Ανάλυση Επίδοσης Υπολογιστικών Συστημάτων

Αναλυτικά μοντέλα, προσομοίωση, μετρήσεις

Ανδρέας-Γεώργιος Σταφυλοπάτης

Γεώργιος Σιόλας

Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράμματα και Βοηθήματα
www.kallipos.gr

HEALLINK
Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Ελληνικά Ακαδημαϊκά Ηλεκτρονικά
Συγγράμματα και Βοηθήματα
www.kallipos.gr

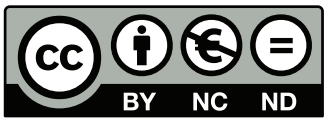
Συγγραφέας: Ανδρέας-Γεώργιος Σταφυλοπάτης

Συν-συγγραφέας: Γεώργιος Σιόλας

ΑΝΑΛΥΣΗ ΕΠΙΔΟΣΗΣ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Αναλυτικά μοντέλα, προσομοίωση, μετρήσεις

Κριτική Ανάγνωση: Ελένη Καρατζά

Τεχνική επιμέλεια: Γεώργιος Σιόλας
Γλωσσική επιμέλεια: Βασίλειος Παππάς



Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών
Εθνικό Μετσόβιο Πολυτεχνείο
Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου
www.kallipos.gr

ISBN: 978-960-603-367-4

Εξώφυλλο: απόσπασμα από τον πίνακα του ζωγράφου Γρηγόρη Φιλιππάτου ‘À Bicyclette’.

ΑΝΑΛΥΣΗ ΕΠΙΔΟΣΗΣ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Αναλυτικά μοντέλα, προσομοίωση, μετρήσεις

Α.-Γ. ΣΤΑΦΥΛΟΠΑΤΗΣ
Καθηγητής Ε.Μ.Π.
Γ. ΣΙΟΛΑΣ
Ε.ΔΙ.Π. Ε.Μ.Π.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Αθήνα 2015

Περιεχόμενα

Συντομεύσεις – Ακρωνύμια

Ε/Ε	Είσοδος/Έξοδος
ΚΜΕ	Κεντρική Μονάδα Επεξεργασίας
ΣΚΠ	Συνάρτηση Κατανομής Πιθανότητας
σμπ	συνάρτηση μάζας πιθανότητας
σππ	συνάρτηση πυκνότητας πιθανότητας
ANOVA	Analysis of Variance
BCMP	Baskett Chandy Muntz Palacios
CPU	Central Procesing Unit
CCDF	Complementary Cumulative Distribution Function
FCFS	First Come-First Served
FIFO	First In-First Out
FB	Foreground-Background
GI	General Independent
GPSS	General Purpose Simulation System
HB	Higher Better
I/O	Input/Output
IS	Infinite Servers
JMT	Java Modelling Tools
K-S	Kolmogorov-Smirnov
LCFS	Last Come-First Served
LCFSPR	Last Come-First Served-Preemptive-Resume
LIFO	Last In-First Out
LD	load-dependent
LI	load-independent
LB	Lower Better
MMPP	Markov-modulated Poisson process
MVA	Mean Value Analysis
MFLOPS	Million Floating-point Operations per Second
MIPS	Million Instructions per Second
NS	Network Simulator
NB	Nominal Better
pps	Packets per Second
PEPSY	Performance Evaluation and Prediction System
PH	phase-type distribution
P-K	Pollaczek-Khinchine
pdf	probability density function
PDF	Probability Distribution Function
pmf	probability mass function
PS	Processor Sharing
QoS	Quality of Service
RR	Round-Robin
SLA	Service Level Agreement
SPT	Shortest-Processing-Time-first
SPEC	Standard Performance Evaluation Corporation
SWIC	stepwise inclusion of classes
SSE	Sum of Squared Errors
SSR	Sum of Squares explained by Regression
SST	Total Sum of Squares
TPC	Transaction Processing Performance Council

Κεφάλαιο 1

Εισαγωγή στην Ανάλυση Επίδοσης

Σύνοψη
Στο πρώτο κεφάλαιο παρουσιάζονται οι βασικές έννοιες που αφορούν την ποσοτική ανάλυση της Επίδοσης και της Ποιότητας Υπηρεσιών (Quality of Service) των υπολογιστικών συστημάτων. Ορίζεται η έννοια του συστήματος και εξετάζονται οι στόχοι, τα οφέλη και οι δυσκολίες μιας μελέτης επίδοσης (performance engineering). Περιγράφονται οι γενικές μεθοδολογίες ανάλυσης, δηλαδή τα αναλυτικά μοντέλα, η προσομοίωση και οι τεχνικές μετρήσεων, και συζητούνται τα κυριότερα χαρακτηριστικά τους. Επίσης, εισάγονται οι έννοιες του φορτίου και των δεικτών επίδοσης, που αποτελούν την είσοδο και την έξοδο, αντίστοιχα, σε κάθε μοντέλο επίδοσης. Εξετάζονται διάφοροι τύποι δεικτών επίδοσης (μέτρα χρόνου, χώρου και ταχύτητας) και αναλύονται τα χαρακτηριστικά τους. Γίνεται ιδιαίτερη αναφορά στα χαρακτηριστικά της επίδοσης συστημάτων βασισμένων στον Ιστό (Web-based).

Καμία άλλη βιομηχανία δεν έχει εξελιχθεί με τον απίστευτο ρυθμό που ακολουθεί η εξέλιξη της βιομηχανίας των υπολογιστών, ούτε έχει διεισδύσει σε ανάλογο εύρος και βάθος στη ζωή των ανθρώπων. Σε αυτή τη διαδικασία αλλαγής και εξάπλωσης, σταθερή είναι η ανάγκη τόσο των κατασκευαστών όσο και των χρηστών για κατανόηση της επίδοσης των υπολογιστικών συσκευών. Όμως, αν και η τεχνολογία των υπολογιστών οδηγεί τα σύγχρονα συστήματα σε ασύλληπτες ταχύτητες, δεν υπάρχουν αυστηρά καθορισμένοι και γενικά αποδεκτοί τρόποι μέτρησης και αποτίμησης των επιδόσεών τους. Ενώ ο υπολογισμός του κόστους ενός συστήματος κατά κανόνα γίνεται με άμεσο τρόπο, το ζήτημα της επίδοσης αντιμετωπίζεται συνήθως με ad hoc τεχνικές και εργαλεία. Πίσω από αυτά, όμως, κρύβονται κοινές θεμελιώδεις αρχές, έννοιες και υποθέσεις, η κατανόηση των οποίων αποτελεί προϋπόθεση για την επιτυχή πρακτική εφαρμογή τους [1, 7, 2, 4, 17, 8, 9, 12, 10, 15, 6, 3, 13, 14, 16, 18, 5].

1.1 Μεθοδολογία Ανάλυσης και Ποιότητα Υπηρεσιών

Η επίδοση είναι ένα βασικό κριτήριο κατά τη σχεδίαση, την προμήθεια ή τη χρήση ενός υπολογιστικού συστήματος: ο στόχος είναι συνήθως η εξασφάλιση της μέγιστης επίδοσης για δεδομένο κόστος. Η επίτευξη του στόχου αυτού μπορεί να βασιστεί σε ένα σύνολο μεθόδων που επιτρέπουν την επίλυση μιας πληθώρας προβλημάτων. Ως παραδείγματα τέτοιων προβλημάτων μπορούν να αναφερθούν:

- προσδιορισμός των απαιτήσεων επίδοσης,
- χαρακτηρισμός του φορτίου (workload characterization),
- αποτίμηση εναλλακτικών λύσεων,
- σύγκριση δύο ή περισσότερων συστημάτων (σχετική επίδοση),
- προσδιορισμός της επίδρασης ενός χαρακτηριστικού,
- βελτιστοποίηση προγράμματος,

- εύρεση της βέλτιστης τιμής μιας παραμέτρου (ρύθμιση συστήματος – tuning),
- εύρεση της στένωσης για την επίδοση ενός συστήματος (bottleneck identification),
- προσδιορισμός του αριθμού και του μεγέθους των συστατικών ενός συστήματος (capacity planning),
- πρόβλεψη της επίδοσης σε μελλοντικά φορτία (forecasting).

Ο όρος *σύστημα* νοείται εδώ ως μία συλλογή συνιστωσών υλικού ή λογισμικού: θα μπορούσε να είναι μία *Κεντρική Μονάδα Επεξεργασίας – ΚΜΕ* (Central Processing Unit – CPU), μια μονάδα *Εισόδου/Εξόδου – Ε/Ε* (Input/Output – I/O), ένα σύστημα διαχείρισης βάσεων δεδομένων ή ένα δίκτυο υπολογιστών.

Η ανάλυση της επίδοσης ενός συστήματος βασίζεται στην επιλογή της κατάλληλης μεθόδου, του κατάλληλου *φορτίου* (workload) και των κατάλληλων *δεικτών επίδοσης* (performance metrics) [11]. Το φορτίο εκφράζει τις αιτήσεις των χρηστών για εξυπηρέτηση από το σύστημα. Οι δείκτες επίδοσης επιτρέπουν τον ποσοτικό προσδιορισμό της επίδοσης του συστήματος.

Ποιότητα Υπηρεσιών Ένα υπολογιστικό σύστημα πρέπει να ανταποκρίνεται στις ανάγκες των χρηστών, όπως αυτές καθορίζονται μέσω ποσοτικών παραμέτρων για κάθε τύπο φορτίου που εξυπηρετείται από το σύστημα. Συνεπώς, το σύστημα θα πρέπει να διαθέτει τα τεχνικά χαρακτηριστικά που θα μπορούν να υποστηρίξουν το φορτίο, και αυτό θα τεκμηριώνεται με ποσοτικό τρόπο (θα μπορεί να μετρηθεί και να αποτελεί εκ των προτέρων αντικείμενο εγγύησης). Η αντιστοίχιση αυτή ανάμεσα στις ανάγκες και την ικανοποίησή τους αποτελεί προϋπόθεση για την ορθή παροχή υπηρεσιών εκ μέρους του συστήματος και εκφράζει την *Ποιότητα Υπηρεσιών* (Quality of Service — QoS), δηλαδή τη συνολική επίδοση του συστήματος όπως την αντιλαμβάνονται οι χρήστες. Βέβαια, υπάρχουν και απαιτήσεις που εκφράζονται ποιοτικά (φιλικότητα, ευκολία χρήσης), αλλά μια ακριβής τεχνική μελέτη επίδοσης θα πρέπει να εστιάζει σε μετρήσιμα χαρακτηριστικά.

Μελέτη επίδοσης Το ζήτημα της επίδοσης πρέπει να λαμβάνεται υπόψη σε όλες τις φάσεις του κύκλου ζωής ενός συστήματος και όχι μόνο στα τελικά στάδια της ανάπτυξης και στη φάση της λειτουργίας, όπως συμβαίνει συχνά. Η διαπίστωση αυτή καθορίζει τις σύγχρονες τάσεις στον σχεδιασμό και την ανάπτυξη συστημάτων. Η γενική μεθοδολογία που προκύπτει αναφέρεται ως *τεχνολογία επίδοσης* (performance engineering) και μπορεί να θεωρηθεί ως ένα σύνολο μεθόδων που υποστηρίζουν την ανάπτυξη συστημάτων σε όλες τις φάσεις του κύκλου ζωής. Στις αρχικές φάσεις της ανάπτυξης, στις οποίες υπάρχουν λίγα δεδομένα για το σύστημα και το φορτίο, η ανάλυση της επίδοσης είναι σχετικά αδρομερής. Εν συνεχεία, με την πρόοδο του έργου, ο όγκος των διαθέσιμων πληροφοριών αυξάνει και επιτρέπει την ανάλυση σε μεγαλύτερο βαθμό λεπτομέρειας.

Ανεξάρτητα από τις επιμέρους ακολουθούμενες τεχνικές ανάλυσης, μια συστηματική μελέτη επίδοσης θα πρέπει να περιλαμβάνει τα ακόλουθα βήματα:

- καθορισμός των στόχων της μελέτης και των ορίων του συστήματος,
- καταγραφή των υπηρεσιών που παρέχονται από το σύστημα και των δυνατών αποτελεσμάτων,
- επιλογή φορτίου,
- επιλογή δεικτών επίδοσης,
- καταγραφή των παραμέτρων του συστήματος και του φορτίου,
- επιλογή τεχνικής για την ανάλυση της επίδοσης,
- σχεδίαση πειραμάτων,
- ανάλυση και ερμηνεία των δεδομένων,
- παρουσίαση των αποτελεσμάτων της μελέτης.

1.2 Επιλογή Τεχνικής και Μοντελοποίηση

Οι τεχνικές για την ανάλυση επίδοσης υπολογιστικών συστημάτων μπορούν να ενταχθούν σε τρεις γενικές κατηγορίες: *αναλυτικά μοντέλα*, *προσομοίωση* και *μετρήσεις*. Συνδυασμός τεχνικών μπορεί επίσης να χρησιμοποιηθεί.

Οι πραγματικές μετρήσεις συνήθως παρέχουν τα καλύτερα αποτελέσματα, καθόσον δεν απαιτούν απλοποιητικές υποθέσεις. Για τον λόγο αυτό, τα αποτελέσματα των μετρήσεων έχουν γενικά υψηλή απήχηση και αξιοπιστία. Οι τεχνικές μετρήσεων, όμως, υστερούν ως προς το κόστος, την ευελιξία και την πληρότητα (σε ένα πραγματικό σύστημα είναι δύσκολο έως αδύνατο να μελετηθεί η επίδοση για όλους τους συνδυασμούς τιμών των παραμέτρων). Οι άλλες δύο κατηγορίες μεθόδων δεν αντιμετωπίζουν αυτό το πρόβλημα, διότι βασίζονται στη *μοντελοποίηση* (modelling) του υπό μελέτη συστήματος. Ορισμένες τεχνικές μετρήσεων αναφέρονται και ως *εμπειρικές τεχνικές*, διότι δεν στηρίζονται σε ισχυρή μαθηματική θεμελίωση. Ως ακραίο παράδειγμα μπορούν να αναφερθούν διαισθητικές προσεγγίσεις (βασισμένες στην εμπειρία ή το ένστικτο του ειδικού), οι οποίες δίνουν γρήγορα και —συνήθως— ανακριβή αποτελέσματα.

Ένα μοντέλο αποτελεί μια γενικευμένη αναπαράσταση ενός πραγματικού συστήματος. Η μοντελοποίηση πραγματοποιεί ένα είδος αφαίρεσης, η οποία περιλαμβάνει τα σημαντικά χαρακτηριστικά του συστήματος αφαιρώντας λεπτομέρειες που δεν κρίνονται ουσιώδεις για την ανάλυση επίδοσης. Προκειμένου να επιτευχθεί το επιθυμητό επίπεδο λεπτομέρειας, απαιτείται συνήθως η υιοθέτηση απλοποιητικών υποθέσεων στον ορισμό του μοντέλου. Όσο μεγαλύτερος ο βαθμός αφαίρεσης, τόσο το μοντέλο απέχει από την πραγματικότητα. Κατά μια έννοια, το μοντέλο παρέχει ένα «εργαστηριακό» περιβάλλον για τη μελέτη του συστήματος —ακόμη και πριν το σύστημα αυτό υπάρξει. Εξάλλου, ακόμη και στην περίπτωση υπάρχοντος συστήματος, δεν είναι εύκολο να υλοποιηθούν φυσικές τροποποιήσεις και δοκιμές εναλλακτικών επιλογών. Από την άλλη πλευρά, το «σύστημα» είναι ένα κομμάτι του πραγματικού κόσμου, το οποίο περιγράφεται μέσα από την υποκειμενική οπτική της μοντελοποίησης. Η διαδικασία κατασκευής του μοντέλου ορίζει τους επιδιωκόμενους στόχους και μέσω αυτών προσδιορίζει τα όρια του συστήματος και τον βαθμό λεπτομέρειας του μοντέλου. Μια βασική αρχή είναι ότι το μοντέλο δεν πρέπει να είναι περισσότερο πολύπλοκο απ' ό,τι απαιτούν οι στόχοι του.

Τα αναλυτικά μοντέλα παρέχουν μια μαθηματική περιγραφή του συστήματος, η οποία –γενικά– χαρακτηρίζεται από ευελιξία και αποδοτικότητα. Προκειμένου να είναι εφικτή η μαθηματική επίλυσή τους, τα αναλυτικά μοντέλα χαρακτηρίζονται από μικρό βαθμό λεπτομέρειας σε σύγκριση με τις άλλες δύο τεχνικές. Η λύση παρέχεται είτε μέσω μαθηματικών τύπων είτε μέσω υπολογιστικών αλγορίθμων. Τα αναλυτικά μοντέλα που χρησιμοποιούνται στην πράξη βασίζονται κυρίως στη *θεωρία αναμονής* (queueing theory). Τα αποτελέσματα της επίλυσης ενός αναλυτικού μοντέλου δεν είναι τόσο ακριβή, αλλά επιτρέπουν την ταχεία κατανόηση της συμπεριφοράς του συστήματος, και μπορούν να κατευθύνουν τους στόχους λεπτομερέστερων μελετών με χρήση προσομοίωσης ή μετρήσεων.

Σύμφωνα με τη μέθοδο της προσομοίωσης, το υπό μελέτη σύστημα παριστάνεται μέσω ενός προγράμματος υπολογιστή. Το πρόγραμμα παρέχει ευελιξία στην περιγραφή της δυναμικής συμπεριφοράς του συστήματος και δυνατότητα για σημαντικό βαθμό λεπτομέρειας ανάλογα και με τα χαρακτηριστικά και την πολυπλοκότητα του υπό μελέτη συστήματος. Η υψηλή πολυπλοκότητα του μοντέλου μπορεί να οδηγήσει σε υπερβολική αύξηση του προγράμματος με επακόλουθες δυσχέρειες. Το κόστος της προσομοίωσης περιλαμβάνει τόσο το κόστος ανάπτυξης του προγράμματος όσο και το κόστος εκτέλεσης, και –κατά περίπτωση– μπορεί να είναι υψηλό.

Θα πρέπει να σημειωθεί ότι υπάρχουν περιπτώσεις συστημάτων των οποίων η συμπεριφορά δεν μπορεί να περιγραφεί αποδοτικά —ή καθόλου— από ακριβή αναλυτικά μοντέλα. Εξάλλου, η προσφυγή στην προσομοίωση μπορεί να είναι ασύμφορη. Σε τέτοιες περιπτώσεις, μια αποτελεσματική επιλογή είναι η ανάπτυξη προσεγγίσεων με βάση τα αναλυτικά μοντέλα. Οι προσεγγίσεις είναι συνήθως αποδοτικές από υπολογιστική άποψη, αλλά μπορεί να μειονεκτούν σε ακρίβεια. Εντούτοις, έστω και μειωμένη ακρίβεια μπορεί να είναι ικανοποιητική για τους στόχους της μελέτης επίδοσης.

Τα σημαντικότερα κριτήρια που πρέπει να ληφθούν υπόψη κατά την επιλογή της κατάλληλης τεχνικής για μια μελέτη επίδοσης συνοψίζονται στον Πίνακα 1.1.

Η διαδικασία επίλυσης του μοντέλου δέχεται ως είσοδο ένα σύνολο δεδομένων που αποτελούν παρα-

Κριτήριο	Αναλυτικό Μοντέλο	Προσομοίωση	Μετρήσεις
1. Στάδιο κύκλου ζωής	Οποιοδήποτε	Οποιοδήποτε	Υπάρχον σύστημα
2. Απαιτούμενος χρόνος	Μικρός	Μέτριος	Ποικίλλει
3. Απαιτούμενα εργαλεία	Θεωρία αναμονής	Γλώσσες προγραμματισμού	Οργανα μέτρησης
4. Ακρίβεια	Χαμηλή	Μέτρια	Ποικίλλει
5. Αποτίμηση εναλλακτικών λύσεων	Εύκολη	Μέτρια	Δύσκολη
6. Κόστος	Χαμηλό	Μέτριο	Υψηλό
7. Απήχηση	Χαμηλή	Μέτρια	Υψηλή

Πίνακας 1.1: *Κριτήρια επιλογής.*

μέτρους του συστήματος και του φορτίου. Οι πρώτες αφορούν τόσο χαρακτηριστικά του συστήματος ως συνόλου (π.χ. πρωτόκολλα, περιορισμοί χωρητικότητας) όσο και ιδιότητες των επιμέρους συνιστωσών (πόρων) του συστήματος (π.χ. ταχύτητα ΚΜΕ, χρόνος αναζήτησης στο δίσκο). Οι δεύτερες διακρίνονται σε παραμέτρους έντασης φορτίου (αριθμός χρηστών ή ρυθμός άφιξης αιτημάτων προς το σύστημα) και παραμέτρους απαίτησης εξυπηρέτησης (απαιτούμενος χρόνος εξυπηρέτησης σε κάθε συστατικό του συστήματος). Οι έξοδοι του μοντέλου είναι δείκτες επίδοσης (Σχήμα 1.1).

1.3 Δείκτες Επίδοσης

Σε κάθε μελέτη επίδοσης θα πρέπει να επιλεγεί ένα σύνολο δεικτών που αντιπροσωπεύουν τα αποτελέσματα της ανάλυσης επίδοσης. Ένας *δείκτης* ή *μετρική επίδοσης* (performance index/metric) είναι μια ποσοτική μεταβλητή που αντιστοιχεί σε χαρακτηριστικό του συστήματος σχετικό με την επίδοση [6, 11]. Η αντιστοίχιση αυτή βασίζεται κατά κύριο λόγο στην καταγραφή των υπηρεσιών που παρέχει το σύστημα.

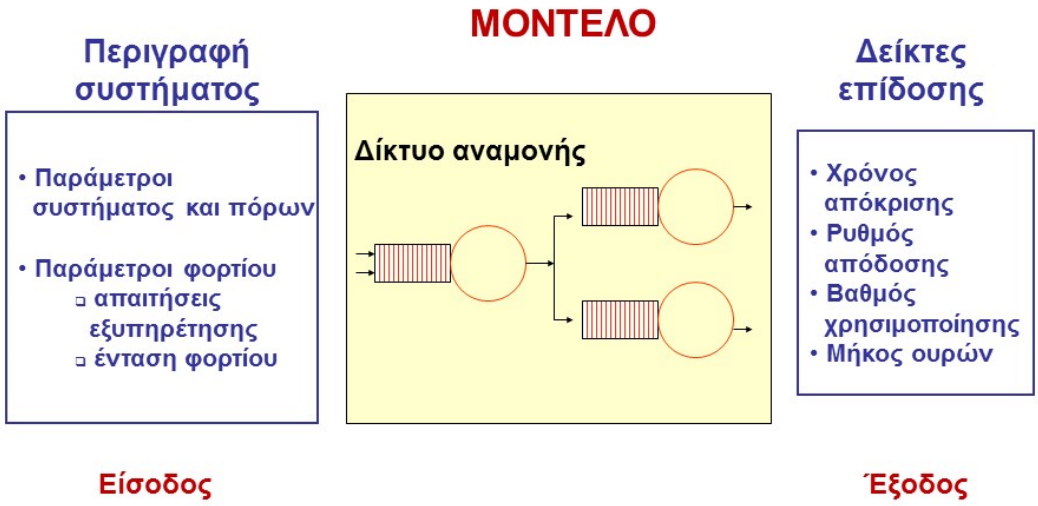
Βασικοί τύποι Οι τιμές των χαρακτηριστικών που περιγράφουν την επίδοση προκύπτουν συνήθως —αμέσως ή εμμέσως— από ποσότητες βασικών τύπων, όπως:

- Μετρητής (πόσες φορές συμβαίνει ένα γεγονός). Π.χ., ο αριθμός των προσπελάσεων στο δίσκο για ανάγνωση/εγγραφή που πραγματοποιούνται κατά την εκτέλεση ενός προγράμματος.
- Διάρκεια ενός χρονικού διαστήματος. Π.χ., η χρονική διάρκεια μιας προσπέλασης στο δίσκο.
- Ποσότητα ενός μεγέθους/μιας παραμέτρου. Π.χ., ο αριθμός των bits που γράφονται στο δίσκο στη διάρκεια μιας προσπέλασης.

Συνήθως, ένας μετρητής κανονικοποιείται ως προς μια βάση χρόνου για να δώσει μια μετρική που ονομάζεται *ρυθμός* (rate) ή *διέλευση* (throughput), π.χ. ο ρυθμός άφιξης εργασιών στο σύστημα ή ο ρυθμός αναχώρησης εργασιών. Άλλοι παράγωγοι τύποι μετρικών, χρήσιμοι για τη σύγκριση συστημάτων, είναι η *επιτάχυνση* (speedup) και η *σχετική μεταβολή* (relative change). Η επιτάχυνση εκφράζει πόσες φορές ταχύτερο είναι ένα σύστημα σε σχέση με ένα άλλο που λαμβάνεται ως βάση αναφοράς. Ανάλογα, η σχετική μεταβολή εκφράζει την επίδοση ενός συστήματος ως ποσοστιαία μεταβολή σε σχέση με την επίδοση ενός συστήματος αναφοράς.

Δείκτες και παροχή υπηρεσιών Γενικά, όταν υποβάλλεται στο σύστημα μία αίτηση εξυπηρέτησης, τα δυνατά αποτελέσματα μπορεί να είναι τριών ειδών: (1) *σωστή εξυπηρέτηση*, (2) *εσφαλμένη εξυπηρέτηση* ή (3) *αδυναμία εξυπηρέτησης*.

Στην περίπτωση (1), η επίδοση του συστήματος σχετίζεται κυρίως με την ταχύτητα και χαρακτηρίζεται από τον χρόνο που απαιτήθηκε (*χρόνος απόκρισης* – response time), το ρυθμό με τον οποίο εκτελείται η εξυπηρέτηση (*ρυθμός απόδοσης* – throughput) και τους πόρους του συστήματος που καταναλώθηκαν



Σχήμα 1.1: *Μοντέλο συστήματος.*

(*βαθμός χρησιμοποίησης* – utilization). Ο χρόνος απόκρισης είναι το διάστημα ανάμεσα στην αίτηση του χρήστη για εξυπηρέτηση και την απόκριση του συστήματος. Ο ορισμός αυτός είναι γενικός και εξειδικεύεται ανάλογα με τα χαρακτηριστικά του συστήματος (π.χ. διαλογικό, batch). Ο ρυθμός απόδοσης εκφράζεται, ανάλογα με το σύστημα, σε εργασίες ανά δευτερόλεπτο (jobs per sec), συναλλαγές ανά δευτερόλεπτο (transactions per sec), πακέτα ανά δευτερόλεπτο (pps) ή bits ανά δευτερόλεπτο (bps) για δίκτυα, εκατομμύρια εντολές ανά δευτερόλεπτο (MIPS) ή εκατομμύρια πράξεις κινητής υποδιαστολής ανά δευτερόλεπτο (MFLOPS) για υπολογιστικές μονάδες. Ο βαθμός χρησιμοποίησης ενός πόρου (resource) είναι το ποσοστό του χρόνου κατά το οποίο ο πόρος είναι απασχολημένος. Ο πόρος με τον υψηλότερο βαθμό χρησιμοποίησης είναι η *στένωση* (bottleneck) του συστήματος.

Στην περίπτωση (2), η *αξιοπιστία* (reliability)του συστήματος χαρακτηρίζεται από την *πιθανότητα σφάλματος* και το *χρόνο μεταξύ σφαλμάτων* για κάθε κατηγορία σφάλματος που μπορεί να συμβεί.

Στην περίπτωση (3), το σύστημα δεν μπορεί να εκτελέσει την εξυπηρέτηση, διότι κάποια βλάβη το έχει θέσει εκτός λειτουργίας. Η *διαθεσιμότητα* (availability) του συστήματος χαρακτηρίζεται από τη *διάρκεια της βλάβης* και τον *χρόνο μεταξύ βλαβών* για κάθε κατηγορία βλάβης που μπορεί να συμβεί.

Η διαχείριση ενός συστήματος απαιτεί την εξασφάλιση των προϋποθέσεων για επίτευξη των στόχων επίδοσης. Για παράδειγμα, όταν το σύστημα πραγματοποιεί "online" συναλλαγές, ο στόχος επίδοσης εκφράζεται ως άνω φράγμα (επιτρεπτή τιμή) του μέσου χρόνου απόκρισης. Αντίστοιχα, για φορτίο τύπου "batch", ο στόχος ορίζεται ως κάτω φράγμα του ρυθμού απόδοσης. Τέτοιου είδους απαιτήσεις αντιπροσωπεύουν μια συμφωνία ανάμεσα στον πάροχο και στον χρήστη μια υπηρεσίας και αποτελούν τη *Συμφωνία Επιπέδου Υπηρεσιών* (Service Level Agreement — SLA). Είναι φανερό ότι οι έννοιες της Ποιότητας Υπηρεσιών (QoS) και της Συμφωνίας Επιπέδου Υπηρεσιών σχετίζονται στενά, και συχνά οι δύο όροι χρησιμοποιούνται αδιακρίτως.

Ιδιότητες δεικτών Υπάρχουν διάφοροι δείκτες επίδοσης που χρησιμοποιούνται ευρέως. Η εμπειρία δείχνει, όμως, ότι δεν είναι όλοι πάντοτε «καλοί», με την έννοια ότι σε ορισμένες περιπτώσεις η χρήση ενός δείκτη μπορεί να οδηγήσει σε εσφαλμένα ή παραπλανητικά συμπεράσματα. Η κατανόηση των χαρακτηριστικών ιδιοτήτων των δεικτών βοηθάει στην επιλογή των πλέον κατάλληλων για κάθε περίπτωση. Οι επιθυμητές ιδιότητες των δεικτών μπορούν να ελεγχθούν μέσω των παρακάτω κριτηρίων [11].

- *Γραμμικότητα.* Οι άνθρωποι τείνουν να σκέπτονται «γραμμικά», επομένως, η τιμή ενός δείκτη θα πρέπει να είναι ευθέως ανάλογη της πραγματικής επίδοσης. Π.χ., αν ο δείκτης διπλασιαστεί, θα περιμέναμε οι χρόνοι εκτέλεσης να μειωθούν στο μισό. Η γραμμικότητα καθιστά έναν δείκτη διαισθητικά εύληπτο, σε αντίθεση με δείκτες που δεν ικανοποιούν αυτήν την ιδιότητα. Παράδειγμα μη γραμμικότητας αποτελούν οι λογαριθμικοί δείκτες, π.χ. η κλίμακα dB για την ένταση του ήχου.

- *Αξιοπιστία.* Ένας δείκτης θεωρείται αξιόπιστος αν το σύστημα A υπερέχει πάντα σε επίδοση του συστήματος B, όταν οι αντίστοιχες τιμές του δείκτη υποδεικνύουν ότι το A θα πρέπει να υπερέχει του B. Αν και —εκ πρώτης όψεως— η απαίτηση αυτή φαίνεται προφανής και περιττή, διάφοροι γνωστοί δείκτες επίδοσης δεν την ικανοποιούν στην πράξη, όπως, π.χ., ο δείκτης MIPS (εκατομμύρια εντολές ανά δευτερόλεπτο). Συχνά, ένας επεξεργαστής A, με υψηλότερη τιμή MIPS από έναν επεξεργαστή B, χρειάζεται περισσότερο χρόνο από τον B για την εκτέλεση ενός συγκεκριμένου προγράμματος.

- *Επαναληψιμότητα.* Ο δείκτης είναι ντετερμινιστικός, δηλαδή σε κάθε επανάληψη ενός πειράματος μετριέται η ίδια τιμή του δείκτη.

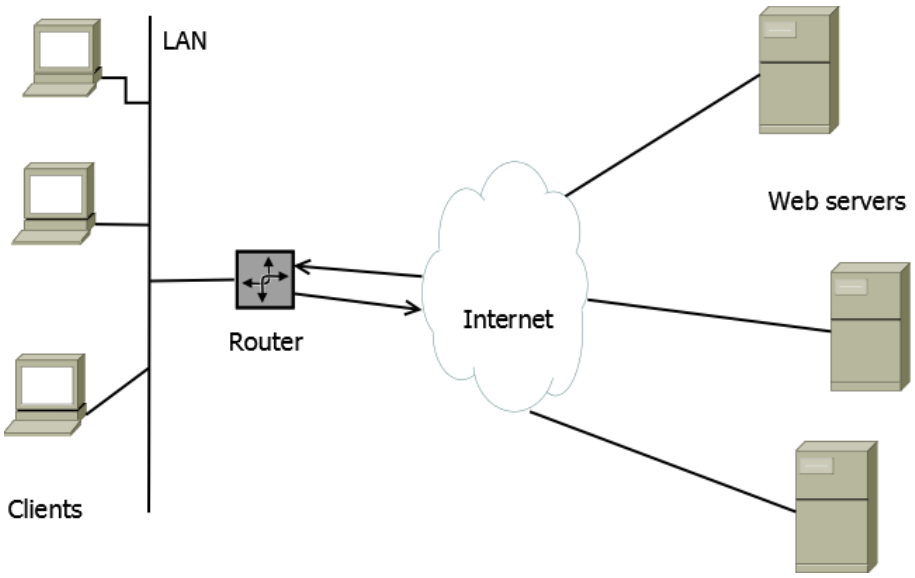
- *Ευχέρεια μέτρησης.* Όταν ένας δείκτης μετριέται δύσκολα (άμεσα ή έμμεσα), η μέτρησή του είναι επιρρεπής σε σφάλματα.

- *Συνέπεια.* Ένας δείκτης είναι συνεπής όταν ορίζεται με τον ίδιο τρόπο και έχει τις ίδιες μονάδες μέτρησης σε διαφορετικά συστήματα. Και αυτή η απαίτηση, αν και προφανής, συχνά παραβιάζεται, π.χ. MIPS.

- *Ανεξαρτησία.* Πολλές φορές, οι κατασκευαστές υπολογιστικών συστημάτων σχεδιάζουν τα συστήματά τους με έμφαση στη βελτιστοποίηση της τιμής ενός συγκεκριμένου δείκτη, αλλοιώνοντας τη σημασία του δείκτη προς όφελός τους. Επομένως, το περιεχόμενο των δεικτών πρέπει να είναι ανεξάρτητο από ιδιαίτερα χαρακτηριστικά των συστημάτων.

Λειτουργική κατηγοριοποίηση Οι περισσότεροι δείκτες επίδοσης εμπίπτουν σε δύο γενικές κατηγορίες: *με προσανατολισμό στο σύστημα* (system-oriented) και *με προσανατολισμό στον χρήστη* (user-oriented). Η πρώτη κατηγορία αντιπροσωπεύει την οπτική γωνία του διαχειριστή του συστήματος και αφορά δείκτες που σχετίζονται με τον βαθμό χρησιμοποίησης των πόρων, καθώς και με το μήκος των ουρών αναμονής στα διάφορα συστατικά του συστήματος. Οι τιμές των δεικτών αυτών συνδέονται άμεσα με τη ρύθμιση καλής λειτουργίας. Ο βαθμός χρησιμοποίησης δεν πρέπει να είναι ούτε υπερβολικά χαμηλός (κακή χρήση των πόρων) ούτε υπερβολικά υψηλός (κίνδυνος κορεσμού). Το μήκος ουρών επηρεάζει τις καθυστερήσεις και σχετίζεται με το μέγεθος των ενταμιευτών (buffers). Η δεύτερη κατηγορία περιγράφει την αντίληψη του χρήστη και περιλαμβάνει δείκτες που σχετίζονται με τον χρόνο απόκρισης και τον ρυθμό απόδοσης. Είναι φανερό ότι δείκτες αυτού του τύπου μπορούν να οριστούν με διάφορους τρόπους ανάλογα με τη φύση των παρεχόμενων υπηρεσιών και τα χαρακτηριστικά του συστήματος. Ο ρυθμός απόδοσης, πάντως, μπορεί να θεωρηθεί και δείκτης με προσανατολισμό στο σύστημα.

Τέλος, ανάλογα με την ωφελιμότητα που εκφράζει ένας δείκτης επίδοσης, μπορεί να είναι προτιμότερες οι υψηλές τιμές του (Higher Better – HB) ή οι χαμηλές τιμές του (Lower Better – LB) ή κάποια ενδιάμεση (ονομαστική) τιμή του (Nominal Best – NB). Παραδείγματα των τριών αυτών κατηγοριών είναι ο ρυθμός απόδοσης, ο χρόνος απόκρισης και ο βαθμός χρησιμοποίησης, αντίστοιχα.



Σχήμα 1.2: *Μοντελοποίηση Ιστού.*

1.4 Υπηρεσίες Παγκόσμιου Ιστού

Ο Ιστός εξελίσσεται ταχύτατα ενσωματώνοντας διαρκώς νέες συνιστώσες, εφαρμογές και υπηρεσίες. Αποτέλεσμα είναι η ραγδαία αύξηση της κυκλοφορίας στο Διαδίκτυο και του αριθμού των αιτήσεων εξυπηρέτησης που διεκπεραιώνονται καθημερινά. Ένα από τα σημαντικότερα προβλήματα της διαχείρισης ενός Ιστότοπου είναι ο προσδιορισμός του κατάλληλου μεγέθους του πληροφοριακού συστήματος, ώστε να παρέχεται στους χρήστες η απαιτούμενη Ποιότητα Υπηρεσιών. Η κάλυψη αυτής της ανάγκης προϋποθέτει τη δυνατότητα εποπτείας της επίδοσης των ιστότοπων και των προσφερόμενων *υπηρεσιών Ιστού* (Web Services). Οι απαιτήσεις λαμβάνουν ακόμη μεγαλύτερες διαστάσεις με τη χρήση των ασύρματων τεχνολογιών και των ηλεκτρονικών συσκευών χειρός. Τα σύγχρονα συστήματα του Ιστού παρουσιάζουν διάφορα ενδιαφέροντα χαρακτηριστικά, όπως δυναμική δημιουργία σελίδων, προσωποποιημένο περιεχόμενο, σύνδεση με βάσεις δεδομένων κλπ.

Οι περισσότεροι Ιστότοποι παρέχουν Υπηρεσίες Ιστού που εκτείνονται σε διάφορες κατηγορίες ([13, 14]) όπως:

- πληροφοριακές (εφημερίδες, βιβλία, διαφημήσεις),
- διαλογικές (φόρμες εγγραφής, παιχνίδια),
- συναλλακτικές (ηλεκτρονική αγορά, τραπεζικές συναλλαγές),
- ροή εργασίας (σχεδιασμός και χρονοδρομολόγηση),
- συνεργατικές (κατανεμημένη συγγραφή και σχεδίαση),
- κοινότητες και κοινωνικά δίκτυα (ομάδες συζήτησης, συστήματα συστάσεων, αγορές, πληστηριασμοί),
- πύλες του Ιστού (Web portals, — ηλεκτρονικές υπηρεσίες, μηχανές αναζήτησης).

Πέραν του Παγκόσμιου Ιστού, οι τεχνολογίες του Διαδικτύου χρησιμοποιούνται για την ανάπτυξη *εσωτερικών εταιρικών δικτύων* (intranets), τα οποία χρησιμοποιούνται κυρίως για την παροχή εκπαιδευτικού περιεχομένου στους χρήστες μέσω φυλλομετρητών Ιστού (Web browsers). Θα πρέπει να σημειωθεί ότι τόσο ο Ιστός όσο και οι εφαρμογές τύπου intranet μπορούν να θεωρηθούν ειδικές περιπτώσεις του μοντέλου *πελάτη–εξυπηρετητή* (client–server), σύμφωνα με το οποίο το υπολογιστικό έργο μιας εφαρμογής διασπάται σε δύο επιμέρους διεργασίες που εκτελούνται σε διαφορετικές διασυνδεδεμένες μηχανές. Οι ποσοτικές

τεχνικές που αναφέρθηκαν νωρίτερα μπορούν να χρησιμοποιηθούν για την ανάλυση της επίδοσης υπηρεσιών στο σύνολο των καταστάσεων που σχετίζονται με Ιστότοπους, περιβάλλοντα πελάτη–εξυπηρετητή, εταιρικές πύλες, Παρόχους Υπηρεσιών Διαδικτύου κλπ. (Σχήμα 1.2).

Η Ποιότητα Υπηρεσιών —και κατ΄ επέκταση η Συμφωνία Επιπέδου Υπηρεσιών— έχει ιδιαίτερη σημασία όσον αφορά τη σχέση μεταξύ παρόχου και χρήστη. Το επίπεδο ποιότητας μιας υπηρεσίας περιγράφεται μέσω ενός συνόλου δεικτών, οι οποίοι θα πρέπει να ορίζονται έτσι ώστε να εκφράζουν την οπτική γωνία του χρήστη. Σε περιβάλλον Ιστού, ένας συνδεδεμένος χρήστης συνήθως απαιτεί χαμηλό —και προβλέψιμο— χρόνο απόκρισης, υψηλή διαθεσιμότητα (ιδανικά 24×7) και αξιοπιστία. Η διαθεσιμότητα παριστάνει το ποσοστό του χρόνου κατά το οποίο η υπηρεσία παρέχεται σύμφωνα με τις απαιτήσεις του χρήστη. Η αξιοπιστία ορίζεται ως η πιθανότητα παροχής της υπηρεσίας με ικανοποιητικό τρόπο υπό καθορισμένες συνθήκες φορτίου για δεδομένο χρονικό διάστημα. Η τήρηση της Συμφωνίας Επιπέδου Υπηρεσιών πρέπει να εποπτεύεται, ώστε να εξασφαλίζεται η ομαλή λειτουργία του συστήματος και να εντοπίζονται τυχόν προβλήματα. Ανάλογα με την επίτευξη των στόχων της Ποιότητας Υπηρεσιών, όπως αποτιμάται από την ανάλυση επίδοσης, προγραμματίζεται ενδεχόμενη αύξηση της χωρητικότητας του συστήματος. Γενικά, η τεχνολογική υποδομή θα πρέπει να βρίσκεται πάντα μπροστά από την αυξανόμενη ζήτηση των υπηρεσιών.

Βιβλιογραφία

[1] Ferrari, D., *Computer Systems Performance Evaluation*, Prentice-Hall, 1978.

[2] Ferrari, D., Serazzi, G. and Zeigner, A., *Measurement and Tuning of Computer Systems*, Prentice-Hall, 1983.

[3] Fortier, P.J., and Michel, H.E., *Computer Systems Performance Evaluation and Prediction*, Elsevier Science, 2003.

[4] Gelenbe, E. and Mitrani, I., *Analysis and Synthesis of Computer Systems*, Academic Press, 1980.

[5] Gregg, B., *Systems Performance: Enterprise and the Cloud*, Prentice Hall, 2013.

[6] Jain, R., *The Art of Computer Systems Performance Analysis - Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, 1991.

[7] Kobayashi, H., *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*, Addison–Wesley, 1978.

[8] Lavenberg, S.S.,*Computer Performance Modeling Handbook*, Academic Press, 1983.

[9] Lazowska, E.D., Zahorjan, J., Scott Graham, G. and Sevcik, K.C., *Quantitative System Performance - Computer System Analysis Using Queueing Network Models*, Prentice-Hall, 1984.

[10] Leung, C.H.C., *Quantitative Analysis of Computer Systems*, John Wiley & Sons, 1988.

[11] Lilja, D.J., *Measuring Computer Performance: A Practitioner's Guide*, Cambridge University Press, 2000.

[12] MacNair, E.A. and Sauer, C.H., *Elements of Practical Performance Modeling*, Prentice-Hall, 1985.

[13] Menasce, D.A., and Almeida, V.A.F., *Capacity Planning for Web Services: Metrics, Models and Methods*, Prentice-Hall, 2002.

[14] Menasce, D.A., Almeida, V.A.F., and Dowdy, L.W., *Performance by Design, Computer Capacity Planning by Example*, Prentice-Hall PTR, 2004.

[15] Molloy, M.K., *Fundamentals of Performance Modeling*, Macmillan, 1989.

[16] Obaidat, M.S., and Boudriga, N.A., *Performance Evaluation of Computer and Telecommunication Systems*, Wiley, 2010.

[17] Sauer, C.H. and Chandy, K.M., *Computer Systems Performance Modelling*, Prentice-Hall, 1981.

[18] Wescott, B., *Every Computer Performance Book*, CreateSpace Independent Publishing Platform, 2013.

Κεφάλαιο 2

Τυχαίες Μεταβλητές και Στοχαστικές Διαδικασίες

Σύνοψη
Το κεφάλαιο αυτό παρέχει καταρχάς μια σύντομη υπενθύμιση βασικών εννοιών της θεωρίας πιθανοτήτων (τυχαίες μεταβλητές, κατανομές πιθανότητας, συνόψιση, μέση τιμή, διασπορά). Εν συνεχεία, με βάση τις πιθανότητες ορίζονται οι στοχαστικές διαδικασίες και μελετώνται ιδιαιτέρως η διαδικασία Poisson, οι διαδικασίες γεννήσεων-θανάτων και οι διαδικασίες Markov (συνεχούς και διακριτού χρόνου). Παρουσιάζονται οι βασικές ιδιότητες και τα κυριότερα θεωρήματα για την επίλυση μαρκοβιανών μοντέλων στη μόνιμη κατάσταση και δίνονται παραδείγματα χρήσης τους για την ανάλυση υπολογιστικών συστημάτων.

Τα περισσοτερα φαινόμενα που χαρακτηρίζουν τη λειτουργία των υπολογιστικών συστημάτων μπορούν να περιγραφούν με τη χρήση τυχαίων μεταβλητών. Μπορούμε να αναφέρουμε, για παράδειγμα, τον αριθμό προγραμμάτων σε αναμονή επεξεργασίας, τον απαιτούμενο χρόνο για κάποια λειτουργία εισόδου/εξόδου ή τον χρόνο απόκρισης ενός συστήματος. Στην πραγματικότητα, ενδιαφερόμαστε για τη μεταβολή της συμπεριφοράς των τυχαίων αυτών μεταβλητών στο χρόνο, οπότε καταλήγουμε στη μελέτη μιας οικογένειας τυχαίων μεταβλητών. Μια τέτοια οικογένεια τυχαίων μεταβλητών ονομάζεται *τυχαία ή στοχαστική διαδικασία*. Οι περισσότερες μέθοδοι ανάλυσης και αξιολόγησης υπολογιστικών συστημάτων στηρίζονται στη θεωρία των στοχαστικών διαδικασιών [2, 7, 9, 6, 1, 3, 13, 5].

2.1 Βασικές Έννοιες και Ορισμοί

Πριν προχωρήσουμε σε ορισμούς που αφορούν τις στοχαστικές διαδικασίες, θα υπενθυμίσουμε επιγραμματικά ορισμένες (γνωστές) βασικές έννοιες από τη θεωρία των πιθανοτήτων.

Τυχαία μεταβλητή. Θεωρούμε έναν χώρο πιθανότητας $(\Omega, \mathcal{E}, P)$, αποτελούμενο από τον δειγματικό χώρο Ω, ένα σύνολο ενδεχομένων $\mathcal{E}$ και ένα μέτρο πιθανότητας P. Μια τυχαία μεταβλητή X είναι μια συνάρτηση $X : \Omega \to R$, η οποία αναθέτει μια πραγματική τιμή $X(\omega)$ σε κάθε σημείο ω του δειγματικού χώρου. Άρα, μια τυχαία μεταβλητή είναι μια απεικόνιση που αντικατοπτρίζει το αποτέλεσμα ενός τυχαίου πειράματος. Μια *διακριτή τυχαία μεταβλητή* λαμβάνει μόνο διακριτές τιμές από σύνολο πεπερασμένο ή αριθμήσιμο. Μια *συνεχής τυχαία μεταβλητή* λαμβάνει τιμές από την ευθεία των πραγματικών αριθμών ή διαστήματά της.

Συνάρτηση Κατανομής Πιθανότητας –ΣΚΠ (Probability Distribution Function –PDF). Ονομάζεται και *αθροιστική ή σωρευτική συνάρτηση* κατανομής. Για μια δεδομένη τιμή x, η ΣΚΠ ισούται με την πιθανότητα η τυχαία μεταβλητή X να είναι μικρότερη ή ίση του x:

$$F_X(x) = \Pr[X \leq x] \tag{2.1}$$

Συνάρτηση Πυκνότητας Πιθανότητας –σππ (Probability Density Function –pdf). Ορίζεται για συνεχείς τυχαίες μεταβλητές και είναι η παράγωγος f_X της ΣΚΠ F_X. Με δεδομένη τη σππ, η πιθανότητα το X να βρίσκεται στο διάστημα (c_1, c_2) υπολογίζεται ως:

$$\Pr[c_1 < X \leq c_2) = F_X(c_2) - F_X(c_1) = \int_{c_1}^{c_2} f_X(x)\,\mathrm{d}x \tag{2.2}$$

Συνάρτηση Μάζας Πιθανότητας –σμπ (Probability Mass Function –pmf). Ορίζεται για διακριτές τυχαίες μεταβλητές για τις οποίες η ΣΚΠ είναι μη συνεχής, άρα και μη παραγωγίσιμη. Αν θεωρήσουμε διακριτή τυχαία μεταβλητή X, η οποία μπορεί να λάβει τις τιμές $x_1, x_2, \ldots, x_n$, με αντίστοιχες πιθανότητες $p_1, p_2, \ldots, p_n$, η σμπ απεικονίζει κάθε τιμή x_i στην αντίστοιχη p_i.

Η πιθανότητα το X να βρίσκεται στο διάστημα (c_1, c_2) υπολογίζεται με την άθροιση:

$$\Pr[c_1 < X \leq c_2) = F_X(c_2) - F_X(c_1) = \sum_{\substack{i \\ c_1 < x_i \leq c_2}} p_i \tag{2.3}$$

Μέση τιμή

$$\mu = \mathrm{E}\left[X\right] = \begin{cases} \sum_{i=1}^{n} p_i x_i & \text{διακριτή κατανομή} \\ \int_{-\infty}^{\infty} x f_X(x)\,\mathrm{d}x & \text{συνεχής κατανομή} \end{cases} \tag{2.4}$$

Διασπορά

$$\sigma^2 = \mathrm{Var}\left(X\right) = \mathrm{E}\left[(X-\mu)^2\right] = \begin{cases} \sum_{i=1}^{n} p_i (x_i - \mu)^2 & \text{διακριτή κατανομή} \\ \int_{-\infty}^{\infty} (x-\mu)^2 f_X(x)\,\mathrm{d}x & \text{συνεχής κατανομή} \end{cases} \tag{2.5}$$

Στοχαστικές διαδικασίες

Όπως παραπάνω, θεωρούμε έναν χώρο πιθανότητας $(\Omega, \mathcal{E}, P)$. Μια *στοχαστική διαδικασία* μπορεί να οριστεί ως εξής: σε κάθε δείγμα $\omega \in \Omega$ αντιστοιχίζουμε μια συνάρτηση του χρόνου $X(t, \omega)$. Αυτή η οικογένεια συναρτήσεων αποτελεί μια στοχαστική διαδικασία. Διαφορετικά, θα μπορούσαμε να πούμε ότι για κάθε τιμή t που ανήκει σε κάποιο δοσμένο σύνολο τιμών επιλέγουμε μια τυχαία μεταβλητή $X(t, \omega)$, ορίζοντας έτσι μια συλλογή τυχαίων μεταβλητών που εξαρτώνται από τον χρόνο. Συνήθως, μια στοχαστική διαδικασία συμβολίζεται απλά σαν μια συνάρτηση $X(t)$, της οποίας οι τιμές είναι τυχαίες μεταβλητές.

Για να χαρακτηρίσουμε μια στοχαστική διαδικασία $X(t)$, ορίζουμε για κάθε επιτρεπτή τιμή του t τη συνάρτηση κατανομής πιθανότητας:

$$F_X(x; t) = \Pr[X(t) \leq x] \tag{2.6}$$

Στη συνέχεια, ορίζουμε για ένα σύνολο n επιτρεπτών τιμών του t, την από κοινού συνάρτηση κατανομής πιθανότητας (Joint PDF):

$$F_{X_1 X_2 \ldots X_n}(x_1, x_2, \ldots, x_n; t_1, t_2, \ldots, t_n) \triangleq \Pr[X(t_1) \leq x_1, X(t_2) \leq x_2, \ldots, X(t_n) \leq x_n] \tag{2.7}$$

όπου το σύμβολο $\triangleq$ διαβάζεται «ισούται εξ ορισμού». Θα συμβολίζουμε τη συνάρτηση αυτή με τη διανυσματική μορφή $F_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{t})$. Για τον πλήρη χαρακτηρισμό μιας στοχαστικής διαδικασίας, θα πρέπει κανείς να δώσει τη συνάρτηση $F_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{t})$ για όλα τα n και όλα τα δυνατά υποσύνολα τιμών $\{x_i\}$, $\{t_i\}$, πράγμα που φαίνεται πρακτικά αδύνατο. Ευτυχώς, οι περισσότερες στοχαστικές διαδικασίες που μας ενδιαφέρουν στην πράξη μπορούν να χαρακτηριστούν με πολύ απλό τρόπο.

Η συνάρτηση πυκνότητας πιθανότητας μιας στοχαστικής διαδικασίας ορίζεται ως:

$$f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{t}) \triangleq \frac{\partial F_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{t})}{\partial \boldsymbol{x}} \tag{2.8}$$

και, από αυτήν, η μέση τιμή της στοχαστικής διαδικασίας:

$$\overline{X}(t) = E[X(t)] = \int_{-\infty}^{\infty} x f_X(x;t)\,\mathrm{d}x \tag{2.9}$$

και η συνάρτηση αυτοσυσχέτισης:

$$\begin{aligned} R_{XX}(t_1,t_2) &= E[X(t_1)X(t_2)] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 f_{X_1X_2}(x_1,x_2;t_1,t_2)\,\mathrm{d}x_1\,\mathrm{d}x_2 \end{aligned} \tag{2.10}$$

2.2 Ταξινόμηση Στοχαστικών Διαδικασιών

Η ταξινόμηση των στοχαστικών διαδικασιών εξαρτάται από τρεις παράγοντες, οι οποίοι είναι:

(i) **Ο χώρος καταστάσεων**, δηλαδή το σύνολο των δυνατών τιμών που μπορεί να πάρουν οι τυχαίες μεταβλητές $X(t)$. Ο χώρος καταστάσεων είναι *διακριτός*, αν είναι πεπερασμένος ή απαριθμητός. Μια διαδικασία διακριτών καταστάσεων αναφέρεται συχνά ως *αλυσίδα*. Διαφορετικά, ο χώρος καταστάσεων είναι *συνεχής*, αν αποτελείται από ένα πεπερασμένο ή άπειρο συνεχές διάστημα της ευθείας των πραγματικών αριθμών (ή από ένα σύνολο τέτοιων διαστημάτων).

(ii) **Η παράμετρος του χρόνου**, η οποία χαρακτηρίζεται από το σύνολο των επιτρεπτών τιμών του χρόνου για τις οποίες ορίζεται η στοχαστική διαδικασία. Όπως και ο χώρος καταστάσεων, το σύνολο αυτό μπορεί να έιναι διακριτό ή συνεχές, οπότε αναφερόμαστε σε διαδικασίες *διακριτής παραμέτρου* ή διαδικασίες *συνεχούς παραμέτρου*.

(iii) **Οι στατιστικές εξαρτήσεις** μεταξύ των τυχαίων μεταβλητών $X(t)$ για διαφορετικές τιμές της παραμέτρου t, οι οποίες περιγράφονται από την από κοινού συνάρτηση κατανομής πιθανότητας $F_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{t})$ των τυχαίων μεταβλητών $\boldsymbol{X} = (X(t_1), X(t_2), \ldots, X(t_n))$ για όλα τα $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, $\boldsymbol{t} = (t_1, t_2, \ldots, t_n)$ και όλες τις τιμές του n.

Παραθέτουμε ορισμένα παραδείγματα που αντιστοιχούν στις διαφορετικές επιλογές για τον χώρο καταστάσεων και την παράσταση του χρόνου.

- Ο χρόνος αναμονής ενός μηνύματος μέχρι να αρχίσει η επεξεργασία του, $\{W(t), t \geq 0\}$, όπου t ο χρόνος άφιξης του μηνύματος. Πρόκειται για διαδικασία συνεχούς παραμέτρου και συνεχούς χώρου καταστάσεων.

- Ο αριθμός μηνυμάτων που φθάνουν στο διάστημα από 0 έως t, $\{N(t), t \geq 0\}$. Έχουμε διαδικασία συνεχούς παραμέτρου και διακριτού χώρου καταστάσεων.

- Έστω $\{X_n, n = 1, 2, 3, 4, 5, 6, 7\}$ ο μέσος χρόνος εκτέλεσης ενός προγράμματος τη n-στή μέρα της εβδομάδας. Έχουμε διαδικασία διακριτής παραμέτρου και συνεχούς χώρου καταστάσεων.

- Έστω $\{X_n, n = 1, \ldots, 365(366)\}$ ο αριθμός προγραμμάτων που εκτελούνται τη n-στή μέρα του χρόνου. Πρόκειται για διαδικασία διακριτής παραμέτρου και διακριτού χώρου καταστάσεων.

Όπως αναφέρθηκε και παραπάνω, ο πλήρης χαρακτηρισμός μιας στοχαστικής διαδικασίας όσον αφορά τις στατιστικές εξαρτήσεις δεν είναι καθόλου απλός. Θα αναφερθούμε στη συνέχεια σε ορισμένους σημαντικούς τύπους στοχαστικών διαδικασιών, με ευρεία χρήση στη μοντελοποίηση συστημάτων, οι οποίοι χαρακτηρίζονται από διάφορα είδη σχέσεων εξάρτησης μεταξύ των τυχαίων μεταβλητών [7].

(i) **Στατικές διαδικασίες**

Μια στοχαστική διαδικασία $X(t)$ ονομάζεται *στατική* (stationary), αν όλες οι συναρτήσεις $F_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{t})$ μένουν αμετάβλητες σε μετατοπίσεις στον χρόνο, δηλαδή ισχύει:

$$F_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{t}+\tau) = F_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{t}) \tag{2.11}$$

όπου τ σταθερά και $\boldsymbol{t} + \tau = (t_1 + \tau, t_2 + \tau, \ldots, t_n + \tau)$.
Αν μια στοχαστική διαδικασία είναι στατική θα ισχύει:

$$\overline{X(t)} = \overline{X} \tag{2.12}$$

και

$$R_{XX}(t_1, t_2) = R_{XX}(t_2 - t_1) \tag{2.13}$$

δηλαδή η μέση τιμή είναι ανεξάρτητη του t και η συνάρτηση αυτοσυσχέτισης εξαρτάται μόνο από τη διαφορά $\tau = t_2 - t_1$. Μια διαδικασία λέγεται *στατική με την ευρεία έννοια* (wide-sense stationary), αν ισχύουν οι σχέσεις (2.12) και (2.13). Πρέπει να σημειωθεί ότι κάθε στατική διαδικασία είναι και στατική με την ευρεία έννοια, αλλά όχι αντίστροφα.

(ii) **Ανεξάρτητες διαδικασίες**

Η απλούστερη περίπτωση στοχαστικής διαδικασίας είναι όταν οι τυχαίες μεταβλητές που αντιστοιχούν σε διαφορετικές τιμές της παραμέτρου t είναι ανεξάρτητες μεταξύ τους, οπότε θα ισχύει:

$$f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{t}) = f_{X_1}(x_1; t_1) \ldots f_{X_n}(x_n; t_n) \tag{2.14}$$

Μια τέτοια διαδικασία στερείται δομής και αποτελεί ακραία περίπτωση, η οποία για συνεχή παράμετρο αναφέρεται ως *λευκός θόρυβος*.

(iii) **Διαδικασίες Markov**

Οι ιδιότητες των διαδικασιών αυτών μελετήθηκαν αρχικά από τον A.A. Markov σε άρθρο του που δημοσιεύθηκε το 1907 [10]. Μια διαδικασία Markov (ή «μαρκοβιανή» διαδικασία) διακριτού χώρου καταστάσεων αναφέρεται ως αλυσίδα Markov. Η βασική ιδιότητα μιας διαδικασίας Markov είναι ότι, σε κάθε χρονική στιγμή, η επίδραση ολόκληρου του παρελθόντος πάνω στο μέλλον της διαδικασίας εκφράζεται αποκλειστικά μέσα από την τρέχουσα τιμή της διαδικασίας. Η ιδιότητα αυτή είναι γνωστή ως *έλλειψη μνήμης* (memoryless property) και περιορίζει τη γενικότητα των διαδικασιών Markov. Η μελέτη των διαδικασιών αυτών, όμως, είναι βασική για τη θεωρία αναμονής και γι' αυτό θα ασχοληθούμε ιδιαίτερα στη συνέχεια με τις αλυσίδες Markov διακριτής και συνεχούς παραμέτρου (χρόνου).

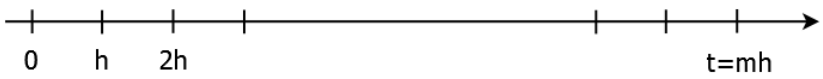
(iv) **Διαδικασίες γεννήσεων–θανάτων**

Οι διαδικασίες αυτές αποτελούν υποσύνολο των αλυσίδων Markov και έχουν παίξει σημαντικό ρόλο στην ανάπτυξη της θεωρίας αναμονής. Η βασική συνθήκη που ισχύει είναι ότι μεταβάσεις γίνονται μόνο μεταξύ γειτονικών καταστάσεων. Εάν θεωρήσουμε το σύνολο των μη αρνητικών ακεραίων ως χώρο καταστάσεων της διαδικασίας, τότε από την κατάσταση i η διαδικασία μπορεί να μεταβεί μόνο σε μία από τις καταστάσεις $i-1$ ή $i+1$, πράγμα το οποίο θα συμβόλιζε ένα «θάνατο» ή μια «γέννηση» αντίστοιχα, αν υποθέταμε ότι η διαδικασία παριστάνει τις μεταβολές στο μέγεθος κάποιου πληθυσμού.

(v) **Ημιμαρκοβιανές διαδικασίες**

Στις διαδικασίες Markov η ιδιότητα της έλλειψης μνήμης ισχύει σε οποιαδήποτε χρονική στιγμή. Η συνθήκη αυτή επιβάλλει, όπως θα δούμε στη συνέχεια, ότι το χρονικό διάστημα μεταξύ διαδοχικών αλλαγών κατάστασης (ή ισοδύναμα το χρονικό διάστημα κατά το οποίο η διαδικασία παραμένει σε μια κατάσταση) ακολουθεί μια κατανομή πιθανότητας που εξασφαλίζει την έλλειψη μνήμης. Η κατανομή αυτή πρέπει να είναι η γεωμετρική κατανομή για διακριτό χρόνο ή η εκθετική κατανομή για συνεχή χρόνο.

Αν θελήσουμε να χαλαρώσουμε αυτόν τον περιορισμό, οδηγούμαστε στις *ημιμαρκοβιανές διαδικασίες* (semi-Markov processes) στις οποίες ο χρόνος, κατά τον οποίο η διαδικασία παραμένει σε μια κατάσταση, μπορεί να ακολουθεί οποιαδήποτε κατανομή πιθανότητας. Παρ' όλα αυτά, στις χρονικές στιγμές αλλαγής κατάστασης η διαδικασία συμπεριφέρεται σαν μια κοινή διαδικασία Markov, οπότε αναφερόμαστε στην ενσωματωμένη διαδικασία (ή αλυσίδα) Markov (imbedded Markov chain). Προφανώς, οι διαδικασίες Markov αποτελούν υποσύνολο των ημιμαρκοβιανών διαδικασιών.



Σχήμα 2.1: *Υπολογισμός κατανομής Poisson.*

(vi) **Τυχαίοι περίπατοι**

Μια ακολουθία τυχαίων μεταβλητών $\{S_n\}$ ονομάζεται τυχαίος περίπατος (Random Walk), αν ισχύει:

$$S_n = X_1 + X_2 + \ldots + X_n \; n = 1, 2, \ldots \tag{2.15}$$

όπου $S_0 = 0$ και $X_1, X_2, \ldots$ είναι μια ακολουθία τυχαίων μεταβλητών ανεξάρτητων και με την ίδια κατανομή πιθανότητας. Ένας τυχαίος περίπατος θα μπορούσε να θεωρηθεί ως η κίνηση ενός σωματιδίου σε ένα διακριτό χώρο καταστάσεων, έτσι ώστε κάθε φορά η επόμενη θέση να καθορίζεται από την προηγούμενη θέση συν μια τυχαία μεταβλητή. Ο δείκτης n απλώς μετρά τον αριθμό των αλλαγών κατάστασης για τη στοχαστική διαδικασία. Οι τυχαίοι περίπατοι είναι υποσύνολο των ημιμαρκοβιανών διαδικασιών.

(vii) **Ανανεωτικές διαδικασίες**

Οι ανανεωτικές διαδικασίες (Renewal Processes) μπορούν να θεωρηθούν ειδική περίπτωση των τυχαίων περιπάτων. Οι διαδικασίες αυτές περιγράφουν τον αριθμό των αλλαγών κατάστασης (μεταβάσεων) ως συνάρτηση του χρόνου. Έτσι, αν οι τυχαίες μεταβλητές X_n της Εξίσωσης (2.15) παριστάνουν τους χρόνους μεταξύ μεταβάσεων, τότε η τυχαία μεταβλητή S_n παριστάνει το χρόνο στον οποίο έγινε η n-στή μετάβαση. Η διαφορά σε σχέση με τους τυχαίους περιπάτους είναι ότι γι' αυτούς η Εξίσωση (2.15) περιγράφει την κατάσταση της διαδικασίας, ενώ ο χρόνος μεταξύ μεταβάσεων είναι κάποια άλλη τυχαία μεταβλητή.

2.3 Η Διαδικασία Poisson

Θεωρούμε μια ακολουθία από τυχαία γεγονότα, όπως π.χ. αφίξεις εργασιών σε ένα υπολογιστικό κέντρο, κλήσεις σε ένα τηλεφωνικό κέντρο, διακοπές υλικού ή λογισμικού σε ένα υπολογιστικό σύστημα. Μια τέτοια ακολουθία γεγονότων μπορεί να περιγραφεί από τη στοχαστική διαδικασία $\{N(t), t \geq 0\}$, όπου $N(t)$ ο αριθμός των γεγονότων που συνέβησαν στο διάστημα $(0, t]$.

Η διαδικασία $\{N(t), t \geq 0\}$ είναι διαδικασία Poisson με ρυθμό $\lambda > 0$, αν ισχύουν οι πιο κάτω συνθήκες:

(i) Γεγονότα που συμβαίνουν σε μη επικαλυπτόμενα χρονικά διαστήματα είναι ανεξάρτητα μεταξύ τους.

(ii) Οι μεταβολές της διαδικασίας είναι στατικές. (Η κατανομή του αριθμού γεγονότων σε οποιοδήποτε χρονικό διάστημα εξαρτάται μόνο από το μήκος του διαστήματος και όχι από την αρχή του.)

(iii) Η πιθανότητα να συμβεί ακριβώς ένα γεγονός σε οποιοδήποτε χρονικό διάστημα μήκους h είναι $\lambda h + o(h)$. (Μια συνάρτηση f είναι $o(h)$ αν ισχύει $\lim_{h\to 0} \frac{f(h)}{h} = 0$.)

(iv) Η πιθανότητα να συμβούν περισσότερα από ένα γεγονότα σε οποιοδήποτε χρονικό διάστημα μήκους h είναι $o(h)$.

Από τις δύο τελευταίες συνθήκες συνεπάγεται ότι η πιθανότητα να μη συμβεί κανένα γεγονός σε διάστημα μήκους h είναι $1 - \lambda h + o(h)$.

Με βάση τις πιο πάνω συνθήκες θα αναζητήσουμε την κατανομή πιθανότητας της τυχαίας μεταβλητής $N(t)$. Διαιρούμε το διάστημα $(0, t]$ σε m υποδιαστήματα, έτσι ώστε το κάθε ένα από αυτά να έχει μήκος $h = t/m$ (Σχ. 2.1). Σύμφωνα με τις συνθήκες, η ύπαρξη γεγονότος σε ένα υποδιάστημα μπορεί να θεωρηθεί σαν «επιτυχία» ενός πειράματος Bernoulli, και ο αριθμός των γεγονότων στο διάστημα $(0, t]$ ως το αποτέλεσμα μιας ακολουθίας από πειράματα Bernoulli. Συνεπώς η πιθανότητα να συμβούν ακριβώς n γεγονότα στα m υποδιαστήματα θα δίνεται από τη διωνυμική κατανομή: $\binom{m}{n}[\lambda h + o(h)]^n[1 - \lambda h + o(h)]^{m-n}$.

Παίρνοντας τα όρια $h \to 0$ και $m \to \infty$, διατηρώντας $mh = t$ σταθερό, βρίσκουμε:

$$\begin{aligned} \Pr[N(t) = n] &= \lim_{m\to\infty} \frac{m!}{n!(m-n)!} \cdot \left(\frac{\lambda t}{m}\right)^n \left(1 - \frac{\lambda t}{m}\right)^{m-n} = \\ &= \frac{(\lambda t)^n}{n!} \cdot \lim_{m\to\infty} \frac{m!}{m^n(m-n)!} \cdot \lim_{m\to\infty} \left(1 - \frac{\lambda t}{m}\right)^{m-n} \end{aligned} \tag{2.16}$$

Έχουμε όμως:

$$\lim_{m\to\infty} \frac{m!}{m^n(m-n)!} = 1 \tag{2.17}$$

και

$$\begin{aligned} \lim_{m\to\infty} \left(1 - \frac{\lambda t}{m}\right)^m &= \lim_{m\to\infty} \sum_{i=0}^{m} \binom{m}{i} \left(\frac{-\lambda t}{m}\right)^i = \\ &= \lim_{m\to\infty} \frac{m!}{m^i(m-i)!} \cdot \lim_{m\to\infty} \sum_{i=0}^{m} \frac{(-\lambda t)^i}{i!} = e^{-\lambda t} \end{aligned} \tag{2.18}$$

οπότε η (2.16) γίνεται:

$$\Pr[N(t) = n] = \frac{(\lambda t)^n}{n!} e^{-\lambda t},\ t \geq 0,\ n = 0, 1, 2, \ldots \tag{2.19}$$

που είναι η γνωστή κατανομή Poisson.

Εύκολα μπορεί κανείς να υπολογίσει ότι:

$$\mathrm{E}\left[N(t)\right] = \lambda t \text{ και } \mathrm{Var}\left[N(t)\right] = \lambda t \tag{2.20}$$

Επομένως, ο μέσος αριθμός γεγονότων ανά χρονική μονάδα θα είναι $\lambda t/t = \lambda$, πράγμα που δικαιολογεί το όνομα *ρυθμός* για την παράμετρο λ.

2.3.1 Ιδιότητες της Διαδικασίας Poisson

(i) **Διαστήματα μεταξύ γεγονότων – Η εκθετική κατανομή**

Έστω X το χρονικό διάστημα από τη στιγμή 0 (επιλεγμένη αυθαίρετα) μέχρι το πρώτο γεγονός. Έχουμε:

$$\Pr[X > x] = \Pr[N(x) = 0] = e^{-\lambda x} \tag{2.21}$$

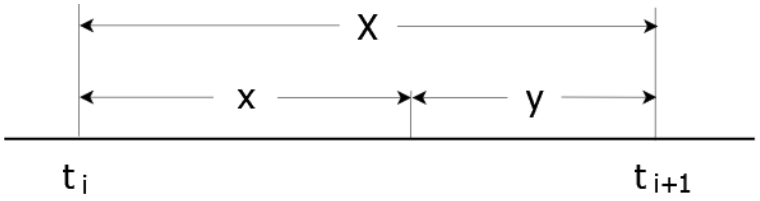
σύμφωνα με την (2.19). Άρα η συνάρτηση κατανομής πιθανότητας του X και η αντίστοιχη συνάρτηση πυκνότητας πιθανότητας θα είναι:

$$F_X(x) = \Pr[X \leq x] = 1 - e^{-\lambda x}, \quad x \geq 0 \tag{2.22}$$

$$f_X(x) = F'_X(x) = \lambda e^{-\lambda x}, \quad x \geq 0 \tag{2.23}$$

Αν επιλέξουμε ως χρονική στιγμή 0 τη στιγμή ενός γεγονότος, τότε η τυχαία μεταβλητή X παριστάνει το διάστημα μεταξύ δύο διαδοχικών γεγονότων. Συμπεραίνουμε ότι τα διαστήματα μεταξύ γεγονότων της διαδικασίας Poisson ακολουθούν εκθετική κατανομή με μέση τιμή $1/\lambda$.

Αποδεικνύεται ότι ισχύει και αντίστροφα: έστω $\{N(t), t \geq 0\}$ μια στοχαστική διαδικασία που μετρά τον αριθμό γεγονότων στο διάστημα $(0, t]$, τέτοια ώστε τα διαστήματα μεταξύ γεγονότων να είναι ανεξάρτητες τυχαίες μεταβλητές κατανεμημένες εκθετικά με μέση τιμή $1/\lambda$. Τότε η $\{N(t), t \geq 0\}$ είναι διαδικασία Poisson με ρυθμό λ.



Σχήμα 2.2: *Έλλειψη μνήμης.*

Έλλειψη μνήμης. Η σπουδαιότερη ιδιότητα της εκθετικής κατανομής είναι η *έλλειψη μνήμης*. Έστω t_i η στιγμή του i-στού γεγονότος και έστω ότι έχει ήδη παρέλθει διάστημα x πριν συμβεί το επόμενο γεγονός (Σχ. 2.2).

Ενδιαφερόμαστε για την πιθανότητα το διάστημα που υπολείπεται μέχρι το επόμενο γεγονός να είναι μεγαλύτερο από y, δεδομένου ότι έχει ήδη παρέλθει διάστημα x από το τελευταίο γεγονός. Αν X ο χρόνος μεταξύ γεγονότων, θα έχουμε σύμφωνα με τον ορισμό της πιθανότητας υπό συνθήκη:

$$\begin{aligned} \Pr[X > x+y / X > x] &= \frac{\Pr[X > x+y, X > x]}{\Pr[X > x]} = \frac{\Pr[X > x+y]}{\Pr[X > x]} = \\ &= \frac{e^{-\lambda(x+y)}}{e^{-\lambda x}} = e^{-\lambda y} = \Pr[X > y] \end{aligned} \tag{2.24}$$

δηλαδή η υπό συνθήκη κατανομή του υπολειπόμενου διαστήματος είναι ανεξάρτητη του x και είναι ίδια με την κατανομή του X. Με άλλα λόγια, η κατανομή του χρόνου μέχρι το επόμενο γεγονός δεν εξαρτάται από το πότε συνέβη το τελευταίο γεγονός. Αποδεικνύεται ότι η εκθετική κατανομή είναι η μόνη συνεχής κατανομή με την ιδιότητα έλλειψης μνήμης.

Έστω τώρα X_1 και X_2 δύο ανεξάρτητες τυχαίες μεταβλητές κατανεμημένες εκθετικά με παραμέτρους λ_1, λ_2 αντίστοιχα. Τα X_1, X_2 μπορούν να θεωρηθούν ως οι διάρκειες δύο διεργασιών που εκτελούνται ταυτόχρονα. Αν κάποια χρονική στιγμή καμμία από τις δύο διεργασίες δεν έχει τελειώσει, μας ενδιαφέρει η κατανομή του διαστήματος X μέχρι να τελειώσει κάποια από τις δύο, ή ισοδύναμα η κατανομή του $\min(X_1, X_2)$ σύμφωνα με την ιδιότητα έλλειψης μνήμης. Έχουμε:

$$\Pr[X > x] = \Pr[X_1 > x, X_2 > x] = e^{-\lambda_1 x} e^{-\lambda_2 x} = e^{-(\lambda_1+\lambda_2)x}$$

ή

$$\Pr[X \le x] = 1 - e^{-(\lambda_1+\lambda_2)x} \tag{2.25}$$

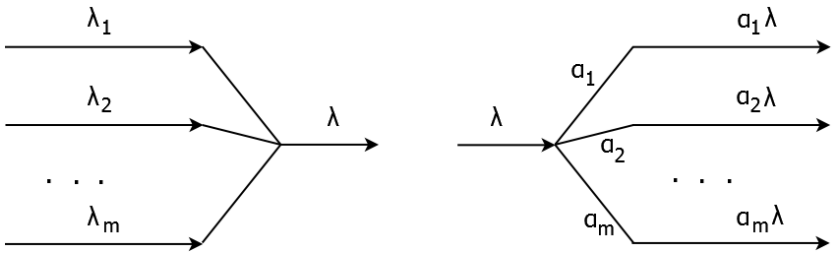
Άρα το διάστημα X είναι κατανεμημένο εκθετικά με παράμετρο $\lambda_1 + \lambda_2$. Η πιθανότητα να τελειώσει πρώτη η διεργασία 1 θα είναι:

$$\Pr[X_1 < X_2] = \int_0^\infty e^{-\lambda_2 x} \lambda_1 e^{-\lambda_1 x} \, \mathrm{d}x = \lambda_1/(\lambda_1+\lambda_2) \tag{2.26}$$

Αντίστοιχα $\Pr[X_2 < X_1] = \lambda_2/(\lambda_1+\lambda_2)$. Η κατανομή του διαστήματος $X = \min(X_1, X_2)$ δεν εξαρτάται από το ποια διεργασία τελειώνει πρώτη. Τα αποτελέσματα αυτά, τα οποία γενικεύονται εύκολα για οποιονδήποτε αριθμό διεργασιών, έχουν άμεση εφαρμογή στις διαδικασίες Markov, όπως θα δούμε στη συνέχεια.

(ii) **Η κατανομή Erlang**

Ας υπολογίσουμε την κατανομή του διαστήματος T_n από την αρχή του χρόνου μέχρι τη στιγμή του n-στού γεγονότος. Η τυχαία μεταβλητή T_n αποτελεί το άθροισμα n ανεξάρτητων τυχαίων μεταβλητών



Σχήμα 2.3: *Υπέρθεση και διάσπαση.*

(διαστημάτων) που ακολουθούν την ίδια εκθετική κατανομή με παράμετρο λ. Έχουμε:

$$\begin{aligned} G_n(x) &= \Pr[T_n \le x] = \Pr[N(x) \ge n] = \sum_{i=n}^{\infty} e^{-\lambda x}(\lambda x)^i/i! \\ &= 1 - \sum_{i=0}^{n-1} e^{-\lambda x}(\lambda x)^i/i! \end{aligned} \tag{2.27}$$

Πρόκειται για τη γνωστή κατανομή Erlang n-σταδίων με αντίστοιχη συνάρτηση πυκνότητας πιθανότητας:

$$g_n(x) = \frac{\lambda(\lambda x)^{n-1}}{(n-1)!}e^{-\lambda x},\ x \ge 0 \tag{2.28}$$

(iii) **Η χρονική κατανομή των γεγονότων**

Η Εξίσωση (2.19) δίνει την κατανομή πιθανότητας για τον αριθμό γεγονότων σε ένα διάστημα μήκους t. Αν τώρα γνωρίζουμε ότι ακριβώς k γεγονότα συνέβησαν στο διάστημα αυτό, είναι ενδιαφέρον να αναζητήσουμε την από κοινού κατανομή των χρονικών στιγμών κατά τις οποίες συνέβησαν τα k γεγονότα. Αποδεικνύεται [7] ότι η κατανομή αυτή είναι η ίδια με την κατανομή k σημείων κατανεμημένων ομοιόμορφα στο ίδιο διάστημα. Με άλλα λόγια, η χρονική στιγμή κάθε γεγονότος ισοδυναμεί με ένα σημείο που επιλέγεται ανεξάρτητα σύμφωνα με μια ομοιόμορφη κατανομή πάνω στο διάστημα μήκους t.

(iv) **Υπέρθεση και διάσπαση διαδικασιών Poisson**

Όπως αναφέρθηκε παραπάνω, αν θεωρήσουμε m ανεξάρτητες τυχαίες μεταβλητές εκθετικά κατανεμημένες με παραμέτρους $\lambda_i, i = 1, 2, \ldots, m$, τότε το ελάχιστο αυτών των μεταβλητών ακολουθεί επίσης εκθετική κατανομή με παράμετρο $\lambda = \sum_{i=1}^{m} \lambda_i$. Αν τώρα θεωρήσουμε την υπέρθεση m ανεξάρτητων διαδικασιών Poisson με ρυθμούς $\lambda_i, i = 1, 2, \ldots, m$, τότε το διάστημα από μια τυχαία χρονική στιγμή μέχρι το επόμενο γεγονός θα ισοδυναμεί με το ελάχιστο m ανεξάρτητων τυχαίων μεταβλητών εκθετικά κατανεμημένων με τις αντίστοιχες παραμέτρους λ_i. Συμπεραίνουμε ότι η διαδικασία που προκύπτει από την υπέρθεση ανεξάρτητων διαδικασιών Poisson είναι επίσης διαδικασία Poisson με ρυθμό το άθροισμα των ρυθμών των επιμέρους διαδικασιών (Σχ. 2.3).

Αντίστροφα, ας θεωρήσουμε τη διάσπαση μιας διαδικασίας Poisson $\{N(t), t \ge 0\}$ σε δύο επιμέρους διαδικασίες $\{N_1(t), t \ge 0\}$, $\{N_2(t), t \ge 0\}$. Η διάσπαση πραγματοποιείται με μια ακολουθία πειραμάτων Bernoulli: κάθε γεγονός της διαδικασίας N ανατίθεται στη διαδικασία N_1 με πιθανότητα α_1 και στη N_2 με πιθανότητα α_2 ($\alpha_1 + \alpha_2 = 1$). Η από κοινού κατανομή πιθανότητας των $N_1(t)$, $N_2(t)$ θα είναι:

$$\begin{aligned} &\Pr[N_1(t) = n_1, N_2(t) = n_2] = \\ &\quad = \Pr[N_1(t) = n_1, N_2(t) = n_2 / N(t) = n_1 + n_2] \cdot \Pr[N(t) = n_1 + n_2] \\ &\quad = \frac{(n_1+n_2)!}{n_1!n_2!}\alpha_1^{n_1}\alpha_2^{n_2}\frac{(\lambda t)^{n_1+n_2}}{(n_1+n_2)!}e^{-\lambda t} \\ &\quad = \frac{(\alpha_1\lambda t)^{n_1}}{n_1!}e^{-\alpha_1\lambda t} \cdot \frac{(\alpha_2\lambda t)^{n_2}}{n_2!}e^{-\alpha_2\lambda t} \end{aligned} \tag{2.29}$$

δηλαδή οι διαδικασίες που προκύπτουν από τη διάσπαση είναι επίσης Poisson με ρυθμούς $\alpha_1\lambda$ και $\alpha_2\lambda$ και επιπλέον ανεξάρτητες μεταξύ τους. Και το αποτέλεσμα αυτό γενικεύεται εύκολα για διάσπαση σε οποιονδήποτε αριθμό επιμέρους διαδικασιών.

2.4 Αλυσίδες Markov Διακριτού Χρόνου

Ορισμός. *Η ακολουθία των τυχαίων μεταβλητών $X_1, X_2, \ldots$ αποτελεί μια αλυσίδα Markov διακριτού χρόνου αν για $n = 1, 2, \ldots$ και για όλες τις δυνατές τιμές των τυχαίων μεταβλητών ισχύει:*

$$\Pr[X_{n+1} = j / X_1 = i_1, X_2 = i_2, \ldots, X_n = i_n] = \Pr[X_{n+1} = j / X_n = i_n] \tag{2.30}$$

Ως παράδειγμα μπορούμε να θεωρήσουμε έναν ταξιδιώτη, ο οποίος περιπλανάται από πόλη σε πόλη μέσα σε μια χώρα. Έστω ότι η τυχαία μεταβλητή X_n παριστάνει την πόλη στην οποία βρίσκεται ο ταξιδιώτης το μεσημέρι της n-στής ημέρας. Ο ορισμός μάς λέει απλά ότι η επόμενη πόλη την οποία θα επισκεφθεί ο ταξιδιώτης εξαρτάται μόνο από την πόλη στην οποία βρίσκεται τώρα και όχι από τις πόλεις τις οποίες έχει ήδη επισκεφθεί, ή αλλιώς ότι η πορεία του ταξιδιώτη χαρακτηρίζεται από την ιδιότητα της έλλειψης μνήμης.

Η έκφραση στο δεξιό μερος της (2.30) ονομάζεται *πιθανότητα μετάβασης* (ενός βήματος) και δίνει την υπό συνθήκη πιθανότητα να γίνει μετάβαση από την κατάσταση i_n στο n-στό βήμα προς την κατάσταση j στο $n+1$ βήμα της διαδικασίας. Για να είναι πλήρως ορισμένη η εξέλιξη της διαδικασίας θα πρέπει να δίνεται κάποια αρχική κατανομή πιθανότητας $\Pr[X_0 = i]$.

Εάν οι πιθανότητες μετάβασης είναι ανεξάρτητες του n, τότε έχουμε μια *ομοιογενή* (homogeneous) αλυσίδα Markov και ορίζουμε:

$$p_{ij} \triangleq \Pr[X_{n+1} = j / X_n = i] \ \forall n \tag{2.31}$$

την πιθανότητα μετάβασης σε ένα βήμα από την κατάσταση i στην κατάσταση j. Όμοια μπορούμε να ορίσουμε τις πιθανότητες μετάβασης σε m βήματα:

$$p_{ij}^{(m)} \triangleq \Pr[X_{n+m} = j / X_n = i] \tag{2.32}$$

για τις οποίες εύκολα μπορούμε να γράψουμε την πιο κάτω αναδρομική σχέση:

$$p_{ij}^{(m)} = \sum_k p_{ik}^{(m-1)} p_{kj}, \ m = 2, 3, \ldots \tag{2.33}$$

Στη συνέχεια θα αναφερθούμε σε ομοιογενείς αλυσίδες Markov, δηλαδή σε αλυσίδες των οποίων οι πιθανότητες μετάβασης είναι στατικές στον χρόνο.

Ορίζουμε τη *μήτρα πιθανοτήτων μετάβασης* $\boldsymbol{P}$ με στοιχεία p_{ij}:

$$\boldsymbol{P} \triangleq [p_{ij}] \tag{2.34}$$

Παρατηρούμε ότι ισχύει για τα στοιχεία της μήτρας $\boldsymbol{P}$:

$$p_{ij} \geq 0, \qquad \forall i, j \tag{2.35}$$

$$\sum_j p_{ij} = 1, \qquad \forall i \tag{2.36}$$

Μια μήτρα που ικανοποιεί τις συνθήκες (2.35) και (2.36) ονομάζεται *στοχαστική μήτρα*.

Η «κίνηση» της διαδικασίας στον χώρο καταστάσεων μπορεί να απεικονιστεί γραφικά μέσω μιας δομής προσανατολισμένου γράφου, του οποίου οι κορυφές παριστάνουν τις καταστάσεις της διαδικασίας. Οι ακμές του γράφου δηλώνουν επιτρεπτές μεταβάσεις μεταξύ καταστάσεων και χαρακτηρίζονται από αριθμητικά βάρη τα οποία παριστάνουν τις πιθανότητες μετάβασης p_{ij}. Ένας τέτοιος γράφος ονομάζεται *γράφος μεταβάσεων* (state-transition graph).

Μια αλυσίδα Markov είναι *αμείωτη* (irreducible), αν από κάθε κατάσταση μπορούμε να φθάσουμε σε οποιαδήποτε άλλη κατάσταση. Έστω A το σύνολο όλων των καταστάσεων μιας αλυσίδας. Ένα υποσύνολο A_1 του A λέγεται *κλειστό*, αν δεν υπάρχει δυνατή μετάβαση ενός βήματος από οποιαδήποτε κατάσταση στο A_1 προς οποιαδήποτε κατάσταση στο A_1^C (συμπλήρωμα του A_1). Αν το A_1 περιλαμβάνει μόνο μια κατάσταση, τότε η κατάσταση αυτή λέγεται *απορροφητική* (absorbing) (για μια απορροφητική κατάσταση i θα ισχύει $p_{ii} = 1$). Προφανώς, σε μια αμείωτη αλυσίδα Markov δεν υπάρχουν κλειστά υποσύνολα.

2.4.1 Ταξινόμηση των Καταστάσεων

Σύμφωνα με την ιδιότητα έλλειψης μνήμης, η διαδικασία μπορεί να επανέλθει σε μια κατάσταση την οποία έχει ήδη επισκεφθεί. Ορίζουμε, λοιπόν, τις πιο κάτω ποσότητες:

$$f_j^{(n)} \triangleq \Pr[\text{η πρώτη επάνοδος στην κατάσταση } j \text{ γίνεται } n \text{ βήματα μετά την αναχώρηση από την κατάσταση } j] \tag{2.37}$$

$$f_j = \sum_{n=1}^{\infty} f_j^{(n)} = \Pr[\text{κάποτε γίνεται επάνοδος στην κατάσταση } j] \tag{2.38}$$

Ανάλογα με την τιμή της πιθανότητας επανόδου f_j μπορούμε να χαρακτηρίσουμε τις καταστάσεις μιας αλυσίδας Markov:

- Αν $f_j = 1$, η κατάσταση j λέγεται *επαναληπτική* (recurrent)
- Αν $f_j < 1$, η κατάσταση λέγεται *μεταβατική* (transient)
- Επιπλέον, αν οι μόνοι δυνατοί αριθμοί βημάτων στους οποίους μπορεί να γίνει επάνοδος στην κατάσταση j είναι $\gamma, 2\gamma, 3\gamma \ldots$ (όπου $\gamma > 1$ και είναι ο μεγαλύτερος ακέραιος για τον οποίο ισχύει αυτό), τότε η κατάσταση j λέγεται *περιοδική* με περίοδο γ. Αν $\gamma = 1$, τότε η κατάσταση j λέγεται *απεριοδική*.

Στη συνέχεια θεωρούμε τις επαναληπτικές καταστάσεις και ορίζουμε τον *μέσο χρόνο επανάληψης* (mean recurrence time) της κατάστασης j:

$$M_j \triangleq \sum_{n=1}^{\infty} nfwj^{(n)} \tag{2.39}$$

Αν $M_j = \infty$, η κατάσταση j λέγεται *μηδενική επαναληπτική* (null recurrent), ενώ αν $M_j < \infty$, η κατάσταση j λέγεται *θετική επαναληπτική* (positive recurrent).

Τέλος, ορίζουμε την πιθανότητα $\pi_j^{(n)}$ να βρεθεί η διαδικασία στην κατάσταση j στο n-στό βήμα:

$$\pi_j^{(n)} \triangleq \Pr[X_n = j] \tag{2.40}$$

Η ταξινόμηση των καταστάσεων χαρακτηρίζεται από το πιο κάτω βασικό θεώρημα [2].

Θεώρημα 2.1. *Οι καταστάσεις μιας αμείωτης αλυσίδας Markov είναι είτε όλες μεταβατικές είτε όλες θετικές επαναληπτικές. Επιπλέον, αν είναι περιοδικές, τότε όλες έχουν την ίδια περίοδο γ.*

2.4.2 Μόνιμη Κατάσταση

Ένα δεύτερο σημαντικό θεώρημα αναφέρεται στην ύπαρξη μιας *στατικής κατανομής πιθανότητας* $\{\pi_j\}$, η οποία περιγράφει την πιθανότητα να βρεθεί η διαδικασία στην κατάσταση j σε κάποια μακρινή χρονική στιγμή. Η στατική κατανομή αναφέρεται συχνά και ως *κατανομή μόνιμης κατάστασης* (steady-state distribution) ή *κατανομή κατάστασης ισορροπίας* (equilibrium-state distribution).

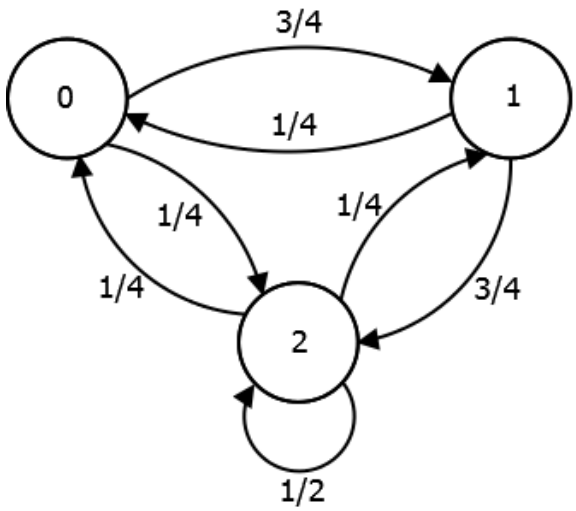
Θεώρημα 2.2. *Σε μια αμείωτη και απεριοδική ομοιογενή αλυσίδα Markov οι οριακές πιθανότητες:*

$$\pi_j = \lim_{n \to \infty} \pi_j^{(n)} \tag{2.41}$$

υπάρχουν πάντα και είναι ανεξάρτητες από την αρχική κατανομή πιθανότητας.

Επιπλέον,

(i) είτε όλες οι καταστάσεις είναι μεταβατικές ή όλες είναι μηδενικές επαναληπτικές, οπότε $\pi_j = 0$ για όλα τα j και δεν υπάρχει στατική κατανομή,



Σχήμα 2.4: *Αλυσίδα Markov — Γράφος μεταβάσεων.*

(ii) είτε όλες οι καταστάσεις είναι θετικές επαναληπτικές, οπότε $\pi_j > 0$ *για όλα τα* j *και οι πιθανότητες* $\{\pi_j\}$ *αποτελούν στατική κατανομή.*

Στην περίπτωση αυτή ισχύει:

$$\pi_j = \frac{1}{M_j} \tag{2.42}$$

και οι πιθανότητες π_j *καθορίζονται μονοσήμαντα από τη λύση του συστήματος:*

$$\pi_j = \sum_i \pi_i p_{ij},\ \forall j \tag{2.43}$$

$$\sum_j \pi_j = 1 \tag{2.44}$$

Αν ορίσουμε τώρα το διάνυσμα πιθανοτήτων $\boldsymbol{\pi}$:

$$\boldsymbol{\pi} \triangleq [\pi_j] \tag{2.45}$$

τότε οι Εξισώσεις (2.43) μπορούν να γραφούν με τη μορφή:

$$\boldsymbol{\pi} = \boldsymbol{\pi}\boldsymbol{P} \tag{2.46}$$

Σχετικά με την περίπτωση ii του Θεωρήματος 2.2 θα πρέπει να εισαγάγουμε την έννοια της *εργοδικότητας* (ergodicity). Μια κατάσταση j λέγεται *εργοδική*, αν είναι απεριοδική και θετική επαναληπτική. Μια αλυσίδα Markov λέγεται εργοδική, αν όλες οι καταστάσεις της είναι εργοδικές. Για μια αμείωτη εργοδική αλυσίδα Markov οι πιθανότητες $\{\pi_j^{(n)}\}$ συγκλίνουν πάντα σε μια οριακή στατική κατανομή. Ένα σημαντικό πόρισμα του Θεωρήματος 2.2 είναι ότι μια αμείωτη απεριοδική αλυσίδα Markov με πεπερασμένο πλήθος καταστάσεων είναι εργοδική [2].

Παράδειγμα 2.1. *Ας γυρίσουμε στο παράδειγμα του ταξιδιώτη του οποίου η περιπλάνηση παριστάνεται από μια αλυσίδα Markov. Υποθέτουμε ότι ο ταξιδιώτης μπορεί να επισκεφθεί τις πόλεις* 0, 1 *και* 2, *οι οποίες αντιπροσωπεύουν τις καταστάσεις της διαδικασίας. Στο Σχ.* 2.4 *απεικονίζεται ο γράφος μεταβάσεων.*

Εύκολα διαπιστώνουμε ότι πρόκειται για μια εργοδική αλυσίδα Markov (αμείωτη και με πεπερασμένο πλήθος καταστάσεων), άρα μπορούμε να αναζητήσουμε τη στατική κατανομή πιθανότητας. Έχουμε:

$$\boldsymbol{P} = \begin{bmatrix} 0 & 3/4 & 1/4 \\ 1/4 & 0 & 3/4 \\ 1/4 & 1/4 & 1/2 \end{bmatrix} \tag{2.47}$$

και από την (2.46),

$$\left.\begin{array}{ll} \pi_0 & = 0\pi_0 + \frac{1}{4}\pi_1 + \frac{1}{4}\pi_2 \\ \pi_1 & = \frac{3}{4}\pi_0 + 0\pi_1 + \frac{1}{4}\pi_2 \\ \pi_2 & = \frac{1}{4}\pi_0 + \frac{3}{4}\pi_1 + \frac{1}{2}\pi_2 \end{array}\right\} \tag{2.48}$$

Παρατηρούμε ότι οι Εξισώσεις (2.48) δεν είναι γραμμικά ανεξάρτητες. Γενικά, σε κάθε σύστημα με στοχαστική μήτρα μια εξίσωση θα είναι γραμμικά εξαρτημένη από τις υπόλοιπες. Για τη λύση του συστήματος θα πρέπει επομένως να χρησιμοποιηθεί και η συνθήκη (2.44) ή στο παράδειγμα:

$$1 = \pi_0 + \pi_1 + \pi_2 \tag{2.49}$$

Λύνοντας, λοιπόν, οποιεσδήποτε δύο από τις Εξισώσεις (2.48) μαζί με την (2.49) παίρνουμε την κατανομή:

$$\pi_0 = \frac{1}{5}\ ,\ \pi_1 = \frac{7}{25}\ ,\ \pi_2 = \frac{13}{25} \tag{2.50}$$

που αποτελεί τη στατική κατανομή πιθανότητας για την αλυσίδα Markov του παραδείγματος. □

2.4.3 Μεταβατική Κατάσταση

Πολλές φορές ενδιαφερόμαστε για τη μεταβατική συμπεριφορά ενός συστήματος, δηλαδή για τις πιθανότητες $\pi_j^{(n)}$ να βρεθεί η διαδικασία στην κατάσταση j στον χρόνο (βήμα) n. Αν ορίσουμε το διάνυσμα πιθανοτήτων στον χρόνο n:

$$\boldsymbol{\pi}^{(n)} = \left[\pi_j^{(n)}\right] \tag{2.51}$$

μπορούμε να γράψουμε γενικά:

$$\boldsymbol{\pi}^{(n)} = \boldsymbol{\pi}^{(n-1)}\boldsymbol{p}\ n = 1, 2, \ldots \tag{2.52}$$

ή λύνοντας αναδρομικά:

$$\boldsymbol{\pi}^{(n)} = \boldsymbol{\pi}^{(0)}\boldsymbol{P}^n\ n = 1, 2, \ldots \tag{2.53}$$

Η (2.53) δίνει τη γενική μέθοδο επίλυσης αν γνωρίζουμε τη μήτρα $\boldsymbol{P}$ και την αρχική κατανομή $\boldsymbol{\pi}^{(0)}$. Σύμφωνα με τα προηγούμενα, η στατική κατανομή θα είναι το όριο:

$$\boldsymbol{\pi} = \lim_{n\to\infty} \boldsymbol{\pi}^{(n)} \tag{2.54}$$

εφόσον υπάρχει (με την προϋπόθεση ότι η αλυσίδα Markov είναι εργοδική). Παίρνοντας τα όρια στα δύο μέλη της (2.52) καταλήγουμε στην (2.46), ανεξάρτητα από την αρχική κατανομή.

Αν θέλουμε να έχουμε τη μεταβατική απόκριση $\boldsymbol{\pi}^{(n)}$ από τις (2.52) ή (2.53) στη γενική της μορφή (τις πιθανότητες $\pi_j^{(n)}$ ως συναρτήσεις του n) καταφεύγουμε συνήθως στη χρήση μετασχηματισμών. Ειδικότερα στην περίπτωση αυτή, ο υπολογισμός γίνεται εύκολα με εφαρμογή του μετασχηματισμού z στην Εξίσωση (2.52) που είναι μια διανυσματική εξίσωση διαφορών [7, 8].

Τελειώνοντας, θα πρέπει να αναφερθούμε στην ιδιότητα έλλειψης μνήμης όσον αφορά τον χρόνο που περνάει η διαδικασία σε μια δεδομένη κατάσταση. Έστω ότι η διαδικασία μόλις εισήλθε στην κατάσταση j. Θα παραμείνει στην κατάσταση αυτή και στο επόμενο βήμα με πιθανότητα p_{jj} ή θα φύγει στο επόμενο βήμα με πιθανότητα $1 - p_{jj}$. Εφόσον παραμείνει στην κατάσταση j θα ισχύουν τα ίδια και στα επόμενα βήματα, ανεξάρτητα κάθε φορά, σύμφωνα με τον ορισμό της αλυσίδας Markov. Άρα η πιθανότητα να παραμείνει η διαδικασία στην κατάσταση j για m βήματα ακριβώς, δεδομένου ότι μόλις εισήλθε στην κατάσταση j, θα είναι:

$$P(m) = (1 - p_{jj})p_{jj}^{m-1} \tag{2.55}$$

Άρα ο αριθμός των χρονικών βημάτων που περνάει η διαδικασία σε μια κατάσταση ακολουθεί τη γεωμετρική κατανομή. Αποδεικνύεται εύκολα ότι, σε αντιστοιχία με την εκθετική κατανομή, η γεωμετρική κατανομή είναι η μόνη διακριτή κατανομή η οποία εμφανίζει την ιδιότητα της έλλειψης μνήμης.

2.5 Αλυσίδες Markov Συνεχούς Χρόνου

Στις αλυσίδες Markov συνεχούς χρόνου ο χώρος καταστάσεων παραμένει διακριτός, αλλά οι αλλαγές κατάστασης μπορούν να γίνουν σε οποιαδήποτε χρονική στιγμή και όχι σε διακριτά χρονικά βήματα. Έστω $X(t)$ η κατάσταση της διαδικασίας τη χρονική στιγμή t.

Ορισμός. *Η στοχαστική διαδικασία $\{X(t), t \geq 0\}$ αποτελεί μια αλυσίδα Markov συνεχούς χρόνου αν για $n = 1, 2, \ldots$ και για κάθε ακολουθία χρονικών στιγμών $t_1, t_2, \ldots, t_{n+1}$, όπου $t_1 < t_2 < \ldots < t_{n+1}$, ισχύει:*

$$\Pr[X(t_{n+1}) = j / X(t_1) = i_1, X(t_2) = t_2, \ldots, X(t_n) = i_n] = \Pr[X(t_{n+1}) = j / X(t_n) = i_n] \tag{2.56}$$

Ο πιο πάνω ορισμός αποτελεί απλή επέκταση του ορισμού (2.30) στην περίπτωση του συνεχούς χρόνου. Γενικά, η θεωρία των αλυσίδων Markov συνεχούς χρόνου είναι αντίστοιχη με αυτήν των αλυσίδων Markov διακριτού χρόνου. Για το λόγο αυτό, θα αναφερθούμε συνοπτικά σε ορισμένα βασικά χαρακτηριστικά τους [2].

Καταρχάς, γνωρίζουμε ότι για κάθε διαδικασία Markov, ο χρόνος τον οποίο περνά η διαδικασία σε μια δεδομένη κατάσταση θα πρέπει να είναι «χωρίς μνήμη». Αναφέρθηκε ήδη στην Ενότητα 2.3, ότι η μόνη συνεχής κατανομή με αυτή την ιδιότητα είναι η εκθετική κατανομή. Πράγματι, έστω ότι τη χρονική στιγμή t η διαδικασία βρίσκεται στην κατάσταση i, και έστω ότι η τυχαία μεταβλητή X_i παριστάνει το διάστημα μέχρι να φύγει η διαδικασία από την κατάσταση i. Μπορούμε να δείξουμε ότι η τυχαία μεταβλητή X_i ακολουθεί εκθετική κατανομή [2, 4]. Αν αντί για μια τυχαία χρονική στιγμή t, θεωρήσουμε τη στιγμή εισόδου της διαδικασίας στην κατάσταση i, συμπεραίνουμε ότι ο χρόνος παραμονής της διαδικασίας στην κατάσταση i ακολουθεί εκθετική κατανομή

Σε αντιστοιχία με τις αλυσίδες Markov διακριτού χρόνου, ορίζουμε στην περίπτωση του συνεχούς χρόνου τις πιο κάτω πιθανότητες μετάβασης για *ομοιογενή* αλυσίδα Markov:

$$p_{ij}(h) \triangleq \Pr[X(t+h) = j / X(t) = i] \ \forall t \tag{2.57}$$

και τη μήτρα πιθανοτήτων μετάβασης:

$$\boldsymbol{P}(h) \triangleq [p_{ij}(h)] \tag{2.58}$$

Επίσης, σε αντιστοιχία με τις πιθανότητες $\{\pi_j^{(n)}\}$ του διακριτού χρόνου ορίζουμε τις πιθανότητες:

$$\pi_j(t) \triangleq \Pr[X(t) = j] \tag{2.59}$$

και το διάνυσμα πιθανοτήτων:

$$\boldsymbol{\pi}(t) \triangleq [\pi_j(t)] \tag{2.60}$$

Σε αντιστοιχία με την Εξίσωση (2.52), έχουμε για τη μεταβατική συμπεριφορά της διαδικασίας:

$$\boldsymbol{\pi}(t+h) = \boldsymbol{\pi}(t) \cdot \boldsymbol{P}(h) \tag{2.61}$$

Η (2.61) μπορεί να γραφεί:

$$\frac{\boldsymbol{\pi}(t+h) - \boldsymbol{\pi}(t)}{h} = \boldsymbol{\pi}(t) \frac{\boldsymbol{P}(h) - \boldsymbol{I}}{h} \tag{2.62}$$

και παίρνοντας το όριο $h \to 0$ έχουμε:

$$\frac{\mathrm{d}\boldsymbol{\pi}(t)}{\mathrm{d}t} = \boldsymbol{\pi}(t) \cdot \boldsymbol{Q} \tag{2.63}$$

όπου:

$$\boldsymbol{Q} = \lim_{h \to 0} \frac{\boldsymbol{P}(h) - \boldsymbol{I}}{h} \tag{2.64}$$

Η μήτρα $\boldsymbol{Q}$ ονομάζεται *μήτρα ρυθμών μετάβασης* (Transition rate matrix) και τα στοιχεία της q_{ij} ορίζονται ως εξής:

$$q_{ii} = \lim_{h \to 0} \frac{p_{ii}(h) - 1}{h} \tag{2.65}$$

$$q_{ij} = \lim_{h \to 0} \frac{p_{ij}(h)}{h} \ i \neq j \tag{2.66}$$

Παρατηρούμε ότι, εφόσον $\sum_j p_{ij}(h) = 1$ για όλα τα i, θα ισχύει:

$$-q_{ii} = \sum_{j \neq i} q_{ij} \text{ ή } \sum_j q_{ij} = 0 \text{ για όλα τα } i \tag{2.67}$$

Μπορούμε να δώσουμε την ακόλουθη ερμηνεία στα όρια (2.65), (2.66):

- Δεδομένου ότι η διαδικασία βρίσκεται στην κατάσταση i, η υπό συνθήκη πιθανότητα να συμβεί μετάβαση σε άλλη κατάσταση εκτός της i σε διάστημα h θα είναι $-q_{ii}h + o(h)$. Έτσι μπορούμε να θεωρήσουμε ότι η ποσότητα $-q_{ii}$ είναι ο ρυθμός με τον οποίο η διαδικασία φεύγει από την κατάσταση i, όταν βρίσκεται σε αυτή την κατάσταση.

- Όμοια, δεδομένου ότι η διαδικασία βρίσκεται στην κατάσταση i, η υπό συνθήκη πιθανότητα να συμβεί μετάβαση από την κατάσταση αυτή στην κατάσταση j σε διάστημα h θα είναι $q_{ij}h + o(h)$. Έτσι, q_{ij} θα είναι ο ρυθμός με τον οποίο η διαδικασία περνάει από την κατάσταση i στην κατάσταση j, όταν βρίσκεται στην κατάσταση i.

Σύμφωνα με τα προηγούμενα, όμως, ο χρόνος παραμονής της διαδικασίας στην κατάσταση i ακολουθεί εκθετική κατανομή με παράμετρο λ_i. Για την εκθετική κατανομή, η πιθανότητα να φύγει η διαδικασία από την κατάσταση i σε ένα μικρό διάστημα h θα είναι:

$$\Pr[X_i \leq h] = 1 - e^{-\lambda_i h} = \lambda_i h + o(h) \tag{2.68}$$

ανεξάρτητα από τον χρόνο που έχει ήδη περάσει η διαδικασία στην κατάσταση i. Συμπεραίνουμε, λοιπόν, ότι ισχύει:

$$\lambda_i = -q_{ii} \tag{2.69}$$

Επίσης, η πιθανότητα να παραμείνει η διαδικασία στην κατάσταση i για διάστημα x και μετά να μεταβεί στην κατάσταση j στο διάστημα $(x, x + \mathrm{d}x)$ θα είναι $e^{-\lambda_i x} q_{ij}\,\mathrm{d}x$. Ολοκληρώνοντας την έκφραση αυτή για $x \geq 0$ βρίσκουμε την πιθανότητα μετάβασης από την κατάσταση i στην κατάσταση j ανεξάρτητα από τον χρόνο:

$$p_{ij} = \int_0^\infty e^{-\lambda_i x} q_{ij}\,\mathrm{d}x = \frac{q_{ij}}{\lambda_i} = -\frac{q_{ij}}{q_{ii}} \; i \neq j \tag{2.70}$$

Μπορούμε να θεωρήσουμε ότι οι διαδοχικές καταστάσεις τις οποίες επισκέπτεται η διαδικασία σχηματίζουν μια αλυσίδα Markov διακριτής παραμέτρου με πιθανότητες μετάβασης p_{ij} (γι' αυτό και χρησιμοποιήσαμε τον ίδιο συμβολισμό). Στην περίπτωση αυτή, η αλυσίδα διακριτού χρόνου είναι *ενσωματωμένη* (imbedded) στη διαδικασία Markov συνεχούς χρόνου.

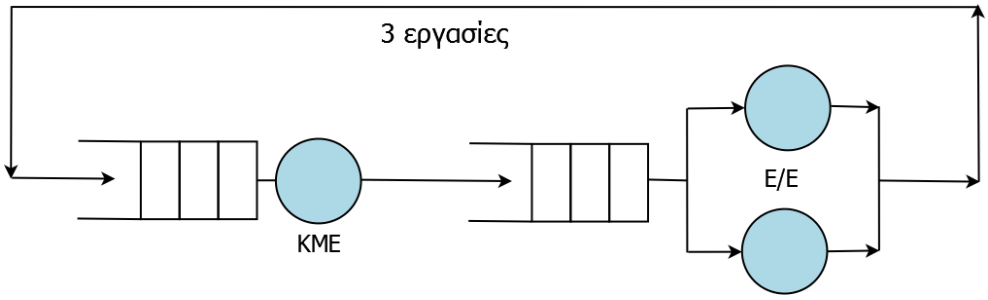
Όταν η διαδικασία μπαίνει στην κατάσταση i, μπορούμε να φανταστούμε ότι οι μεταβάσεις από την κατάσταση i στις καταστάσεις $j, j \neq i$, παριστάνονται από ανεξάρτητες διεργασίες που αρχίζουν την ίδια στιγμή και εξελίσσονται ταυτόχρονα. Οι διάρκειες των διεργασιών αυτών ακολουθούν εκθετικές κατανομές με αντίστοιχες παραμέτρους q_{ij}. Ο χρόνος παραμονής της διαδικασίας στην κατάσταση i θα είναι το διάστημα μέχρι να τελειώσει κάποια από τις διεργασίες, ή ισοδύναμα το ελάχιστο εκθετικά κατανεμημένων τυχαίων μεταβλητών. Όπως είδαμε στην Παράγραφο 2.3, σχετικά με την εκθετική κατανομή (Εξίσωση (2.25)), το διάστημα αυτό είναι εκθετικά κατανεμημένο με παράμετρο $\lambda_i = \sum_{j \neq i} q_{ij}$, σε συμφωνία με τις (2.69), (2.67). Επιπλέον, η πιθανότητα να τελειώσει πρώτη η διεργασία με παράμετρο q_{ij} (Εξίσωση (2.26)), ή ισοδύναμα, να γίνει μετάβαση στην κατάσταση j, θα είναι ίση με q_{ij}/λ_i, όπως δίνεται από την (2.70).

Η μεταβατική απόκριση $\boldsymbol{\pi}(t)$ της διαδικασίας δίνεται από τη διαφορική Εξίσωση (2.63). Για μια *εργοδική* αλυσίδα Markov συνεχούς χρόνου θα υπάρχει στατική κατανομή, η οποία θα δίνεται από το όριο:

$$\pi_j = \lim_{t \to \infty} \pi_j(t) \text{ ή } \boldsymbol{\pi} = \lim_{t \to \infty} \boldsymbol{\pi}(t) \tag{2.71}$$

ανεξάρτητα από την αρχική κατανομή. Εφαρμογή του ορίου στην (2.63) καθορίζει μονοσήμαντα τη στατική κατανομή πιθανότητας:

$$\boldsymbol{\pi}\boldsymbol{Q} = 0 \tag{2.72}$$



Σχήμα 2.5: *Κλειστό δίκτυο.*

$$\sum_j \pi_j = 1 \tag{2.73}$$

Η Εξίσωση (2.72) είναι αντίστοιχη της (2.46) για τον διακριτό χρόνο, με τη διαφορά ότι η $\boldsymbol{P}$ ήταν μήτρα *πιθανοτήτων* μετάβασης, ενώ η $\boldsymbol{Q}$ είναι η μήτρα *ρυθμών* μετάβασης.

Θα κλείσουμε με ένα παράδειγμα εφαρμογής της θεωρίας των αλυσίδων Markov συνεχούς χρόνου. Στο επόμενο κεφάλαιο, τα αποτελέσματα αυτά θα εξειδικευθούν στην περίπτωση των διαδικασιών γεννήσεων–θανάτων, οι οποίες παίζουν βασικό ρόλο στην ανάλυση συστημάτων αναμονής. Ήδη, η διαδικασία Poisson που εξετάστηκε αποτελεί μια σημαντική ειδική περίπτωση των διαδικασιών αυτών. Όσον αφορά άλλες κατηγορίες διαδικασιών που αναφέρθηκαν, θα αναπτυχθούν όταν αυτό θα είναι απαραίτητο στη συνέχεια.

Παράδειγμα 2.2. *Θεωρούμε το κλειστό δίκτυο ουρών του* Σχ. 2.5*, το οποίο μπορεί να χρησιμοποιηθεί ως μοντέλο ενός απλού υπολογιστικού συστήματος ομαδικής επεξεργασίας (batch processing).*

Υποθέτουμε ότι υπάρχει μια Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) και 2 *Μονάδες Εισόδου/Εξόδου (Ε/Ε). Κάθε στιγμή υπάρχουν στο σύστημα* 3 *εργασίες, κάθε μια από τις οποίες βρίσκεται στην ουρά της ΚΜΕ ή στην ουρά των μονάδων Ε/Ε (σε αναμονή ή εξυπηρέτηση). Οι εργασίες εξυπηρετούνται με τη σειρά άφιξής τους. Κάθε εργασία εξυπηρετείται από την ΚΜΕ μέχρι να χρειαστεί κάποια λειτουργία εισόδου/εξόδου, οπότε περνά στην ουρά των μονάδων Ε/Ε. Μετά την εξυπηρέτησή της εκεί, επιστρέφει στην ουρά της ΚΜΕ (Πολυπρογραμματισμός βαθμού* 3*).*

Υποθέτουμε ότι οι χρόνοι εξυπηρέτησης σε κάθε ουρά είναι ανεξάρτητες τυχαίες μεταβλητές που ακολουθούν την ίδια εκθετική κατανομή. Οι αντίστοιχες παράμετροι είναι α *για την ΚΜΕ και* β *για κάθε μονάδα Ε/Ε.*

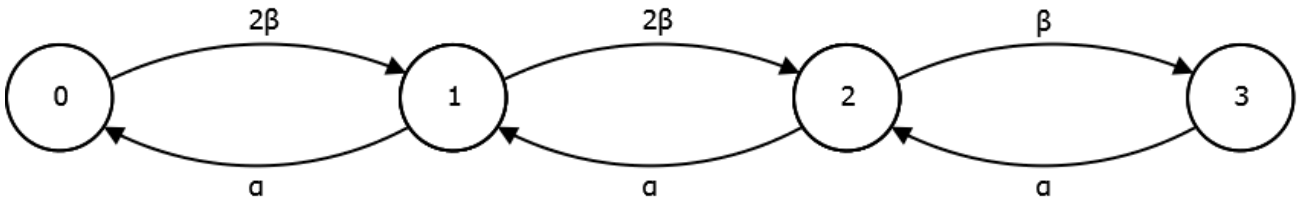
Μπορούμε να ορίσουμε ως κατάσταση του συστήματος κάθε στιγμή τον αριθμό i των εργασιών στην ουρά της ΚΜΕ, $i = 0, 1, 2, 3$. Προφανώς $3 - i$ εργασίες θα βρίσκονται αντίστοιχα στην ουρά των μονάδων Ε/Ε. Η συμπεριφορά του συστήματος μπορεί να περιγραφεί από μια αλυσίδα Markov συνεχούς χρόνου. Θα αναζητήσουμε τους ρυθμούς μετάβασης q_{ij}, $0 \leq i, j \leq 3$. Παρατηρούμε ότι μια αλλαγή κατάστασης καθορίζεται είτε από το τέλος μιας εξυπηρέτησης στην ΚΜΕ (εφόσον $i > 0$), το οποίο συμβαίνει με ρυθμό α, είτε από το τέλος μιας εξυπηρέτησης στις μονάδες Ε/Ε (εφόσον $i < 3$), το οποίο συμβαίνει με ρυθμό β αν εργάζεται μόνο η μια μονάδα Ε/Ε (άρα αν $i = 2$) και με ρυθμό 2β αν εργάζονται και οι δύο μονάδες Ε/Ε (άρα αν $i < 2$). Οδηγούμαστε, λοιπόν, στην ακόλουθη μήτρα ρυθμών μετάβασης:

$$\boldsymbol{Q} = \begin{bmatrix} -2\beta & 2\beta & 0 & 0 \\ \alpha & -(\alpha+2\beta) & 2\beta & 0 \\ 0 & \alpha & -(\alpha+\beta) & \beta \\ 0 & 0 & \alpha & -\alpha \end{bmatrix} \tag{2.74}$$

Σε αντιστοιχία με την περίπτωση του διακριτού χρόνου, η αλυσίδα Markov συνεχούς χρόνου μπορεί να παρασταθεί γραφικά από τον *γράφο μεταβάσεων* του Σχ. 2.6, όπου στις ακμές σημειώνονται οι ρυθμοί μετάβασης (δεν σημειώνουμε τους ρυθμούς q_{ii}).

Η αλυσίδα Markov είναι αμείωτη με πεπερασμένο πλήθος καταστάσεων, άρα θα υπάρχει η στατική κατανομή πιθανότητας $\boldsymbol{\pi} = [\pi_0, \pi_1, \pi_2, \pi_3]$, η οποία θα δίνεται από τη λύση του συστήματος

$$\boldsymbol{\pi Q} = 0 \tag{2.75}$$



Σχήμα 2.6: *Κλειστό δίκτυο — Γράφος μεταβάσεων.*

σε συνδυασμό με τη σχέση

$$\pi_0 + \pi_1 + \pi_2 + \pi_3 = 1 \tag{2.76}$$

Επιλύοντας το σύστημα, βρίσκουμε:

$$\pi_1 = \frac{2\beta}{\alpha}\pi_0, \quad \pi_2 = \frac{4\beta^2}{\alpha^2}\pi_0, \quad \pi_3 = \frac{4\beta^3}{\alpha^3}\pi_0 \tag{2.77}$$

όπου

$$\pi_0 = \left(1 + \frac{2\beta}{\alpha} + \frac{4\beta^2}{\alpha^2} + \frac{4\beta^3}{\alpha^3}\right)^{-1} \tag{2.78}$$

Αν υποθέσουμε ότι οι μέσοι χρόνοι εξυπηρέτησης στην ΚΜΕ και σε κάθε μονάδα Ε/Ε είναι αντίστοιχα $1/\alpha = 1$ sec και $1/\beta = 0,5$ sec, έχουμε:

$$\boldsymbol{\pi} = [0,019,\ 0,075,\ 0,302,\ 0,604]$$

Χρησιμοποιώντας τις στατικές πιθανότητες μπορούμε εύκολα να υπολογίσουμε μερικά ενδιαφέροντα μέτρα επίδοσης:

- Βαθμός χρησιμοποίησης (utilization) της ΚΜΕ =
  $= \pi_1 + \pi_2 + \pi_3 = 0,981$
- Ρυθμός απόδοσης (throughput) της ΚΜΕ =
  $= \alpha(\pi_1 + \pi_2 + \pi_3) = 0,981$ εργασίες/sec
- Μέσο μήκος ουράς στην ΚΜΕ =
  $= \pi_1 + 2\pi_2 + 3\pi_3 = 2,491$
- Βαθμός χρησιμοποίησης κάθε μονάδας Ε/Ε =
  $= \pi_0 + \pi_1 + \frac{1}{2}\pi_2 = 0,245$

□

Βιβλιογραφία

[1] Bolch, G., Greiner, S., De Meer, H., and Trivedi, K.S., *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, Wiley-Interscience, 2006.

[2] Çinlar, E., *Introduction to Stochastic Processes*, Prentice–Hall, 1975.

[3] Fortier, P.J., and Michel, H.E., *Computer Systems Performance Evaluation and Prediction*, Elsevier Science, 2003.

[4] Gelenbe, E. and Mitrani, I., *Analysis and Synthesis of Computer Systems*, Academic Press, 1980.

[5] Harchol-Balter, M., *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, Cambridge University Press, 2013.

[6] Jain, R., *The Art of Computer Systems Performance Analysis - Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, 1991.

[7] Kleinrock, L., *Queueing Systems, Vol. I: Theory, Vol. II: Computer Applications*, John Wiley, 1975–76.

[8] Kobayashi, H., *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*, Addison–Wesley, 1978.

[9] Leung, C.H.C., *Quantitative Analysis of Computer Systems*, John Wiley & Sons, 1988.

[10] Markov, A.A., *Extension of the Limit Theorems of Probability Theory to a Sum of Variables Connected in a Chain*, The Notes of the Imperial Academy of Sciences of St. Petersburg VIII Series, Physio-Mathematical College, Vol. XXII, No. 9, Dec. 1907.

[11] Menasce, D.A., and Almeida, V.A.F., *Capacity Planning for Web Services: Metrics, Models and Methods*, Prentice-Hall, 2002.

[12] Menasce, D.A., Almeida, V.A.F., and Dowdy, L.W., *Performance by Design, Computer Capacity Planning by Example*, Prentice-Hall PTR, 2004.

[13] Obaidat, M.S., and Boudriga, N.A., *Performance Evaluation of Computer and Telecommunication Systems*, Wiley, 2010.

Κεφάλαιο 3

Απλά Συστήματα Αναμονής

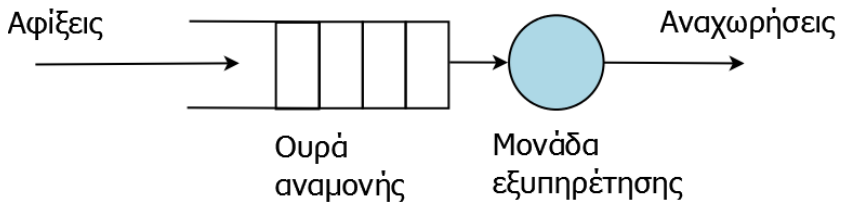
Σύνοψη
Εισάγονται τα βασικά στοιχεία της θεωρίας των ουρών αναμονής και αναπτύσσονται οι τεχνικές ανάλυσης απλών συστημάτων αναμονής (αποτελούμενων από μια ουρά). Εξετάζονται οι βασικές αρχές λειτουργίας των ουρών, οι συνθήκες ευστάθειας και ο Νόμος του Little. Περιγράφονται οι διάφορες επιλογές που αφορούν τον πληθυσμό, τον χώρο αναμονής και τον ρυθμό εξυπηρέτησης. Αναλύονται τα διάφορα μαρκοβιανά συστήματα αναμονής γεννήσεων-θανάτων (μοντέλα $M/M/1$, $M/M/c$, $M/M/\infty$, $M/M/1/K$, $M/M/1/K/K$ κλπ), καθώς και γενικότερα (μη μαρκοβιανά) μοντέλα, όπως το σύστημα $M/G/1$. Γίνεται αναφορά και σε άλλα μοντέλα με ειδικά χαρακτηριστικά (προτεραιότητες, ομαδικές αφίξεις, ομαδικές εξυπηρετήσεις κλπ). Αναπτύσσονται παραδείγματα μοντελοποίησης και ανάλυσης ενιαίων υπολογιστικών συστημάτων (system-level performance models).

Όπως αναφέρθηκε ήδη, η λειτουργία ενός υπολογιστικού συστήματος χαρακτηρίζεται από φαινόμενα αναμονής. Διαφόρων ειδών *πελάτες* (εργασίες, προγράμματα, διαδικασίες, μηνύματα κλπ.) αναγκάζονται να σχηματίσουν ουρές αναμονής, προκειμένου να εξυπηρετηθούν από διάφορους *σταθμούς εξυπηρέτησης* (ΚΜΕ, κύρια μνήμη, μονάδες εισόδου/εξόδου, μονάδες δευτερεύουσας μνήμης κλπ.). Η μελέτη των φαινομένων αυτών στηρίζεται σε μοντέλα της θεωρίας αναμονής [10, 14, 2, 16, 11]. Στο κεφάλαιο αυτό, θα αναφερθούμε σε απλά μοντέλα αναμονής (συστήματα με έναν σταθμό εξυπηρέτησης), ενώ τα μοντέλα δικτύων αναμονής θα εξεταστούν στο επόμενο κεφάλαιο. Τα απλά συστήματα αναμονής μπορούν να χρησιμοποιηθούν για τη μοντελοποίηση των υπολογιστικών συστημάτων ως ενιαίων οντοτήτων, χωρίς να περιγράφονται οι επιμέρους συνιστώσες τους. Για τον λόγο αυτό, —σε αντιδιαστολή προς τα μοντέλα δικτύων αναμονής— τα απλά μοντέλα αναμονής αναφέρονται και ως *μοντέλα επιπέδου συστήματος* (system-level models).

3.1 Βασικά Χαρακτηριστικά Συστημάτων Αναμονής

Η κλασική δομή ενός απλού συστήματος αναμονής απεικονίζεται στο Σχ. 3.1.

Μία ακολουθία *πελατών* φθάνει σε έναν *σταθμό εξυπηρέτησης* (service station) ο οποίος περιλαμβάνει μία ή περισσότερες *μονάδες εξυπηρέτησης* (servers). Αν ένας πελάτης που φθάνει στο σύστημα βρει



Σχήμα 3.1: *Απλό σύστημα αναμονής.*

όλες τις μονάδες εξυπηρέτησης απασχολημένες, περιμένει στην *ουρά* (queue) μέχρι να έρθει η στιγμή να εξυπηρετηθεί σύμφωνα με κάποιον *κανονισμό αναμονής* (queueing discipline) ή *αλγόριθμο χρονοδρομολόγησης* (scheduling algorithm). Όταν τελειώσει η εξυπηρέτησή του, ο πελάτης αναχωρεί από το σύστημα. Επομένως, τα τρία κύρια στοιχεία που χαρακτηρίζουν ένα τέτοιο σύστημα αναμονής θα είναι *η διαδικασία αφίξεων*, ο *μηχανισμός εξυπηρέτησης* και ο *κανονισμός αναμονής*.

Όσον αφορά τις αφίξεις, θεωρούμε ότι οι πελάτες προέρχονται από κάποιον *πληθυσμό*, δηλαδή από ένα σύνολο διακεκριμένων ατόμων, τα οποία θα μπορούσαν να επισκεφθούν το σύστημα. Ο πληθυσμός μπορεί να είναι πεπερασμένος ή άπειρος. Το δεύτερο σημείο που μας ενδιαφέρει είναι τα στατιστικά χαρακτηριστικά της διαδικασίας αφίξεων. Η πιο απλή και χρήσιμη περίπτωση είναι η τελείως τυχαία διαδικασία αφίξεων, με άλλα λόγια η διαδικασία Poisson.

Κάθε πελάτης που φθάνει στο σύστημα χαρακτηρίζεται από την ποσότητα εξυπηρέτησης που απαιτεί. Η μονάδα μέτρησης αυτής της ποσότητας ποικίλλει ανάλογα με τη φύση των πελατών και της εξυπηρέτησης. Θα μπορούσε να είναι εντολές, αν οι πελάτες είναι προγράμματα σε μία ΚΜΕ ή (bits), αν οι πελάτες είναι μηνύματα σε κάποιο κανάλι μετάδοσης. Στις περισσότερες περιπτώσεις, υποθέτουμε ότι οι απαιτήσεις εξυπηρέτησης των πελατών είναι τυχαίες μεταβλητές με κοινή κατανομή πιθανότητας, που την ονομάζουμε *κατανομή εξυπηρέτησης*. Εκείνο που ενδιαφέρει, όμως, είναι η χρονική διάρκεια της εξυπηρέτησης, η οποία εξαρτάται από τον *ρυθμό εργασίας* (processing rate) της μονάδας εξυπηρέτησης. Αν ο ρυθμός αυτός είναι C (μονάδες εξυπηρέτησης/sec) και S (μονάδες εξυπηρέτησης) είναι η ποσότητα εξυπηρέτησης που απαιτεί ένας πελάτης, τότε ο *χρόνος εξυπηρέτησης* (service time) του πελάτη θα είναι S/C (sec). Το αντίστροφο της μέσης τιμής του χρόνου εξυπηρέτησης $\mu = C/\overline{S}$ ονομάζεται *ρυθμός εξυπηρέτησης* (service rate). Αν ο ρυθμός εργασίας C είναι σταθερός, τότε δεν έχει ιδιαίτερη σημασία η διάκριση ανάμεσα στις ποσότητες S και S/C και μπορούμε να θέσουμε $C = 1$ χωρίς βλάβη της γενικότητας. Στη συνέχεια μπορούμε να θεωρούμε ότι οι απαιτήσεις εξυπηρέτησης των πελατών (κατανομή εξυπηρέτησης) εκφράζονται ως χρονικές διάρκειες.

Ο κανονισμός αναμονής καθορίζει τη σειρά με την οποία εξυπηρετούνται οι πελάτες που βρίσκονται στο σύστημα. Αναφέρουμε μερικούς από τους πιο συνηθισμένους κανονισμούς εξυπηρέτησης:

- FIFO (First In, First Out) ή FCFS (First Come, First Served): οι πελάτες εξυπηρετούνται σύμφωνα με τη σειρά άφιξής τους.
- LIFO (Last in, First Out) ή LCFS (Last Come, First Served): κάθε φορά εξυπηρετείται ο πελάτης με τον πιο πρόσφατο χρόνο άφιξης.
- FIRO (First In, Random Out): τυχαία σειρά εξυπηρέτησης των πελατών.
- Χρονοδρομολόγηση με προτεραιότητες (Priority Scheduling): οι πελάτες χωρίζονται σε κατηγορίες με διαφορετικές προτεραιότητες. Διακρίνουμε δύο γενικούς τύπους προτεραιοτήτων:
  - *Απλή προτεραιότητα ή προτεραιότητα χωρίς διακοπή* (Nonpreemptive): μετά από κάθε τέλος εξυπηρέτησης επιλέγεται για την επόμενη εξυπηρέτηση ο πελάτης με την υψηλότερη προτεραιότητα (μεταξύ πελατών με ίση προτεραιότητα ακολουθείται ο κανονισμός FCFS).
  - *Απόλυτη προτεραιότητα ή προτεραιότητα με διακοπή* (Preemptive): όταν ένας πελάτης που φθάνει στο σύστημα βρίσκει έναν πελάτη με χαμηλότερη προτεραιότητα να εξυπηρετείται, τον διακόπτει και αρχίζει τη δική του εξυπηρέτηση. Αργότερα, ο πελάτης που διακόπηκε, είτε συνεχίζει την εξυπηρέτησή του από το σημείο που την είχε διακόψει (Preemptive–resume), είτε ξαναρχίζει από την αρχή την ίδια εξυπηρέτηση (Preemptive–repeat without resampling), είτε αρχίζει μία νέα ανεξάρτητη εξυπηρέτηση (Preemptive–repeat with resampling).
- LCFSPR (Last Come, First Served–Preemptive–Resume): υπάρχει απόλυτη προτεραιότητα μεταξύ των πελατών αντίστροφη της σειράς άφιξής τους.
- RR (Round-Robin): είναι ένας από τους πιο διαδεδομένους αλγόριθμους χρονοδρομολόγησης για συστήματα καταμερισμού χρόνου (Time-Sharing). Οι πελάτες εξυπηρετούνται σε διάταξη FCFS, εφόσον ο χρόνος εξυπηρέτησής τους δεν ξεπερνά ένα σταθερό χρονικό διάστημα Q (quantum).

Όταν ο χρόνος εξυπηρέτησης φτάσει το Q, ο πελάτης διακόπτεται και τοποθετείται στο τέλος της ουράς. Η διαδικασία επαναλαμβάνεται μέχρι να συμπληρωθεί ο απαιτούμενος χρόνος εξυπηρέτησης. Στην οριακή περίπτωση $Q \to \infty$ ο αλγόριθμος συμπίπτει με τον FCFS, ενώ στην οριακή περίπτωση $Q \to 0$ έχουμε τον αλγόριθμο PS (Processor-Sharing) σύμφωνα με τον οποίο όσοι πελάτες βρίσκονται στο σύστημα εξυπηρετούνται εξίσου και ταυτόχρονα με ρυθμό αντιστρόφως ανάλογο προς τον αριθμό τους.

Παρατηρούμε ότι οι περισσότεροι κανονισμοί, αν και σχετίζονται με γενικά μοντέλα αναμονής, είναι άμεσα εμπνευσμένοι από τα υπολογιστικά συστήματα. Ενδεικτικά αναφέρουμε ακόμη τους αλγορίθμους FB (Foreground–Background), SPT (Shortest-Processing-Time-first), SEPT (Shortest-Expected-Processing-Time-first) και SRPT (Shortest-Remaining-Processing-Time-first).

Τα χαρακτηριστικά ενός συστήματος αναμονής μπορούν να παρασταθούν σύντομα με τη βοήθεια ενός απλού συμβολισμού, τον οποίο εισήγαγε πρώτος ο D.G.Kendall, $A/B/c/K/m/Z$, όπου:

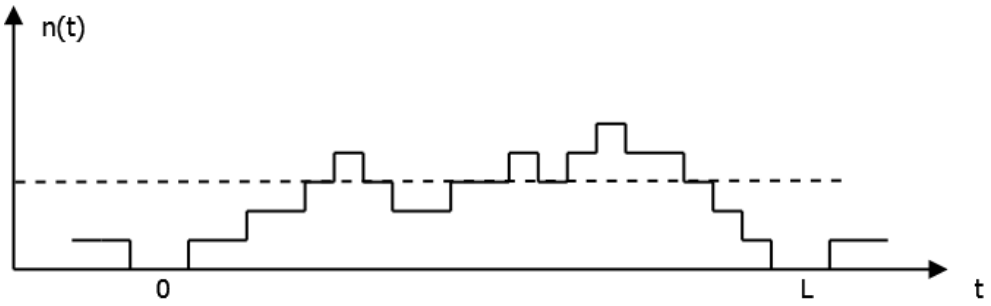
- A και B δηλώνουν αντίστοιχα την κατανομή των διαστημάτων μεταξύ αφίξεων και την κατανομή εξυπηρέτησης. Τα ακόλουθα σύμβολα χρησιμοποιούνται για την περιγραφή των κατανομών: M (εκθετική), E_k (Erlang-k), H_k (υπερεκθετική τάξης k), D (σταθερή), G (γενική), GI (γενική ανεξάρτητη).
- c είναι ο αριθμός των μονάδων εξυπηρέτησης στο σταθμό,
- K είναι η μέγιστη χωρητικότητα του συστήματος (περίπτωση πεπερασμένου χώρου αναμονής),
- m είναι το μέγεθος του πληθυσμού πελατών,
- Z είναι ένα από τα σύμβολα κανονισμών αναμονής, όπως ορίστηκαν πιο πάνω.

Όταν κάποιο από τα τρία τελευταία στοιχεία του συμβολισμού παραλείπεται, υπονοείται αντίστοιχα ότι ισχύει $K = \infty, m = \infty$ και Z=FCFS. Για παράδειγμα, στο σύστημα $D/M/2/20/50$ οι χρόνοι μεταξύ αφίξεων είναι σταθεροί (ντετερμινιστικοί), οι χρόνοι εξυπηρέτησης ακολουθούν εκθετική κατανομή, υπάρχουν 2 μονάδες εξυπηρέτησης, υπάρχει χώρος για 20 πελάτες, συνολικά 50 πελάτες μπορούν να επισκεφθούν το σύστημα, και ο κανονισμός αναμονής είναι FCFS.

3.2 Ντετερμινιστική Ανάλυση ενός Απλού Συστήματος Αναμονής

Η *ντετερμινιστική ή επιχειρησιακή ανάλυση* (operational analysis) είναι μία γενική μέθοδος για τη μελέτη των επιδόσεων συστημάτων σε δεδομένα χρονικά διαστήματα. Σε αντίθεση με τη θεωρία αναμονής, η επιχειρησιακή ανάλυση δεν στηρίζεται σε πιθανοτικές υποθέσεις, αλλά στη συμπεριφορά ποσοτήτων που μπορούν να μετρηθούν κατά τη λειτουργία ενός συστήματος. Η μέθοδος αυτή έχει εφαρμοστεί [3, 7] στη μελέτη φαινομένων αναμονής στα υπολογιστικά συστήματα, επιβεβαιώνοντας σε διαφορετικό πλαίσιο αποτελέσματα της θεωρίας αναμονής, αλλά και εισάγοντας νέα αποτελέσματα. Στην παράγραφο αυτή θα μελετήσουμε τη ντετερμινιστική συμπεριφορά ενός απλού συστήματος, για να διαπιστώσουμε στη συνέχεια ότι τα αποτελέσματα έχουν μία άμεση αντιστοιχία με αυτά που προκύπτουν από τη στοχαστική ανάλυση του ίδιου συστήματος [9].

Θεωρούμε μία ακολουθία από πελάτες αριθμημένους $1, 2, 3, \ldots$, οι οποίοι φθάνουν σε ένα απλό σύστημα αναμονής τις χρονικές στιγμές $a_1 < a_2 < a_3 < \ldots$ και αναχωρούν τις αντίστοιχες χρονικές στιγμές $c_i,\ i \geq 1$. Ο σταθμός εξυπηρέτησης περιλαμβάνει μία μονάδα εξυπηρέτησης, η οποία εξυπηρετεί τους πελάτες στην ουρά με κάποια τυχαία διάταξη. Παρατηρούμε την κατάσταση του συστήματος για ένα δεδομένο χρονικό διάστημα μήκους L, τέτοιο ώστε στην αρχή και στο τέλος του το σύστημα να είναι άδειο (Σχ. 3.2). Ως κατάσταση του συστήματος λαμβάνεται ο αριθμός $n(t)$ των πελατών στην ουρά (αναμονή ή εξυπηρέτηση) τη στιγμή t Για το χρονικό διάστημα παρατήρησης ορίζουμε τις ποσότητες:
$L(n)$, συνολικός χρόνος κατά τον οποίο το σύστημα βρίσκεται στην κατάσταση n,



Σχήμα 3.2: *Διάγραμμα αριθμού πελατών.*

$\alpha(n)$, αριθμός αφίξεων που γίνονται στην κατάσταση n,
$\gamma(n)$, αριθμός αναχωρήσεων που γίνονται στην κατάσταση n.

Εφόσον το σύστημα είναι άδειο στα άκρα του διαστήματος, θα ισχύει υποχρεωτικά:

$$\alpha(n) = \gamma(n+1) \tag{3.1}$$

Έστω $p(n)$ το ποσοστό του χρόνου στο διάστημα παρατήρησης κατά το οποίο το σύστημα βρίσκεται στην κατάσταση n:

$$p(n) = \frac{L(n)}{L} \tag{3.2}$$

οπότε με χρήση της Εξίσωσης (3.1) έχουμε:

$$p(n)\frac{\alpha(n)}{L(n)} = p(n+1)\frac{\gamma(n+1)}{L(n+1)} \tag{3.3}$$

Για να απλοποιήσουμε τους συμβολισμούς ορίζουμε τον ρυθμό αφίξεων $\lambda(n)$ και τον ρυθμό αναχωρήσεων $\mu(n)$ όταν το σύστημα βρίσκεται στην κατάσταση n:

$$\lambda(n) = \alpha(n)/L(n) \tag{3.4}$$
$$\mu(n) = \gamma(n)/L(n) \tag{3.5}$$

οπότε παίρνουμε τελικά:

$$p(n+1) = p(n)\frac{\lambda(n)}{\mu(n+1)},\ n = 0, 1, 2, \ldots \tag{3.6}$$

ή

$$p(n) = p(0)\prod_{i=0}^{n-1}\frac{\lambda(i)}{\mu(i+1)},\ n = 1, 2, \ldots \tag{3.7}$$

Η ποσότητα $p(0)$ μπορεί να υπολογιστεί εύκολα από τη σχέση:

$$\sum_{n=0}^{\infty} p(n) = \sum_{n=0}^{\infty}\frac{L(n)}{L} = 1 \tag{3.8}$$

από την οποία έχουμε:

$$p(0) = \left[1 + \sum_{n=1}^{\infty}\prod_{i=0}^{n-1}\frac{\lambda(i)}{\mu(i+1)}\right]^{-1} \tag{3.9}$$

Πρακτικά, οι δυνατές τιμές του n φράσσονται από τη μέγιστη τιμή που σημειώνεται στο διάστημα παρατήρησης.

Επομένως, τα ποσοστά χρόνου που πέρασε το σύστημα στις καταστάσεις $n = 0, 1, 2, \ldots$ υπολογίζονται από τις σχέσεις (3.7) και (3.9), οι οποίες θα ισχύουν για κάθε χρονικό διάστημα που αρχίζει και τελειώνει

με «άδεια» ουρά. Οι ποσότητες $\lambda(n), \mu(n)$ μπορούν να μετρηθούν στο χρονικό διάστημα παρατήρησης. Θα πρέπει, όμως, να τονίσουμε ότι οι ποσότητες $p(n)$ χαρακτηρίζουν το διάστημα αυτό, αλλά δεν έχουν καμία ισχύ πρόβλεψης έξω από το συγκεκριμένο διάστημα.

Συνεχίζοντας την ανάλυση του συστήματος, θα καταλήξουμε σε ένα αποτέλεσμα πολύ γενικό και εξαιρετικά χρήσιμο, τον **τύπο του Little**. Ας συμβολίσουμε με $a(t)$ και $c(t)$ αντίστοιχα τον αριθμό των αφίξεων και τον αριθμό των αναχωρήσεων που πραγματοποιήθηκαν μέχρι τη στιγμή t, υποθέτοντας $a(0) = 0$. Θα ισχύει επομένως:

$$n(t) = a(t) - c(t) \tag{3.10}$$

Θεωρούμε και πάλι το διάστημα παρατήρησης μήκους L, και θέτουμε τη χρονική στιγμή 0 στην αρχή του διαστήματος. Ο μέσος αριθμός πελατών στο σύστημα για το διάστημα αυτό θα είναι:

$$\mathrm{E}[n] = \frac{1}{L}\int_0^L n(t)\,\mathrm{d}t \tag{3.11}$$

Ορίζουμε τώρα τον *χρόνο απόκρισης* του i-στού πελάτη, δηλαδή τον χρόνο που περνάει ο i-στός πελάτης στο σύστημα (αναμονή και εξυπηρέτηση):

$$T_i = c_i - a_i \tag{3.12}$$

Αν υποθέσουμε ότι ακριβώς K αφίξεις (άρα και K αναχωρήσεις) έγιναν στο διάστημα παρατήρησης, τότε ο μέσος χρόνος απόκρισης των πελατών στο διάστημα αυτό θα είναι:

$$T = \mathrm{E}[T_i] = \frac{1}{K}\sum_{i=1}^{K}(c_i - a_i) \tag{3.13}$$

Έστω, τέλος, λ ο μέσος ρυθμός αφίξεων στο διάστημα μήκους L:

$$\lambda = K/L \tag{3.14}$$

θα έχουμε από τις δύο τελευταίες σχέσεις:

$$\lambda T = \frac{1}{L}\sum_{i=1}^{K}(c_i - a_i) \tag{3.15}$$

Μπορούμε, όμως, εύκολα να επαληθεύσουμε στο Σχ. 3.2 ότι ισχύει:

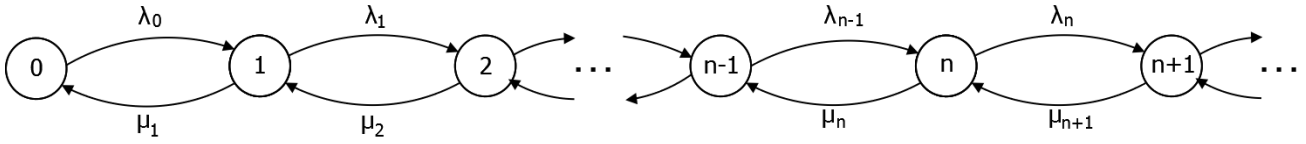
$$\sum_{i=1}^{K}(c_i - a_i) = \int_0^L (a(t) - c(t))\,\mathrm{d}t = \int_0^L n(t)\,\mathrm{d}t \tag{3.16}$$

οπότε με χρήση της Εξίσωσης (3.11) παίρνουμε τελικά τη σχέση:

$$\mathrm{E}[n] = \lambda T \tag{3.17}$$

η οποία δίνει τον *τύπο του Little* στο πλαίσιο της ντετερμινιστικής ανάλυσης.

Το αποτέλεσμα αυτό ήταν γνωστό από παλιά ως εμπειρικός κανόνας και αποδείχθηκε για πρώτη φορά από τον J.D.C. Little (1961) [15] για στοχαστικά συστήματα αναμονής. Η ισχύς του ειναι τελείως γενική και μπορεί να εφαρμοστεί σε οποιοδήποτε σύστημα χωρίς ιδιαίτερους περιορισμούς όσον αφορά τα χαρακτηριστικά του. Η βασική προϋπόθεση για την εφαρμογή του τύπου είναι ότι η στοχαστική διαδικασία $\{n(t), t \geq 0\}$, που περιγράφει τον αριθμό πελατών στο σύστημα, πρέπει να είναι μια *αναγεννητική διαδικασία* (regenerative process) [4]. Για μια τέτοια διαδικασία υπάρχει μία ακολουθία χρονικών στιγμών $t_0, t_1, \ldots$, τέτοια ώστε για κάθε στιγμή t_i, το μέλλον της διαδικασίας να είναι ανεξάρτητο από το παρελθόν. Οι στιγμές αυτές ονομάζονται *σημεία αναγέννησης* (regeneration points) της διαδικασίας. Στις περισσότερες περιπτώσεις συστημάτων αναμονής, ως σημεία αναγέννησης μπορούν να θεωρηθούν οι χρονικές στιγμές αφίξεων που βρίσκουν το σύστημα άδειο.



Σχήμα 3.3: *Διαδικασία γεννήσεων–θανάτων — Γράφος μεταβάσεων.*

3.3 Μοντέλα Αναμονής Γεννήσεων–Θανάτων

Όπως αναφέρθηκε νωρίτερα, μία *διαδικασία γεννήσεων–θανάτων* είναι μία ειδική περίπτωση αλυσίδας Markov, στην οποία μεταβάσεις γίνονται μόνο μεταξύ γειτονικών καταστάσεων. Μία τέτοια διαδικασία είναι ιδιαίτερα χρήσιμη ως μοντέλο των μεταβολών ενός πληθυσμού. Πράγματι, θεωρώντας ότι ο διακριτός χώρος καταστάσεων $0, 1, 2, \ldots$ παριστάνει το μέγεθος ενός πληθυσμού, οι επιτρεπτές μεταβάσεις από την κατάσταση n στις καταστάσεις $n+1$ ή $n-1$ δηλώνουν αντίστοιχα μία «γέννηση» ή ένα «θάνατο» μέσα στον πληθυσμό. Ενδιαφερόμαστε κυρίως για τις διαδικασίες γεννήσεων–θανάτων συνεχούς χρόνου.

Καταρχάς, εισάγουμε τις παραμέτρους λ_n και μ_n, οι οποίες αντιπροσωπεύουν τον *ρυθμό γεννήσεων* και τον *ρυθμό θανάτων*, αντίστοιχα, όταν η διαδικασία βρίσκεται στην κατάσταση n. Οι ρυθμοί λ_n, μ_n είναι ανεξάρτητοι του χρόνου, άρα η αλυσίδα Markov είναι ομοιογενής. Ακολουθώντας τους συμβολισμούς για τις αλυσίδες Markov συνεχούς χρόνου μπορούμε να γράψουμε:

$$\left.\begin{array}{l} \lambda_n = q_{n,n+1} \\ \mu_n = q_{n,n-1} \\ q_{nn} = -(\lambda_n + \mu_n) \end{array}\right\} \tag{3.18}$$

εφόσον ο ορισμός των διαδικασιών γεννήσεων–θανάτων επιβάλλει ότι $q_{nj} = 0$ για $|n-j| > 1$. Επίσης είναι προφανές ότι $\mu_0 = 0$. (Εύκολα μπορεί κανείς να διαπιστώσει ότι η διαδικασία Poisson είναι μία καθαρή διαδικασία γεννήσεων με $\mu_n = 0$ και $\lambda_n = \lambda$ για όλα τα n).

Στο Σχ. 3.3 παριστάνεται ο γράφος μεταβάσεων για τη διαδικασία γεννήσεων–θανάτων.

Ονομάζουμε $p_n(t)$ την πιθανότητα να βρίσκεται η διαδικασία στην κατάσταση n τη χρονική στιγμή t. Ενδιαφερόμαστε για την ύπαρξη των οριακών πιθανοτήτων:

$$p_n \triangleq \lim_{t\to\infty} p_n(t) \tag{3.19}$$

Αν υποθέσουμε προς το παρόν ότι η κατανομή p_n υπάρχει και αντιπροσωπεύει τη στατική κατανομή για τη διαδικασία, τότε θα πρέπει να ισχύουν οι εξισώσεις:

$$-(\lambda_n + \mu_n)p_n + \lambda_{n-1}p_{n-1} + \mu_{n+1}p_{n+1} = 0,\ n \geq 1 \tag{3.20}$$

$$-\lambda_0 p_0 + \mu_1 p_1 = 0 \tag{3.21}$$

Οι εξισώσεις αυτές αντιστοιχούν απλά στο σύστημα $\boldsymbol{\pi Q} = 0$ (Εξίσωση 2.72), το οποίο πρέπει να ικανοποιούν οι στατικές πιθανότητες μιας εργοδικής αλυσίδας Markov συνεχούς χρόνου.

Επιπλέον, θα πρέπει να ισχύει και η σχέση κανονικοποίησης:

$$\sum_{n=0}^{\infty} p_n = 1 \tag{3.22}$$

Στο σημείο αυτό θα ήταν χρήσιμο να κάνουμε μια διαφορετική ερμηνεία των Εξισώσεων (3.20) και (3.21). Οι εξισώσεις αυτές περιγράφουν τη δυναμική εξέλιξη της διαδικασίας και θα μπορούσαμε να πούμε ότι εκφράζουν τη «ροή» της πιθανότητας μέσα από τις διάφορες καταστάσεις. Παρατηρώντας τα βέλη στις ακμές του γράφου (Σχ. 3.3) μπορούμε να θεωρήσουμε ότι:

$$\begin{array}{rcl} \text{Ρυθμός ροής προς την κατάσταση } n & = & \lambda_{n-1}p_{n-1} + \mu_{n+1}p_{n+1} \\ \text{Ρυθμός ροής από την κατάσταση } n & = & (\lambda_n + \mu_n)p_n \end{array}$$

Εφόσον ενδιαφερόμαστε για τη μόνιμη (στατική) κατάσταση της διαδικασίας, το σύστημα πρέπει να βρίσκεται σε ισορροπία και οι ρυθμοί αυτοί να είναι ίσοι, πράγμα το οποίο εκφράζεται απο την Εξίσωση (3.20). Για τον λόγο αυτό, οι Εξισώσεις (3.20), (3.21), ονομάζονται *εξισώσεις ισορροπίας* του συστήματος. Η έννοια της ισορροπίας της ροής δεν ισχύει μόνο για κάθε κατάσταση, αλλά για οποιοδήποτε σύνολο καταστάσεων, θεωρώντας τις ροές από και προς το συγκεκριμένο σύνολο. Για παράδειγμα, αν θεωρήσουμε το σύνολο των καταστάσεων $\{0, 1, \ldots, n-1\}$, η ισορροπία της ροής δίνει:

$$\lambda_{n-1} p_{n-1} = \mu_n p_n \tag{3.23}$$

Εύκολα διαπιστώνει κανείς ότι το σύνολο των εξισώσεων που προκύπτουν από την Εξίσωση (3.23) για $n = 1, 2, \ldots$ είναι ισοδύναμο με το σύνολο των εξισώσεων που περιγράφεται από τις (3.20), (3.21). Η απλή αυτή τεχνική επισκόπησης είναι ιδιαίτερα χρήσιμη στην ανάλυση των διαδικασιών Markov.

Η γενική λύση του συστήματος εξισώσεων βρίσκεται εύκολα:

$$p_n = p_0 \prod_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}},\ n = 1, 2, \ldots \tag{3.24}$$

όπου η πιθανότητα p_0 προσδιορίζεται με βάση τη συνθήκη (3.22):

$$p_0 = \left[1 + \sum_{n=1}^{\infty} \prod_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}} \right]^{-1} \tag{3.25}$$

Οι στατικές πιθανότητες p_n θα υπάρχουν εάν ισχύει:

$$S = 1 + \sum_{n=1}^{\infty} \prod_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}} < \infty \tag{3.26}$$

δηλαδή, εάν ισχύει $p_0 > 0$. Η σημασία της τελευταίας συνθήκης γίνεται πιο εμφανής διαισθητικά, αν θεωρήσουμε ότι η διαδικασία γεννήσεων–θανάτων περιγράφει ένα απλό σύστημα αναμονής, όπου οι γεννήσεις και οι θάνατοι αντιπροσωπεύουν αντίστοιχα αφίξεις και αναχωρήσεις πελατών. Η συνθήκη $p_0 > 0$ εκφράζει απλά ότι για να είναι μια ουρά ευσταθής θα πρέπει κατά καιρούς να αδειάζει. Από την (3.26) παρατηρούμε ότι η σειρά S συγκλίνει αν υπάρχει κάποιο n_0 τέτοιο, ώστε για κάθε $n \geq n_0$ να ισχύει $\lambda_n/\mu_n < 1$, ή, σε σχέση με το σύστημα αναμονής, οι αφίξεις να γίνονται με μικρότερο ρυθμό απο τις αναχωρήσεις.

Τέλος, θα πρέπει να παρατηρήσουμε την ομοιότητα των (3.24) και (3.25) που αναφέρονται σε μια στοχαστική διαδικασία, με τις (3.7) και (3.9) που υπολογίστηκαν σε ένα καθαρά ντετερμινιστικό πλαίσιο.

Στη συνέχεια, θα εφαρμόσουμε τη γενική λύση (3.24), (3.25), σε ορισμένες από τις πιο σημαντικές περιπτώσεις απλών συστημάτων αναμονής.

3.3.1 Το Σύστημα $M/M/1$

Το σύστημα $M/M/1$ είναι το πιο απλό σύστημα αναμονής και χαρακτηρίζεται από αφίξεις Poisson (εκθετικά κατανεμημένους χρόνους μεταξύ αφίξεων) και εκθετικά κατανεμημένους χρόνους εξυπηρέτησης. Οι ρυθμοί γεννήσεων–θανάτων είναι ανεξάρτητοι από την κατάσταση του συστήματος:

$$\left.\begin{array}{ll} \lambda_n = \lambda, & n = 0, 1, 2, \ldots \\ \mu_n = \mu, & n = 1, 2, \ldots \end{array}\right\} \tag{3.27}$$

Αντικαθιστώντας τους ρυθμούς στη γενική λύση βρίσκουμε:

$$p_n = p_0 \left(\frac{\lambda}{\mu}\right)^n,\ n \geq 0 \tag{3.28}$$

$$p_0 = \left[\sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^n \right]^{-1} \tag{3.29}$$

εφόσον ισχύει η συνθήκη:

$$S = \sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^n < \infty \tag{3.30}$$

Για $\lambda < \mu$ η σειρά S συγκλίνει προς την τιμή $1/(1-\lambda/\mu)$, οπότε:

$$p_0 = 1 - \lambda/\mu \tag{3.31}$$

Η ποσότητα $\rho = \lambda/\mu$ ονομάζεται *ένταση κυκλοφορίας* (traffic intensity) του συστήματος. Σύμφωνα με την Εξίσωση (3.31) η ένταση κυκλοφορίας εκφράζει την πιθανότητα να μην είναι άδειο το σύστημα ή ισοδύναμα να είναι απασχολημένη η μονάδα εξυπηρέτησης, γι' αυτό ονομάζεται και *βαθμός χρησιμοποίησης* (utilization) της μονάδας εξυπηρέτησης. Με χρήση του ρ, η στατική πιθανότητα να υπάρχουν n πελάτες στο σύστημα θα είναι:

$$p_n = (1-\rho)\rho^n,\ n = 0, 1, 2, \ldots \tag{3.32}$$

δηλαδή, ο αριθμός πελατών στο σύστημα ακολουθεί γεωμετρική κατανομή.

Ο *μέσος όρος πελατών στο σύστημα* θα είναι:

$$\mathrm{E}[n] = \sum_{n=0}^{\infty} n p_n = (1-\rho) \sum_{n=0}^{\infty} n\rho^n$$

ή

$$\mathrm{E}[n] = \frac{\rho}{1-\rho} \tag{3.33}$$

και, με βάση τον τύπο του Little, ο *μέσος χρόνος παραμονής* στο σύστημα (*μέσος χρόνος απόκρισης*) θα είναι

$$T = \mathrm{E}[n]/\lambda = \frac{1}{\mu} \cdot \frac{1}{1-\rho} \tag{3.34}$$

Ο χρόνος παραμονής ενός πελάτη στο σύστημα είναι το άθροισμα του χρόνου αναμονής και του χρόνου εξυπηρέτησης. Άρα ο *μέσος χρόνος αναμονής* θα είναι

$$W = T - \frac{1}{\mu} = \frac{1}{\mu} \cdot \frac{\rho}{1-\rho} \tag{3.35}$$

και, με νέα εφαρμογή του τύπου του Little, ο *μέσος αριθμός πελατών σε αναμονή* θα είναι:

$$N_W = \lambda W = \frac{\rho^2}{1-\rho} \tag{3.36}$$

Ένα ενδιαφέρον αποτέλεσμα είναι η συνάρτηση κατανομής πιθανότητας του χρόνου απόκρισης στο σύστημα $M/M/1$, την οποία παραθέτουμε χωρίς απόδειξη (βλ. [12, 13, 8, 1]):

$$F_T(x) = \Pr[T \le x] = 1 - e^{-(\mu-\lambda)x} = 1 - e^{-\mu(1-\rho)x},\ x \ge 0 \tag{3.37}$$

δηλαδή ο χρόνος απόκρισης είναι εκθετικά κατανεμημένος.

Παράδειγμα 3.1. *Σε ένα μικρό υπολογιστικό σύστημα ομαδικής επεξεργασίας (Batch system) ο χρόνος επεξεργασίας ανά εργασία είναι εκθετικά κατανεμημένος με μέση τιμή* 3 sec. *Οι εργασίες φθάνουν σύμφωνα με μια διαδικασία Poisson με μέσο ρυθμό μία εργασία κάθε* 4 sec *και εξυπηρετούνται με κανονισμό FCFS. Ζητούνται η πιθανότητα ο χρόνος απόκρισης μίας εργασίας να ξεπερνά τα* 20 sec *και ο μέσος αριθμός εργασιών σε αναμονή.*

Αποφασίστηκε ότι, όταν ο φόρτος εργασίας φθάσει σε επίπεδο τέτοιο, ώστε ο μέσος χρόνος απόκρισης να γίνει 30 sec, *θα πρέπει να αυξηθούν οι δυνατότητες του συστήματος. Κατά πόσο θα έχει αυξηθεί ο σημερινός ρυθμός αφίξεων όταν θα συμβεί αυτό;*

- Έχουμε $\lambda = \frac{1}{4\text{ sec}} = 0,25$ εργασίες/sec και $\frac{1}{\mu} = 3$ sec, οπότε $\rho = \frac{\lambda}{\mu} = 0,75$. Σύμφωνα με την (3.37), η ζητούμενη πιθανότητα είναι:
  $\Pr[T > 20\text{ sec}] = e^{-\mu(1-\rho)\cdot 20\text{ sec}} = e^{-\frac{1}{3}\cdot 0,25\cdot 20} = 0,1889$

- Από την (3.36) ο μέσος αριθμός πελατών σε αναμονή θα είναι:
  $N_W = \frac{\rho^2}{1-\rho} = \frac{0,75^2}{0,25} = 2,25$ εργασίες

- Ο μέσος χρόνος απόκρισης δίνεται από την (3.34):

$$T = \frac{1}{\mu}\frac{1}{1-\rho} = \frac{1}{\mu - \lambda}$$

  Θεωρώντας ότι ο μέσος χρόνος εξυπηρέτησης παραμένει ο ίδιος βρίσκουμε: $30 = \frac{1}{1/3-\lambda}$ ή $\lambda = 0,3$ εργασίες/sec. Επομένως, θα πρέπει να γίνει αναβάθμιση του συστήματος όταν ο φόρτος εργασίας αυξηθεί κατά 20%.

□

Παράδειγμα 3.2. *Σε ένα υπολογιστικό κέντρο διατίθεται ένας μεγάλος υπολογιστής για την εξυπηρέτηση on-line εφαρμογών διαφόρων χρηστών που είναι γεωγραφικά διασκορπισμένοι. Οι αφίξεις αιτήσεων εξυπηρέτησης προς τον κεντρικό υπολογιστή ακολουθούν διαδικασία Poisson, ενώ οι χρόνοι εξυπηρέτησης είναι κατανεμημένοι εκθετικά. Έγινε μία πρόταση να μοιραστεί ο φόρτος εργασίας εξίσου σε n μικρότερους υπολογιστές, ο καθένας από τους οποίους θα διαθέτει το $1/n$ της υπολογιστικής ισχύος του σημερινού. Συμφέρει η πραγματοποίηση της πρότασης;*

Έστω λ, μ οι ρυθμοί αφίξεων και εξυπηρετήσεων για το σημερινό σύστημα και $\rho = \lambda/\mu$ ο βαθμός χρησιμοποίησης του υπολογιστή. Για καθένα από τα προτεινόμενα συστήματα, ο ρυθμός αφίξεων θα είναι λ/n και ο ρυθμός εξυπηρέτησης μ/n, οπότε ο βαθμός χρησιμοποίησης παραμένει ο ίδιος. Αν συγκρίνουμε όμως τους αντίστοιχους μέσους χρόνους απόκρισης, θα έχουμε σύμφωνα με την (3.34):

$$\frac{T_{\text{προτεινόμενο}}}{T_{\text{σημερινό}}} = \frac{\frac{n/\mu}{(1-\rho)}}{\frac{1/\mu}{(1-\rho)}} = n$$

δηλαδή ο μέσος χρόνος απόκρισης στο προτεινόμενο σύστημα θα είναι n φορές μεγαλύτερος από το σημερινό! Αν και τα n μικρά συστήματα θα εξυπηρετούν τον ίδιο αριθμό αιτήσεων ανά μονάδα χρόνου με το σημερινό σύστημα, κάθε αίτηση θα χρειάζεται κατά μέσο όρο n φορές περισσότερο χρόνο για να εξυπηρετηθεί. Το φαινόμενο αυτό είναι γνωστό ως *φαινόμενο κλίμακας* (scaling effect) και εκφράζει την αρχή ότι η συγκέντρωση υπολογιστικής ισχύος βελτιώνει τον χρόνο απόκρισης. □

Παράδειγμα 3.3. *Σε ένα κέντρο μεταγωγής μηνυμάτων ο μέσος ρυθμός αφίξεων σε μία επικοινωνιακή γραμμή είναι* 240 *μηνύματα ανά* min. *Η ταχύτητα μετάδοσης της γραμμής είναι* 800 bytes/sec *και η κατανομή του μήκους των μηνυμάτων είναι εκθετική με μέσο μήκος* 176 bytes. *Ζητούνται τα κύρια μέτρα επιδοσεων του συστήματος υποθέτοντας ότι υπάρχει πολύ μεγάλος αριθμός διαθέσιμων ενταμιευτών (buffers). Ποια είναι η πιθανότητα να περιμένουν στην ουρά τουλάχιστον* 10 *μηνύματα;*

Ο μέσος χρόνος εξυπηρέτησης θα είναι ο μέσος χρόνος μετάδοσης ενός μηνύματος:

$$\frac{1}{\mu} = \frac{176\text{ bytes}}{800\text{ bytes/sec}} = 0,22\text{ sec}$$

Ο ρυθμός αφίξεων είναι $\lambda = 240$ μηνύματα/min=4 μηνύματα/sec οπότε $\rho = \lambda/\mu = 0,88$.

Χρησιμοποιώντας τα αποτελέσματα της ουράς $M/M/1$ βρίσκουμε:
$\mathrm{E}[n] = \frac{\rho}{1-\rho} = 7,33$ μηνύματα (Μέσος αριθμός μηνυμάτων στο σύστημα).

$N_W = \frac{\rho^2}{1-\rho} = 6,45$ μηνύματα (Μέσος αριθμός μηνυμάτων σε αναμονή μετάδοσης).
$T = \frac{1}{\mu}\frac{1}{1-\rho} = 1,83$ sec (Μέσος χρόνος παραμονής μηνυμάτων στο σύστημα).
$W = \frac{1}{\mu}\frac{\rho}{1-\rho} = 1,61$ sec (Μέσος χρόνος αναμονής μηνυμάτων).
Από την (3.32) η κατανομή του αριθμού μηνυμάτων στο σύστημα είναι $p_n = (1-\rho)\rho^n,\ n = 0, 1, 2, \ldots$ Ζητάμε την πιθανότητα να περιμένουν τουλάχιστον 10 μηνύματα στην ουρά, άρα να βρίσκονται τουλάχιστον 11 μηνύματα στο σύστημα:

$$\Pr[n \geq 11] = \sum_{n=11}^{\infty}(1-\rho)\rho^n = \rho^{11}\cdot\sum_{n=0}^{\infty}(1-\rho)\cdot\rho^n = \rho^{11} = 0,245$$

□

3.3.2 Το Σύστημα $M/M/c$

Θεωρούμε τώρα το σύστημα με c ίδιες μονάδες εξυπηρέτησης. Έχουμε:

$$\left.\begin{array}{l} \lambda_n = \lambda,\ n = 0, 1, 2, \ldots \\ \mu_n = \min(n\mu, c\mu) = \left[\begin{array}{l} n\mu,\ n = 1, 2, \ldots, c \\ c\mu,\ n \geq c \end{array}\right. \end{array}\right\} \tag{3.38}$$

Η εφαρμογή της γενικής λύσης δίνει τη στατική κατανομή του αριθμού πελατών στο σύστημα:

$$p_n = \left\{\begin{array}{ll} p_0\frac{(c\rho)^n}{n!}, & n \leq c \\ p_0\frac{\rho^n c^c}{c!}, & n \geq c \end{array}\right. \tag{3.39}$$

όπου $\rho = \lambda/(c\mu) < 1$. Η ποσότητα αυτή εκφράζει το αναμενόμενο ποσοστό απασχολημένων μονάδων εξυπηρέτησης ή ισοδύναμα το βαθμό χρησιμοποίησης κάθε μονάδας εξυπηρέτησης. Τέλος, η πιθανότητα p_0 θα είναι:

$$p_0 = \left[\sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!(1-\rho)}\right]^{-1} \tag{3.40}$$

Ένα χρήσιμο μέτρο επίδοσης είναι η πιθανότητα $\Pr[n \geq c]$, δηλαδή η πιθανότητα ένας πελάτης που φθάνει στο σύστημα να χρειαστεί να περιμένει (τύπος C του Ερλανγ).

Γενικότερα, οι εκφράσεις για τα διάφορα μέτρα επιδόσεων του συστήματος $M/M/c$ είναι μάλλον πολύπλοκες [1]. Ένα σύστημα $M/M/c$ έχει κατώτερες επιδόσεις από ένα σύστημα $M/M/1$ με ισοδύναμο ρυθμό εξυπηρέτησης, δηλαδή με ρυθμό εξυπηρέτησης $c\mu$. Αυτό συμβαίνει γιατί στο σύστημα $M/M/c$ δεν χρησιμοποιούνται πάντα όλες οι δυνατότητές του (όταν υπάρχουν λιγότεροι από c πελάτες, κάποιες μονάδες εξυπηρέτησης δεν εργάζονται). Αντίθετα, το σύστημα $M/M/c$ έχει καλύτερες επιδόσεις από το σύστημα c ανεξάρτητων μονάδων εξυπηρέτησης με ξεχωριστές ουρές (όπως στο Παράδειγμα 3.2).

3.3.3 Το Σύστημα $M/M/\infty$ (Άπειρες μονάδες εξυπηρέτησης)

Στο σύστημα αυτό μπορούμε να θεωρήσουμε είτε ότι υπάρχει μία μονάδα εξυπηρέτησης, η οποία επιταχύνει το ρυθμό της γραμμικά όσο έρχονται περισσότεροι πελάτες, είτε ως την περίπτωση όπου διατίθεται πάντα μία καινούρια μονάδα για κάθε πελάτη που φθάνει στο σύστημα. Εδώ, ο αριθμός πελατών στο σύστημα ισοδυναμεί με τον αριθμό πελατών που εξυπηρετούνται, εφόσον δεν υπάρχει αναμονή. Έχουμε:

$$\left.\begin{array}{l} \lambda_n = \lambda,\ n = 0, 1, 2, \ldots \\ \mu_n = n\mu,\ n = 1, 2, \ldots \end{array}\right\} \tag{3.41}$$

Η επίλυση των εξισώσεων ισορροπίας δίνει:

$$p_n = \frac{\rho^n}{n!}p_0,\ n = 0, 1, 2 \tag{3.42}$$

όπου $\rho = \lambda/\mu$. Από την εξίσωση κανονικοποίησης παίρνουμε:

$$p_0 = \left[\sum_{n=0}^{\infty} \frac{\rho^n}{n!}\right]^{-1} = e^{-\rho} \tag{3.43}$$

Επομένως, η στατική κατανομή υπάρχει πάντα και έχει τη μορφή κατανομής Poisson. Εφόσον δεν υπάρχει συνθήκη σύγκλισης, το ρ μπορεί να πάρει και τιμές μεγαλύτερες του 1. Ο μέσος αριθμός πελατών στο σύστημα είναι:

$$\mathrm{E}[n] = \rho \tag{3.44}$$

ενώ ο μέσος χρόνος απόκρισης θα είναι $T = 1/\mu$, ίσος με το μέσο χρόνο εξυπηρέτησης. Στην ουσία, εφόσον δεν υπάρχει αναμονή στον σταθμό, κάθε πελάτης υφίσταται καθυστέρηση ανεξάρτητα από τους άλλους. Για τον λόγο αυτό, ένα σύστημα αναμονής με άπειρες μονάδες εξυπηρέτησης αναφέρεται ως *σταθμός καθυστέρησης* (delay station).

3.3.4 Το Σύστημα $M/M/1/K$ (Πεπερασμένος χώρος αναμονής)

Το μοντέλο του Παραδείγματος 3.3 δεν φαίνεται ιδιαίτερα ρεαλιστικό, γιατί κανένα σύστημα δεν μπορεί να διαθέτει άπειρο αριθμό ενταμιευτών. Το σύστημα $M/M/1/K$, στο οποίο ένας πελάτης μπορεί να εξυπηρετείται και το πολύ $K-1$ να περιμένουν, αποτελεί ακριβέστερο μοντέλο τέτοιου είδους συστημάτων. Υποθέτουμε ότι οι πελάτες που φθάνοντας βρίσκουν K πελάτες στο σύστημα φεύγουν και «χάνονται». Οι ρυθμοί γεννήσεων–θανάτων θα είναι:

$$\left.\begin{aligned} \lambda_n &= \begin{cases} \lambda & , \quad n < K \\ 0 & , \quad n \geq K \end{cases} \\ \mu_n &= \begin{cases} \mu & , \quad n \leq K \\ 0 & , \quad n > K \end{cases} \end{aligned}\right\} \tag{3.45}$$

Παρατηρούμε ότι πρόκειται για αμείωτη αλυσίδα Markov με πεπερασμένο αριθμό καταστάσεων, η οποία είναι πάντα εργοδική, και επομένως υπάρχει πάντα η στατική κατανομή πιθανότητας:

$$p_n = p_0 \rho^n,\ n = 0, 1, 2, \ldots, K \tag{3.46}$$

όπου $\rho = \lambda/\mu$. Εύκολα βρίσκουμε ότι:

$$p_0 = \frac{1-\rho}{1-\rho^{K+1}},\ \rho \neq 1 \tag{3.47}$$

Για την περίπτωση $\lambda = \mu$ ή $\rho = 1$ ισχύει:

$$p_n = \frac{1}{K+1},\ n = 0, 1, 2, \ldots, K \tag{3.48}$$

Παράδειγμα 3.4. *Θεωρούμε το κέντρο μεταγωγής μηνυμάτων του* Παραδ. 3.3*, με τα ίδια χαρακτηριστικά. Επιθυμούμε, όμως, να διατεθούν τόσοι buffers, ώστε η πιθανότητα να είναι όλοι γεμάτοι κάποια στιγμή να είναι μικρότερη από* $0,005$*. Ζητείται ο απαιτούμενος αριθμός buffers.*

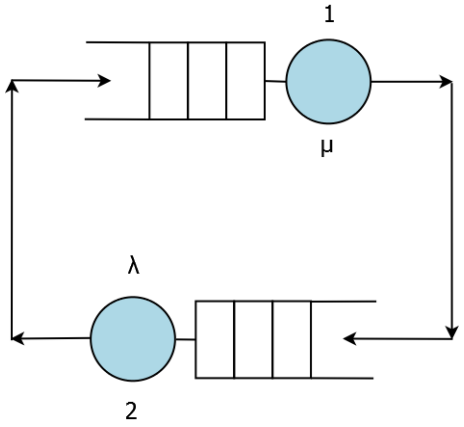
Έχουμε σύστημα $M/M/1/K$ με $\rho = 0,88$. Στο σύστημα θα υπάρχουν το πολύ K μηνύματα, από τα οποία ένα εξυπηρετείται (μεταδίδεται) και $K-1$ είναι αποθηκευμένα σε buffers. Άρα η πιθανότητα να είναι όλοι οι buffers γεμάτοι (δεδομένου ότι διατίθενται $K-1$ buffers) θα είναι σύμφωνα με τις (3.46), (3.47):

$$p_K = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}},\ \text{όπου } \rho = 0,88.$$

Με δοκιμές βρίσκουμε ότι ισχύει:

$$p_{26} = 0,004464 < 0,005 < p_{25} = 0,005095$$

Άρα χρειάζονται 25 ενταμιευτές. □



Σχήμα 3.4: *Κλειστό κυκλικό σύστημα.*

Παράδειγμα 3.5. *Το μοντέλο $M/M/1/K$ παρουσιάζει ιδιαίτερο ενδιαφέρον, γιατί είναι ισοδύναμο με το εξής κλειστό κυκλικό σύστημα: K πελάτες κυκλοφορούν συνεχώς ανάμεσα σε δύο σταθμούς εξυπηρέτησης, 1 και 2, στους οποίους οι χρόνοι εξυπηρέτησης είναι εκθετικά κατανεμημένοι με μέσες τιμές $1/\mu$ και $1/\lambda$ αντίστοιχα, και η σειρά εξυπηρέτησης FCFS* (Σχ. 3.4).

Η ισοδυναμία ανάμεσα στο σύστημα αυτό και το σύστημα $M/M/1/K$ είναι προφανής, αν παρατηρήσουμε ότι, όσο ο αριθμός πελατών στο σταθμό 1 είναι μικρότερος από K, φθάνουν στο σταθμό 1 πελάτες με ρυθμό λ. Οι αφίξεις σταματούν όταν οι K πελάτες είναι όλοι στο σταθμό 1.

Το κυκλικό αυτό μοντέλο μπορεί να εφαρμοστεί σε ένα υπολογιστικό σύστημα αποτελούμενο από μία ΚΜΕ και μία μονάδα Ε/Ε, στο οποίο K εργασίες μοιράζονται την κύρια μνήμη. □

3.3.5 Το Σύστημα $M/M/c/c$

Στο σύστημα αυτό γίνονται δεκτοί μόνο οι πελάτες που βρίσκουν διαθέσιμη μονάδα εξυπηρέτησης, διαφορετικά οι πελάτες χάνονται. Κλασική εφαρμογή ενός τέτοιου μοντέλου είναι ένα τηλεφωνικό κέντρο με c γραμμές. Για το σύστημα αυτό οι ρυθμοί γεννήσεων–θανάτων είναι:

$$\left.\begin{aligned} \lambda_n &= \begin{cases} \lambda & n < c \\ 0 & n \geq c \end{cases} \\ \mu_n &= \begin{cases} n\mu & n \leq c \\ 0 & n > c \end{cases} \end{aligned}\right\} \tag{3.49}$$

Και στην περίπτωση αυτή, η αλυσίδα Markov είναι πάντα εργοδική, οπότε η στατική κατανομή πιθανότητας βρίσκεται:

$$p_n = p_0 \frac{\rho^n}{n!},\ n = 0, 1, 2, \ldots, c \tag{3.50}$$

θέτοντας, όπως συνήθως $\rho = \lambda/\mu$. Η πιθανότητα p_0 δίνεται από τη σχέση:

$$p_0 = \left[\sum_{i=0}^{c} \frac{\rho^i}{i!} \right]^{-1} \tag{3.51}$$

Ιδιαίτερο ενδιαφέρον παρουσιάζει η πιθανότητα p_c να είναι απασχολημένες όλες οι μονάδες εξυπηρέτησης, η οποία ισοδυναμεί με το ποσοστό των πελατών που χάνονται:

$$p_c = \frac{\rho^c/c!}{1 + \rho + \rho^2/2! + \ldots + \rho^c/c!} \tag{3.52}$$

Η σχέση (3.52) είναι γνωστή στην τηλεφωνία ως ο τύπος B ή τύπος *απώλειας* του Erlang (loss formula) και υπολογίστηκε για πρώτη φορά από τον Δανό μαθηματικό και μηχανικό A.K. Erlang το 1917.

3.3.6 Το Σύστημα $M/M/1/K/K$ (Επισκευή μηχανών)

Το μοντέλο αυτό, στο οποίο ο πληθυσμός των πελατών είναι πεπερασμένος, είναι γνωστό ως το μοντέλο επισκευής μηχανών (machine repair model) και αποτελεί ένα από τα πιο χρήσιμα μοντέλα της θεωρίας αναμονής. Υπάρχουν συνολικά K πελάτες, καθένας από τους οποίους είτε βρίσκεται στο σύστημα, είτε κατά κάποιο τρόπο βρίσκεται «καθ' οδόν» προς το σύστημα. Μπορούμε να φανταστούμε ότι οι πελάτες είναι K ίδιες μηχανές και κάθε μηχανή παθαίνει κατά διαστήματα βλάβες. Το διάστημα λειτουργίας μεταξύ βλαβών είναι εκθετικά κατανεμημένο με παράμετρο α. Οι μηχανές επισκευάζονται από ένα επισκευαστή με εκθετικά κατανεμημένο χρόνο επισκευής μέσης τιμής $1/\mu$. Όταν βρίσκονται n μηχανές στο σύστημα (σε αναμονή ή εξυπηρέτηση), ο ρυθμός αφίξεων θα είναι $(K-n)\alpha$, εφόσον κάθε μηχανή λειτουργεί ανεξάρτητα. Έτσι οι ρυθμοί γεννήσεων–θανάτων θα είναι:

$$\left.\begin{array}{l} \lambda = \left[\begin{array}{ccc} (K-n)\alpha & , & n \le K \\ 0 & , & n > K \end{array}\right. \\ \mu_n = \mu,\ n = 1, 2, \ldots, K \end{array}\right\} \tag{3.53}$$

Η αλυσίδα Markov είναι πάντα εργοδική και η στατική κατανομή πιθανότητας δίνεται από τις σχέσεις:

$$p_n = p_0 \frac{K!}{(K-n)!} \left(\frac{\alpha}{\mu}\right)^n,\ n = 0, 1, 2, \ldots, K \tag{3.54}$$

$$p_0 = \left[\sum_{n=0}^{K} \frac{K!}{(K-n)!} \left(\frac{\alpha}{\mu}\right)^n\right]^{-1} \tag{3.55}$$

Στο σημείο αυτό, είναι ενδιαφέρον να δείξουμε ένα αποτέλεσμα που ισχύει γενικά για κάθε σύστημα αναμονής $G/G/1$ σε ισορροπία, και το οποίο θα είναι χρήσιμο στη συνέχεια της ανάλυσης.

Αν σε ένα γενικό σύστημα ο μέσος ρυθμός αφίξεων είναι λ και ο μέσος χρόνος εξυπηρέτησης είναι S, τότε ο βαθμός χρησιμοποίησης του συστήματος ορίζεται:

$$\rho = \lambda S \tag{3.56}$$

σε αντιστοιχία με το συμβολισμό που χρησιμοποιήσαμε ως τώρα. Αν θεωρήσουμε ένα πολύ μεγάλο χρονικό διάστημα L, τότε ο συνολικός αριθμός αφίξεων στο διάστημα αυτό θα είναι περίπου λL. Επίσης, έστω p_0 η πιθανότητα το σύστημα να είναι άδειο κάποια τυχαία χρονική στιγμή. Μπορούμε, λοιπόν, να πούμε ότι στο διάστημα L η μονάδα εξυπηρέτησης ήταν απασχολημένη για χρόνο $L - Lp_0$, και ο αριθμός πελατών που εξυπηρετήθηκαν είναι περίπου $(L - Lp_0)/S$. Εφόσον το σύστημα είναι σε ισορροπία θα πρέπει για μεγάλο L όσοι πελάτες ήρθαν να εξυπηρετήθηκαν, άρα μπορούμε να θεωρήσουμε ότι:

$$\lambda L \cong (L - Lp_0)/S \tag{3.57}$$

από την οποία με χρήση της (3.56) παίρνουμε:

$$\rho = 1 - p_0 \tag{3.58}$$

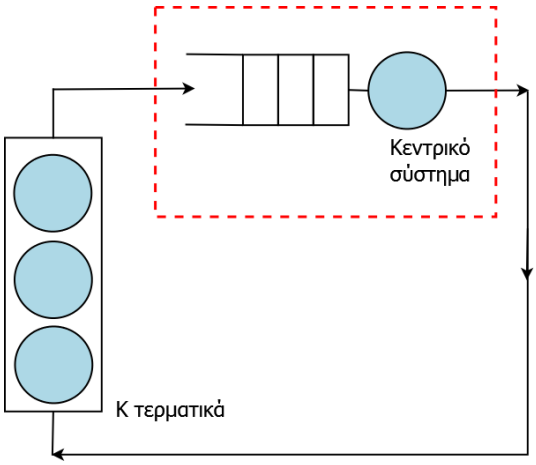
Η σχέση (3.58) επιβεβαιώνει ότι η ποσότητα ρ είναι το ποσοστό του χρόνου κατά το οποίο η μονάδα εξυπηρέτησης είναι απασχολημένη.

Επιστρέφοντας στο σύστημα $M/M/1/K/K$, συμπεραίνουμε με βάση τις (3.55), (3.58) ότι ο βαθμός χρησιμοποίησης του επισκευαστή είναι:

$$\rho = 1 - \frac{1}{\sum_{n=0}^{K} \frac{K!}{(K-n)!} \left(\frac{\alpha}{\mu}\right)^n} \tag{3.59}$$

Σύμφωνα με την (3.58) ο πραγματικός μέσος ρυθμός αφίξεων στο σύστημα θα είναι:

$$\lambda = \rho\mu \tag{3.60}$$



Σχήμα 3.5: *Μοντέλο κεντρικού συστήματος.*

Για τον υπολογισμό του μέσου χρόνου αναμονής W σκεπτόμαστε ως εξής: για κάθε μηχανή ένας πλήρης κύκλος αποτελείται από την περίοδο λειτουργίας, τον χρόνο αναμονής και τον χρόνο εξυπηρέτησης. Έτσι ο μέσος ρυθμός με τον οποίο οι K μηχανές παθαίνουν βλάβη (δηλαδή ο μέσος ρυθμός αφίξεων στο σύστημα) θα είναι:

$$\lambda = K/(1/\alpha + W + 1/\mu) \tag{3.61}$$

ή

$$W = K/\lambda - 1/\alpha - 1/\mu \tag{3.62}$$

Με χρήση του τύπου του Little βρίσκουμε το μέσο αριθμό μηχανών σε αναμονή:

$$N_W = K - \lambda/\alpha - \lambda/\mu \tag{3.63}$$

Επίσης, ο μέσος χρόνος απόκρισης και ο μέσος αριθμός μηχανών στο σύστημα θα είναι αντίστοιχα:

$$T = W + 1/\mu = K/\lambda - 1/\alpha \tag{3.64}$$

$$\mathrm{E}[n] = K - \lambda/\alpha \tag{3.65}$$

Τέλος, η πιθανότητα μία οποιαδήποτε μηχανή να μη λειτουργεί θα είναι:

$$\Pr[\text{η μηχανή } i \text{ δεν λειτουργεί}] = T/(T + 1/\alpha) \tag{3.66}$$

Ανάλογα αποτελέσματα μπορούν να υπολογιστούν και για το σύστημα $M/M/c/K/K$ (μοντέλο επισκευής μηχανών με πολλούς επισκευαστές) [1].

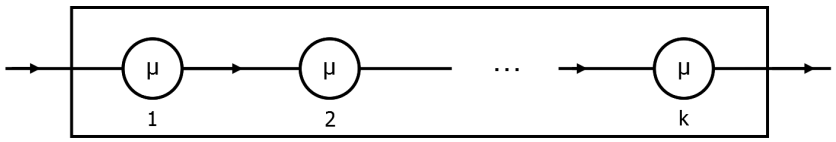
Παράδειγμα 3.6. *Το σύστημα $M/M/1/K/K$ μπορεί να χρησιμοποιηθεί ως ένα απλό μοντέλο διαλογικού (interactive) υπολογιστικού συστήματος με K τερματικά* (Σχ. 3.5).

Κάθε πελάτης (χρήστης) βρίσκεται κάθε στιγμή σε μία από τις εξής τρεις καταστάσεις: (1) *σε «σκέψη» μπροστά στο τερματικό,* (2) *σε αναμονή εξυπηρέτησης και* (3) *σε εξυπηρέτηση από το κεντρικό σύστημα. Έτσι ο χρήστης δεν μπορεί να ζητήσει εξυπηρέτηση αν η προηγούμενη αίτησή του δεν έχει ικανοποιηθεί. Αν ο χρόνος σκέψης και ο χρόνος εξυπηρέτησης είναι εκθετικά κατανεμημένοι με αντίστοιχες παραμέτρους α και μ, το σύστημα μπορεί να περιγραφεί από το μοντέλο επισκευής μηχανών με έναν επισκευαστή.*

Ας θεωρήσουμε ένα τέτοιο σύστημα με 20 *τερματικά, στο οποίο ο μέσος χρόνος σκέψης είναι* 20 sec *και ο μέσος χρόνος εξυπηρέτησης στην ΚΜΕ είναι* 2 sec. *Ζητούνται ο μέσος χρόνος απόκρισης και ο μέσος ρυθμός απόδοσης (throughput) του συστήματος. Ποια επίδραση θα είχε η προσθήκη* 5 *τερματικών;*

Έχουμε από την (3.55):

$$p_0 = \left[\sum_{n=0}^{20} \frac{20!}{(20-n)!}\left(\frac{2}{20}\right)^n\right]^{-1} = 0,001869$$



Σχήμα 3.6: *Κατανομή* Erlang k *σταδίων.*

οπότε
$\rho = 1 - p_0 = 0,998131$ (βαθμός χρησιμοποίησης)
$\lambda = 0,49907$ εργασίες/sec (ρυθμός απόδοσης)
$T = 20/0,49907 - 20 = 20,075$ sec (μέσος χρόνος απόκρισης)

Για 25 τερματικά παίρνουμε αντίστοιχα:
$p_0 = 0,00002927,\ \rho = 0,99997073,$
$\lambda = 0,499985365$ εργασίες/sec και $T = 30$ sec
Επομένως, η προσθήκη 5 τερματικών αυξάνει το μέσο ρυθμό απόδοσης μόνο κατά 0,18%, ενώ ταυτόχρονα αυξάνει το μέσο χρόνο απόκρισης κατά 49,44%. □

3.4 Άλλα Μαρκοβιανά Μοντέλα Αναμονής

Τα μοντέλα αναμονής που εξετάστηκαν στη προηγούμενη παράγραφο χαρακτηρίζονται από χρόνους μεταξύ αφίξεων και χρόνους εξυπηρέτησης κατανεμημένους εκθετικά, πράγμα το οποίο μας επιτρέπει να επωφεληθούμε από την απλότητα των διαδικασιών γεννήσεων–θανάτων. Πριν αναφερθούμε σε συστήματα με γενικές κατανομές, θα εξετάσουμε σύντομα ορισμένες ειδικές κατανομές, οι οποίες επιτρέπουν την ανάλυση των αντίστοιχων μοντέλων με τη βοήθεια των αλυσίδων Markov.

Αναφερθήκαμε νωρίτερα στην κατανομή Erlang-k, η οποία εκφράζει την κατανομή του αθροίσματος k ανεξάρτητων τυχαίων μεταβλητών που ακολουθούν την ίδια εκθετική κατανομή. Αν υποθέσουμε ότι ο χρόνος εξυπηρέτησης σε ένα σύστημα αναμονής ακολουθεί κατανομή Erlang-k, μπορούμε να παραστήσουμε την κατάσταση όπως στο Σχ. 3.6.

Σύμφωνα με τη παράσταση αυτή, ένας πελάτης θα πρέπει να διασχίσει k στάδια εξυπηρέτησης, καθένα από τα οποία είναι κατανεμημένο εκθετικά με παράμετρο μ. Κάθε στιγμή, μόνο ένας πελάτης μπορεί να βρίσκεται σε κάποιο από τα k στάδια της εξυπηρέτησης. Σύμφωνα με την ερμηνεία αυτή, αν έχουμε αφίξεις Poisson (σύστημα $M/E_k/1$), η συμπεριφορά του συστήματος μπορεί να περιγραφεί από μία αλυσίδα Markov συνεχούς χρόνου. Η κατάσταση της αλυσίδας Markov θα παριστάνεται από το ζεύγος (n, j), όπου n ο αριθμός πελατών στο σύστημα και j ο αριθμός του σταδίου στο οποίο βρίσκεται ο πελάτης που εξυπηρετείται. Μία ισοδύναμη περιγραφή θα προέκυπτε, αν θεωρούσαμε ότι κάθε πελάτης που φθάνει στο σύστημα ισοδυναμεί με k στάδια εξυπηρέτησης. Έτσι θα μπορούσαμε αντί για την προηγούμενη διδιάστατη περιγραφή, να θεωρήσουμε ως κατάσταση του συστήματος τον αριθμό σταδίων εξυπηρέτησης που απομένουν κάθε στιγμή.

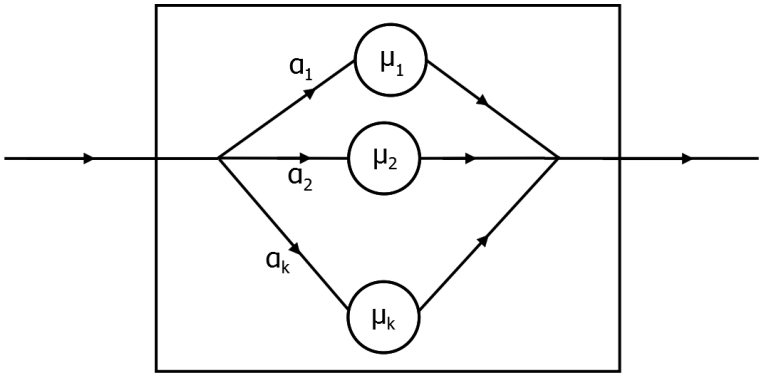
Μία τελείως αντίστοιχη περιγραφή προκύπτει στην περίπτωση που τα διαστήματα μεταξύ αφίξεων ακολουθούν κατανομή Erlang-k, αν φανταστούμε ότι κάθε τέτοιο διάστημα ισοδυναμεί με τη διάσχιση k *σταδίων άφιξης*. Για την πλήρη ανάλυση των συστημάτων αναμονής $M/E_k/1$ και $E_k/M/1$ παραπέμπουμε στο [12]. Θυμίζουμε ότι η κατανομή Erlang-k έχει συνάρτηση πυκνότητας πιθανότητας:

$$f(x) = \frac{\mu(\mu x)^{k-1}}{(k-1)!} e^{-\mu x},\ x \geq 0 \tag{3.67}$$

με μετασχηματισμό Laplace:

$$\Phi(s) = \left(\frac{\mu}{\mu + s}\right)^k \tag{3.68}$$

Συχνά η κατανομή του χρόνου εξυπηρέτησης μπορεί να παρασταθεί ως «μείγμα» εκθετικών κατανομών. Έστω ότι υπάρχουν k τύποι πελατών που εμφανίζονται με πιθανότητες $\alpha_i,\ i = 1, 2, \ldots, k$, και κάθε πελάτης



Σχήμα 3.7: *Υπερεκθετική κατανομή k σταδίων.*

τύπου i έχει χρόνο εξυπηρέτησης κατανεμημένο εκθετικά με μέση τιμή $1/\mu_i$, $i = 1, 2, \ldots, k$. Προκύπτει η *υπερεκθετική κατανομή τάξης* k (συμβολίζεται H_k) με συνάρτηση πυκνότητας πιθανότητας:

$$f(x) = \sum_{i=1}^{k} \alpha_i \mu_i e^{-\mu_i x},\ x > 0 \tag{3.69}$$

της οποίας ο μετασχηματισμός Laplace είναι:

$$\Phi(s) = \sum_{i=1}^{k} \alpha_i \frac{\mu_i}{\mu_i + s} \tag{3.70}$$

Και στην περίπτωση αυτή, ο χρόνος εξυπηρέτησης μπορεί να παρασταθεί με τη βοήθεια k εκθετικών σταδίων, μόνο που εδώ τα στάδια δεν είναι στη σειρά αλλά παράλληλα, και κάθε φορά επιλέγεται κάποιο από τα k στάδια σύμφωνα με τις πιθανότητες α_i (Σχ. 3.7). Κάθε στιγμή μόνο ένας πελάτης μπορεί να βρίσκεται σε κάποιο από το k στάδια.

Συνδυασμός των προηγούμενων περιπτώσεων μας οδηγεί στη *γενικευμένη μέθοδο των σταδίων*, σύμφωνα με την οποία η αντίστοιχη κατανομή παριστάνεται από k παράλληλες γραμμές, κάθε μία από τις οποίες περιλαμβάνει έναν αριθμό σταδίων στη σειρά (Σχ. 3.8).

Ο μετασχηματισμός Laplace για την κατανομή αυτή θα είναι:

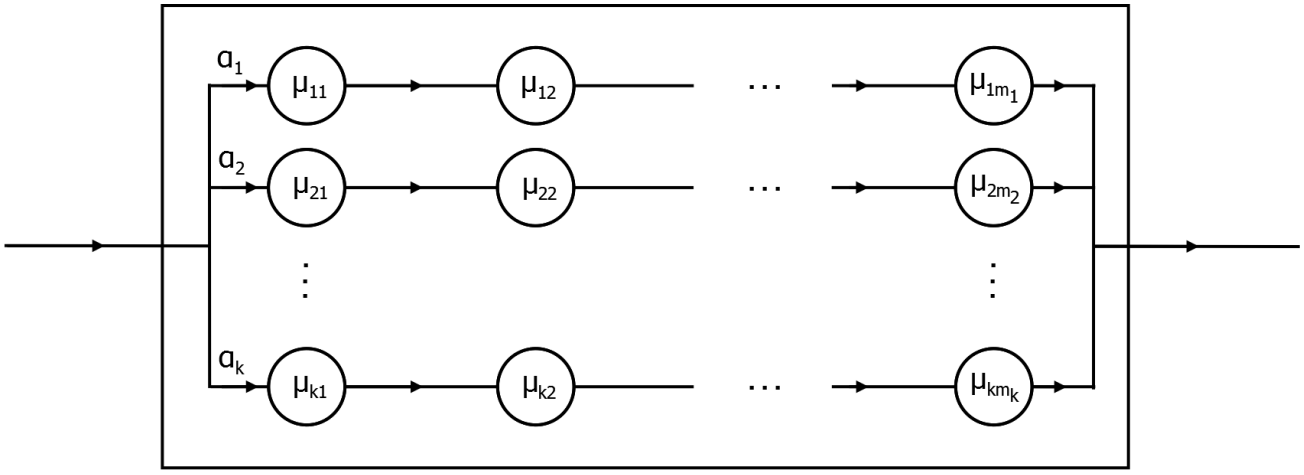
$$\Phi(s) = \sum_{i=1}^{k} \alpha_i \prod_{j=1}^{m_i} \frac{\mu_{ij}}{\mu_{ij} + s} \tag{3.71}$$

Οι κατανομές αυτές ανήκουν σε μία γενική κατηγορία κατανομών, οι οποίες είναι γνωστές ως *κατανομές Cox* [6] και επιτρέπουν την ανάλυση των αντίστοιχων συστημάτων αναμονής με βάση τη θεωρία των διαδικασιών Markov. Οι κατανομές αυτές έχουν ιδιαίτερη χρησιμότητα στη μελέτη των δικτύων αναμονής.

Τελειώνοντας, αναφέρουμε δύο ειδικές περιπτώσεις μαρκοβιανών συστημάτων αναμονής με ιδιαίτερο ενδιαφέρον, η μελέτη των οποίων χρησιμοποιεί αναλυτικές μεθόδους ανάλογες με αυτές που χρησιμοποιούνται στην παράσταση με εκθετικά στάδια. Πρόκειται για συστήματα με ομαδικές αφίξεις (Bulk arrivals) ή ομαδική εξυπηρέτηση (Bulk service), των οποίων λεπτομερής ανάλυση μπορεί να βρεθεί στο [12].

3.5 Το Σύστημα Αναμονής $M/G/1$

Κλείνοντας το κεφάλαιο, θα εξετάσουμε την περίπτωση του συστήματος αναμονής $M/G/1$ (αφίξεις Poisson, γενική κατανομή του χρόνου εξυπηρέτησης s), του οποίου η συμπεριφορά δεν μπορεί να περιγραφεί από μία διαδικασία Markov. Επομένως, η ανάλυσή του απαιτεί τη χρήση άλλων μεθόδων [12, 4, 13, 1, 5], στις οποίες θα αναφερθούμε συνοπτικά εστιάζοντας στις βασικές ιδέες.



Σχήμα 3.8: *Γενικευμένη μέθοδος των σταδίων.*

Ήδη στην προηγούμενη παράγραφο αναφέρθηκε η *μέθοδος των σταδίων*, σύμφωνα με την οποία, αν ο μετασχηματισμός Laplace μιας γενικής κατανομής είναι ρητή συνάρτηση, μπορούμε να εφαρμόσουμε τεχνικές από τη θεωρία των διαδικασιών Markov.

Μία δεύτερη μέθοδος ανάλυσης θα ήταν η ειδική εφαρμογή της θεωρίας που έχει αναπτυχθεί για τη γενική περίπτωση του συστήματος $G/G/1$ (*Ολοκληρωτική εξίσωση του Lindley* [12]).

Τρίτο, αν θεωρήσουμε το σύστημα $M/G/1$, παρατηρούμε ότι για να περιγράψουμε πλήρως την κατάστασή του τη χρονική στιγμή t, ώστε να συνοψίζεται στην περιγραφή αυτή ολόκληρο το παρελθόν του συστήματος χρειαζόμαστε δύο μεταβλητές: τον αριθμό πελατών $N(t)$ στο σύστημα και το χρόνο εξυπηρέτησης $X_0(t)$, τον οποίο έχει ήδη λάβει ο πελάτης που εξυπηρετείται τη στιγμή t. (Εξαιτίας της ιδιότητας έλλειψης μνήμης, η δεύτερη μεταβλητή δεν χρειάζεται για την περιγραφή του συστήματος $M/M/1$). Επομένως, η στοχαστική διαδικασία $N(t)$ δεν είναι διαδικασία Markov. Αντίθετα, η διαδικασία $[N(t), X_0(t)]$ είναι διαδικασία Markov δύο διαστάσεων, η δεύτερη, όμως, από τις οποίες περιγράφεται από συνεχή χώρο καταστάσεων. Με χρήση της κατάλληλης θεωρίας μπορεί να γίνει ανάλυση του συστήματος βασισμένη στην περιγραφή $[N(t), X_0(t)]$. Αυτή η μέθοδος επίλυσης είναι γνωστή ως η μέθοδος των *συμπληρωματικών μεταβλητών*.

Μία τέταρτη μέθοδος επίλυσης είναι η μέθοδος της ενσωματωμένης αλυσίδας Markov (imbedded Markov chain). Η έννοια της ενσωματωμένης αλυσίδας Markov αναφέρθηκε στο προηγούμενο κεφάλαιο σε σχέση με τις ημιμαρκοβιανές διαδικασίες και συνδέεται με την παρατήρηση του συστήματος σε κατάλληλα επιλεγμένες χρονικές στιγμές. Η μεθοδολογία ανάλυσης του συστήματος $M/G/1$ ισχύει και για το σύστημα $G/M/1$ (και γενικότερα το σύστημα $G/M/c$),το οποίο παρουσιάζει εντελώς αντίστοιχη συμπεριφορά [12, 1].

Η ανάλυση του συστήματος $M/G/1$ βασίζεται σε δύο σημαντικές ιδιότητες, οι οποίες σχετίζονται με τη χρήση της μεθόδου της ενσωματωμένης αλυσίδας Markov.

- Σε κάθε σύστημα αναμονής με αφίξεις Poisson ισχύει $p_n(t) = r_n(t)$, όπου $p_n(t)$ είναι η πιθανότητα το σύστημα να βρίσκεται στην κατάσταση n την στιγμή t και $r_n(t)$ είναι η πιθανότητα ένας πελάτης που φθάνει τη στιγμή t να βρει το σύστημα στην κατάσταση n. Η ιδιότητα αυτή, που ισχύει για κάθε t, θα ισχύει και για τις αντίστοιχες κατανομές στη μόνιμη κατάσταση. Συμπεραίνουμε ότι στη διαδικασία Poisson κάθε πελάτης συμπεριφέρεται τη στιγμή της άφιξής του ως *τυχαίος* παρατηρητής του συστήματος.

- Η δεύτερη ιδιότητα μας λέει ότι, εάν σε κάποιο σύστημα αναμονής ο αριθμός πελατών μεταβάλλεται μόνο κατά βήματα μεγέθους συν ή πλην ένα, τότε στη μόνιμη κατάσταση (εφόσον υπάρχει) η πιθανότητα ένας πελάτης που αναχωρεί να αφήνει πίσω του n πελάτες είναι ίση με την πιθανότητα ένας πελάτης που φθάνει να βρει στο σύστημα n πελάτες. Διαισθητικά μπορούμε να ερμηνεύσουμε την ιδιότητα αυτή παρατηρώντας ότι, μετά από ένα πολύ μεγάλο χρονικό διάστημα λειτουργίας του συστήματος, ο αριθμός των μεταβάσεων από την κατάσταση n προς την κατάσταση $n+1$ θα πρέπει

να είναι ίσος με τον αριθμό μεταβάσεων από την $n+1$ προς την n. (Ένας αντίστοιχος συλλογισμός είχε χρησιμοποιηθεί στη ντετερμινιστική ανάλυση του συστήματος αναμονής).

Όπως θα δούμε στο επόμενο κεφάλαιο, οι ιδιότητες αυτές έχουν ιδιαίτερη σημασία στη θεμελίωση της θεωρίας των δικτύων αναμονής (Θεώρημα των αφίξεων, καθολική και τοπική ισορροπία).

Η επίλυση του μοντέλου $M/G/1$ μπορεί να οδηγήσει στην κατανομή πιθανότητας p_n στη μόνιμη κατάσταση. Επειδή κατά κανόνα ο υπολογισμός αυτός είναι δύσκολος, συνήθως αρκούμαστε στη μέση τιμή της κατανομής, η οποία μπορεί να υπολογιστεί με διάφορους τρόπους:

$$\mathrm{E}[n] = \rho + \frac{\lambda^2 \mathrm{E}\left[s^2\right]}{2(1-\rho)} = \rho + \frac{\rho^2(1+C_s^2)}{2(1-\rho)} \tag{3.72}$$

όπου
λ ο ρυθμός αφίξεων,
S ο μέσος χρόνος εξυπηρέτησης,
$\rho = \lambda S$ η ένταση κυκλοφορίας (συνθήκη ισορροπίας $\rho < 1$),
$C_s^2 = \frac{\mathrm{E}[s^2]}{S^2} - 1$ είναι το τετράγωνο του συντελεστή μεταβολής (coefficient of variation) του χρόνου εξυπηρέτησης s. (Το τετράγωνο του συντελεστή μεταβολής ισούται με το πηλίκο της διασποράς προς το τετράγωνο της μέσης τιμής.)

Η σχέση (3.72) είναι ένα από τα πιο χρήσιμα αποτελέσματα της θεωρίας αναμονής και είναι γνωστή ως ο τύπος των Pollaczek-Khinchine (P-K).

Τελειώνοντας, θα πρέπει να τονίσουμε το πλήθος εφαρμογών του μοντέλου $M/G/1$, καθώς και τον μεγάλο αριθμό θεωρητικών αποτελεσμάτων σχετικών με το σύστημα αυτό. Ιδιαίτερο ενδιαφέρον παρουσιάζουν ειδικές περιπτώσεις του μοντέλου, όπως τα συστήματα $M/E_k/1$, $M/H_k/1$, $M/D/1$ [12, 1], καθώς και επεκτάσεις όπως το μοντέλο $M/G/1$ με πεπερασμένη χωρητικότητα [17] (μοντέλο συστημάτων πολυπρογραμματισμού), το μοντέλο $M/G/1$ με κανονισμό Processor-Sharing και τα μοντέλα $M/G/1$ με διάφορους τύπους προτεραιοτήτων [12, 8, 1].

Παράδειγμα 3.7. *Σε ένα υπολογιστικό σύστημα είναι συνδεδεμένες* 4 *γραμμές επικοινωνίας, καθεμία από τις οποίες έχει μέσο χρόνο μετάδοσης ανά μήνυμα* 2,4 sec *και βαθμό χρησιμοποίησης* $0,8$. *Η κατανομή των χρόνων μετάδοσης μηνυμάτων είναι διαφορετική για κάθε γραμμή. Για την πρώτη γραμμή η κατανομή είναι υπερεκθετική με* $\alpha_1 = 0,4$, $\alpha_2 = 0,6$, $1/\mu_1 = 4,8$ sec *και* $1/\mu_2 = 0,8$ sec. *Για τις υπόλοιπες τρεις γραμμές η κατανομή είναι αντίστοιχα εκθετική,* Erlang-3 *και σταθερή. Να συγκριθούν οι μέσοι χρόνοι αναμονής των μηνυμάτων στις* 4 *γραμμές.*

Εφαρμόζουμε το μοντέλο $M/G/1$ για κάθε μία από τις γραμμές. Σε όλες τις περιπτώσεις έχουμε $S = 2,4$ sec και $\rho = 0,8$, οπότε $\lambda = 1/3$ μηνύματα/sec. Ο μέσος χρόνος αναμονής θα δίνεται από τη σχέση $W = T - S = \mathrm{E}[n]/\lambda - S$, όπου το $\mathrm{E}[n]$ προσδιορίζεται με χρήση του τύπου των Pollaczek-Khinchine.

- Γραμμή 1: Για την υπερεκθετική κατανομή έχουμε:

$$\mathrm{E}[S^2] = \alpha_1 \frac{2}{\mu_1^2} + \alpha_2 \frac{2}{\mu_2^2} = 19,2 \text{ sec}^2$$

  απ' όπου βρίσκουμε $W = 16,0$ sec.

- Γραμμή 2: Για την εκθετική κατανομή βρίσκουμε κατευθείαν από την Εξίσωση (3.35): $W = \frac{1}{\mu}\frac{\rho}{1-\rho} = 9,6$ sec.

- Γραμμή 3: Για την κατανομή Erlang-k έχουμε:
  $\mathrm{Var}[s] = \frac{\mathrm{E}[s]^2}{k}$, οπότε $\mathrm{E}[s^2] = \mathrm{Var}[s] + S^2 = \frac{k+1}{k}\mathrm{E}[s]^2$
  Άρα για την Erlang-3: $\mathrm{E}[s^2] = \frac{4}{3} \cdot 2,4^2 = 7,68$ και $W = 6,4$ sec.

- Γραμμή 4: Για σταθερή κατανομή έχουμε:
  $\mathrm{E}[s^2] = S^2 = 2,4^2 = 5,76$, οπότε W=4,8 sec.

Παρατηρούμε, στο παράδειγμα αυτό, την επίδραση της *ανωμαλίας* μιας κατανομής (της απομάκρυνσης των τιμών της από την μέση τιμή, όπως εκφράζεται από τον συντελεστή μεταβολής C_s) στον μέσο χρόνο αναμονής. Πράγματι, για τις 4 περιπτώσεις που εξετάσαμε, το τετράγωνο του συντελεστή μεταβολής C_s^2 έχει τις τιμές αντίστοιχα 2,33, 1, 1/3 και 0 με αποτέλεσμα αντίστοιχες διαφορές στο χρόνο αναμονής W. □

Βιβλιογραφία

[1] Allen, A.O., *Probability, Statistics and Queueing Theory with Computer Science Applications*, Academic Press, 1978.

[2] Bolch, G., Greiner, S., De Meer, H., and Trivedi, K.S., *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, Wiley-Interscience, 2006.

[3] Buzen, J.P., *Fundamental Operational Laws of Computer System Performance*, Acta Informatica, Vol. 7, 1976.

[4] Çinlar, E., *Introduction to Stochastic Processes*, Prentice–Hall, 1975.

[5] Cohen, J.W., *The Single Server Queue*, North-Holland, 1969.

[6] Cox, D.R., Miller, H.D., *The Theory of Stochastic Processes*, Chapman and Hall, 1965.

[7] Denning, P.J. and Buzen, J.P., *The Operational Analysis of Queueing Network Models*, Computing Surveys, Vol. 10, No. 3, pp. 225-261, 1978.

[8] Gelenbe, E. and Mitrani, I., *Analysis and Synthesis of Computer Systems*, Academic Press, 1980.

[9] Gelenbe, E. and Pujolle, G., *Introduction to Queueing Networks*, John Wiley, 1987.

[10] Gross, D. and Harris, C., *Fundamentals of Queueing Theory*, John Wiley, 1985.

[11] Harchol-Balter, M., *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, Cambridge University Press, 2013.

[12] Kleinrock, L., *Queueing Systems, Vol. I: Theory, Vol. II: Computer Applications*, John Wiley, 1975–76.

[13] Kobayashi, H., *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*, Addison–Wesley, 1978.

[14] Leung, C.H.C., *Quantitative Analysis of Computer Systems*, John Wiley & Sons, 1988.

[15] Little, J.D.C., *A Proof of the Queueing Formula* $L = \lambda W$, Operations Research, Vol. 9, pp. 383–387, 1961.

[16] Obaidat, M.S., and Boudriga, N.A., *Performance Evaluation of Computer and Telecommunication Systems*, Wiley, 2010.

[17] Reiser, M., Kobayashi, H., *The Effects of Service Time Distributions on System Performance*, Information Processing, North-Holland, 1974.

Κεφάλαιο 4

Δίκτυα Αναμονής

Σύνοψη

Εξετάζονται τα σημαντικότερα μοντέλα δικτύων αναμονής, περιλαμβάνοντας τα ανοικτά και κλειστά δίκτυα Jackson, τις ιδιότητες της καθολικής και τοπικής ισορροπίας, το Θεώρημα BCMP και τα γενικά δίκτυα με λύση μορφής γινομένου (product-form). Αναπτύσσονται οι βασικοί επιχειρησιακοί νόμοι, η επιχειρησιακή ανάλυση και οι αρχές λειτουργίας της μεθόδου της Ανάλυσης Μέσης Τιμής (MVA). Δίνονται ο αλγόριθμος της συνέλιξης (convolution algorithm) και η ακριβής διατύπωση του αλγορίθμου MVA για μία κατηγορία και πολλές κατηγορίες πελατών, καθώς επίσης για σταθερούς ρυθμούς εξυπηρέτησης και μεταβλητούς ρυθμούς εξυπηρέτησης (εξαρτώμενους από το φορτίο). Γίνεται εφαρμογή των ανωτέρω στη μελέτη υπολογιστικών συστημάτων με μοντελοποίηση των συνιστωσών τους (component-level performance models). Αναπτύσσονται μοντέλα συστημάτων βασισμένων στον Ιστό, τόσο από την πλευρά του χρήστη (client-side models) όσο και από την πλευρά του εξυπηρετητή (server-side models).

Τα υπολογιστικά και τηλεπικοινωνιακά συστήματα, των οποίων οι επιδόσεις μάς ενδιαφέρουν συνήθως στην πράξη, είναι πολύπλοκα συστήματα πολλαπλών πόρων (multiple resource systems). Στα συστήματα αυτά, μια εργασία χρειάζεται διάφορα είδη εξυπηρέτησης από διαφορετικούς σταθμούς και, επομένως, μπορεί να χρειαστεί να περιμένει σε διάφορες ουρές αναμονής μέσα στο σύστημα ανταγωνιζόμενη τις υπόλοιπες εργασίες. Για παράδειγμα, σε ένα υπολογιστικό σύστημα πολυπρογραμματισμού οι εργασίες περιλαμβάνουν διάφορες επιμέρους δραστηριότητες στην ΚΜΕ και τις μονάδες I/O. Για την παράσταση ενός τέτοιου συστήματος θεωρούμε ένα δίκτυο σταθμών εξυπηρέτησης με χωριστή ουρά σε κάθε κόμβο. Οι πελάτες κινούνται από κόμβο σε κόμβο του δικτύου, για να εξυπηρετηθούν στους διάφορους σταθμούς, και στο τέλος ενδεχομένως εγκαταλείπουν το δίκτυο. Όπως είναι φανερό, ένα απλό σύστημα μιας ουράς αναμονής δεν θα επαρκούσε για την παράσταση και μελέτη ενός κατανεμημένου υπολογιστικού περιβάλλοντος, το οποίο απαιτεί τη χρήση μοντέλου σε επίπεδο συνιστωσών (Component-level performance model).

Η ανάλυση ενός δικτύου αναμονής λαμβάνει ως είσοδο δεδομένα που περιγράφουν το σύστημα και το φορτίο του και υπολογίζει ως έξοδο ένα σύνολο μέτρων επίδοσης. Η κατάλληλη χρήση μοντέλων δικτύων αναμονής παρέχει συνήθως ακρίβεια αποτελεσμάτων με χαμηλό κόστος, πράγμα που τα καθιστά ιδιαίτερα αποδοτική επιλογή στην ανάλυση επίδοσης.

Θα αναφερθούμε καταρχάς σε ορισμένα βασικά αποτελέσματα της θεωρίας των δικτύων αναμονής και, ειδικότερα, σε μοντέλα τα οποία επιδέχονται αναλυτική λύση κλειστής μορφής [8, 22, 2]. Στα μοντέλα αυτά βασίζονται κατά κύριο λόγο οι αναλυτικές τεχνικές, ακριβείς ή προσεγγιστικές, που χρησιμοποιούνται στην πράξη για τη μελέτη υπολογιστικών συστημάτων, όπως θα δούμε στη συνέχεια. Τα θεωρήματα των δικτύων αναμονής επεκτείνουν τις βασικές έννοιες της θεωρίας των αλυσίδων Markov, καθώς και τα βασικά αποτελέσματα της ανάλυσης απλών μαρκοβιανών συστημάτων αναμονής (διαδικασίες γεννήσεων-θανάτων) (Κεφ.3).

4.1 Στοιχεία Θεωρίας

Τα δίκτυα αναμονής εισήχθηκαν και μελετήθηκαν για πρώτη φορά από τον J.R. Jackson [13, 14]. Στη συνέχεια, η εξέλιξη των υπολογιστικών συστημάτων οδήγησε στην ανάπτυξη μεγάλου όγκου σχετικών θεωρητικών αποτελεσμάτων.

4.1.1 Δίκτυα Jackson

Ας θεωρήσουμε ένα δίκτυο αναμονής που αποτελείται από M σταθμούς εξυπηρέτησης, καθένας από τους οποίους έχει τη δική του ουρά αναμονής. Μπορούμε να παραστήσουμε το δίκτυο σαν ένα προσανατολισμένο γράφο $Q = (V, E)$, όπου $V = \{1, 2, \ldots, M\}$ το σύνολο των κόμβων, οι οποίοι αντιπροσωπεύουν τους σταθμούς του δικτύου και $E = V \times V$ το σύνολο των ακμών, οι οποίες αντιπροσωπεύουν τις συνδέσεις μεταξύ των σταθμών.

Διακρίνουμε δύο κατηγορίες δικτύων αναμονής:

- Σε ένα *ανοικτό δίκτυο*, θεωρούμε ότι οι πελάτες προέρχονται από τον «έξω κόσμο» και τελικά επιστρέφουν σ' αυτόν μετά από κάποιο διάστημα παραμονής στο δίκτυο. Συμβολίζουμε την επίδραση του έξω κόσμου με δύο ειδικούς κόμβους: τον κόμβο 0 (πηγή), από τον οποίο έρχονται πελάτες και τον κόμβο $M + 1$ (προορισμό), ο οποίος απορροφά τους πελάτες που εγκαταλείπουν το δίκτυο. (Προφανώς, θα μπορούσαμε να συμπτύξουμε τους κόμβους 0 και $M + 1$ σε έναν ενιαίο κόμβο που θα παρίστανε τον έξω κόσμο.)

- Σε ένα *κλειστό δίκτυο*, θεωρούμε ότι δεν υπάρχουν εξωτερικές αφίξεις ή αναχωρήσεις από το δίκτυο. Άρα λείπουν οι κόμβοι 0 και $M + 1$ και ο συνολικός αριθμός των πελατών που κυκλοφορούν στο δίκτυο είναι σταθερός.

Το γενικό μοντέλο δικτύων αναμονής που αναπτύχθηκε από τον Jackson στηρίζεται στις ακόλουθες υποθέσεις:

(i) Οι πελάτες φθάνουν από τον κόμβο 0 σύμφωνα με μία διαδικασία γεννήσεων ρυθμού $\lambda(N)$, όπου N ο συνολικός αριθμός πελατών στο δίκτυο τη στιγμή της άφιξης. Κάθε πελάτης που προέρχεται από τον κόμβο 0 κατευθύνεται με πιθανότητα q_{0i}, $i = 1, 2, \ldots, M$, σε καθέναν από τους σταθμούς i. Επομένως, με βάση την ιδιότητα της διάσπασης διαδικασιών Poisson, μπορούμε να πούμε ότι οι εξωτερικές αφίξεις στους σταθμούς $i = 1, 2, \ldots, M$ ακολουθούν ανεξάρτητες διαδικασίες γεννήσεων με αντίστοιχους ρυθμούς $\lambda(N)q_{0i}$. (Εννοείται ότι η υπόθεση αυτή αφορά μόνο τα ανοικτά δίκτυα αναμονής.)

(ii) Ο χρόνος εξυπηρέτησης ενός πελάτη στον σταθμό i ($i = 1, 2, \ldots, M$) ακολουθεί εκθετική κατανομή με παράμετρο $\mu_i(n_i)$, όπου n_i ο αριθμός πελατών στον κόμβο i (σε αναμονή ή εξυπηρέτηση). Η εξάρτηση αυτή του χρόνου εξυπηρέτησης από τον αριθμό πελατών στον σταθμό καλύπτει την ειδική περίπτωση κατά την οποία ο σταθμός i περιλαμβάνει c_i μονάδες εξυπηρέτησης, οπότε ο συνολικός ρυθμός εξυπηρέτησης του σταθμού δεν είναι πάντοτε σταθερός.

(iii) Οι πελάτες εξυπηρετούνται με κανονισμό FCFS (αν και ο περιορισμός αυτός μπορεί να αρθεί εύκολα).

(iv) Φεύγοντας από τον κόμβο i ($i = 1, 2, \ldots, M$) ένας πελάτης μπορεί να κατευθυνθεί σε οποιονδήποτε κόμβο j ($j = 1, 2, \ldots, M+1$) με πιθανότητα q_{ij}, ($q_{ij} \geq 0$ και $\sum_{j=1}^{M+1} q_{ij} = 1$). Οι πιθανότητες q_{ij}, $0 \leq i \leq M$, $1 \leq j \leq M + 1$ ονομάζονται *πιθανότητες δρομολόγησης* (routing probabilities). Θεωρούμε ότι $q_{0,M+1} = 0$, δηλαδή οι πέλατες επισκέπτονται οπωσδήποτε κάποιο σταθμό, πριν εγκαταλείψουν το δίκτυο. (Προφανώς, για κλειστό δίκτυο ορίζονται μόνο οι πιθανότητες q_{ij}, $1 \leq i, j \leq M$.) Γενικά, η διαδρομή που ακολουθεί ένας πελάτης στο δίκτυο περιγράφεται από μία αλυσίδα Markov με χώρο καταστάσεων $\{0, 1, \ldots, M + 1\}$ και πιθανότητες μετάβασης q_{ij}. Υποθέτουμε ότι η αλυσίδα Markov έχει μόνο μία απορροφητική κατάσταση, την κατάσταση $M + 1$.

Η κατάσταση του δικτύου μπορεί να παρασταθεί από το διάνυσμα $\boldsymbol{n} = [n_1, n_2, \ldots, n_M]$ όπου n_i, $i = 1, 2, \ldots, M$, ο αριθμός πελατών στον σταθμό i. Ο συνολικός αριθμός πελατών στο δίκτυο θα είναι $N = \|\boldsymbol{n}\| = \sum_{i=1}^{M} n_i$. Επομένως, σε ένα κλειστό δίκτυο αναμονής θα έχουμε N σταθερό.

Ορίζουμε τώρα την πιθανότητα $p(\boldsymbol{n}; t)$ να βρίσκεται το σύστημα στην κατάσταση $\boldsymbol{n} = [n_1, n_2, \ldots, n_M]$ τη χρονική στιγμή t. Θα αναζητήσουμε τη στατική κατανομή πιθανότητας $p(\boldsymbol{n}) = \lim_{t\to\infty} p(\boldsymbol{n}; t)$ για ανοικτά και κλειστά δίκτυα αναμονής.

4.1.1.1 Ανοικτά Δίκτυα Jackson

Για ανοικτό δίκτυο αναμονής σε κατάσταση ισορροπίας οι πιθανότητες $p(\boldsymbol{n})$ θα πρέπει να ικανοποιούν το πιο κάτω σύστημα εξισώσεων:

$$\begin{aligned}\left[\lambda(N) + \sum_{i=1}^{M} \mu_i(n_i)(1 - q_{ii})\right] p(\boldsymbol{n}) &= \lambda(N-1) \sum_{i=1}^{M} q_{0i} p(\boldsymbol{n} - \mathbf{1}_i) + \\ &\quad \sum_{i=1}^{M} \mu_i(n_i + 1) q_{i,M+1} p(\boldsymbol{n} + \mathbf{1}_i) + \\ &\quad \sum_{j=1}^{M} \sum_{\substack{i=1 \\ i\neq j}}^{M} \mu_i(n_i + 1) q_{ij} p(\boldsymbol{n} + \mathbf{1}_i - \mathbf{1}_j) \qquad (4.1)\end{aligned}$$

$$\lambda(0) p(\mathbf{0}) = \sum_{i=1}^{M} \mu_i(1) q_{i,M+1} p(\mathbf{1}_i) \qquad (4.2)$$

όπου $N = \sum_{i=1}^{M} n_i$, $\mathbf{0}$ είναι το διάνυσμα με M μηδενικές συνιστώσες και $\mathbf{1}_i$ είναι το διάνυσμα διάστασης M του οποίου η i-στη συνιστώσα έχει τιμή 1 και όλες οι άλλες συνιστώσες έχουν τιμή 0. Η Εξίσωση (4.1) ισχύει για όλες τις δυνατές τιμές του διανύσματος $\boldsymbol{n}$ για τις οποίες $n_i \geq 0$, $i = 1, 2, \ldots, M$, και εκφράζει την ισορροπία της ροής, σε αναλογία με τα μονοδιάστατα συστήματα αναμονής. Πράγματι στο αριστερό μέλος της (4.1) έχουμε τον ρυθμό εξόδου από την κατάσταση $\boldsymbol{n}$, ενώ στο δεξιό μέλος έχουμε τους ρυθμούς εισόδου στην κατάσταση $\boldsymbol{n}$ λόγω εξωτερικών αφίξεων, αναχωρήσεων από το δίκτυο και μετακινήσεων από κόμβο σε κόμβο αντίστοιχα.

Θεωρούμε τώρα τις ποσότητες $\{e_i\}$, $i = 1, 2, \ldots, M$, οι οποίες αποτελούν τη λύση του συστήματος εξισώσεων:

$$e_i = q_{0i} + \sum_{j=1}^{M} e_j q_{ji}, \quad i = 1, 2, \ldots, M \qquad (4.3)$$

Εύκολα μπορεί κανείς να αντιληφθεί τη φυσική ερμηνεία των ποσοστήτων αυτών: e_i είναι ο μέσος αριθμός επισκέψεων στο σταθμο i που πραγματοποιεί ένας πελάτης κατά τη διάρκεια της παραμονής του στο δίκτυο.

Για το μοντέλο που περιγράψαμε ισχύει το ακόλουθο θεώρημα (Jackson, 1963).

Θεώρημα 4.1. *Εάν το σύστημα Εξισώσεων* (4.3) *έχει μία μοναδική μη αρνητική λύση και* $G < \infty$, *τότε η στατική κατανομή* $p(\boldsymbol{n})$ *υπάρχει και δίνεται από τη σχέση:*

$$p(\boldsymbol{n}) = \frac{1}{G} \Lambda(N) \prod_{i=1}^{M} \frac{e_i^{n_i}}{M_i(n_i)} \qquad (4.4)$$

όπου

$$\Lambda(N) = \prod_{n=1}^{N} \lambda(n-1) \tag{4.5}$$

$$M_i(n_i) = \prod_{n=1}^{n_i} \mu_i(n) \tag{4.6}$$

$$G = \sum_{\boldsymbol{n}} \Lambda(N) \prod_{i=1}^{M} \frac{e_i^{n_i}}{M_i(n_i)} \tag{4.7}$$

Η άθροιση στην (4.7) γίνεται για όλες τις δυνατές τιμές του $\boldsymbol{n}$, και εξασφαλίζει τη συνθήκη ότι το άθροισμα των πιθανοτήτων $p(\boldsymbol{n})$ είναι ίσο με 1.

Η απόδειξη του θεωρηματος μπορεί να γίνει με αντικατάσταση στις εξισώσεις ισορροπίας (4.1) και (4.2), οι οποίες ονομάζονται εξισώσεις *καθολικής ισορροπίας* (global balance). Οι εξισώσεις αυτές εκφράζουν την ισορροπία του ρυθμού αναχώρησης της μαρκοβιανής διαδικασίας από τη συγκεκριμένη κατάσταση με τον ρυθμό εισόδου στην κατάσταση αυτή.

4.1.1.2 Η Μορφή Γινομένου

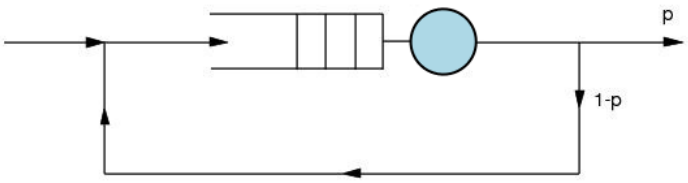
Δεν είναι δύσκολο να επιβεβαιώσει κανείς ότι οι εξισώσεις καθολικής ισορροπίας (4.1),(4.2) μπορούν να διασπαστούν σε μικρότερα συστήματα εξισώσεων τα οποία μπορούν να λυθούν πιο εύκολα [9, 10, 20]. Οι τροποποιημένες εξισώσεις εκφράζουν την ισορροπία της ροής σε σχέση με κάθε κόμβο του δικτύου χωριστά: ο ρυθμός εξόδου από την κατάσταση $\boldsymbol{n}$ λόγω μιας αναχώρησης από τον κόμβο i είναι ίσος με τον ρυθμό εισόδου στην κατάσταση $\boldsymbol{n}$ λόγω μιας άφιξης στον κόμβο i. Οι εξισώσεις αυτές ονομάζονται εξισώσεις *τοπικής ισορροπίας* (local balance) και συνδέονται άμεσα με τη *μορφή γινομένου* (product form) της γενικής λύσης [22]. Η έννοια της τοπικής ισορροπίας είναι χαρακτηριστική των εργοδικών διαδικασιών Markov, και συναντάται στη γενική λύση των διαδικασιών γεννήσεων-θανάτων με τη μορφή απλοποιημένων εξισώσεων. Στην ουσία οι Εξισώσεις (4.1) και (4.2) περιγράφουν μία πολυδιάστατη διαδικασία γεννήσεων-θανάτων. Γενικά, κάθε λύση που ικανοποιεί τις εξισώσεις τοπικής ισορροπίας θα ικανοποιεί και τις εξισώσεις καθολικής ισορροπίας, χωρίς να ισχύει πάντα το αντίστροφο.

Ο προσδιορισμός των εξισώσεων τοπικής ισορροπίας είναι περισσότερο εμπειρική προσέγγιση και λιγότερο αυστηρή αλγοριθμική διαδικασία. Μια ευρέως χρησιμοποιούμενη ιδέα στηρίζεται στη διάσπαση του αριστερού μέλους και του δεξιού μέλους της γενικής εξίσωσης καθολικής ισορροπίας σε $M+1$ αντίστοιχους όρους: ένα ζεύγος όρων που αφορούν εξωτερική άφιξη και εξωτερική αναχώρηση, και M ζεύγη όρων που αφορούν άφιξη ή αναχώρηση σε καθέναν από τους M κόμβους του δικτύου.Αν μπορεί να βρεθεί μια λύση που ικανοποιεί την ισότητα κάθε ζεύγους αντίστοιχων όρων (τοπική ισορροπία), αυτή θα αποτελεί και λύση της συνολικής εξίσωσης. Εφόσον οι εξισώσεις τοπικής ισορροπίας είναι πολύ απλούστερες, είναι ευκολότερος ο προσδιορισμός και ο έλεγχος υποψήφιων λύσεων.

Η τοπική ισορροπία σχετίζεται άμεσα με δύο άλλες χαρακτηριστικές ιδιότητες των δικτύων αναμονής μορφής γινομένου [18, 6].

Η πρώτη ιδιότητα συμβολίζεται ως $M \Rightarrow M$ (Markov implies Markov) και αναφέρεται στους σταθμούς αναμονής. Ένας σταθμός έχει την ιδιότητα $M \Rightarrow M$, εάν και μόνο εάν ο σταθμός μετατρέπει μια διαδικασία αφίξεων Poisson σε διαδικασία αναχωρήσεων Poisson. Αποδεικνύεται ότι, αν όλοι οι κόμβοι ενός δικτύου έχουν την ιδιότητα $M \Rightarrow M$, τότε το δίκτυο έχει λύση μορφής γινομένου. Η ιδιότητα ισχύει για απλά συστήματα αναμονής τύπου $M/M/1$, $M/M/k$ και $M/M/\infty$. Για περισσότερες πληροφορίες ο αναγνώστης παραπέμπεται στη θεωρία της *χρονικής αναστρεψιμότητας* (time-reversibility) και το θεώρημα του P.J. Burke [3, 9].

Η δεύτερη ιδιότητα ονομάζεται *ισορροπία σταθμού* (station balance) και αναφέρεται στους κανονισμούς εξυπηρέτησης. Ένας κανονισμός έχει την ιδιότητα της ισορροπίας σταθμού, αν οι ρυθμοί εξυπηρέτησης με τους οποίους εξυπηρετούνται οι πελάτες σε μια θέση της ουράς είναι ανάλογοι προς την πιθανότητα να εισέλθει ένας πελάτης σε αυτή τη θέση. Με άλλα λόγια, η ουρά ενός κόμβου διαμερίζεται σε θέσεις και ο ρυθμός με τον οποίο ένας πελάτης εισέρχεται σε μια θέση είναι ίσος με τον ρυθμό με τον οποίο αφήνει



Σχήμα 4.1: *Σύστημα $M/M/1$ με ανάδραση.*

αυτή τη θέση. Αποδεικνύεται ότι, όπως και η πρώτη ιδιότητα, η ισορροπία σταθμού αποτελεί ικανή (αλλά όχι αναγκαία) συνθήκη για την ύπαρξη λύσης σε μορφή γινομένου.

Η κατανομή πιθανότητας των δικτύων Jackson εκφράζεται ως γινόμενο όρων (ενός όρου για κάθε σταθμό) συνοδευόμενων από μια σταθερά κανονικοποίησης. Η ιδιότητα αυτή είναι πολύ σημαντική, διότι επιτρέπει τον ευχερή υπολογισμό της λύσης, και χαρακτηρίζει (ενδεχομένως σε παραλλαγές ως προς τη μορφή) και τα θεωρήματα που θα εξετάσουμε στη συνέχεια. Η μορφή γινομένου της λύσης συνεπάγεται ότι οι καταστάσεις n_i των επί μέρους κόμβων ($i = 1, 2, \ldots, M$) στην κατάσταση ισορροπίας συμπεριφέρονται σαν ανεξάρτητες τυχαίες μεταβλητές. Η αποσύζευξη αυτή των σταθμών είναι περισσότερο εμφανής, αν θεωρήσουμε την ειδική περίπτωση κατά την οποία ο εξωτερικός ρυθμός αφίξεων είναι ανεξάρτητος από τον πληθυσμό του δικτύου, δηλαδή $\lambda(N) = \lambda$ για κάθε N. Εχουμε:

$$\Lambda(N) = \lambda^N = \prod_{i=1}^{M} \lambda^{n_i} \tag{4.8}$$

και με αντικατάσταση στην (4.4) βρίσκουμε εύκολα:

$$p(\boldsymbol{n}) = \prod_{i=1}^{M} p_i(n_i) \tag{4.9}$$

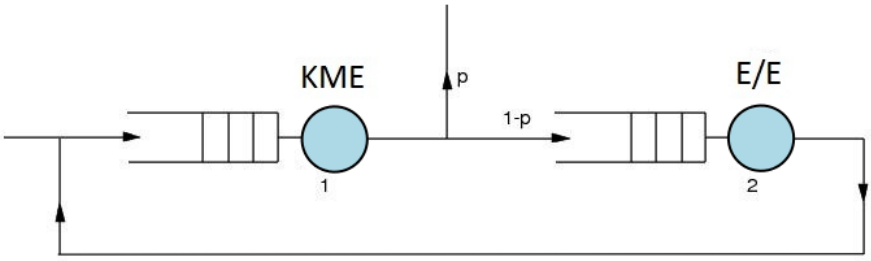
όπου:

$$p_i(n_i) = \frac{(\lambda e_i)^{n_i}}{M_i(n_i)} p_i(0) \tag{4.10}$$

$$p_i(0) = \left[\sum_{n=0}^{\infty} \frac{(\lambda e_i)^n}{M_i(n)} \right]^{-1} \tag{4.11}$$

Επομένως, κάθε σταθμός συμπεριφέρεται σαν ανεξάρτητο σύστημα αναμονής γεννήσεων-θανάτων με σταθερό ρυθμό αφίξεων λe_i και ρυθμό εξυπηρέτησης $\mu_i(n_i)$ εξαρτώμενο από την κατάσταση του σταθμού ($i = 1, 2, \ldots, M$). Η λύση απλοποιείται ακόμη περισσότερο, αν θεωρήσουμε σταθμούς με μία ή περισσότερες μονάδες εξυπηρέτησης, καθεμία από τις οποίες έχει σταθερό ρυθμό εξυπηρέτησης. Στην περίπτωση αυτή οι πιθανότητες $p_i(n_i)$ μπορούν να αντικατασταθούν κατευθείαν από τις κλασικές εκφράσεις για απλά συστήματα αναμονής $M/M/1$ (ή $M/M/c$) σε ισορροπία. Η περίπτωση αυτή ήταν η πρώτη που εξετάστηκε από τον Jackson το 1957, και αναφέρεται συχνά ως το πρώτο θεώρημα του Jackson. Θα πρέπει να σημειωθεί ότι –προς έκπληξη– η διαδικασία αφίξεων στους επιμέρους σταθμούς του δικτύου γενικά *δεν* είναι διαδικασία Poisson.

Παράδειγμα 4.1. *Ας θεωρήσουμε ένα απλό σύστημα αναμονής $M/M/1$ με ανάδραση, το οποίο μπορεί να παραστήσει ένα σταθμό μεταγωγής μηνυμάτων* (Σχ. 4.1)*. Ο χρόνος που απαιτείται για τη μετάδοση ενός μηνύματος και τη λήψη επιβεβαίωσης από τον προορισμό υποτίθεται εκθετικά κατανεμημένος με μέση τιμή $1/\mu$. Η πιθανότητα σωστής μετάδοσης ενός μηνύματος είναι p. Με πιθανότητα $1-p$ το μήνυμα πρέπει να ξαναμεταδοθεί. Τα μηνύματα φθάνουν στον σταθμό σύμφωνα με μία διαδικασία Poisson ρυθμού λ.*



Σχήμα 4.2: *Ανοικτό δίκτυο δύο σταθμών με ανάδραση.*

Πρόκειται για δίκτυο Jackson με ένα σταθμό ($M = 1$), για το οποίο ισχύει $q_{01} = 1$, $q_{11} = 1 - p$ και $q_{12} = p$, οπότε βάσει της (4.3) βρίσκουμε $e_1 = 1 + (1 - p)e_1$ ή $e_1 = 1/p$. Άρα η κατανομή $p(n)$ θα δίνεται από τη λύση του συστήματος $M/M/1$, αν θέσουμε $\rho = (\lambda e_1)/\mu = \lambda/p\mu$ (πρέπει $\rho < 1$) . Ο μέσος αριθμός μηνυμάτων στο σύστημα και ο μέσος χρόνος απόκρισης θα είναι αντίστοιχα $E[n] = \rho/(1-\rho) = \lambda/(p\mu-\lambda)$ και $T = 1/\mu(1 - \rho) = p/(p\mu - \lambda)$. Για παράδειγμα, αν θεωρήσουμε $\lambda = 4$ μηνύματα/sec, $1/\mu = 0,22$ sec και $p = 0,99$ βρίσκουμε: $\lambda e_1 = 4,0404$, $\rho = 0,889$, $E[n] = 8$ μηνύματα και $T = 1,98$ sec. Για το σύστημα $M/M/1$ χωρίς ανάδραση θα βρίσκαμε αντίστοιχα $E[n] = 7,33$ μηνύματα και $T = 1,83$ sec. □

Παράδειγμα 4.2. *Το δίκτυο του* Σχ. 4.2 *με δύο σταθμούς χρησιμοποιείται ως μοντέλο υπολογιστικού συστήματος με πολυπρογραμματισμό. Η εκτέλεση ενός προγράμματος στην ΚΜΕ διακόπτεται από δραστηριότητες εισόδου/εξόδου (π.χ. αιτήσεις σελίδων στην περίπτωση εικονικής μνήμης) και επομένως περιλαμβάνει πολλές κυκλικές μετακινήσεις ανάμεσα στην ΚΜΕ και τη μονάδα Ε/Ε. Υποθέτουμε αφίξεις Poisson με ρυθμό λ και εκθετικά κατανεμημένους χρόνους εξυπηρέτησης στην ΚΜΕ και τη μονάδα Ε/Ε με μέσες τιμές $1/\mu_1$ και $1/\mu_2$ αντίστοιχα. Φεύγοντας από την ΚΜΕ ένα πρόγραμμα εγκαταλείπει το σύστημα με πιθανότητα p ή περνάει στη μονάδα Ε/Ε με πιθανότητα $1 - p$. Μετά την εξυπηρέτηση στη μονάδα Ε/Ε το πρόγραμμα επιστρέφει στην ΚΜΕ.*

Έχουμε δίκτυο Jackson με $M = 2$ και $q_{01} = 1$, $q_{02} = 0$, $q_{11} = 0$, $q_{12} = 1-p$, $q_{13} = p$, $q_{21} = 1$, $q_{22} = 0$, $q_{23} = 0$. Από την (4.3) βρίσκουμε $e_1 = 1 + 0e_1 + 1e_2$, $e_2 = 0 + (1 - p)e_1 + 0e_2$ ή $e_1 = 1/p$, $e_2 = (1 - p)/p$.

Βάσει της (4.9) η πιθανότητα $p(n_1, n_2)$ θα δίνεται από τη σχέση $p(n_1, n_2) = p_1(n_1)p_2(n_2)$, όπου $p_1(n_1)$, $p_2(n_2)$ αντιστοιχούν σε απλά συστήματα $M/M/1$ με $\rho_1 = (\lambda e_1)/\mu_1 = \lambda/p\mu_1$ και $\rho_2 = (\lambda e_2)/\mu_2 = (1 - p)\lambda/p\mu_2$. Άρα $p(n_1, n_2) = (1 - \rho_1)\rho_1^{n_1}(1 - \rho_2)\rho_2^{n_2}$ (όπου $\rho_1 < 1$ και $\rho_2 < 1$. Ο μέσος αριθμός προγραμμάτων στο σύστημα θα είναι $E[n] = E[n_1] + E[n_2] = \rho_1/(1 - \rho_1) + \rho_2/(1 - \rho_2)$ και ο μέσος χρόνος απόκρισης $T = E[n]/\lambda$, σύμφωνα με τον τύπο του Little. Για παράδειγμα, ας υποθέσουμε ότι κάθε πρόγραμμα απαιτεί κατά μέσο όρο 4 sec συνολικού χρόνου στην ΚΜΕ με διακοπές Ε/Ε κάθε 0,25sec παραμονής στην ΚΜΕ, και κάθε δραστηριότητα Ε/Ε έχει μέση διάρκεια 0,2 sec. Το σύστημα επεξεργάζεται κατά μέσο όρο 8000 προγράμματα σε 10 ώρες ημερήσιας λειτουργίας. Εχουμε $\lambda = 8000/36000 = 2/9$ προγράμματα/sec. Ο μέσος αριθμός διακοπών Ε/Ε για ένα πρόγραμμα είναι 4/0,25=16, άρα $p = 1/16$. Επίσης $1/\mu_1 = 0,25$ sec και $1/\mu_2 = 0,2$ sec. Επομένως:

- Για την ΚΜΕ και τη μονάδα Ε/Ε θα έχουμε αντίστοιχα $\rho_1 = 8/9$ και $\rho_2 = 2/3$.
- Ο μέσος αριθμός προγραμμάτων στην ΚΜΕ και στη μονάδα Ε/Ε είναι $E[n_1] = 8$ και $E[n_2] = 2$, άρα συνολικά $E[n] = 10$ προγράμματα.
- Ο συνολικός μέσος χρόνος απόκρισης του συστήματος είναι $T = 45$ sec.

□

4.1.1.3 Κλειστά Δίκτυα Jackson

Για κλειστό δίκτυο αναμονής οι εξισώσεις καθολικής ισορροπίας παίρνουν τη μορφή:

$$\sum_{i=1}^{M} \mu_i(n_i)(1-q_{ii})p(\boldsymbol{n}) = \sum_{j=1}^{M} \sum_{\substack{i=1 \\ i\neq j}}^{M} \mu_i(n_i+1)q_{ij}p(\boldsymbol{n}+\mathbf{1}_i-\mathbf{1}_j) \tag{4.12}$$

εφόσον δεν υπάρχουν εξωτερικές αφίξεις ή αναχωρήσεις από το δίκτυο.

Θεωρούμε και πάλι τις ποσότητες $\{e_i\}$, $i = 1, 2, \ldots, M$ ως λύση του συστήματος:

$$e_i = \sum_{j=1}^{M} e_j q_{ji}, \quad i = 1, 2, \ldots, M \tag{4.13}$$

Παρατηρούμε, όμως, ότι στην περίπτωση αυτή η λύση του συστήματος δεν είναι μοναδική, καθόσον η μήτρα $\boldsymbol{Q} = [q_{ij}]$ είναι στοχαστική μήτρα. Αν εισαγάγουμε τον πρόσθετο περιορισμό $\sum_{i=1}^{M} e_i = 1$, τότε μπορούμε να δώσουμε διαφορετική ερμηνεία στις ποσότητες $\{e_i\}$ για τα κλειστά δίκτυα: πρόκειται για τις στατικές πιθανότητες της αλυσίδας Markov που περιγράφει τη μετακίνηση ενός πελάτη στο δίκτυο (αν θέσουμε $\boldsymbol{e} = [e_i]$ έχουμε σύστημα της γνωστής μορφής $\boldsymbol{e} = \boldsymbol{eQ}$). Υποθέτουμε ότι η αλυσίδα αυτή είναι αμείωτη, οπότε θα είναι εργοδική καθόσον έχει πεπερασμένο χώρο καταστάσεων. Πάντως, όπως θα δούμε στη συνέχεια, ο περιορισμός αυτός δεν είναι απαραίτητος, γιατί οι ποσότητες $\{e_i\}$ εμφανίζονται στη γενική λύση μόνο σε σχέση πηλίκου.

Το πιο κάτω θεώρημα διατυπώθηκε από τους Gordon και Newell (1967) [11], αλλά προκύπτει άμεσα ως ειδική περίπτωση του γενικού θεωρήματος Jackson.

Θεώρημα 4.2. *Έστω $\{e_i\}$, $i = 1, 2, \ldots, M$ μία μη μηδενική λύση του συστήματος* (4.13). *Τότε η στατική κατανομή $p(\boldsymbol{n})$ με $\|\boldsymbol{n}\| = \sum_{i=1}^{M} n_i = N$, $n_i \geq 0$, υπάρχει και δίνεται από την σχέση:*

$$p(\boldsymbol{n}) = \frac{1}{G(N,M)} \prod_{i=1}^{M} \frac{e_i^{n_i}}{M_i(n_i)} \tag{4.14}$$

όπου οι ποσότητες $M_i(n_i)$ ορίζονται όπως στην (4.6) *και*

$$G(N,M) = \sum_{\|\boldsymbol{n}\|=N} \prod_{i=1}^{M} \frac{e_i^{n_i}}{M_i(n_i)} \tag{4.15}$$

Η ποσότητα $G(N, M)$ αναφέρεται ως *σταθερά κανονικοποίησης* (normalization constant).

Η απόδειξη του θεωρήματος μπορεί να γίνει με απευθείας αντικατάσταση στις Εξισώσεις (4.12). Παρατηρούμε και πάλι την ύπαρξη της μορφής γινομένου, ως συνέπεια της τοπικής ισορροπίας. Στην περίπτωση αυτή όμως δεν πρόκειται για γινόμενο ανεξάρτητων κατανομών για τους κόμβους, όπως στα ανοικτά δίκτυα. Πράγματι, οι κόμβοι δεν συμπεριφέρονται ανεξάρτητα εφόσον ο συνολικός αριθμός πελατών στο δίκτυο είναι σταθερός.

4.1.1.4 Ο Αλγόριθμος της Συνέλιξης

Η βασική δυσκολία για τον καθορισμό της κατανομής (4.14) είναι η άθροιση που απαιτείται για τον υπολογισμό της σταθεράς $G(N, M)$. Μπορεί κανείς εύκολα να διαπιστώσει ότι ο αριθμός των δυνατών καταστάσεων $\boldsymbol{n}$ (άρα ο αριθμός των όρων στην άθροιση) είναι ίσος με το πλήθος των διαφορετικών τρόπων με τους οποίους μπορούν να τοποθετηθούν N πελάτες στους M κόμβους. Ο αριθμός αυτός είναι ίσος με $\begin{pmatrix} N+M-1 \\ M-1 \end{pmatrix}$ και είναι τεράστιος ακόμη και για σχετικά μικρές τιμές των M και N. Ευτυχώς, έχουν αναπτυχθεί αποτελεσματικοί αλγόριθμοι για τον υπολογισμό της σταθεράς κανονικοποίησης. Ένας από

αυτούς είναι ο *αλγόριθμος της συνέλιξης* (convolution algorithm) που αναπτύχθηκε από τον J.P. Buzen [4].

Θεωρούμε τη γεννήτρια συνάρτηση της ακολουθίας $\{\frac{e_i^n}{M_i(n)},\ n = 1, 2, \ldots\}$, η οποία ορίζεται:

$$g_i(z) = \sum_{n=0}^{\infty} \frac{(e_i z)^n}{M_i(n)}, \quad i = 1, 2, \ldots, M$$

και το γινόμενο των γεννητριών συναρτήσεων:

$$g(z) = \prod_{i=1}^{M} g_i(z)$$

Από την (4.15) είναι εύκολο να διαπιστώσουμε ότι η σταθερά $G(N, M)$ είναι ο συντελεστής του όρου z^N στην $g(z)$. Για τον υπολογισμό της σταθεράς, ορίζουμε καταρχάς τα μερικά γινόμενα:

$$\begin{aligned} \gamma_1(z) &= g_1(z) \\ \gamma_i(z) &= \gamma_{i-1}(z) g_i(z), \quad i = 2, 3, \ldots, M \end{aligned}$$

και ονομάζουμε $G(j, i)$ τον συντελεστή του όρου z^j στο γινόμενο $\gamma_i(z)$, ο οποίος θα είναι ίσος με:

$$G(j, i) = \sum_{k=0}^{j} G(k, i-1) \frac{e_i^{j-k}}{M_i(j-k)} \tag{4.16}$$

Η αναδρομική σχέση (4.16) προσφέρεται για τον υπολογισμό της σταθεράς $G(N, M)$ με χρήση των οριακών συνθηκών:

$$G(0, i) = 1, \quad i = 1, 2, \ldots, M \tag{4.17}$$

$$G(j, 1) = \frac{e_1^j}{M_1(j)}, \quad j = 1, 2, \ldots, N \tag{4.18}$$

Ο υπολογισμός αυτός απαιτεί $O(MN^2)$ αριθμητικές πράξεις.

Τα πράγματα είναι πολύ πιο εύκολα στην ειδική περίπτωση που οι ρυθμοί εξυπηρέτησης $\mu_i(n_i)$ είναι ανεξάρτητοι από το μήκος της ουράς, δηλαδή $\mu_i(n_i) = \mu_i$ για κάθε $n_i > 0$. Πράγματι, αν θέσουμε $\tau_i = e_i/\mu_i$, $i = 1, 2, \ldots, M$, η (4.16) γράφεται:

$$\begin{aligned} G(j, i) &= \sum_{k=0}^{j} G(k, i-1) \tau_i^{j-k} \\ &= G(j, i-1) + \sum_{k=0}^{j-1} G(k, i-1) \tau_i^{j-k} \\ &= G(j, i-1) + \tau_i \sum_{k=0}^{j-1} G(k, i-1) \tau_i^{j-k-1} \end{aligned}$$

ή

$$G(j, i) = G(j, i-1) + \tau_i G(j-1, i) \tag{4.19}$$

Η τελευταία σχέση μαζί με τις (4.17)– (4.18) που γίνεται $G(j, 1) = \tau_1^j$, $j = 1, 2, \ldots, N$, επιτρέπουν τον υπολογισμό της σταθεράς $G(N, M)$ με $O(MN)$ αριθμητικές πράξεις. Ιδιαίτερα εύχρηστη είναι η παράσταση του υπολογισμού σε μορφή δομής με $N + 1$ γραμμές και M στήλες, όπως ο Πίνακας 4.1. Δεδομένου ότι ενδιαφερόμαστε μόνο για τα στοιχεία της τελευταίας στήλης, δεν είναι απαραίτητη η αποθήκευση ολόκληρου του πίνακα αλλά μόνο μιας στήλης, η οποία ενημερώνεται κατά την εκτέλεση του αλγορίθμου.

	1	2	⋯	i	⋯	M
0	1	1	⋯	1	⋯	1
1	τ_1					
2	τ_1^2					
⋮	⋮					
				$G(j-1,i)$		
				$\downarrow \times \tau_i$		
j	τ_1^j		$G(j,i-1) \longrightarrow$	$G(j,i)$		
⋮	⋮					
N	τ_1^N					$G(N,M)$

Πίνακας 4.1: *Υλοποίηση του αλγορίθμου της συνέλιξης.*

Ο αλγόριθμος βασίζεται στην ιδιότητα της συνέλιξης ότι, αν δύο πεπερασμένες ακολουθίες αποτελούν τους συντελεστές δύο πολυωνύμων, τότε οι συντελεστές του συνήθους γινομένου των δύο πολυωνύμων είναι η συνέλιξη των αρχικών ακολουθιών (γνωστό και ως *γινόμενο Cauchy*).

Χρησιμοποιώντας τον αλγόριθμο της συνέλιξης στη γενική περίπτωση μπορούμε να υπολογίσουμε διάφορους δείκτες επίδοσης του δικτύου. Θα αναφέρουμε τα πιο κάτω αποτελέσματα, τα οποία ισχύουν για κάθε σταθμό i με σταθερό ρυθμό εξυπηρέτησης μ_i (χωρίς να έχουν υποχρεωτικά όλοι οι σταθμοί του δικτύου σταθερό ρυθμό εξυπηρέτησης):

- *Βαθμός χρησιμοποίησης του σταθμού*

$$U_i = \tau_i \frac{G(N-1,M)}{G(N,M)}$$

- *Ρυθμός απόδοσης του σταθμού*

$$X_i = e_i \frac{G(N-1,M)}{G(N,M)}$$

(Το αποτέλεσμα αυτό ισχύει και για μη σταθερό ρυθμό εξυπηρέτησης.)

- *Μέσος αριθμός πελατών στον σταθμό*

$$E[n_i] = \frac{1}{G(N,M)} \sum_{n=1}^{N} G(N-n,M)\tau_i^n$$

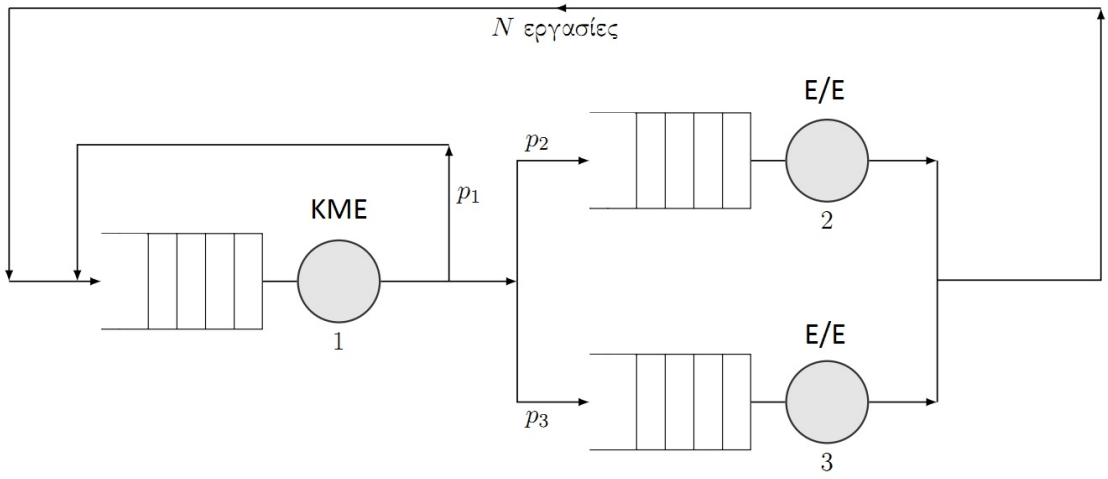
- *Μέσος χρόνος απόκρισης του σταθμού* (Τύπος του Little)

$$T_i = \frac{E[n_i]}{\lambda_i} = \frac{1}{e_i G(N-1,M)} \sum_{n=1}^{N} G(N-n,M)\tau_i^n$$

Παράδειγμα 4.3. *Έστω το κλειστό δίκτυο του* Σχ. 4.3 *το οποίο μπορεί να χρησιμοποιηθεί ως μοντέλο υπολογιστικού συστήματος με σταθερό βαθμό πολυπρογραμματισμού N (υποθέτουμε ότι N προγράμματα κυκλοφορούν συνεχώς ανάμεσα στην ΚΜΕ και τις δύο μονάδες Ε/Ε). Έχουμε $N = 4$, $M = 3$ και σταθερούς ρυθμούς εξυπηρέτησης $1/\mu_1 = 28$ msec, $1/\mu_2 = 40$ msec και $1/\mu_3 = 280$ msec. Επίσης υποθέτουμε ότι $p_1 = 0,1$, $p_2 = 0,7$ και $p_3 = 0,2$.*

Η μήτρα $\boldsymbol{Q}$ των πιθανοτήτων δρομολόγησης έχει τη μορφή:

$$\boldsymbol{Q} = \begin{bmatrix} p_1 & p_2 & p_3 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$



Σχήμα 4.3: *Επίλυση κλειστού δικτύου με τον αλγόριθμο της συνέλιξης.*

οπότε μία λύση του συστήματος $\boldsymbol{e} = \boldsymbol{eQ}$ θα είναι $e_1 = 1$, $e_2 = p_2$ και $e_3 = p_3$.

Η κατανομή πιθανότητας για το δίκτυο θα είναι:

$$p(n_1, n_2, n_3) = \frac{1}{G(4,3)} \left(\frac{1}{\mu_1}\right)^{n_1} \left(\frac{p_2}{\mu_2}\right)^{n_2} \left(\frac{p_3}{\mu_3}\right)^{n_3} = \frac{1}{G(4,3)} \tau_1^{n_1} \tau_2^{n_2} \tau_3^{n_3}$$

Η σταθερά $G(4,3)$ υπολογίζεται με βάση τον αλγόριθμο:

$$\begin{aligned} G(0,i) &= 1, \quad i = 1,2,3 \\ G(j,1) &= \tau_1^j, \quad j = 1,2,3,4 \\ G(j,i) &= G(j,i-1) + \tau_i G(j-1,i), \quad i = 2,3, \quad j = 1,2,3,4 \end{aligned}$$

Αντικαθιστώντας τις αριθμητικές τιμές έχουμε $\tau_1 = 28$, $\tau_2 = 28$, $\tau_3 = 56$ (msec), οπότε με εφαρμογή του αλγορίθμου βρίσκουμε τελικά $G(1,3) = 4 \times 28$, $G(2,3) = 11 \times 28^2$, $G(3,3) = 26 \times 28^3$, $G(4,3) = 57 \times 28^4$. Επομένως, ο βαθμός χρησιμοποίησης καθενός από τους 3 σταθμούς θα είναι $U_1 = 26/57$, $U_2 = 26/57$, $U_3 = 52/57$.

Ο ρυθμός απόδοσης της KME θα είναι $X_1 = 0,01629$ προγράμματα/msec=16,29 προγράμματα/sec. Η πιθανότητα p_1 εκφράζει το τέλος της εκτέλεσης ενός προγράμματος (και την αντικατάστασή του με ένα άλλο, πράγμα που συμβολίζεται με επιστροφή στην KME). Άρα ο ρυθμός απόδοσης του συστήματος θα είναι $X = p_1 \lambda_1 = 1,629$ προγράμματα/sec. Τέλος, ο μέσος χρόνος απόκρισης του συστήματος θα είναι $T = N/\lambda = 2,455$ sec. □

4.1.2 Δίκτυα BCMP

Η ανάπτυξη των μοντέλων Jackson έγινε κυρίως στις αρχές της δεκαετίας του '60, με βασικό κίνητρο την επίλυση προβλημάτων της επιχειρησιακής έρευνας. Τα μοντέλα αυτά χρησιμοποιήθηκαν στη συνέχεια και μέχρι τα μέσα της δεκαετίας του '70 ως μοναδικά εργαλεία για την ανάλυση υπολογιστικών συστημάτων χάρη στην απλότητα και την αποτελεσματικότητά τους. Η εξέλιξη, όμως, των υπολογιστικών συστημάτων δημιούργησε την ανάγκη για αναζήτηση νέων μοντέλων, ικανών να παραστήσουν ιδιαίτερα χαρακτηριστικά των υπολογιστικών συστημάτων που δεν καλύπτονται από τα δίκτυα Jackson. Αποδείχθηκε ότι άλλες κατηγορίες δικτύων, πιο γενικές από τα δίκτυα Jackson, επιδέχονται επίσης λύση σε μορφή γινομένου. Οι γενικεύσεις αυτές περιλαμβάνουν πολλές κατηγορίες πελατών, κανονισμούς εξυπηρέτησης διαφορετικούς από τον κανονισμό FCFS και γενικές κατανομές των χρόνων εξυπηρέτησης. Τα κυριότερα αποτελέσματα παρουσιάστηκαν το 1975 στο συλλογικό άρθρο των F.Baskett, K.M.Chandy, R.R.Muntz και F.G.Palacios [1]. Τα δίκτυα αυτά αναφέρονται συνήθως ως *γενικά δίκτυα με λύση μορφής γινομένου* ή *διαχωρίσιμα* (separable) *δίκτυα* ή *δίκτυα BCMP*.

4.1.2.1 Βασικές Ιδιότητες

Η τοπολογία ενός δικτύου BCMP μπορεί να περιγραφεί από ένα γράφο M κόμβων (καθώς και τους κόμβους 0 και $M+1$ αν πρόκειται για ανοικτό δίκτυο). Υπάρχουν C κατηγορίες πελατών και ένας πελάτης μπορεί να αλλάξει κατηγορία καθώς κινείται από κόμβο σε κόμβο. Ειδικότερα, ένας πελάτης της κατηγορίας r ο οποίος τελειώνει την εξυπηρέτησή του στον κόμβο i, πηγαίνει στον κόμβο j ως πελάτης της κατηγορίας s με πιθανότητα $q_{ir,js}$. Στην περίπτωση ανοικτού δικτύου, ένας πελάτης που προέρχεται από τον έξω κόσμο κατευθύνεται στον κόμβο i ως πελάτης της κατηγορίας r με πιθανότητα $q_{0,ir}$, ενώ ένας πελάτης της κατηγορίας r, ο οποίος φεύγει από τον κόμβο i, εγκαταλείπει οριστικά το δίκτυο με πιθανότητα $q_{ir,M+1}$. Τα δυνατά ζεύγη (i,r) μπορούν να κατανεμηθούν σε ένα ή περισσότερα υποσύνολα τα οποία ονομάζονται *αλυσίδες* του δικτύου: δύο ζεύγη ανήκουν στην ίδια αλυσίδα αν υπάρχει μη μηδενική πιθανότητα ένας πελάτης να βρεθεί στις δύο αντίστοιχες καταστάσεις κατά τη διάρκεια της παραμονής του στο δίκτυο. Για παράδειγμα, αν οι πελάτες δεν αλλάζουν ποτέ κατηγορία, τότε θα υπάρχουν τουλάχιστον C αλυσίδες. Οι αλυσίδες του δικτύου μπορεί να είναι κλειστές (να περιέχουν σταθερό αριθμό πελατών) ή ανοικτές (με εξωτερικές αφίξεις και αναχωρήσεις). Αν όλες οι αλυσίδες του δικτύου είναι κλειστές, τότε το δίκτυο είναι κλειστό.

Στην περίπτωση ανοικτού δικτύου, οι εξωτερικές αφίξεις γίνονται σύμφωνα με μία διαδικασία γεννήσεων ρυθμού $\lambda(N)$, όπου N ο συνολικός αριθμός πελατών στο δίκτυο. (Στο [1] ορίζεται και ένας δεύτερος μηχανισμός αφίξεων, με χωριστή διαδικασία για κάθε αλυσίδα του δικτύου.)

Απομένει να περιγράψουμε τους κόμβους του δικτύου, όσον αφορά την κατανομή των χρόνων εξυπηρέτησης και τον μηχανισμό εξυπηρέτησης.

Πέραν των εκθετικών κατανομών που χαρακτήριζαν το μοντέλο Jackson, το μοντέλο BCMP θεωρεί κατανομές χρόνων εξυπηρέτησης που ανήκουν στην ευρύτερη οικογένεια των κατανομών Cox. Οι κατανομές αυτές είναι γενικές και έχουν ιδιαίτερη χρησιμότητα γιατί επιτρέπουν την ανάλυση των συστημάτων αναμονής με βάση τη θεωρία των διαδικασιών Markov.

4.1.2.2 Κατανομές Cox

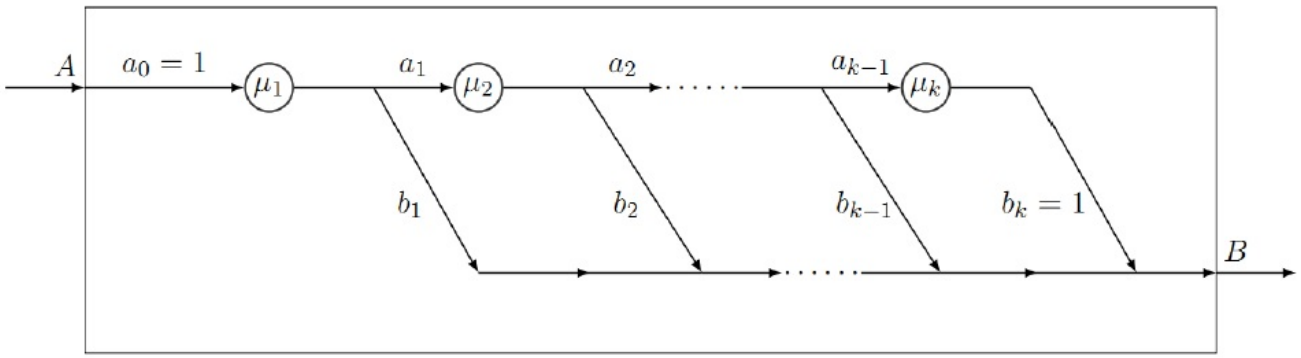
Οι κατανομές Cox εντάσσονται σε μια ευρύτερη τεχνική που ονομάζεται «μέθοδος των σταδίων» ή «μέθοδος των φάσεων». Η ιδέα είναι ότι κάθε κατανομή μπορεί να παρασταθεί με ακρίβεια μέσω ενός μίγματος εκθετικών κατανομών, που αναφέρεται ως *κατανομή φάσεων* (phase-type distribution – PH). Εφόσον οι κατανομές PH αποτελούνται από εκθετικές κατανομές, ένα σύστημα αναμονής του οποίου οι αφίξεις και εξυπηρετήσεις περιγράφονται από κατανομές PH μπορεί να αναλυθεί ως μαρκοβιανή διαδικασία συνεχούς χρόνου.

Βασικές κατανομές φάσεων είναι η *κατανομή Erlang-k*, που αποτελείται από k παρόμοια (με την ίδια παράμετρο) εκθετικά στάδια «εν σειρά» (ονομάζεται και *Υποεκθετική* στη γενικευμένη διατύπωσή της) και η *Υπερεκθετική κατανομή k-φάσεων*, που αποτελείται από k εκθετικά στάδια «εν παραλλήλω». Οι κατανομές φάσεων εκφράζουν τη μίξη εκθετικών κατανομών στην πιο γενική μορφή της.

Ορισμός [12]. Θεωρούμε αλυσίδα Markov συνεχούς χρόνου $k+1$ καταστάσεων, $k \geq 1$ με τις παρακάτω ιδιότητες:

- Οι καταστάσεις 1 έως k είναι μεταβατικές και η κατάσταση 0 είναι απορροφητική.
- Αρχική κατανομή πιθανότητας $\boldsymbol{a} = (a_0, a_1, \ldots, a_k)$, $\sum_{i=0}^{k} a_i = 1$.
- Μήτρα ρυθμών μετάβασης M, διαστάσεων $k \times (k+1)$, από τις καταστάσεις $\{1, 2, \ldots, k\}$ στις καταστάσεις $\{0, 1, 2, \ldots, k\}$.
- Δεν υπάρχει μετάβαση εξόδου από την κατάσταση 0 ούτε από μια κατάσταση στον εαυτό της.

Η κατανομή του χρόνου μέχρι την απορρόφηση είναι κατανομή PH k-φάσεων. Κάθε κατάσταση της διαδικασίας Markov αντιστοιχεί σε μια από τις φάσεις της κατανομής. Αν $a_0 \neq 0$, η κατανομή θα περιλαμβάνει μάζα πιθανότητας στο 0. □



Σχήμα 4.4: *Κατανομή Cox.*

Αποδεικνύεται ότι μια κατανομή PH με επαρκή αριθμό φάσεων μπορεί να προσεγγίσει οποιαδήποτε μη-αρνητική κατανομή με αυθαίρετα μεγάλη ακρίβεια. Οι κατανομές Cox αποτελούν υποσύνολο των κατανομών PH.

Μια κατανομή Cox μπορεί να παρασταθεί με τη βοήθεια (ανεξάρτητων) εκθετικών σταδίων εξυπηρέτησης, όπως στο Σχήμα 4.4. Υποθέτουμε ότι κάθε στιγμή μόνο ένας πελάτης βρίσκεται ανάμεσα στα σημεία A και B. Πριν από κάθε στάδιο $i+1$ ($i = 1, \ldots, k-1$), ο πελάτης επιλέγει αν θα συνεχίσει τη διάσχιση ή θα σταματήσει, σύμφωνα με τις αντίστοιχες πιθανότητες a_i και b_i, όπου $a_i + b_i = 1$. Όπως αναφέρθηκε σχετικά με τις κατανομές PH, κατ' αντιστοιχία, αν $b_0 \neq 0$, η κατανομή θα περιλαμβάνει μάζα πιθανότητας στο 0. Ο μετασχηματισμός Laplace της κατανομής δίνεται από τη σχέση

$$\Phi(s) = \sum_{i=1}^{k} A_i b_i \prod_{j=1}^{i} \frac{\mu_j}{\mu_j + s}$$

όπου $A_i = a_0 \cdots a_{i-1}$ ($i = 1, \ldots, k$) είναι η πιθανότητα να φθάσει ο πελάτης στο στάδιο i. Ο μέσος χρόνος εξυπηρέτησης για την κατανομή Cox θα δίνεται από τη σχέση $1/\mu = \sum_{i=1}^{k} A_i/\mu_i$.

Η $\Phi(s)$ είναι ρητή συνάρτηση της οποίας ο παρονομαστής έχει πραγματικές ρίζες και βαθμό μεγαλύτερο ή ίσο του βαθμού του αριθμητή. Αντίστροφα, κάθε κατανομή της οποίας ο μετασχηματισμός Laplace ικανοποιεί την παραπάνω συνθήκη μπορεί να παρασταθεί με εκθετικά στάδια. Κάθε γραμμικός συνδυασμός κατανομών Cox είναι επίσης κατανομή Cox. Επιπλέον, οποιαδήποτε κατανομή πιθανότητας μπορεί να προσεγγιστεί σε οποιοδήποτε βαθμό ακρίβειας από μια κατανομή Cox.

4.1.2.3 Οι Τέσσερις Τύποι Σταθμών

Όσον αφορά τους κανονισμούς εξυπηρέτησης, εκτός από τον κανονισμό FIFO ενδιαφερόμαστε για τους πιο κάτω κανονισμούς:

- PS (Processor Sharing). Αποτελεί οριακή περίπτωση του κανονισμού Round-Robin, ο οποίος παραχωρεί εξυπηρέτηση σε quanta σταθερής διάρκειας Q. Για $Q \to 0$ έχουμε έναν τρόπο λειτουργίας χωρίς αναμονή, όπου όλοι οι πελάτες εξυπηρετούνται παράλληλα και εξίσου με ρυθμό αντιστρόφως ανάλογο προς το πλήθος τους.

- IS (Infinite Servers). Κάθε στιγμή διατίθενται τόσες μονάδες εξυπηρέτησης, όσοι και οι πελάτες στον σταθμό. Επομένως δεν υπάρχει αναμονή γιατί, μόλις φθάσει ένας πελάτης, μία νέα μονάδα αναλαμβάνει την εξυπηρέτησή του. Ο κανονισμός αυτός αναφέρεται συχνά και ως Server-per-job ή Delay (εφόσον στην ουσία κάθε πελάτης καθυστερεί στον σταθμό ανεξάρτητα από τους άλλους).

- LCFSPR (Last Come-First Served-Preemptive-Resume). Ο σταθμός εξυπηρέτησης συμπεριφέρεται σαν στοίβα εξυπηρετώντας κάθε στιγμή τον πελάτη που έφθασε τελευταίος. Ένας πελάτης που

εξυπηρετείται, διακόπτεται από την άφιξη ενός νέου πελάτη (απόλυτη προτεραιότητα) και συνεχίζει την εξυπηρέτησή του αργότερα από το σημείο διακοπής.

Κάθε κόμβος ενός δικτύου BCMP μπορεί να ανήκει σε έναν από τους πιο κάτω τέσσερις τύπους:

- **Τύπος 1**. Ο σταθμός περιλαμβάνει μία μονάδα εξυπηρέτησης και ο χρόνος εξυπηρέτησης είναι εκθετικά κατανεμημένος με παράμετρο $\mu_i(n_i)$ για όλες τις κατηγορίες πελατών, όπου n_i ο αριθμός πελατών στον κόμβο. Ο κανονισμός εξυπηρέτησης είναι FCFS. (Η εξάρτηση της παραμέτρου μ_i από τον αριθμό πελατών επιτρέπει να περιλάβουμε και την περίπτωση πολλών μονάδων εξυπηρέτησης.)

- **Τύπος 2**. Οι χρόνοι εξυπηρέτησης ακολουθούν κατανομή Cox, η οποία μπορεί να είναι διαφορετική για κάθε κατηγορία πελατών. Υπάρχει μία μονάδα εξυπηρέτησης και ο κανονισμός εξυπηρέτησης είναι PS.

- **Τύπος 3**. Ο κανονισμός εξυπηρέτησης είναι IS και οι χρόνοι εξυπηρέτησης όπως ορίστηκαν για τον τύπο 2.

- **Τύπος 4**. Ο κανονισμός εξυπηρέτησης είναι LCFSPR και οι χρόνοι εξυπηρέτησης όπως για τους τύπους 2 και 3.

4.1.2.4 Επίλυση του Μοντέλου

Η κατάσταση $\boldsymbol{S}$ του δικτύου κάθε στιγμή ορίζεται ως το διάνυσμα $\boldsymbol{S} = [\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_M]$, όπου $\boldsymbol{S}_i$ η κατάσταση του σταθμού i, η οποία ορίζεται διαφορετικά για κάθε τύπο σταθμού και περιέχει πολύ περισσότερες πληροφορίες από όσες συνήθως χρειάζονται στην πράξη. Επειδή ενδιαφερόμαστε για μία περισσότερο συγκεντρωτική περιγραφή της κατάστασης του δικτύου, παραπέμπουμε στη βιβλιογραφία για τον λεπτομερή ορισμό των καταστάσεων $\boldsymbol{S}_i$ [1, 9, 10]. Αναφέρουμε απλά ότι οι κατανομές των χρόνων εξυπηρέτησης (εκθετικά στάδια) σε συνδυασμό με τους αντίστοιχους κανονισμούς εξυπηρέτησης εξασφαλίζουν ότι η $\boldsymbol{S}$ (ως συνάρτηση του χρόνου) είναι διαδικασία Markov. Ενδιαφερόμαστε για τη στατική κατανομή της διαδικασίας αυτής, η οποία θα πρέπει να ικανοποιεί τις εξισώσεις καθολικής ισορροπίας. Αποδεικνύεται ότι οι εξισώσεις καθολικής ισορροπίας μπορούν να διασπαστούν σε ένα είδος εξισώσεων τοπικής ισορροπίας, πράγμα το οποίο μας επιτρέπει να περιμένουμε λύση σε μορφή γινομένου.

Η ύπαρξη της στατικής κατανομής εξαρτάται από τη λύση του πιο κάτω συστήματος εξισώσεων:

$$e_{ir} = q_{0,ir} + \sum_{j=1}^{M}\sum_{s=1}^{C} e_{js}q_{js,ir}, \quad 1 \le i,j \le M, \;\; 1 \le r,s \le C \tag{4.20}$$

Είναι φανερό ότι οι Εξισώσεις (4.20) αποτελούν γενίκευση των Εξισώσεων (4.3) για την περίπτωση πολλών κατηγοριών πελατών. Η ποσότητα e_{ir} είναι ανάλογη με τη συχνότητα επισκέψεων των πελατών της κατηγορίας r στον κόμβο i. Σύμφωνα με τον ορισμό των αλυσίδων ενός δικτύου, στην πραγματικότητα το σύστημα (4.20) αποτελείται από ανεξάρτητα υποσυστήματα, ένα για κάθε αλυσίδα του δικτύου.

Το γενικό θεώρημα BCMP [1] καθορίζει τη λύση των εξισώσεων καθολικής ισορροπίας σύμφωνα με τη λεπτομερή περιγραφή των καταστάσεων που αναφέρθηκε πιο πάνω. Θα παρουσιάσουμε εδώ μία απλούστερη μορφή του θεωρήματος, η οποία και χρησιμοποιείται περισσότερο στις πρακτικές εφαρμογές.

Ορίζουμε την κατάσταση του δικτύου $\boldsymbol{n} = [\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \boldsymbol{n}_M]$ με τη βοήθεια των καταστάσεων των κόμβων $\boldsymbol{n}_i = [n_{i1}, n_{i2}, \ldots, n_{iC}]$, όπου n_{ir} $(r = 1, 2, \ldots, C)$ ο αριθμός των πελατών της κατηγορίας r στον κόμβο i $(i = 1, 2, \ldots, M)$. Συμβολίζουμε με $n_i = \|\boldsymbol{n}_i\| = \sum_{r=1}^{C} n_{ir}$ τον συνολικό αριθμό πελατών στον κόμβο i και με $N = \sum_{i=1}^{M} n_i$ τον συνολικό αριθμό πελατών στο δίκτυο. Επίσης, έστω $\mu_i(n_i)$ ο ρυθμός εξυπηρέτησης του κόμβου i (ίδιος για όλες τις κατηγορίες πελατών) αν πρόκειται για κόμβο τύπου 1, και $1/\mu_{ir}$ ο μέσος χρόνος εξυπηρέτησης πελατών της κατηγορίας r στον κόμβο i, αν ο κόμβος i ανήκει σε έναν από τους τύπους 2,3 ή 4.

Θεώρημα 4.3. *Έστω* $\{e_{ir}\}$, $i = 1, 2, \ldots, M$, $r = 1, 2, \ldots, C$, *μία μη αρνητική λύση του συστήματος* (4.20). *Η στατική κατανομή* $p(\boldsymbol{n})$ *υπάρχει εάν* $G < \infty$ *και έχει τη μορφή:*

$$p(\boldsymbol{n}) = \frac{1}{G}\Lambda(N)\prod_{i=1}^{M} g_i(\boldsymbol{n}_i) \tag{4.21}$$

όπου

$$\Lambda(N) = \begin{cases} \prod_{n=1}^{N}\lambda(n-1) & \textit{για ανοικτό δίκτυο} \\ 1 & \textit{για κλειστό δίκτυο} \end{cases} \tag{4.22}$$

$$G = \sum_{\boldsymbol{n}}\Lambda(N)\prod_{i=1}^{M} g_i(\boldsymbol{n}_i) \tag{4.23}$$

και

$$g_i(\boldsymbol{n}_i) = \begin{cases} n_i!\prod_{r=1}^{C}\left[\dfrac{e_{ir}^{n_{ir}}}{n_{ir}!}\right] / \prod_{n=1}^{n_i}\mu_i(n) & \textit{Τύπος 1} \\ n_i!\prod_{r=1}^{C}\left[\dfrac{1}{n_{ir}!}\left(\dfrac{e_{ir}}{\mu_{ir}}\right)^{n_{ir}}\right] & \textit{Τύπος 2 ή 4} \\ \prod_{r=1}^{C}\left[\dfrac{1}{n_{ir}!}\left(\dfrac{e_{ir}}{\mu_{ir}}\right)^{n_{ir}}\right] & \textit{Τύπος 3} \end{cases} \tag{4.24}$$

Παρατηρούμε ότι οι πιθανότητες $p(\boldsymbol{n})$ εξαρτώνται μόνο από τις μέσες τιμές των χρόνων εξυπηρέτησης και ότι η μορφή γινομένου ισχύει όχι μόνο για τους κόμβους αλλά και για τις κατηγορίες πελατών.

Μία περισσότερο συγκεντρωτική περιγραφή μπορεί να γίνει αν θεωρήσουμε την κατάσταση του δικτύου $\boldsymbol{n} = [n_1, n_2, \ldots, n_M]$, η οποία περιλαμβάνει μόνο τον συνολικό αριθμό πελατών σε κάθε κόμβο (για όλες τις κατηγορίες). Στην περίπτωση αυτή, η κατανομή παίρνει απλούστερη μορφή για δίκτυο χωρίς κλειστές αλυσίδες, αν υποθέσουμε ότι ο ρυθμός εξωτερικών αφίξεων λ και οι ρυθμοί εξυπηρέτησης μ_i (Τύπος 1) είναι ανεξάρτητοι από την κατάσταση του δικτύου. Το σύστημα (4.20) θα έχει τότε μία μοναδική λύση $\{e_{ir}\}$, με βάση την οποία ορίζουμε τη συνολική ένταση κυκλοφορίας για κάθε κόμβο i ($i = 1, 2, \ldots, M$):

$$\rho_i = \begin{cases} \sum_{r=1}^{C}(\lambda e_{ir}/\mu_i) & \text{Τύπος 1} \\ \sum_{r=1}^{C}(\lambda e_{ir}/\mu_{ir}) & \text{Τύπος 2, 3 ή 4} \end{cases} \tag{4.25}$$

Η στατική κατανομή $p(\boldsymbol{n})$ εμφανίζεται τώρα σαν γινόμενο ανεξάρτητων κατανομών για τους κόμβους:

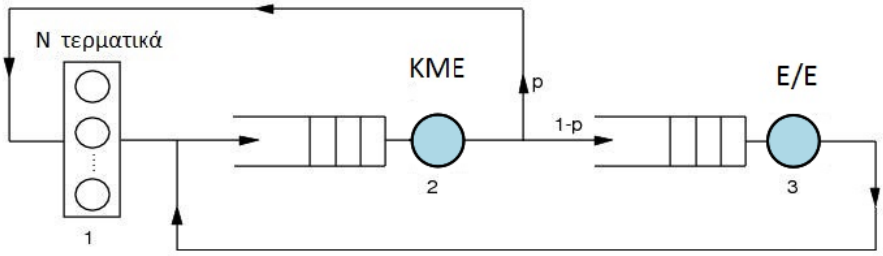
$$p(\boldsymbol{n}) = p_1(n_1)p_2(n_2)\ldots p_M(n_M) \tag{4.26}$$

όπου:

$$p_i(n_i) = \begin{cases} (1-\rho_i)\rho_i^{n_i} & \text{Τύπος 1, 2 ή 4} \\ e^{-\rho_i}\dfrac{\rho_i^{n_i}}{n_i!} & \text{Τύπος 3} \end{cases} \tag{4.27}$$

με την προϋπόθεση ότι $\rho_i < 1$ για τους τύπους 1,2 και 4. Οι επιμέρους κατανομές έχουν την ίδια μορφή όπως το σύστημα $M/M/1$ (για τους τύπους 1,2 και 4) ή το σύστημα $M/M/\infty$ (για τον τύπο 3).

Όσον αφορά τα κλειστά δίκτυα BCMP, υπάρχει το πρόβλημα του υπολογισμού της σταθεράς κανονικοποίησης G, όπως και στα δίκτυα Jackson. Εάν θεωρήσουμε μία μόνο κατηγορία πελατών και σταθερούς ρυθμούς εξυπηρέτησης, μπορεί να εφαρμοστεί η αναδρομική σχέση 4.19, εφόσον όμως δεν υπάρχουν κόμβοι τύπου 3 (IS). Πράγματι, για κόμβους τύπου 3, στους οποίους υπάρχουν τόσες μονάδες εξυπηρέτησης όσοι και οι πελάτες, ο ρυθμός αναχωρήσεων εξαρτάται αναγκαστικά από τον αριθμό πελατών στον σταθμό. Η διαφορά είναι φανερή στις αντίστοιχες εκφράσεις για τον τύπο 3 στις 4.24 και 4.27. Επομένως για τους σταθμούς αυτού του τύπου θα πρέπει να εφαρμοστεί η αναδρομική σχέση 4.16, ενώ η συνθήκη 4.18 παίρνει τη μορφή $G(j,1) = \tau_1^j/j!$, $j = 1, 2, \ldots, N$, εάν ο σταθμός 1 είναι τύπου 3 και $\tau_1 = e_1/\mu_1$. Μία ευνοϊκή περίπτωση είναι όταν υπάρχει μόνο ένας σταθμός τύπου 3 στο δίκτυο, όπως στο παράδειγμα που ακολουθεί. Πράγματι, αν θεωρήσουμε ότι ο σταθμός αυτός είναι ο κόμβος 1 του δικτύου, η επίδρασή του εμφανίζεται



Σχήμα 4.5: *Κλειστό δίκτυο BCMP.*

μόνο στην ποσότητα $G(j,1)$, όπως πιο πάνω, και από εκεί και πέρα μπορεί να εφαρμοστεί η 4.19. Επίσης, στις απλές περιπτώσεις κόμβων με σταθερό ρυθμό, μπορούν να υπολογιστούν δείκτες επίδοσης με βάση τις σχέσεις που ισχύουν για τα κλειστά δίκτυα Jackson. Βέβαια, όλα αυτά ισχύουν όταν υπάρχει μόνο μία κατηγορία πελατών στο δίκτυο.

Για τη γενική περίπτωση κλειστού δικτύου έχουν αναπτυχθεί διάφοροι αλγόριθμοι, οι οποίοι επιτρέπουν τον αριθμητικό υπολογισμό της σταθεράς κανονικοποίησης και την επίλυση του δικτύου. Μία άλλη μέθοδος υπολογισμού δεικτών επίδοσης του δικτύου, χωρίς υπολογισμό της σταθεράς κανονικοποίησης, είναι η ανάλυση μέσης τιμής, η οποία θα εξεταστεί στην επόμενη παράγραφο.

Θα ήταν χρήσιμο να κάνουμε μερικές παρατηρήσεις σχετικά με τη γενικότητα και τη χρησιμότητα του μοντέλου BCMP. Η δυνατότητα ορισμού πολλών κατηγοριών πελατών επιτρέπει την περιγραφή συστημάτων με ιδιαίτερα χαρακτηριστικά, όπως πολύπλοκα κατανεμημένα συστήματα ή δίκτυα επικοινωνίας υπολογιστών. Όσον αφορά τους τύπους των κόμβων, οι περισσότεροι μπορούν να εφαρμοστούν άμεσα σε καταστάσεις πρακτικού ενδιαφέροντος. Οι κόμβοι τύπου 2 (PS) χρησιμοποιούνται συνήθως για την παράσταση κεντρικών υπολογιστικών μονάδων, ενώ οι κόμβοι τύπου 3 (IS) μπορούν να παραστήσουν μονάδες τερματικών (χρόνος σκέψης). Τέλος, οι κόμβοι τύπου 1 (FCFS) χρησιμοποιούνται ιδίως ως μοντέλα μονάδων δευτερεύουσας μνήμης ή συσκευών εισόδου/εξόδου. Όσον αφορά τους κόμβους τύπου 4 (LCFSPR) η χρήση τους είναι μάλλον περιορισμένη, εκτός από τη δυνατότητα παράστασης στοίβας σε μοντέλα δομών δεδομένων.

Παράδειγμα 4.4. *Ας θεωρήσουμε το απλό υπολογιστικό σύστημα του* Σχ.4.5*, το οποίο αποτελείται από* $N = 5$ *τερματικά (τα οποία μπορούν να παρασταθούν από έναν κόμβο τύπου* 3*), μία ΚΜΕ (κόμβος τύπου* 2*) και μία μονάδα δίσκου (κόμβος τύπου* 1*). Πρόκειται για κλειστό δίκτυο με* $M = 3$ *κόμβους, στο οποίο κυκλοφορούν* $N = 5$ *προγράμματα. Υποθέτουμε ότι υπάρχει μία κατηγορία προγραμμάτων και ότι τα προγράμματα που φεύγουν από την ΚΜΕ πηγαίνουν στα τερματικά με πιθανότητα* $p = 0,2$ *και στον δίσκο με πιθανότητα* $1 - p = 0,8$*. Ο μέσος χρόνος σκέψης στα τερματικά είναι* $1/\mu_1 = 10$ sec*, ενώ οι μέσοι χρόνοι εξυπηρέτησης στην ΚΜΕ και στον δίσκο είναι αντίστοιχα* $1/\mu_2 = 0,06$ sec *και* $1/\mu_3 = 0,4$ sec*. Οι αντίστοιχες κατανομές μπορεί να είναι γενικές κατανομές Cox για τους κόμβους* 1 *και* 2*, ενώ για τον κόμβο* 3 *υποτίθεται εκθετική κατανομή.*

Το σύστημα (4.20) γράφεται: $e_1 = pe_2$, $e_2 = e_1 + e_3$, $e_3 = (1-p)e_2$. Μία λύση του προκύπτει αν θέσουμε $e_2 = 1$, οπότε $e_1 = p = 0,2$ και $e_3 = 1-p = 0,8$. Η κατανομή για την κατάσταση του συστήματος $\boldsymbol{n} = [n_1, n_2, n_3]$, όπου n_i ο αριθμός προγραμμάτων στον κόμβο i $(n_1 + n_2 + n_3 = 5)$, θα δίνεται με βάση τις (4.21)– (4.24) από τη σχέση:

$$p(\boldsymbol{n}) = \frac{1}{G}(\tau_1^{n_1}/n_1!)\tau_2^{n_2}\tau_3^{n_3}$$

όπου $\tau_i = e_i/\mu_i$, $i = 1,2,3$, εφόσον έχουμε $C = 1$ και σταθερό ρυθμό εξυπηρέτησης για τον κόμβο τύπου 1. Έχουμε $\tau_1 = 2$, $\tau_2 = 0,06$ και $\tau_3 = 0,32$ (sec). Θα υπολογίσουμε τη σταθερά κανονικοποίησης $G = G(5,3)$ σύμφωνα με τις αναδρομικές σχέσεις:

$$\begin{aligned} G(0,i) &= 1, \quad i = 1,2,3 \\ G(j,1) &= \tau_1^j/j!, \quad j = 1,2,3,4,5 \\ G(j,i) &= G(j,i-1) + \tau_i G(j-1,i), \quad i = 2,3, \quad j = 1,2,3,4,5 \end{aligned}$$

Αντικαθιστώντας βρίσκουμε τελικά: $G(4,3) = 1,5172$ και $G(5,3) = 0,7974$.

Επομένως, οι βαθμοί χρησιμοποίησης της ΚΜΕ και του δίσκου θα είναι αντίστοιχα: $\rho_2 = \tau_2 \frac{G(4,3)}{G(5,3)} = 0,1142$, $\rho_3 = \tau_3 \frac{G(4,3)}{G(5,3)} = 0,6089$, και ο ρυθμός απόδοσης της ΚΜΕ: $\lambda_2 = e_2 \frac{G(4,3)}{G(5,3)} = \mu_2 \rho_2 = 1,9027$ προγράμματα/sec.

Οι ρυθμοί απόδοσης των άλλων κόμβων θα είναι αντίστοιχα: $\lambda_1 = p\lambda_2 = 0,3805$ προγράμματα/sec και $\lambda_3 = (1-p)\lambda_2 = 1,5221$ προγράμματα/sec.

Ο μέσος αριθμός προγραμμάτων στα τερματικά θα είναι $E[n_1] = \lambda_1/\mu_1 = 3,805$ (μέσος αριθμός απασχολημένων τερματικών), άρα ο μέσος αριθμός προγραμμάτων στο υπόλοιπο σύστημα θα είναι $N - E[n_1] = 1,195$. Τέλος, ο μέσος χρόνος απόκρισης του συστήματος, δηλαδή ο μέσος χρόνος που χρειάζεται ένα πρόγραμμα από τη στιγμή που θα φύγει από το τερματικό ως το τέλος της επεξεργασίας του (επιστροφή στο τερματικό), θα είναι σύμφωνα με τον τύπο του Little: $T = (N - E[n_1])/\lambda_1 = N/\lambda_1 - 1/\mu_1 = 3,1406$ sec. □

4.1.2.5 Ανάλυση Μέσης Τιμής

Η λύση μορφής γινομένου στα δίκτυα BCMP χαρακτηρίζεται από μία ενδιαφέρουσα ιδιότητα: από το πλήθος των παραμέτρων που χρειάζονται για τον καθορισμό του δικτύου ελάχιστες τελικά εμφανίζονται στη λύση. Για παράδειγμα, ας θεωρήσουμε ένα κλειστό δίκτυο με μία κατηγορία πελατών, για το οποίο η μήτρα πιθανοτήτων δρομολόγησης $\boldsymbol{Q}$ έχει διαστάσεις $M \times M$ (εάν M ο αριθμός των κόμβων). Αντί για τις M^2 ποσότητες q_{ij}, μόνο οι M ποσότητες e_i που προκύπτουν από τη λύση του συστήματος (4.20) εμφανίζονται στη λύση. Επίσης, όλες οι παράμετροι των γενικών κατανομών Cox εξαφανίζονται από την τελική λύση, στην οποία υπάρχουν μόνο μέσες τιμές των χρόνων εξυπηρέτησης. Τέλος, μπορούμε να πούμε ότι συνήθως στην πράξη δεν ενδιαφερόμαστε τόσο για την ακριβή κατανομή πιθανότητας του αριθμού πελατών σε κάθε σταθμό, αλλά κυρίως για τις μέσες τιμές διαφόρων μεγεθών, όπως είναι ο μέσος αριθμός πελατών στους σταθμούς του δικτύου, ο μέσος χρόνος παραμονής ή ο μέσος ρυθμός απόδοσης. Οι παρατηρήσεις αυτές οδήγησαν στην ανάπτυξη μιας μεθόδου επίλυσης δικτύων, η οποία στηρίζεται αποκλειστικά σε μέσες τιμές. Η μέθοδος αναπτύχθηκε αρχικά από τον M. Reiser [19] και ονομάστηκε *ανάλυση μέσης τιμής* (Mean Value Analysis, MVA).

Η ανάλυση μέσης τιμής στηρίζεται σε μία βασική αρχή που ονομάζεται *Θεώρημα των Αφίξεων* (Arrival (Instant) Theorem). Το Θεώρημα χαρακτηρίζεται από το γεγονός ότι έχει απλή διατύπωση και επιδέχεται πολύ αποτελεσματική διαισθητική ερμηνεία.

Θεώρημα 4.4. *(Reiser) Σε ένα κλειστό δίκτυο με λύση μορφής γινομένου, η κατανομή πιθανότητας της κατάστασης του δικτύου την οποία «βλέπει» ένας πελάτης που φθάνει σε ένα σταθμό, είναι η ίδια με τη στατική κατανομή του δικτύου αν ο πελάτης αυτός έλειπε από το δίκτυο. Ειδικότερα, ο μέσος αριθμός πελατών που βλέπει ο πελάτης κατά την άφιξή του στον σταθμό είναι ο μέσος αριθμός πελατών ανεξαρτήτως άφιξης σε ένα δίκτυο με έναν πελάτη λιγότερο στην κατηγορία του συγκεκριμένου πελάτη.*

Σύμφωνα με το Θεώρημα των Αφίξεων, ο μέσος χρόνος αναμονής ενός πελάτη της κατηγορίας j στον σταθμό i, εξαρτάται από τον μέσο αριθμό πελατών που θα υπήρχε στον σταθμό i αν ο αριθμός πελατών της κατηγορίας j στο δίκτυο ήταν μικρότερος κατά ένα. Όπως θα δούμε στη συνέχεια, από το Θεώρημα προκύπτουν εύχρηστες σχέσεις για τον χρόνο απόκρισης των σταθμών, ειδικά για την περίπτωση σταθερών ρυθμών εξυπηρέτησης. Οι σχέσεις αυτές γενικεύονται και για την περίπτωση μη σταθερών ρυθμών [19]. Επίσης, υπάρχει αντίστοιχη διατύπωση για ανοικτά δίκτυα [21], η οποία διευκολύνει την ανάλυση και ευνοεί την ανάπτυξη αποδοτικών μεθόδων.

Εκτός από το Θεώρημα των Αφίξεων, βασικό ρόλο στην ανάπτυξη της Ανάλυσης Μέσης Τιμής παίζει ο τύπος του Little, ο οποίος μπορεί να εφαρμοστεί σε ολόκληρο το δίκτυο και σε κάθε σταθμό του δικτύου χωριστά.

4.2 Επιχειρησιακή Ανάλυση Δικτύων Αναμονής

Όπως είδαμε σε προηγούμενο κεφάλαιο, η *ντετερμινιστική ή επιχειρησιακή ανάλυση* επιτρέπει τη μελέτη της συμπεριφοράς ενός συστήματος βάσει μετρήσεων που έχουν ληφθεί κατά τη λειτουργία του [5, 7]. Η ανάλυση αυτή, η οποία δεν στηρίζεται σε πιθανοτικές υποθέσεις, μπορεί να οδηγήσει σε ισχυρά αποτελέσματα ευρείας εφαρμογής και χρησιμότητας. Ένα βασικό αποτέλεσμα που μπορεί να εξαχθεί ντετερμινιστικά είναι ο Τύπος του Little.

4.2.1 Συμβολισμοί

Πριν εξετάσουμε ορισμένες σημαντικές επιχειρησιακές σχέσεις, θα συνοψίσουμε τους συμβολισμούς που θα χρησιμοποιηθούν συστηματικά στη συνέχεια σε ό,τι αφορά τα μοντέλα δικτύων αναμονής (πιθανοτικά ή ντετερμινιστικά). Οι συμβολισμοί αυτοί —γενικά— εφαρμόζονται με συνέπεια στα σχετικά κεφάλαια, συνδυάζοντας την ευχρηστία με την τήρηση της συμβατότητας προς την παραδοσιακή βιβλιογραφία.

4.2.1.1 Μια Κατηγορία

M Αριθμός σταθμών στο σύστημα.

N Αριθμός εργασιών (πληθυσμός) στο σύστημα (για κλειστά δίκτυα).

Z Μέσος χρόνος σκέψης των χρηστών (για διαλογικά συστήματα).

λ Μέσος ρυθμός εξωτερικών αφίξεων στο σύστημα (για ανοικτά δίκτυα).

v_i Μέσος αριθμός επισκέψεων μιας εργασίας στον σταθμό i. (Θα χρησιμοποιήσουμε τον μέσο αριθμό επισκέψεων αντί της πιθανότητας δρομολόγησης. Η επιλογή αυτή διευκολύνει και απλοποιεί τη διατύπωση των αλγορίθμων. Εξάλλου, οι ποσότητες αυτές προκύπτουν άμεσα από δεδομένα μετρήσεων.)

S_i Μέση απαίτηση (χρόνος) εξυπηρέτησης ανά επίσκεψη μιας εργασίας στον σταθμό i.

D_i Μέση συνολική απαίτηση εξυπηρέτησης μιας εργασίας στον σταθμό i. (Ισχύει $D_i = v_i S_i$.)

T_i Μέσος χρόνος παραμονής (αναμονή+εξυπηρέτηση) ανά επίσκεψη μιας εργασίας στον σταθμό i.

R_i Μέσος συνολικός χρόνος παραμονής μιας εργασίας στον σταθμό i. (Ισχύει $R_i = v_i T_i$.)

R Μέσος χρόνος απόκρισης του συστήματος. (Ισχύει $R = \sum_i v_i T_i = \sum_i R_i$.)

X_i Ρυθμός απόδοσης του σταθμού i.

X Συνολικός ρυθμός απόδοσης του συστήματος. (Συμπίπτει με τον μέσο ρυθμό αφίξεων λ για ανοικτό δίκτυο σε ισορροπία.) (Ισχύει $X_i = X v_i$.)

U_i Βαθμός χρησιμοποίησης του σταθμού i. (Ισχύει $U_i = X_i S_i = X D_i$.)

Q_i Μέσος αριθμός εργασιών στον σταθμό i. (Ισχύει $Q_i = X_i T_i = X R_i$ — Τύπος του Little.)

Q Μέσος συνολικός αριθμός εργασιών στο σύστημα. (Ισχύει $Q = \sum_i Q_i = XR$. Για κλειστό δίκτυο είναι $Q = N$ σταθερό.)

4.2.1.2 Πολλές Κατηγορίες

M Αριθμός σταθμών στο σύστημα.

C Αριθμός κατηγοριών εργασιών στο σύστημα.

N^j Αριθμός εργασιών (πληθυσμός) της κατηγορίας j στο σύστημα (για κλειστές κατηγορίες). Για κλειστό δίκτυο θέτουμε $\boldsymbol{N} = [N^1, \ldots, N^C]$ και $N = \|\boldsymbol{N}\|$.

Z^j Μέσος χρόνος σκέψης των χρηστών κατηγορίας j (για διαλογικές κατηγορίες).

λ^j Μέσος ρυθμός εξωτερικών αφίξεων της κατηγορίας j στο σύστημα (για ανοικτές κατηγορίες). Για ανοικτό δίκτυο θέτουμε $\boldsymbol{\lambda} = [\lambda^1, \ldots, \lambda^C]$, οπότε $\lambda = \|\boldsymbol{\lambda}\|$.

v_{ij} Μέσος αριθμός επισκέψεων μιας εργασίας της κατηγορίας j στον σταθμό i.

S_{ij} Μέση απαίτηση (χρόνος) εξυπηρέτησης ανά επίσκεψη μιας εργασίας της κατηγορίας j στον σταθμό i.

D_{ij} Μέση συνολική απαίτηση εξυπηρέτησης μιας εργασίας της κατηγορίας j στον σταθμό i. (Ισχύει $D_{ij} = v_{ij}S_{ij}$.)

T_{ij} Μέσος χρόνος παραμονής (αναμονή+εξυπηρέτηση) ανά επίσκεψη μιας εργασίας της κατηγορίας j στον σταθμό i.

R_{ij} Μέσος συνολικός χρόνος παραμονής μιας εργασίας της κατηγορίας j στον σταθμό i. (Ισχύει $R_{ij} = v_{ij}T_{ij}$.)

R^j Μέσος χρόνος απόκρισης του συστήματος για την κατηγορία j. (Ισχύει $R^j = \sum_i v_{ij}T_{ij} = \sum_i R_{ij}$.)

X_{ij} Ρυθμός απόδοσης του σταθμού i για την κατηγορία j.

X^j Συνολικός ρυθμός απόδοσης του συστήματος για την κατηγορία j. (Συμπίπτει με τον μέσο ρυθμό αφίξεων λ^j για ανοικτό δίκτυο σε ισορροπία.) (Ισχύει $X_{ij} = X^j v_{ij}$.)

X Συνολικός ρυθμός απόδοσης του συστήματος. (Συμπίπτει με τον συνολικό μέσο ρυθμό αφίξεων λ για ανοικτό δίκτυο.) (Ισχύει $X = \sum_j X^j$.)

U_{ij} Βαθμός χρησιμοποίησης του σταθμού i για την κατηγορία j. (Ισχύει $U_{ij} = X_{ij}S_{ij} = X^j D_{ij}$.)

Q_{ij} Μέσος αριθμός εργασιών της κατηγορίας j στον σταθμό i. (Ισχύει $Q_{ij} = X_{ij}T_{ij} = X^j R_{ij}$ — Τύπος του Little.)

Q^j Μέσος συνολικός αριθμός εργασιών της κατηγορίας j στο σύστημα. (Ισχύει $Q^j = \sum_i Q_{ij} = X^j R^j$. (Για κλειστή κατηγορία είναι $Q^j = N^j$ σταθερό.)

4.2.2 Επιχειρησιακοί Νόμοι

Ορισμένες απλές σχέσεις έχουν ευρύτατη ισχύ στα συστήματα αναμονής χωρίς να εξαρτώνται από υποθέσεις σχετικές με τις κατανομές των χρόνων εξυπηρέτησης ή των χρόνων μεταξύ αφίξεων. Οι σχέσεις αυτές αναφέρονται ως *επιχειρησιακοί νόμοι* (operational laws) και αφορούν ποσότητες που μπορούν να μετρηθούν κατά τη διάρκεια ενός πεπερασμένου διαστήματος παρατήρησης. Η επιχειρησιακή ανάλυση επιβεβαιώνει σε ντετερμινιστικό πλαίσιο αποτελέσματα που μπορούν να εξαχθούν με στοχαστική ανάλυση.

Έστω ότι κατά την παρατήρηση του σταθμού i για χρονικό διάστημα διάρκειας L μετρήθηκαν α_i αφίξεις, γ_i αναχωρήσεις και διάρκεια απασχόλησης β_i. Θα υποθέσουμε ότι το διάστημα παρατήρησης είναι τέτοιο ώστε ο αριθμός των αφίξεων να είναι ίσος με τον αριθμό των αναχωρήσεων, δηλ. $\alpha_i = \gamma_i$, οπότε ο ρυθμός αφίξεων είναι ίσος με το ρυθμό αναχωρήσεων (ρυθμό απόδοσης). Η υπόθεση αυτή, η οποία εκφράζει την *ισορροπία της ροής* και ελέγχεται με τη βοήθεια μέτρησης, συνεπάγεται ότι δεν δημιουργούνται νέες εργασίες

μέσα στο σύστημα ούτε χάνονται εργασίες στο εσωτερικό του. Από τις παραπάνω μετρήσιμες ποσότητες υπολογίζονται άμεσα ο ρυθμός απόδοσης $X_i = \alpha_i/L = \gamma_i/L$, ο βαθμός χρησιμοποίησης $U_i = \beta_i/L$ και ο μέσος χρόνος εξυπηρέτησης $S_i = \beta_i/\gamma_i$.

(i) **Ο νόμος της χρησιμοποίησης.** Όπως προκύπτει άμεσα από τους ορισμούς

$$U_i = X_i S_i \tag{4.28}$$

Ο νόμος αυτός μπορεί να θεωρηθεί και ως ειδική περίπτωση του νόμου του Little.

(ii) **Ο νόμος (τύπος) του Little.** Είναι ο πιο διαδεδομένος τύπος της θεωρίας αναμονής [15] και ισχύει για κάθε σύστημα ή μέρος συστήματος για το οποίο ικανοποιείται η υπόθεση της ισορροπίας της ροής. Αν Q_i και T_i είναι ο μέσος αριθμός εργασιών στον σταθμό i και ο μέσος χρόνος απόκρισης (παραμονής) των εργασιών ανά επίσκεψη στον σταθμό i, αντίστοιχα, ο νόμος του Little μπορεί να διατυπωθεί ως εξής:

$$Q_i = X_i T_i \tag{4.29}$$

Στο προηγούμενο κεφάλαιο είδαμε μια απόδειξη του τύπου του Little στο πλαίσιο της επιχειρησιακής ανάλυσης. Εδώ θα παραθέσουμε μια απλούστερη ερμηνεία που παρέχει κυρίως διαισθητική αιτιολόγηση του νόμου. Έστω τυχαίος πελάτης που φθάνει στον σταθμό i. Κατά την άφιξή του, θα πρέπει να βρεί κατά μέσο όρο Q_i πελάτες. Παραμένει στον σταθμό για μέσο χρόνο T_i. Άρα, $X_i T_i$ πελάτες κατά μέσο όρο πρέπει να αφίχθηκαν κατά τη διάρκεια της παραμονής του. Σε κατάσταση ισορροπίας, ο μέσος αριθμός πελατών που παραμένουν στον σταθμό κατά την αναχώρηση του πελάτη πρέπει να ισούται με τον μέσο αριθμό πελατών που βρήκε κατά την άφιξή του.

(iii) **Ο νόμος της υποχρεωτικής ροής.** Ο νόμος αυτός συσχετίζει τον ρυθμό απόδοσης ενός συστήματος (δικτύου) ως συνόλου με τους ρυθμούς απόδοσης των επιμέρους σταθμών που περιλαμβάνονται στο σύστημα.

Σε ένα ανοικτό σύστημα ο ρυθμός απόδοσης ορίζεται ως ο αριθμός εργασιών που εγκαταλείπουν το σύστημα ανά μονάδα χρόνου. Σε κλειστά συστήματα οι εργασίες δεν εγκαταλείπουν πραγματικά το σύστημα. Υπάρχει όμως πάντοτε κάποια σύνδεση η οποία συμπεριφέρεται ως «εξωτερική» σύνδεση, με την έννοια ότι μία εργασία που διασχίζει τη σύνδεση αυτή είναι σαν να εγκαταλείπει το σύστημα και να επανέρχεται στιγμιαία (ή ισοδύναμα σαν να αντικαθίσταται από μία άλλη παρόμοια). Συνεπώς, ο ρυθμός απόδοσης του κλειστού συστήματος ορίζεται ως ο αριθμός εργασιών που διασχίζουν την *εξωτερική σύνδεση ή σύνδεση αναφοράς* στη μονάδα του χρόνου . Σε κλειστά συστήματα με τερματικά (διαλογικά συστήματα), η σύνδεση αναφοράς είναι αυτή που διέρχεται από τα τερματικά.

Σε σχέση με τα παραπάνω, θα πρέπει να αναφερθεί η βασική αρχή ότι, στα μοντέλα υπολογιστικών συστημάτων, ο μέσος αριθμός επισκέψεων μιας εργασίας στην ΚΜΕ υπερβαίνει κατά 1 τον μέσο συνολικό αριθμό επισκέψεων στις περιφερειακές μονάδες. Αυτό οφείλεται στο γεγονός ότι οι εργασίες που ολοκληρώνονται εγκαταλείπουν το σύστημα από την ΚΜΕ, οδεύοντας προς την εξωτερική σύνδεση αναφοράς, άρα πρόκειται για μια αναχώρηση από την ΚΜΕ επιπλέον των αναχωρήσεων με προορισμό τις περιφερειακές μονάδες.

Αν συμβολίσουμε με γ τον αριθμό των εργασιών που εγκαταλείπουν το σύστημα στην περίοδο παρατήρησης L (προς τον έξω κόσμο ή από την «εξωτερική» σύνδεση), τότε ο ρυθμός απόδοσης του συνολικού συστήματος θα είναι $X = \gamma/L$. Αν κάθε εργασία πραγματοποιεί v_i επισκέψεις στον σταθμό i και ισχύει η υπόθεση της ισορροπίας της ροής, θα έχουμε $\gamma_i = \gamma v_i$, απ' όπου προκύπτει ο νόμος της υποχρεωτικής ροής:

$$X_i = X v_i \tag{4.30}$$

Οι ποσότητες v_i εκφράζουν τη δρομολόγηση των εργασιών στο δίκτυο και σχετίζονται άμεσα με τις πιθανότητες δρομολόγησης όπως αυτές χρησιμοποιούνται στα κλασικά αποτελέσματα της θεωρίας αναμονής.

Συνδυασμός του νόμου αυτού και του νόμου της χρησιμοποίησης δίνει:

$$U_i = XD_i \tag{4.31}$$

όπου $D_i = v_i S_i$ είναι η μέση συνολική απαίτηση εξυπηρέτησης μιας εργασίας στον σταθμό i (για όλες τις επισκέψεις της εργασίας στον σταθμό αυτό). Η Εξίσωση (4.31) συνεπάγεται ότι ο σταθμός με τη μεγαλύτερη μέση απαίτηση εξυπηρέτησης D_i θα έχει και τον υψηλότερο βαθμό χρησιμοποίησης. Ο σταθμός αυτός αποτελεί τη *στένωση* (bottleneck) του συστήματος.

(iv) **Ο νόμος του χρόνου απόκρισης.** Έστω $Q = \sum_{i=1}^{M} Q_i$ ο μέσος συνολικός αριθμός εργασιών σε δίκτυο M σταθμών και R ο μέσος χρόνος απόκρισης του δικτύου, ο οποίος εκφράζει το μέσο συνολικό χρόνο παραμονής μιας εργασίας στο δίκτυο (ανοικτό σύστημα) ή το μέσο χρόνο ανάμεσα σε δύο διαδοχικές διελεύσεις μιας εργασίας από τη σύνδεση αναφοράς (κλειστό σύστημα). Ο νόμος του Little για το συνολικό δίκτυο δίνει $Q = XR$. Θεωρώντας την εφαρμογή του νόμου του Little σε καθέναν από τους σταθμούς, όπως δίνεται από την Εξίσωση (4.29), καθώς και το νόμο της υποχρεωτικής ροής βρίσκουμε τον νόμο του χρόνου απόκρισης:

$$R = \sum_{i=1}^{M} v_i T_i \tag{4.32}$$

Σε ένα διαλογικό σύστημα ο νόμος του χρόνου απόκρισης διατυπώνεται ελαφρώς διαφορετικά, αν ορίσουμε τον μέσο χρόνο απόκρισης ως τον μέσο χρόνο από τη στιγμή που μία εργασία φεύγει από τα τερματικά μέχρι τη στιγμή που επιστρέφει σε αυτά (σύνδεση αναφοράς). Ο χρόνος σκέψης Z των χρηστών στα τερματικά δεν περιλαμβάνεται στον χρόνο απόκρισης R του συστήματος. (Ο χρόνος $R + Z$ είναι η μέση διάρκεια ενός πλήρους κύκλου μιας εργασίας στο σύστημα.) Αν N είναι τώρα ο συνολικός αριθμός χρηστών (τερματικών) στο σύστημα θα έχουμε

$$X = \frac{N}{R + Z}$$

ή

$$R = N/X - Z \tag{4.33}$$

Η τελευταία εξίσωση εκφράζει τον νόμο του χρόνου απόκρισης για διαλογικά συστήματα.

Παράδειγμα 4.5. *Ο εξυπηρετητής ενός μεγάλου εταιρικού δικτύου (intranet server) δέχεται δύο τύπους φορτίου: ερωτήσεις από* 150 *πελάτες και εργασίες ενημέρωσης από* 23 *διαχειριστές. Ο μέσος χρόνος σκέψης των πελατών είναι* 38 sec *και των διαχειριστών* 82 sec*. Σύμφωνα με μετρήσεις, ο μέσος αριθμός προσπελάσεων στο δίσκο είναι* 8 *ανά ερώτηση πελάτη και* 103 *ανά εργασία ενημέρωσης. Ο μέσος χρόνος εξυπηρέτησης ανά προσπέλαση στον δίσκο είναι* 17 msec *και* 25 msec *για τις δύο κατηγορίες, αντίστοιχα. Αν σε διάστημα* 30 *min ο χρόνος απασχόλησης του δίσκου ήταν* 1671 *εμπησec και ολοκληρώθηκαν συνολικά* 5482 *εργασίες των δύο κατηγοριών, ζητείται ο μέσος χρόνος απόκρισης και ο ρυθμός απόδοσης κάθε κατηγορίας.*

Συμβολίζουμε με δ τον δίσκο και θεωρούμε δύο κατηγορίες εργασιών, A και B, που αντιστοιχούν στους πελάτες και στους διαχειριστές. Σύμφωνα με τις δοθείσες τιμές έχουμε:
$Z^A = 38$ sec, $Z^B = 82$ sec, $S_{\delta A} = 0,017$ sec, $S_{\delta B} = 0,025$ sec, $v_{\delta A} = 8$, $v_{\delta B} = 103$, απ' όπου $D_{\delta A} = 0,136$ sec, $D_{\delta B} = 2,575$ sec. Επίσης, $U_\delta = 1671/1800 = 0,928$ και $X = 5482/1800 = 3,046$ εργασίες/sec. Συνεπώς, με εφαρμογή των επιχειρησιακών νόμων:

$$\begin{aligned} X^A D_{\delta A} + X^B D_{\delta B} &= U_\delta \\ X^A + X^B &= X \end{aligned}$$

Από την επίλυση του συστήματος προκύπτει: $X^A = 2,835$ εργασίες/sec και $X^B = 0,211$ εργασίες/sec. Από τον νόμο του χρόνου απόκρισης για διαλογικά συστήματα παίρνουμε τελικά: $R^A = 14,91$ sec και $R^B = 27,005$ sec. □

4.3 Αλγόριθμοι για την Επίλυση Δικτύων Αναμονής

Οι βασικοί αλγόριθμοι που χρησιμοποιούνται αφορούν μοντέλα ανοικτών, κλειστών και μικτών δικτύων και στηρίζονται κατά το κύριο μέρος στην *ανάλυση μέσης τιμής*. Ειδικότερα, χρησιμοποιείται το *θεώρημα των αφίξεων* (arrival instant theorem) [19, 21, 23], το οποίο ισχύει για δίκτυα με λύση μορφής γινομένου και χαρακτηρίζει την κατανομή των πελατών σε ένα σταθμό του δικτύου όπως τη βλέπει ένας πελάτης που φθάνει σε αυτόν τον σταθμό. Στους βασικούς αλγορίθμους επίλυσης δικτύων αναμονής στηρίζεται η ανάπτυξη προσεγγιστικών τεχνικών, οι οποίες επιτρέπουν την ανάλυση πολύπλοκων συστημάτων ή συστημάτων με ιδιαίτερα χαρακτηριστικά.

Γενικά, οι είσοδοι ενός μοντέλου υπολογιστικού συστήματος υποδηλώνουν το *φορτίο* του συστήματος (workload) και αποτελούνται από την *ένταση φορτίου* (workload intensity) και τις *απαιτήσεις εξυπηρέτησης* (service demands) . Οι έξοδοι του μοντέλου αφορούν δείκτες επίδοσης του συστήματος όπως τον βαθμό χρησιμοποίησης, τον ρυθμό απόδοσης, τον χρόνο απόκρισης και τον μέσο αριθμό εργασιών.

Διακρίνουμε τις περιπτώσεις μοντέλων με μία κατηγορία και με πολλές κατηγορίες εργασιών. Υποθέτουμε σταθερούς ρυθμούς εξυπηρέτησης στους σταθμούς και ότι οι εργασίες δεν αλλάζουν κατηγορία. Η πρώτη υπόθεση θα ξανασυζητηθεί αργότερα, ενώ η δεύτερη αποδεικνύεται ότι δεν βλάπτει τη γενικότητα.

Συνεπώς, θα θεωρήσουμε τρεις τύπους δικτύων και θα αναπτύξουμε τους αντίστοιχους αλγορίθμους επίλυσης.

- *Ανοικτά δίκτυα*. Εργασίες (jobs) μιας ή πολλών κατηγοριών εισέρχονται στο δίκτυο προερχόμενες από τον έξω κόσμο. Όταν ολοκληρώνουν την εξυπηρέτησή τους αναχωρούν για τον έξω κόσμο. Συνεπώς, ο πληθυσμός των εργασιών στο δίκτυο μεταβάλλεται με τον χρόνο. Οι εργασίες (αιτήσεις εξυπηρέτησης) που υποβάλλονται σε ανοικτό σύστημα χαρακτηρίζονται συνήθως ως *συναλλαγές* (transactions).

- *Κλειστά δίκτυα*. Σε ένα κλειστό σύστημα περιέχεται ένας σταθερός αριθμός εργασιών μιας ή πολλών κατηγοριών, που κυκλοφορούν μεταξύ των σταθμών του δικτύου. Στην πράξη, οι εργασίες έχουν πεπερασμένη διάρκεια παραμονής στο δίκτυο και, όταν μια εργασία ολοκληρώνεται, αντικαθίσταται στιγμιαία από μια άλλη στατιστικά παρόμοια. Συνήθως, οι εργασίες είναι *διαλογικές* (interactive), προερχόμενες από έναν σταθερό αριθμό ενεργών *τερματικών σταθμών εργασίας* (terminal workstations), και χαρακτηρίζονται από *χρόνο σκέψης*, δηλαδή χρόνο χρήσης του τερματικού σταθμού μεταξύ διαδοχικών αλληλεπιδράσεων με το σύστημα. Άλλο είδος εργασιών είναι αυτές που προέρχονται από ένα πλήθος *έτοιμων* εργασιών (batch) που εκτελούνται με σταθερό βαθμό πολυπρογραμματισμού, ώστε πάντα να υπάρχει σταθερός αριθμός ενεργών εργασιών στο δίκτυο.

- *Μικτά δίκτυα*. Ένα μικτό δίκτυο περιλαμβάνει πολλές κατηγορίες πελατών και αποτελεί συνδυασμό των δύο προηγούμενων τύπων. Οι πελάτες κάθε κατηγορίας (ανοικτής ή κλειστής) συμπεριφέρονται με κοινό τρόπο.

Θα θεωρήσουμε δύο τύπους σταθμών εξυπηρέτησης: *σταθμούς αναμονής* και *σταθμούς καθυστέρησης*. Στην πρώτη περίπτωση, οι εργασίες ανταγωνίζονται για τη χρήση του εξυπηρετητή. Ο χρόνος παραμονής της εργασίας στον σταθμό αποτελείται από τον χρόνο αναμονής και τον χρόνο εξυπηρέτησης. Ο χρόνος αναμονής αντιπροσωπεύει την επιβάρυνση καθεμιάς εργασίας λόγω της παρουσίας των υπολοίπων. Δεν εκφράζεται πάντα ως χρονική διάρκεια, αλλά μπορεί να λάβει διάφορες εκφράσεις ανάλογα με τον ισχύοντα κανονισμό εξυπηρέτησης. Αντιθέτως, σε έναν σταθμό καθυστέρησης, οι εργασίες δεν μοιράζονται την ισχύ του εξυπηρετητή όπως παραπάνω, αλλά σε κάθε εργασία διατίθεται αποκλειστικά ένας εξυπηρετητής. Συνεπώς, δεν υπάρχει ανταγωνισμός και ο χρόνος παραμονής ταυτίζεται με τον χρόνο εξυπηρέτησης. Οι σταθμοί καθυστέρησης είναι χρήσιμοι για τη μοντελοποίηση καταστάσεων όπου μια εργασία υφίσταται καθυστέρηση γνωστής μέσης τιμής ανεξάρτητα από τις υπόλοιπες εργασίες. Η συνηθέστερη χρήση των σταθμών καθυστέρησης αφορά την παράσταση του χρόνου σκέψης σε τερματικούς σταθμούς εργασίας.

Όπως αναφέρθηκε παραπάνω, θα θεωρήσουμε καταρχάς σταθερούς ρυθμούς εξυπηρέτησης για τους σταθμούς αναμονής. Στην περίπτωση αυτή, έχουμε σταθμό (με ρυθμό εξυπηρέτησης) *ανεξάρτητο από το*

φορτίο (load-independent, LI). Στην αντίθετη περίπτωση, που θα εξεταστεί χωριστά, ο ρυθμός εξυπηρέτησης ενός σταθμού εξαρτάται από το φορτίο, δηλαδή είναι συνάρτηση του αριθμού εργασιών στον σταθμό. Πρόκειται για σταθμό *με εξάρτηση από το φορτίο* (load-dependent, LD).

4.3.1 Ανοικτά Δίκτυα

Τα μοντέλα ανοικτών δικτύων χρησιμοποιούνται για την ανάλυση συστημάτων στα οποία οι εργασίες φθάνουν με ένα δεδομένο ρυθμό προερχόμενες από ένα πληθυσμό που μεταβάλλεται (transactions). Τα αποτελέσματα που παραθέτουμε διατυπώνονται εδώ με έμφαση στη φυσική ερμηνεία τους, αλλά βασίζονται άμεσα στο θεώρημα BCMP.

4.3.1.1 Μία Κατηγορία

Ο ρυθμός απόδοσης X του συστήματος σε ισορροπία ταυτίζεται με τον ρυθμό αφίξεων λ, ο οποίος αποτελεί δεδομένο εισόδου του μοντέλου και χαρακτηρίζει την ένταση φορτίου. Θα θεωρήσουμε το λ ως όρισμα για τις ποσότητες που μας ενδιαφέρουν.

Ρυθμός απόδοσης

$$X_i(\lambda) = \lambda v_i$$

Βαθμός χρησιμοποίησης

$$U_i(\lambda) = X_i(\lambda)S_i = \lambda D_i$$

Χρόνος παραμονής

Διακρίνουμε σταθμούς που προκαλούν μόνο καθυστέρηση (Τύπος 3 του μοντέλου BCMP) και σταθμούς που περιλαμβάνουν αναμονή (Τύποι 1,2 και 4 του μοντέλου BCMP).

Για σταθμούς με καθυστέρηση ισχύει:

$$R_i(\lambda) = v_i S_i = D_i$$

Για σταθμούς με αναμονή ο χρόνος R_i είναι το άθροισμα του χρόνου εξυπηρέτησης και του χρόνου αναμονής. Η πρώτη συνιστώσα είναι $v_i S_i$. Για σταθμούς με κανονισμό FIFO η δεύτερη συνιστώσα είναι ο χρόνος αναμονής μέχρι να εξυπηρετηθούν οι εργασίες που είναι ήδη στην ουρά, όταν φθάνει μία εργασία. Σύμφωνα με τις υποθέσεις του μοντέλου BCMP αποδεικνύεται ότι *η κατανομή πελατών σε ένα σταθμό κατά την στιγμή άφιξης ενός πελάτη στον σταθμό, είναι η κατανομή ισορροπίας των πελατών στον σταθμό για τυχούσα χρονική στιγμή (ανεξάρτητα από την άφιξη)*. Αυτή είναι η διατύπωση του θεωρήματος των αφίξεων για ανοικτό δίκτυο [23]. Άρα έχουμε:

$$R_i(\lambda) = v_i S_i + v_i S_i n_i(\lambda) = D_i[1 + n_i(\lambda)] = D_i[1 + \lambda R_i(\lambda)]$$

με χρήση του τύπου του Little, και τελικά:

$$R_i(\lambda) = \frac{D_i}{1 - U_i(\lambda)}$$

Η σχέση αυτή ισχύει και για σταθμούς τύπου 2 ή 4, για τους οποίους όμως η φυσική ερμηνεία δεν είναι προφανής, όπως για τους σταθμούς FIFO. Στην περίπτωση αυτή, ο παράγων $1-U_i(\lambda)$ εκφράζει τη «διαστολή» του χρόνου εξυπηρέτησης ενός πελάτη, η οποία οφείλεται στην παρουσία άλλων πελατών (expansion factor). Άρα γενικά:

$$R_i(\lambda) = \begin{cases} D_i & \text{(καθυστέρηση)} \\ \dfrac{D_i}{1 - U_i(\lambda)} & \text{(αναμονή)} \end{cases} \tag{4.34}$$

Μέσος αριθμός εργασιών στον σταθμό

$$Q_i(\lambda) = \lambda R_i(\lambda) = \begin{cases} U_i(\lambda) & \text{(καθυστέρηση)} \\ \dfrac{U_i(\lambda)}{1 - U_i(\lambda)} & \text{(αναμονή)} \end{cases} \tag{4.35}$$

(Η μορφή της τελευταίας σχέσης για σταθμούς αναμονής είναι γνωστή από το σύστημα $M/M/1$).
Μέσος χρόνος απόκρισης του συστήματος

$$R(\lambda) = \sum_{i=1}^{M} R_i(\lambda) \tag{4.36}$$

Μέσος αριθμός εργασιών στο σύστημα

$$Q(\lambda) = \sum_{i=1}^{M} Q_i(\lambda) = \lambda R(\lambda) \tag{4.37}$$

Παράδειγμα 4.6. *Ένας εξυπηρετητής αρχείων (file server) περιλαμβάνει μία ΚΜΕ και δύο δίσκους. Κατά τη λειτουργία του συστήματος σε σύνδεση με άλλα υπολογιστικά συστήματα (clients) μετρήθηκαν σε διάστημα* 1 h *οι ακόλουθες ποσότητες:*

Αριθμός εργασιών από πελάτες προς εξυπηρετητή	8530
Χρόνος απασχόλησης ΚΜΕ	2148 sec
Χρόνος απασχόλησης Δίσκου 1	1836 sec
Χρόνος απασχόλησης Δίσκου 2	2707 sec

Με τη βοήθεια μοντέλου ανοικτού δικτύου υπολογίζεται ο μέσος χρόνος απόκρισης του συστήματος. Εν συνεχεία, θα εξεταστεί αν βελτιώνεται ο μέσος χρόνος απόκρισης με χρήση λανθάνουσας μνήμης (cache) για τον Δίσκο 2*, δεδομένου ότι στην περίπτωση αυτή ο αριθμός των επισκέψεων στον Δίσκο* 2 *θα μειωθεί κατά* 35% *με ταυτόχρονη επιβάρυνση του χρόνου ΚΜΕ κατά* 20% *και του χρόνου εξυπηρέτησης ανά επίσκεψη στον Δίσκο* 2 *κατά* 10%*.*

Αντιστοιχίζουμε τους αριθμούς 1, 2 και 3 στην ΚΜΕ και τους δύο δίσκους. Από τα δεδομένα των μετρήσεων έχουμε: $X = 8530/3600 = 2,3694$ εργασίες/sec, $D_1 = 2148/8530 = 0,2518$ sec, $D_2 = 1836/8530 = 0,2152$ sec, $D_3 = 2707/8530 = 0,3174$ sec.

Από τον νόμο της χρησιμοποίησης έχουμε $U_1 = 0,597$, $U_2 = 0,51$ και $U_3 = 0,752$ και χρησιμοποιώντας τις (4.34) και (4.36) βρίσκουμε τον χρόνο απόκρισης: $R_1 = 0,625$ sec, $R_2 = 0,439$ sec, $R_3 = 1,280$ sec και $R = 2,344$ sec.

Αν χρησιμοποιηθεί λανθάνουσα μνήμη, τα δεδομένα τροποποιούνται ως εξής: $D_1' = 1,20 \times D_1 = 0,3022$ sec και $D_3' = v_3' S_3' = (0,65 \times v_3)(1,10 \times S_3) = 0,65 \times 1,10 \times D_3 = 0,2269$ sec, απ' όπου βρίσκουμε $U_1' = 0,716$, $U_3' = 0,538$ και τελικά $R_1' = 1,064$, $R_3' = 0,491$ και $R' = 1,994$. Άρα, ο μέσος χρόνος απόκρισης βελτιώνεται. □

4.3.1.2 Πολλές Κατηγορίες

Η είσοδος του συστήματος εκφράζεται από το διάνυσμα $\boldsymbol{\lambda} = [\lambda^1, \ldots, \lambda^C]$, όπου λ^j ο ρυθμός άφιξης της κατηγορίας j. Τα αποτελέσματα του απλού μοντέλου μιας κατηγορίας γενικεύονται άμεσα.
Ρυθμός απόδοσης

$$X_{ij}(\boldsymbol{\lambda}) = \lambda^j v_{ij}$$

Βαθμός χρησιμοποίησης

$$U_{ij}(\boldsymbol{\lambda}) = X_{ij}(\boldsymbol{\lambda}) S_{ij} = \lambda^j D_{ij}$$

Χρόνος παραμονής

Για σταθμό με καθυστέρηση έχουμε:

$$R_{ij}(\boldsymbol{\lambda}) = D_{ij}$$

Για την περίπτωση σταθμού με αναμονή έχουμε, σύμφωνα με το θεώρημα των αφίξεων:

$$R_{ij}(\boldsymbol{\lambda}) = D_{ij}\left[1 + \sum_{k=1}^{C} Q_{ik}(\boldsymbol{\lambda})\right] = D_{ij}\left[1 + \sum_{k=1}^{C} \lambda^k R_{ik}(\boldsymbol{\lambda})\right] \tag{4.38}$$

οπότε:

$$\frac{R_{ij}(\boldsymbol{\lambda})}{R_{ik}(\boldsymbol{\lambda})} = \frac{D_{ij}}{D_{ik}} \tag{4.39}$$

ή

$$R_{ik}(\boldsymbol{\lambda}) = \frac{D_{ik}}{D_{ij}} R_{ij}(\boldsymbol{\lambda}) \tag{4.40}$$

και με αντικατάσταση προκύπτει:

$$R_{ij}(\boldsymbol{\lambda}) = \frac{D_{ij}}{1 - \sum_{k=1}^{C} U_{ik}(\boldsymbol{\lambda})} \tag{4.41}$$

Ο παράγων διαστολής περιλαμβάνει τώρα την επίδραση όλων των κατηγοριών. Άρα γενικά:

$$R_{ij}(\boldsymbol{\lambda}) = \begin{cases} D_{ij} & \text{(καθυστέρηση)} \\ \dfrac{D_{ij}}{1 - \sum_{k=1}^{C} U_{ik}(\boldsymbol{\lambda})} & \text{(αναμονή)} \end{cases} \tag{4.42}$$

Μέσος αριθμός εργασιών στον σταθμό

$$Q_{ij}(\boldsymbol{\lambda}) = \lambda^j R_{ij}(\boldsymbol{\lambda}) = \begin{cases} U_{ij}(\boldsymbol{\lambda}) & \text{(καθυστέρηση)} \\ \dfrac{U_{ij}(\boldsymbol{\lambda})}{1 - \sum_{k=1}^{C} U_{ik}(\boldsymbol{\lambda})} & \text{(αναμονή)} \end{cases} \tag{4.43}$$

Μέσος χρόνος απόκρισης του συστήματος

$$R^j(\boldsymbol{\lambda}) = \sum_{i=1}^{M} R_{ij}(\boldsymbol{\lambda}) \tag{4.44}$$

Μέσος αριθμός εργασιών στο σύστημα

$$Q_j(\boldsymbol{\lambda}) = \sum_{i=1}^{M} Q_{ij}(\boldsymbol{\lambda}) = \lambda^j R^j(\boldsymbol{\lambda}) \tag{4.45}$$

Παράδειγμα 4.7. *Ένας εξυπηρετητής βάσης δεδομένων (database server) περιλαμβάνει μια ΚΜΕ και έναν δίσκο και δέχεται φορτίο δύο κατηγοριών. Ο ρυθμός αφίξεων είναι* $8,7$ *εργασίες/*sec *συνολικά για τις δύο κατηγορίες. Από μετρήσεις έχουν προκύψει οι παρακάτω τιμές για τη μέση συνολική απαίτηση εξυπηρέτησης:*

Κατηγορία	*Απαίτηση εξυπηρέτησης* (sec)	
	ΚΜΕ	*Δίσκος*
A	$0,05$	$0,04$
B	$0,12$	$0,09$

Ζητείται το ποσοστό των αφίξεων που ανήκει στην κατηγορία A*, αν ο μέσος χρόνος απόκρισης της κατηγορίας* B *είναι* $0,5$ sec.

Έχουμε $\lambda = \lambda^A + \lambda^B = 8,7$ εργασίες/sec. Αν x είναι το ποσοστό των αφίξεων που ανήκουν στην κατηγορία A, θα είναι $\lambda^A = 8,7x$ και $\lambda^B = 8,7(1-x)$. Έστω ότι 1 και 2 παριστάνουν την ΚΜΕ και τον δίσκο, αντίστοιχα. Τα δεδομένα είναι $D_{1A} = 0,05$ sec, $D_{1B} = 0,12$ sec, $D_{2A} = 0,04$ sec, $D_{2B} = 0,09$ sec, απ' όπου λαμβάνουμε με χρήση του νόμου της χρησιμοποίησης: $U_{1A} = 0,435x$, $U_{1B} = 1,044(1-x)$, $U_{2A} = 0,348x$, $U_{2B} = 0,783(1-x)$. Προκειμένου να υπάρχει λύση στη μόνιμη κατάσταση, θα πρέπει να ισχύει η συνθήκη ισορροπίας στους σταθμούς του δικτύου: $U_1 = U_{1A} + U_{1B} < 1$ και $U_2 = U_{2A} + U_{2B} < 1$. Από τις συνθήκες προκύπτουν οι περιορισμοί: $x > 0,0723$ και $x > -0,4989$, ή τελικά $x > 0,0723$.

Υπό την αίρεση ικανοποίησης του περιορισμού, εφαρμογή των (4.42) και (4.44) δίνει:

$$R^B = R_{1B} + R_{2B} = \frac{0,12}{0,609x - 0,044} + \frac{0,09}{0,435x + 0,217} = 0,5$$

ή $0,1325x^2 - 0,0505x - 0,0268 = 0$, απ' όπου $x_1 = 0,679$ και $x_2 = -0,298$. Προφανώς, εξετάζεται μόνο η x_1, η οποία ικανοποιεί τον περιορισμό και δίνει τελικά $\lambda^A = 5,9$ εργασίες/sec και $\lambda^B = 2,8$ εργασίες/sec. □

4.3.2 Κλειστά Δίκτυα

Για κλειστά δίκτυα το θεώρημα των αφίξεων διατυπώνεται ως εξής: *η κατανομή της κατάστασης του δικτύου κατά τη στιγμή άφιξης ενός πελάτη σε ένα σταθμό είναι η κατανομή ισορροπίας του δικτύου αν ο πελάτης αυτός έλειπε από το δίκτυο* [19, 21]. Οι βασικές τεχνικές επίλυσης κλειστών δικτύων στηρίζονται στην *ανάλυση μέσης τιμής* (MVA), της οποίας θα δώσουμε εδώ μία απλή διατύπωση, χάρη στη χρήση των ποσοτήτων v_i και v_{ij} που θεωρούνται δεδομένα εισόδου.

Τα κλειστά δίκτυα χρησιμοποιούνται για την ανάλυση συστημάτων, στα οποία το φορτίο αποτελείται είτε από ένα σταθερό αριθμό ενεργών εργασιών (batch), είτε από ένα σταθερό αριθμό ενεργών τερματικών. Στην πρώτη περίπτωση ουσιαστικά εννοούμε ένα σύστημα με σταθερό μέσο βαθμό πολυπρογραμματισμού, στο οποίο πάντα υπάρχουν εργασίες έτοιμες για εκτέλεση. Στη δεύτερη περίπτωση λαμβάνεται υπόψη και ο χρόνος σκέψης των χρηστών στα τερματικά, τα οποία παριστάνονται σαν ένας σταθμός που απλά προκαλεί καθυστέρηση (Τύπος 3 του μοντέλου BCMP).

4.3.2.1 Μία Κατηγορία

Η είσοδος του συστήματος παριστάνεται από τον αριθμό N των εργασιών στο σύστημα, ο οποίος εμφανίζεται σαν όρισμα των ποσοτήτων που υπολογίζονται.

Οι βασικές σχέσεις είναι:

$$R_i(N) = \begin{cases} D_i & \text{(καθυστέρηση)} \\ D_i[1 + Q_i(N-1)] & \text{(αναμονή)} \end{cases} \tag{4.46}$$

$$X(N) = \frac{N}{\sum_{i=1}^{M} R_i(N)} \tag{4.47}$$

$$Q_i(N) = X(N)R_i(N) \tag{4.48}$$

Η σχέση (4.46) βασίζεται στο θεώρημα των αφίξεων, ενώ οι σχέσεις (4.47) και (4.48) αποτελούν εφαρμογή του τύπου του Little σε ολόκληρο το σύστημα και σε κάθε σταθμό αντίστοιχα. Η εξάρτηση των ποσοτήτων με όρισμα N από την ποσότητα $Q_i(N-1)$ δεν επιτρέπει τον άμεσο υπολογισμό και επιβάλλει επαναληπτική εφαρμογή των εξισώσεων ξεκινώντας από μηδενική κατάσταση. Παρακάτω, η διαδικασία διατυπώνεται προγραμματιστικά με χρήση ψευδογλώσσας προγραμματισμού (Αλγόριθμος 4.1).

Αλγόριθμος 4.1. *Ανάλυση Μέσης Τιμής (Μία κατηγορία) — Ακριβής λύση*

for $i \Leftarrow 1$ to M do $Q_i \Leftarrow 0$
for $k \Leftarrow 1$ to N do
begin
for $i \Leftarrow 1$ to M do $R_i \Leftarrow \begin{cases} D_i & \text{(καθυστέρηση)} \\ D_i(1 + Q_i) & \text{(αναμονή)} \end{cases}$
$X \Leftarrow \dfrac{k}{\sum_{i=1}^{M} R_i}$
for $i \Leftarrow 1$ to M do $Q_i \Leftarrow XR_i$
end

Οι απαιτήσεις του αλγορίθμου είναι $O(NM)$ σε χρόνο και $O(M)$ σε χώρο.

Παράδειγμα 4.8. *Ο εξυπηρετητής ιστού (web server) μιας εταιρείας περιλαμβάνει μια ΚΜΕ και έναν δίσκο. Κάθε ερώτηση* http *προς το σύστημα απαιτεί κατά μέσο όρο* 124 msec *συνολικής εξυπηρέτησης στην ΚΜΕ και* 230 msec *συνολικής εξυπηρέτησης στον δίσκο. Υποθέτουμε ότι κάθε στιγμή υπάρχει ένας σταθερός αριθμός N ερωτήσεων που είναι φορτωμένες και εκτελούνται στον εξυπηρετητή. Από τη στιγμή κατά την οποία ολοκληρώνεται η επεξεργασία μιας ερώτησης μέχρι την αντικατάστασή της από μια άλλη μεσολαβεί χρόνος διαχείρισης με μέση τιμή* $3,1$ sec. *Ζητείται ο ρυθμός απόδοσης $X(N)$ του συστήματος για $N = 1, 2, 3$.*

Θα εφαρμόσουμε τον αλγόριθμο της Ανάλυσης Μέσης Τιμής, όπως περιγράφτηκε παραπάνω, για $M = 3$, $N = 3$ και $D_1 = 0,124$ sec, $D_2 = 0,230$ sec. Ο χρόνος διαχείρισης θα παρασταθεί ως χρόνος εξυπηρέτησης ενός σταθμού καθυστέρησης $D_3 = 3,1$ (όπως κατά κανόνα παριστάνουμε τον χρόνο σκέψης).

Η τιμή του N προδιαγράφει την εκτέλεση των επαναλήψεων για $k = 1, 2, 3$. Τα ενδιάμεσα αποτελέσματα για έναν δεδομένο πληθυσμό ($k < N$) παρέχουν αυτομάτως την τελική λύση για μικρότερους πληθυσμούς. Παραθέτουμε τις τιμές που υπολογίζονται σε κάθε βήμα (οι χρόνοι σε sec και ο ρυθμός απόδοσης σε εργασίες/sec).

Αρχικοποίηση: $Q_i = 0$, $i = 1, 2, 3$

k=1:	R_1=0,124		Q_1=0,0359
	R_2=0,230	$X(1)$=0,2895	Q_2=0,0666
	R_3=3,1		Q_3=0.8975
k=2:	R_1=0,1285		Q_1=0,074
	R_2=0,2453	$X(2)$=0,5757	Q_2=0,1413
	R_3=3,1		Q_3=1,7847
k=3:	R_1=0,1332		Q_1=0,1143
	R_2=0,2625	$X(3)$=0,8582	Q_2=0,2253
	R_3=3,1		Q_3=2.6604

□

4.3.2.2 Πολλές Κατηγορίες

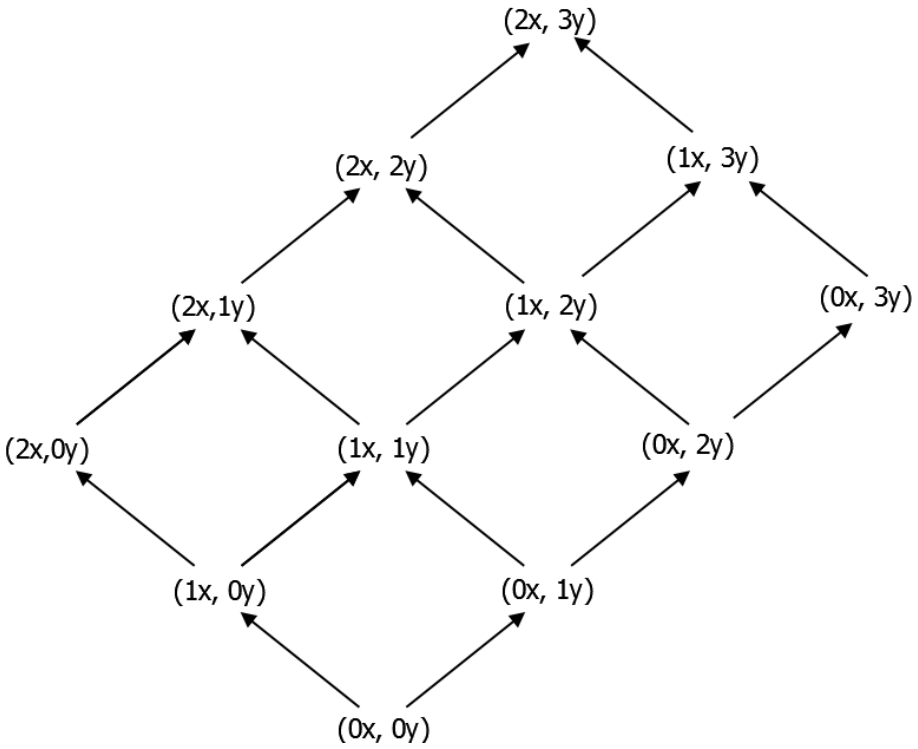
Η είσοδος εκφράζεται από το διάνυσμα $\boldsymbol{N} = [N_1, \ldots, N_C]$, όπου N_j ο αριθμός εργασιών της κατηγορίας j στο σύστημα. Οι βασικές σχέσεις και η ακριβής λύση προκύπτουν άμεσα ως γενίκευση του μοντέλου μιας κατηγορίας. Η διαδικασία περιγράφεται προγραμματιστικά από τον Αλγόριθμο 4.2.

$$R_{ij}(\boldsymbol{N}) = \begin{cases} D_{ij} & \text{(καθυστέρηση)} \\ D_{ij}\left[1 + \sum_{k=1}^{C} Q_{ik}(\boldsymbol{N} - \mathbf{1}_j)\right] & \text{(αναμονή)} \end{cases} \tag{4.49}$$

$$X^j(\boldsymbol{N}) = \frac{N_j}{\sum_{i=1}^{M} R_{ij}(\boldsymbol{N})} \tag{4.50}$$

$$Q_{ij}(\boldsymbol{N}) = X^j(\boldsymbol{N}) R_{ij}(\boldsymbol{N}) \tag{4.51}$$

Αλγόριθμος 4.2. *Ανάλυση Μέσης Τιμής (Πολλές κατηγορίες) — Ακριβής λύση*



Σχήμα 4.6: *MVA — Σχέσεις προήγησης.*

for $i \Leftarrow 1$ to M do
  for $j \Leftarrow 1$ to C do $Q_{ij}(\mathbf{0}) \Leftarrow 0$
for all $\boldsymbol{k} \leq \boldsymbol{N}$ do
  begin
  for $i \Leftarrow 1$ to M do
    for $j \Leftarrow 1$ to C do $R_{ij} \Leftarrow \begin{cases} D_{ij} & \text{(καθυστέρηση)} \\ D_{ij}[1 + \sum_{l=1}^{C} Q_{il}(\boldsymbol{k} - \mathbf{1}_j)] & \text{(αναμονή)} \end{cases}$ /* $k_j > 0$ */
  for $j \Leftarrow 1$ to C do $X^j \Leftarrow \dfrac{k_j}{\sum_{i=1}^{M} R_{ij}}$
  for $i \Leftarrow 1$ to M do
    for $j \Leftarrow 1$ to C do $Q_{ij}(\boldsymbol{k}) \Leftarrow \lambda^j R_{ij}$
  end

Η αναδρομική εξάρτηση από τις τιμές $Q_{il}(\boldsymbol{k} - \mathbf{1}_j)$, $1 \leq j, l \leq C$ επιβαρύνει σημαντικά τις απαιτήσεις του αλγορίθμου σε χώρο. Το παράδειγμα του Σχήματος 4.6 παριστάνει τις εξαρτήσεις προήγησης των ενδιάμεσων πληθυσμών για την περίπτωση ενός μικρού δικτύου (2 κατηγορίες x και y, με 2 και 3 πελάτες, αντίστοιχα). Μια λύση είναι να αποθηκεύουμε τις συγκεντρωτικές τιμές $Q_i(\boldsymbol{k}) = \sum_{j=1}^{C} Q_{ij}(\boldsymbol{k})$, δεδομένου ότι αυτές αρκούν για την ορθή εκτέλεση του αλγορίθμου (εκτός αν υπάρχει λόγος αποθήκευσης όλων των ενδιάμεσων τιμών). Σε αυτή την ιδέα βασίζεται μια αποδοτική παραλλαγή του αλγορίθμου η οποία περιγράφεται προγραμματιστικά παρακάτω (Αλγόριθμος 4.3).

Αλγόριθμος 4.3. *Ανάλυση Μέσης Τιμής (Πολλές κατηγορίες) — Ακριβής λύση, Οικονομική χρήση μνήμης*

for $i \Leftarrow 1$ to M do $Q_i(\mathbf{0}) \Leftarrow 0$
for all $\boldsymbol{k} \leq \boldsymbol{N}$ do
begin
for $i \Leftarrow 1$ to M do
for $j \Leftarrow 1$ to C do $R_{ij} \Leftarrow \begin{cases} D_{ij} & \text{(καθυστέρηση)} \\ D_{ij}[1 + Q_i(\boldsymbol{k} - \mathbf{1}_j) & \text{(αναμονή)} \end{cases}$ /* $k_j > 0$ */
for $j \Leftarrow 1$ to C do $X^j \Leftarrow \dfrac{k_j}{\sum_{i=1}^{M} R_{ij}}$
for $i \Leftarrow 1$ to M do
$Q_i(\boldsymbol{k}) \Leftarrow \sum_{j=1}^{C} X^j R_{ij}$
end

Οι απαιτήσεις του αλγορίθμου είναι $O(CM \prod_{j=1}^{C} N_j)$ σε χρόνο και $O(M \prod_{j=1}^{C} N_j)$ σε χώρο. Αυτό σημαίνει ότι —παρά την οικονομική χρήση της μνήμης— είναι πρακτικά ασύμφορη η ακριβής επίλυση δικτύων με πολλές κατηγορίες. Για παράδειγμα, η επίλυση ενός δικτύου με 10 σταθμούς, 5 κατηγορίες και 10 πελάτες ανά κατηγορία απαιτεί πάνω από 5 000 000 αριθμητικές πράξεις και 1 000 000 θέσεις μνήμης. (Αντίθετα, ένα δίκτυο μιας κατηγορίας με 10 σταθμούς και 50 πελάτες απαιτεί περίπου 500 πράξεις και 10 θέσεις μνήμης.)

4.3.3 Μικτά Δίκτυα

Τα μικτά δίκτυα περιλαμβάνουν ανοικτές και κλειστές κατηγορίες. Συμβολίζουμε την ένταση φορτίου με το διάνυσμα $\boldsymbol{I} = [N_1 \text{ ή } \lambda^1, \ldots, N_C \text{ ή } \lambda^C]$, όπου ανάλογα με το είδος της κατηγορίας υπάρχει ο αριθμός εργασιών ή ο ρυθμός αφίξεων. Επίσης, συμβολίζουμε με O το σύνολο των ανοικτών κατηγοριών και με C το σύνολο των κλειστών κατηγοριών. Θα εφαρμόσουμε το θεώρημα των αφίξεων για ανοικτές και κλειστές κατηγορίες [23].

Ανοικτές Κατηγορίες ($j \in O$) Εφαρμογή του θεωρήματος των αφίξεων για σταθμούς με αναμονή:

$$R_{ij}(\boldsymbol{I}) = D_{ij}\left[1 + \sum_{k \in C} Q_{ik}(\boldsymbol{I}) + \sum_{k \in O} Q_{ik}(\boldsymbol{I})\right] = D_{ij}\left[1 + \sum_{k \in C} Q_{ik}(\boldsymbol{I}) + \sum_{k \in O} X^k R_{ik}(\boldsymbol{I})\right]$$

Όπως και στα ανοικτά δίκτυα έχουμε:

$$\frac{R_{ij}(\boldsymbol{I})}{R_{ik}(\boldsymbol{I})} = \frac{D_{ij}}{D_{ik}}$$

ή

$$R_{ik}(\boldsymbol{I}) = \frac{D_{ik}}{D_{ij}} R_{ij}(\boldsymbol{I}) \quad j, k \in O$$

οπότε αντικαθιστώντας προκύπτει:

$$R_{ij}(\boldsymbol{I}) = \frac{D_{ij}[1 + \sum_{k \in C} Q_{ik}(\boldsymbol{I})]}{1 - \sum_{k \in O} U_{ik}}, \quad j \in O \tag{4.52}$$

Κλειστές Κατηγορίες ($j \in C$) Εφαρμογή του θεωρήματος των αφίξεων για σταθμούς με αναμονή:

$$R_{ij}(\boldsymbol{I}) = D_{ij}\left[1 + \sum_{k \in C} Q_{ik}(\boldsymbol{I} - \mathbf{1}_j) + \sum_{k \in O} Q_{ik}(\boldsymbol{I} - \mathbf{1}_j)\right]$$

Αλλά, βάσει της (4.52) ισχύει:

$$\sum_{k \in O} Q_{ik}(\boldsymbol{I} - \mathbf{1}_j) = \sum_{k \in O} X^k D_{ik} \frac{1 + \sum_{k \in C} Q_{ik}(\boldsymbol{I} - \mathbf{1}_j)}{1 - \sum_{k \in O} U_{ik}} = \frac{1 + \sum_{k \in C} Q_{ik}(\boldsymbol{I} - \mathbf{1}_j)}{1 - \sum_{k \in O} U_{ik}} \sum_{k \in O} U_{ik}$$

οπότε

$$R_{ij}(\boldsymbol{I}) = D_{ij}\left[1 + \sum_{k\in C} Q_{ik}(\boldsymbol{I} - \mathbf{1}_j) + \frac{[1 + \sum_{k\in C} Q_{ik}(\boldsymbol{I} - \mathbf{1}_j)]\sum_{k\in O} U_{ik}}{1 - \sum_{k\in O} U_{ik}}\right]$$

ή τελικά:

$$R_{ij}(\boldsymbol{I}) = \frac{D_{ij}[1 + \sum_{k\in C} Q_{ik}(\boldsymbol{I} - \mathbf{1}_j)]}{1 - \sum_{k\in O} U_{ik}}, \quad j \in C \tag{4.53}$$

Το αποτέλεσμα (4.53) είναι ιδιαίτερα ενδιαφέρον διότι δείχνει (σε σύγκριση με την Εξίσωση (4.49)) ότι οι δείκτες επίδοσης των κλειστών κατηγοριών στο μικτό μοντέλο είναι οι ίδιοι που θα παίρναμε αν επιλύαμε το κλειστό δίκτυο που προκύπτει αν εξαλείψουμε τις ανοικτές κατηγορίες και «διαστείλουμε» τις απαιτήσεις εξυπηρέτησης των κλειστών κατηγοριών κατά τον παράγοντα $1 - \sum_{k\in O} U_{ik}$ σε κάθε σταθμό i. Στην ουσία, ο παράγων αυτός υποδηλώνει την ταχύτητα εξυπηρέτησης του σταθμού, όπως τη βλέπουν οι κλειστές κατηγορίες, δεδομένου ότι μέρος του χρόνου του αφιερώνεται στις ανοικτές κατηγορίες. Η τεχνική αυτή η οποία ουσιαστικά συνίσταται σε «διαστολή» του χρόνου εξυπηρέτησης αναφέρεται συχνά ως *απόκρυψη φορτίου* (load concealment) και χρησιμοποιείται ευρύτατα για τη μείωση της πολυπλοκότητας των μοντέλων. Ένα ανάλογο φαινόμενο διαστολής μπορεί να επισημανθεί και στο αποτέλεσμα (4.52).

Με βάση τα παραπάνω μπορούμε να διατυπώσουμε έναν αλγόριθμο επίλυσης μικτών μοντέλων. Τα βήματα της διαδικασίας περιγράφονται αδρομερώς παρακάτω (Αλγόριθμος 4.4).

Αλγόριθμος 4.4. *Επίλυση μικτών δικτύων*

(i) Για κάθε σταθμό i υπολογίζουμε τον βαθμό χρησιμοποίησης για κάθε ανοικτή κατηγορία

$$U_{ij}(\boldsymbol{I}) = \lambda^j D_{ij}, \quad j \in O$$

(ii) Επιλύουμε το κλειστό μοντέλο που αποτελείται μόνο από τις κλειστές κατηγορίες διαστέλλοντας τους χρόνους εξυπηρέτησης όπως γίνεται στην (4.53). Για κάθε σταθμό, ο ρυθμός απόδοσης, ο χρόνος παραμονής και ο μέσος αριθμός εργασιών που υπολογίζονται για τις κλειστές κατηγορίες είναι και τα αποτελέσματα για το μικτό μοντέλο. Οι βαθμοί χρησιμοποίησης υπολογίζονται θεωρώντας τους αρχικούς χρόνους εξυπηρέτησης (χωρίς διαστολή).

(iii) Για τις ανοικτές κατηγορίες και για κάθε σταθμό υπολογίζουμε το χρόνο παραμονής με βάση την (4.52) και το μέσο αριθμό εργασιών.

4.4 Σταθμοί με Ρυθμό Εξυπηρέτησης Εξαρτώμενο από το Φορτίο

Τα μοντέλα που εξετάσαμε ως τώρα βασίζονται στην υπόθεση ότι οι ρυθμοί εξυπηρέτησης των σταθμών είναι σταθεροί, δηλαδή ανεξάρτητοι από τον αριθμό εργασιών που βρίσκονται στην ουρά του σταθμού. Οι σταθμοί αυτοί αναφέρονται ως *ανεξάρτητοι του φορτίου* (Load Independent, LI), σε αντιδιαστολή προς τους σταθμούς *με εξάρτηση από το φορτίο* (Load Dependent, LD), στους οποίους ο ρυθμός εξυπηρέτησης μεταβάλλεται συναρτήσει του αριθμού εργασιών. Υπάρχουν πολλά παραδείγματα συσκευών με μεταβλητούς ρυθμούς εξυπηρέτησης, όπως δίκτυα ευρείας ζώνης με μηχανισμούς ελέγχου συμφόρησης (congestion control), τοπικά δίκτυα, εξυπηρετητές δίσκων, πολυεπεξεργαστές κλπ. Γενικά, τα μοντέλα LD εκφράζουν με επιτυχία τη δυναμική συμπεριφορά διαφόρων συνιστωσών των υπολογιστκών συστημάτων.

Θα εξετάσουμε καταρχάς τις βασικές τροποποιήσεις που πρέπει να γίνουν στον (ακριβή) αλγόριθμο MVA για να καλύπτεται και η περίπτωση των σταθμών με ρυθμό εξυπηρέτησης εξαρτώμενο από το φορτίο, περιορίζοντας έτσι την προσοχή μας σε κλειστά δίκτυα. Στο επόμενο κεφάλαιο θα δούμε την προσεγγιστική διατύπωση του αλγορίθμου για κλειστά δίκτυα που περιέχουν σταθμούς LI και LD, καθώς και την αντίστοιχη επέκταση της διαδικασίας επίλυσης ανοικτών δικτύων. Η εφαρμογή της ανάλυσης μέσης τιμής περιλαμβάνει τυπικά 3 στάδια σε κάθε βήμα του αλγορίθμου:

- υπολογισμός του χρόνου παραμονής,
- υπολογισμός του ρυθμού απόδοσης,
- υπολογισμός του μέσου αριθμού πελατών.

Οι αναγκαίες τροποποιήσεις, προκειμένου να περιληφθεί και η περίπτωση σταθμών LD, αφορούν το πρώτο και το τρίτο στάδιο του αλγορίθμου. Οι τεχνικές που προκύπτουν παρέχουν ακριβή λύση, στην πράξη όμως χρησιμοποιούνται σε προσεγγίσεις και κυρίως για την υλοποίηση της ιεραρχικής μοντελοποίησης (ισοδύναμοι σταθμοί), που θα μας απασχολήσει στο επόμενο κεφάλαιο.

4.4.1 Μία Κατηγορία

Στη διατύπωση που θα χρησιμοποιήσουμε, ο χρόνος εξυπηρέτησης θα αντικατασταθεί από τον ρυθμό εξυπηρέτησης, ο οποίος πρακτικά είναι ίσος με το αντίστροφο του χρόνου εξυπηρέτησης ανά επίσκεψη. Συμβολίζουμε με $\mu_i(k)$ τον ρυθμό εξυπηρέτησης του σταθμού i, όταν υπάρχουν k πελάτες στον σταθμό. Επίσης, συμβολίζουμε με $p_i(k|N)$ την πιθανότητα να υπάρχουν k πελάτες στον σταθμό i, όταν ο συνολικός αριθμός πελατών στο δίκτυο είναι N.

Οι ακόλουθες σχέσεις θα πρέπει να χρησιμοποιηθούν στο πρώτο και τρίτο στάδιο του αλγορίθμου MVA για σταθμούς με ρυθμό εξαρτώμενο από το φορτίο [19]. (Για σταθμούς με σταθερό ρυθμό ισχύουν οι αρχικές σχέσεις.)

- Υπολογισμός του χρόνου παραμονής:

$$R_i(N) = v_i \sum_{k=1}^{N} \frac{k}{\mu_i(k)} p_i(k-1|N-1) \tag{4.54}$$

- Υπολογισμός της κατανομής του αριθμού πελατών στον σταθμό (αντί του μέσου αριθμού που χρησιμοποιείται στην περίπτωση σταθερών ρυθμών):

$$p_i(k|N) = \begin{cases} \dfrac{X_i(N)}{\mu_i(k)} p_i(k-1|N-1) & k = 1, \ldots, N \\ 1 - \sum_{l=1}^{N} p_i(l|N) & k = 0 \end{cases} \tag{4.55}$$

(Προφανώς, ισχύει $p_i(0|0) = 1$ για όλα τα i.)

4.4.2 Πολλές Κατηγορίες

Για την περίπτωση των πολλών κατηγοριών θα θεωρήσουμε ότι ο ρυθμός εξυπηρέτησης εξαρτάται από τον συνολικό αριθμό πελατών στον σταθμό (και όχι από τον αριθμό πελατών κάθε κατηγορίας) και ότι η εξάρτηση είναι ίδια για όλες τις κατηγορίες. Ουσιαστικά, ο ρυθμός εξυπηρέτησης (που ορίστηκε ως αντίστροφο του μέσου χρόνου εξυπηρέτησης ανά επίσκεψη) χαρακτηρίζεται από μία σταθερή ποσότητα εξυπηρέτησης και την ταχύτητα εξυπηρέτησης του σταθμού, η οποία μεταβάλλεται ως συνάρτηση του συνολικού αριθμού πελατών. Για παράδειγμα, έστω ότι ο ρυθμός εξυπηρέτησης της κατηγορίας A σε ένα σταθμό είναι 2 φορές μεγαλύτερος όταν υπάρχουν 5 πελάτες (οποιασδήποτε κατηγορίας) στο σταθμό απ' ό,τι όταν υπάρχουν 2 πελάτες. Τότε και για την κατηγορία B θα πρέπει να ισχύει το ίδιο. Αυτή η μορφή εξάρτησης είναι η απλούστερη δυνατή για την περίπτωση πολλών κατηγοριών. Στη γενικότερη περίπτωση, ο ρυθμός εξυπηρέτησης μιας κατηγορίας μπορεί να εξαρτάται από την πλήρη κατάσταση του σταθμού (αριθμό πελατών για κάθε κατηγορία). Η λύση όμως στην περίπτωση αυτή υπόκειται σε ορισμένους περιορισμούς και είναι αρκετά πολύπλοκης μορφής, οπότε δεν θα μας απασχολήσει. Επιπλέον, θα πρέπει να σημειωθεί ότι στην πράξη η προσεγγιστική επίλυση δικτύων ανάγεται συνήθως σε επίλυση μοντέλων μιας κατηγορίας με μη σταθερούς ρυθμούς, αποφεύγοντας τα μοντέλα πολλών κατηγοριών που έχουν μεγάλες απαιτήσεις σε χρόνο και χώρο.

Για την απλή μορφή εξάρτησης που περιγράψαμε, οι σχέσεις που προκύπτουν είναι απλή γενίκευση του μοντέλου μιας κατηγορίας. Συμβολίζουμε με $\mu_{ij}(k)$ τον ρυθμό εξυπηρέτησης της κατηγορίας j στον σταθμό i, όταν υπάρχουν συνολικά k πελάτες στον σταθμό και με $p_i(k|\boldsymbol{N})$ την πιθανότητα να υπάρχουν k πελάτες στο σταθμό i (ανεξάρτητα από κατηγορία) όταν ο πληθυσμός του δικτύου είναι $\boldsymbol{N}$.

- Υπολογισμός του χρόνου παραμονής:

$$R_{ij}(\boldsymbol{N}) = v_{ij} \sum_{k=1}^{N} \frac{k}{\mu_{ij}(k)} p_i(k-1|\boldsymbol{N}-\mathbf{1}_j) \tag{4.56}$$

  όπου $N = \|\boldsymbol{N}\|$.

- Υπολογισμός της κατανομής του συνολικού αριθμού πελατών στον σταθμό:

$$p_i(k|\boldsymbol{N}) = \begin{cases} \sum_{j=1}^{C} \frac{X_{ij}(\boldsymbol{N})}{\mu_{ij}(k)} p_i(k-1|\boldsymbol{N}-\mathbf{1}_j) & k = 1,\ldots,N \\ 1 - \sum_{l=1}^{N} p_i(l|\boldsymbol{N}) & k = 0 \end{cases} \tag{4.57}$$

4.5 Μοντελοποίηση του Ιστού

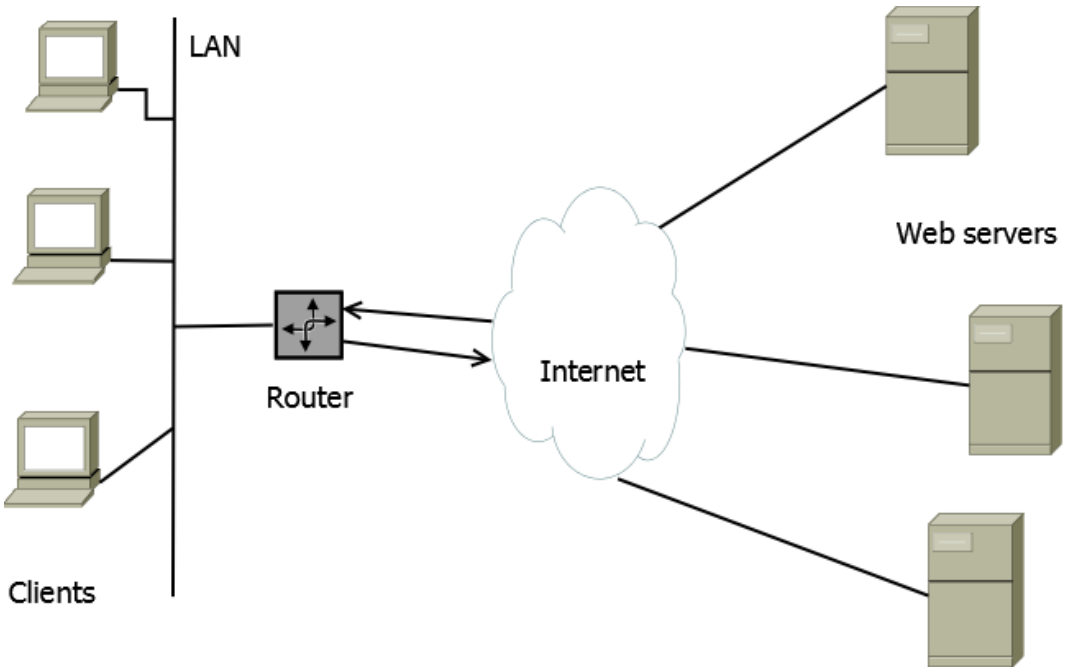
Τα μοντέλα επίδοσης που εξετάστηκαν μέχρι τώρα μπορούν να εφαρμοστούν σε πληθώρα υπολογιστικών συστημάτων. Οι *υπηρεσίες ιστού* (Web services) αποτελούν εφαρμογές με ιδιαίτερα χαρακτηριστικά, τα οποία λαμβάνονται υπόψη κατά την ανάπτυξη εξειδικευμένων μοντέλων [16, 17].

Τα μοντέλα που χρησιμοποιούνται για την ανάλυση συστημάτων στον Παγκόσμιο Ιστό αποτελουν κατά κύριο λόγο γενικεύσεις του μοντέλου *πελάτη-εξυπηρετητή* (client-server). Η διαφορά είναι ότι —στη γενική περίπτωση— δεν είναι δυνατόν να συνυπάρχουν ισότιμα πελάτες και εξυπηρετητές. Πράγματι, ένας πελάτης μπορεί να επικοινωνεί με πολλούς εξυπηρετητές σε παγκόσμια κλίμακα, χωρίς, όμως να γνωρίζει τις λεπτομέρειες που απαιτούνται για την ενσωμάτωσή τους στο μοντέλο. Αντίστοιχα, ένας εξυπηρετητής δέχεται αιτήματα από μεγάλο αριθμό πελατών, σε μια ενιαία ροή εισόδου. Η ύπαρξη αυτής της διπλής ασυμμετρίας, οδηγεί στην ανάπτυξη χωριστών μοντέλων για την πλευρά των πελατών και την πλευρά των εξυπηρετητών, αντίστοιχα. Μια περίπτωση που αποτελεί εξαίρεση στην παραπάνω θεώρηση αφορά τα μοντέλα *ιδιωτικών (εταιρικών) δικτύων* (intranet), δηλαδή μεγάλων δικτύων με πεπερασμένο αριθμό πελατών και εξυπηρετητών, που χρησιμοποιούν τεχνολογίες Διαδικτύου και Παγκόσμιου Ιστού. Στη συνέχεια, θα εξετάσουμε βασικά μοντέλα των δύο πλευρών.

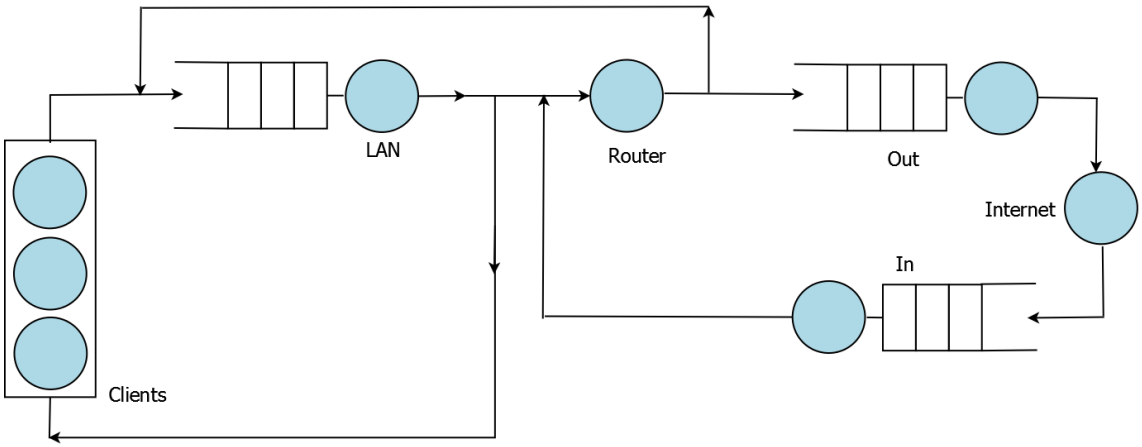
4.5.1 Μοντέλα από την Πλευρά των Πελατών

Το Σχήμα 4.7 παριστάνει τη δομή ενός συστήματος από την οπτική γωνία του πελάτη. Η αρχιτεκτονική περιλαμβάνει ένα σύνολο σταθμών εργασίας (πελατών) στους οποίους εκτελούνται φυλλομετρητές Ιστού (Web browsers). Οι σταθμοί εργασίας συνδέονται σε ένα τοπικο δίκτυο (LAN), το οποίο συνδέεται στο Διαδίκτυο μέσω ενός δρομολογητή (router) και ενός παρόχου υπηρεσιών Διαδικτύου (εισερχόμενη και εξερχόμενη σύνδεση). Το Σχήμα 4.8 απεικονίζει ένα μοντέλο κλειστού δικτύου αναμονής που αντιστοιχεί στην αρχιτεκτονική του Σχήματος 4.7. Οι σταθμοί εργασίας (clients) λειτουργούν ως σταθμός καθυστέρησης, στον οποίο ο χρόνος που δαπανάται είναι ο χρόνος σκέψης των χρηστών. Επίσης, σταθμοί με σχετικά μικρό χρόνο εξυπηρέτησης (όπως ο δρομολογητής) μπορούν να παρασταθούν ως σταθμοί καθυστέρησης. Τέλος, το Διαδίκτυο παριστάνεται ως σταθμός καθυστέρησης και περιλαμβάνει τη σύνδεση σε κάποιον πάροχο υπηρεσιών Διαδικτύου, τη διάσχιση του Διαδικτύου και την εξυπηρέτηση σε καποιον απομακρυσμένο εξυπηρετητή. Οι υπόλοιποι σταθμοί (τοπικό δίκτυο, εισερχόμενος και εξερχόμενος σύνδεσμος) είναι σταθμοί αναμονής με ρυθμούς αναξάρτητους από το φορτίο.

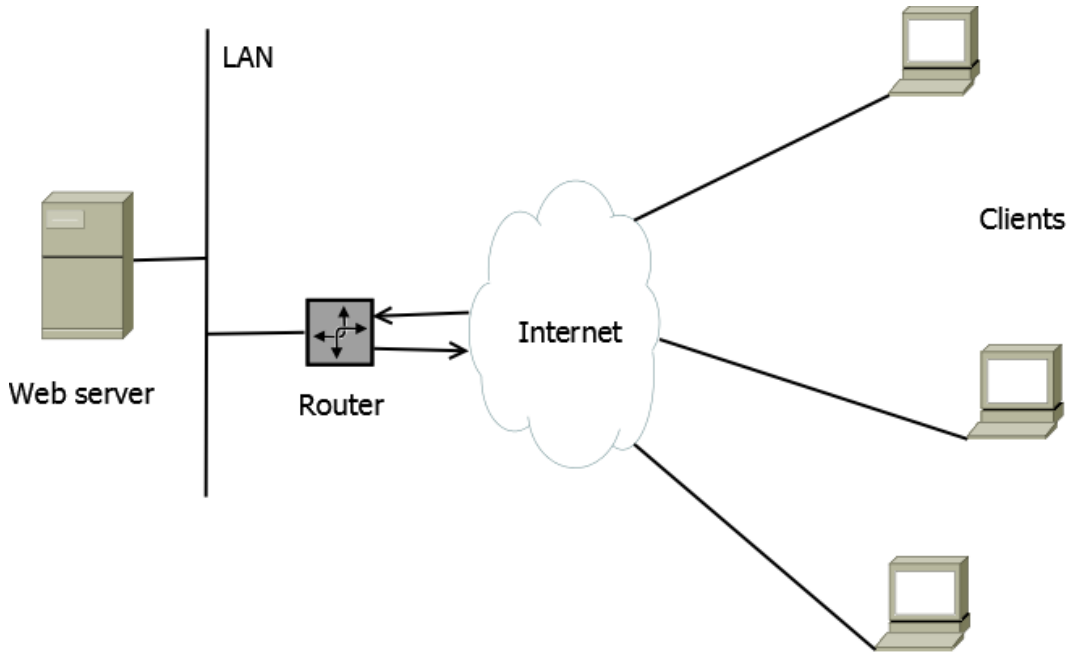
Η επίλυση του μοντέλου προϋποθέτει ότι έχουν προσδιοριστεί οι πραγματικές τιμές των παραμέτρων και των χρόνων εξυπηρέτησης. Οι ποσότητες αυτές συνήθως υπολογίζονται από μετρήσεις και από τα τεχνικά χαρακτηριστικά των συνιστωσών του συστήματος, με βάση τη λεπτομερή αναπαράσταση των διαφόρων λειτουργιών κατά περίπτωση [16]. Αν και ο υπολογισμός είναι προσεγγιστικός, η επίλυση του μοντέλου παρέχει τελικά αποτελέσματα με πολύ ικανοποιητική ακρίβεια.



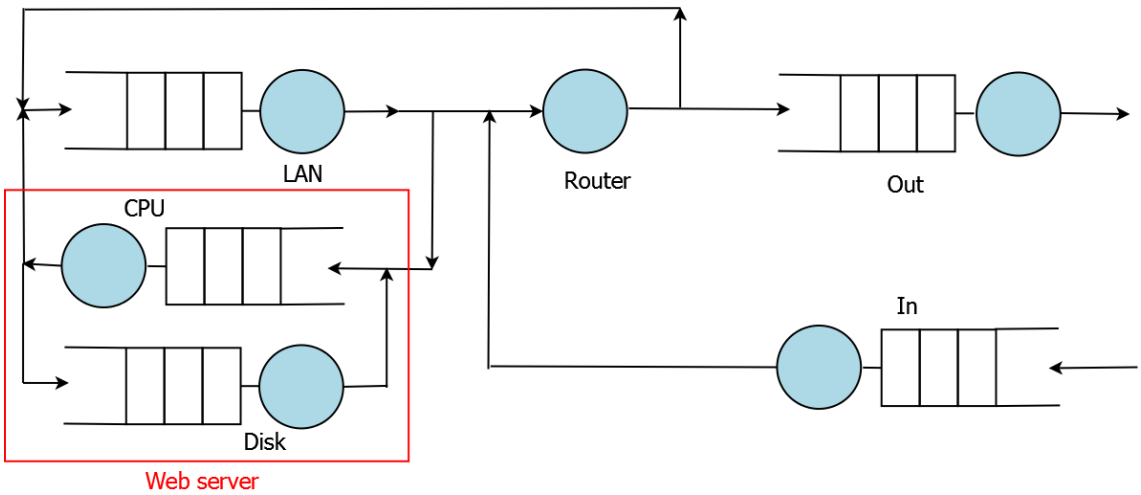
Σχήμα 4.7: *Από την πλευρά των πελατών.*



Σχήμα 4.8: *Μοντέλο κλειστού δικτύου (πλευρά πελατών).*



Σχήμα 4.9: *Από την πλευρά του εξυπηρετητή.*



Σχήμα 4.10: *Μοντέλο ανοικτού δικτύου (πλευρά εξυπηρετητή).*

4.5.2 Μοντέλα από την Πλευρά του Εξυπηρετητή

Όπως αναφέρθηκε ήδη, το μοντέλο είναι διαφορετικό από την οπτική γωνία του εξυπηρετητή. Το Σχήμα 4.9 παριστάνει τη δομή ενός τυπικού περιβάλλοντος με έναν εξυπηρετητή Ιστού, ο οποίος είναι συνδεδεμένος σε ένα τοπικό δίκτυο. Το δίκτυο συνδέεται σε έναν δρομολογητή, ο οποίος με τη σειρά του συνδέεται με εναν πάροχο υπηρεσιών Διαδικτύου. Τα αρχεία που διαχειρίζεται ο εξυπηρετητής είναι αποθηκευμένα στον ίδιο χώρο με αυτόν. Το αντίστοιχο δίκτυο αναμονής φαίνεται στο Σχήμα 4.10. Αν υποθέσουμε ότι ο εξυπηρετητής είναι δημόσια διαθέσιμος στο Διαδίκτυο, τότε θα υπάρχει ένας πολύ μεγάλος πληθυσμός άγνωστων πελατών που θα μπορούν να τον προσπελάσουν. Συνεπώς, το σύστημα μπορεί να παρασταθεί ως ανοικτό δίκτυο αναμονής πολλών κατηγοριών, οι οποίες θα αντιστοιχούν σε αιτήματα (http) για διάφορα μεγέθη αρχείων. Όπως και στα μοντέλα που αναφέρθηκαν στην προηγούμενη παράγραφο, το τοπικό δίκτυο, ο εισερχόμενος και ο εξερχόμενος σύνδεσμος θα παρασταθούν ως σταθμοί αναμονής με ρυθμό ανεξάρτητο από το φορτίο, ενώ ο δρομολογητής ως σταθμός καθυστέρησης. Ο εξυπηρετητής περιλαμβάνει ΚΜΕ και έναν δίσκο, σταθμούς αναμονής (θα μπορούσαμε να έχουμε πολλούς δίσκους και πολλούς επεξεργαστές). Και στην περίπτωση αυτή θα πρέπει να γίνει λεπτομερής προσδιορισμός των παραμέτρων του δικτύου [16].

Βιβλιογραφία

[1] Baskett, F., Chandy, K.M., Muntz, R.R. and Palacios, F.G., *Open, Closed, and Mixed Networks of Queues with Different Classes of Customers*, Journal of the ACM, Vol. 22, No. 2, pp. 248–260, April 1975.

[2] Bolch, G., Greiner, S., De Meer, H., and Trivedi, K.S., *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, Wiley-Interscience, 2006.

[3] Burke, P.J., *The Output of a Queuing System*, Operations Research, Vol. 4, No. 6, pp. 699-704, Dec. 1956.

[4] Buzen, J.P., *Computational Algorithms for Closed Queueing Networks with Exponential Servers*, Communications of the ACM, Vol. 16, No. 9, pp. 527–531, Sept. 1973.

[5] Buzen, J.P., *Fundamental Operational Laws of Computer System Performance*, Acta Informatica, Vol. 7, 1976.

[6] Chandy, K., Howard, J., and Towsley, D., *Product Form and Local Balance in Queueing Networks*, Journal of the ACM, Vol. 24, No. 2, pp. 250-263, April 1977.

[7] Denning, P.J. and Buzen, J.P., *The Operational Analysis of Queueing Network Models*, Computing Surveys, Vol. 10, No. 3, pp. 225-261, 1978.

[8] Gelenbe, E. and Muntz, R.R., *Probabilistic Models of Computer Systems –Part I (Exact Results)*, Acta Informatica, Vol. 7, pp. 35–60, 1976.

[9] Gelenbe, E. and Mitrani, I., *Analysis and Synthesis of Computer Systems*, Academic Press, 1980.

[10] Gelenbe, E. and Pujolle, G., *Introduction to Queueing Networks*, John Wiley, 1987.

[11] Gordon, W.J. and Newell, G.F., *Closed Queueing Systems with Exponential Servers*, Operations Research, Vol. 15, No. 2, pp. 254–265, 1967.

[12] Harchol-Balter, M., *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, Cambridge University Press, 2013.

[13] Jackson, J.R., *Networks of Waiting Lines*, Operations Research, 5, 1957.

[14] Jackson, J.R., *Jobshop-like Queueing Systems*, Management Science, Vol. 10, No. 1, pp. 131–142, 1963.

[15] Little, J.D.C., *A Proof of the Queueing Formula* $L = \lambda W$, Operations Research, Vol. 9, pp. 383–387, 1961.

[16] Menasce, D.A., and Almeida, V.A.F., *Capacity Planning for Web Services: Metrics, Models and Methods*, Prentice-Hall, 2002.

[17] Menasce, D.A., Almeida, V.A.F., and Dowdy, L.W., *Performance by Design, Computer Capacity Planning by Example*, Prentice-Hall PTR, 2004.

[18] Muntz, R., *Poisson Departure Process and Queueing Networks*, Proc. 7th Annual Princeton Conf. on Information Sciences and Systems, pp. 435-440, Princeton University Press, Princeton, March 1973.

[19] Reiser, M. and Lavenberg, S.S., *Mean-Value Analysis of Closed Multichain Queueing Networks*, Journal of the ACM, Vol. 27, No. 2, pp. 313–322, April 1980.

[20] Sauer, C.H. and Chandy, K.M., *Computer Systems Performance Modelling*, Prentice-Hall, 1981.

[21] Sevcik, K.C. and Mitrani, I., *The Distribution of Queueing Network States at Input and Output Instants*, Journal of the ACM, Vol. 28, No. 2, April 1981.

[22] Walrand, J., *An Introduction to Queueing Networks*, Prentice-Hall, 1988.

[23] Zahorjan, J.and Wong, E., *The Solution of Separable Queueing Network Models Using Mean Value Analysis*, ACM SIGMETRICS Conference on Measurement and Modelling of Computer Systems, Las Vegas, Nevada, Sep. 1981.

Κεφάλαιο 5

Προσεγγιστικές Τεχνικές

Σύνοψη
Αναπτύσσονται προσεγγιστικές τεχνικές βασισμένες στη θεωρία των δικτύων αναμονής. Περιλαμβάνονται η προσεγγιστική διατύπωση της μεθόδου MVA (μοντέλα μιας κατηγορίας και πολλών κατηγοριών) για σταθερούς και μεταβλητούς ρυθμούς εξυπηρέτησης, η ιεραρχική μοντελοποίηση με βάση την ισοδυναμία της ροής (ορισμός εξυπηρετητών ισοδύναμων ως προς τη ροή), φράγματα (ασυμπτωτικά φράγματα και φράγματα ισορροπημένων συστημάτων), ανάλυση στένωσης. Περιγράφονται παραδείγματα προσεγγιστικών μεθόδων για την ανάλυση συστημάτων που δεν επιδέχονται λύση μορφής γινομένου (non product-form), όπως συστήματα με περιορισμούς μνήμης/πληθυσμού, συστήματα με αποκλεισμό (blocking), συστήματα με προτεραιότητες (μέθοδος SWIC), συστήματα Fork–Join κλπ. Γίνεται αναφορά στην ανάπτυξη μοντέλων για συστήματα βασισμένα στον Ιστό, με έμφαση σε ιδιαίτερα χαρακτηριστικά, όπως η εκρηκτικότητα (burstiness) του φορτίου, η ύπαρξη τεχνικών proxing, caching, mirroring κλπ.

Στο κεφάλαιο αυτό θα παρουσιάσουμε ορισμένες τεχνικές ανάλυσης υπολογιστικών συστημάτων βασισμένες στα αποτελέσματα της θεωρίας των δικτύων αναμονής. Οι τεχνικές αυτές διατυπώνονται σε μορφή που προσφέρεται για πρακτική εφαρμογή και είναι κατά κύριο λόγο προσεγγιστικές [4, 8, 2].

5.1 Προσέγγιση MVA για Σταθμούς με Σταθερό Ρυθμό

Ο αλγόριθμος MVA χαρακτηρίζεται από την αναδρομική σχέση σύμφωνα με την οποία οι τιμές των μεταβλητών στην τρέχουσα κατάσταση του δικτύου εξαρτώνται από τις τιμές που αφορούν το δίκτυο με έναν πελάτη λιγότερο. Ο στόχος μιας προσεγγιστικής αντιμετώπισης θα ήταν η απαλλαγή από την αναδρομική εξάρτηση και η αντικατάστασή της με μια προσέγγιση που θα περιοριζόταν στην τρέχουσα κατάσταση πληθυσμού. Στο πνεύμα αυτό έχουν προταθεί διάφορες εκφράσεις για την προσεγγιστική συνάρτηση, οι οποίες απλουστεύουν και επιταχύνουν τον αλγόριθμο. Στη συνέχεια θα περιγράψουμε μια αποδοτική μέθοδο που χρησιμοποιείται ευρύτατα στις εφαρμογές. Η προσέγγιση αυτή οφείλεται στους Bard και Schweitzer [1, 11].

5.1.1 Μία Κατηγορία

Σύμφωνα με την προσέγγιση των Bard και Schweitzer καταργούμε την εξάρτηση από το $Q_i(N-1)$ εισάγοντας στη θέση του μία συνάρτηση του $Q_i(N)$. Στην περίπτωση αυτή καταλήγουμε σε ένα μη γραμμικό σύστημα ως προς τις ποσότητες με όρισμα N, το οποίο μπορεί να λυθεί επαναληπτικά μέχρι να επιτευχθεί η επιθυμητή ακρίβεια. Η μέθοδος βασίζεται στην υπόθεση ότι, καθώς αυξάνει ο συνολικός αριθμός εργασιών στο δίκτυο, αυξάνει ανάλογα και το μήκος των ουρών στους σταθμούς:

$$Q_i(N-1) \simeq \frac{N-1}{N} Q_i(N) \tag{5.1}$$

Η χρήση της προσέγγισης οδηγεί στην παρακάτω τροποποίηση του βασικού αλγορίθμου (Αλγόριθμος 5.1):

Αλγόριθμος 5.1. *Αλγόριθμος Μέσης Τιμής (Μία κατηγορία) — Προσέγγιση Bard–Schweitzer*

(i) Αρχικές τιμές: $Q_i(N) \Leftarrow \dfrac{N}{M}$ για όλα τα i.

(ii) Εφαρμογή των Εξισώσεων (4.46)–(4.48) με χρήση της προσέγγισης (5.1) στην (4.46) και υπολογισμός των νέων $Q_i(N)$.

(iii) Αν οι τιμές $Q_i(N)$ που προκύπτουν από το βήμα (ii) διαφέρουν από τις προηγούμενες περισσότερο από ένα δεδομένο ποσοστό, επιστροφή στο βήμα (ii) με τις νέες τιμές.

Οι απαιτήσεις του αλγορίθμου είναι $O(M)$ σε χρόνο και χώρο ανά επανάληψη, ενώ ο αριθμός των απαιτούμενων επαναλήψεων είναι γενικά μικρός.

5.1.2 Πολλές Κατηγορίες

Στην περίπτωση των πολλών κατηγοριών, η οποία όπως είδαμε χαρακτηρίζεται από υψηλό υπολογιστικό κόστος, είναι ιδιαίτερα χρήσιμη η προσεγγιστική τεχνική των Bard και Schweitzer, η οποία σ΄ αυτή την περίπτωση στηρίζεται στην προσέγγιση:

$$\sum_{k=1}^{C} Q_{ik}(\boldsymbol{N} - \mathbf{1}_j) \simeq \frac{N^j - 1}{N^j} Q_{ij}(\boldsymbol{N}) + \sum_{\substack{k=1 \\ k \neq j}}^{C} Q_{ik}(\boldsymbol{N}) \tag{5.2}$$

και οδηγεί στη παρακάτω διατύπωση (5.2):

Αλγόριθμος 5.2. *Αλγόριθμος Μέσης Τιμής (Πολλές κατηγορίες) — Προσέγγιση Bard–Schweitzer*

(i) Αρχικές τιμές: $Q_{ij}(\boldsymbol{N}) \Leftarrow \dfrac{N^j}{M}$ για όλα τα i, j.

(ii) Εφαρμογή των Εξισώσεων (4.49)–(4.51) με χρήση της προσέγγισης (5.2) στην (4.49) και υπολογισμός των νέων $Q_{ij}(\boldsymbol{N})$.

(iii) Αν οι τιμές $Q_{ij}(\boldsymbol{N})$ που προκύπτουν από το βήμα (ii) διαφέρουν από τις προηγούμενες περισσότερο από ένα δεδομένο ποσοστό, επιστροφή στο βήμα (ii) με τις νέες τιμές.

Ο αλγόριθμος έχει πολύ καλή συμπεριφορά και απαιτεί $O(CM)$ χρόνο και χώρο ανά επανάληψη.

5.2 Φράγματα

Η απλούστερη τεχνική που χρησιμοποιείται για την ανάλυση υπολογιστικών συστημάτων με χρήση μοντέλων δικτύων αναμονής στηρίζεται στον υπολογισμό φραγμάτων για δείκτες επίδοσης. Με απλούς υπολογισμούς μπορούν να προσδιοριστούν φράγματα του ρυθμού απόδοσης και του χρόνου απόκρισης ενός συστήματος ως συναρτήσεις της έντασης φορτίου (ρυθμός αφίξεων ή αριθμός εργασιών). Θα διακρίνουμε δύο κατηγορίες φραγμάτων: *ασυμπτωτικά φράγματα* και *φράγματα ισορροπημένων συστημάτων*. Ο υπολογισμός φραγμάτων είναι ιδιαίτερα χρήσιμος για την αρχική μελέτη και αποτίμηση εναλλακτικών λύσεων και επιτρέπει την κατανόηση των πρωταρχικών παραγόντων που επηρεάζουν τη συμπεριφορά ενός συστήματος. Θα περιοριστούμε στην παρουσίαση μοντέλων με μία κατηγορία εργασιών, αν και υπάρχουν αποτελέσματα για μοντέλα με πολλές κατηγορίες. Ο κύριος λόγος είναι ότι τα μοντέλα μιας κατηγορίας είναι ικανοποιητικά στην πράξη. Εξάλλου, το κύριο πλεονέκτημα της μεθόδου είναι η απλότητά της, η οποία περιορίζεται όταν θεωρήσουμε πολλές κατηγορίες.

Θα χρησιμοποιήσουμε τους εξής πρόσθετους συμβολισμούς:

$D_{\max}$ Η μέγιστη απαίτηση εξυπηρέτησης στους σταθμούς του δικτύου. Ο σταθμός που αντιστοιχεί σε αυτήν την απαίτηση αποτελεί *στένωση* (bottleneck) του συστήματος.

$D = \sum_{i=1}^{M} D_i$ Το άθροισμα των απαιτήσεων εξυπηρέτησης σε όλους τους σταθμούς.

Z Ο μέσος χρόνος σκέψης των χρηστών (για κλειστό δίκτυο με τερματικά).

5.2.1 Ασυμπτωτικά Φράγματα

Τα ασυμπτωτικά φράγματα προσδιορίζονται θεωρώντας (ασυμπτωτικά) οριακές συνθήκες λειτουργίας του συστήματος (ελαφρύ και βαρύ φορτίο).

5.2.1.1 Ανοικτά Δίκτυα

Από τους βασικούς νόμους έχουμε $U_i = XD_i$, για το βαθμό χρησιμοποίησης κάθε σταθμού i, όπου ισχύει $X = \lambda$ για ανοικτό δίκτυο σε ισορροπία. Ο ρυθμός αφίξεων λ μπορεί να αυξηθεί μέχρις ότου κάποιος σταθμός φθάσει σε κορεσμό (βαθμός χρησιμοποίησης ίσος με 1). Συνεπώς, ένα άνω φράγμα για τον ρυθμό λ θα καθορίζεται από τον σταθμό με τη μεγαλύτερη απαίτηση εξυπηρέτησης (στένωση):

$$\lambda \leq \frac{1}{D_{\max}}$$

Ένα κάτω φράγμα για τον χρόνο απόκρισης προκύπτει αν θεωρήσουμε την καλύτερη δυνατή περίπτωση, κατά την οποία οι εργασίες δεν καθυστερούν καθόλου λόγω αναμονής:

$$T(\lambda) \geq D$$

5.2.1.2 Κλειστά Δίκτυα

Για να υπολογίσουμε φράγματα για τον ρυθμό απόδοσης λ θα θεωρήσουμε δύο περιπτώσεις: *βαρύ φορτίο* και *ελαφρύ φορτίο*.

Στην πρώτη περίπτωση, όσο ο αριθμός N των πελατών αυξάνει, αυξάνει και ο βαθμός χρησιμοποίησης των σταθμών, οπότε όπως και για τα ανοικτά δίκτυα βρίσκουμε:

$$X(N) \leq \frac{1}{D_{\max}}$$

Για ελαφρύ φορτίο, μπορούμε να σκεφθούμε ως εξής: ο χαμηλότερος ρυθμός απόδοσης προκύπτει όταν ένας πελάτης υποχρεώνεται να αναμένει όλους τους άλλους πελάτες σε κάθε σταθμό, οπότε δαπανά χρόνο $(N-1)D$ σε αναμονή, χρόνο D σε εξυπηρέτηση και χρόνο Z σε σκέψη. Ο ρυθμός απόδοσης θα είναι τότε $N/(ND+Z)$. Ο μέγιστος ρυθμός απόδοσης θα προκύπτει όταν οι πελάτες δεν καθυστερούν καθόλου λόγω αναμονής οπότε θα έχουμε ρυθμό $N/(D+Z)$.

Οι παραπάνω περιπτώσεις συνοψίζονται στη σχέση:

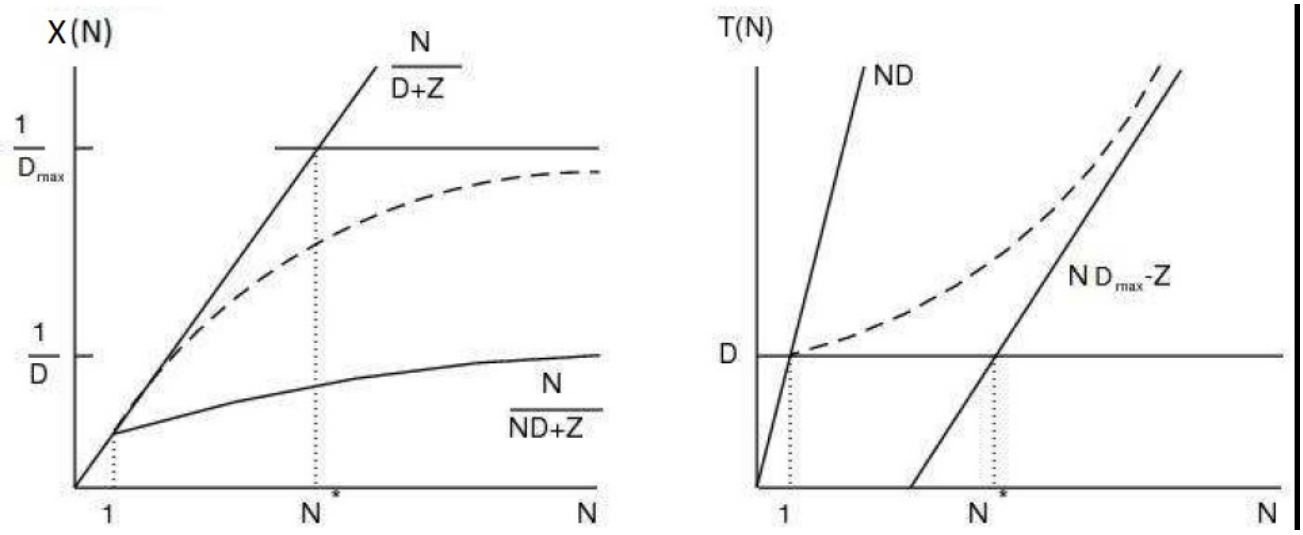
$$\frac{N}{ND+Z} \leq X(N) \leq \min\left(\frac{1}{D_{\max}}, \frac{N}{D+Z}\right) \tag{5.3}$$

Εφαρμόζοντας τον τύπο του Little στα φράγματα του ρυθμού απόδοσης υπολογίζουμε φράγματα στον χρόνο απόκρισης. (Ισχύει $X(N) = N/(T(N)+Z)$.) Παίρνουμε τελικά:

$$\max(D, ND_{\max} - Z) \leq T(N) \leq ND$$

Το άνω φράγμα του ρυθμού απόδοσης (αντίστοιχα το κάτω φράγμα του χρόνου απόκρισης) αποτελείται από δύο συνιστώσες, μία για βαρύ και μία για ελαφρύ φορτίο. Το σημείο τομής των δύο συνιστωσών ορίζει μία τιμή του πληθυσμού $N^\star = (D+Z)/D_{\max}$.

Η γενική μορφή των ασυμπτωτικών φραγμάτων για τον ρυθμό απόδοσης και τον χρόνο απόκρισης φαίνεται στο Σχήμα 5.1. Τα φράγματα καθορίζουν μία περιοχή τιμών μέσα στην οποία βρίσκονται οι τιμές



Σχήμα 5.1: *Ασυμπτωτικά Φράγματα για Κλειστά Δίκτυα.*

των ως άνω δεικτών. (Οι διακεκομμένες καμπύλες στο εσωτερικό των περιοχών παριστάνουν τυπικές μορφές μεταβολής των μεγεθών.)

Οι εκφράσεις και οι αντίστοιχες γραφικές παραστάσεις απλουστεύονται για κλειστό σύστημα χωρίς τερματικά ($Z = 0$).

Παράδειγμα 5.1. *Σχεδιάζεται ένα πείραμα ελέγχου του φορτίου σε ένα διαλογικό σύστημα, το οποίο περιλαμβάνει μια ΚΜΕ και δύο δίσκους με αντίστοιχη μέση συνολική απαίτηση εξυπηρέτησης εργασιών* 0, 42 sec, 0, 18 sec *και* 0, 25 sec. *Κατά τη διάρκεια του πειράματος, εικονικοί χρήστες στέλνουν ερωτήσεις στο σύστημα με μέσο χρόνο σκέψης* 12 sec.
(α) Με βάση τα ασυμπτωτικά φράγματα, πόσους τουλάχιστον εικονικούς χρήστες απαιτεί το πείραμα ώστε να μπορεί να επιτευχθεί ρυθμός απόδοσης 2 *ερωτήσεις*/sec*;*
(β) Με βάση τα ασυμπτωτικά φράγματα, πόσοι το πολύ εικονικοί χρήστες μπορούν να λάβουν μέρος στο πείραμα ώστε να μπορεί να επιτευχθεί μέσος χρόνος απόκρισης 5, 5 sec*;*
(γ) Αν ο αριθμός ενεργών τερματικών είναι 48*, κατά πόσο θα πρέπει να αυξηθεί η ταχύτητα του Δίσκου* 2 *ή/και της ΚΜΕ, ώστε με βάση τα ασυμπτωτικά φράγματα ο μέσος χρόνος απόκρισης να μπορεί να πάρει την τιμή* 4, 5 sec*; Να εξεταστεί το ίδιο για* 70 *τερματικά.*

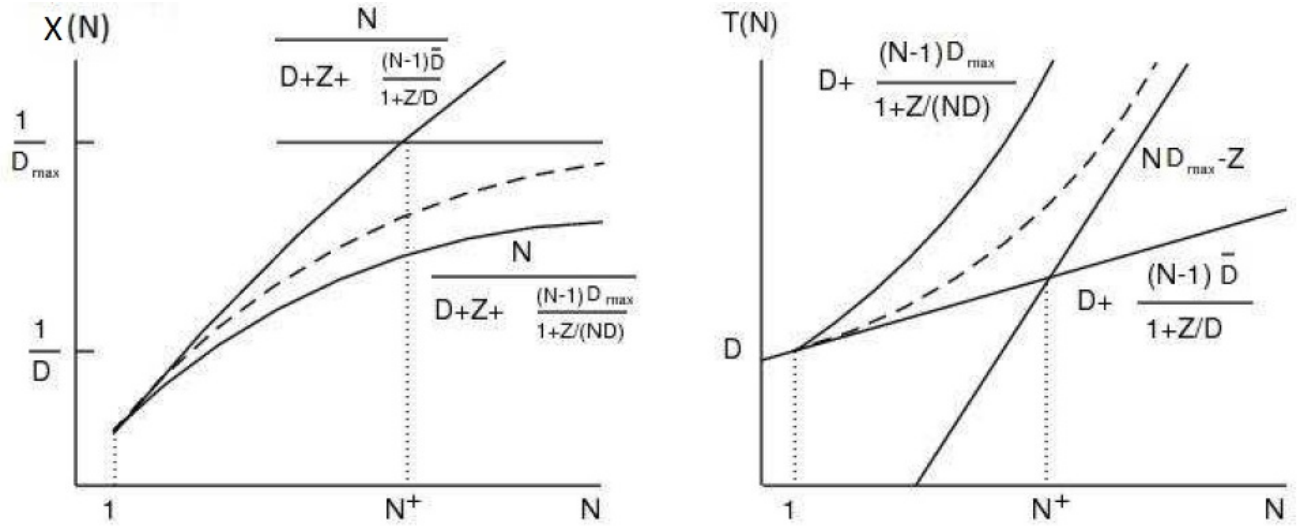
Για λόγους οικονομίας αναφερόμαστε στις τυπικές γραφικές παραστάσεις του Σχ. 5.1, των οποίων τη γενική μορφή ακολουθούν τα συγκεκριμένα παραδείγματα. Έχουμε από τα δεδομένα: Z=12 sec, D_{CPU}=0,42 sec=$D_{\max}$, D_{d1}=0,18 sec και D_{d2}=0,25 sec, οπότε D=0,85, $1/D$= 1,1765, $1/D_{\max}$=2,381, $N^{\star} = (D + Z)/D_{\max}$=30,6.

(α) Από τη γραφική απεικόνιση είναι φανερό ότι η ικανοποίηση του περιορισμού καθορίζεται από την τιμή του N στην οποία το άνω φράγμα ελαφρού φορτίου τέμνεται από την $X(N)$=2, δηλαδή $N/12, 85 = 2$ ή $N = 25, 7$. Άρα, ο ελάχιστος (ακέραιος) αριθμός χρηστών είναι $N = 26$.

(β) Όπως προηγουμένως, από τη γραφική απεικόνιση προκύπτει ότι η ικανοποίηση του περιορισμού καθορίζεται από την τιμή του N στην οποία το κάτω φράγμα βαρέος φορτίου τέμνεται από την $T(N)$=5,5, δηλαδή $N \times 0, 42 - 12 = 5, 5$ ή $N = 41, 67$. Άρα, ο μέγιστος αριθμός χρηστών είναι $N = 41$.

(γ) Έχουμε N=48. Όπως παραπάνω, ο περιορισμός σχετίζεται με το κάτω φράγμα βαρέος φορτίου. Έχουμε $48 \times D_{\max} - 12 = 4, 5$ ή $D_{\max}$= 0,3438 sec. Η ΚΜΕ είναι η στένωση του συστήματος, άρα οποιαδήποτε βελτίωση πρέπει να αρχίζει από την ΚΜΕ. Στην περίπτωση αυτή, θα πρέπει να αυξηθεί η ταχύτητα της ΚΜΕ κατά 0,42/0,3438=1,222 ή κατά 22,2%. Η ΚΜΕ παραμένει η στένωση, άρα δεν έχει νόημα η αύξηση της ταχύτητας του δίσκου ή άλλου συστατικού του συστήματος.

Εργαζόμενοι ομοίως στη δεύτερη περίπτωση, έχουμε $70 \times D_{\max} - 12 = 4, 5$ ή $D_{\max}$= 0,2357 sec, συνεπώς απαιτείται αύξηση της ταχύτητας της ΚΜΕ κατά 0,42/0,2357=1,782 ή κατά 78,2%. Όμως, αν



Σχήμα 5.2: *Φράγματα Ισορροπημένων Συστημάτων για Κλειστά Δίκτυα.*

γίνει η αλλαγή αυτή, η ΚΜΕ παύει να είναι η στένωση και τον ρόλο αυτό αναλαμβάνει ο Δίσκος 2 (δεύτερη στένωση). Για να επιτευχθεί το ζητούμενο, θα πρέπει και ο Δίσκος 2 να αποκτήσει μεγαλύτερη ταχύτητα, ώστε να φθάσει στο ίδιο επίπεδο ($D_{\max}$), άρα χρειάζεται αύξηση της ταχύτητας του Δίσκου 2 κατά 0,25/0,2357=1,0607 ή κατά 6,07%. □

5.2.2 Φράγματα Ισορροπημένων Συστημάτων

Ένα σύστημα ονομάζεται *ισορροπημένο* όταν η απαίτηση εξυπηρέτησης είναι ίδια σε όλους τους σταθμούς: $D_1 = D_2 = \ldots = D_M$. Συνεπώς, σε ένα ισορροπημένο σύστημα δεν υπάρχει σταθμός που να αποτελεί στένωση. Τα ισορροπημένα συστήματα χαρακτηρίζονται από ορισμένες ιδιότητες, οι οποίες μπορούν να χρησιμοποιηθούν για τον υπολογισμό φραγμάτων. Ειδικότερα, όταν δίνεται ένα σύστημα, μπορούν να οριστούν κατάλληλα ισορροπημένα συστήματα των οποίων ο ρυθμός απόδοσης και ο χρόνος απόκρισης αποτελούν φράγματα για τους αντίστοιχους δείκτες του δεδομένου συστήματος [7, 12].

Θα παραθέσουμε χωρίς απόδειξη μερικές εκφράσεις φραγμάτων για ανοικτά και κλειστά δίκτυα. Όπως φαίνεται και στο Σχήμα 5.2 για κλειστά δίκτυα, η μέθοδος των ισορροπημένων συστημάτων παρέχει σαφώς στενότερα φράγματα σε σχέση με τα αντίστοιχα ασυμπτωτικά, ενώ το πρόσθετο υπολογιστικό κόστος είναι σχετικά μικρό. (Τυπικές καμπύλες μεταβολής των μεγεθών εμφανίζονται διακεκομμένες.) Και στην περίπτωση αυτή, οι εκφράσεις και οι γραφικές παραστάσεις απλουστεύονται για κλειστό σύστημα χωρίς τερματικά (εργασίες batch). Το άνω φράγμα του ρυθμού απόδοσης (αντίστοιχα, το κάτω φράγμα του χρόνου απόκρισης) αποτελείται και εδώ από δύο συνιστώσες, το σημείο τομής των οποίων ορίζει μία άλλη χαρακτηριστική τιμή του πληθυσμού που θα συμβολίσουμε με N^+. Συμβολίζουμε με $\overline{D}$ τη μέση τιμή της απαίτησης εξυπηρέτησης για όλους τους σταθμούς του υπό μελέτη δικτύου.

5.2.2.1 Ανοικτά Δίκτυα

Για τον ρυθμό απόδοσης έχουμε:

$$\lambda \leq \frac{1}{D_{\max}}$$

(που ταυτίζεται με το αντίστοιχο ασυμπτωτικό) και για τον χρόνο απόκρισης:

$$\frac{D}{1-\lambda\overline{D}} \leq T(\lambda) \leq \frac{D}{1-\lambda D_{\max}}$$

5.2.2.2 Κλειστά Δίκτυα

Ρυθμός απόδοσης:

$$\frac{N}{D+Z+\frac{(N-1)D_{\max}}{1+Z/(ND)}} \leq X(N) \leq \min\left(\frac{1}{D_{\max}}, \frac{N}{D+Z+\frac{(N-1)\overline{D}}{1+Z/D}}\right)$$

Χρόνος απόκρισης:

$$\max\left(ND_{\max}-Z, D+\frac{(N-1)\overline{D}}{1+Z/D}\right) \leq T(N) \leq D+\frac{(N-1)D_{\max}}{1+Z/(ND)}$$

5.3 Προσέγγιση MVA για Σταθμούς με Ρυθμό Εξαρτώμενο από το Φορτίο

5.3.1 Κλειστά Δίκτυα

Ο αλγόριθμος MVA μπορεί να επιταχυνθεί μέσω προσεγγίσεων και στην περίπτωση μεταβλητών ρυθμών, η οποία χαρακτηρίζεται από αυξημένο υπολογιστικό κόστος. Ο συνδυασμός με την περίπτωση σταθερών ρυθμών επιτρέπει τη διατύπωση γενικού προσεγγιστικού αλγορίθμου για όλους τους τύπους σταθμών σε κλειστά δίκτυα πολλών κατηγοριών [10].

Όπως στην ακριβή διατύπωση του αλγορίθμου, θα θεωρήσουμε ότι ο ρυθμός εξυπηρέτησης εξαρτάται από τον συνολικό αριθμό πελατών στο σταθμό (και όχι από τον αριθμό πελατών κάθε κατηγορίας) και ότι η εξάρτηση είναι ίδια για όλες τις κατηγορίες. Σε αναλογία με τους σταθμούς σταθερού ρυθμού, η προσέγγιση για σταθμούς με ρυθμό εξαρτώμενο από το φορτίο βασίζεται στην τροποποίηση της περιθώριας κατανομής πιθανότητας $p_i(k|\boldsymbol{N})$ να υπάρχουν k πελάτες στο σταθμό i (ανεξάρτητα από κατηγορία), όταν ο πληθυσμός του δικτύου είναι $\boldsymbol{N}$:

$$p_i(k|\boldsymbol{N}) = \sum_{j=1}^{C} \frac{X_{ij}(\boldsymbol{N})}{\mu_{ij}(k)} p_i(k-1|\boldsymbol{N}-\mathbf{1}_j), k = 1, \ldots, N$$

Η προσέγγιση θα βασιστεί στην έκφραση:

$$p_i(k|\boldsymbol{N}-\mathbf{1}_j) \approx p_i(k|\boldsymbol{N}), k = 1, \ldots, N-1 \tag{5.4}$$

δηλαδή, θα υποθέσουμε ότι η αφαίρεση ενός πελάτη της κατηγορίας j δεν επηρεάζει σημαντικά τη συνολική κατανομή πελατών στον σταθμό i.

Καταρχάς ορίζουμε τον *πολλαπλασιαστή ρυθμού εξυπηρέτησης*:

$$\alpha_{ij}(k) = \frac{\mu_{ij}(k)}{\mu_{ij}(1)} = \mu_{ij}(k) \cdot S_{ij}$$

Εφόσον έχουμε υποθέσει ίδια εξάρτηση για όλες τις κατηγορίες j θα ισχύει:

$$\alpha_{ij}(k) = \alpha_i(k)$$

Επομένως μπορούμε να γράψουμε:

$$\frac{X_{ij}(\boldsymbol{N})}{\mu_{ij}(k)} = \frac{X_{ij}(\boldsymbol{N}) \cdot S_{ij}}{\alpha_{ij}(k)} = \frac{U_{ij}(\boldsymbol{N})}{\alpha_i(k)} = \frac{X^j(\boldsymbol{N}) D_{ij}}{\alpha_i(k)}$$

Σύμφωνα με την προσέγγιση έχουμε:

$$p_i(k|\boldsymbol{N}) = \sum_{j=1}^{C} \frac{X_{ij}(\boldsymbol{N})}{\mu_{ij}(k)} p_i(k-1|\boldsymbol{N}-\mathbf{1}_j) \approx \sum_{j=1}^{C} \frac{X^j(\boldsymbol{N}) \cdot D_{ij}}{\alpha_i(k)} p_i(k-1|\boldsymbol{N}), k = 1, \ldots, N \tag{5.5}$$

Με αναδρομική εφαρμογή της τελευταίας παίρνουμε κλειστή μορφή για τις πιθανότητες $p_i(k|\boldsymbol{N})$ συναρτήσει της $p_i(0|\boldsymbol{N})$, η οποία προσδιορίζεται από την εξίσωση κανονικοποίησης:

$$p_i(k|\boldsymbol{N}) = p_i(0|\boldsymbol{N}) \cdot \prod_{l=1}^{k} \frac{\sum_{j=1}^{C} X^j(\boldsymbol{N}) \cdot D_{ij}}{\alpha_i(l)}, k = 1, \ldots, N \tag{5.6}$$

$$p_i(0|\boldsymbol{N}) = \left[1 + \sum_{k=1}^{N} \prod_{l=1}^{k} \frac{\sum_{j=1}^{C} X^j(\boldsymbol{N}) \cdot D_{ij}}{\alpha_i(l)}\right]^{-1} \tag{5.7}$$

όπου $N = \|\boldsymbol{N}\|$.

Διαθέτοντας τις πιθανότητες μπορούμε εύκολα να υπολογίσουμε τον συνολικό χρόνο απόκρισης

$$R_{ij}(\boldsymbol{N}) = v_{ij} \sum_{k=1}^{N} \frac{k}{\mu_{ij}(k)} p_i(k-1|\boldsymbol{N}-\mathbf{1}_j) \approx D_{ij} \sum_{k=1}^{N} \frac{k}{\alpha_i(k)} p_i(k-1|\boldsymbol{N}) \tag{5.8}$$

Ο ρυθμός απόδοσης θα υπολογίζεται κατά τα γνωστά:

$$X^j(\boldsymbol{N}) = \frac{N_j}{\sum_{i=1}^{M} R_{ij}(\boldsymbol{N})} \tag{5.9}$$

Συνδυασμός των σχέσεων για σταθμούς με σταθερούς ρυθμούς και σταθμούς με ρυθμό εξαρτώμενο από το φορτίο δίνει τον γενικό αλγόριθμο για προσεγγιστική επίλυση κλειστών δικτύων (Αλγόριθμος 5.3).

Αλγόριθμος 5.3. *Προσέγγιση για κλειστά δίκτυα με ρυθμούς εξυπηρέτησης εξαρτώμενους από το φορτίο (Πολλές κατηγορίες)*

Αρχικοποίηση

- Για κάθε σταθμό i και κατηγορία j θέτουμε ως αρχική τιμή του μέσου αριθμού εργασιών:

$$Q_{ij} = \frac{N^j}{M}$$

- Για κάθε κατηγορία j θέτουμε ως αρχική τιμή του ρυθμού απόδοσης το ασυμπτωτικό άνω φράγμα, κατ' αναλογία προς το μοντέλο μιας κατηγορίας (Εξίσωση (5.3)):

$$X^j = \min\left(\frac{1}{\max_i D_{ij}}, \frac{N^j}{\sum_{i=1}^{M} D_{ij}}\right)$$

Επανάληψη

Επαναλαμβάνουμε τα παρακάτω βήματα μέχρι την επίτευξη επιθυμητής ακρίβειας ως προς τις τιμές του ρυθμού απόδοσης X^j:

- Υπολογισμός πιθανοτήτων $p_i(k|\boldsymbol{N})$ για σταθμούς με ρυθμό εξαρτώμενο από το φορτίο.
- Υπολογισμός χρόνων παραμονής για όλους τους σταθμούς. Εφαρμογή προσέγγισης με χρήση του μέσου αριθμού πελατών Q_{ij} ή των πιθανοτήτων $p_i(k|\boldsymbol{N})$ αναλόγως του τύπου του σταθμού.
- Υπολογισμός ρυθμού απόδοσης X^j ανά κατηγορία.

- Υπολογισμός μέσου αριθμού πελατών Q_{ij} για σταθμούς σταθερού ρυθμού.

5.3.2 Ανοικτά Δίκτυα

Η μέθοδος που θα περιγράψουμε είναι ακριβής και επεκτείνει τη βασική μέθοδο επίλυσης ανοικτών δικτύων πολλών κατηγοριών ενσωματώνοντας σταθμούς με ρυθμούς εξυπηρέτησης εξαρτώμενους από το φορτίο [10]. Όπως και για τα κλειστά δίκτυα παραπάνω, υποθέτουμε και εδώ ότι ο πολλαπλασιαστής ρυθμού εξυπηρέτησης $\alpha_{ij}(k)$ είναι ανεξάρτητος της κατηγορίας, δηλαδή $\alpha_{ij}(k) = \alpha_i(k)$ για όλες τις κατηγορίες j. Θα θεωρήσουμε την πιθανότητα $p_i(k|\boldsymbol{\lambda})$, $k \geq 0$, να υπάρχουν k εργασίες στον σταθμό i (ανεξαρτήτως κατηγορίας), όταν το διάνυσμα αφίξεων είναι $\boldsymbol{\lambda} = [\lambda^1, \ldots, \lambda^C]$, όπου λ^j ο ρυθμός άφιξης της κατηγορίας j. Για την επίλυση του δικτύου απαιτείται ο υπολογισμός της κατανομής πιθανότητας $p_i(k|\boldsymbol{\lambda})$.

Ο *ονομαστικός βαθμός χρησιμοποίησης* του σταθμού i αναφέρεται στην κατάσταση κατά την οποία υπάρχει μία μόνο εργασία στον σταθμό ($k = 1$):

$$U_i(\boldsymbol{\lambda}) = \sum_j U_{ij}(\boldsymbol{\lambda}) = \sum_j \lambda^j D_{ij} \tag{5.10}$$

Εφόσον υπάρχει κατάσταση ισορροπίας, η κατανομή πιθανότητας $p_i(k|\boldsymbol{\lambda})$ θα ικανοποιεί τις εξισώσεις:

$$\begin{aligned} p_i(k|\boldsymbol{\lambda}) &= \sum_{j=1}^{C} \frac{X_{ij}}{\mu_{ij}(k)} p_i(k-1|\boldsymbol{\lambda}) \\ &= \sum_{j=1}^{C} \frac{\lambda^j D_{ij}}{\alpha_i(k)} p_i(k-1|\boldsymbol{\lambda}) \\ &= \frac{U_i}{\alpha_i(k)} p_i(k-1|\boldsymbol{\lambda}), k \geq 1 \end{aligned} \tag{5.11}$$

όπου για λόγους απλότητας παραλείπουμε την εξάρτηση του βαθμού χρησιμοποίησης U_i από την είσοδο $\boldsymbol{\lambda}$.

Η επίλυση των εξισώσεων παρέχει τις πιθανότητες $p_i(k|\boldsymbol{\lambda})$ συναρτήσει της $p_i(0|\boldsymbol{\lambda})$, η οποία προσδιορίζεται από την εξίσωση κανονικοποίησης:

$$p_i(k|\boldsymbol{\lambda}) = p_i(0|\boldsymbol{\lambda}) \frac{U_i^k}{A_i(k)}, k \geq 1 \tag{5.12}$$

$$p_i(0|\boldsymbol{\lambda}) = \left[1 + \sum_{k=1}^{\infty} \frac{U_i^k}{A_i(k)} \right]^{-1} \tag{5.13}$$

όπου $A_i(k) = \prod_{l=1}^{k} \alpha_i(l)$.

Θα υποθέσουμε ότι, για κάθε σταθμό i με εξάρτηση από το φορτίο, ο πολλαπλασιαστής ρυθμού εξυπηρέτησης $\alpha_i(k)$ είναι σταθερός μετά από κάποια τιμή c_i του φορτίου:

$$\alpha_i(k) = \alpha_i(c_i), k \geq c_i$$

Η υπόθεση αυτή, η οποία ικανοποιείται συνήθως στην πράξη, επιτρέπει τον υπολογισμό κλειστής μορφής για τις πιθανότητες $p_i(k|\boldsymbol{\lambda})$. Εξάλλου, στην περίπτωση αυτή, η συνθήκη ευστάθειας για τον σταθμό i ανάγεται στην απαίτηση:

$$\frac{U_i(\boldsymbol{\lambda})}{\alpha_i(c_i)} < 1$$

Με βάση την παραπάνω υπόθεση, η κατανομή πιθανότητας $p_i(k|\boldsymbol{\lambda})$ παίρνει τη μορφή:

$$p_i(k|\boldsymbol{\lambda}) = \begin{cases} p_i(0|\boldsymbol{\lambda})\frac{U_i^k}{A_i(k)}, & k = 1, \ldots, c_i \\ p_i(0|\boldsymbol{\lambda})\frac{U_i^k}{A_i(c_i)[\alpha_i(c_i)]^{k-c_i}}, & k > c_i \end{cases} \tag{5.14}$$

$$p_i(0|\boldsymbol{\lambda}) = \left[1 + \sum_{k=1}^{c_i} \frac{U_i^k}{A_i(k)} + \frac{U_i^{c_i+1}}{A_i(c_i)\alpha(c_i)} \frac{1}{1 - \frac{U_i}{\alpha_i(c_i)}}\right]^{-1} \tag{5.15}$$

Είμαστε τώρα σε θέση να υπολογίσουμε τον μέσο αριθμό εργασιών στον σταθμό i (ανεξαρτήτως κατηγορίας):

$$\begin{aligned} Q_i(\boldsymbol{\lambda}) &= \sum_{k=1}^{\infty} k p_i(k|\boldsymbol{\lambda}) \\ &= p_i(0|\boldsymbol{\lambda}) \left[\sum_{k=1}^{c_i} k \frac{U_i^k}{A_i(k)} + \frac{U_i^{c_i+1}}{A_i(c_i)\alpha_i(c_i)} \frac{1 + c_i\left(1 - \frac{U_i}{\alpha_i(c_i)}\right)}{\left(1 - \frac{U_i}{\alpha_i(c_i)}\right)^2}\right] \end{aligned} \tag{5.16}$$

Μπορούμε εύκολα να δείξουμε ότι

$$Q_{ij}(\boldsymbol{\lambda}) = \frac{U_{ij}}{U_i} Q_i(\boldsymbol{\lambda}) \tag{5.17}$$

δηλαδή, ισχύει αντίστοιχη αναλογία με την περίπτωση των σταθερών ρυθμών (Εξίσωση (4.39)). Τέλος, ο χρόνος απόκρισης προκύπτει με απευθείας εφαρμογή του Τύπου του Little:

$$R_{ij}(\boldsymbol{\lambda}) = v_{ij} T_{ij}(\boldsymbol{\lambda}) = v_{ij} \frac{Q_{ij}(\boldsymbol{\lambda})}{\lambda_{ij}} = \frac{Q_{ij}(\boldsymbol{\lambda})}{\lambda^j} \tag{5.18}$$

Με βάση τα παραπάνω καταλήγουμε σε μία γενική μέθοδο επίλυσης ανοικτών δικτύων πολλών κατηγοριών που περιλαμβάνουν σταθμούς εξαρτώμενους από το φορτίο (Αλγόριθμος 5.4):

Αλγόριθμος 5.4. *Προσέγγιση για ανοικτά δίκτυα με ρυθμούς εξυπηρέτησης εξαρτώμενους από το φορτίο (Πολλές κατηγορίες)*

- Υπολογισμός του βαθμού χρησιμοποίησης U_{ij} για κάθε σταθμό i και κατηγορία j.
- Έλεγχος της συνθήκης ευστάθειας για κάθε σταθμό.
- Υπολογισμός του μέσου αριθμού εργασιών $Q_i(\boldsymbol{\lambda})$ για τους σταθμούς με ρυθμό εξαρτώμενο από το φορτίο (LD) ανεξαρτήτως κατηγορίας.
- Υπολογισμός του μέσου αριθμού εργασιών $Q_{ij}(\boldsymbol{\lambda})$ για κάθε κατηγορία και σταθμό αναλόγως του τύπου του σταθμού:

$$Q_{ij}(\boldsymbol{\lambda}) = \begin{cases} U_{ij} & \text{Καθυστέρηση} \\ \frac{U_{ij}}{1-U_i} & \text{Αναμονή LI} \\ \frac{U_{ij}}{U_i} Q_i(\boldsymbol{\lambda}) & \text{Αναμονή LD} \end{cases}$$

- Υπολογισμός του μέσου χρόνου παραμονής $R_{ij}(\boldsymbol{\lambda})$ για κάθε κατηγορία και σταθμό αναλόγως

του τύπου του σταθμού:

$$R_{ij}(\boldsymbol{\lambda}) = \begin{cases} D_{ij} & \text{Καθυστέρηση} \\ \frac{D_{ij}}{1-U_i} & \text{Αναμονή LI} \\ \frac{Q_{ij}(\boldsymbol{\lambda})}{\lambda^j} & \text{Αναμονή LD} \end{cases}$$

5.4 Η Ισοδυναμία της Ροής — Ιεραρχική Μοντελοποίηση

Συχνά είναι απαραίτητη η κατασκευή πολύπλοκων μοντέλων που περιγράφουν με λεπτομέρεια τα χαρακτηριστικά υπολογιστικών συστημάτων. Μία μέθοδος που χρησιμοποιείται ευρύτατα για την ανάπτυξη και επίλυση τέτοιων μοντέλων είναι η *ιεραρχική μοντελοποίηση* ή *ιεραρχική διάσπαση*, η οποία στηρίζεται στις αρχές της *συνάθροισης* και *απομόνωσης*. Σύμφωνα με την τεχνική αυτή, μια ομάδα σταθμών ενός δικτύου μπορούν να συναθροιστούν και να αποτελέσουν μικρότερο υποδίκτυο, το οποίο επιλύεται ανεξάρτητα (σε απομόνωση). Στη συνέχεια, η λύση αυτή χρησιμοποιείται για την επίλυση του αρχικού μοντέλου, αντικαθιστώντας το υποδίκτυο με ένα μοναδικό σταθμό, στον οποίο ο ρυθμός εξυπηρέτησης εξαρτάται από το μήκος της ουράς. Ο σταθμός αυτός θα πρέπει να συμπεριφέρεται προς το υπόλοιπο δίκτυο, όπως το υποδίκτυο που αντικαθιστά και θα ονομάζεται *ισοδύναμος σταθμός ως προς τη ροή* (flow equivalent service center). Η διαδικασία αυτή μπορεί να εφαρμοστεί σε πολλές ομάδες σταθμών του αρχικού δικτύου.

Χρησιμοποιώντας την έννοια του ισοδύναμου σταθμού μπορούμε να παραστήσουμε ιεραρχικά ένα σύστημα σε διάφορα επίπεδα λεπτομέρειας, όπου το κατώτατο επίπεδο είναι το αρχικό μοντέλο και κάθε επίπεδο προκύπτει από το χαμηλότερό του, αν μία ομάδα σταθμών αντικατασταθεί από έναν ισοδύναμο σταθμό. Έχουμε έτσι ένα είδος αφαίρεσης από τα χαμηλότερα προς τα υψηλότερα επίπεδα και βαθμιαίας συγκεκριμενοποίησης από τα υψηλότερα προς τα χαμηλότερα. Η επίλυση του μοντέλου πραγματοποιείται από τα χαμηλότερα προς τα υψηλότερα επίπεδα.

Ο σκοπός του ισοδύναμου σταθμού είναι να μιμηθεί τη συμπεριφορά της ομάδας των σταθμών που αντικαθιστά. Μία προσέγγιση της ισοδυναμίας της ροής στηρίζεται στην υπόθεση ότι ο μέσος ρυθμός αναχωρήσεων από την ομάδα των σταθμών εξαρτάται μόνο από τον συνολικό αριθμό πελατών στους σταθμούς της ομάδας (για κάθε κατηγορία, αν πρόκειται για μοντέλο πολλών κατηγοριών) και όχι από την ακριβή θέση των πελατών στους σταθμούς. Η υπόθεση αυτή υποδηλώνει χαλαρή σύζευξη μεταξύ της ομάδας των σταθμών και του υπόλοιπου συστήματος, δηλαδή ότι ο ρυθμός μετακινήσεων πελατών μεταξύ σταθμών της ομάδας είναι πολύ υψηλότερος του ρυθμού αφίξεων πελατών στην ομάδα από το υπόλοιπο σύστημα. Κατά την επίλυση των μοντέλων υψηλού επιπέδου θεωρούμε ότι ο ρυθμός εξυπηρέτησης στους ισοδύναμους σταθμούς εξαρτάται από το φορτίο (τον αριθμό των πελατών). Ο ρυθμός αυτός τίθεται ίσος με τον ρυθμό απόδοσης της ομάδας των σταθμών, αν το αντίστοιχο υποδίκτυο επιλυθεί σε απομόνωση για τις δυνατές τιμές του πληθυσμού. Ένα πρόβλημα προκύπτει στην περίπτωση ανοικτών δικτύων, για τα οποία οι δυνατές τιμές του πληθυσμού σε ένα υποδίκτυο είναι άπειρες. Στην πράξη, όμως, θεωρούμε διαφορετικούς ρυθμούς για τιμές του πληθυσμού μέχρι κάποιο όριο και σταθερούς ρυθμούς πέρα από το όριο αυτό.

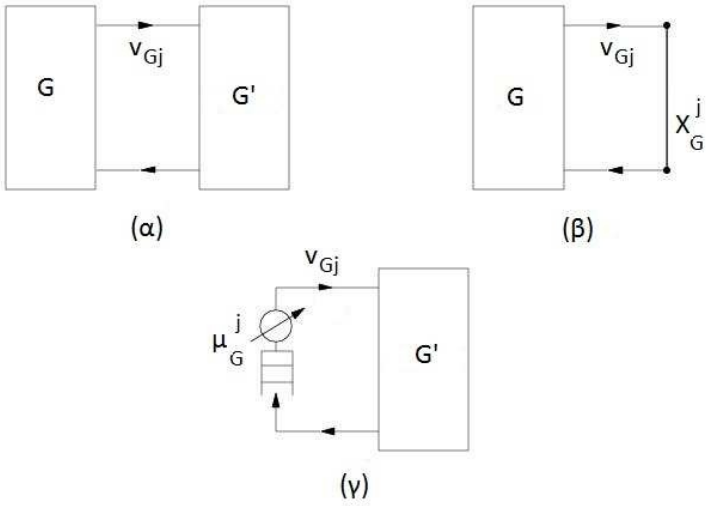
Κατά την επίλυση ενός υποδικτύου (ομάδας σταθμών) σε απομόνωση θεωρούμε ότι η σύνδεσή του με το υπόλοιπο σύστημα *βραχυκυκλώνεται* και υπολογίζουμε το ρυθμό απόδοσης στη βραχυκυκλωμένη σύνδεση ως συνάρτηση του πληθυσμού του υποδικτύου (μοντέλο χαμηλού επιπέδου). Είναι φανερή η αναλογία που υπάρχει ανάμεσα στην ιδέα του ισοδύναμου σταθμού και στο ισοδύναμο Norton στα ηλεκτρικά κυκλώματα.

Η επίλυση των μοντέλων υψηλού επιπέδου απαιτεί τεχνικές επίλυσης δικτύου με ρυθμούς εξαρτώμενους από το φορτίο. Οι τεχνικές αυτές, όπως είδαμε, είναι επεκτάσεις των βασικών αλγορίθμων, οι οποίοι υποθέτουν σταθερούς ρυθμούς.

Θα πρέπει να σημειωθεί ότι η μέθοδος της συνάθροισης και απομόνωσης, η οποία μελετήθηκε από τον P.-J. Courtois [3], είναι ακριβής, όταν το δίκτυο πληροί τις προυποθέσεις για λύση μορφής γινομένου, διαφορετικά χρησιμοποιείται ως προσεγγιστική τεχνική.

Η γενική μέθοδος της ιεραρχικής διάσπασης μπορεί να διατυπωθεί αδρά με τη μορφή του ακόλουθου αλγορίθμου για κλειστά δίκτυα (Αλγόριθμος 5.5).

Θεωρούμε ένα κλειστό δίκτυο M σταθμών με πληθυσμό $\boldsymbol{N}$ και υποθέτουμε ότι η ομάδα σταθμών G θα αντικατασταθεί από έναν ισοδύναμο σταθμό. (Θα συμβολίζουμε με G την ομάδα σταθμών και τον



Σχήμα 5.3: *Ιεραρχική μοντελοποίηση: (α) αρχικό δίκτυο, (β) μοντέλο χαμηλού επιπέδου, (γ) μοντέλο υψηλού επιπέδου.*

ισοδύναμο σταθμό. Με G' θα συμβολίζουμε το υπόλοιπο σύστημα.)

Αλγόριθμος 5.5. *Ιεραρχική διάσπαση*

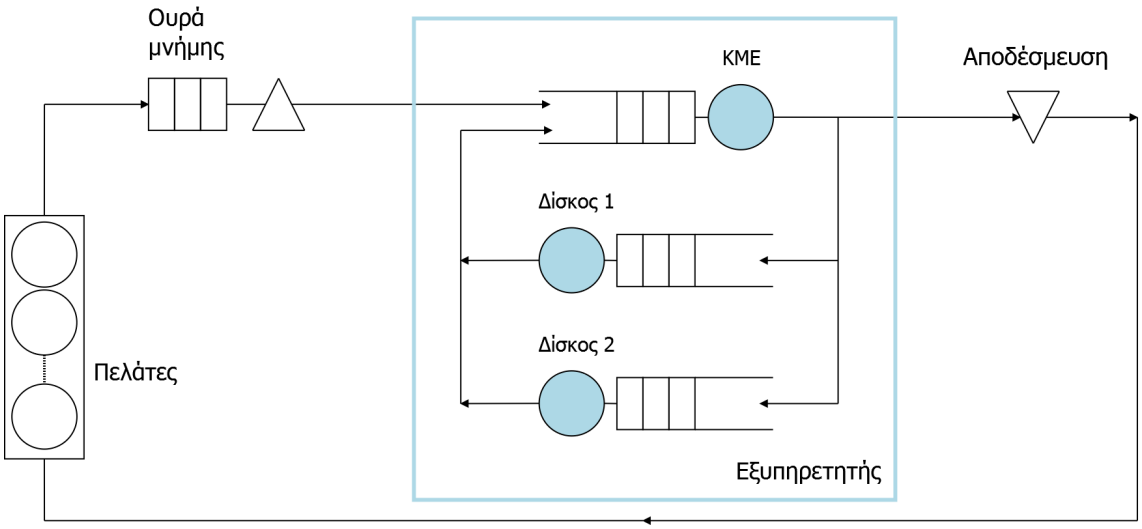
(i) Κατασκευάζουμε υποδίκτυο που αποτελείται μόνο από τους σταθμούς της ομάδας G (μοντέλο χαμηλού επιπέδου) και το επιλύουμε σε απομόνωση με τις γνωστές τεχνικές (MVA). Για κάθε κατηγορία j υπολογίζουμε τον ρυθμό απόδοσης του υποδικτύου $X_G^j(\boldsymbol{k})$ στη βραχυκυκλωμένη σύνδεση για κάθε $\boldsymbol{k} \leq \boldsymbol{N}$ (οι τιμές αυτές προκύπτουν ως ενδιάμεσα αποτελέσματα του αλγορίθμου MVA).

(ii) Κατασκευάζουμε το μοντέλο υψηλού επιπέδου αντικαθιστώντας στο αρχικό δίκτυο την ομάδα των σταθμών G με έναν ισοδύναμο σταθμό, τέτοιον ώστε ο ρυθμός εξυπηρέτησης $\mu_G^j(\boldsymbol{k})$ της κατηγορίας j για πληθυσμό $\boldsymbol{k}$ στο σταθμό να είναι ίσος με $X_G^j(\boldsymbol{k})$, όπως υπολογίστηκε στο προηγούμενο βήμα. Επιλύουμε το μοντέλο υψηλού επιπέδου χρησιμοποιώντας τη διατύπωση του αλγορίθμου MVA για ρυθμούς εξυπηρέτησης εξαρτώμενους από το φορτίο.

(iii) Οι δείκτες επίδοσης για τους σταθμούς εκτός του ισοδύναμου σταθμού προκύπτουν άμεσα, ενώ για τους σταθμούς στο εσωτερικό της ομάδας που αντιπροσωπεύει ο ισοδύναμος σταθμός υπολογίζονται συνδυάζοντας τα αποτελέσματα των μοντέλων χαμηλού και υψηλού επιπέδου.

Κατά την κατασκευή των μοντέλων χαμηλού και υψηλού επιπέδου θα πρέπει να υπάρχει συμβατότητα μεταξύ των δύο μοντέλων όσον αφορά τον μέσο αριθμό επισκέψεων των εργασιών στο υποδίκτυο (Σχήμα 5.3). Ειδικότερα, έστω στο αρχικό μοντέλο v_{Gj} ο μέσος αριθμός επισκέψεων μιας εργασίας της κατηγορίας j στο υποδίκτυο G (μέσος αριθμός διελεύσεων από τη σύνδεση που ενώνει την ομάδα σταθμών με το υπόλοιπο σύστημα). Κατά την επίλυση του μοντέλου χαμηλού επιπέδου θεωρούμε ότι κάθε εργασία διέρχεται κατά μέσο όρο v_{Gj} φορές από τη βραχυκυκλωμένη σύνδεση. Συνεπώς, κατά τον προσδιορισμό του ρυθμού απόδοσης $X_G^j(\boldsymbol{k})$ της βραχυκυκλωμένης σύνδεσης λαμβάνεται υπόψη το σύνολο των επισκέψεων που πραγματοποιεί μια εργασία στο υποδίκτυο. Έτσι, ο ρυθμός εξυπηρέτησης $\mu_G^j(\boldsymbol{k}) = X_G^j(\boldsymbol{k})$ του ισοδύναμου σταθμού αντιστοιχεί σε μια επίσκεψη στο υποδίκτυο. Κατά την επίλυση του μοντέλου υψηλού επιπέδου, ο μέσος αριθμός επισκέψεων στον ισοδύναμο σταθμό λαμβάνεται ίσος με v_{Gj}.

Οι εξισώσεις των επόμενων παραγράφων είναι προσαρμοσμένες στον παραπάνω τρόπο αντιμετώπισης του μέσου αριθμού επισκέψεων στα δύο μοντέλα.



Σχήμα 5.4: *Ουρά μνήμης (παθητική ουρά).*

5.5 Παραδείγματα Ανάλυσης Μη Διαχωρίσιμων Δικτύων

5.5.1 Συστήματα με Περιορισμούς Μνήμης

Σαν μία πρώτη εφαρμογή προσεγγιστικής επίλυσης θα εξετάσουμε την περίπτωση συστήματος με περιορισμένη μνήμη, το οποίο δεν μπορεί να παρασταθεί με δίκτυο που δέχεται λύση μορφής γινομένου. Σε ένα τέτοιο σύστημα, εξαιτίας του πεπερασμένου χώρου μνήμης σχηματίζεται ουρά από τις εργασίες που δεν μπορούν να φορτωθούν στο κεντρικό υποσύστημα (επεξεργαστής, δίσκοι, μονάδες εισόδου/εξόδου κλπ) για να εκτελεστούν. Οι εργασίες αυτές αναμένουν την απελευθέρωση (release) χώρου μνήμης λόγω της αναχώρησης εργασιών που εξυπηρετούνται (Σχήμα 5.4). Η *ουρά μνήμης* είναι μία *παθητική ουρά* (passive queue) δεδομένου ότι η παραμονή σε αυτήν καθορίζεται από τη δραστηριότητα σε άλλες ουρές που χαρακτηρίζονται *ενεργητικές* (active). Αντίστοιχα, η μνήμη θεωρείται *παθητικός πόρος* του συστήματος, καθόσον δεσμεύεται από μία εργασία ταυτόχρονα με κάποιον άλλο *ενεργητικό* πόρο. Το φαινόμενο αυτό είναι γενικότερο και αναφέρεται ως *αποκλεισμός* (blocking). Πρόκειται για διακοπή της λειτουργίας μιας συνιστώσας του συστήματος εξαιτίας της μη διαθεσιμότητας άλλων πόρων του συστήματος.

Σύμφωνα με την αρχή της διάσπασης χωρίζουμε το σύστημα σε δύο τμήματα: (α) το κεντρικό υποσύστημα μαζί με την ουρά μνήμης και (β) το εξωτερικό περιβάλλον (τερματικά). Στη συνέχεια, ορίζουμε έναν ισοδύναμο σταθμό, ο οποίος παριστάνει το κεντρικό υποσύστημα και περιλαμβάνει τους περιορισμούς μνήμης (σαν περιορισμούς στις δυνατές τιμές του πληθυσμού του). Η ανάλυση αυτή συμπίπτει με την αντίληψη του συστήματος από την πλευρά του χρήστη. Κάθε πελάτης μπορεί να βρίσκεται σε δύο βασικές καταστάσεις: *σε σκέψη* (στα τερματικά) ή *έτοιμος*. Λόγω των περιορισμών μνήμης, ένας έτοιμος πελάτης μπορεί να βρίσκεται σε δύο υπο-καταστάσεις: *σε αναμονή* (αποκλεισμένος στην ουρά μνήμης) ή *ενεργός* (σε εκτέλεση, δηλαδή σε κάποιον από τους σταθμούς του κεντρικού υποσυστήματος). Η συμπεριφορά των έτοιμων πελατών θα πρέπει να εκφράζεται μέσα από τα χαρακτηριστικά του ισοδύναμου σταθμού.

5.5.1.1 Μία Κατηγορία

Υποθέτουμε ότι όλοι οι πελάτες (εργασίες) έχουν τις ίδιες απαιτήσεις σε μνήμη και θεωρούμε ότι ο περιορισμός μνήμης επιβάλλει ότι L το πολύ πελάτες μπορεί να είναι φορτωμένοι στη μνήμη. Μπορούμε να διατυπώσουμε έναν απλό προσεγγιστικό αλγόριθμο επίλυσης του μοντέλου.

(i) Ορίζουμε ένα μοντέλο χαμηλού επιπέδου που αποτελείται από τους σταθμούς του κεντρικού υποσυστήματος (επεξεργαστές, δίσκοι κλπ.) σε απομόνωση (βραχυκυκλώνοντας τη σύνδεση των τερματικών). Επιλύουμε το μοντέλο χαμηλού επιπέδου (το οποίο δέχεται λύση μορφής γινομένου) για κάθε

δυνατό πληθυσμό $k = 1, \ldots, L$ και υπολογίζουμε τον ρυθμό απόδοσης ως συνάρτηση του φορτίου $X(k)$.

(ii) Κατασκευάζουμε ένα σταθμό με ρυθμό εξαρτώμενο από το φορτίο, ο οποίος είναι ισοδύναμος με το κεντρικό υποσύστημα συν την ουρά μνήμης, θέτοντας τον ρυθμό εξυπηρέτησής του:

$$\mu(k) = \begin{cases} X(k) & k = 1, \ldots, L \\ X(L) & k > L \end{cases}$$

Ορίζουμε ένα μοντέλο υψηλού επιπέδου, που αποτελείται από τον ισοδύναμο σταθμό και το εξωτερικό περιβάλλον (τερματικά) και το επιλύουμε υπολογίζοντας δείκτες επίδοσης του συστήματος.

5.5.1.2 Πολλές Κατηγορίες

Στην περίπτωση πολλών κατηγοριών θα υποθέσουμε ότι κάθε κατηγορία j έχει περιορισμό μνήμης L_j ανεξάρτητα από τις άλλες κατηγορίες. Ο αλγόριθμος που περιγράφηκε για μία κατηγορία γενικεύεται με προφανή τρόπο. Το πρόβλημα είναι ότι η γενίκευση αυτή έχει μεγάλο υπολογιστικό κόστος, όπως αναφέρθηκε ήδη για τα μοντέλα πολλών κατηγοριών με ρυθμούς εξυπηρέτησης εξαρτώμενους από το φορτίο. Πράγματι, για τον χαρακτηρισμό του ισοδύναμου σταθμού θα πρέπει να επιλυθεί το μοντέλο χαμηλού επιπέδου για όλους τους δυνατούς πληθυσμούς με κόστος ανάλογο του $CM \prod_{j=1}^{C}(L_j + 1)$. Επιπλέον, το μοντέλο υψηλού επιπέδου δεν δέχεται λύση μορφής γινομένου.

Για να υπερπηδήσουμε αυτές τις δυσκολίες εισάγουμε την ακόλουθη απλοποιητική υπόθεση:

- Υποθέτουμε ότι ο ρυθμός απόδοσης μιας κατηγορίας στο κεντρικό υποσύστημα εξαρτάται από τον πληθυσμό (αριθμό ενεργών εργασιών) της κατηγορίας αυτής και μόνο από τον μέσο πληθυσμό των άλλων κατηγοριών στο κεντρικό υποσύστημα. Επιπλέον, κάθε κατηγορία βλέπει τις άλλες κατηγορίες σαν ο πληθυσμός καθεμιάς στο κεντρικό υποσύστημα να ήταν ανεξάρτητος από τον πληθυσμό των άλλων.

Με βάση αυτήν την υπόθεση μπορούμε να διατυπώσουμε έναν ικανοποιητικό προσεγγιστικό αλγόριθμο [6, 7].

(i) Αγνοώντας τους περιορισμούς μνήμης έχουμε ένα δίκτυο με λύση μορφής γινομένου, το οποίο επιλύουμε. Θέτουμε A_j ίσο με το ελάχιστο μεταξύ του L_j και του μέσου πληθυσμού της κατηγορίας j στο κεντρικό υποσύστημα, όπως προκύπτει από τη λύση του δικτύου για όλες τις κατηγορίες j με περιορισμό μνήμης. Οι τιμές αυτές θα χρησιμοποιηθούν ως αρχικές τιμές στην επανάληψη που ακολουθεί.

(ii) Στη συνέχεια τροποποιούμε το αρχικό μοντέλο, μετατρέποντας κάθε κατηγορία j με περιορισμό μνήμης σε μία κατηγορία με σταθερό πληθυσμό ίσο με A_j χωρίς να λαμβάνουμε υπόψη τα τερματικά (κατηγορία τύπου batch), δηλαδή ουσιαστικά απομονώνουμε τις κατηγορίες με περιορισμό στο κεντρικό υποσύστημα. Οι κατηγορίες χωρίς περιορισμό παραμένουν όπως έχουν. Προκύπτει, έτσι, ένα μοντέλο πολλών κατηγοριών με λύση μορφής γινομένου. (Η ιδιαιτερότητα του μοντέλου αυτού είναι ότι οι κατηγορίες με περιορισμούς μπορεί να έχουν μη ακέραιο πληθυσμό. Η επίλυση, όμως, γίνεται εύκολα με χρήση του προσεγγιστικού αλγορίθμου MVA.)

(iii) Για κάθε κατηγορία j με περιορισμό μνήμης:

(α′) Στο μοντέλο που ορίστηκε στο βήμα (ii), αντικαθιστούμε τον πληθυσμό A_j με κάθε δυνατό πληθυσμό $k = 1, \ldots, L_j$ και επιλύουμε το μοντέλο υπολογίζοντας το ρυθμό απόδοσης $X^j(k)$ της κατηγορίας (διέλευση από τη βραχυκυκλωμένη σύνδεση).

(β′) Ορίζουμε ένα μοντέλο υψηλού επιπέδου μιας κατηγορίας που αποτελείται από το εξωτερικό περιβάλλον της κατηγορίας j (τερματικά) και έναν ισοδύναμο σταθμό με ρυθμό εξυπηρέτησης εξαρτώμενο από το φορτίο θέτοντας:

$$\mu_j(k) = \begin{cases} X^j(k) & k = 1, \ldots, L_j \\ X^j(L_j) & k > L_j \end{cases}$$

Επιλύουμε το μοντέλο υψηλού επιπέδου και υπολογίζουμε την κατανομή $p_j(k)$ του αριθμού πελατών στον ισοδύναμο σταθμό, την οποία χρησιμοποιούμε για να υπολογίσουμε μία νέα τιμή του μέσου πληθυσμού της κατηγορίας j στο κεντρικό υποσύστημα:

$$A_j = \sum_{k=1}^{L_j} k p_j(k) + [1 - \sum_{k=0}^{L_j} p_j(k)] L_j$$

(iv) Επαναλαμβάνουμε το βήμα (iii) μέχρι να επιτύχουμε επιθυμητή ακρίβεια για τις ποσότητες A_j.

(v) Υπολογίζουμε δείκτες επίδοσης για τις κατηγορίες με περιορισμούς από την επίλυση των μοντέλων υψηλού επιπέδου της τελευταίας επανάληψης. Για τις κατηγορίες χωρίς περιορισμούς, οι δείκτες επίδοσης υπολογίζονται επιλύοντας το μοντέλο που ορίστηκε στο βήμα (ii) θέτοντας τις τελικές τιμές των A_j.

Άλλες ενδιαφέρουσες ποσότητες που προκύπτουν από την εκτέλεση του αλγορίθμου είναι ο μέσος αριθμός εργασιών της κατηγορίας j σε αναμονή στην ουρά μνήμης (blocked jobs):

$$B_j = \sum_{k=L_j+1}^{N_j} (k - L_j) p_j(k)$$

και ο μέσος αριθμός έτοιμων εργασιών της κατηγορίας j (σε εκτέλεση στο κεντρικό υποσύστημα ή σε αναμονή στην ουρά μνήμης):

$$E_j = A_j + B_j$$

5.5.2 Υποσυστήματα με Περιορισμούς Πληθυσμού

Η μνήμη δεν είναι ο μόνος πόρος του συστήματος που επιβάλλει περιορισμούς πληθυσμού. Στην ουσία, η μνήμη είναι μία ειδική περίπτωση *ταυτόχρονης δέσμευσης πόρων*, ενός γενικότερου φαινομένου που παραβιάζει τις συνθήκες μορφής γινομένου. Η διαφορά ανάμεσα στους περιορισμούς μνήμης και σε γενικότερους περιορισμούς πληθυσμού βρίσκεται στο μοντέλο υψηλού επιπέδου: στην πρώτη περίπτωση, οι πελάτες δεν μοιράζονται άλλους πόρους έξω από το κεντρικό υποσύστημα που υπόκειται στον περιορισμό (το εξωτερικό περιβάλλον αποτελείται μόνο από τερματικά), ενώ στη δεύτερη περίπτωση ισχύει το αντίθετο (π.χ. οι εργασίες μοιράζονται τη χρήση της ΚΜΕ όταν δεν βρίσκονται στο υποσύστημα Ε/Ε που υπόκειται σε περιορισμούς πληθυσμού). Ο προσεγγιστικός αλγόριθμος που περιγράφηκε προηγουμένως για συστήματα πολλών κατηγοριών με περιορισμούς μνήμης μπορεί να γενικευθεί ώστε να καλύπτει και την περίπτωση υποσυστημάτων με περιορισμούς πληθυσμού. Η βασική ιδέα είναι η αντικατάσταση του υποσυστήματος που υπόκειται σε περιορισμούς με R ισοδύναμους σταθμούς, έναν για κάθε κατηγορία.

Υποθέτουμε ότι στο υποσύστημα G του συστήματος κάθε κατηγορία j έχει περιορισμό πληθυσμού L_j ανεξάρτητα από τις άλλες κατηγορίες. (Αν υπάρχουν κατηγορίες χωρίς περιορισμό πληθυσμού, μπορούμε να τις αντιμετωπίσουμε με ενιαίο τρόπο θέτοντας τεχνητούς περιορισμούς που δεν εφαρμόζονται ποτέ.) Μπορούμε να διατυπώσουμε τον ακόλουθο προσεγγιστικό αλγόριθμο [6].

(i) Αγνοώντας τους περιορισμούς πληθυσμού για το υποσύστημα G έχουμε ένα δίκτυο με λύση μορφής γινομένου, το οποίο επιλύουμε. Για κάθε κατηγορία j θέτουμε A_j ίσο με το ελάχιστο μεταξύ του L_j και του μέσου πληθυσμού της κατηγορίας j στο υποσύστημα G, όπως προκύπτει από τη λύση του δικτύου. Οι τιμές αυτές θα χρησιμοποιηθούν ως αρχικές τιμές στην επανάληψη που ακολουθεί.

(ii) Στη συνέχεια κατασκευάζουμε δύο μοντέλα δικτύου αναμονής:

- ένα μοντέλο χαμηλού επιπέδου που παριστάνει το υποσύστημα G σε απομόνωση. Στο μοντέλο αυτό κάθε κατηγορία παριστάνεται ως κατηγορία με σταθερό πληθυσμό ίσο με A_j (κατηγορία τύπου batch), όπου A_j ο μέσος πληθυσμός της κατηγορίας αυτής στο υποσύστημα G.

- ένα μοντέλο υψηλού επιπέδου το οποίο θα περιλαμβάνει τους σταθμούς του υπόλοιπου συστήματος (το εξωτερικό περιβάλλον του υποσυστήματος G) και C ισοδύναμους σταθμούς (έναν για κάθε κατηγορία) με ρυθμούς εξυπηρέτησης εξαρτώμενους από το φορτίο. Κάθε κατηγορία j επισκέπτεται τον δικό της ισοδύναμο σταθμό που παριστάνει το υποσύστημα G. Οι ρυθμοί εξυπηρέτησης των ισοδύναμων σταθμών υπολογίζονται σε κάθε βήμα της επανάληψης.

(iii) Επανάληψη:

(α′) Θεωρούμε το μοντέλο χαμηλού επιπέδου. Για κάθε κατηγορία j:

- Για κάθε δυνατό πληθυσμό $k = 1, \ldots, L_j$ της κατηγορίας j επιλύουμε το μοντέλο θεωρώντας τον πληθυσμό των υπόλοιπων κατηγοριών σταθερό και ίσο με A_j και υπολογίζουμε το ρυθμό απόδοσης $X^j(k)$ της κατηγορίας (διέλευση από τη βραχυκυκλωμένη σύνδεση).
- Ορίζουμε το ρυθμό εξυπηρέτησης του ισοδύναμου σταθμού που επισκέπτεται η κατηγορία j θέτοντας:

$$\mu_j(k) = \begin{cases} X^j(k) & k = 1, \ldots, L_j \\ X^j(L_j) & k > L_j \end{cases}$$

(β′) Επιλύουμε το μοντέλο υψηλού επιπέδου θεωρώντας τους R ισοδύναμους σταθμούς μιας κατηγορίας που ορίστηκαν προηγουμένως. Για τους ισοδύναμους σταθμούς χρησιμοποιούμε τις σχέσεις (4.56) και (4.57) λαμβάνοντας υπόψη το γεγονός ότι κάθε ισοδύναμος σταθμός δέχεται επισκέψεις μόνο από μία κατηγορία (και συνεπώς ο συνολικός αριθμός πελατών στο σταθμό δεν ξεπερνά τον πληθυσμό της αντίστοιχης κατηγορίας). Από την επίλυση του μοντέλου υπολογίζουμε για κάθε κατηγορία j την κατανομή $p_j(k)$ του αριθμού πελατών στον αντίστοιχο ισοδύναμο σταθμό, την οποία χρησιμοποιούμε για να υπολογίσουμε μία νέα τιμή του μέσου πληθυσμού της κατηγορίας j στο υποσύστημα G:

$$A_j = \sum_{k=1}^{L_j} k p_j(k) + [1 - \sum_{k=0}^{L_j} p_j(k)] L_j$$

(iv) Επαναλαμβάνουμε το βήμα (*iii*) μέχρι να επιτύχουμε επιθυμητή ακρίβεια για τις ποσότητες A_j.

(v) Υπολογίζουμε δείκτες επίδοσης από την επίλυση του μοντέλου υψηλού επιπέδου.

Σχετικά με το μέσο αριθμό επισκέψεων εργασιών στο υποσύστημα G, στα μοντέλα υψηλού και χαμηλού επιπέδου ισχύουν οι παρατηρήσεις της Ενότητας 5.4.

5.5.3 Κανονισμοί Εξυπηρέτησης με Προτεραιότητες

Η ύπαρξη προτεραιοτήτων στον κανονισμό εξυπηρέτησης δεν επιτρέπει τη λύση σε μορφή γινομένου. Θα αναπτύξουμε μία απλή προσεγγιστική τεχνική για συστήματα με προτεραιότητες στη χρονοδρομολόγηση της ΚΜΕ. Υποθέτουμε απόλυτη προτεραιότητα, δηλαδή ότι μία εργασία υψηλής προτεραιότητας που φθάνει στην ΚΜΕ διακόπτει την εξυπηρέτηση μιας εργασίας χαμηλής προτεραιότητας για να εξυπηρετηθεί.

Έστω ότι υπάρχουν C κατηγορίες εργασιών, τις οποίες αριθμούμε με αύξουσα προτεραιότητα. Θα προσεγγίσουμε τον χρόνο παραμονής των εργασιών της κατηγορίας j στην ΚΜΕ, θεωρώντας διαδοχικά την επίδραση εργασιών με χαμηλότερη, ίση και υψηλότερη προτεραιότητα.

- Χαμηλότερη προτεραιότητα (κατηγορίες 1 έως $j-1$). Εφόσον έχουμε απόλυτη προτεραιότητα οι εργασίες των κατηγοριών 1 έως $j-1$ δεν έχουν καμία επίδραση στον χρόνο παραμονής των εργασιών της κατηγορίας j.

- Ίση προτεραιότητα (κατηγορία j). Μεταξύ εργασιών της ίδιας κατηγορίας ακολουθείται κανονισμός FIFO, άρα η επίδραση των εργασιών ίσης προτεραιότητας στον χρόνο παραμονής μιας εργασίας της κατηγορίας j μπορεί να εκφραστεί με βάση το θεώρημα των αφίξεων:

$$R_{\mathrm{CPU},j}(\boldsymbol{N}) \simeq D_{\mathrm{CPU},j}[1 + Q_{\mathrm{CPU},j}(\boldsymbol{N} - \mathbf{1}_j)]$$

υποθέτοντας κλειστό δίκτυο. (Αγνοούμε προς το παρόν την επίδραση των κατηγοριών υψηλότερης προτεραιότητας.)

- Υψηλότερη προτεραιότητα (κατηγορίες $j+1$ έως C). Μπορούμε να περιλάβουμε την επίδραση εργασιών υψηλότερης προτεραιότητας διαστέλλοντας τον χρόνο εξυπηρέτησης των εργασιών της κατηγορίας j στη CPU κατά τον παράγοντα $1-\sum_{k=j+1}^{C} U_{\mathrm{CPU},k}(\boldsymbol{N})$, ο οποίος εκφράζει το ποσοστό του χρόνου, κατά το οποίο η ΚΜΕ διατίθεται στις εργασίες της κατηγορίας j. Έτσι, μία τελική προσέγγιση που λαμβάνει υπόψη όλες τις επιδράσεις θα είναι:

$$R_{\mathrm{CPU},j}(\boldsymbol{N}) \simeq \frac{D_{\mathrm{CPU},j}[1+Q_{\mathrm{CPU},j}(\boldsymbol{N}-\mathbf{1}_j)]}{1-\sum_{k=j+1}^{C} U_{\mathrm{CPU},k}(\boldsymbol{N})} \tag{5.19}$$

Θα μπορούσαμε να αναπτύξουμε μία παραλλαγή του αλγορίθμου MVA ενσωματώνοντας την Εξίσωση (5.19) για τον χρόνο παραμονής. Εντούτοις, αντί να τροποποιούμε κάθε φορά τους βασικούς αλγορίθμους, είναι προτιμότερο να τους χρησιμοποιούμε όπως έχουν αναπτύσσοντας νέους αλγορίθμους σε υψηλότερο επίπεδο. Στην περίπτωση των προτεραιοτήτων μπορούμε να επιλύσουμε το μοντέλο με βάση τη σχέση (5.19) εφαρμόζοντας την ακόλουθη τεχνική: αντικαθιστούμε την ΚΜΕ με C σταθμούς κανονισμού FIFO, έτσι ώστε κάθε κατηγορία να επισκέπτεται μόνο έναν από αυτούς τους σταθμούς [7]. Η λειτουργία της μεθόδου περιγράφεται με τη βοήθεια του ακόλουθου αλγορίθμου.

(i) Αντικαθιστούμε την ΚΜΕ με C εικονικές ΚΜΕ κανονισμού *FIFO*, έτσι ώστε σε κάθε εικονική ΚΜΕ να εξυπηρετείται μόνο μία κατηγορία. Θέτουμε αρχικά τον ρυθμό απόδοσης κάθε κατηγορίας j ίσο με μηδέν, $X^j = 0$.

(ii) Επανάληψη.

(α′) Υπολογίζουμε τον βαθμό χρησιμοποίησης της ΚΜΕ για κάθε κατηγορία j:

$$U_{\mathrm{CPU},j} = X^j D_{\mathrm{CPU},j}$$

όπου $D_{\mathrm{CPU},j}$ είναι η 'πραγματική' απαίτηση εξυπηρέτησης της κατηγορίας j στην ΚΜΕ.

(β′) Επιλύουμε το μοντέλο με τις εικονικές ΚΜΕ χρησιμοποιώντας ανάλυση μέσης τιμής και θέτοντας την απαίτηση εξυπηρέτησης της κατηγορίας j στην αντίστοιχη εικονική ΚΜΕ ίση με τη διεσταλμένη τιμή

$$D_{\mathrm{CPU},j} = \frac{D_{\mathrm{CPU},j}}{1-\sum_{k=j+1}^{C} U_{\mathrm{CPU},k}}$$

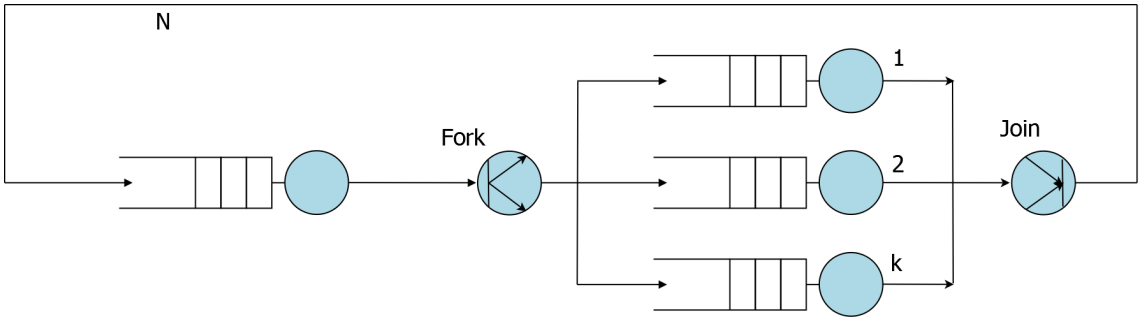
(iii) Επαναλαμβάνουμε το βήμα (ii) μέχρι να επιτύχουμε επιθυμητή ακρίβεια για τους ρυθμούς απόδοσης X^j.

(iv) Από την τελευταία επανάληψη προκύπτουν δείκτες επίδοσης για το σύστημα συνολικά και για τους σταθμούς εκτός της ΚΜΕ. Ο χρόνος παραμονής και ο μέσος αριθμός εργασιών στην ΚΜΕ για κάθε κατηγορία δίνονται άμεσα από τις εικονικές ΚΜΕ. (Είναι φανερό ότι η (5.19) ισχύει για κάθε εικονική ΚΜΕ.) Ο βαθμός χρησιμοποίησης κάθε κατηγορίας προκύπτει από τη σχέση $U_{\mathrm{CPU},j} = X^j D_{\mathrm{CPU},j}$ (θεωρώντας την αρχική τιμή της απαίτησης εξυπηρέτησης).

Οι εικονικές ΚΜΕ αναφέρονται στη βιβλιογραφία ως *σκιώδεις* (shadow CPU) [10]. Η συνολική προσέγγιση αναφέρεται ως *σταδιακή εισαγωγή των κατηγοριών* (stepwise inclusion of classes, SWIC).

5.5.4 Σταθμοί *FIFO* με Χρόνους Εξυπηρέτησης Εξαρτώμενους από την Κατηγορία

Αν σε ένα σταθμό κανονισμού FIFO (Τύπος 1 του μοντέλου BCMP) έχουμε διαφορετικό χρόνο εξυπηρέτησης ανά επίσκεψη για κάθε κατηγορία, παραβιάζονται οι προϋποθέσεις για λύση μορφής γινομένου.



Σχήμα 5.5: *Αρχιτεκτονική Fork–Join.*

Μπορούμε να αναπτύξουμε μία απλή προσεγγιστική τεχνική βασισμένη στον αλγόριθμο MVA τροποποιώντας κατάλληλα την εξίσωση του χρόνου παραμονής. Η αρχική μορφή της εξίσωσης είναι:

$$R_{ij}(\boldsymbol{N}) = D_{ij}\left[1 + \sum_{k=1}^{C} Q_{ik}(\boldsymbol{N} - \mathbf{1}_j)\right] = v_{ij}\left[S_{ij} + S_{ij}\sum_{k=1}^{C} Q_{ik}(\boldsymbol{N} - \mathbf{1}_j)\right]$$

Εφόσον όλες οι κατηγορίες πρέπει να έχουν τον ίδιο μέσο χρόνο εξυπηρέτησης ανά επίσκεψη, μπορούμε να φανταστούμε ότι η πιο πάνω εξίσωση είναι ειδική περίπτωση της εξίσωσης

$$R_{ij}(\boldsymbol{N}) = v_{ij}\left[S_{ij} + \sum_{k=1}^{C} S_{ik}Q_{ik}(\boldsymbol{N} - \mathbf{1}_j)\right]$$

την οποία μπορούμε να χρησιμοποιήσουμε στον αλγόριθμο της μέσης τιμής. Η τροποποίηση αυτή έχει απλή φυσική ερμηνεία και δίνει πολύ ικανοποιητικά αποτελέσματα.

5.5.5 Δίκτυα Fork–Join

Ο μηχανισμός Fork–Join μπορεί να παραστήσει φαινόμενα ταυτοχρονισμού και παραλληλίας σε υπολογιστικά συστήματα. Όταν μια εργασία εισέρχεται σε στάδιο ταυτοχρονισμού, διακλαδίζεται σε υπο-εργασίες (fork), οι οποίες εκτελούνται αναξάρτητα στον ίδιο ή σε διαφορετικούς εξυπηρετητές. Όταν ολοκληρωθεί η εκτέλεσή της, κάθε υπο-εργασία περιμένει στο σημείο συνένωσης (join) μέχρι να τελειώσουν όλες οι συγγενείς υπο-εργασίες. Οι σταθμοί Fork–Join περιγράφουν τους περιορισμούς συγχρονισμού μεταξύ υπο-εργασιών κατά την εκτέλεση και είναι απαραίτητα στοιχεία για την ανάλυση διαφόρων υπολογιστικών και τηλεπικοινωνιακών συστημάτων, όπως είναι για παράδειγμα οι συστοιχίες δίσκων με σύγχρονα φορτία. Όμως, τα μοντέλα δικτύων αναμονής με δομές Fork–Join δεν ικανοποιούν τις προϋποθέσεις για λύση σε μορφή γινομένου. Συνεπώς, αφού η ακριβής λύση είναι συνήθως δυσχερής, αναζητούνται προσεγγιστικές τεχνικές. Θα περιγράψουμε στη συνέχεια μια προσέγγιση για μοντέλα αναμονής που περιλαμβάνουν συγχρονισμό Fork–Join [10].

Θα θεωρήσουμε κλειστό δίκτυο μιας κατηγορίας, το οποίο περιλαμβάνει διασυνδεδεμένα υποσυστήματα δύο τύπων (Σχήμα 5.5):

(i) *Σειριακό υποσύστημα*, αποτελούμενο από έναν συνήθη σταθμό αναμονής.

(ii) *Παράλληλο υποσύστημα*, αποτελούμενο από k, $k > 1$, παράλληλους σταθμούς αναμονής σε διάταξη Fork–Join, δηλαδή κάθε εργασία που υποβάλλεται στο υποσύστημα διασπάται σε k υπο-εργασίες, μία για κάθε επιμέρους σταθμό. Καθώς η εργασία ολοκληρώνεται όταν ολοκληρωθούν όλες οι παράλληλες υπο-εργασίες, ο συνολικός χρόνος εκτέλεσης θα είναι το μέγιστο των χρόνων που απαιτήθηκαν για εξυπηρέτηση σε καθέναν από τους k παράλληλους σταθμούς.

Θα διατυπώσουμε μια προσεγγιστική μέθοδο MVA προσαρμόζοντας τη βασική σχέση της ανάλυσης (υπολογισμός χρόνου απόκρισης σύμφωνα με το Θεώρημα των Αφίξεων), ώστε να ενσωματώνει την αρχή

λειτουργίας του μηχανισμού Fork–Join. Έστω ένα παράλληλο υποσύστημα, το οποίο παριστάνεται ως ο σταθμός i του δικτύου και περιλαμβάνει k_i πανομοιότυπους παράλληλους σταθμούς. Υποθέτουμε ότι ο μέσος χρόνος εξυπηρέτησης είναι ο ίδιος σε καθέναν από τους παράλληλους σταθμούς. Επιπλέον, θα υποθέσουμε ότι οι ως άνω χρόνοι εξυπηρέτησης (ανά επίσκεψη) είναι εκθετικά κατανεμημένοι με μέση τιμή S_i. Ο χρόνος απόκρισης μπορεί να προσεγγιστεί από την εξίσωση:

$$R_i(N) = v_i S_i[H_{k_i} + Q_i(N-1)] \tag{5.20}$$

όπου το όρισμα στην παρένθεση αφορά ως συνήθως τον πληθυσμό του δικτύου και H_{k_i} είναι ο αρμονικός αριθμός τάξης k_i:

$$H_{k_i} = \sum_{j=1}^{k_i} \frac{1}{j}$$

Η παρουσία του αρμονικού αριθμού εκφράζει την επίδραση του συγχρονισμού Fork–Join: Μπορεί εύκολα να αποδειχθεί ότι η μέση τιμή μιας τυχαίας μεταβλητής που είναι το μέγιστο k αναξάρτητων τυχαίων μεταβλητών εκθετικά κατανεμημένων με μέση τιμή S είναι ίση με SH_k. Η παραπάνω σχέση θα εφαρμοστεί για όλους τους σταθμούς i του δικτύου που παριστάνουν παράλληλα υποσυστήματα. Οι λοιπές σχέσεις του αλγορίθμου MVA δεν μεταβάλλονται.

Θα πρέπει να παρατηρήσουμε ότι η μέθοδος αυτή βασίζεται σε μια «απαισιόδοξη» εκτίμηση του χρόνου απόκρισης. Πράγματι, πέραν του συγχρονισμού στην έξοδο, η μέθοδος ουσιαστικά υποθέτει την ύπαρξη συγχρονισμού και στην είσοδο του σταθμού, εφόσον οι επιμέρους ταυτόχρονες υπο-εργασίες μιας εργασίας αρχίζουν να εκτελούνται μαζί, αντί να εισέρχεται καθεμιά στην αντίστοιχη ουρά αναμένοντας εκτέλεση. (Μπορούμε να πούμε ότι η προσέγγιση ισοδυναμεί με την περίπτωση μιας κοινής ουράς αναμονής μπροστά στους k παράλληλους σταθμούς.) Επομένως, στα παράλληλα υποσυστήματα, η προσέγγιση αυτή λαμβάνει υπόψη έναν χρόνο απόκρισης μεγαλύτερο από τον πραγματικό. Επιπλέον, η υπόθεση εκθετικά κατανεμημένων χρόνων οδηγεί σε υψηλή σχετική εκτίμηση του μεγίστου (αρμονικός αριθμός) λόγω της μεγάλης διασποράς της εκθετικής κατανομής. Παρ' όλα αυτά, η μέθοδος έχει ικανοποιητική επίδοση —ειδικά σε καταστάσεις ελαφρού φορτίου— και αποτελεί χαρακτηριστικό παράδειγμα προσέγγισης που εντάσσεται με φυσικό τρόπο στο βασικό μοντέλο της Ανάλυσης Μέσης Τιμής.

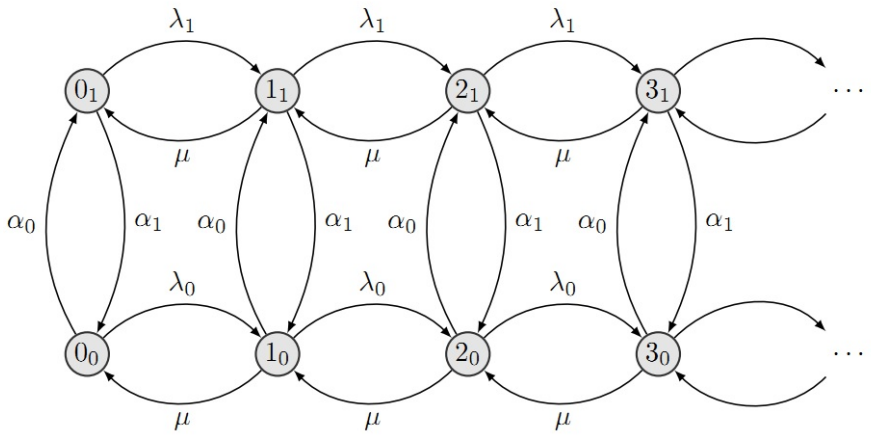
5.6 Ειδικά Χαρακτηριστικά Υπηρεσιών Ιστού

Οι *υπηρεσίες ιστού* (Web services) αποτελούν εφαρμογές με ιδιαίτερα χαρακτηριστικά, τα οποία λαμβάνονται υπόψη κατά την ανάπτυξη εξειδικευμένων μοντέλων [9, 10].

5.6.1 Εκρηκτικότητα

Ένα πρώτο χαρακτηριστικό είναι η *εκρηκτικότητα* (burstiness) του φορτίου σε μεγάλα κατανεμημένα συστήματα όπως το Διαδίκτυο, ο Παγκόσμιος Ιστός και τα εταιρικά δίκτυα (intranets). Διάφορες μελέτες έχουν δείξει ότι η συνολική κίνηση (σε πακέτα ανά sec ή αιτήσεις http ανά sec) χαρακτηρίζεται από διαστήματα αιχμής με υψηλή μεταβλητότητα ρυθμών. Πράγματι, οι τιμές αιχμής του ρυθμού αφίξεων μπορούν να υπερβούν και κατά δέκα φορές τον μέσο ρυθμό. Η εκρηκτικότητα αυτή μπορεί να επιβαρύνει την επίδοση αν δεν ληφθεί υπόψη. Θα πρέπει, επομένως, να περιληφθεί στον χαρακτηρισμό φορτίου και στα αντίστοιχα μοντέλα. Το φαινόμενο μπορεί να παρασταθεί με διάφορους τρόπους, είτε με επιχειρησιακή είτε με μαθηματική μοντελοποίηση.

Στην περίπτωση επιχειρησιακής ανάλυσης, η εκρηκτικότητα περιγράφεται συνήθως με τη βοήθεια παραμέτρων που προσδιορίζονται από παρατηρήσεις σε δεδομένο διάστημα λειτουργίας του συστήματος. Τέτοιες παράμετροι είναι: (α) ο λόγος του μέγιστου παρατηρηθέντος ρυθμού αφίξεων προς τον μέσο ρυθμό αφίξεων, (β) το ποσοστό του χρόνου κατά το οποίο ο στιγμιαίος ρυθμός αφίξεων υπερβαίνει τον μέσο ρυθμό αφίξεων. Εν συνεχεία,η επίδραση της εκρηκτικότητας στον ρυθμό απόδοσης του συστήματος αποτιμάται μέσω της κατάλληλης ενσωμάτωσης των παραπάνω παραμέτρων στην απαίτηση εξυπηρέτησης των αιτήσεων (διαστολή του χρόνου εξυπηρέτησης) [10].



Σχήμα 5.6: *Μαρκοβιανή ουρά με (εκρηκτικές) αφίξεις MMPP.*

Η χρονική μεταβολή του ρυθμού αφίξεων μπορεί να παρασταθεί και με τη χρήση μαρκοβιανής διαδικασίας. Ας υποθέσουμε ότι η διαδικασία των αφίξεων εναλλάσσεται μεταξύ δύο καταστάσεων: υψηλού ρυθμού λ_1 (κατάσταση 1) και χαμηλού ρυθμού λ_0 (κατάσταση 0). Το διάστημα που περνά η διαδικασία στην κατάσταση υψηλού (χαμηλού) ρυθμού αφίξεων ακολουθεί εκθετική κατανομή με παράμετρο α_1 (α_0). Πρόκειται, επομένως, για *χρονικά μεταβαλλόμενη διαδικασία Poisson* ή *διαδικασία Poisson με μαρκοβιανή διαμόρφωση* (Markov-modulated Poisson process, MMPP) [5]. Είναι φανερό ότι μια τέτοια διαδικασία αντιστοιχεί σε μια *κατανομή φάσεων*.

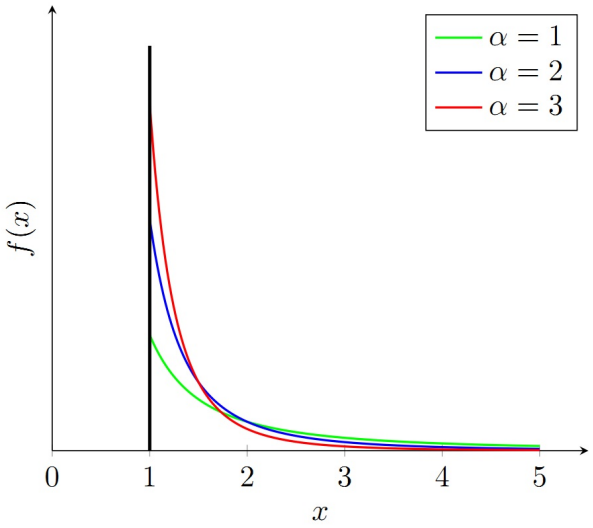
Η λειτουργία ενός σταθμού αναμονής που δέχεται αφίξεις εναλλασσόμενου ρυθμού όπως παραπάνω και εκθετικούς χρόνους εξυπηρέτησης σταθερού ρυθμού μ μπορεί να παρασταθεί ως αλυσίδα Markov δύο (άπειρων) επιπέδων που θα αντιστοιχούν στις δύο καταστάσεις των αφίξεων, όπως φαίνεται στο Σχήμα 5.6. Το μοντέλο αυτό χαρακτηρίζεται από αρκετά περίπλοκες εξισώσεις ισορροπίας σε σχέση με τα απλά συστήματα αναμονής που έχουμε εξετάσει και η επίλυσή του συνήθως βασίζεται σε αριθμητικές τεχνικές.

5.6.2 Κατανομές Αρχείων

Ένα δεύτερο χαρακτηριστικό αφορά το γεγονός ότι ένα μεγάλο ποσοστό αιτήσεων http αφορά μικρά έγγραφα, ενώ ένα μικρό ποσοστό αφορά έγγραφα που είναι μερικές τάξεις μεγαλύτερα από τα προηγούμενα. Αποτέλεσμα αυτού είναι ότι ένα ελάχιστο μέρος των εργασιών αντιπροσωπεύει ένα πολύ μεγάλο μέρος του συνολικού φορτίου. Διάφορες μελέτες έχουν δείξει ότι τα μεγέθη των αρχείων που διακινούνται ακολουθούν *κατανομές βαριάς ουράς* (heavy-tailed distributions). Οι κατανομές αυτές έχουν την ιδιότητα ότι η *συμπληρωματική αθροιστική συνάρτηση κατανομής πιθανότητας* (Complementary Cumulative Distribution Function — CCDF) $P[X > x] = 1 - F(x)$ απομειώνεται βραδύτερα από την αντίστοιχη της εκθετικής κατανομής.

Η ιδιότητα αυτή χαρακτηρίζει πολλά φαινόμενα στην πράξη. Μια σχετική κατανομή είναι ο *νόμος του Zipf*, ο οποίος προβλέπει ότι η συχνότητα ζήτησης ενός αντικειμένου είναι αντιστρόφως ανάλογη προς τη θέση του στη σειρά που προκύπτει, αν τα αντικείμενα διαταχθούν σύμφωνα με τη συχνότητα ζήτησης. Ο νόμος αυτός χρησιμοποιείται για να χαρακτηριστεί η «δημοφιλία» και εφαρμόστηκε αρχικά για να περιγράψει τη σχέση ανάμεσα στις λέξεις ενός κειμένου και τη συχνότητα χρήσης τους. Ανάλογη λειτουργία έχει και ο παλιός *κανόνας 80-20* που εκτιμά ότι, σε πολλά φαινόμενα, περίπου το 80% των αποτελεσμάτων προέρχεται από το 20% των αιτίων.

Μια οικογένεια κατανομών που χαρακτηρίζεται από συμπεριφορά βαριάς ουράς είναι οι *κατανομές ή νόμοι δύναμης* power-laws, που ακολουθούν τη μορφή $f(x) \propto x^{\alpha}$. Ειδική περίπτωση αποτελεί η κατανομή Pareto, για την οποία ισχύει $P[X > x] = x^{-\alpha}$, δηλαδή ο αριθμός των στιγμιοτύπων που είναι μεγαλύτερα από x είναι αντίστροφος μιας δύναμης του x (Σχήμα 5.7). Εμπειρικές μελέτες έχουν δείξει ότι τα μεγέθη των αρχείων σε διαδικτυακούς τόπους μπορούν να παρασταθούν από κατανομή Pareto με $alpha = 1,1$.



Σχήμα 5.7: *Κατανομή Pareto.*

Για παράδειγμα, με την τιμή αυτή του α, προκύπτει ότι στο 1% των εργασιών αντιστοιχεί περίπου το 50% του φορτίου. Χαρακτηριστικά παρατηρούμε ότι σε εκθετική κατανομή με την ίδια μέση τιμή, το 1% των εργασιών αφορά μόλις το 5% του συνολικής απαίτησης εξυπηρέτησης.

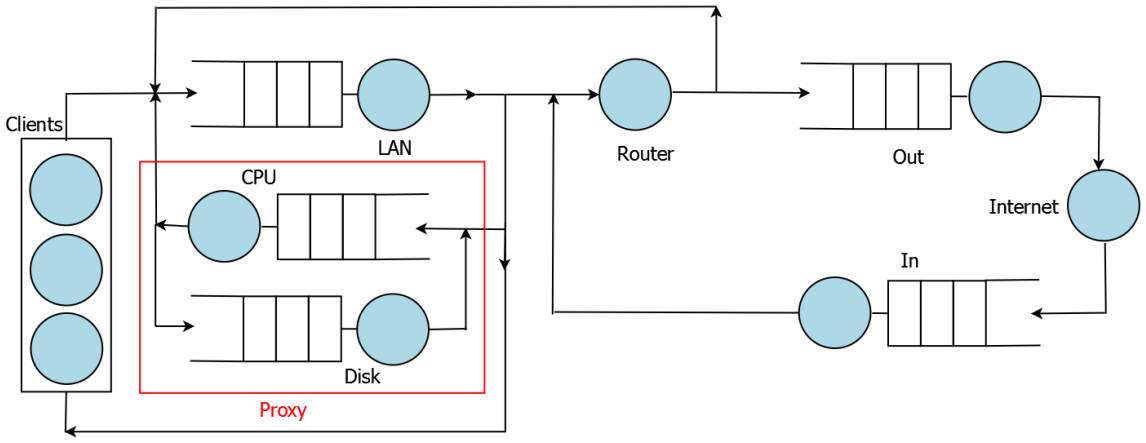
Σε μεγάλη έκταση, οι κατανομές δύναμης χαρακτηρίζουν τη συμπεριφορά των χρηστών στον Παγκόσμιο Ιστό. Οι κατανομές αυτές μπορούν να χρησιμοποιηθούν για να περιγράψουν το φορτίο (χρόνοι εξυπηρέτησης) στα μοντέλα των δικτύων αναμονής. Πράγματι, όπως γνωρίζουμε, μπορούμε να χρησιμοποιήσουμε γενικές κατανομές θεωρώντας ότι αυτές μπορούν να προσεγγιστούν στον επιθυμητό βαθμό από σύνθετες κατανομές σταδίων. Εξάλλου, είναι σαφές ότι η μεταβλητότητα στο μέγεθος των ζητούμενων αρχείων δεν μπορεί να περιγραφεί από μια ενιαία κατανομή. Είναι, όμως, σχετικά εύκολο να οριστούν πολλές κατηγορίες χρηστών (αιτήσεων) που θα αντιστοιχούν στις διαφορετικές περιοχές τιμών του μεγέθους των αρχείων.

5.6.3 Τεχνικές Μεσολάβησης

Με στόχο τη βελτίωση της επίδοσης συστημάτων που παρέχουν υπηρεσίες Ιστού, έχουν αναπτυχθεί διάφορες τεχνικές. Οι τεχνικές αυτές επιτυγχάνουν μείωση της ζήτησης των δημοφιλών αρχείων και —γενικότερα— μείωση του χρόνου πρόσβασης στα αρχεία και μείωση του απαιτούμενου εύρους ζώνης για μεταφορά των δεδομένων. Επίσης, συμβάλλουν στην ενίσχυση του συστήματος από πλευράς ασφάλειας.

Ένας *εξυπηρετητής μεσολάβησης* (proxy server) είναι ένας ειδικός τύπος εξυπηρετητή Ιστού, ο οποίος μπορεί να λειτουργεί τόσο ως εξυπηρετητής όσο και ως πελάτης. Ο μεσολαβητής δέχεται αιτήματα από τους πελάτες και τα προωθεί στους εξυπηρετητές Ιστού. Αντίστοιχα, όταν λαμβάνει απάντηση από απομακρυσμένους εξυπηρετητές, την προωθεί στους πελάτες. Αρχικά, οι μεσολαβητές παρείχαν πρόσβαση στον Ιστό για χρήστες ιδιωτικών δικτύων, οι οποίοι μπορούσαν να προσπελάσουν το Διαδίκτυο μόνο μέσω *τείχους προστασίας* (firewall).

Η λειτουργία του εξυπηρετητή μεσολάβησης δεν περιορίζεται στον ρόλο του αντιπροσώπου που αναμεταδίδει μηνύματα. Οι μεσολαβητές Ιστού μπορούν να αποθηκεύουν τις αναμεταδιδόμενες απαντήσεις με μηχανισμούς *λανθάνουσας μνήμης* (caching), συμβάλλοντας έτσι στη βελτίωση της επίδοσης του συστήματος. Η βασική αρχή λειτουργίας είναι η εξής: τα αρχεία που αναζητούνται συχνά αποθηκεύονται σε τοπικά αντίγραφα για μελλοντική χρήση, ώστε να μη χρειάζεται η ανάκτησή τους από τον εξυπηρετητή την επόμενη φορά που θα ζητηθούν. Η τεχνική λανθάνουσας μνήμης μειώνει τον χρόνο προσπέλασης φέρνοντας τα δεδομένα όσο γίνεται πιο κοντά στους πελάτες. Αυτό έχει ως αποτέλεσμα τη μείωση της κίνησης στο δίκτυο και την αύξηση της διαθεσιμότητας του εξυπηρετητή. Συγχρόνως, όμως, ο μηχανισμός αυτός παρουσιάζει και διάφορα προβλήματα, που σχετίζονται με την επιλογή των αρχείων που θα αποθηκευτούν και —κυρίως— με την επικαιρότητα των αποθηκευμένων αρχείων.



Σχήμα 5.8: *Μοντέλο κλειστού δικτύου (πλευρά πελατών) με εξυπηρετητή μεσολάβησης.*

Η τεχνική της λανθάνουσας μνήμης είναι ευρύτατα διαδεδομένη στον Παγκόσμιο Ιστό και εφαρμόζεται με διάφορους τρόπους. Από την πλευρά του πελάτη, ο φυλλομετρητής (browser) διατηρεί στον τοπικό δίσκο αντίγραφα σελίδων τις οποίες επισκεφθηκε πρόσφατα ο χρήστης (π.χ. όταν πιέζεται το πλήκτρο «back»). Μια άλλη εφαρμογή της μεσολάβησης, είναι η περίπτωση ενός εξυπηρετητή με λανθάνουσα μνήμη, ο οποίος τοποθετείται στο δίκτυο ενδιάμεσα στη διαδρομή μεταξύ μιας κοινότητας πελατών και ενός συνόλου εξυπηρετητών (π.χ. σε ένα εταιρικό ή πανεπιστημιακό δίκτυο, ή σε έναν πάροχο υπηρεσιών Διαδικτύου). Όταν κάποιος πελάτης ζητάει ένα αρχείο, ο μεσολαβητής λειτουργεί ως πελάτης του Ιστού και ζητάει με τη σειρά του το αρχείο από κάποιον απομακρυσμένο εξυπηρετητή, για να το επιστρέψει εν συνεχεία στον αρχικό πελάτη. Επόμενες αιτήσεις για το ίδιο αρχείο ικανοποιούνται από το αντίγραφο που αποθηκεύεται στη λανθάνουσα μνήμη (cache memory) του μεσολαβητή. Η αποδοτικότητα της λανθάνουσας μνήμης εκφράζεται συνήθως με το *ποσοστό ευστοχίας* (hit ratio), δηλαδή τον λόγο του αριθμού των αιτήσεων που ικανοποιήθηκαν από τη λανθάνουσα μνήμη προς τον συνολικό αριθμό αιτήσεων. Συμπληρωματικά χρησιμοποιείται το *ποσοστό αστοχίας* (miss ratio).

Μια άλλη συναφής μέθοδος κατανομής των δεδομένων είναι ο *αντικατοπτρισμός* (mirroring), δηλαδή η δημιουργία πανομοιότυπων (κατοπτρικών) αντιγράφων του περιεχομένου ενός εξυπηρετητή σε άλλους εξυπηρετητές. Η τεχνική αυτή απαιτεί τακτική ενημέρωση του περιεχομένου (μέσω Διαδικτύου), ώστε τα αντίγραφα να είναι συνεπή, και αναδρομολόγηση των αιτημάτων προς τους δευτερεύοντες εξυπηρετητές, όταν ο πρωτεύων είναι απασχολημένος. Ο αντικατοπτρισμός βελτιώνει την ποιότητα των υπηρεσιών αυξάνοντας τη διαθεσιμότητα του συστήματος και ενισχύοντας την ισορροπία του φορτίου.

5.6.3.1 Μοντελοποίηση Ιστού με Μεσολάβηση

Οι τεχνικές μεσολάβησης μπορούν εύκολα να ενσωματωθούν στα βασικά μοντέλα συστημάτων Ιστού που παρουσιάστηκαν στο προηγούμενο κεφάλαιο και παριστάνουν την αρχιτεκτονική συστημάτων από τις οπτικές γωνίες του πελάτη και του εξυπηρετητή [9, 10].

Στο Σχήμα 5.8 φαίνεται η αρχιτεκτονική δικτύου αναμονής του Σχήματος 4.8, στην οποία έχει προστεθεί ένας εξυπηρετητής μεσολάβησης με λανθάνουσα μνήμη. Ο μεσολαβητής στο παράδειγμα του σχήματος περιλαμβάνει επεξεργαστή και έναν δίσκο (σταθμοί αναμονής). Τα λοιπά στοιχεία (σταθμοί εργασίας πελατών, τοπικό δίκτυο, δρομολογητής, συνδέσεις, Διαδίκτυο) θεωρούνται παρόμοια στις δύο αρχιτεκτονικές και μοντελοποιούνται με τον ίδιο τρόπο.

Έστω p_{hit} το ποσοστό ευστοχίας των αιτήσεων των πελατών στη λανθάνουσα μνήμη. Θα εξετάσουμε πώς τροποποιούνται οι τιμές της απαίτησης εξυπηρέτησης σε σχέση με τις αντίστοιχες τιμές στο σύστημα χωρίς μεσολαβητή.

Οι αιτήσεις θα προωθούνται προς το Διαδίκτυο μόνο στην περίπτωση αστοχίας. Συνεπώς, η νέα τιμή της απαίτησης εξυπηρέτησης θα είναι

$$D_i' = (1 - p_{\text{hit}})D_i$$

όπου i αναφέρεται στον δρομολογητή, τους συνδέσμους και το Διαδίκτυο. Όσον αφορά τη χρήση του τοπικού δικτύου, σε περίπτωση επιτυχίας ο χρόνος είναι ίδιος όπως προηγουμένως. Στην περίπτωση αποτυχίας, όμως, η αίτηση του πελάτη πραγματοποιεί διπλάσιες διαδρομές στο τοπικό δίκτυο. Συνεπώς, η απαίτηση θα είναι

$$D'_{\mathrm{NET}} = p_{\mathrm{hit}} D_{\mathrm{NET}} + (1 - p_{\mathrm{hit}}) 2 D_{\mathrm{NET}} = (2 - p_{\mathrm{hit}}) D_{\mathrm{NET}}$$

Γενικά, για να είναι συμφέρουσα η χρήση μεσολαβητή, η επιτάχυνση λόγω ευστοχίας θα πρέπει να είναι μια τάξη μεγέθους μεγαλύτερη από την επιβράδυνση λόγω αστοχίας.

Ανάλογα μπορούμε να θεωρήσουμε την επέκταση του μοντέλου της πλευράς του εξυπηρετητή (ανοικτό δίκτυο) εφαρμόζοντας την τεχνική του αντικατοπτρισμού. Πράγματι, η αρχιτεκτονική του Σχήματος 4.10 μπορεί να γενικευθεί με χρήση n εξυπηρετητών που περιέχουν πανομοιότυπα αντίγραφα του συνόλου των αρχείων. Κάθε εξυπηρετητής είναι συνδεδεμένος απευθείας στο τοπικό δίκτυο και περιλαμβάνει ΚΜΕ και έναν δίσκο. Υποθέτουμε ότι σε μια τέτοια διάταξη το φορτίο θα ισοκατανέμεται μεταξύ των εξυπηρετητών. Υποθέτοντας ότι τα λοιπά χαρακτηριστικά των δύο μοντέλων εναι όμοια, θα εξετάσουμε πώς τροποποιούνται οι χρόνοι εξυπηρέτησης σε σχέση με τις αντίστοιχες τιμές στο σύστημα του απλού εξυπηρετητή χωρίς αντικατοπτρισμό. Εύκολα διαπιστώνουμε ότι μόνο οι απαιτήσεις εξυπηρέτησης της ΚΜΕ και των δίσκων του εξυπηρετητή χρειάζονται τροποποίηση. Πράγματι, η πιθανότητα να επισκεφθεί μια εργασία έναν συγκεκριμένο εξυπηρετητή i είναι ίση με $1/n$. Αν j είναι μια συνιστώσα (π.χ. επεξεργαστής, δίσκος) του εξυπηρετητή i, τότε ο μέσος αριθμός επισκέψεων σε αυτήν θα είναι $v'_j = v_j/n$ όπου v_j ο μέσος αριθμός επισκέψεων στην περίπτωση απλού εξυπηρετητή. (Θεωρήσαμε μια κατηγορία εργασιών για απλούστευση του συμβολισμού.) Εφόσον ο μέσος χρόνος εξυπηρέτησης ανά επίσκεψη S_i δεν αλλάζει, η συνολική απαίτηση εξυπηρέτησης στη συνιστώσα j θα είναι

$$D'_j = v'_j S_j = v_j S_j / n = D_j / n$$

Μια άλλη επιλογή θα ήταν ο συνδυασμός των n εξυπηρετητών Ιστού με έναν εξυπηρετητή αρχείων (file server), ο οποίος θα περιέχει ΚΜΕ και m δίσκους. Στην περίπτωση αυτή, που δεν περιλαμβάνει αντικατοπτρισμό, οι εξυπηρετητές θα μοιράζονται το σύστημα αρχείων. Η διάταξη αυτή είναι απαλλαγμένη από το πρόβλημα της ενημέρωσης των κατοπτρικών αντιγράφων, αλλά συνεπάγεται αυξημένη απασχόληση του τοπικού δικτύου και του εξυπηρετητή αρχείων.

Βιβλιογραφία

[1] Bard, Y., *Some Extensions to Multiclass Queueing Network Analysis*, Proceedings of the 4th International Symposium on Modelling and Performance Evaluation of Computer Systems: Performance of Computer Systems (North-Holland Publishing Co.): 51–62, 1979.

[2] Bolch, G., Greiner, S., De Meer, H., and Trivedi, K.S., *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, Wiley-Interscience, 2006.

[3] Courtois, P.-J., *Decomposability: Queueing and Computer System Applications*, Academic Press,1977.

[4] Gelenbe, E., *On Approximate Computer System Models*, Journal of the ACM, Vol. 22, No. 2, pp. 261–269, 1975.

[5] Harchol-Balter, M., *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, Cambridge University Press, 2013.

[6] Lazowska, E.D. and Zahorjan, J., *Multiple Class Memory Constrained Queueing Networks*, ACM SIGMETRICS Conference on Measurement and Modelling of Computer Systems, Seattle, Washington, Aug. 1982.

[7] Lazowska, E.D., Zahorjan, J., Scott Graham, G. and Sevcik, K.C., *Quantitative System Performance - Computer System Analysis Using Queueing Network Models*, Prentice-Hall, 1984.

[8] MacNair, E.A. and Sauer, C.H., *Elements of Practical Performance Modeling*, Prentice-Hall, 1985.

[9] Menasce, D.A., and Almeida, V.A.F., *Capacity Planning for Web Services: Metrics, Models and Methods*, Prentice-Hall, 2002.

[10] Menasce, D.A., Almeida, V.A.F., and Dowdy, L.W., *Performance by Design, Computer Capacity Planning by Example*, Prentice-Hall PTR, 2004.

[11] Schweitzer, P., *Approximate analysis of multiclass closed networks of queues*, Proceedings of International Conference on Stochastic Control and Optimization, 1979.

[12] Zahorjan, J., Sevcik, K.C., Eager D.L. and Galler, B., *Balanced Job Bound Analysis of Queueing Networks*, Communications of the ACM, Vol. 25, No. 2, pp. 134–141, Feb. 1982.

Κεφάλαιο 6

Δημιουργία Τυχαίων Αριθμών

Σύνοψη

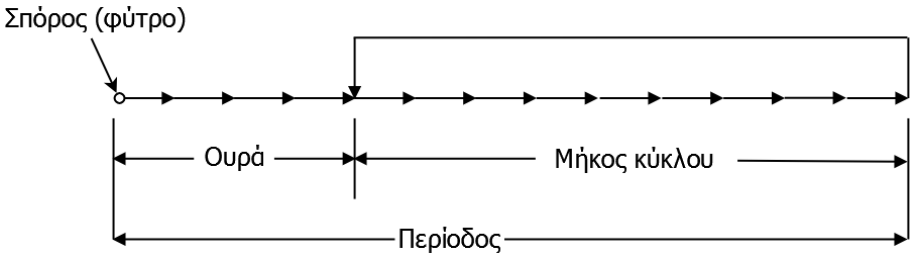
Παρουσιάζονται οι κυριότεροι τύποι γεννητριών για τη δημιουργία ψευδο-τυχαίων αριθμών ομοιόμορφα κατανεμημένων στο διάστημα (0, 1). Έμφαση δίνεται στις γεννήτριες που βασίζονται στον γραμμικό μετασχηματισμό ισοδυναμίας υπολοίπου (linear congruential generators), και τους συνδυασμούς αυτών. Εξετάζονται οι ιδιότητες των γεννητριών σε σχέση με τη μέγιστη περίοδο και την επιλογή του φύτρου. Περιγράφονται οι κυριότεροι στατιστικοί έλεγχοι προσαρμογής όσον αφορά την ομοιομορφία και την ανεξαρτησία των δειγμάτων (τεστ χ^2, τεστ Kolmogorov-Smirnov, τεστ σειριακής συσχέτισης κλπ). Εν συνεχεία, με βάση τους ομοιόμορφα κατανεμημένους τυχαίους αριθμούς, αναπτύσσονται οι κυριότερες τεχνικές δημιουργίας τυχαίων μεταβλητών, με εφαρμογή στις ευρύτερα χρησιμοποιούμενες διακριτές και συνεχείς κατανομές πιθανότητας. Εξετάζονται η μέθοδος της αντιστροφής, η δειγματοληψία απόρριψης, οι μέθοδοι σύνθεσης και συνέλιξης, και διάφορες τεχνικές χαρακτηρισμού βασισμένες στις ιδιότητες των κατανομών.

Τυχαίες μεταβλητές μπορούν να παριστάνουν τη λειτουργία διαφόρων συνιστωσών ενός συστήματος, καθώς και την επίδραση του περιβάλλοντος. Στη γενική περίπτωση ενδιαφερόμαστε για τη δημιουργία μιας ακολουθίας τυχαίων αριθμών που ακολουθούν κάποιο δεδομένο νόμο πιθανότητας. Ο νόμος αυτός μπορεί να δίνεται είτε με τη μορφή ιστογράμματος, είτε με κάποια αναλυτική έκφραση της συνάρτησης κατανομής πιθανότητας [8, 2, 7, 1, 11, 9, 3]. Η δυνατότητα δημιουργίας τυχαίων αριθμών είναι βασικό δομικό στοιχείο της προσομοίωσης και χρησιμοποιείται σε πολλές περιοχές εφαρμογών, όπως δειγματοληψία, αριθμητική ανάλυση και υπολογισμός, λήψη αποφάσεων, αναψυχή. Η τελευταία βασίζεται παραδοσιακά σε χειροκίνητους ή μηχανικούς τρόπους παραγωγής τυχαίων αποτελεσμάτων (ζάρια, παιγνιόχαρτα, περιστρεφόμενος τροχός κλπ). Η χρήση αυτή οδήγησε στον χαρακτηρισμό «μέθοδος Monte Carlo», ο οποίος αποδίδεται συνήθως σε αλγορίθμους που χρησιμοποιούν τυχαίους αριθμούς. Είναι αυτονόητη, όμως, η ανάγκη κατασκευής μηχανισμών αυτόματης παραγωγής τυχαίων αριθμών για την κάλυψη μεγάλων υπολογιστικών εφαρμογών. Οι αριθμοί που δημιουργούνται από τον υπολογιστή χαρακτηρίζονται ως «ψευδοτυχαίοι», διότι, ενώ κατασκευάζονται ντετερμινιστικά, οι ακολουθίες τους έχουν όλα τα χαρακτηριστικά ανεξάρτητων τιμών μιας καθορισμένης τυχαίας κατανομής. Η διαδικασία περιλαμβάνει δύο στάδια. Πρώτα δημιουργείται μια ακολουθία τυχαίων αριθμών που ακολουθούν ομοιόμορφη κατανομή στο διάστημα μεταξύ 0 και 1 (random number generation). Στη συνέχεια με χρήση της ακολουθίας αυτής δημιουργούνται τυχαίες τιμές που ακολουθούν την επιθυμητή κατανομή πιθανότητας (random variate generation).

6.1 Γεννήτριες Τυχαίων Αριθμών

Όπως αναφέρθηκε παραπάνω, σε κάθε περίπτωση, το πρόβλημα ανάγεται στη δημιουργία ακολουθιών τυχαίων πραγματικών αριθμών $\{U_n\}$, ομοιόμορφα κατανεμημένων στο διάστημα $(0, 1)$. Αν διαθέτουμε μία ακολουθία ακέραιων αριθμών $\{X_n\}$ ομοιόμορφα κατανεμημένων σε κάποιο μεγάλο διάστημα $[0, m-1]$, τότε οι αριθμοί U_n μπορούν να υπολογιστούν διαιρώντας διά m:

$$U_n = X_n/m \tag{6.1}$$



Σχήμα 6.1: *Ορισμός της περιόδου.*

Η πιο συνηθισμένη επιλογή που ακολουθείται είναι ο υπολογισμός κάθε αριθμού της ακολουθίας με βάση κάποιο μετασχηματισμό του προηγουμένου του. Έτσι ξεκινώντας από κάποιο αρχικό αριθμό X_0 που ονομάζεται *σπόρος* ή *φύτρο* (seed) της ακολουθίας, θα έχουμε:

$$X_0,\ X_1 = f(X_0), \ldots, X_{n+1} = f(X_n)$$

όπου, βέβαια, ο μετασχηματισμός f θα πρέπει να δίνει τιμές στο ίδιο διάστημα $[0, m-1]$. Μία πρώτη αδυναμία της μεθόδου είναι ότι οι ακολουθίες που δημιουργούνται είναι περιοδικές, εφόσον η τιμή κάποιου X_i θα επανεμφανιστεί σίγουρα μετά από το πολύ m βήματα. Άρα, καταρχήν θα πρέπει να επιλέγονται πολύ μεγάλες τιμές του m. Το τμήμα της ακολουθίας το οποίο επαναλαμβάνεται είναι ο *κύκλος* (cycle) της γεννήτριας. Συχνά υπάρχει ένα αρχικό τμήμα που δεν επαναλαμβάνεται. Το τμήμα αυτό είναι η *ουρά* (tail). Η *περίοδος* (period)της γεννήτριας είναι το μήκος του ωφέλιμου τμήματος της ακολουθίας, δηλαδή το άθροισμα της ουράς και του κύκλου (Σχήμα 6.1).

Στη συνέχεια θα αναφερθούμε σε μια οικογένεια γεννητριών, βασισμένη στη θεωρία της ισοδυναμίας κατά μέτρο. Μια άλλη ενδιαφέρουσα προσέγγιση αφορά την οικογένεια των γεννητριών Tausworthe που δημιουργούν τυχαίους αριθμούς χρησιμοποιώντας τυχαίες ακολουθίες δυαδικών ψηφίων [13, 4].

6.1.1 Γραμμικοί Αλγόριθμοι Ισοδυναμίας κατά Μέτρο

Ένας τύπος γεννητριών που εφαρμόζεται με επιτυχία στην πράξη στηρίζεται στον *γραμμικό μετασχηματισμό ισοδυναμίας κατά μέτρο* (linear congruential tranformation) [4, 5]. Δύο ακέραιοι ονομάζονται *ισοδύναμοι κατά μέτρο* m (modulo m) ή *ισοϋπόλοιποι*, αν διαιρούμενοι διά m αφήνουν το ίδιο υπόλοιπο.

$$X_{n+1} = (aX_n + c) \bmod m \tag{6.2}$$

όπου a και c είναι μη αρνητικές σταθερές. Το ακόλουθο θεώρημα καθορίζει την επιλογή των παραμέτρων a, c και m [5]:

Θεώρημα 6.1. *Μία ακολουθία που δημιουργείται από τον μετασχηματισμό* (6.2) *έχει περίοδο m εάν, και μόνον εάν,*

(i) τα c και m δεν έχουν κοινούς παράγοντες μεγαλύτερους από το 1,

(ii) κάθε πρώτος παράγοντας του m είναι και παράγοντας του $a-1$,

(iii) αν το 4 είναι παράγοντας του m είναι και παράγοντας του $a-1$.

Μία λογική επιλογή είναι η τιμή $m = 2^b$, όπου b είναι ο αριθμός των bits (εκτός από το πρόσημο) στην παράσταση ακεραίων του υπολογιστή. Στην περίπτωση αυτή, η πράξη mod ανάγεται στην αποκοπή των b δεξιότερων ψηφίων. Οι απαιτήσεις του θεωρήματος πληρούνται αν είναι $m = 2^b$, $a = 4k+1$ και c περιττός, π.χ. για $a = 2^{34}+1$, $c = 1$ και $m = 2^{35}$, η γεννήτρια έχει περίοδο 2^{35}.

Μια άλλη επιλογή για την τιμή του m είναι ο μεγαλύτερος πρώτος αριθμός που είναι μικρότερος του 2^b. Κατόπιν, η επιλογή των a και c πρέπει να είναι τέτοια, ώστε, εκτός από τη μέγιστη περίοδο, να εξασφαλίζεται κατά το δυνατό και η ανεξαρτησία των αριθμών X_n. Για τη λεπτομερή ανάλυση των διαφόρων επιλογών παραπέμπουμε στον Knuth [5].

6.1.2 Πολλαπλασιαστικοί Αλγόριθμοι Ισοδυναμίας κατά Μέτρο

Ειδική περίπτωση των γραμμικών γεννητριών αποτελούν οι αλγόριθμοι που βασίζονται στον *πολλαπλασιαστικό μετασχηματισμό ισοδυναμίας κατά μέτρο* (multiplicative congruential):

$$X_{n+1} = (aX_n) \bmod m \tag{6.3}$$

Ο πολλαπλασιαστικός μετασχηματισμός ήταν ο πρώτος που προτάθηκε (D.H. Lehmer, 1951). Στην πραγματικότητα, οι περισσότερες γεννήτριες που χρησιμοποιούνται σήμερα αποτελούν επεκτάσεις της πρότασης του Lehmer, ο οποίος παρατήρησε ότι τα υπόλοιπα της διαίρεσης των δυνάμεων ενός αριθμού a με έναν αριθμό m έχουν χαρακτηριστικά τυχαιότητας. Παρατηρούμε ότι η περίοδος μιας πολλαπλασιαστικής γεννήτριας δεν μπορεί να περιλαμβάνει το 0, διότι θα μηδενίζονταν όλοι οι επόμενοι όροι της ακολουθίας. Άρα η μέγιστη περίοδος θα είναι $m-1$. Το παρακάτω θεώρημα συνοψίζει τη συμπεριφορά της πολλαπλασιαστικής γεννήτριας διακρίνοντας δύο περιπτώσεις ως προς τον διαιρέτη m.

Θεώρημα 6.2. *Όσον αφορά την περίοδο της ακολουθίας που δημιουργείται από τον μετασχηματισμό* (6.3) *ισχύουν τα εξής:*

- *Αν $m = 2^k$, η μέγιστη δυνατή περίοδος της ακολουθίας είναι 2^{k-2} και επιτυγχάνεται αν $a = 8i \pm 3$ και ο σπόρος είναι περιττός.*
- *Αν $m \neq 2^k$, η μέγιστη δυνατή περίοδος της ακολουθίας είναι $m-1$ και επιτυγχάνεται αν m πρώτος αριθμός και a πρωτογενής ρίζα κατά μέτρο m, δηλαδή $a^n \bmod m \neq 1,\ n = 1, \ldots, m-2$.*

Παράδειγμα της πρώτης περίπτωσης είναι η γεννήτρια $X_{n+1} = 5X_n \bmod 2^5$, με περίοδο 8. Παράδειγμα της δεύτερης περίπτωσης είναι η γεννήτρια $X_{n+1} = 7^5 X_n \bmod (2^{31}-1)$, με πλήρη περίοδο $2^{31}-2$. Η τελευταία είναι μια από τις πλέον αποδοτικές και διαδεδομένες γεννήτριες, και οφείλεται στους Park και Miller [10]. Είναι γνωστή με την ονομασία «Minimal Standard».

6.1.3 Υλοποίηση Γραμμικών Γεννητριών

Δύο προβλήματα που σχετίζονται με την υλοποίηση γραμμικών γεννητριών είναι ότι πρώτον οι πράξεις πρέπει να γίνονται ακριβώς, χωρίς λάθη στρογγύλευσης, και δεύτερον ότι υπάρχει κίνδυνος ακέραιας υπερχείλισης κατά την εκτέλεση του πολλαπλασιασμού aX_n. Τα προβλήματα αυτά αποφεύγονται είτε με τη χρήση αριθμητικής πραγματικών αριθμών, εφόσον παρέχεται η επιθυμητή ακρίβεια, είτε με κατάλληλα τεχνάσματα που βασίζονται σε ιδιότητες της αριθμητικής ακεραίων αριθμών.

Μια τεχνική που αντιμετωπίζει το πρόβλημα της υπερχείλισης προτάθηκε από τον Schrage [12]. Η λύση αυτή βασίζεται στην ταυτότητα:

$$ax \bmod m = g(x) + mh(x)$$

όπου

$$\begin{aligned} g(x) &= a(x \bmod q) - r(x \div q) \\ h(x) &= (x \div q) - (ax \div m) \end{aligned}$$

και $q = m \div a$, $r = m \bmod a$. (Ο τελεστής $\div$ παριστάνει ακέραια διαίρεση.) Αποδεικνύεται ότι για κάθε $x \in \{1, 2, \ldots, m-1\}$ οι απαιτούμενες εκφράσεις για τον υπολογισμό του $g(x)$ είναι μικρότερες από $m-1$. Επίσης, αν $r < q$, το $h(x)$ εξαρτάται μόνο από το $g(x)$ και είναι ίσο με 1 αν $g(x) < 0$ διαφορετικά είναι ίσο με 0. Έτσι αποφεύγεται η εκτέλεση της πράξης ax που θα μπορούσε να προκαλέσει υπερχείλιση.

Ακολουθεί (Αλγόριθμος 6.1) υλοποίηση της γεννήτριας Minimal Standard που βασίζεται στην παραπάνω τεχνική, για συστήματα 32-bits με χρήση αριθμητικής ακεραίων (γεννήτρια Park και Miller) [10]. Υποτίθεται ότι έχει προηγηθεί η δήλωση της καθολικής ακέραιας μεταβλητής x, η οποία αρχικοποιείται με μία τιμή στο διάστημα από το 1 έως το 2147483646.

Αλγόριθμος 6.1. *Γεννήτρια Park και Miller, 1988 (Minimal Standard).*

```
double random()
{
#define A 16807
#define M 2147483647
#define Q 127773
#define R 2836
int l, h, t;
h = x / Q;
l = x % Q;
t = A * l - R * h;
if (t > 0)
x = t;
else
x = t + m;
return ((double) x) / m;
}
```

Η παραπάνω υλοποίηση είναι σωστή σε οποιοδήποτε σύστημα ο μέγιστος ακέραιος (MAXINT) είναι τουλάχιστον $2^{31}-1$. Αν αυτό δεν ισχύει, μπορεί να χρησιμοποιηθεί ισοδύναμη διατύπωση με πραγματικούς αριθμούς.

Η ορθότητα της υλοποίησης αυτής της γεννήτριας ελέγχεται με τον υπολογισμό του x_{10000} αρχίζοντας με $x_0 = 1$. Η random θα είναι σωστή αν το αποτέλεσμα είναι 1043618065.

Τελειώνοντας αναφέρουμε γεννήτριες που κατασκευάζουν την ακολουθία με μετασχηματισμό των δύο τελευταίων αριθμών, όπως είναι για παράδειγμα ο *προσθετικός μετασχηματισμός ισοδυναμίας κατά μέτρο* (additive congruential):

$$X_{n+1} = (X_n + X_{n-1}) \bmod m \tag{6.4}$$

Ο αλγόριθμος αυτός, ο οποίος αναφέρεται και ως *γεννήτρια Fibonacci* λόγω της μορφής της αναδρομικής σχέσης, δεν έχει πολύ καλές επιδόσεις. Η κατάσταση βελτιώνεται, αν αντί για τους δύο τελευταίους, χρησιμοποιηθούν οι k τελευταίοι αριθμοί ή κάποιοι μη διαδοχικοί πρόσφατοι αριθμοί της ακολουθίας.

6.1.4 Συνδυασμός Γεννητριών

Μία μέθοδος, η οποία βελτιώνει οποιαδήποτε γεννήτρια και φαίνεται ότι είναι η καλύτερη που προτάθηκε μέχρι τώρα, στηρίζεται στο «ανακάτεμα» (shuffling) μιας ακολουθίας τυχαίων αριθμών (σε αναλογία με το ανακάτεμα μιας τράπουλας). Σύμφωνα με την τεχνική αυτή, δημιουργείται αρχικά ένας πίνακας με k τυχαίους αριθμούς $X_1, X_2, \ldots, X_k$ (με k της τάξης του 100). Στη συνέχεια κάθε τυχαίος αριθμός X_{k+i} που δημιουργείται χρησιμοποιείται για την τυχαία επιλογή ενός στοιχείου του πίνακα, το οποίο κατόπιν αντικαθίσταται με τον X_{k+i}. Μία παραλλαγή της μεθόδου χρησιμοποιεί δύο γεννήτριες τυχαίων αριθμών $\{X_n\}$ και $\{Y_n\}$, μία για να γεμίζει τον πίνακα και μία για να επιλέγει τυχαία στοιχεία του πίνακα (συνδυασμός γεννητριών). Η μέθοδος αποδεικνύεται πολύ αποτελεσματική, ακόμη και αν οι ακολουθίες που χρησιμοποιούνται δεν είναι από μόνες τους ιδιαίτερα ικανοποιητικές.

Μία άλλη κατηγορία μεθόδων βασίζεται στην πρόσθεση τυχαίων αριθμών που προέρχονται από διαφορετικές γεννήτριες. Με κατάλληλη επιλογή των παραμέτρων μπορεί να προκύψουν γεννήτριες πολύ καλών επιδόσεων, τόσο ως προς την περίοδο όσο και ως προς την τυχαιότητα.

Ακολουθεί η υλοποίηση μιας γεννήτριας για συστήματα 32-bits, η οποία θεωρείται ιδιαίτερα αποτελεσματική [6]. Η γεννήτρια αυτή οφείλεται στον L'Ecuyer και πραγματοποιεί αφαίρεση τυχαίων αριθμών από

διαφορετικές γεννήτριες:

$$\begin{aligned} x_{n+1} &= 40014x_n \bmod 2147483563 \\ y_{n+1} &= 40692y_n \bmod 2147483399 \\ z_{n+1} &= (x_{n+1} - y_{n+1}) \bmod 2147483562 \end{aligned}$$

Οι ακέραιες μεταβλητές s1 και s2 είναι καθολικές και αρχικοποιούνται σε τιμές που ανήκουν στις περιοχές [1,2147483562] και [1,2147483398] αντίστοιχα (Αλγόριθμος 6.2).

Αλγόριθμος 6.2. *Γεννήτρια L' Ecuyer, 1988.*

```
double uniform()
{
int z, k;
k = s1 / 53668;
s1 = 40014 * (s1 - k * 53668) - k * 12211;
if (s1 < 0)
s1 = s1 + 2147483563;
k = s2 / 52774;
s2 = 40692 * (s2 - k * 52774) - k * 3791;
if (s2 < 0)
s2 = s2 + 2147483399;
z = s1 - s2;
if (z < 1)
z = z + 2147483562;
return z * 4.656613e-10;
}
```

6.1.5 Έλεγχος Γεννητριών

Όλοι οι αλγόριθμοι (γεννήτριες) που δημιουργούν αριθμούς στο διάστημα $[0, m-1]$ δεν εξασφαλίζουν ικανοποιητικά αποτελέσματα, όσον αφορά την τυχαιότητα. Πριν χρησιμοποιηθεί μία γεννήτρια τυχαίων αριθμών θα πρέπει οπωσδήποτε να περάσει κάποια από τα βασικά στατιστικά τεστ, όπως π.χ. το τεστ χ^2, που είναι και το πιο διαδεδομένο στην πράξη, το τεστ Kolmogorov-Smirnov και άλλα [4, 11].

6.1.5.1 Τεστ χ^2

Επιτρέπει να ελέγξουμε κατά πόσο ένα σύνολο δεδομένων ακολουθεί μία δεδομένη κατανομή. Είναι γενικό και μπορεί να χρησιμοποιηθεί για οποιαδήποτε κατανομή.

Οι συχνότητες (πλήθος εμφανίσεων) ενδεχομένων που βασίζονται σε ένα ιστόγραμμα των δεδομένων συγκρίνονται με αυτές που υπολογίζονται από την κατανομή πιθανότητας που ελέγχεται. Έστω ότι το ιστόγραμμα περιλαμβάνει k διαστήματα και o_i, e_i οι παρατηρηθείσες και αναμενόμενες συχνότητες (διακριτές πιθανότητες) αντίστοιχα για το i διάστημα. Ο έλεγχος βασίζεται στον υπολογισμό της ποσότητας

$$D = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$$

Για απόλυτη προσαρμογή θα πρέπει να προκύπτει $D = 0$, λόγω της τυχαιότητας των δεδομένων όμως η τιμή του D είναι μη μηδενική. Αποδεικνύεται ότι το D ακολουθεί κατανομή χ^2 με $k-1$ βαθμούς ελευθερίας. Η υπόθεση ότι τα δεδομένα προέρχονται από την καθορισμένη κατανομή απορρίπτεται με βαθμό

σημαντικότητας α αν η τιμή του D είναι μεγαλύτερη από την τιμή $\chi^2_{[1-\alpha;k-1]}$ των πινάκων της κατανομής χ^2, διαφορετικά η υπόθεση γίνεται δεκτή.

Το τεστ χ^2 ενδείκνυται για διακριτές κατανομές και μεγάλα δείγματα. Απαραίτητη προϋπόθεση είναι να έχουμε τουλάχιστον 5 τιμές σε κάθε διάστημα του ιστογράμματος, διαφορετικά ενώνουμε γειτονικά διαστήματα. Επίσης, το τεστ αποδίδει καλύτερα όταν τα μεγέθη των διαστημάτων επιλεγούν έτσι ώστε οι αναμενόμενες συχνότητες e_i να είναι ίσες μεταξύ τους (πράγμα που ισχύει για ίσα διαστήματα στην περίπτωση της ομοιόμορφης κατανομής). Για μικρά δείγματα και συνεχείς κατανομές ενδείκνυται το τεστ Kolmogorov-Smirnov, το οποίο είναι ελαφρώς πιο πολύπλοκο και δίνει ακριβέστερα αποτελέσματα.

Παράδειγμα 6.1. *Χίλιοι τυχαίοι αριθμοί προερχόμενοι από μία γεννήτρια ομαδοποιούνται σε ένα ιστόγραμμα με* 10 *διαστήματα μήκους* 0,1. *Οι παρατηρηθείσες συχνότητες* o_i *στα* 10 *διαστήματα είναι:* 106, 103, 96, 92, 94, 102, 87, 119, 91, 110. *Εφαρμόζοντας το τεστ* χ^2 *θέλουμε να ελέγξουμε με βαθμό σημαντικότητας* $\alpha = 0,2$*,αν οι αριθμοί είναι ομοιόμορφα κατανεμημένοι στο* (0,1).

Αν οι αριθμοί ήταν ομοιόμορφα κατανεμημένοι στο (0,1), κάθε διάστημα θα έπρεπε να περιλαμβάνει $e_i = 100$ αριθμούς. Ο υπολογισμός της ποσότητας D δίνει 8,76, ενώ από τους πίνακες της κατανομής χ^2 έχουμε $\chi^2_{[0,8;9]} = 12,24$. Επομένως η υπόθεση γίνεται δεκτή. □

6.1.5.2 Τεστ Kolmogorov-Smirnov

Το τεστ K-S αναπτύχθηκε από τους Ρώσους μαθηματικούς A.N. Kolmogorov και N.V. Smirnov τη δεκαετία του 1930 και ελέγχει κατά πόσο ένα δείγμα παρατηρήσεων προέρχεται από μια δεδομένη συνεχή κατανομή. Σε αντιστοιχία με το τεστ χ^2 που βασίζεται στη διαφορά ανάμεσα σε παρατηρηθείσες και αναμενόμενες πιθανότητες, το τεστ K-S βασίζεται στη διαφορά ανάμεσα στην παρατηρηθείσα Συνάρτηση Κατανομής Πιθανότητας (ΣΚΠ) $F_o(x)$ και την αναμενόμενη ΣΚΠ $F_e(x)$. Η διαφορά εκφράζεται μέσω των ποσοτήτων K^+ και K^-, οι οποίες, σε δείγμα μεγέθους n, παριστάνουν τις μέγιστες παρατηρηθείσες αποκλίσεις πάνω και κάτω από την αναμενόμενη ΣΚΠ, αντίστοιχα:

$$\begin{aligned} K^+ &= \sqrt{n}\max_x [F_o(x) - F_e(x)] \\ K^- &= \sqrt{n}\max_x [F_e(x) - F_o(x)] \end{aligned}$$

Η υπόθεση ότι τα δεδομένα προέρχονται από την καθορισμένη κατανομή απορρίπτεται με βαθμό σημαντικότητας α, αν οι τιμές K^+ και K^- είναι μεγαλύτερες από την τιμή $K_{[1-\alpha;n]}$ των πινάκων της κατανομής $K - S$, διαφορετικά θεωρούμε ότι η υπόθεση γίνεται δεκτή.

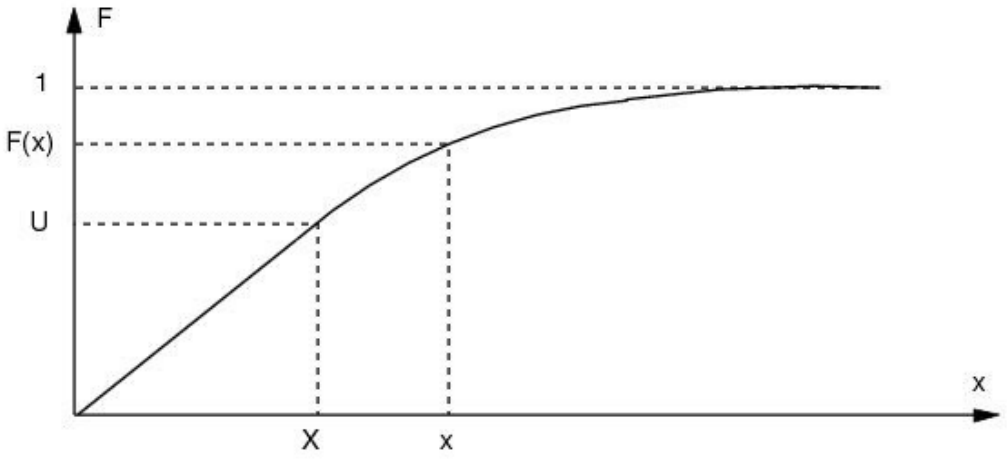
Όπως ήδη αναφέρθηκε, το τεστ Kolmogorov-Smirnov αναπτύχθηκε για μικρά δείγματα και συνεχείς κατανομές, ενώ το τεστ χ^2 για μεγάλα δείγματα και διακριτές κατανομές. Το τεστ χ^2 εκτιμά πιθανότητες ομαδοποιώντας τα δεδομένα σε διαστήματα αυθαίρετου μήκους (με την έννοια ότι δεν υπάρχουν αυστηροί κανόνες επιλογής του μήκους). Αντίθετα, το τεστ K-S χρησιμοποιεί τα δεδομένα με άμεσο τρόπο (χωρίς ομαδοποίηση), άρα αξιοποιεί καλύτερα τη διαθέσιμη πληροφορία. Γενικά, το τεστ χ^2 είναι προσεγγιστική μέθοδος, ενώ το τεστ K-S μπορεί να θεωρηθεί ακριβής μέθοδος.

6.1.5.3 Τεστ Σειριακής Συσχέτισης

Τα δύο προηγούμενα τεστ ελέγχουν την καλή προσαρμογή (goodness-of-fit) των τυχαίων αριθμών στην επιθυμητή κατανομή πιθανότητας. Πέραν αυτού, όμως, πρέπει να ελεγχθεί και η ανεξαρτησία των αριθμών. Μία μέθοδος ελέγχου της ανεξαρτησίας δύο τυχαίων μεταβλητών είναι το τεστ σειριακής συσχέτισης (serial-correlation test), το οποίο ελέγχει την εξάρτηση των δεδομένων μέσω του υπολογισμού της συνδιακύμανσης (covariance).

Αν διαθέτουμε μια ακολουθία τυχαίων αριθμών, υπολογίζουμε τη συνδιακύμανση μεταξύ αριθμών που απέχουν κατά k θέσεις ($k \geq 1$) μέσα στην ακολουθία. Η ποσότητα αυτή ονομάζεται αυτοσυνδιακύμανση υστέρησης k (autocovariance at lag k):

$$R_k = \frac{1}{n-k}\sum_{i=1}^{n-k}\left(U_i - \frac{1}{2}\right)\left(U_{i+k} - \frac{1}{2}\right)$$



Σχήμα 6.2: *Αντιστροφή συνεχούς κατανομής.*

Για μεγάλο μέγεθος δείγματος n, η αυτοσυνδιακύμανση R_k ακολουθεί κατά καλή προσέγγιση την κανονική κατανομή με μέση τιμή 0 και διασπορά $1/[144(n-k)]$. Συνεπώς, το διάστημα εμπιστοσύνης με βαθμό εμπιστοσύνης $(1-\alpha)$ θα είναι:

$$\left[R_k \mp \frac{z_{1-\alpha/2}}{12\sqrt{n-k}}\right]$$

Αν το διάστημα εμπιστοσύνης δεν περιλαμβάνει το 0, μπορούμε να θεωρήσουμε ότι η ακολουθία τυχαίων αριθμών εμφανίζει σημαντική συσχέτιση. Το αντίστροφο δεν ισχύει, δηλαδή, αν το 0 περιλαμβάνεται στο διάστημα εμπιστοσύνης, μπορεί επίσης να υπάρχει συσχέτιση.

Από τα παραπάνω είναι φανερό ότι μια ακολουθία μπορεί να περάσει ένα τεστ στατιστικής επικύρωσης και να αποτύχει σε ένα άλλο. Επίσης, τα διάφορα τεστ συχνά δίνουν απάντηση μόνο στην περίπτωση κατά την οποία η ακολουθία τυχαίων αριθμών αποτυγχάνει στο τεστ, ενώ δεν υπάρχει βεβαιότητα στην περίπτωση επιτυχίας στο τεστ. Κατά συνέπεια, είναι αναγκαίο να χρησιμοποιηθούν όσο το δυνατόν περισσότερα τεστ, ώστε να υπάρχει αλληλοκάλυψη των αποφάσεων.

6.2 Δημιουργία Τυχαίων Μεταβλητών

Οι ακολουθίες τυχαίων αριθμών $\{U_n\}$ ομοιόμορφα κατανεμημένων στο διάστημα $(0,1)$ χρησιμοποιούνται για τη δημιουργία άλλων τυχαίων ποσοτήτων με διαφορετικές κατανομές. Στη συνέχεια θα αναφερθούμε στις διάφορες μεθόδους δημιουργίας τέτοιων τυχαίων δεδομένων, όταν διαθέτουμε τους τυχαίους αριθμούς U με την ομοιόμορφη κατανομή:

$$\Pr[U \le x] = \begin{cases} 0 & x < 0 \\ x & 0 \le x < 1 \\ 1 & x \ge 1 \end{cases} \tag{6.5}$$

6.2.1 Αντιστροφή Συνεχών Κατανομών

Έστω ότι θέλουμε να δημιουργήσουμε μία τυχαία μεταβλητή X με συνεχή συνάρτηση κατανομής πιθανότητας $F(x) = \Pr[X \le x]$. Για τη συνάρτηση αυτή υπάρχει η αντίστροφη συνάρτηση $F^{-1}(y)$, η οποία είναι συνεχής και μονοτόνως αύξουσα και ικανοποιεί τη σχέση $F^{-1}(F(x)) = x$. Επιλέγουμε την τιμή $X = F^{-1}(U)$ και είναι εύκολο να αποδείξουμε ότι η μεταβλητή X ακολουθεί πράγματι τη δεδομένη κατανομή (Σχήμα 6.2):

$$\Pr[X \le x] = \Pr[F^{-1}(U) \le x] = \Pr[U \le F(x)] = F(x)$$

με χρήση της μονοτονίας της F^{-1} και της (6.5).

Για παράδειγμα, ας θεωρήσουμε την *ομοιόμορφη κατανομή* στο διάστημα (a, b) για την οποία ισχύει

$$F(x) = (x - a)/(b - a)$$

Με απλή αντιστροφή βρίσκουμε:

$$X = F^{-1}(U) = (b - a)U + a \tag{6.6}$$

Έστω, τώρα ότι θέλουμε να δημιουργήσουμε την τυχαία μεταβλητή X με *εκθετική συνάρτηση κατανομής πιθανότητας* $F(x) = 1 - e^{-\lambda x}$, $x \geq 0$. Με χρήση της αντιστροφής έχουμε:

$$X = F^{-1}(U) = -\frac{1}{\lambda}\ln(1 - U) = -\frac{1}{\lambda}\ln(U) \tag{6.7}$$

όπου θέσαμε U στη θέση του $1-U$, όπως αναφέρθηκε και για τη γεωμετρική κατανομή, εφόσον και οι δύο ποσότητες ακολουθούν την ίδια κατανομή.

Για να εφαρμοστεί η μέθοδος της αντιστροφής θα πρέπει να μπορεί να προσδιοριστεί εύκολα η αντίστροφη της συνάρτησης κατανομής. Για τις περιπτώσεις όπου αυτό δεν είναι δυνατό υπάρχουν διάφορες άλλες εναλλακτικές μέθοδοι.

Παράδειγμα 6.2. *Διαθέτουμε γεννήτρια τυχαίων αριθμών U, ομοιόμορφα κατανεμημένων στο διάστημα $(0, 1)$. Να εφαρμοστεί η μέθοδος της αντιστροφής για τη δημιουργία τυχαίας μεταβλητής με συνάρτηση πυκνότητας πιθανότητας:*

$$f(x) = e^x/(e - 1),\ 0 \leq x < 1$$

Υπολογίζουμε τη συνάρτηση κατανομής πιθανότητας:

$$F(x) = \int_0^x \frac{e^z}{e-1}dz = \frac{e^x - 1}{e - 1},\ 0 \leq x < 1$$

και εν συνεχεία υπολογίζουμε την αντίστροφή της $X = F^{-1}(U)$

$$U = \frac{e^X - 1}{e - 1} \Rightarrow e^X = (e - 1)U + 1$$

ή τελικά

$$X = \ln[(e - 1)U + 1]$$

Μπορούμε να επαληθεύσουμε την αντιστοίχιση τιμών των U και X:

$$\begin{aligned} 0 \leq X < 1 \Rightarrow 0 \leq \ln[(e-1)U + 1] < 1 &\Rightarrow 1 \leq (e-1)U + 1 < e \\ &\Rightarrow 0 \leq (e-1)U < e - 1 \Rightarrow 0 \leq U < 1 \end{aligned}$$

□

Παράδειγμα 6.3. *Ομοίως, να εφαρμοστεί η μέθοδος της αντιστροφής για τη δημιουργία τυχαίας μεταβλητής με συνάρτηση πυκνότητας πιθανότητας:*

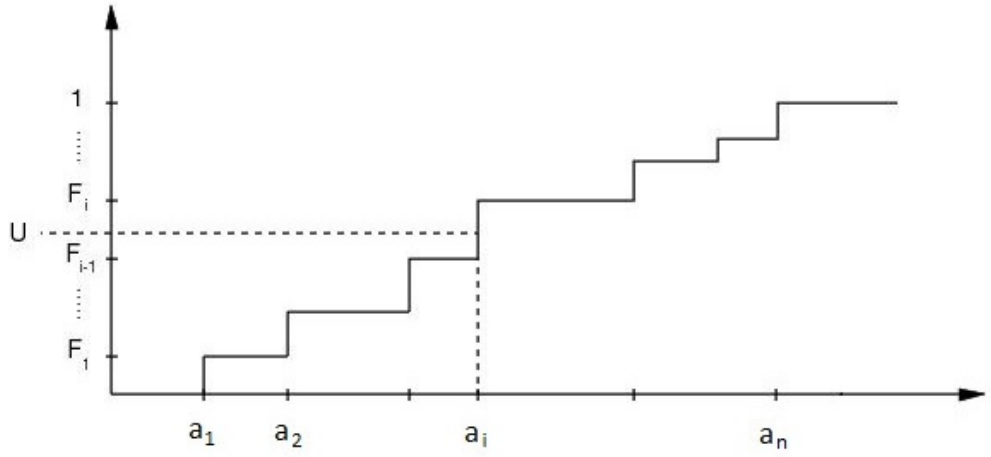
$$f(x) = \min(x, 2 - x),\ 0 \leq x < 2$$

Η συνάρτηση $f(x)$ μπορεί να γραφτεί ισοδύναμα:

$$f(x) = \begin{cases} x & 0 \leq x < 1 \\ 2 - x & 1 \leq x < 2 \end{cases}$$

Υπολογίζουμε τη συνάρτηση κατανομής πιθανότητας $F(x)$ για τις δύο περιπτώσεις ορισμού της $f(x)$:

$$\begin{aligned} 0 \leq x < 1: &\quad F(x) = \textstyle\int_0^x z dz = \frac{x^2}{2} \\ 1 \leq x < 2: &\quad F(x) = \textstyle\int_0^1 z dz + \int_1^x (2 - z)dz = 2x - \frac{x^2}{2} - 1 \end{aligned}$$



Σχήμα 6.3: *Αντιστροφή διακριτής κατανομής.*

Εν συνεχεία αντιστρέφουμε την $F(x)$:

$$\begin{array}{ll} 0 \le x < 1: & U = \frac{X^2}{2} \Rightarrow X = \pm\sqrt{2U} \qquad\qquad\quad \text{Δεκτό το } +. \\ 1 \le x < 2: & U = 2X - \frac{X^2}{2} - 1 \Rightarrow X = 2 \pm \sqrt{2(1-U)} \quad \text{Δεκτό το -.} \end{array}$$

Ελέγχουμε την αντιστοίχιση τιμών των U και X:

$$\begin{array}{ll} 0 \le X < 1 & \Rightarrow 0 \le \sqrt{2U} < 1 \Rightarrow 0 \le U < \frac{1}{2} \\ 1 \le X < 2 & \Rightarrow 1 \le 2 - \sqrt{2(1-U)} < 2 \Rightarrow 0 < \sqrt{2(1-U)} \le 1 \Rightarrow \frac{1}{2} \le U < 1 \end{array}$$

Τελικά:

$$X = \begin{cases} \sqrt{2U} & 0 \le U < \frac{1}{2} \\ 2 - \sqrt{2(1-U)} & \frac{1}{2} \le U < 1 \end{cases}$$

□

6.2.2 Αντιστροφή Διακριτών Κατανομών

Θεωρούμε διακριτή κατανομή με n δυνατές τιμές $a_1, a_2, \ldots, a_n$ και αντίστοιχες πιθανότητες $p_1, p_2, \ldots, p_n$ ($\sum_{k=1}^{n} p_k = 1$). Αν ορίσουμε τις αθροιστικές πιθανότητες $F_i = p_1 + p_2 + \cdots + p_i$ ($i = 1, 2, \ldots, n$), τότε επιλέγουμε τη μικρότερη τιμή a_i για την οποία ισχύει $U < F_i$ (Σχήμα 6.3). Παρατηρούμε ότι στην ουσία γίνεται αντιστροφή της αθροιστικής συνάρτησης πιθανότητας, δηλαδή, δεδομένου του U ζητάμε το i για το οποίο ισχύει:

$$F_{i-1} \le U < F_i$$

Για παράδειγμα, αν είχαμε δύο δυνατές τιμές a_1 και a_2 με πιθανότητες p και $1-p$ αντίστοιχα, θα επιλέγαμε την a_1 αν $U < p$ αλλιώς την a_2.

Η μέθοδος μπορεί να εφαρμοστεί και σε διακριτές κατανομές με άπειρο πλήθος τιμών. Ας θεωρήσουμε για παράδειγμα τη *γεωμετρική κατανομή* με συνάρτηση μάζας πιθανότητας $p_k = (1-a)a^{k-1}$, $k = 1, 2, \ldots$ ($0 < a < 1$) της οποίας η αθροιστική κατανομή πιθανότητας είναι $F_i = 1 - a^i$, $i = 1, 2, \ldots$. Αντικατάσταση στην ανισότητα της αντιστροφής δίνει:

$$i - 1 \le \ln(1-U)/\ln a < i$$

οπότε επιλέγουμε την τιμή $i = 1 + \lfloor \ln(1-U)/\ln a \rfloor$. Στην τελευταία έκφραση η ποσότητα $1-U$ μπορεί να αντικατασταθεί με την ποσότητα U, εφόσον και οι δύο ακολουθούν την ίδια ομοιόμορφη κατανομή, οπότε παίρνουμε τελικά την απλούστερη σχέση:

$$i = 1 + \lfloor \ln(U)/\ln a \rfloor \tag{6.8}$$

Παράδειγμα 6.4. *Αν διαθέτουμε γεννήτρια τυχαίων αριθμών ομοιόμορφα κατανεμημένων στο διάστημα* $(0,1)$*, να εφαρμοστεί η μέθοδος της αντιστροφής για τη δημιουργία τυχαίων μεταβλητών* X *που ακολουθούν τη διωνυμική κατανομή με συνάρτηση μάζας πιθανότητας*

$$p_k = \left(\begin{array}{c} n \\ k \end{array} \right) q^k (1-q)^{n-k},\ k = 0, 1, \ldots, n$$

Η διωνυμική κατανομή χαρακτηρίζει τον αριθμό k των επιτυχιών σε n ανεξάρτητα πειράματα Bernoulli με πιθανότητα επιτυχίας q. Για να εφαρμόσουμε τη μέθοδο της αντιστροφής, απαιτείται η αθροιστική κατανομή πιθανότητας, η οποία στην περίπτωση αυτή δεν μπορεί να υπολογιστεί σε κλειστή μορφή. Θα καταφύγουμε, λοιπόν, σε αλγοριθμική λύση υπολογίζοντας την αθροιστική κατανομή με επαναληπτικό τρόπο. Η επανάληψη θα βασιστεί στον παρακάτω αναδρομικό τύπο που εκφράζει την πιθανότητα p_{k+1} συναρτήσει της p_k:

$$p_{k+1} = \frac{n!}{(n-k-1)!(k+1)!} q^{k+1} (1-q)^{n-k-1}$$

απ' όπου

$$p_{k+1} = \frac{n-k}{k+1} \frac{q}{1-q} p_k,\ 0 \leq k < n$$

Σε κάθε βήμα του αλγορίθμου που ακολουθεί προστίθεται ένας όρος στην αθροιστική πιθανότητα F_i, μέχρι να προσδιοριστεί το i για το οποίο ισχύει $F_{i-1} \leq U < F_i$, όπου χρησιμοποιείται ένας τυχαίος αριθμός U. Ο αλγόριθμος είναι γραμμένος σε ψευδογλώσσα τύπου C.

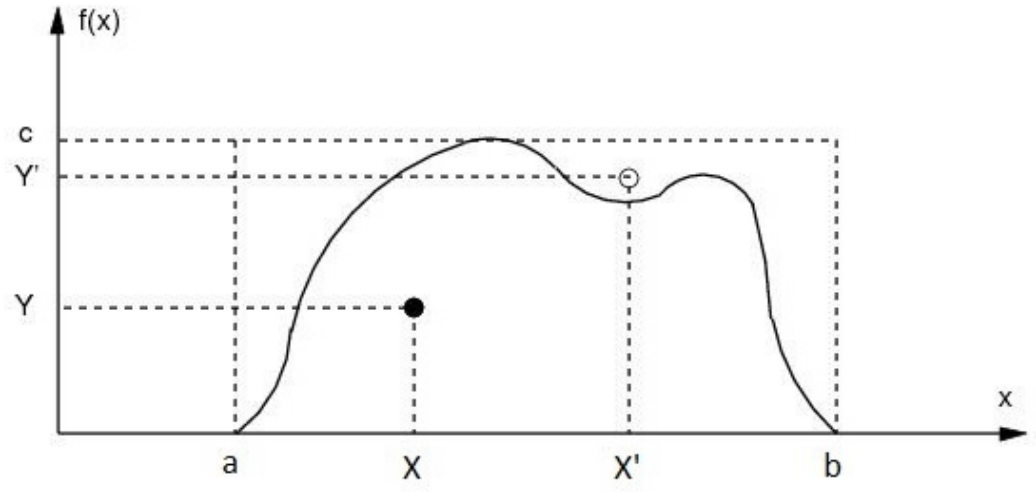
```
U=random();
c=q/(1-q); pr=(1-q)^n;
k=0; F=pr;
while (F<=U) {
pr=pr*c*(n-k)/(k+1);
F=F+pr; k=k+1;
}
return X=k;
}
```

Παρατηρούμε ότι άλλος τρόπος δημιουργίας διωνυμικής μεταβλητής X μπορεί να βασιστεί στη φυσική ερμηνεία της, ως του αριθμού k των επιτυχιών σε n ανεξάρτητα πειράματα Bernoulli. Άρα, δημιουργούμε n τυχαίους U ομοιόμορφους στο (0,1) (αντί του ενός U που χρησιμοποιήσαμε στην αντιστροφή) και μετράμε για πόσους ισχύει $U < q$. Πρόκειται για τη *μέθοδο του χαρακτηρισμού* στην οποία θα αναφερθούμε παρακάτω (Ενότητα 6.2.4). □

6.2.3 Η Μέθοδος της Απόρριψης

Όταν η συνάρτηση πυκνότητας πιθανότητας $f(x)$ της ζητούμενης μεταβλητής X περιλαμβάνεται ολόκληρη σε ένα ορθογώνιο με βάση (a,b) και ύψος c (Σχήμα 6.4) μπορούμε να εφαρμόσουμε τη *μέθοδο της απόρριψης* (rejection (sampling) method), σύμφωνα με την οποία:

- δημιουργούμε δύο τυχαίους αριθμούς U_1 και U_2,
- θέτουμε $X = a + (b-a)U_1$ και $Y = cU_2$,
- δεχόμαστε το X αν $Y < f(X)$, αλλιώς το απορρίπτουμε και επαναλαμβάνουμε τη διαδικασία.



Σχήμα 6.4: *Μέθοδος της απόρριψης.*

Με άλλα λόγια, δημιουργούμε τυχαία σημεία (X, Y) ομοιόμορφα κατανεμημένα στο ορθογώνιο $(a, b) \times (0, c)$, μέχρις ότου κάποιο σημείο βρεθεί κάτω από την καμπύλη της $f(x)$. Η συντεταγμένη X του σημείου αυτού είναι η μεταβλητή που ζητάμε, πράγμα το οποίο αποδεικνύεται ως εξής:

$$\begin{aligned} \Pr[x \le X < x + dx / Y < f(X)] &= \frac{\Pr[x \le X < x + dx,\ Y < f(X)]}{\Pr[Y < f(X)]} \\ &= \frac{[dx/(b-a)]\,[f(x)/c]}{1/[(b-a)c]} = f(x)dx, \quad a < x < b \end{aligned}$$

εφόσον τα X και Y είναι ομοιόμορφα κατανεμημένα στα διαστήματα (a, b) και $(0, c)$ αντίστοιχα, και η πιθανότητα $\Pr[Y < f(X)]$ να βρεθεί το σημείο (X, Y) κάτω από την καμπύλη είναι ίση με το λόγο της επιφάνειας κάτω από την καμπύλη (δηλαδή 1) προς την επιφάνεια ολόκληρου του ορθογωνίου.

6.2.3.1 Η Γενική Περίπτωση

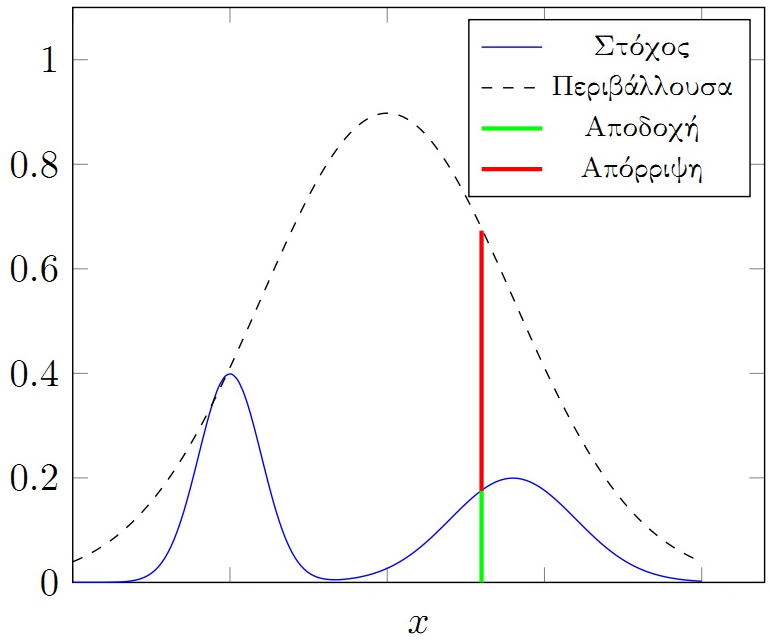
Η μέθοδος που περιγράψαμε προηγουμένως αποτελεί ειδική περίπτωση μιας γενικότερης διατύπωσης της μεθόδου της απόρριψης. Θεωρούμε τη συνάρτηση πυκνότητας πιθανότητας $f(x)$ της συνεχούς κατανομής της οποίας θέλουμε να δημιουργήσουμε δείγματα (κατανομή *στόχος*). Έστω, επίσης, ότι διαθέτουμε τη συνεχή κατανομή με σππ $q(x)$, της οποίας μπορούμε «εύκολα» να δημιουργήσουμε τυχαία δείγματα. Αν υπάρχει σταθερά M τέτοια ώστε

$$M \cdot q(x) > f(x),\ \forall x$$

η συνάρτηση $M \cdot q(x)$ (περιβάλλουσα κατανομή) περικλείει εντελώς τη συνάρτηση στόχο $f(x)$.

Είναι σαφές ότι, στην περίπτωση που εξετάστηκε προηγουμένως, τον ρόλο της κατανομής $q(x)$ έπαιζε η ομοιόμορφη κατανομή. Πέραν αυτού, η μέθοδος, όπως προηγουμένως, δημιουργεί δείγματα τα οποία υποβάλλονται στο κριτήριο αποδοχής – απόρριψης:

- Με τη μέθοδο που διαθέτουμε, δημιουργούμε τυχαία μεταβλητή X που ακολουθεί την κατανομή $q(x)$.
- Δημιουργούμε τυχαίο αριθμό U, ομοιόμορφα κατανεμημένο στο διάστημα $(0, 1)$.
- Δεχόμαστε το X αν $U \le \frac{f(X)}{M \cdot q(x)}$, αλλοιώς το απορρίπτουμε και επαναλαμβάνουμε τη διαδικασία.



Σχήμα 6.5: *Η γενική μέθοδος της απόρριψης.*

Προκειμένου να γίνει κατανοητή η λειτουργία της μεθόδου, μπορούμε καταρχάς να υπολογίσουμε την *πιθανότητα αποδοχής* μιας τυχαίας μεταβλητής σύμφωνα με το κριτήριο:

$$\begin{aligned}
\Pr[\text{accept}] &= \int_x \Pr[\text{accept} \mid x \le X < x + dx] \Pr[x \le X < x + dx] \\
&= \int_x \Pr[U \le \frac{f(X)}{M \cdot q(x)} \mid x \le X < x + dx] q(x) dx \\
&= \int_x \frac{f(X)}{M \cdot q(x)} q(x) dx = \frac{1}{M} \int_x f(x) dx = \frac{1}{M}
\end{aligned}$$

Το αποτέλεσμα αυτό συνεπάγεται ότι, κατά μέσο όρο, απαιτούνται M βήματα (με δημιουργία δύο τυχαίων μεταβλητών ανά βήμα) για τη δημιουργία μιας μεταβλητής της κατανομής στόχου. Κλείνοντας, θα δείξουμε ότι η μεταβλητή X, που δημιουργήσαμε, ακολουθεί την κατανομή στόχο $f(x)$:

$$\begin{aligned}
\Pr[x \le X < x + dx \mid \text{accept}] &= \frac{\Pr[x \le X < x + dx, \text{accept}]}{\Pr[\text{accept}]} \\
&= \frac{\Pr[\text{accept} \mid x \le X < x + dx] \ \Pr[x \le X < x + dx]}{\Pr[\text{accept}]} \\
&= \frac{\frac{f(x)}{M \cdot q(x)} q(x) dx}{\frac{1}{M}} = f(x) dx
\end{aligned}$$

Η μέθοδος της απόρρριψης μπορεί να εφαρμοστεί με ανάλογο τρόπο στην περίπτωση διακριτης κατανομής στόχου χρησιμοποιώντας διακριτή περιβάλλουσα κατανομή.

6.2.4 Συνθετικές Μέθοδοι

Τυχαίες μεταβλητές που ακολουθούν διάφορες πολύπλοκες κατανομές μπορούν να δημιουργηθούν με συνδυασμό των προηγουμένων μεθόδων.

Ας θεωρήσουμε την *υπερεκθετική κατανομή τάξης* k για την οποία ισχύει $f(x) = \sum_{i=1}^{k} \alpha_i \mu_i e^{-\mu_i x}$, $x \geq 0$ ($\sum_{i=1}^{k} \alpha_i = 1$). Για να δημιουργήσουμε τη μεταβλητή X, επιλέγουμε πρώτα μία τυχαία τιμή i σύμφωνα με τη διακριτή κατανομή α_i. Στη συνέχεια δημιουργούμε μία τυχαία μεταβλητή με εκθετική κατανομή παραμέτρου μ_i η οποία θα είναι η ζητούμενη μεταβλητή X.

Προφανώς, η μέθοδος αυτή, η οποία αναφέρεται ως *σύνθεση* (ή και *διάσπαση*), μπορεί να εφαρμοστεί για οποιαδήποτε κατανομή η οποία είναι γραμμικός συνδυασμός άλλων κατανομών.

Αντίστοιχα, μπορούμε να δημιουργήσουμε τη μεταβλητή X με κατανομή Erlang-k προσθέτοντας k εκθετικά κατανεμημένες τυχαίες μεταβλητές:

$$X = \sum_{i=1}^{k} (-\frac{1}{\lambda}) \ln(U_i) = -\frac{1}{\lambda} \ln \left(\prod_{i=1}^{k} U_i \right) \tag{6.9}$$

Η μέθοδος αυτή ονομάζεται και μέθοδος της *συνέλιξης* (convolution), καθόσον η συνάρτηση πυκνότητας πιθανότητας του αθροίσματος τυχαίων μεταβλητών εκφράζεται αναλυτικά ως η συνέλιξη των επιμέρους συναρτήσεων πυκνότητας πιθανότητας.

Συνδυάζοντας κατάλληλα διακριτές κατανομές και εκθετικές κατανομές μπορούμε να δημιουργήσουμε οποιαδήποτε κατανομή Cox (μέθοδος των εκθετικών σταδίων).

Άλλο παράδειγμα εφαρμογής της μεθόδου της συνέλιξης αποτελεί η *κανονική κατανομή*. Σύμφωνα με το Κεντρικό Οριακό Θεώρημα, το άθροισμα n ανεξάρτητων τυχαίων μεταβλητών που ακολουθούν την ίδια κατανομή με μέση τιμή m και διασπορά s^2 συγκλίνει, όσο αυξάνει το n, προς την κανονική κατανομή με μέση τιμή nm και διασπορά ns^2. Γνωρίζοντας ότι για την ομοιόμορφη κατανομή στο διάστημα $(0, 1)$ ισχύει $m = 1/2$ και $s^2 = 1/12$ διαπιστώνουμε εύκολα ότι η μεταβλητή

$$X = \frac{\sum_{i=1}^{n} U_i - n/2}{\sqrt{n/12}} \sigma + \mu \tag{6.10}$$

αποτελεί προσέγγιση κανονικής μεταβλητής με μέση τιμή μ και διασπορά σ^2, για σχετικά μεγάλο n. Θα πρέπει να αναφερθεί ότι υπάρχουν διάφορες μέθοδοι για τη δημιουργία τυχαίων μεταβλητών με κανονική κατανομή, μερικές από τις οποίες είναι ιδιαίτερα αποτελεσματικές.

Τελειώνοντας σημειώνουμε ότι η δημιουργία τυχαίων μεταβλητών ορισμένων κατανομών μπορεί να βασιστεί σε χαρακτηριστικά των κατανομών που επιτρέπουν τη διατύπωση ειδικών αλγορίθμων. Οι τεχνικές αυτές αναφέρονται γενικά ως τεχνικές *χαρακτηρισμού* (characterization).

Βιβλιογραφία

[1] Bolch, G., Greiner, S., De Meer, H., and Trivedi, K.S., *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, Wiley-Interscience, 2006.

[2] Bratley, P., Fox, B.L. and Schrage, L.E., *A Guide to Simulation*, Springer-Verlag, 1986.

[3] Harchol-Balter, M., *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, Cambridge University Press, 2013.

[4] Jain, R., *The Art of Computer Systems Performance Analysis - Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, 1991.

[5] Knuth, D.E., *The Art of Computer Programming, Vol.2: Seminumerical Algorithms*, Addison-Wesley, 1981.

[6] L'Ecuyer, P., *Efficient and Portable Combined Random Number Generators*, Communications of the ACM, Vol. 31, No. 6, pp. 742–774, June 1988.

[7] Leung, C.H.C., *Quantitative Analysis of Computer Systems*, John Wiley & Sons, 1988.

[8] Mitrani, I., *Simulation Techniques for Discrete-Event Systems*, Cambridge University Press, 1982.

[9] Obaidat, M.S., and Boudriga, N.A., *Performance Evaluation of Computer and Telecommunication Systems*, Wiley, 2010.

[10] Park, S.K. and Miller, K.W., *Random Number Generators: Good Ones Are Hard to Find*, Communications of the ACM, Vol. 31, No. 10, pp. 1192–1201, Oct. 1988.

[11] Ross, S.M., *Simulation* (Fourth edition), Academic Press, 2006.

[12] Schrage, L.E., *A More Portable Fortran Random Number Generator*, ACM Trans. on Mathematical Software, Vol. 5, No.2, pp. 132–138, 1979.

[13] Tausworthe, R.C., *Random Numbers Generated by Linear Recurrence Mod Two*, Mathematics of Computation, Vol. 19, pp. 201–209, 1965.

Κεφάλαιο 7

Η Μέθοδος της Προσομοίωσης

Σύνοψη

Εισάγονται οι βασικές αρχές της προσομοίωσης με έμφαση στην προσομοίωση διακριτών γεγονότων. Εξετάζονται η κατασκευή, επαλήθευση και επικύρωση του προγράμματος προσομοίωσης, η διαχείριση του χρόνου, καθώς και οι κατάλληλες τεχνικές και δομές δεδομένων για τη δημιουργία γεγονότων και την υλοποίηση της λίστας γεγονότων (λίστα, δέντρο, σωρός κλπ). Περιγράφονται τεχνικές συλλογής μετρήσεων καθώς και τεχνικές που χρησιμοποιούνται για την απαλοιφή του μεταβατικού φαινομένου κατά την εκτέλεση του προγράμματος. Εξετάζονται τα διάφορα κριτήρια τερματισμού και οι σχετικές τεχνικές για τη στατιστική ανάλυση των αποτελεσμάτων και τον υπολογισμό διαστημάτων εμπιστοσύνης με εκτίμηση της διασποράς (ανεξάρτητες επαναλήψεις, τμηματικές μέσες τιμές, αναγεννητική μέθοδος). Παρουσιάζονται οι κυριότερες τεχνικές μείωσης της διασποράς. Δίνονται παραδείγματα προγράμματος προσομοίωσης και περιγράφονται ορισμένες από τις πιο σημαντικές γλώσσες προσομοίωσης.

Η *προσομοίωση* (simulation) είναι μία μέθοδος, η οποία μας επιτρέπει να κατασκευάσουμε ένα αφηρημένο μοντέλο της πραγματικότητας και να παρακολουθήσουμε την εξέλιξή του στο χρόνο με τη βοήθεια ενός ψηφιακού υπολογιστή (ψηφιακή προσομοίωση). Πρόκειται για την πιο δημοφιλή μέθοδο ανάλυσης υπολογιστικών συστημάτων και αποτελεί τη λογική επιλογή στις περιπτώσεις όπου η χρήση αναλυτικών λύσεων ή προσεγγίσεων φαίνεται ανεπαρκής. Συνεπάγεται, όμως, σημαντικό υπολογιστικό κόστος και θα πρέπει να χρησιμοποιείται σωστά για να δίνει αξιόπιστα αποτελέσματα [3, 1, 6, 5].

Για την κατασκευή ενός μοντέλου χρειάζεται να καθοριστούν όλες οι συνιστώσες του πραγματικού συστήματος, οι οποίες θεωρούνται απαραίτητες για την περιγραφή της λειτουργίας του, καθώς και οι αλληλεπιδράσεις τους. Με τις συνιστώσες αυτές, τις οποίες ονομάζουμε *οντότητες* (entities), συνδέονται ορισμένα *χαρακτηριστικά* (attributes) τα οποία περιγράφουν την κατάσταση κάθε οντότητας. Η συλλογή όλων των χαρακτηριστικών τη χρονική στιγμή t ορίζει την κατάσταση του συστήματος $\boldsymbol{S}(t)$, η οποία γενικά παριστάνεται ως ένα διάνυσμα τυχαίων μεταβλητών. Μία συγκεκριμένη υλοποίηση του $\boldsymbol{S}(t)$ σε κάποιο διάστημα T $\{\boldsymbol{S}(t),\ \ 0 \leq t \leq T\}$ ονομάζεται *δειγματικό μονοπάτι* (sample path) για την περίοδο παρατήρησης T. Κάθε εκτέλεση ενός προγράμματος προσομοίωσης αντιστοιχεί σε ένα δειγματικό μονοπάτι για κάποια περίοδο παρατήρησης. Γενικά, η κατάσταση του συστήματος αλλάζει στη διάρκεια της περιόδου παρατήρησης. Οι αλλαγές κατάστασης του συστήματος ονομάζονται *γεγονότα*.

Ένα σύστημα είναι *συνεχές* όταν η κατάστασή του μεταβάλλεται συνεχώς με το χρόνο. Συνήθως τέτοια συστήματα περιγράφονται από συστήματα διαφορικών εξισώσεων και τα δειγματικά μονοπάτια (συναρτήσεις που ικανοποιούν τις εξισώσεις) καθορίζονται αποκλειστικά από τις οριακές συνθήκες. Αντίθετα, τα *διακριτά* συστήματα χαρακτηρίζονται από πεπερασμένες μεταβολές της κατάστασης σε διακεκριμένες χρονικές στιγμές (στιγμές γεγονότων). Τα συστήματα αυτά αναφέρονται ως *συστήματα διακριτών γεγονότων* (discrete event systems) και αποτελούν το αντικείμενο του ενδιαφέροντός μας σχετικά με τη μέθοδο της προσομοίωσης. Στα συστήματα αυτά ο χρόνος μπορεί να είναι συνεχής ή διακριτός. Στη δεύτερη περίπτωση είναι σαν να επιτρέπονται γεγονότα μόνο σε καθορισμένες (και συνήθως ισαπέχουσες) χρονικές στιγμές. Το χαρακτηριστικό ενός συστήματος διακριτών γεγονότων-συνεχούς χρόνου είναι ότι ένα δειγματικό μονοπάτι

καθορίζεται τελείως από την ακολουθία των χρονικών στιγμών γεγονότων και από τις αλλαγές κατάστασης που συμβαίνουν τις στιγμές αυτές. Ανάμεσα σε διαδοχικές στιγμές γεγονότων η κατάσταση του συστήματος μπορεί να μεταβάλλεται συνεχώς, αλλά η μεταβολή αυτή είναι απόλυτα καθορισμένη από τα γεγονότα και τις χρονικές στιγμές τους [2, 4, 6].

Ένα πρόγραμμα προσομοίωσης επιτελεί τρεις κύριες λειτουργίες:

(i) δημιουργεί δειγματικά μονοπάτια για το σύστημα,

(ii) συλλέγει στατιστικά στοιχεία για διάφορες ποσότητες που μας ενδιαφέρουν,

(iii) υπολογίζει καλές εκτιμήσεις για τους ζητούμενους δείκτες επίδοσης.

Συνήθως, το τμήμα του προγράμματος το οποίο σχετίζεται με την πρώτη λειτουργία αναφέρεται ως *προσομοιωτής* (simulator).

Για να είναι ικανοποιητικά τα αποτελέσματα μιας προσομοίωσης, το πρόγραμμα θα πρέπει να έχει πρόσβαση σε κατάλληλα δεδομένα εισόδου. Μία επιλογή θα ήταν να τροφοδοτείται το πρόγραμμα από δεδομένα που προέρχονται από μετρήσεις σε ένα πραγματικό σύστημα. Στην περίπτωση αυτή λέμε ότι η προσομοίωση οδηγείται από αποτυπώματα μετρήσεων (trace-driven simulation). Για να πραγματοποιηθεί αυτό, θα πρέπει αφενός να υπάρχει πρόσβαση στο περιβάλλον του συστήματος ή δυνατότητα δημιουργίας αυτού του περιβάλλοντος, και αφετέρου τα δεδομένα των μετρήσεων να επαρκούν για την προβλεπόμενη διάρκεια της προσομοίωσης. Συνήθως οι συνθήκες αυτές δεν ικανοποιούνται εύκολα και, επιπλέον, δεν εξασφαλίζουν την πλήρη κατανόηση της συμπεριφοράς του συστήματος. Για το λόγο αυτό, καταφεύγουμε στη δημιουργία τεχνητών δεδομένων με βάση στατιστικά μοντέλα της πραγματικότητας, οπότε έχουμε *αυτο-οδηγούμενη προσομοίωση* (self-driven simulation). Στην περίπτωση αυτή, τα δεδομένα δημιουργούνται με τη βοήθεια *γεννητριών τυχαίων αριθμών*.

7.1 Ανάπτυξη του Προγράμματος Προσομοίωσης

Η ανάπτυξη του προγράμματος προσομοίωσης πρέπει να ακολουθεί τις γενικές αρχές της τεχνολογίας λογισμικού και να καλύπτει τις βασικές απαιτήσεις της *επαλήθευσης* και *επικύρωσης*. Η ποιότητα του μοντέλου προσομοίωσης εκφράζεται με την πιστότητα με την οποία περιγράφει την πραγματικότητα. Η επικύρωση του μοντέλου αφορά τον έλεγχο της καταλληλότητας των υποθέσεων στις οποίες έχει βασιστεί το μοντέλο για να παραστήσει την πραγματικότητα. Η επαλήθευση αφορά τον έλεγχο του κατά πόσο οι υποθέσεις αυτές έχουν αποτυπωθεί σωστά κατά την υλοποίηση του μοντέλου, δηλαδή σχετίζεται με τη διόρθωση λαθών (debugging).

Κατά την επαλήθευση χρησιμοποιούνται όλες οι γνωστές τεχνικές ανάπτυξης, διόρθωσης και συντήρησης μεγάλων προγραμμάτων. Ειδικότερα, δεδομένου ότι η προσομοίωση αναπαριστά τη λειτουργία ενός συστήματος, μπορούν να χρησιμοποιηθούν διάφοροι μηχανισμοί ελέγχου είτε γενικοί είτε εξαρτώμενοι από τα χαρακτηριστικά του συστήματος. Συνήθως, οι τεχνικές αυτές βασίζονται στην παρακολούθηση ενός αποτυπώματος (trace) του προγράμματος, σε γραφικές απεικονίσεις, σε δοκιμές που αφορούν απλουστευμένες ή οριακές περιπτώσεις, στον έλεγχο συνθηκών λογικής ασυνέπειας, σε ελέγχους σχετικούς με τη συμπεριφορά των τυχαίων μεταβλητών κλπ.

Οι τεχνικές επικύρωσης σχετίζονται με τις υποθέσεις του μοντέλου, τις τιμές των παραμέτρων εισόδου, τις κατανομές των τυχαίων μεταβλητών, τις τιμές και τα χαρακτηριστικά των αποτελεσμάτων εξόδου. Συνήθως, η διαδικασία της επικύρωσης είτε βασίζεται απευθείας στην πείρα και τη διαισθητική ικανότητα του αναλυτή, είτε στη σύγκριση των αποτελεσμάτων του προγράμματος προσομοίωσης με δεδομένα από μετρήσεις σε πραγματικό σύστημα ή με τα αποτελέσματα της επίλυσης αναλυτικών μοντέλων.

7.1.1 Η Διαχείριση του Χρόνου

Η δομή ενός προγράμματος προσομοίωσης εξαρτάται από τα χαρακτηριστικά του συστήματος και από τα στατιστικά δεδομένα για τα οποία ενδιαφερόμαστε. Γενικά, όμως, θα πρέπει να λαμβάνεται υπόψη το πέρασμα του χρόνου, το οποίο παριστάνεται με την αύξηση της τιμής ενός «ρολογιού» [2, 4]. Διακρίνουμε δύο βασικούς τύπους προσομοίωσης ανάλογα με την τεχνική διαχείρισης του χρόνου.

- *Σύγχρονη προσομοίωση* ή προσομοίωση οδηγούμενη από το ρολόι (clock-driven simulation). Η τιμή του ρολογιού αυξάνεται κατά σταθερά βήματα, ελέγχονται τα γεγονότα που συνέβησαν σε κάθε βήμα και γίνονται οι αντίστοιχες ενέργειες. Η τεχνική αυτή είναι κατάλληλη για συστήματα διακριτού χρόνου, αλλά μπορεί να αποδειχθεί ανεπαρκής για συστήματα συνεχούς χρόνου.

- *Ασύγχρονη προσομοίωση* ή προσομοίωση οδηγούμενη από τα γεγονότα (event-driven simulation). Η τιμή του ρολογιού αυξάνει από τη μία στιγμή γεγονότος στην επόμενη, άρα κατά βήματα μεταβλητού μήκους, και σε κάθε στιγμή γεγονότος γίνονται οι κατάλληλες ενέργειες. Η τεχνική αυτή χρησιμοποιείται σε όλες τις γλώσσες προσομοίωσης για συστήματα διακριτών γεγονότων.

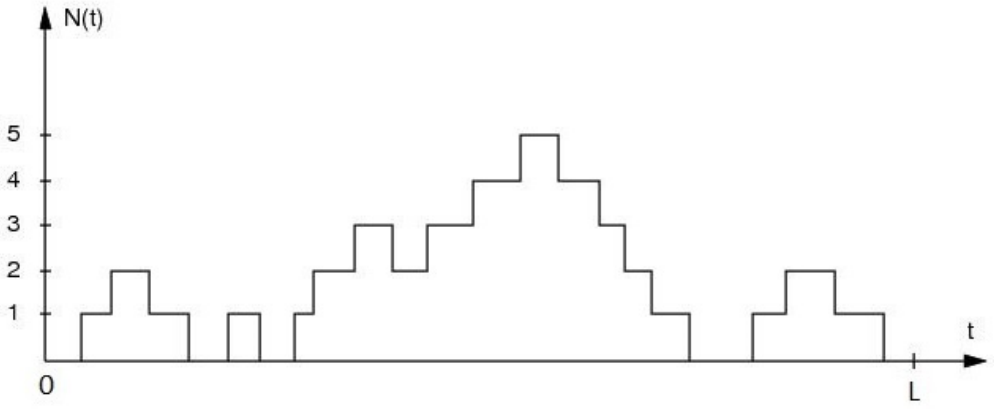
7.1.2 Χρονοδρομολόγηση

Η διαχείριση των γεγονότων και η αντίστοιχη ενημέρωση της κατάστασης του συστήματος μπορούν να γίνουν με διάφορους τρόπους, ανάλογα με τους οποίους μπορεί να ποικίλλει ριζικά η φιλοσοφία του προγράμματος προσομοίωσης[4, 6].

(i) *Χρονοδρομολόγηση γεγονότων* (event scheduling)
Σύμφωνα με τη μέθοδο αυτή υπάρχει ένα υποπρόγραμμα συνδεδεμένο με κάθε τύπο γεγονότος, το οποίο καλείται κάθε φορά που συμβαίνει το αντίστοιχο γεγονός. Το πρόγραμμα τηρεί ένα είδος «ημερολογίου», το οποίο ονομάζεται *λίστα γεγονότων* και περιλαμβάνει κάθε στιγμή τα γεγονότα που είναι προγραμματισμένα να συμβούν διατεταγμένα σύμφωνα με τις αντίστοιχες χρονικές στιγμές. Συνήθως, η λίστα περιλαμβάνει τόσα γεγονότα, όσοι και οι τύποι γεγονότων στο σύστημα. Κάθε φορά το ρολόι παίρνει ως τιμή τη χρονική στιγμή του πλησιέστερου γεγονότος (αρχή της λίστας) και καλεί το υποπρόγραμμα του αντίστοιχου τύπου. Το τελευταίο ενημερώνει την κατάσταση του συστήματος, συλλέγει στατιστικά στοιχεία και ενδεχομένως δρομολογεί νέα γεγονότα ενημερώνοντας αντίστοιχα τη λίστα γεγονότων.

(ii) *Χρονοδρομολόγηση διεργασιών* (process scheduling)
Μία διεργασία ορίζεται ως μία ακολουθία γεγονότων, καθένα από τα οποία συνοδεύεται από ένα σύνολο ενεργειών, π.χ. η ακολουθία των αφίξεων σε ένα σύστημα αναμονής. Κάθε γεγονός μπορεί να ανήκει σε περισσότερες από μία διεργασίες, αλλά κάθε ενέργεια ανήκει μόνο σε μία διεργασία. Έτσι, η συμπεριφορά του συστήματος μπορεί να παρασταθεί από ένα σύνολο διεργασιών, η συγχώνευση των οποίων θα έδινε την ακολουθία όλων των γεγονότων που συμβαίνουν στο σύστημα. Βέβαια, υπάρχει το ρολόι, το οποίο προχωρεί από γεγονός σε γεγονός, καθώς και μία λίστα (ισοδύναμη με τη λίστα γεγονότων) η οποία περιλαμβάνει τις προγραμματισμένες διεργασίες διατεταγμένες σύμφωνα με τις αντίστοιχες χρονικές στιγμές ενεργοποίησής τους (στιγμές γεγονότων). Κάθε φορά επιλέγεται η διεργασία στην αρχή της λίστας και εκτελούνται οι αντίστοιχες ενέργειες. Κάθε διεργασία μπορεί να δρομολογεί ή να ακυρώνει άλλες διεργασίες ή και τον εαυτό της. Οι διεργασίες αυτές μπορούν να υλοποιηθούν με τη χρήση συρρουτινών (coroutines), δημιουργώντας έτσι ένα είδος παραλληλίας στην εκτέλεση του προγράμματος.

Όπως είναι φανερό, το μεγαλύτερο μέρος της δραστηριότητας ενός προσομοιωτή είναι αφιερωμένο στη διαχείριση της δομής δεδομένων που χρησιμοποιείται από τον μηχανισμό χρονοδρομολόγησης. Η λίστα γεγονότων θα μπορούσε καταρχήν να υλοποιηθεί με τη βοήθεια ενός *πίνακα* (array), έτσι ώστε κάθε φορά να αναζητείται το μικρότερο στοιχείο του πίνακα που θα αντιστοιχεί στη χρονική στιγμή του επόμενου γεγονότος, ενώ η θέση του στοιχείου αυτού θα δηλώνει τον τύπο του γεγονότος. Μία άλλη δομή δεδομένων, ίσως η πιο ενδεδειγμένη, θα ήταν η *διατεταγμένη γραμμική λίστα* (απλά ή διπλά συνδεδεμένη). Κάθε στοιχείο της λίστας θα πρέπει να περιέχει οπωσδήποτε τη χρονική στιγμή και τον τύπο του αντίστοιχου γεγονότος. Χρειάζονται οι διαδικασίες για εισαγωγή ενός νέου στοιχείου στην κατάλληλη θέση και εξαγωγή του εκάστοτε πρώτου στοιχείου της λίστας που θα αντιστοιχεί στο επόμενο γεγονός. Οι λειτουργίες αυτές χαρακτηρίζουν την αφηρημένη δομή δεδομένων που ονομάζεται *ουρά προτεραιότητας* (priority queue). Σε προσομοιώσεις με πολλούς τύπους γεγονότων, μπορούν να αναζητηθούν δομές δεδομένων με καλύτερες επιδόσεις από τη γραμμική λίστα, δηλαδή δομές στις οποίες η εισαγωγή στοιχείων γίνεται σε χρόνο μικρότερο του $O(N)$, αν N είναι ο αριθμός στοιχείων στη λίστα. Αναφέρουμε τη *δεικτοδοτημένη γραμμική*



Σχήμα 7.1: *Αριθμός πελατών στο σύστημα κατά την περίοδο παρατήρησης.*

λίστα (indexed linear list) με ένα ή δύο επίπεδα κλειδιών, καθώς και διάφορους τύπους *δυαδικών δέντρων* με ιδιαίτερη χρησιμότητα, όπως είναι π.χ. ο *σωρός* (heap).

7.1.3 Συλλογή Δεδομένων

Εκτός από το μηχανισμό διαχείρισης της λίστας γεγονότων, το πρόγραμμα προσομοίωσης πρέπει να συλλέγει στατιστικά στοιχεία που θα χρησιμεύσουν στην εκτίμηση δεικτών επίδοσης του συστήματος. Θα αναφερθούμε ενδεικτικά στη συλλογή στατιστικών στοιχείων για τους βασικούς δείκτες επίδοσης ενός απλού συστήματος αναμονής, χωρίς να ασχοληθούμε, προς το παρόν, με τη στατιστική ανάλυση αυτών των στοιχείων.

Το Σχήμα 7.1 παριστάνει ένα δειγματικό μονοπάτι για την περίοδο παρατήρησης L (διάρκεια προσομοίωσης). Η κατάσταση του συστήματος κάθε στιγμή t, περιγράφεται από τον αριθμό $N(t)$ των πελατών στο σύστημα (σε αναμονή ή εξυπηρέτηση).

- Ο *βαθμός χρησιμοποίησης* (utilization) ορίζεται ως το ποσοστό του χρόνου κατά το οποίο μία μονάδα εξυπηρέτησης είναι απασχολημένη. Μπορούμε να εκτιμήσουμε την ποσότητα αυτή αθροίζοντας όλες τις περιόδους απασχόλησης της μονάδας εξυπηρέτησης και διαιρώντας το άθροισμα με το L:

$$U = \frac{1}{L}\int_0^L B(t)\,dt \tag{7.1}$$

  όπου

$$B(t) = \begin{cases} 1 & N(t) > 0 \\ 0 & N(t) = 0 \end{cases}$$

  Η άθροιση μπορεί να γίνει εύκολα, αν κάθε φορά που αρχίζει μία περίοδος απασχόλησης σημειώνουμε το χρόνο, για να τον αφαιρέσουμε αργότερα από το χρόνο λήξης της περιόδου και να προσθέσουμε τη διαφορά στο άθροισμα. Αν ο σταθμός εξυπηρέτησης περιλαμβάνει c μονάδες εξυπηρέτησης, μπορούμε να αθροίσουμε τις περιόδους απασχόλησης για όλες τις μονάδες και να εκτιμήσουμε τον βαθμό χρησιμοποίησης κάθε μονάδας διαιρώντας το συνολικό άθροισμα με το γινόμενο cL.

- Ο *ρυθμός απόδοσης* (throughput) μπορεί να εκτιμηθεί αν μετρήσουμε τον συνολικό αριθμό K των πελατών που πέρασαν από το σύστημα και τον διαιρέσουμε με το L:

$$\lambda = K/L \tag{7.2}$$

- Ο *μέσος αριθμός πελατών* στο σύστημα μπορεί να εκτιμηθεί εύκολα αν διαιρέσουμε το ολοκλήρωμα της συνάρτησης $N(t)$ με το L:

$$E[N] = \frac{1}{L}\int_0^L N(t)\,dt \tag{7.3}$$

Το ολοκλήρωμα ισοδυναμεί με το εμβαδόν της επιφάνειας κάτω από την καμπύλη $N(t)$, και μπορεί να υπολογιστεί όπως και το άθροισμα των περιόδων απασχόλησης. Κάθε φορά που αλλάζει η τιμή του N, σημειώνουμε το χρόνο για να τον αφαιρέσουμε αργότερα από το χρόνο της επόμενης αλλαγής. Η διαφορά αυτή πολλαπλασιάζεται με την προηγούμενη τιμή του N και το γινόμενο προστίθεται στο άθροισμα που θα δώσει τελικά το ζητούμενο ολοκλήρωμα.

- Τέλος, ο *μέσος χρόνος απόκρισης* εκτιμάται εύκολα, με χρήση του τύπου του Little:

$$T = \frac{E[N]}{\lambda} = \frac{1}{K}\int_0^L N(t)\,dt \tag{7.4}$$

διαιρώντας το ολοκλήρωμα της συνάρτησης $N(t)$ με τον συνολικό αριθμό K των πελατών που πέρασαν από το σύστημα.

Στο σημείο αυτό θα πρέπει να σημειώσουμε την ομοιότητα της τεχνικής για την εκτίμηση μέτρων επίδοσης με τη μέθοδο που ακολουθείται στη ντετερμινιστική ανάλυση ενός συστήματος αναμονής.

Παρακάτω (Αλγόριθμος 7.1) παρουσιάζεται σε γενική διατύπωση ο πυρήνας του προγράμματος προσομοίωσης. Κάθε πρόγραμμα προσομοίωσης με χρονοδρομολόγηση γεγονότων αποτελεί συγκεκριμενοποίηση του βασικού αυτού σχήματος.

Αλγόριθμος 7.1. *Ο γενικός αλγόριθμος της προσομοίωσης*

- αρχικοποίηση μεταβλητών κατάστασης
- αρχικοποίηση χρόνου
- δρομολόγηση πρώτου γεγονότος
- εφόσον (δεν ισχύει η συνθήκη τερματισμού) επανάλαβε:
  - εξαγωγή του επόμενου γεγονότος από τη λίστα
  - προχώρηση του χρόνου
  - εκτέλεση ειδικών πράξεων
  - ενημέρωση μεταβλητών κατάστασης
  - συλλογή μετρήσεων
  - ενημέρωση στατιστικών στοιχείων
  - δρομολόγηση νέων γεγονότων που προκαλούνται από το τρέχον
- προβολή αποτελεσμάτων

Παράδειγμα 7.1. *Πελάτες φθάνουν σε ένα κατάστημα ηλεκτρονικών ειδών σύμφωνα με διαδικασία Poisson με ρυθμό* 112 *αφίξεις/ώρα. Κάθε πελάτης με πιθανότητα* $0,45$ *αγοράζει ένα τεμάχιο μιας συγκεκριμένης συσκευής που διατίθεται σε προσφορά, με πιθανότητα* $0,2$ *αγοράζει δύο τεμάχια της συσκευής αυτής και με πιθανότητα* $0,35$ *δεν αγοράζει κανένα τεμάχιο. Το αρχικό απόθεμα της εν λόγω συσκευής είναι* 500 *τεμάχια. Αν διαθέτουμε γεννήτρια τυχαίων αριθμών, ομοιόμορφα κατανεμημένων στο* $(0,1)$*, να γραφτεί σε γλώσσα (ή ψευδογλώσσα) προγραμματισμού αλγόριθμος που προσομοιώνει τη λειτουργία του καταστήματος μέχρι να εξαντληθεί το απόθεμα της συσκευής και προσδιορίζει τον συνολικό αριθμό των πελατών που αγόρασαν τη συσκευή και το διάστημα μέχρι την εξάντληση του αποθέματος.*

Το παράδειγμα αποτελεί απλοποιημένο στιγμιότυπο του γενικού αλγορίθμου προσομοίωσης. Στο σύστημα συμβαίνει ένας τύπος γεγονότος: άφιξη, και συνεπώς η λίστα γεγονότων ανάγεται στον χρόνο της επόμενης άφιξης. Σε κάθε γεγονός πραγματοποιούνται οι κατάλληλες πράξεις και ενημερώσεις. Ακολουθεί το πρόγραμμα σε ψευδογλώσσα τύπου C.

```
clock=0.0;   /* μεταβλητή χρόνου */
n=0;         /* μετρητής */
s=500;       /* μεταβλητή κατάστασης */
over=0;      /* συνθήκη τερματισμού */
while (!over) {
clock=clock-log(random())/112.0;    /* επόμενο γεγονός */
U=random();
if (U<0.45)                         /* επιλογή ενέργειας */
x=1;
else if (U<0.65)
x=2;
else
x=0;
if (x<=s) {
s=s-x;
if (x>0) n++;
}
else over=1;
}
```

□

Παράδειγμα 7.2. *Θέλουμε να προσομοιώσουμε την ουρά αναμονής μπροστά σε μια αυτόματη ταμειολογιστική μηχανή. Οι χρόνοι μεταξύ αφίξεων ακολουθούν εκθετική κατανομή με παράμετρο* $1/3$ $\min^{-1}$ *και οι χρόνοι εξυπηρέτησης είναι ομοιόμορφα κατανεμημένοι μεταξύ* 2 *και* 4 min. *Δίνονται οι παρακάτω τυχαίοι αριθμοί* U_i *ομοιόμορφα κατανεμημένοι στο* $(0, 1)$*:*

0,71	0,47	0,26	0,76	0,46	0,91	0,12
0,40	0,13	0,59	0,92	0,09	0,78	0,53

Αρχικά το σύστημα είναι άδειο. Χρησιμοποιώντας την πρώτη σειρά τυχαίων αριθμών για τη δημιουργία αφίξεων και τη δεύτερη σειρά για τη δημιουργία εξυπηρετήσεων, να προσομοιωθεί η λειτουργία του συστήματος για διάστημα 10 min. *Με βάση την προσομοίωση να εκτιμηθούν:*
- ο μέσος αριθμός πελατών στο σύστημα,
- το ποσοστό του χρόνου π_1*, κατά το οποίο υπάρχει ένας πελάτης στο σύστημα,*
- το ποσοστό των πελατών q_1*, που κατά την άφιξή τους βρίσκουν έναν πελάτη στο σύστημα.*

Με χρήση των τυχαίων αριθμών U_i μπορούμε να δημιουργήσουμε τους χρόνους μεταξύ αφίξεων $A_i = -3\ln(U_i)$ και τους χρόνους εξυπηρέτησης $S_i = 2U_i + 2$, καταλήγοντας στα παρακάτω δεδομένα:

U_i	0,71	0,47	0,26	0,76	0,46	0,91	0,12
A_i	1,03	2,27	4,04	0,82	2,33	0,28	6,36
U_i	0,40	0,13	0,59	0,92	0,09	0,78	0,53
S_i	2,80	2,26	3,18	3,84	2,18	3,56	3,06

Στο σύστημα έχουμε δύο τύπους γεγονότων: άφιξη και τέλος εξυπηρέτησης (αναχώρηση). Η αλληλουχία των γεγονότων απεικονίζεται στον παρακάτω πίνακα. Η πρώτη στήλη αριθμεί τα γεγονότα, ενώ οι δύο επόμενες στήλες του πίνακα υλοποιούν κατ' ουσίαν τη λίστα γεγονότων, δηλαδή περιέχουν τη χρονική στιγμή για την οποία είναι δρομολογημένο το επόμενο γεγονός του κάθε τύπου. (Αν δεν υπάρχει δρομολογημένο γεγονός, θεωρούμε την τιμή άπειρο.) Σε κάθε γραμμή του πίνακα επιλέγεται το επόμενο γεγονός (με έντονα στοιχεία), δηλαδή το ελάχιστο μεταξύ των A και C, το οποίο αποτελεί τη νέα τιμή του ρολογιού. Στην τέταρτη στήλη δίνεται η νέα τιμή του αριθμού πελατών N, όπως διαμορφώνεται από το τρέχον γεγονός.

Γεγονός	A	C	N
1	**1,03**	∞	1
2	**3,30**	3,83	2
3	7,34	**3,83**	1
4	7,34	**6,09**	0
5	**7,34**	∞	1
6	**8,16**	10,52	2
	10,49	10,52	

Με βάση τις τιμές που περιέχονται στον πίνακα, είμαστε σε θέση να παρακολουθήσουμε την εξέλιξη του αριθμού πελατών συναρτήσει του χρόνου και να υπολογίσουμε τα στατιστικά στοιχεία που ζητούνται.
- Ο μέσος αριθμός πελατών στο σύστημα θα είναι ίσος με $E[N] = \frac{1}{10}\int_0^{10} N(t)\,dt$, δηλαδή το εμβαδόν της επιφάνειας κάτω από την καμπύλη $N(t)$ (σε αναλογία με το Σχήμα 7.1):
$E[N]$ = [(3,3-1,03)×1+(3,83-3,3)×2+(6,09-3,83)×1+(8,16-7,34)×1+(10,0-8,16)×2]/10,0 = 1,009.
- Το ποσοστό του χρόνου κατά το οποίο υπάρχει ένας πελάτης στο σύστημα υπολογίζεται με παρόμοιο τρόπο:
π_1 = [(3,3-1,03)+(6,09-3,83)+(8,16-7,34)]/10,0=0,535.
- Δύο από τους τέσσερις πελάτες βρίσκουν έναν πελάτη στο σύστημα κατά την άφιξή τους. Άρα: q_1 = 0,50.
[Όπως γνωρίζουμε, στη μόνιμη κατάσταση, θα ισχύει $\pi_1 = q_1$ ή γενικά $\pi_i = q_i$, $\forall i$.] □

7.2 Γλώσσες Προσομοίωσης

Για να πραγματοποιηθεί μία προσομοίωση, δεν χρειάζεται να διαθέτει κανείς κάποια γλώσσα προσομοίωσης. Λίγο ως πολύ, οποιαδήποτε γλώσσα προγραμματισμού γενικής χρήσης θα μπορούσε να χρησιμοποιηθεί. Απλώς, οι γλώσσες προσομοίωσης διευκολύνουν την ανάπτυξη του προγράμματος, καθόσον περιλαμβάνουν ως ενσωματωμένα στοιχεία τους τα βασικά χαρακτηριστικά της προσομοίωσης: διαχείριση οντοτήτων (αντικειμένων), διαχείριση χρόνου και γεγονότων, δημιουργία τυχαίων αριθμών, συλλογή στατιστικών δεδομένων καθώς και δυνατότητες αριθμητικών υπολογισμών. Οι σύγχρονες εκδόσεις πολλών γλωσσών προσομοίωσης αποτελούν ολοκληρωμένα πακέτα λογισμικού με περιβάλλον γραφικών, εξελιγμένους μηχανισμούς ελέγχου, και πληθώρα άλλων χαρακτηριστικών (visualization, animation).

Θα αναφερθούμε σύντομα σε ορισμένες από τις πλέον σημαντικές και δημοφιλείς γλώσσες προσομοίωσης που έχουν αναπτυχθεί ως τώρα. Στο Κεφάλαιο 10 θα γίνει εκτενέστερη αναφορά σε εργαλεία και γλώσσες μοντελοποίησης.

7.2.1 Simscript

Αναπτύχθηκε για πρώτη φορά στις αρχές της δεκαετίας του 1960 από τη RAND Corporation και από τότε έχει εξελιχθεί σημαντικά. Είναι οργανωμένη ιεραρχικά σε 5 επίπεδα. Τα τρία πρώτα επίπεδα παρέχουν τα χαρακτηριστικά μιας γλώσσας προγραμματισμού γενικής χρήσης του τύπου της Algol. Το τέταρτο επίπεδο αναφέρεται στη διαχείριση οντοτήτων, ενώ το πέμπτο παρέχει όλα τα υπόλοιπα χαρακτηριστικά της προσομοίωσης που αναφέρθηκαν πιο πάνω.

Η Simscript χρησιμοποιεί χρονοδρομολόγηση γεγονότων. Επομένως, ένα πρόγραμμα προσομοίωσης σε Simscript περιλαμβάνει ρουτίνες διαχείρισης των γεγονότων, οι οποίες είναι τόσες, όσοι και οι τύποι των γεγονότων και φέρουν το αντίστοιχο όνομα του τύπου. Ακόμη, περιλαμβάνει εντολές για δημιουργία, καταστροφή ή μετακίνηση οντοτήτων σε διατεταγμένα σύνολα (ουρές). Επίσης παρέχει δυνατότητες συλλογής στατιστικών δεδομένων και αριθμητικών υπολογισμών, καθώς και δημιουργίας ακολουθιών τυχαίων αριθμών.

Η σημερινή έκδοση της γλώσσας είναι η Simscript III, εμπορικό προϊόν με πληθώρα νέων χαρακτηριστικών. Συγκεκριμένα, η Simscript III είναι μια ισχυρή αντικειμενοστρεφής γλώσσα που περιλαμβάνει τάξεις και αντικείμενα, οντότητες, σύνολα, διεργασίες, διαχείριση ταυτοχρονισμού, μηχανισμό χρόνου, δυ-

νατότητες 2-Δ και 3-Δ γραφικών απεικονίσεων και video, καθώς και διάφορα εργαλεία μοντελοποίησης σε ένα ολοκληρωμένο προγραμματιστικό περιβάλλον.

7.2.2 GPSS

Η GPSS (General Purpose Simulation System) είναι βασικά ένα προιόν IBM (1961), αν και έχει επίσης υλοποιηθεί σε άλλα συστήματα. Η φιλοσοφία σχεδιασμού της στηρίζεται σε μία δομή από blocks. Προσωρινές οντότητες (transactions) δημιουργούνται, ακολουθούν μία πορεία μέσα στη δομή και τελικά καταστρέφονται. Για την προσομοίωση ενός συστήματος αναμονής οι ενεργές οντότητες αντιπροσωπεύουν τους πελάτες και τα παθητικά blocks αντιπροσωπεύουν τα διάφορα στάδια που ακολουθεί ένας πελάτης στο σύστημα (άφιξη, αναμονή, εξυπηρέτηση, αναχώρηση). Η γλώσσα χρησιμοποιεί χρονοδρομολόγηση διεργασιών (πολυνηματική υλοποίηση) και περιλαμβάνει διάφορα χαρακτηριστικά της προσομοίωσης, πάντα όμως σε σχέση με την ιδέα της μετακίνησης οντοτήτων μέσα σε ένα δίκτυο από blocks (transaction-flow modeling). Επιτρέπει επίσης τη συλλογή στατιστικών δεδομένων, καθώς και δυνατότητες αριθμητικού υπολογισμού. Το σύγχρονο εμπορικό προϊόν είναι η GPSS/H, η οποία έχει ενισχυθεί ώστε να αποτελεί μια πλούσια και ευέλικτη γλώσσα προγραμματισμού. Έχει εύχρηστη διεπαφή με τον χρήστη και μπορεί να διαχειριστεί μοντέλα πολύ μεγάλης κλίμακας.

7.2.3 Simula

Η γλώσσα Simula αναπτύχθηκε από τους K. Nygaard και O.-J. Dahl στο Πανεπιστήμιο του Όσλο (Norwegian Computing Center) στις αρχές της δεκαετίας του 1960. Αν και σχεδιάστηκε ως ειδική γλώσσα προσομοίωσης, η Simula κατέληξε να είναι μία πολύ εύχρηστη γλώσσα προγραμματισμού γενικής χρήσης. Στηρίζεται σε μία ελαφρά τροποποιημένη μορφή της Algol 60, την οποία περιέχει ως υποσύνολο, και περιλαμβάνει χαρακτηριστικά όπως η τάξη (class) και τα αντικείμενα (objects). Η Simula μπορεί να θεωρηθεί η πρώτη αντικειμενοστρεφής γλώσσα προγραμματισμού. Χρησιμοποιώντας τάξεις μπορεί κανείς να δημιουργήσει πολύπλοκα ιεραρχικά δεδομένα και δομές προγράμματος, επεκτείνοντας έτσι τις δυνατότητες της γλώσσας. Δύο τέτοιες επεκτάσεις ενσωματωμένες στο σύστημα είναι οι τάξεις simset και simulation. Η πρώτη παρέχει δυνατότητες διαχείρισης οντοτήτων και συνόλων ενώ η δεύτερη αφορά παράλληλες διεργασίες, διαχείριση του χρόνου και χρονοδρομολόγηση. Επίσης περιλαμβάνει διάφορες γεννήτριες τυχαίων αριθμών. Όπως και η GPSS, η Simula στηρίζεται στη χρονοδρομολόγηση διεργασιών.

Ένα πρόγραμμα Simula έχει τη μορφή ενός block, που περιέχει δηλώσεις και εντολές, και κάθε εντολή μπορεί να είναι με τη σειρά της ένα block. Ένα πρόγραμμα προσομοίωσης είναι ένα block που συνοδεύεται από το πρόθεμα simulation και περιλαμβάνει τη δήλωση διεργασιών, οι οποίες δημιουργούνται και ενεργοποιούνται κατά τη διάρκεια της εκτέλεσης του προγράμματος. Το κύριο πρόγραμμα συμπεριφέρεται σαν διεργασία, η οποία μπορεί να χρονοδρομολογηθεί όπως όλες οι άλλες. Η προσομοίωση αρχίζει τοποθετώντας το κύριο πρόγραμμα στην αρχή της λίστας γεγονότων τη χρονική στιγμή 0 και τελειώνει όταν το κύριο πρόγραμμα ολοκληρώνει τις ενέργειές του.

Η Simula παραμένει μια γλώσσα δημοφιλής σε ακαδημαϊκούς κύκλους. Διατίθεται ελεύθερα και χρησιμοποιείται για τη διδασκαλία του προγραμματισμού σε αρκετά πανεπιστήμια. Ένα πρόβλημα είναι ότι δεν υπάρχει μεγάλη διαθεσιμότητα σε εκπαιδευτικό υλικό (βιβλία και υλοποιήσεις). Ο πλέον διαδεδομένος μεταγλωττιστής της Simula είναι γραμμένος σε C (Simula in C, CIM).

7.2.4 Γλώσσες Βασισμένες στην Java

Ένας αριθμός γλωσσών ή εργαλείων προσομοίωσης έχουν βασιστεί στην Java. Οι περισσότερες αποτελούν επεκτάσεις της Java, ενισχυμένες με δυνατότητες που υπήρχαν σε παλαιότερες γλώσσες προσομοίωσης, δίνοντας έμφαση σε διάφορα χαρακτηριστικά. Θα αναφερθούμε σε δύο τέτοιες υλοποιήσεις προερχόμενες από ακαδημαϊκά περιβάλλοντα. Και οι δύο βασίστηκαν σε υπάρχουσες βιβλιοθήκες προγραμμάτων C++.

Η SimJava είναι γλώσσα προσομοίωσης βασισμένη σε χρονοδρομολόγηση διεργασιών σε αναλογία με τις άλλες γλώσσες αυτού του τύπου. Αναπτύχθηκε στο Πανεπιστήμιο του Εδιμβούργου στα μέσα της δεκαετίας του 1990 και αποτελεί επέκταση της Java με δομές που επιτρέπουν τον ορισμό και την

εκτέλεση προσομοίωσης. Το πρόγραμμα είναι ένα σύνολο διεργασιών (νημάτων) που ονομάζονται οντότητες (entities) στην ορολογία της SimJava. Οι οντότητες εκφράζουν συμπεριφορές και επικοινωνούν μεταξύ τους με μεταβίβαση γεγονότων. Ο έλεγχος των διεργασιών, η διαχείριση του χρόνου και η δημιουργία των γεγονότων ελέγχονται από ένα κεντρικό σύστημα. Η γλώσσα είναι ενισχυμένη με καλής ποιότητας γεννήτρια τυχαίων αριθμών, δυνατότητες συλλογής δεδομένων και στατιστικής ανάλυσης, διαφόρων ειδών αυτοματισμούς και αποδοτικούς μηχανισμούς διεπαφής με τον χρήστη. Σημαντικό χαρακτηριστικό είναι η δυνατότητα αναπαράστασης των αντικειμένων της προσομοίωσης σε μορφή κινούμενων σχεδίων. Διατίθεται ελεύθερα.

Η JavaSIM αναπτύχθηκε στο Πανεπιστήμιο του Newcastle στα τέλη της δεκαετίας του 1990 και χρησιμοποιείται σε πολλούς οργανισμούς, ακαδημαϊκούς και μη. Πρόκειται για ένα πακέτο λογισμικού αντικειμενοστρεφούς τύπου βασισμένο στη C++, το οποίο παρέχει δυνατότητες όμοιες με αυτές της Simula. Περιλαμβάνονται μηχανισμοί ταυτοχρονισμού (νήματα), διαχείρισης οντοτήτων, συνόλων και προσομοίωσης ανάλογοι με τις τάξεις simset και simulation της Simula. Επίσης, παρέχονται ρουτίνες στατιστικής επεξεργασίας, γεννήτριες τυχαίων αριθμών και διαχείριση διακοπών (interrupts). Διατίθεται ελεύθερα μέσω υπηρεσιών ανοικτού κώδικα.

7.3 Ανάλυση των Αποτελεσμάτων της Προσομοίωσης

Μέχρι τώρα επικεντρώσαμε την προσοχή μας στην κατασκευή του προγράμματος προσομοίωσης, και ειδικότερα στη χρήση των μηχανισμών που επιτρέπουν τη δημιουργία δειγματικών μονοπατιών της λειτουργίας ενός συστήματος. Απομένει το εξίσου σημαντικό πρόβλημα της χρησιμοποίησης των αποτελεσμάτων αυτών για την εκτίμηση δεικτών επίδοσης του συστήματος. Είναι πολύ συνηθισμένο να αρκείται κανείς στην κατασκευή του εξομοιωτή, θεωρώντας ότι το έργο του έχει ολοκληρωθεί. Ατυχώς, όμως, είναι πολύ εύκολο να εξαχθούν λανθασμένα συμπεράσματα από μία προσομοίωση, όχι γιατί το πρόγραμμα δεν είναι σωστό, αλλά γιατί δεν έγινε σωστή ερμηνεία των αποτελεσμάτων του.

7.3.1 Αφαίρεση της Επίδρασης του Μεταβατικού Φαινομένου

Κατά την εκτέλεση πειραμάτων προσομοίωσης, ενδιαφερόμαστε συνήθως για τη συμπεριφορά του συστήματος σε κατάσταση ισορροπίας (μόνιμη κατάσταση). Στην περίπτωση αυτή δεν θα πρέπει να λαμβάνονται υπόψη αποτελέσματα του αρχικού μέρους της προσομοίωσης που συνήθως αφορούν τη μεταβατική κατάσταση [2].

Δεδομένου ότι δεν είναι δυνατό να οριστούν σαφώς τα όρια ανάμεσα στη μεταβατική και τη μόνιμη κατάσταση η αφαίρεση της μεταβατικής κατάστασης βασίζεται σε διάφορες τεχνικές ευρετικού χαρακτήρα. Μία απλή προσέγγιση συνίσταται στην εκτέλεση πειραμάτων μεγάλης διάρκειας, ώστε η επίδραση των αρχικών συνθηκών στα τελικά αποτελέσματα να είναι αμελητέα. Η μέθοδος αυτή κοστίζει πολύ, χωρίς να παρέχει καμία εγγύηση για το αποτέλεσμα, συνεπώς θα πρέπει να αποφεύγεται. Μία άλλη τεχνική βασίζεται στη χρήση αρχικών συνθηκών που βρίσκονται κοντά στην αναμενόμενη μόνιμη κατάσταση, ώστε να μειώνεται η διάρκεια του μεταβατικού φαινομένου.

Οι περισσότερες τεχνικές βασίζονται στην υπόθεση ότι στη μόνιμη κατάσταση μειώνεται η μεταβλητότητα των παρατηρήσεων. Στη συνέχεια αναφέρονται ορισμένες από τις τεχνικές αυτές.

- **Αποκοπή με βάση το εύρος τιμών**
  Σύμφωνα με τη μέθοδο αυτή, η μεταβλητότητα αποτιμάται ως το εύρος τιμών των παρατηρήσεων. Έχοντας m παρατηρήσεις αγνοούμε τις πρώτες l και υπολογίζουμε το ελάχιστο και το μέγιστο των υπόλοιπων $m-l$ παρατηρήσεων. Το βήμα αυτό επαναλαμβάνεται για $l = 1, \ldots, m-1$ μέχρις ότου η $(l+1)$-στή παρατήρηση να μην είναι ούτε το ελάχιστο ούτε το μέγιστο των υπόλοιπων παρατηρήσεων. Η τιμή αυτή του l μπορεί να ληφθεί ως το μήκος της μεταβατικής κατάστασης.

- **Αποκοπή με βάση τη συνολική μέση τιμή**
  Η μέθοδος αυτή βασίζεται στη μεταβολή της συνολικής μέσης τιμής όταν αποκόπτονται παρατηρήσεις

από την αρχή του δείγματος. Υποτίθεται ότι στη μόνιμη κατάσταση η μέση τιμή δεν αλλάζει σημαντικά με την αποκοπή αρχικών παρατηρήσεων.

Έστω ότι διαθέτουμε m παρατηρήσεις x_j, $j = 1, \ldots, m$. Για $l = 1, \ldots, m-1$, αποκόπτονται οι πρώτες l παρατηρήσεις και υπολογίζεται η μέση τιμή των υπολοίπων:

$$\overline{x}_l = \frac{1}{m-l} \sum_{j=l+1}^{m} x_j \tag{7.5}$$

Ως μήκος της μεταβατικής κατάστασης λαμβάνεται η τιμή του l από την οποία αρχίζει να σταθεροποιείται η μέση τιμή $\overline{x}_l$.

- **Μετατοπιζόμενη μέση τιμή** (moving average)
  Σύμφωνα με τη μέθοδο αυτή, υπολογίζεται η μέση τιμή των παρατηρήσεων σε ένα μετατοπιζόμενο χρονικό παράθυρο του οποίου το μέγεθος αυξάνει από βήμα σε βήμα. Ειδικότερα, όπως προηγουμένως, θεωρούμε m παρατηρήσεις και για $k = 1, 2, 3, \ldots$ υπολογίζουμε τις μετατοπιζόμενες μέσες τιμές:

$$\overline{x}_j = \frac{1}{2k+1} \sum_{l=j-k}^{j+k} x_l, \quad j = k+1, \ldots, m-k \tag{7.6}$$

  Η επανάληψη συνεχίζεται για αυξανόμενα μεγέθη παραθύρου, μέχρις ότου η καμπύλη μεταβολής της μετατοπιζόμενης μέσης τιμής $\overline{x}_j$ γίνει αρκετά ομαλή. Ως μήκος της μεταβατικής κατάστασης λαμβάνεται η τιμή του j από την οποία αρχίζει να σταθεροποιείται η μέση τιμή $\overline{x}_j$.

- **Τμηματικές μέσες τιμές** (Batch means)
  Εκτελείται ένα πείραμα προσομοίωσης πολύ μεγάλης διάρκειας και οι παρατηρήσεις χωρίζονται σε τμήματα ίσου μεγέθους. Η μέθοδος βασίζεται στη μεταβολή της διασποράς των τμηματικών μέσων τιμών.

  Έστω N ο συνολικός αριθμός παρατηρήσεων οι οποίες κατανέμονται σε n τμήματα μεγέθους m, όπου $n = \lfloor N/m \rfloor$. Συμβολίζουμε με x_{ij} την j-στη παρατήρηση του i-στου τμήματος. Για $m = 2, 3, \ldots$ υπολογίζουμε τη διασπορά των τμηματικών μέσων τιμών:

$$\mathrm{Var}(\overline{x}) = \frac{1}{n-1} \sum_{i=1}^{n} (\overline{x}_i - \overline{\overline{x}})^2 \tag{7.7}$$

  όπου

$$\overline{x}_i = \frac{1}{m} \sum_{j=1}^{m} x_{ij}, \quad i = 1, \ldots, n$$

  είναι οι τμηματικές μέσες τιμές και

$$\overline{\overline{x}} = \frac{1}{n} \sum_{i=1}^{n} \overline{x}_i$$

  είναι η συνολική μέση τιμή. Ως μήκος της μεταβατικής κατάστασης λαμβάνεται η τιμή του m από την οποία η διασπορά αρχίζει οριστικά να μειώνεται. Πράγματι, στο σημείο αυτό η επίδραση της μεταβατικής κατάστασης στη διασπορά των τμηματικών μέσων τιμών περιορίζεται μόνο στη μέση τιμή του πρώτου τμήματος και ελαττώνεται όσο μεγαλώνει το μέγεθος των τμημάτων.

7.3.2 Εκτίμηση Δεικτών Επίδοσης – Διαστήματα Εμπιστοσύνης

Χρησιμοποιώντας ορισμένες βασικές έννοιες της στατιστικής μπορούμε να αντιμετωπίσουμε με αυστηρό τρόπο το πρόβλημα της εκτίμησης διαφόρων δεικτών επίδοσης του υπό μελέτη συστήματος. Σκοπός μας είναι ο υπολογισμός διαστημάτων εμπιστοσύνης για τις ποσότητες που εκτιμώνται με βάση τα αποτελέσματα της προσομοίωσης [2, 4].

Έστω $\{X_i,\ 1 \le i \le n\}$, ένα δείγμα μεγέθους n (n ανεξάρτητες παρατηρήσεις) από έναν πληθυσμό του οποίου η κατανομή έχει μέση τιμή μ και διασπορά σ^2. Ορίζουμε τη *δειγματική μέση τιμή* (sample mean) ως τον αριθμητικό μέσο όρο:

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{7.8}$$

Παίρνοντας τη μέση τιμή των δύο μελών της παραπάνω σχέσης έχουμε:

$$E[\overline{X}] = \frac{1}{n}\sum_{i=1}^{n} E[X_i] = \frac{1}{n}\sum_{i=1}^{n} \mu = \mu$$

εφόσον οι τυχαίες μεταβλητές X_i, $i = 1, \ldots, n$, είναι ανεξάρτητες και με την ίδια κατανομή. Η μεταβλητή $\overline{X}$ αποτελεί μία *αμερόληπτη εκτίμηση* (unbiased estimate) του μ. (Μία εκτίμηση λέγεται αμερόληπτη, όταν η μέση τιμή της δίνει την ποσότητα που θέλουμε να εκτιμήσουμε.) Αντίστοιχα η διασπορά του $\overline{X}$ βρίσκεται:

$$\mathrm{Var}[\overline{X}] = \frac{1}{n^2}\sum_{i=1}^{n} \mathrm{Var}[X_i] = \frac{1}{n^2}\sum_{i=1}^{n} \sigma^2 = \frac{\sigma^2}{n}$$

δηλαδή τείνει στο 0 όταν αυξάνει το μέγεθος του δείγματος.

Οι παρατηρήσεις X_i που προέρχονται από προσομοίωση ακολουθούν κατά προσέγγιση κανονική κατανομή. Είδαμε σε προηγούμενη παράγραφο, ότι για τους διάφορους δείκτες επίδοσης, οι τιμές των X_i προκύπτουν από την άθροιση πολλών μικρών ποσοτήτων. Χρησιμοποιώντας μία μορφή του κεντρικού οριακού θεωρήματος, μπορούμε να ισχυριστούμε ότι κάθε παρατήρηση στο δείγμα είναι κανονικά κατανεμημένη με μέση τιμή μ και διασπορά σ^2. Εφόσον το άθροισμα κανονικά κατανεμημένων τυχαίων μεταβλητών είναι επίσης κανονικά κατανεμημένο, συμπεραίνουμε ότι η δειγματική μέση τιμή $\overline{X}$ ακολουθεί κανονική κατανομή με μέση τιμή μ και διασπορά σ^2/n. Άρα, η τυχαία μεταβλητή $U = (\overline{X} - \mu)/(\sigma/\sqrt{n})$ θα ακολουθεί τη μοναδιαία κανονική κατανομή με μέση τιμή 0 και διασπορά 1.

Η μεταβλητή U, όμως, είναι συνάρτηση όχι μόνο του μ αλλά και της παραμέτρου σ^2, η οποία γενικά δεν είναι γνωστή. Μία λογική λύση θα ήταν να εκτιμήσουμε τη διασπορά σ^2 από το δείγμα. Ορίζουμε, επομένως, τη *δειγματική διασπορά* (sample variance):

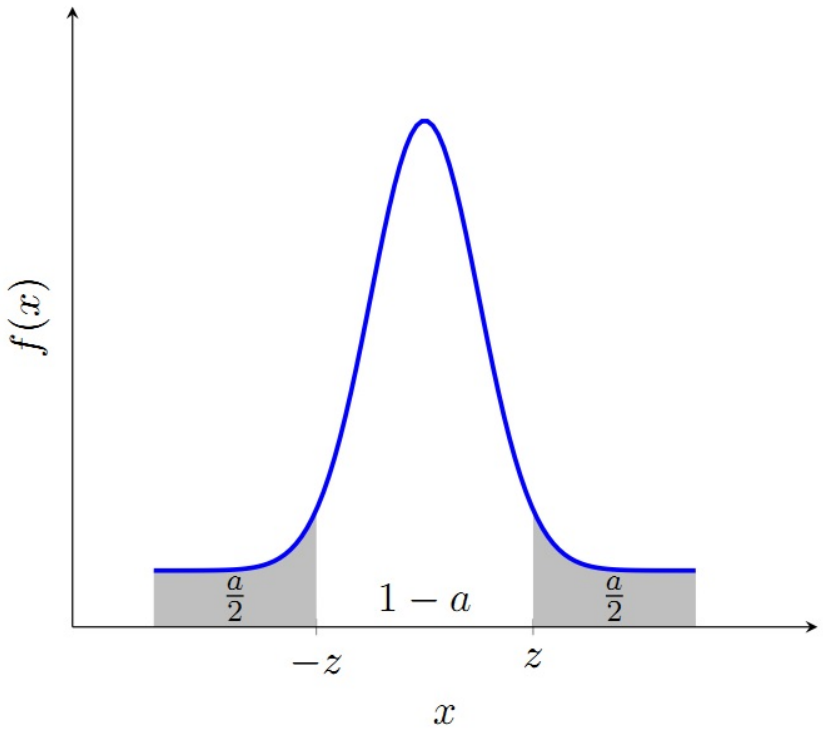
$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n} (X_i - \overline{X})^2 \tag{7.9}$$

Μπορούμε εύκολα να αποδείξουμε ότι το s^2 αποτελεί αμερόληπτη εκτίμηση του σ^2:

$$\begin{aligned} E[s^2] &= \frac{1}{n-1} E\left[\sum_{i=1}^{n} (X_i - \mu + \mu - \overline{X})^2\right] \\ &= \frac{1}{n-1}\left(\sum_{i=1}^{n} E\left[(X_i - \mu)^2\right] - nE\left[(\overline{X} - \mu)^2\right]\right) \\ &= \frac{1}{n-1}\left[n\sigma^2 - n(\sigma^2/n)\right] = \sigma^2 \end{aligned}$$

Επιπλέον, όταν οι παρατηρήσεις X_i ακολουθούν κανονική κατανομή, αποδεικνύεται ότι η τυχαία μεταβλητή $Z = (\overline{X} - \mu)/(s/\sqrt{n})$ ακολουθεί την κατανομή Student με $n-1$ βαθμούς ελευθερίας. Η συνάρτηση πυκνότητας πιθανότητας της κατανομής Student είναι συμμετρική ως προς τον κατακόρυφο άξονα, όπως αυτή της κανονικής κατανομής. Εξάλλου, για μεγάλες τιμές του n (πρακτικά για $n > 30$) η κατανομή Student προσεγγίζει τη μοναδιαία κανονική κατανομή.

Είμαστε τώρα σε θέση να αναζητήσουμε το *διάστημα εμπιστοσύνης* για την εκτίμηση $\overline{X}$ της ποσότητας μ. Η τεχνική που ακολουθούμε γενικά είναι να δημιουργήσουμε μία τυχαία μεταβλητή $Z(\overline{X}, \mu)$, η οποία είναι συνάρτηση του δείγματος και της άγνωστης παραμέτρου μ, και της οποίας γνωρίζουμε την κατανομή.



Σχήμα 7.2: *Διάστημα εμπιστοσύνης.*

Αν υπάρχει μία τέτοια μεταβλητή, μπορούμε να βρούμε δυο τιμές z_1 και z_2, τέτοιες ώστε $\Pr[z_1 \leq Z(\overline{X}, \mu) \leq z_2] = 1 - \alpha$, $(0 < \alpha < 1)$. Λύνοντας τις ανισότητες ως προς μ δημιουργούμε ένα διάστημα μέσα στο οποίο περιέχεται το μ με πιθανότητα $1-\alpha$. Η πιθανότητα $1-\alpha$ ονομάζεται *βαθμός εμπιστοσύνης* και ερμηνεύεται ως εξής: αν υπολογίσουμε ένα μεγάλο αριθμό διαστημάτων εμπιστοσύνης με βάση διαφορετικά δείγματα, τότε ένα ποσοστό $1-\alpha$ των διαστημάτων αυτών θα περιέχει την πραγματική τιμή του μ (Σχήμα 7.2).

Σύμφωνα με όσα αναφέραμε πιο πάνω, η μεταβλητή Z μπορεί να χρησιμοποιηθεί για τον υπολογισμό του διαστήματος εμπιστοσύνης. Η μεταβλητή αυτή ακολουθεί κατανομή Student με $n-1$ βαθμούς ελευθερίας ή προσεγγιστικά μοναδιαία κανονική κατανομή. Έστω, σε κάθε περίπτωση, $f(x)$ και $F(x)$ οι αντίστοιχες συναρτήσεις πυκνότητας πιθανότητας και κατανομής πιθανότητας. Αν θεωρήσουμε $z_1 = -z_2 = z = F^{-1}(y)$, αναζητούμε την τιμή του y για την οποία θα ισχύει:

$$\Pr[-z \leq Z \leq z] = F(z) - F(-z) = F(z) - [1 - F(z)] = 2F(z) - 1 = 2y - 1$$

εφόσον λόγω της συμμετρίας της f ως προς τον κατακόρυφο άξονα θα ισχύει γενικά $F(-x) = 1 - F(x)$. Βρίσκουμε τελικά $1 - \alpha = 2y - 1$ ή $y = 1 - \alpha/2$. Αν, λοιπόν,

$$z = F^{-1}(1 - \alpha/2) = z_{1-\alpha/2} \tag{7.10}$$

είναι το $(1-\alpha/2)$-ποσοστιαίο σημείο της κατανομής Student με $n-1$ βαθμούς ελευθερίας ή της μοναδιαίας κανονικής κατανομής για μεγάλο n, μπορούμε να γράψουμε για το διάστημα εμπιστοσύνης:

$$\Pr\left[-z_{1-\alpha/2} \leq \frac{\overline{X} - \mu}{s/\sqrt{n}} \leq z_{1-\alpha/2}\right] = 1 - \alpha$$

ή τελικά:

$$\Pr\left[\overline{X} - \frac{z_{1-\alpha/2} s}{\sqrt{n}} \leq \mu \leq \overline{X} + \frac{z_{1-\alpha/2} s}{\sqrt{n}}\right] = 1 - \alpha \tag{7.11}$$

Συνοψίζοντας, η διαδικασία υπολογισμού του διαστήματος εμπιστοσύνης είναι η εξής:

- Από το δείγμα $\{X_i,\ 1 \leq i \leq n\}$ υπολογίζονται η δειγματική μέση τιμή $\overline{X}$ και η δειγματική διασπορά s^2.

- Για τον επιθυμητό βαθμό εμπιστοσύνης $1-\alpha$, βρίσκεται από τους πίνακες της κατανομής Student με $n-1$ βαθμούς ελευθερίας ή της μοναδιαίας κανονικής κατανομής για μεγάλο n, η τιμή $z_{1-\alpha/2}$ για την οποία ισχύει $F(z_{1-\alpha/2}) = 1-\alpha/2$. (Ενδεικτικά για $1-\alpha = 0,9$ και 0,5 η κανονική κατανομή δίνει αντίστοιχα $z_{1-\alpha/2} = 1,65$ και 1,96.)

Το μισό του εύρους του διαστήματος εμπιστοσύνης $z_{1-\alpha/2}s/\sqrt{n}$ αποτελεί μέτρο της ακρίβειας της εκτίμησης. Παρατηρούμε ότι γενικά η ακρίβεια αυξάνει αντιστρόφως ανάλογα προς την τετραγωνική ρίζα του μεγέθους του δείγματος. Συνήθως, οι απαιτήσεις σε μία προσομοίωση καθορίζουν από πριν ότι για δεδομένο βαθμό εμπιστοσύνης το μισό του εύρους του διαστήματος δεν πρέπει να ξεπερνά κάποιο ποσοστό της τιμής $\overline{X}$ που εκτιμήθηκε. Για να το επιτύχουμε αυτό, μπορούμε να πραγματοποιούμε τον υπολογισμό του διαστήματος κατά καιρούς στη διάρκεια της προσομοίωσης μέχρις ότου ικανοποιηθεί η συνθήκη. Γενικά, όσο λιγότερες πληροφορίες διαθέτουμε τόσο πιο πλατύ θα είναι το διάστημα εμπιστοσύνης. Επίσης θα πρέπει να τονίσουμε το ρόλο της διασποράς s^2 στο πλάτος του διαστήματος εμπιστοσύνης. Υπάρχουν διάφορες τεχνικές μείωσης της διασποράς, οι οποίες μπορούν να εφαρμοστούν, ώστε η ίδια ακρίβεια να επιτυγχάνεται με λιγότερο φόρτο του προγράμματος προσομοίωσης.

7.3.3 Μείωση Διασποράς

Τα τυχαία φαινόμενα που χαρακτηρίζουν την επίδοση προκαλούν συνήθως σημαντική μεταβλητότητα των τιμών που παρατηρούνται σε ένα πείραμα προσομοίωσης. Το πρόβλημα μπορεί να αντιμετωπιστεί με αύξηση του αριθμού των παρατηρήσεων, ώστε να εξομαλυνθούν οι αποκλίσεις και να αυξηθεί η ακρίβεια. Αυτό, όμως, συνεπάγεται αύξηση του κόστους εκτέλεσης. Όπως είδαμε, το διάστημα εμπιστοσύνης είναι ανάλογο προς την τυπική απόκλιση του δείγματος. Επομένως, η ακρίβεια των αποτελεσμάτων θα βελτιωθεί, αν διαθέτουμε δείγμα με χαμηλή διασπορά (ή ισοδύναμα η ίδια ακρίβεια θα επιτευχθεί με λιγότερες παρατηρήσεις). Το θέμα έχει μελετηθεί εκτενώς τόσο σε σχέση με την προσομοίωση όσο και σε άλλα πεδία εφαρμογών. Στη συνέχεια θα αναφερθούμε εν συντομία σε μερικές αποδοτικές μεθόδους που χρησιμοποιούνται στο πλαίσιο της προσομοίωσης [4, 6]

7.3.3.1 Κοινές Ακολουθίες Τυχαίων Αριθμών

Ας υποθέσουμε ότι ενδιαφερόμαστε για τη σύγκριση των επιδόσεων ενός συστήματος όταν λειτουργεί με δύο διαφορετικούς τρόπους A και B. Έστω ότι η σύγκριση θα βασιστεί στον δείκτη επίδοσης α, οπότε ο στόχος θα είναι η εκτίμηση της διαφοράς $\alpha_A - \alpha_B$. Κατά τη συνήθη πρακτική, η σύγκριση γίνεται με βάση δύο ανεξάρτητα σύνολα παρατηρήσεων (ίσου μεγέθους n) που αντιστοιχούν στους δύο τρόπους. Συμβολίζουμε με $\overline{X}_A$ ($\overline{X}_B$) τη δειγματική μέση τιμή και με S_A^2 (S_B^2) τη δειγματική διασπορά του δείγματος A (B). Η εκτίμηση της ποσότητας $\alpha_A - \alpha_B$ θα είναι $\overline{X}_A - \overline{X}_B$. Η διασπορά της τελευταίας είναι $\mathrm{Var}\,[\overline{X}_A] + \mathrm{Var}\,[\overline{X}_B]$, της οποίας εκτίμηση θα είναι

$$(S_A^2 + S_B^2)/n$$

Ας θεωρήσουμε, τώρα, ότι τα δύο δείγματα A και B δεν είναι ανεξάρτητα, αλλά δημιουργήθηκαν με χρήση της ίδιας ακολουθίας τυχαίων αριθμών (ίδια φύτρα). Στο εσωτερικό καθενός από τα δείγματα οι παρατηρήσεις είναι ανεξάρτητες μεταξύ τους, αλλά υπάρχει θετική συσχέτιση ανάμεσα στις αντίστοιχες παρατηρήσεις των δύο συνόλων. Η διαφορά $\overline{X}_A - \overline{X}_B$ εξακολουθεί να είναι αμερόληπτη εκτίμηση του $\alpha_A - \alpha_B$. Η διασπορά της διαφοράς αυτής, όμως, θα είναι τώρα $\mathrm{Var}\,[\overline{X}_A] + \mathrm{Var}\,[\overline{X}_B] - 2\mathrm{Cov}\,[\overline{X}_A, \overline{X}_B]$, όπου $\mathrm{Cov}\,[\overline{X}_A, \overline{X}_B]$ είναι η συνδιασπορά των $\overline{X}_A$ και $\overline{X}_B$. Επομένως, η διασπορά της διαφοράς των μέσων τιμών μπορεί να εκτιμηθεί από την ποσότητα

$$(S_A^2 + S_B^2 - 2S_{AB})/n$$

όπου

$$S_{AB} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i^A - \overline{X}_A)(X_i^B - \overline{X}_B) \tag{7.12}$$

Είναι φανερό ότι έχει επέλθει μείωση της διασποράς

$$S_A^2 + S_B^2 - 2S_{AB} < S_A^2 + S_B^2 \tag{7.13}$$

Παρατηρούμε, συνεπώς, ότι, στην περίπτωση αυτή, η συσχέτιση των παρατηρήσεων είναι επιθυμητή και αξιοποιείται. Θα πρέπει να τονιστεί, όμως, ότι πρόκειται για συσχέτιση μεταξύ δειγμάτων και όχι στο εσωτερικό ενός δείγματος.

7.3.3.2 Αντιθετικές Μεταβλητές

Σε πολλές περιπτώσεις, η μείωση της διασποράς μπορεί να επιτευχθεί με κατάλληλη επιλογή των ακολουθιών τυχαίων αριθμών. Ας θεωρήσουμε ότι διαθέτουμε $2n$ παρατηρήσεις ως εξής: οι n εξ αυτών, $X_1, X_2, \ldots, X_n$, αντιστοιχούν στις ανεξάρτητες ακολουθίες τυχαίων αριθμών $\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_n$, όπου $\boldsymbol{U} = (U_1, U_2, \ldots, U_K)$. Υποθέτουμε ότι και οι υπόλοιπες n παρατηρήσεις, $X_1^*, X_2^*, \ldots, X_n^*$, αντιστοιχούν στις ανεξάρτητες ακολουθίες $\boldsymbol{U}_1^*, \boldsymbol{U}_2^*, \ldots, \boldsymbol{U}_n^*$. Ισχύει, όμως, ότι υπάρχει εξάρτηση μεταξύ $\boldsymbol{U}_1$ και $\boldsymbol{U}_1^*$, $\boldsymbol{U}_2$ και $\boldsymbol{U}_2^*$, κ.ο.κ. Αν $\overline{X}$ και $\overline{X}^*$ είναι οι αντίστοιχες δειγματικές μέσες τιμές, τότε η ποσότητα $\overline{Y} = (\overline{X} + \overline{X}^*)/2$ αποτελεί αμερόληπτη εκτίμηση της μέσης τιμής $\mathrm{E}[X_i] = \mathrm{E}[X_i^*]$. Η διασπορά του $\overline{Y}$ θα είναι:

$$\begin{aligned} \mathrm{Var}[\overline{Y}] &= \frac{1}{4}\{\mathrm{Var}[\overline{X}] + \mathrm{Var}[\overline{X}^*] + 2\mathrm{Cov}[\overline{X}, \overline{X}^*]\} \\ &= \frac{1}{2}\{\mathrm{Var}[\overline{X}] + \mathrm{Cov}[\overline{X}, \overline{X}^*]\} \end{aligned} \tag{7.14}$$

Αν δεν υπήρχε εξάρτηση μεταξύ $\boldsymbol{U}_i$ και $\boldsymbol{U}_i^*$ $(i = 1, 2, \ldots, n)$, η συνδιασπορά των $\overline{X}$ και $\overline{X}^*$ θα ήταν μηδενική και η διασπορά του $\overline{Y}$ θα ήταν ίση με $\mathrm{Var}[\overline{X}]/2$. Αν οι ακολουθίες $\boldsymbol{U}_i^*$ μπορούν να επιλεγούν έτσι ώστε τα $\overline{X}$ και $\overline{X}^*$ να εμφανίζουν αρνητική συσχέτιση, τότε θα έχουμε μείωση της διασποράς του $\overline{Y}$. Οι μεταβλητές που επιτυγχάνουν αρνητική συσχέτιση, και —κατά συνέπεια— μείωση της διασποράς, ονομάζονται *αντιθετικές μεταβλητές* (antithetic variables). Γενικά, όταν μια ακολουθία $\boldsymbol{U}$ έχει ως αποτέλεσμα τη συχνή εμφάνιση κάποιων γεγονότων, η αντίστοιχη ακολουθία $\boldsymbol{U}^*$ θα πρέπει να προκαλεί σπάνια εμφάνιση των συγκεκριμένων γεγονότων και αντίστροφα. Σε ένα σύστημα αναμονής αυτό μπορεί να συμβεί αν γίνει κάποιο είδος ανταλλαγής των φύτρων των ακολουθιών που αντιστοιχούν σε χρόνους μεταξύ αφίξεων και χρόνους εξυπηρέτησης.

7.3.3.3 Μεταβλητές Ελέγχου

Η γενική ιδέα πίσω από τις περισσότερες τεχνικές μείωσης της διασποράς είναι η αξιοποίηση διαθέσιμης πληροφορίας σχετικής με το υπό μελέτη σύστημα, προκειμένου οι παρατηρήσεις που θα εξαχθούν από την προσομοίωση να μην εμφανίζουν μεγάλη μεταβλητότητα. Ένα σαφές παράδειγμα χρήσης πρόσθετης πληροφορίας είναι η παρακάτω μέθοδος.

Διαθέτουμε ένα δείγμα n ανεξάρτητων και ισόνομων παρατηρήσεων, $X_1, X_2, \ldots, X_n$, από το οποίο θα εκτιμηθεί ο δείκτης επίδοσης $\alpha = \mathrm{E}[X_i]$.

Ας υποθέσουμε, επίσης, ότι έχουμε στη διάθεσή μας και ένα δεύτερο δείγμα n ανεξάρτητων και ισόνομων παρατηρήσεων, $Y_1, Y_2, \ldots, Y_n$, το οποίο έχει τις παρακάτω ιδιότητες: (α) Υπάρχει ισχυρή θετική συσχέτιση μεταξύ X_i και Y_i. (β) Γνωρίζουμε την προσδοκητή τιμή $\beta = \mathrm{E}[Y_i]$. Η πρόσθετη πληροφορία που διαθέτουμε μπορεί να χρησιμοποιηθεί για να προσδιορίσουμε μια εκτίμηση με χαμηλότερη διασπορά. Θεωρούμε την έκφραση

$$V = \overline{X} = V + \beta - \overline{Y}$$

όπου $\overline{X}$, $\overline{Y}$, αντίστοιχα, είναι οι δειγματικές μέσες τιμές των X_i και Y_i, οι οποίες αποτελούν άμεση εκτίμηση των α και β. Όμως, και το V είναι αμερόληπτη εκτίμηση του α:

$$\mathrm{E}[V] = \mathrm{E}[\overline{X}] + \beta - \mathrm{E}[\overline{Y}] = \alpha + \beta - \beta = \alpha$$

Η διασπορά του V θα έχει τιμή

$$\mathrm{Var}[V] = \mathrm{Var}[\overline{X}] + \mathrm{Var}[\overline{Y}] - 2\mathrm{Cov}[\overline{X}, \overline{Y}] \tag{7.15}$$

και θα είναι μικρότερη από τη διασπορά $\mathrm{Var}[\overline{X}]$ εφόσον $\mathrm{Cov}[\overline{X}, \overline{Y}] > 1/2\mathrm{Var}[\overline{Y}]$.

Η ποσότητα που παράγει το δείγμα των Y_i αναφέρεται ως *μεταβλητή ελέγχου*. Προκειμένου να ικανοποιήσει τις παραπάνω ιδιότητες (α) και (β), η μεταβλητή ελέγχου θα πρέπει να σχετίζεται στενά με τον δείκτη επίδοσης που θέλουμε να εκτιμήσουμε και να είναι αρκετά απλή, ώστε η προσδοκητή τιμή της να υπολογίζεται εύκολα.

Μια επιλογή είναι να χρησιμοποιηθεί ένας δείκτης επίδοσης παρόμοιος με τον προς εκτίμηση, αλλά σε ένα απλούστερο μοντέλο. Θεωρούμε, για παράδειγμα, ότι προσομοιώνουμε ένα σύστημα αναμονής με έναν εξυπηρετητή, στο οποίο οι χρόνοι μεταξύ αφίξεων και οι χρόνοι εξυπηρέτησης ακολουθούν κατανομές διάφορες της εκθετικής, και ισχύει κανονισμός διάφορος του FIFO. Έστω ότι θέλουμε να εκτιμήσουμε τον μέσο χρόνο παραμονής των εργασιών στο σύστημα. Η μεταβλητή ελέγχου θα μπορούσε να είναι ο μέσος χρόνος παραμονής Y σε ένα σύστημα $M/M/1$, που μπορούμε να υπολογίσουμε με άμεσο τρόπο. [Υπενθυμίζεται ότι για σύστημα $M/M/1$, ισχύει $\beta = 1/(\mu - \lambda)$, όπου λ και μ οι ρυθμοί αφίξεων και εξυπηρέτησης, αντίστοιχα.] Για να επιτύχουμε την επιθυμητή συσχέτιση, τα X_i και Y_i $(i = 1, 2, \ldots, n)$, παράγονται από πειράματα προσομοίωσης που χρησιμοποιούν την ίδια ακολουθία τυχαίων αριθμών, για κάθε i.

Μια άλλη προσέγγιση είναι να επιλέξουμε ένα κατάλληλο μέγεθος από το ίδιο το υπό μελέτη σύστημα. Μια τέτοια μεταβλητή ελέγχου ονομάζεται *συνακόλουθη* (concomitant). Συνεχίζοντας το προηγούμενο παράδειγμα, μπορούμε να λάβουμε ως συνακόλουθη μεταβλητή Y τον συνολικό χρόνο απασχόλησης του εξυπηρετητή σε ένα διάστημα παρατήρησης του συστήματος. Είναι προφανές ότι τα X_i και Y_i είναι συσχετισμένα, για κάθε i. Επιπλέον, μπορούμε εύκολα να υπολογίσουμε την προσδοκητή τιμή του Y: αν κάθε πείραμα προσομοίωσης εκτελείται μέχρι να ολοκληρωθεί η εξυπηρέτηση k πελατών,θα έχουμε $\beta = k\mathrm{E}[S]$, όπου S ο χρόνος εξυπηρέτησης. Η μέθοδος των μεταβλητών ελέγχου χαρακτηρίζεται από σχετικά μεγάλο κόστος υλοποίησης σε σχέση με τις άλλες τεχνικές που αναφέρθηκαν, αλλά φαίνεται ότι είναι και πιο αποτελεσματική.

7.3.4 Εκτίμηση Διασποράς – Κριτήρια Τερματισμού

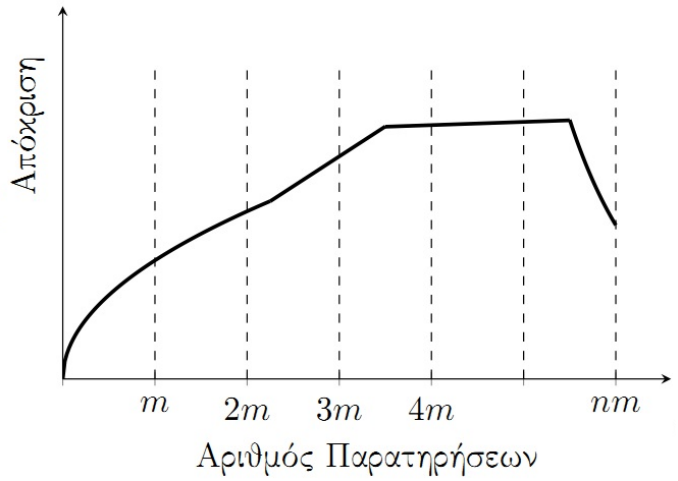
Μέχρι τώρα θεωρήσαμε ως δεδομένο ότι διαθέτουμε ένα δείγμα X από ανεξάρτητες παρατηρήσεις, πράγμα το οποίο δεν συμβαίνει πάντοτε. Θα αναφέρουμε στη συνέχεια διάφορες μεθόδους, οι οποίες εξασφαλίζουν κατά κάποιον τρόπο την ανεξαρτησία των παρατηρήσεων $\{X_i,\ 1 \leq i \leq n\}$.

7.3.4.1 Ανεξάρτητες Επαναλήψεις (Independent replications)

Μία λύση είναι να επαναλάβουμε πολλές φορές το πείραμα της προσομοίωσης, χρησιμοποιώντας κάθε φορά διαφορετικές ακολουθίες τυχαίων αριθμών. Σε κάθε επανάληψη υπολογίζουμε τη μέση τιμή X_i του μεγέθους που θέλουμε να εκτιμήσουμε. Μπορούμε λογικά να υποθέσουμε ότι οι μεταβλητές X_i είναι ανεξάρτητες μεταξύ τους για τις διάφορες επαναλήψεις και συνεπώς η μέση τιμή $\overline{X}$ μπορεί να υπολογιστεί ως μέση τιμή των επιμέρους μέσων τιμών των επαναλήψεων, με αντίστοιχο υπολογισμό του διαστήματος εμπιστοσύνης.

Η μέθοδος αυτή εφαρμόζεται συνήθως όταν ενδιαφερόμαστε για τη μεταβατική συμπεριφορά ενός συστήματος, αρχίζοντας από κάποιες καθορισμένες αρχικές συνθήκες. Για παράδειγμα, θα μπορούσαμε να ζητάμε το μέσο χρόνο μέχρι όλες οι εργασίες να βρεθούν στην ΚΜΕ ενός συστήματος, δεδομένης της τρέχουσας θέσης των εργασιών, ή τον μέσο χρόνο απόκρισης σε ένα διαλογικό σύστημα, όταν δίνεται η τρέχουσα κατάσταση του συστήματος. Και στις δύο περιπτώσεις δεν μας ενδιαφέρει αν το σύστημα βρίσκεται σε κατάσταση ισορροπίας ή όχι. Συνήθως οι επαναλήψεις δεν έχουν μεγάλη διάρκεια και ο αριθμός τους εξαρτάται από την απαιτούμενη ακρίβεια.

Αν η προσομοίωση αφορά την εκτίμηση μεγεθών στη μόνιμη κατάσταση, θα χρειαστεί ενδεχομένως η αφαίρεση του μεταβατικού φαινομένου σε κάθε επανάληψη. Στην περίπτωση αυτή, για να περιοριστεί η απώλεια λόγω των αρχικών μετρήσεων που αποκόπτονται σε κάθε επανάληψη, συμφέρει να εκτελούνται λίγες επαναλήψεις μεγάλης διάρκειας.



Σχήμα 7.3: *Τμηματικές μέσες τιμές.*

7.3.4.2 Τμηματικές Μέσες Τιμές (Batch means)

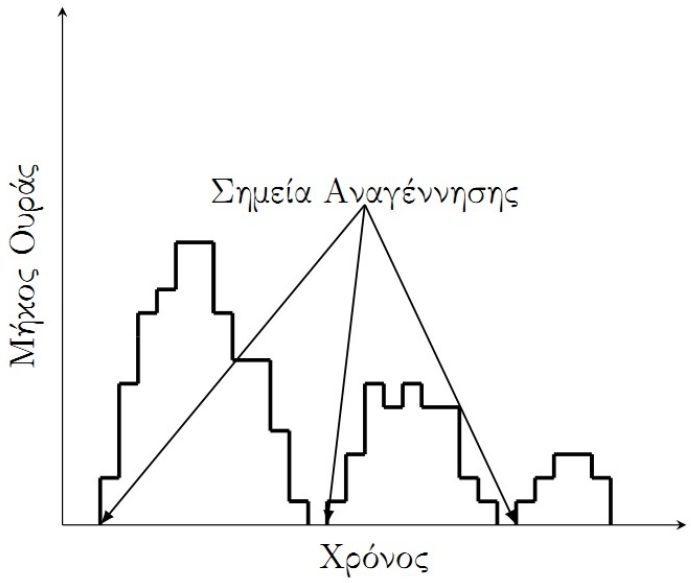
Η μέθοδος αυτή μοιάζει με τη μέθοδο των ανεξάρτητων επαναλήψεων και χρησιμοποιείται κυρίως όταν ενδιαφερόμαστε για τη συμπεριφορά ενός συστήματος στη μόνιμη κατάσταση. Αντί για πολλές ανεξάρτητες προσομοιώσεις, πραγματοποιούμε μία προσομοίωση μεγάλης διάρκειας και τη χωρίζουμε σε διαδοχικά τμήματα (batches) ίσου μήκους. Το μήκος m μπορεί να αναφέρεται είτε σε χρονική διάρκεια, είτε σε αριθμό μετρήσεων (Σχήμα 7.3). Με τον τρόπο αυτό ομαδοποιούμε τις μετρήσεις και υπολογίζουμε τη μέση τιμή X_i για κάθε τέτοια ομάδα μετρήσεων. Με βάση τις τιμές X_i υπολογίζεται η μέση τιμή $\overline{X}$ και το αντίστοιχο διάστημα εμπιστοσύνης.

Κατά την υλοποίηση της μεθόδου πρέπει να αντιμετωπιστούν δύο βασικά προβλήματα. Πρώτον, ο χωρισμός σε τμήματα θα πρέπει να γίνει αφού το σύστημα φθάσει στην κατάσταση ισορροπίας, αλλιώς η μεταβατική κατάσταση μπορεί να αλλοιώσει την ακρίβεια των αποτελεσμάτων. Επομένως, θα πρέπει να εξαλειφθεί η επίδραση της μεταβατικής κατάστασης. Η μέθοδος αυτή συνεπάγεται μικρότερες απώλειες απ' ό,τι η προηγούμενη, καθόσον οι αρχικές μετρήσεις αποκόπτονται μόνο μία φορά.

Το δεύτερο πρόβλημα είναι ότι οι παρατηρήσεις X_i που προκύπτουν από τη μέθοδο δεν είναι γενικά ανεξάρτητες μεταξύ τους. Συνεπώς, δεν μπορεί να εφαρμοστεί η διαδικασία υπολογισμού του διαστήματος εμπιστοσύνης παρά μόνο προσεγγιστικά, είτε λαμβάνοντας υπόψη τη συμμεταβλητότητα (covariance) των παρατηρήσεων στον υπολογισμό της διασποράς, είτε επιλέγοντας το μήκος των τμημάτων, έτσι ώστε οι μέσες τιμές των τμημάτων να έχουν μικρή συσχέτιση. Ένας τρόπος επιλογής του μήκους m βασίζεται στη συμμεταβλητότητα των διαδοχικών παρατηρήσεων:

$$\mathrm{Cov}(X_i, X_{i+1}) = \frac{1}{n-2}\sum_{i=1}^{n-1}(X_i - \overline{X})(X_{i+1} - \overline{X}) \tag{7.16}$$

Συνήθως δοκιμάζονται αυξανόμενες τιμές του μεγέθους m των τμημάτων και επιλέγεται το ελάχιστο απαιτούμενο μέγεθος ώστε ο λόγος της συμμεταβλητότητας των διαδοχικών παρατηρήσεων προς τη διασπορά (αυτοσυσχέτιση τάξης 1) να είναι πολύ μικρότερος της μονάδας (π.χ. μικρότερος του 0,05).



Σχήμα 7.4: *Παράδειγμα αναγεννητικής διαδικασίας.*

7.3.4.3 Η Αναγεννητική Μέθοδος (Regenerative method)

Μία στοχαστική διαδικασία λέγεται *αναγεννητική*, αν υπάρχει κάποια κατάσταση τέτοια, ώστε κάθε φορά που η διαδικασία επανέρχεται στην κατάσταση αυτή το παρελθόν της διαδικασίας δεν έχει καμμία επίδραση στη μελλοντική της εξέλιξη. Μία τέτοια κατάσταση ονομάζεται αναγεννητική και οι χρονικές στιγμές επανόδου σ' αυτήν *σημεία αναγέννησης* (regeneration points) της διαδικασίας. Τα χρονικά διαστήματα μεταξύ διαδοχικών σημείων αναγέννησης ονομάζονται *αναγεννητικοί κύκλοι* (regeneration cycles).

Κατά την προσομοίωση ενός αναγεννητικού συστήματος είναι φανερό ότι τα τμήματα του δειγματικού μονοπατιού που αντιστοιχούν σε διαδοχικούς αναγεννητικούς κύκλους θα είναι ανεξάρτητα και με την ίδια κατανομή. Άρα, αν η χρονική διάρκεια της προσομοίωσης χωριστεί σε τμήματα τα οποία θα συμπίπτουν με αναγεννητικούς κύκλους, οι παρατηρήσεις που αντιστοιχούν σε αυτά τα τμήματα θα είναι ανεξάρτητες και με την ίδια κατανομή. Η μέθοδος αυτή, λοιπόν, λύνει αυτόματα το πρόβλημα της ανεξαρτησίας του δείγματος. Συγχρόνως, αν η προσομοίωση αρχίζει από αναγεννητική κατάσταση, δεν υπάρχει πλέον το πρόβλημα του μεταβατικού φαινομένου. Μία δυσκολία της μεθόδου είναι ο προσδιορισμός της κατάλληλης αναγεννητικής κατάστασης, εφόσον υπάρχει. Για παράδειγμα, σε ένα απλό σύστημα αναμονής η στιγμή άφιξης ενός πελάτη που βρίσκει το σύστημα άδειο είναι σημείο αναγέννησης (Σχήμα 7.4). Τα πράγματα είναι πολύ απλά όταν υπάρχει η ιδιότητα της *έλλειψης μνήμης* (εκθετική κατανομή). Για τις εργοδικές διαδικασίες Markov κάθε κατάσταση είναι αναγεννητική. Επίσης αναγεννητικές καταστάσεις υπάρχουν σε πολλά συστήματα αναμονής που βρίσκονται στη μόνιμη κατάσταση. Δεν είναι πάντα, όμως, προφανής η ύπαρξη αναγεννητικών καταστάσεων σε πολύπλοκα συστήματα. Ακόμα και αν οι καταστάσεις αυτές υπάρχουν, θα πρέπει το μήκος των αναγεννητικών κύκλων να είναι πεπερασμένο και σχετικά μικρό, ώστε να μπορούν να πραγματοποιηθούν αρκετοί κύκλοι στη διάρκεια της προσομοίωσης. Ένα σχετικό πρόβλημα είναι ότι το μήκος των κύκλων είναι απρόβλεπτο και συνεπώς η διάρκεια της προσομοίωσης δεν μπορεί να προγραμματιστεί εκ των προτέρων.

Ο υπολογισμός της διασποράς στην περίπτωση της αναγεννητικής μεθόδου είναι λίγο πιο πολύπλοκος απ' ό,τι στις προηγούμενες μεθόδους. Αυτό συμβαίνει διότι οι κύκλοι έχουν διαφορετικό μήκος και η συνολική μέση τιμή δεν μπορεί να υπολογιστεί ως μέσος όρος των επιμέρους μέσων τιμών.

Έστω ότι ενδιαφερόμαστε για την εκτίμηση ενός δείκτη επίδοσης της μορφής $r = E[V]$, όπου η μεταβλητή $\{V(t),\ t \leq 0\}$ αντιπροσωπεύει ένα χαρακτηριστικό της αναγεννητικής διαδικασίας στη μόνιμη κατάσταση. Υποθέτουμε ότι η προσομοίωση πραγματοποιήθηκε για n αναγεννητικούς κύκλους. Συμβο-

λίζουμε με c_i, $i = 1, 2, \ldots, n$, τη διάρκεια του i-στού αναγεννητικού κύκλου και με y_i, $i = 1, 2, \ldots, n$, το ολοκλήρωμα της $V(t)$ πάνω στον i-στό αναγεννητικό κύκλο. (Π.χ., η $V(t)$ μπορεί να παριστάνει τον αριθμό πελατών σε μία ουρά.) Τα ζεύγη (y_i, c_i), $i = 1, 2, \ldots, n$, είναι ανεξάρτητα και ακολουθούν την ίδια κατανομή (οι μεταβλητές y_i και c_i, όμως, είναι προφανώς εξαρτημένες μεταξύ τους). Οι επιμέρους μέσες τιμές θα δίνονται από τη σχέση $X_i = y_i/c_i$.

Στην περίπτωση που οι παρατηρήσεις του μεγέθους V γίνονται σε διακριτές χρονικές στιγμές, η μεταβλητή y_i μπορεί να εκφράζει άθροισμα τιμών (αντί για ολοκλήρωμα) και η μεταβλητή c_i το πλήθος των παρατηρήσεων στον αναγεννητικό κύκλο (αντί για χρονική διάρκεια του κύκλου).

Ο ζητούμενος δείκτης r μπορεί να εκτιμηθεί με βάση την ποσότητα $R = \dfrac{\overline{y}}{\overline{c}}$, όπου $\overline{y}$ και $\overline{c}$ αντίστοιχα οι δειγματικές μέσες τιμές των $\{y_i\}$, $\{c_i\}$. Η εκτίμηση αυτή δεν είναι αμερόληπτη ($E[R] \neq r$), η πόλωση όμως τείνει στο 0 όταν το $n \to \infty$, επομένως ο αριθμός των αναγεννητικών κύκλων θα πρέπει να είναι αρκετά μεγάλος.

Για τον υπολογισμό του διαστήματος εμπιστοσύνης με βάση την εκτίμηση R, αποδεικνύεται ότι η τυχαία μεταβλητή $Z = (R - r)\overline{c}/(s/\sqrt{n})$ ακολουθεί κατανομή Student με $n - 1$ βαθμούς ελευθερίας (ή μοναδιαία κανονική κατανομή για μεγάλο n), όπου η δειγματική διασπορά s^2 υπολογίζεται από τις σχέσεις:

$$s^2 = s_y^2 - 2Rs_{yc} + R^2 s_c^2 \tag{7.17}$$

$$s_y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2 \tag{7.18}$$

$$s_c^2 = \frac{1}{n-1}\sum_{i=1}^{n}(c_i - \overline{c})^2 \tag{7.19}$$

$$s_{yc} = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})(c_i - \overline{c}) \tag{7.20}$$

Επομένως, το διάστημα εμπιστοσύνης θα δίνεται από τη σχέση:

$$\Pr\left[R - \frac{z_{1-\alpha/2}s}{\overline{c}\sqrt{n}} \leq r \leq R + \frac{z_{1-\alpha/2}s}{\overline{c}\sqrt{n}}\right] = 1 - \alpha \tag{7.21}$$

για βαθμό εμπιστοσύνης $1 - \alpha$, όπου $z_{1-\alpha/2}$ είναι το $(1 - \alpha/2)$-ποσοστιαίο σημείο της κατανομής Student με $n - 1$ βαθμούς ελευθερίας ή της μοναδιαίας κανονικής κατανομής για μεγάλο n.

Τελειώνοντας σχετικά με τη στατιστική ανάλυση, θα πρέπει να παρατηρήσουμε ότι, για τις διάφορες μεθόδους που παρουσιάστηκαν, ο υπολογισμός των απαραίτητων ποσοτήτων μπορεί να ενσωματωθεί σχετικά εύκολα στο βασικό πρόγραμμα της προσομοίωσης. Συνήθως απαιτείται να τηρούνται διάφορα αθροίσματα, ο υπολογισμός των οποίων γίνεται σταδιακά κατά τη διάρκεια της προσομοίωσης, όπως και για τις εκτιμήσεις των διαφόρων δεικτών επίδοσης.

7.4 Ένα Παράδειγμα Προσομοίωσης

Θα παρουσιάσουμε τώρα τα βασικά στοιχεία ενός προγράμματος προσομοίωσης. Το πρόγραμμα είναι γραμμένο σε γλώσσα C και περιγράφει τη λειτουργία ενός κλειστού δικτύου αναμονής.

Παράδειγμα 7.3. *Θεωρούμε κλειστό δίκτυο αναμονής μορφής γινομένου (BCMP) με μία κατηγορία πελατών, το οποίο θα προσομοιώσουμε εφαρμόζοντας την αναγεννητική μέθοδο [7]. Το πρόγραμμα της προσομοίωσης δεν είναι πλήρες, με την έννοια ότι αφήνονται κενές ορισμένες επιλογές του χρήστη που χαρακτηρίζουν τεχνικές λεπτομέρειες της υλοποίησης. Επίσης, παραλείπονται τα σώματα των γενικών συναρτήσεων που αφορούν πράξεις σε διατεταγμένη γραμμική λίστα (διαχείριση απλών δομών δεδομένων). Υποθέτουμε ότι η συνάρτηση random() παρέχει τυχαίους αριθμούς ομοιόμορφα κατανεμημένους στο διάστημα (0,1). Επίσης, διαθέτουμε τη συνάρτηση sqr(x) που υπολογίζει το τετράγωνο ενός αριθμού.*

```
#include  <stdio.h>
.......            /* συμπερίληψη αρχείων */
.......            /* ορισμοί-μακροεντολές */
.......            /* δηλώσεις */

#define  NQ  .....      /* αριθμός σταθμών */
#define  NJ  .....      /* αριθμός εργασιών (πελατών) στο δίκτυο */
#define  EVENTLIMIT  .....         /* μέγιστος αριθμός γεγονότων
(συνθήκη τερματισμού) */

typedef struct JobElement *Jobptr;
typedef struct EventElement *Eventptr;

struct EventElement {        /* στοιχείο της λίστας γεγονότων */
float time;                /* χρονική στιγμή */
Jobptr job;                /* εργασία που σχετίζεται με το γεγονός */
Eventptr next;             /* επόμενο/προηγούμενο γεγονός */
Eventptr previous;         /* (διπλά συνδεδεμένη λίστα) */
};
struct JobElement {          /* εργασία */
int currentQueue;          /* τρέχων σταθμός (αρίθμηση 0..NQ-1) */
float request;             /* (υπολειπόμενος) χρόνος εξυπηρέτησης */
Jobptr nextJob;            /* επόμενη εργασία στη λίστα του σταθμού */
Eventptr event;            /* δρομολογημένο γεγονός
που σχετίζεται με την εργασία */
};

int i;
Eventptr firstEvent, lastEvent;   /* λίστα γεγονότων */
float clock;                      /* ρολόι */
int endCycle;                     /* έλεγχος τέλους αναγεννητικού κύκλου */

struct Queue {          /* σταθμός */
int discipline;       /* κανονισμός εξυπηρέτησης
1:FCFS,2:PS,3:IS,4:LCFSPR */
int numberServers;    /* αριθμός εξυπηρετητών:
>=1 για FCFS, 1 για PS και LCFSPR,
NJ (μέγιστη τιμή) για IS */
float meanService;    /* μέσος χρόνος εξυπηρέτησης,
υποθέτουμε εκθετική κατανομή */
float routing[NQ];    /* πιθανότητες δρομολόγησης */
Jobptr firstInQueue;  /* λίστα εργασιών στον σταθμό */
Jobptr lastInQueue;
int length;           /* αριθμός εργασιών στον σταθμό
float oldClock;       /* χρονική στιγμή τελευταίου γεγονότος στον σταθμό */

/* αθροίσματα μετρήσεων σε κάθε κύκλο */
float sumTimeLength;      /* άθροισμα εμβαδών ορθογωνίων:
χρόνος x αριθμός εργασιών */
```

```
float sumBusyTime;        /* άθροισμα διαστημάτων απασχόλησης */
int numberCompletions;    /* αριθμός αναχωρήσεων */

/* αθροίσματα μετρήσεων για ολους τους κύκλους */
float bt;      /* busyTime */
float tl;      /* timeLength */
float nc;      /* numberCompletions */
float btsq;    /* busyTime squared */
float btxcl;   /* busyTime x cycleLength */
float ncsq;    /* numberCompletions squared */
float ncxcl;   /* numberCompletions x cycleLength */
float tlsq;    /* timeLength squared */
float tlxcl;   /* timeLength x cycleLength */
float tlxnc;   /* timeLength x numberCompletions */

float util, tput, ql, qt;         /* μέσες τιμές δεικτών */
float dutil, dtput, dql, dqt;     /* διαστήματα εμπιστοσύνης */

float varbt, varnc, vartl;
float covarbtcl, covarnccl, covartlcl, covartlnc;
};

struct Queue queues[NQ]

int numberEvents, numberCycles, nocycm1;

float timeCycleStarted, cycleLength;   /* διάρκεια αναγεννητικού κύκλου */

float sumcl, sumclsq, varcl, dcl;

Jobptr tempJob;

/*-----------------------------------------------------------------*/

void insertEvent(float, Jobptr);
/* εισαγωγή γεγονότος στη λίστα γεγονότων
ορίσματα: χρόνος, εργασία */

void removeEvent(Eventptr, float *, Jobptr *);
/* εξαγωγή γεγονότος από τη λίστα γεγονότων
επιστρέφει τον χρόνο και την εργασία του γεγονότος. */

/*-----------------------------------------------------------------*/

void complete(Jobptr j) {
/* τέλος εξυπηρέτησης της εργασίας j. */

int leng;
Jobptr l;
float t;
```

```
struct Queue *q;

q=&queues[j->currentQueue];

/* στατιστικά στοιχεία */
q->numberCompletions=q->numberCompletions+1;
q->sumTimeLength=q->sumTimeLength
+(clock-q->oldClock)*q->length;
q->sumBusyTime=q->sumBusyTime+(clock-q->oldClock)*
min(q->length,q->numberServers);
q->oldClock=clock;

/* ενημέρωση μεταβλητών */
q->length=q->length-1;

if ((q->discipline==1) /* FCFS */ || (q->length==0)) {
q->firstInQueue=q->firstInQueue->nextJob;
if (q->firstInQueue==NULL) q->lastInQueue=NULL;
if (q->length>=q->numberServers) {
leng=1;
l=q->firstInQueue;
while (leng<q->numberServers) {
l=l->nextJob;
leng=leng+1;
}
l->request=-q->meanService*log(random());
insertEvent(clock+l->request,l);
}
}
else if (discipline==2) /* PS */ {
t=j->request;
q->firstInQueue=q->firstInQueue->nextJob;
l=q->firstInQueue;
while (l!=NULL) {
l->request=l->request-t;
l=l->nextJob;
}
insertEvent(clock+q->firstInQueue->request*q->length,
q->firstInQueue);
}
else if (discipline==3) /* IS */ {
t=j->request;
q->firstInQueue=q->firstInQueue->nextJob;
l=q->firstInQueue;
while (l!=NULL) {
l->request=l->request-t;
l=l->nextJob;
}
}
else /* discipline=LCFSPR */ {
```

```
q->firstInQueue=q->firstInQueue->nextJob;
insertEvent(clock+q->firstInQueue->request,
q->firstInQueue);
}
}   /* complete */

/*-------------------------------------------------------------------*/

procedure updateQueue(int i, Jobptr j) {
/* εισάγει την εργασία j στην ουρά i,
στην κατάλληλη θέση σύμφωνα με το j->request.
ουρές τύπου PS και IS θεωρούνται διατεταγμένες.
υποτίθεται ότι η ουρά δεν είναι άδεια. */

Jobptr temp;
struct Queue *q;

q=&queues[i];

if (j->request<q->firstInQueue->request) {
j->nextJob=q->firstInQueue;
firstInQueue=j;
}
else if (j->request>=q->lastInQueue->request) {
q->lastInQueue->nextJob=j;
j->nextJob=NULL;
q->lastInQueue=j;
}
else  {
temp=q->firstInQueue;
while (j->request>=temp->nextjob->request) {
temp=temp->nextjob;
j->nextjob=temp->nextJob;
temp->nextJob=j;
}
}     /* updateQueue*/

/*-------------------------------------------------------------------*/

void arrive(Jobptr j, int c) {
/* άφιξη της εργασίας j στην ουρά c. */

float t; Jobptr dummyJob, temp;

struct Queue *q;

j->currentqueue=c;
q=&queues[c];

/* στατιστικά στοιχεία */
```

```
q->sumTimeLength=q->sumTimeLength
+(clock-q->oldClock)*q->length;
q->sumBusyTime=q->sumBusyTime+(clock-q->oldClock)*
min(q->length,q->numberServers);
q->oldClock=clock;

/* ενημέρωση μεταβλητών */
if ((discipline==1) /* FCFS */ || (firstInQueue==NULL)) {
j->nextJob=NULL;
if (q->firstInQueue==NULL)
q->firstInQueue=j;
else
q->lastInQueue->nextJob=j;
q->lastInQueue=j;
q->length=q->length+1;
if (q->length<=q->numberServers) {
j->request=-q->meanService*log(random());
insertEvent(clock+j->request,j)
}
}
else if (discipline==2) /* PS */ {
removeEvent(q->firstInQueue->event,t,dummyJob);
t=q->firstInQueue->request-(t-clock)/q->length;
temp=firstInQueue;
while (temp!=NULL) {
temp->request=temp->request-t;
temp=temp->nextJob;
}
j->request=-q->meanService*log(random());
updateQueue(c,j);
q->length=q->length+1;
insertEvent(clock+q->firstInQueue->request*q->length,
q->firstInQueue);
}
else if (discipline==3) /* IS */ {
j->request=-q->meanService*log(random());
updateQueue(c,j);
q->length=q->length+1;
insertEvent(clock+j->request,j);
}
else /* discipline=LCFSPR */ {
removeEvent(q->firstInQueue->event,t,dummyjob);
q->firstInQueue->request=t-clock;  /* διακοπή εξυπηρέτησης */
j->nextjob=q->firstInQueue;
q->firstInQueue=j;
q->length=q->length+1;
j->request=-q->meanService*log(random());
insertEvent(clock+j->request,j);
}
}   /* arrive */
```

```
/*-----------------------------------------------------------------*/

int nextNode(Jobptr j) {
/* εύρεση του επόμενου σταθμού που επισκέπτεται η εργασία j. */

float prob;
int i;
struct Queue *q;

q=&queues[j->currentQueue];
prob=random();
i=1;
while ((prob>q->routing[i]) && (i!=NQ)) {
prob=prob-q->routing[i];
i=i+1;
}
return i;
}    /* nextnode */

/*-----------------------------------------------------------------*/

void checkCycle();
/* ΄Ελεγχος για τέλος αναγεννητικού κύκλου. Ενημέρωση συσσωρευτών. */

int i;
struct Queue *q;

endCycle=0;
if ((.....)    /* συνθήκη αναγεννητικής κατάστασης */
&& (numberEvents>0)) {
endCycle=1;
numberCycles=numberCycles+1;
cycleLength=clock-timeCycleStarted;
timeCycleStarted=clock;
sumcl=sumcl+cycleLength;
sumclsq=sumclsq+sqr(cycleLength);
for (i=0; i<NQ; i++) {
q=&queues[i];
q->sumTimeLength=q->sumTimeLength
+(clock-q->oldClock)*q->length;
q->sumBusyTime=(q->sumBusyTime+(clock-q->oldClock)*
min(q->length,q->numberServers))/q->numberServers;
q->oldClock=clock;

q->bt=q->bt+q->sumBusyTime;
q->tl=q->tl+q->sumTimeLength;
q->nc=q->nc+q->numberCompletions;
q->btsq=q->btsq+sqr(q->sumBusyTime);
q->btxcl=q->btxcl+q->sumBusyTime*cycleLength;
q->sumBusyTime=0.0;
```

```
q->ncsq=q->ncsq+sqr(q->numberCompletions);
q->ncxcl=q->ncxcl+q->numberCompletions*cycleLength;
q->tlsq=q->tlsq+sqr(q->sumTimeLength);
q->tlxcl=q->tlxcl+q->sumTimeLength*cycleLength;
q->tlxnc=q->tlxnc+q->sumTimeLength*q->numberCompletions;
q->numberCompletions=0;
q->sumTimeLength=0.0;
}
}
}     /* checkCycle */

/*-----------------------------------------------------------------*/

main()
/* κύριο πρόγραμμα */
.....
/* αρχικοποίηση γεννήτριας τυχαίων αριθμών */
.....

/* αρχικοποιήσεις */
numberEvents=0;
firstEvent=NULL;
lastEvent=NULL;
clock=0.0;
numberCycles=0;
endCycle=0;
timeCycleStarted=0.0;
sumcl=0.0;
sumclsq=0.0;

struct Queue *q;

/* παράμετροι σταθμών */

q=&queues[0];
q->discipline=......
q->numberServers=......
q->meanService=......
for (i=0;i<NQ;i++) scanf("%f", &routing[i]);
.............................
.............................
q=&queues[NQ-1];
q->discipline=......
q->numberServers=......
q->meanservice=......
for (i=0;i<NQ;i++) scanf("%f", &routing[i]);

for (i=0;i<NQ;i++) {
q=&queues[i];
```

```
q->firstInQueue=NULL;
q->lastInQueue=NULL;
q->length=0;
q->oldClock=0.0;
q->sumTimeLength=0.0;
q->sumBusyTime=0.0;
q->numberCompletions=0;
q->bt=0.0;
q->tl=0.0;
q->nc=0.0;
q->btsq=0.0;
q->btxcl=0.0;
q->ncsq=0.0;
q->ncxcl=0.0;
q->tlsq=0.0;
q->tlxcl=0.0;
q->tlxnc=0.0
}

/* αρχικοποίηση σε αναγεννητική κατάσταση:
δημιουργία και δρομολόγηση NJ εργασιών */

tempJob=(Jobptr)malloc(sizeof JobElement); arrive(tempJob,...);
.............................
.............................
tempJob=(Jobptr)malloc(sizeof JobElement); arrive(tempJob,...);


/* προσομοίωση */

while ((firstEvent!=NULL) &&
((numberEvents<EVENTLIMIT) || (!endCycle))) {
/* η προσομοίωση σταματά σε τέλος αναγεννητικού κύκλου,
όταν έχουν συμπληρωθεί τουλάχιστον EVENTLIMIT γεγονότα. */

removeEvent(firstEvent,clock,tempJob);
numberEvents=numberEvents+1;
complete(tempJob);
i=nextNode(tempJob);
arrive(tempJob,i);
checkCycle;
}


/* στατιστικά αποτελέσματα */
printf("\n");
printf("number of events: %f\n simulated time: %f\n",
numberEvents, clock);
printf("\n");
printf("queue utilization throughput queue-length queueing-time\n");
```

```
/* υποθέτουμε ότι numberCycles>1 */

cycleLength=sumcl/numberCycles;
nocycm1=numberCycles-1;
varcl=(sumclsq-sqr(sumcl)/numberCycles)/nocycm1;

/* βαθμός εμπιστοσύνης 1-α=0.9 */

for (i=0;i<NQ;i++) {
q=&queues[i];

if (q->nc>0.0) {
q->util=q->bt/sumcl;
q->varbt=(q->btsq-sqr(q->bt)/numberCycles)
/nocycm1;
q->covarbtcl=(q->btxcl-q->bt*sumcl/numberCycles)
/nocycm1;
q->dutil=1.645*sqrt((q->varbt-2*q->util*q->covarbtcl
+sqr(q->util)*varcl)/numberCycles)
/cycleLength;

q->tput=q->nc/sumcl;
q->varnc=(q->ncsq-sqr(q->nc)/numberCycles)
/nocycm1;
q->covarnccl=(q->ncxcl-q->nc*sumcl/numberCycles)
/nocycm1;
q->dtput=1.645*sqrt((q->varnc-2*q->tput*q->covarnccl
+sqr(q->tput)*varcl)/numberCycles)
/cycleLength;

q->ql=q->tl/sumcl;
q->vartl=(q->tlsq-sqr(q->tl)/numberCycles)
/nocycm1;
q->covartlcl=(q->tlxcl-q->tl*sumcl/numberCycles)
/nocycm1;
q->dql=1.645*sqrt((q->vartl-2*q->ql*q->covartlcl
+sqr(q->ql)*varcl)/numberCycles)
/cycleLength;

q->qt=q->tl/q->nc;
q->covartlnc=(q->tlxnc-q->tl*q->nc/numberCycles)
/nocycm1;
q->dqt=1.645*sqrt((q->vartl-2*q->qt*q->covartlnc
+sqr(q->qt)*q->varnc)/numberCycles)
/(nc/numberCycles);

printf("%d\n",i);
printf("lower %f %f %f %f \n", q->util-q->dutil,q->tput-q->dtput,
q->ql-q->dql,q->qt-q->dqt);
printf("mean %f %f %f %f \n", q->util,q->tput,
```

```
        q->ql,q->qt);
        printf("upper %f %f %f %f \n",
        q->util+q->dutil,q->tput+q->dtput,
        q->ql+q->dql,q->qt+q->dqt);
        printf("\n");
        }

        printf("n");
        printf("number of cycles: %d\n",numberCycles);
        printf("average number of events: %f\n",numberEvents/numberCycles);
        dcl=1.645*sqrt(varcl/numberCycles);
        printf("average cycle length: %f, c.i.: ( %f , %f)",
        cycleLength, cycleLength-dcl,cycleLength+dcl);
        return 0;
        }
```

□

Βιβλιογραφία

[1] Bratley, P., Fox, B.L. and Schrage, L.E., *A Guide to Simulation*, Springer-Verlag, 1986.

[2] Jain, R., *The Art of Computer Systems Performance Analysis - Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, 1991.

[3] MacNair, E.A. and Sauer, C.H., *Elements of Practical Performance Modeling*, Prentice-Hall, 1985.

[4] Mitrani, I., *Simulation Techniques for Discrete-Event Systems*, Cambridge University Press, 1982.

[5] Obaidat, M.S., and Boudriga, N.A., *Performance Evaluation of Computer and Telecommunication Systems*, Wiley, 2010.

[6] Ross, S.M., *Simulation* (Fourth edition), Academic Press, 2006.

[7] Sauer, C.H. and Chandy, K.M., *Computer Systems Performance Modelling*, Prentice-Hall, 1981.

Κεφάλαιο 8

Τεχνικές Μετρήσεων

Σύνοψη
Στο κεφάλαιο αυτό, εξετάζονται οι βασικές έννοιες και αρχές για τη μέτρηση και την πρόβλεψη της επίδοσης. Αναλύονται οι στόχοι και ο σχεδιασμός μιας μελέτης επίδοσης. Ορίζεται το φορτίο (workload) ενός συστήματος και περιγράφονται τεχνικές επιλογής και χαρακτηρισμού του φορτίου για την ανάλυση επίδοσης του συστήματος. Εξετάζονται τεχνικές μετρήσεων και συλλογής δεδομένων, όπως αποτύπωση γεγονότων (event tracing), δειγματοληψία, επόπτες (monitors) και τεχνικές εποπτείας, βελτιστοποιητές προγράμματος, λογιστικά ημερολόγια (accounting logs) και προγράμματα αναφοράς (benchmarks). Περιγράφονται διάφορες κατηγορίες προγραμμάτων αναφοράς, με έμφαση στα τυποποιημένα προγράμματα που αναπτύσσονται και ελέγχονται από διεθνείς οργανισμούς (SPEC, TPC). Εξετάζονται θέματα διαχείρισης και σχεδιασμού παραγωγικής ικανότητας (capacity management/planning) και τεχνικές πρόβλεψης φορτίου με έμφαση στον σχεδιασμό υπηρεσιών Ιστού (Web services). Περιγράφονται τεχνικές παρουσίασης και ερμηνείας των αποτελεσμάτων των μετρήσεων.

8.1 Μετρήσεις και Φορτία

Η τεχνική των μετρήσεων μπορεί να εφαρμοστεί σε ένα πραγματικό σύστημα ή στην πρωτότυπη έκδοση ενός συστήματος που είναι υπό ανάπτυξη. Χρησιμοποιείται συχνά αντί των τεχνικών μοντελοποίησης και θεωρείται η πλέον αξιόπιστη τεχνική αποτίμησης της επίδοσης. Είναι, όμως, και η πλέον δαπανηρή. Η μελέτη ενός συστήματος με τη βοήθεια μετρήσεων περιλαμβάνει την παρακολούθηση της λειτουργίας του κατά την εξυπηρέτηση συγκεκριμένων φορτίων. Η αποτελεσματικότητα της μελέτης εξαρτάται από την επιλογή των κατάλληλων φορτίων, την επιτυχή εκτέλεση των μετρήσεων και τη σωστή ανάλυση και ερμηνεία των αποτελεσμάτων [7, 6, 5, 9, 4, 11].

Το *φορτίο* (workload) εκφράζει τις αιτήσεις των χρηστών για εξυπηρέτηση από το σύστημα. Η έννοια του φορτίου συνδέεται συνήθως με τη σύγκριση της επίδοσης υπολογιστικών συστημάτων. Το φορτίο που χρησιμοποιείται σε μελέτες επίδοσης αναφέρεται και ως *δοκιμαστικό φορτίο* (test workload). Το υπολογιστικό σύστημα, του οποίου εξετάζεται η επίδοση, αναφέρεται ως *σύστημα υπό δοκιμή* (system under test – SUT). Πολλές φορές, η μελέτη επίδοσης αφορά την εξέταση εναλλακτικών επιλογών για ένα συστατικό του συστήματος, που είναι το *συστατικό υπό μελέτη* (component under study – CUS). Όπως θα δούμε στη συνέχεια, είναι σημαντική η διάκριση ανάμεσα στο SUT και στο CUS.

8.1.1 Τύποι φορτίου

Διακρίνουμε *πραγματικά φορτία*, δηλαδή φορτία που σχετίζονται με τη λειτουργία ενός συστήματος υπό πραγματικές συνθήκες, και *συνθετικά φορτία*, που αναπαριστούν τα χαρακτηριστικά των πραγματικών φορτίων. Μια άλλη διάκριση χαρακτηρίζει τα φορτία ως *εκτελέσιμα* και *μη-εκτελέσιμα*.

Τα πραγματικά φορτία είναι κυρίως εκτελέσιμα προγράμματα ή τμήματα προγραμμάτων από πραγματικές εφαρμογές. Μπορούμε, ακόμη, να έχουμε (μη-εκτελέσιμα) μοντέλα πραγματικού φορτίου, τα οποία βασίζονται στη λήψη *αποτυπώματος* (workload trace), δηλαδή στη χρονολογική καταγραφή γεγονότων και

δεδομένων από πραγματική εκτέλεση προγράμματος εφαρμογής. Για παράδειγμα, στο *ημερολόγιο* (log) ενός εξυπηρετητή Ιστού, κάθε εγγραφή περιέχει τα στοιχεία μιας αίτησης http (όνομα του υπολογιστή που κάνει την αίτηση, χρονοσφραγίδα, όνομα του ζητούμενου αρχείου). Τα αποτυπώματα αξιοποιούνται συνήθως κατάλληλα ως είσοδος σε μοντέλα προσομοίωσης και στον χαρακτηρισμό φορτίου.

Ένα συνθετικό φορτίο αποτελεί ουσιαστικά ένα μοντέλο του φορτίου και έχει το πλεονέκτημα ότι μπορεί να εφαρμοστεί πολλές φορές σε διαφορετικά συστήματα με ελεγχόμενο τρόπο. Μπορεί να είναι ένα (εκτελέσιμο) πρόγραμμα του οποίου η εκτέλεση επιτρέπει τη μελέτη της συμπεριφοράς του συστήματος, όπως είναι τα προγράμματα αναφοράς που θα εξεταστούν στη συνέχεια. Μπορεί, επίσης, να είναι ένα μη-εκτελέσιμο μοντέλο (*μοντέλο φορτίου*), κατάλληλο να χρησιμοποιηθεί ως είσοδος σε ένα *μοντέλο συστήματος* (αναλυτικό ή μοντέλο προσομοίωσης). Στην περίπτωση αυτή, το μοντέλο φορτίου περιγράφεται από ένα σύνολο παραμέτρων, που χαρακτηρίζουν την εκτέλεση του φορτίου στο υπό μελέτη σύστημα. Οι μελέτες επίδοσης στηρίζονται συνήθως σε συνθετικά φορτία.

8.1.2 Επιλογή Τύπου Φορτίου

Η επιλογή του κατάλληλου φορτίου είναι βασικό ζήτημα σε κάθε μελέτη αξιολόγησης. Το σημείο εκκίνησης είναι η αποτύπωση των υπηρεσιών που παρέχει το σύστημα. Καταρχάς θα πρέπει να προσδιοριστούν ακριβώς τα όρια του υπό δοκιμή συστήματος. Συχνά εξετάζονται εναλλακτικές λύσεις σε σχέση με κάποιο συγκεκριμένο συστατικό του συστήματος. Π.χ. μελετάται η επίδραση του κεντρικού επεξεργαστή (συστατικό) στην επίδοση ενός υπολογιστή (σύστημα). Η επιλογή του φορτίου και των δεικτών επίδοσης σχετίζεται με τις υπηρεσίες που παρέχονται στο επίπεδο του συστήματος (SUT) και όχι του συστατικού (CUS). Π.χ. οι υπηρεσίες που παρέχονται από έναν επεξεργαστή είναι εντολές, οπότε το αντίστοιχο φορτίο μπορεί να εκφραστεί ως συχνότητα εντολών και ένας κατάλληλος δείκτης επίδοσης θα είναι τα MIPS. Αντίθετα, οι υπηρεσίες που παρέχονται από ένα σύστημα καταμερισμού χρόνου μπορεί να είναι συναλλαγές, οπότε το φορτίο εκφράζεται ως συχνότητα συναλλαγών και ο δείκτης επίδοσης είναι transactions per second. Συνεπώς, αν συγκρίνονται δύο συστήματα που διαφέρουν μόνο ως προς τον κεντρικό επεξεργαστή, το κατάλληλο φορτίο θα είναι η συχνότητα συναλλαγών (επίπεδο συστήματος) και όχι η συχνότητα εντολών (επίπεδο συστατικού). Αν το σύστημα παρέχει πολλών ειδών υπηρεσίες, αυτές θα πρέπει να αποτυπώνονται κατάλληλα στο φορτίο.

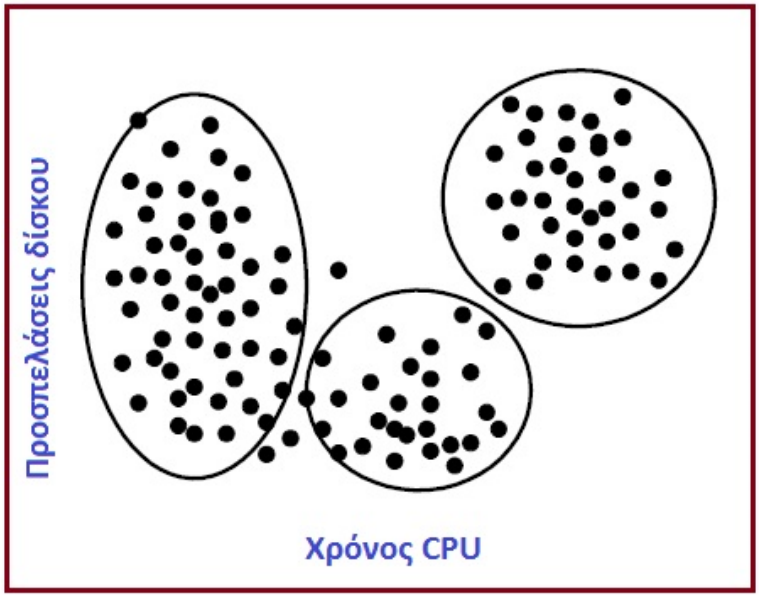
Γενικά, η επιλογή του φορτίου διευκολύνεται από μία ιεραρχική αντίληψη της δομής του συστήματος, η οποία αντικατοπτρίζει την παροχή υπηρεσιών ανά επίπεδο.

Ένα άλλο ζήτημα κατά την επιλογή του φορτίου είναι ο βαθμός λεπτομέρειας στην καταγραφή (και, συνεπώς, στην αναπαραγωγή) των αιτήσεων για εξυπηρέτηση από το σύστημα. Π.χ., η περιγραφή του φορτίου μπορεί να βασίζεται στη συχνότητα των διαφόρων τύπων αιτήσεων, στη μέση απαίτηση εξυπηρέτησης από τους πόρους του συστήματος, στην κατανομή πιθανότητας της απαίτησης ή στη λεπτομερή αποτύπωση μιας χρονικής ακολουθίας αιτήσεων (trace). Η επιλογή της κατάλληλης διατύπωσης σχετίζεται άμεσα με την ακολουθούμενη μέθοδο αξιολόγησης (μετρήσεις, προσομοίωση, αναλυτικό μοντέλο).

8.1.3 Χαρακτηρισμός Φορτίου

Για να ελέγχονται εναλλακτικές λύσεις κάτω από τις ίδιες συνθήκες θα πρέπει να είναι δυνατή η αναπαραγωγή του φορτίου. Για τον λόγο αυτό, στις μελέτες αξιολόγησης χρησιμοποιούνται κυρίως συνθετικά φορτία τα οποία αποτελούν μοντέλα που παριστάνουν τα βασικά χαρακτηριστικά των πραγματικών φορτίων. Η διαδικασία κατασκευής του μοντέλου ονομάζεται *χαρακτηρισμός φορτίου* (workload characterization) [6]. Μετά την κατασκευή του μοντέλου, αλλαγές στο σύστημα ή στο φορτίο μπορούν να μελετηθούν με ελεγχόμενο τρόπο μέσω των παραμέτρων του μοντέλου. Οι παράμετροι του φορτίου που επιλέγονται πρέπει να εξαρτώνται κατά το δυνατό από το ίδιο το φορτίο και όχι από το σύστημα, ώστε να παρέχουν μεταφερτότητα.

Στην περίπτωση μη-εκτελέσιμου συνθετικού φορτίου, το οποίο αποτελεί είσοδο σε αναλυτικό μοντέλο, οι παράμετροι του φορτίου θα περιγράφουν τα χαρακτηριστικά των πελατών (αιτήσεων) και των σταθμών εξυπηρέτησης (συστατικών του συστήματος, π.χ. ΚΜΕ, δίσκος): αριθμός κατηγοριών, χρόνοι μεταξύ αφίξεων, απαιτήσεις εξυπηρέτησης στην ΚΜΕ, αριθμός προσπελάσεων στον δίσκο, αριθμός bytes που γράφον-



Σχήμα 8.1: *Ομαδοποίηση μετρήσεων.*

ται/διαβάζονται κλπ. Το σύνολο των παραμέτρων αυτών ορίζει το μοντέλο του φορτίου, το οποίο —εκτός από είσοδος στο αναλυτικό μοντέλο— μπορεί να είναι η βάση ανάπτυξης ενός συνθετικού προγράμματος. Σε κάθε περίπτωση, οι παράμετροι προκύπτουν από τη διαδικασία του χαρακτηρισμού φορτίου.

Ο χαρακτηρισμός φορτίου βασίζεται κυρίως σε στατιστικές τεχνικές. Οι τεχνικές αυτές περιλαμβάνουν μέτρα θέσης και μεταβλητότητας (μέση τιμή, διασπορά, συντελεστής μεταβλητότητας, ποσοστιαία σημεία), κατανομές συχνοτήτων, ιστογράμματα, μαρκοβιανά μοντέλα, και μεθόδους ανάλυσης δεδομένων, όπως *ομαδοποίηση* (clustering) ή ανάλυση κύριων συνιστωσών (principal-component analysis — PCA).

Έστω ότι διαθέτουμε ένα σύνολο δεδομένων που παριστάνει το φορτίο ενός εξυπηρετητή Ιστού (αιτήσεις http). Υποθέτουμε ότι οι απαιτήσεις εξυπηρέτησης στην ΚΜΕ και στον δίσκο χαρακτηρίζονται από υψηλή μεταβλητότητα, με αποτέλεσμα η χρήση των μέσων τιμών της απαίτησης εξυπηρέτησης ως παραμέτρων να μην είναι επαρκής για την αντιπροσωπευτικότητα ενός μοντέλου φορτίου. Το πρόβλημα αντιμετωπίζεται με την ομαδοποίηση των δεδομένων, ώστε η μεταβλητότητα στο εσωτερικό καθεμιάς από τις ομάδες που θα προκύψουν να είναι πολύ μικρότερη από τη μεταβλητότητα στο αρχικό σύνολο (Σχ. 8.1). Οι ομάδες αυτές αντιστοιχούν στις διαφορετικές κατηγορίες ενός μοντέλου αναμονής πολλών κατηγοριών. Θεωρούμε ότι τα αρχικά δεδομένα του φορτίου αντιστοιχούν σε ένα σύνολο από N σημεία $x_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}]$ σε n-διάστατο χώρο, όπου n το πλήθος των στοιχείων (ιδιοτήτων) που περιγράφουν μια αίτηση http: χρονικές στιγμές αφίξεων, απαιτήσεις εξυπηρέτησης στην ΚΜΕ και στον δίσκο, κλπ. Υπάρχουν διάφοροι αλγόριθμοι ομαδοποίησης που μπορούν να εφαρμοστούν στο σύνολο των σημείων x_i, όπως ο *αλγόριθμος των k-μέσων* (k-means) ή ο *αλγόριθμος ιεραρχικής ομαδοποίησης*, που χρησιμοποιούν Ευκλείδεια απόσταση μεταξύ των σημείων.

Ο αλγόριθμος των k-μέσων παράγει k ομάδες σημείων x_i, όπου k σταθερός (προκαθορισμένος) αριθμός. Κάθε ομάδα Ω_m $(1 \leq m \leq k)$ αντιπροσωπεύεται από το κέντρο βάρους της c_m και παριστάνει μια κατηγορία φορτίου με ιδιαίτερα χαρακτηριστικά. Η μέθοδος αυτή χαρακτηρίζεται ως άκαμπτη ομαδοποίηση (hard clustering), καθόσον κάθε σημείο αντιστοιχίζεται αποκλειστικά σε μια ομάδα. Σε αντιδιαστολή βρίσκεται ο *ασαφής αλγόριθμος των k-μέσων* (fuzzy k-means), ο οποίος πραγματοποιεί εύκαμπτη ομαδοποίηση (soft clustering), σύμφωνα με την οποία κάθε πρότυπο μπορεί να ανήκει σε περισσότερες από μία ομάδες με διαφορετικό βαθμό συμμετοχής. Στη συνέχεια, παρατίθεται ο αλγόριθμος των k-μέσων (Αλγόριθμος 8.1).

Αλγόριθμος 8.1. *Αλγόριθμος k-μέσων*

- Επιλογή αρχικών εκτιμήσεων για τις θέσεις των κέντρων.
- Επανάληψη. Σε κάθε βήμα t:
  - Ανάθεση κάθε σημείου $\boldsymbol{x}_i$ στην ομάδα $\Omega_m^{(t)}$ της οποίας το κέντρο απέχει τη μικρότερη απόσταση από το $\boldsymbol{x}_i$
    $$\mathrm{d}(\boldsymbol{x}_i, \boldsymbol{c}_m^{(t)}) = \min_j \mathrm{d}(\boldsymbol{x}_i, \boldsymbol{c}_j^{(t)})$$
    όπου $\boldsymbol{c}_j^{(t)}$ το τρέχον κέντρο της ομάδας $\Omega_j^{(t)}$ και d Ευκλείδεια απόσταση.
  - Υπολογισμός των νέων θέσεων των κέντρων
    $$\boldsymbol{c}_j^{(t+1)} = \frac{1}{N_j^{(t)}} \sum_{\boldsymbol{x}_i \in \Omega_j^{(t)}} \boldsymbol{x}_i$$
    όπου $N_j^{(t)}$ ο αριθμός σημείων στην ομάδα $\Omega_j^{(t)}$.

  μέχρι την επίτευξη επιθυμητής ακρίβειας για τα $\boldsymbol{c}_j^{(t)}$.

8.1.4 Οδηγοί Φορτίου

Για να μετρηθεί η επίδοση ενός συστήματος απαιτούνται τεχνικές που εφαρμόζουν φορτία στο σύστημα. Αναφέρουμε τις κυριότερες μεθόδους *οδήγησης φορτίου* (load driving).

(i) *Εσωτερική οδήγηση*: απευθείας φόρτωση προγραμμάτων στη μνήμη και εκτέλεση. Μία ακολουθία προγραμμάτων υλοποιείται σαν εργασία batch. Το βασικό πρόβλημα με την εσωτερική οδήγηση φορτίου είναι ότι δεν λαμβάνει υπόψη την επιβάρυνση λόγω επικοινωνίας των τερματικών.

(ii) *Ζωντανός χειρισμός*: υποβολή του φορτίου στο σύστημα από ανθρώπους που κάθονται στα τερματικά και εκτελούν μία προκαθορισμένη σειρά εντολών. Η μέθοδος αυτή λαμβάνει υπόψη την επιβάρυνση λόγω των τερματικών, αλλά κοστίζει πολύ (ειδικά όταν απαιτούνται πολλοί άνθρωποι για τη μελέτη συστήματος πολλών χρηστών) και εισάγει μεγάλες αποκλίσεις στις μετρήσεις λόγω της επίδρασης του ανθρώπινου παράγοντα.

(iii) *Εξομοιωτής τερματικού* (Remote Terminal Emulator – RTE): μίμηση της συμπεριφοράς πολλών χρηστών με ελεγχόμενο τρόπο από έναν υπολογιστή που συνδέεται με το υπό μελέτη σύστημα. Γενικά, ένας εξομοιωτής τερματικού είναι ένας εξειδικευμένος υπολογιστής ο οποίος προσομοιώνει τα τερματικά, την επικοινωνία, τους χρήστες και τις αιτήσεις εξυπηρέτησης που υποβάλλονται στο σύστημα. Οι εξομοιωτές τερματικού έχουν συνήθως ειδικό λειτουργικό σύστημα και διαθέτουν ειδικές γλώσσες προγραμματισμού που επιτρέπουν τον ορισμό «σεναρίων» οδήγησης φορτίου. Κατά την εκτέλεση των πειραμάτων, ο εξομοιωτής μπορεί επίσης να συγκεντρώνει στοιχεία για τις δραστηριότητες του συστήματος, από τα οποία εν συνεχεία εξάγονται στατιστικά αποτελέσματα.

(iv) *Εξομοιωτής φυλλομετρητή* (Remote Browser Emulator – RBE): Η τεχνική των εξομοιωτών τερματικού υπήρξε αρκετά δημοφιλής στην πράξη, αλλά έχασε έδαφος λόγω της ανάπτυξης τεχνικών προσαρμοσμένων σε σύγχρονα χαρακτηριστικά. Οι εξομοιωτές φυλλομετρητή, που αποτελούν εξέλιξη των εξομοιωτών τερματικού, μιμούνται τη συμπεριφορά χρηστών, οι οποίοι επισκέπτονται έναν διαδικτυακό τόπο και ενδεχομένως πραγματοποιούν συναλλαγές. Η λειτουργία του εξομοιωτή παράγει την ίδια κίνηση μηνυμάτων http που θα δημιουργούσε ένας πραγματικός χρήστης μέσω φυλλομετρητή. Ο εξομοιωτής προσομοιώνει συνεδρίες χρηστών με διαφορετικές συνδέσεις προς το υπό μελέτη σύστημα, σύμφωνα με συγκεκριμένα σενάρια που βασίζονται σε αποτελέσματα πραγματικών μετρήσεων. Πέραν του τυχαίου χρόνου σκέψης, το σενάριο κάθε χρήστη υλοποιεί διαδρομές πλοήγησης στις

σελίδες του διαδικτυακού τόπου σύμφωνα με δεδομένη κατανομή πιθανότητας για κάθε επιλογή του χρήστη. Η συμπεριφορά του χρήστη μπορεί να περιγραφεί με τη βοήθεια ενός γράφου (Customer Behaviour Model Graph – CBMG), του οποίου οι κορυφές παριστάνουν σελίδες του τόπου (αντίστοιχα ενέργειες του χρήστη) και οι ακμές παριστάνουν πιθανότητες επιλογής. Η τεχνική αυτή χρησιμοποιείται σε διάφορα σύγχρονα προγράμματα αναφοράς (benchmarks) για την αποτίμηση εξυπηρετητών Ιστού. Μπορούμε να αναφέρουμε τα προγράμματα SPECweb και TPC-W (που ήταν σε ισχύ μέχρι πρόσφατα), τα οποία περιελάμβαναν φορτίο εφαρμογών ηλεκτρονικού εμπορίου.

8.1.5 Συλλογή Δεδομένων

Άμεση καταμέτρηση, δηλαδή απλή μέτρηση του πλήθους των εμφανίσεων ενός γεγονότος, π.χ. μέτρηση του αριθμού προσπελάσεων στον δίσκο που πραγματοποιεί ένα πρόγραμμα.

Έμμεση καταμέτρηση, δηλαδή μέτρηση των τιμών ορισμένων παραμέτρων που σχετίζονται με ένα γεγονός. Με χρήση των τιμών αυτών υπολογίζονται οι δείκτες, π.χ. μέτρηση τιμών του μεγέθους των αρχείων που ζητάει ένα πρόγραμμα με στόχο τον υπολογισμό του μέσου μεγέθους αρχείου. Γενικότερα, όταν ένας δείκτης δεν μπορεί να μετρηθεί άμεσα, υπολογίζεται έμμεσα βάσει κάποιου άλλου δείκτη που είναι άμεσα προσβάσιμος.

Σύνθετη καταμέτρηση, δηλαδή συνδυασμός επιμέρους μετρήσεων για τον προσδιορισμό ενός συγκεντρωτικού δείκτη που χαρακτηρίζει συνολικά την επίδοση ενός αντικειμένου (π.χ. προγράμματος, συστήματος ή συστατικού).

Για τη μέτρηση των διαφόρων τύπων γεγονότων μπορούν να χρησιμοποιηθούν εργαλεία, τα οποία ακολουθούν διάφορους μηχανισμούς και αρχές λειτουργίας. Ένα σημαντικό στοιχείο για κάθε μηχανισμό είναι η *επιβάρυνση* (overhead) που προκαλείται στο υπό μελέτη σύστημα, η οποία αυξάνεται με τον βαθμό λεπτομέρειας και την ανάλυση της μέτρησης.

Μέτρηση οδηγούμενη από τα γεγονότα (event-driven). Ο μηχανισμός μέτρησης ενεργοποιείται κάθε φορά που συμβαίνει γεγονός του συγκεκριμένου τύπου. Συνήθως, η μέτρηση περιλαμβάνει απλές ενέργειες, όπως άμεση καταμέτρηση των γεγονότων, και παρέχει περιορισμένες πληροφορίες. Η επιβάρυνση του συστήματος λόγω της μέτρησης είναι ανάλογη της συχνότητας των γεγονότων, συνεπώς τα εργαλεία αυτού του τύπου θεωρούνται κατάλληλα για γεγονότα χαμηλής συχνότητας.

Αποτύπωση (tracing). Ο μηχανισμός είναι παρόμοιος με τον προηγούμενο, με τη διαφορά ότι —πέραν της καταγραφής/καταμέτρησης των συμβάντων— αποθηκεύεται και ένα τμήμα της κατάστασης του συστήματος με κάθε γεγονός, παρέχοντας έτσι μια χρονικά διατεταγμένη λίστα με λεπτομερείς πληροφορίες. Η τεχνική αυτή, επομένως, προκαλεί σημαντική επιβάρυνση σε χρόνο και χώρο.

Δειγματοληψία (sampling). Σε αντίθεση με τους προηγούμενους τρόπους, πρόκειται για καταγραφή τμήματος της κατάστασης του συστήματος σε τακτά διαστήματα. Επομένως, η επιβάρυνση δεν εξαρτάται από τη συχνότητα των γεγονότων, αλλά από τη συχνότητα δειγματοληψίας. Επίσης, γεγονότα χαμηλής συχνότητας μπορεί να μην εντοπιστούν από τον μηχανισμό μέτρησης, οποίος έχει στατιστικό χαρακτήρα.

8.2 Εποπτεία

Ο *επόπτης* (monitor) είναι εργαλείο που χρησιμοποιείται για την παρακολούθηση των δραστηριοτήτων ενός συστήματος. Γενικά, οι επόπτες παρατηρούν την επίδοση του συστήματος, συλλέγουν στατιστικά στοιχεία, αναλύουν τα δεδομένα και απεικονίζουν τα αποτελέσματα [6].

Όπως συζητήθηκε στην προηγούμενη ενότητα σε σχέση με τις μετρήσεις, συνήθως, η λειτουργία ενός επόπτη διαταράσσει ελαφρά τη λειτουργία του συστήματος καθόσον, π.χ., απαιτεί χρήση της κεντρικής

μονάδας ή χώρο στον δίσκο. Αυτή η κατανάλωση πόρων του συστήματος από τον επόπτη αποτελεί *επιβάρυνση* (overhead) η οποία θα πρέπει να είναι όσο μικρότερη γίνεται. Το *πεδίο* (domain) ενός επόπτη είναι το σύνολο των δραστηριοτήτων του συστήματος που μπορεί να παρακολουθήσει. Οι δραστηριότητες αυτές περιγράφονται με τη βοήθεια *γεγονότων*, δηλαδή αλλαγών της κατάστασης του συστήματος. Η μέγιστη συχνότητα γεγονότων που μπορεί να παρακολουθήσει σωστά ο επόπτης αποτελεί τον *ρυθμό εισόδου*. Η *ανάλυση* (resolution) του επόπτη είναι το ελάχιστο μέγεθος πληροφορίας που μπορεί να διακρίνει.

Οι βασικές εφαρμογές των εποπτών σχετίζονται με την αποδοτικότητα, την ορθότητα και τη διαθεσιμότητα των υπηρεσιών που παρέχει το σύστημα. Διακρίνουμε αντίστοιχα:

(i) *Εποπτεία επίδοσης* (Performance monitoring): ποσοτικός προσδιορισμός της ποιότητας των παρεχομένων υπηρεσιών (ρυθμός απόδοσης, χρόνος απόκρισης, βαθμός χρησιμοποίησης πόρων).

(ii) *Εποπτεία σφαλμάτων* (Error monitoring): παροχή στατιστικών στοιχείων που σχετίζονται με εσφαλμένη παροχή υπηρεσιών (προσδιορισμός των «αναξιόπιστων» συστατικών του συστήματος).

(iii) *Εποπτεία διάταξης* (Configuration monitoring): παρακολούθηση της λειτουργίας ή μη λειτουργίας συστατικών του συστήματος (σύνδεση και αποσύνδεση συστατικών από την τρέχουσα διάταξη).

Ανάλογα με το επίπεδο υλοποίησης διακρίνουμε *επόπτες λογισμικού* (software monitors), *επόπτες υλικού* (hardware monitors), *επόπτες καλωδιωμένης λογικής* (firmware monitors) και *υβριδικούς επόπτες* (hybrid monitors). Οι δύο πρώτες κατηγορίες, που είναι οι βασικές, σχετίζονται με πληροφορίες υψηλού επιπέδου (λειτουργικό σύστημα) και χαμηλού επιπέδου (ηλεκτρικά σήματα), αντίστοιχα.

Ένας άλλος τρόπος διάκρισης βασίζεται στην αμεσότητα απόκρισης του επόπτη. Η έξοδος μπορεί να προβάλλεται σε πραγματικό χρόνο ή να υπολογίζεται μετά τη συλλογή και ανάλυση των δεδομένων. Έχουμε, επομένως, λειτουργία *on-line* και *batch*, αντίστοιχα.

8.2.1 Επόπτες Λογισμικού

Οι επόπτες λογισμικού χρησιμοποιούνται για την παρακολούθηση λειτουργικών συστημάτων και λογισμικού υψηλού επιπέδου (π.χ. δίκτυα, βάσεις δεδομένων). Σε κάθε ενεργοποίηση του επόπτη εκτελείται ένας αριθμός εντολών ο οποίος είναι καθοριστικός για τον ρυθμό εισόδου. Π.χ., αν ο επόπτης εκτελεί 100 εντολές ανά γεγονός, κάθε ενεργοποίησή του θα απαιτεί 0,1 msec σε επεξεργαστή 1 MIPS. Συνεπώς, αν επιθυμούμε η επιβάρυνση να μην ξεπερνά το 1%, ο επόπτης θα πρέπει να ενεργοποιείται ανά διαστήματα μεγαλύτερα των 10 msec, ή ισοδύναμα ο ρυθμός εισόδου του επόπτη θα πρέπει να είναι μικρότερος από 100 γεγονότα ανά sec. Γενικά, ένας επόπτης λογισμικού έχει χαμηλή ανάλυση χρόνου και αδυνατεί να παρακολουθήσει γεγονότα σε επίπεδο υλικού.

Διακρίνουμε τρεις μηχανισμούς ενεργοποίησης ενός επόπτη λογισμικού, σε αναλογία με τους μηχανισμούς μετρήσεων που περιγράφηκαν νωρίτερα.

(i) Διακοπές στο επίπεδο του λογισμικού οι οποίες προκαλούνται από γεγονότα και ενεργοποιούν μία ρουτίνα συλλογής δεδομένων (event-driven monitor).

(ii) Λειτουργία του επεξεργαστή σε κατάσταση αποτύπωσης (trace mode). Ο μηχανισμός αυτός, ο οποίος είναι διαθέσιμος σε πολλούς επεξεργαστές, επιτρέπει την ενεργοποίηση μιας ρουτίνας συλλογής δεδομένων μετά από κάθε εντολή του επεξεργαστή και συνεπάγεται μεγάλη επιβάρυνση.

(iii) Διακοπές προκαλούμενες από το χρονόμετρο του συστήματος οι οποίες ενεργοποιούν μία ρουτίνα συλλογής δεδομένων (clock-driven monitor). Πρόκειται για μηχανισμό δειγματοληψίας που είναι κατάλληλος για υψηλές συχνότητες γεγονότων.

Στη συνέχεια περιγράφονται δύο τύποι εποπτών λογισμικού οι οποίοι είναι ευρύτατα διαδεδομένοι στην πράξη και παρέχουν χρήσιμες πληροφορίες για τη λειτουργία των προγραμμάτων και τη χρήση των πόρων του συστήματος.

8.2.2 Λογιστική Καταγραφή και Εποπτεία Προγραμμάτων

Τα *λογιστικά ημερολόγια* (accounting logs) είναι προγράμματα συνήθως ενσωματωμένα στο σύστημα. Αν και αρχικά αναπτύχθηκαν για λόγους λογιστικής χρέωσης, παρέχουν πληροφορίες σχετικές με τη χρήση και την επίδοση του συστήματος. Η συλλογή δεδομένων γίνεται κατά την κανονική λειτουργία του συστήματος και η πρόσθετη επιβάρυνση είναι μικρή. Τα βασικά μειονεκτήματα είναι ότι γενικά δεν συνοδεύονται από ρουτίνες στατιστικής ανάλυσης, δεν προσφέρουν υψηλό βαθμό λεπτομέρειας και ακρίβειας, και δεν παρέχουν πληροφορίες προσανατολισμένες στο σύστημα (π.χ. μήκη ουρών, βαθμό χρησιμοποίησης μονάδων).

Τυπικά δεδομένα λογιστικής καταγραφής για κάθε ενεργοποίηση προγράμματος είναι:

- Ώρα έναρξης προγράμματος
- Ώρα λήξης προγράμματος
- Χρόνος ΚΜΕ
- Αριθμός εγγραφών στον δίσκο – συνολικός αριθμός bytes
- Αριθμός αναγνώσεων από τον δίσκο – συνολικός αριθμός bytes
- Αριθμός εγγραφών στο τερματικό – συνολικός αριθμός bytes
- Αριθμός αναγνώσεων από το τερματικό – συνολικός αριθμός bytes
- Αριθμός αναγνώσεων σελίδων – συνολικός αριθμός σελίδων

Με βάση τα παραπάνω, η κατανάλωση των πόρων του συστήματος για κάθε πρόγραμμα μπορεί να εκφραστεί με τέσσερις τρόπους:

(i) *ανά ενεργοποίηση*: μοντελοποίηση της συμπεριφοράς των προγραμμάτων, κατασκευή συνθετικών φορτίων.

(ii) *ποσοστό επί του συνόλου των πόρων που καταναλώνονται από όλα τα προγράμματα*: συμβολή του προγράμματος στη συνολική κατανάλωση, δυνατότητα βελτίωσης της επίδοσης του συστήματος.

(iii) *ρυθμός κατανάλωσης πόρων ανά sec*: ένταση κατανάλωσης πόρων, προσδιορισμός του αριθμού χρηστών που μπορεί να υποστηρίξει το σύστημα με βάση τη χωρητικότητα των πόρων.

(iv) *ρυθμός κατανάλωσης πόρων ανά CPU-sec*: ο δείκτης αυτός είναι λιγότερο μεταβλητός από τον προηγούμενο γιατί εξαρτάται μόνο από τον χρόνο ΚΜΕ του προγράμματος και όχι από το συνολικό χρόνο εκτέλεσης που επηρεάζεται από το φορτίο του συστήματος.

Μία δεύτερη κατηγορία επόπτη λογισμικού που χρησιμοποιείται πολύ στην πράξη είναι οι *επόπτες προγραμμάτων* διαφόρων τύπων (program monitors, profilers, analyzers, optimizers). Ορισμένες από τις λειτουργίες τους είναι η *αποτύπωση* της πορείας εκτέλεσης ενός προγράμματος (tracing), ο *προσδιορισμός του χρόνου* που δαπανάται στις διάφορες μονάδες του προγράμματος (timing), η *ρύθμιση* (tuning) του προγράμματος για βέλτιστη επίδοση, ή ο *έλεγχος ισχυρισμών* (assertion checking) για την επαλήθευση ιδιοτήτων του προγράμματος.

Το *προφίλ* (profile) ενός προγράμματος προκύπτει από τη μέτρηση του ποσοστού επί του συνολικού χρόνου, το οποίο αναλώνεται σε συγκεκριμένες καταστάσεις. Οι επόπτες προγράμματος επιτρέπουν την παρακολούθηση του προγράμματος σε διάφορα επίπεδα λεπτομέρειας (ενότητες, υποπρογράμματα, εντολές) και παρέχουν αντίστοιχη απεικόνιση των αποτελεσμάτων. Η μέτρηση μπορεί να πραγματοποιηθεί σύμφωνα με τις τεχνικές αποτύπωσης και δειγματοληψίας που αναφέρθηκαν παραπάνω. Οι μηχανισμοί καταγραφής ή δειγματοληψίας που χρησιμοποιούνται από τον επόπτη μπορούν να υλοποιηθούν με διάφορες τεχνικές:

- ενσωματώνονται από τον προγραμματιστή στο πηγαίο πρόγραμμα με προσθήκη εντολών πριν από τη μετάφραση,

- προστίθενται στον εκτελέσιμο κώδικα κατά τη μετάφραση (μπορεί να υπάρχει αντίστοιχη επιλογή λειτουργίας του μεταγλωττιστή),
- προστίθενται στον εκτελέσιμο κώδικα μετά τη μετάφραση (ενδεχομένως με χρήση κατάλληλων εργαλείων λογισμικού),
- ενεργοποιούνται κατευθείαν στον χρόνο εκτέλεσης (δημιουργία εξαιρέσεων λογισμικού, τροποποίηση κώδικα εικονικής μηχανής).

Η πληροφορία του προφίλ παρέχει συνολική άποψη της συμπεριφοράς ενός προγράμματος κατά την εκτέλεση και χρησιμεύει για τον εντοπισμό των πλέον χρονοβόρων ή συχνά εκτελούμενων τμημάτων του κώδικα. Εν συνεχεία, με βάση την ανάλυση των αποτελεσμάτων, μπορούν να γίνουν κατάλληλες επεμβάσεις, όπως τροποποίηση του κώδικα ή ρύθμιση παραμέτρων (π.χ. μεγέθη ενταμιευτών, καταμερισμός χρόνου επεξεργαστή, μηχανισμοί ανάγνωσης/εγγραφής στους δίσκους κλπ). Ο στόχος είναι να επιτευχθεί βέλτιστη αξιοποίηση των πόρων του προγράμματος και σημαντική βελτίωση της συνολικής επίδοσης. Η διαδικασία αυτή μπορεί να εφαρμοστεί τόσο στο επίπεδο των ενοτήτων στο εσωτερικό ενός προγράμματος εφαρμογής όσο και μεταξύ διεργασιών (εφαρμογών) που εκτελούνται στο επίπεδο του λειτουργικού συστήματος.

8.2.3 Επόπτες Υλικού

Οι επόπτες υλικού αποτελούνται από τμήματα εξοπλισμού τα οποία συνδέονται με το υπό μελέτη σύστημα μέσω ακροδεκτών. Μπορούν να συνδεθούν σε διάφορα σημεία του συστήματος και να καταγράψουν πολλά ταυτόχρονα γεγονότα. Χαρακτηρίζονται από υψηλό ρυθμό εισόδου (τάξης $> 10^5$ ανά sec) και ανάλυση χρόνου της τάξης των λίγων nsec. Προκαλούν μικρή επιβάρυνση και λειτουργούν ανεξάρτητα από δυσλειτουργία ή βλάβη του συστήματος. Συνήθως περιλαμβάνουν τα ακόλουθα στοιχεία:

(i) Ακροδέκτες που τοποθετούνται σε δεδομένα σημεία του συστήματος και παρατηρούν σήματα.

(ii) Λογικά κυκλώματα τα οποία συνδυάζουν σήματα και ελέγχουν γεγονότα.

(iii) Μετρητές, των οποίων η τιμή αυξάνει όταν συμβούν συγκεκριμένα γεγονότα.

(iv) Συγκριτές που συγκρίνουν τιμές μετρητών ή σημάτων.

(v) Χρονόμετρο για την χρονολογική αποτύπωση γεγονότων.

(vi) Ενσωματωμένους οδηγούς για την αποθήκευση δεδομένων σε δίσκο.

Οι πλέον σύγχρονοι επόπτες είναι ολοκληρωμένες υπολογιστικές συσκευές με επεξεργαστή, μνήμη και μονάδες εισόδου/εξόδου.

8.2.4 Ιεραρχική Εποπτεία

Τα περισσότερα σύγχρονα υπολογιστικά συστήματα είναι κατανεμημένα και αποτελούνται από πλήθος συστατικών υλικού και λογισμικού. Συνεπώς, η εποπτεία τέτοιων συστημάτων είναι αναγκαστικά κατανεμημένη και βασίζεται σε μία ιεραρχική δομή λειτουργίας. Η δομή αυτή, που περιγράφεται στη συνέχεια, είναι γενική και εφαρμόζεται σε δίκτυα υπολογιστών, κατανεμημένες βάσεις δεδομένων ή συστήματα κατανεμημένης επεξεργασίας. Οι βασικές αρχές ισχύουν και για μη κατανεμημένα συστήματα. Ακολουθώντας την ιεραρχία από κάτω προς τα πάνω διακρίνουμε τα ακόλουθα επίπεδα:

(i) *Παρατήρηση*: καταγραφή ακατέργαστων δεδομένων από τα διάφορα συστατικά του συστήματος.

(ii) *Συλλογή*: συγκέντρωση δεδομένων από διάφορα σημεία παρατήρησης σε διάφορα σημεία συλλογής.

(iii) *Ανάλυση*: επεξεργασία δεδομένων που προέρχονται από διάφορα σημεία συλλογής με την εφαρμογή στατιστικών μεθόδων.

(iv) *Παρουσίαση*: διασύνδεση με τον χρήστη (απεικονίσεις σε οθόνες, εκτυπώσεις, ειδοποίηση για γεγονότα που απαιτούν επέμβαση).

(v) *Ερμηνεία*: οντότητα (άνθρωπος ή έμπειρο σύστημα) που επεξεργάζεται τα δεδομένα με ευφυή τρόπο (βάσει κανόνων) και εξάγει συμπεράσματα για τη λειτουργία του συστήματος.

(vi) *Έλεγχος*: διασύνδεση με τις διαδικασίες ελέγχου του συστήματος (αλλαγή παραμέτρων, επεμβάσεις σε επιμέρους συστατικά του συστήματος, αναδιάταξη).

(vii) *Διαχείριση*: λήψη αποφάσεων οι οποίες βασίζονται στα συμπεράσματα της εποπτείας και υλοποιούνται μέσω της άσκησης ελέγχου.

Τα δύο τελευταία επίπεδα στην παραγματικότητα δεν αποτελούν μέρος της εποπτείας, συνεργάζονται όμως στενά μαζί της και συχνά χρησιμοποιούν τον ίδιο σταθμό εργασίας. Γενικά, κάθε επίπεδο μπορεί να περιέχει περισσότερες από μία μονάδες της αντίστοιχης λειτουργίας, και υπάρχουν πολλαπλές συνδέσεις μεταξύ μονάδων διαδοχικών επιπέδων. Οι επόπτες κατανεμημένων συστημάτων είναι συνήθως υβριδικοί: η συλλογή ακατέργαστων δεδομένων υλοποιείται με υλικό, ενώ οι λειτουργίες των υψηλότερων επιπέδων με λογισμικό. Στα τρία υψηλότερα επίπεδα ενυπάρχει ο ανθρώπινος παράγων σε συνδυασμό με ειδικό εξοπλισμό και διάφορες αυτοματοποιημένες λειτουργίες.

8.3 Προγράμματα Αναφοράς

Το καταλληλότερο φορτίο για την αποτίμηση της επιδοσης ενός συστήματος θα έπρεπε να αποτελείται από τις εφαρμογές που εκτελεί συχνά ή θα κληθεί να εκτελέσει το σύστημα. Ωστόσο, η επιλογή αυτή είναι συνήθως δύσκολη στην πράξη και δαπανηρή, ιδιαίτερα όταν πρόκειται για νέο σύστημα. Ακόμη, η μελέτη επίδοσης μπορεί να αφορά τη συγκριτική αποτίμηση πολλών συστημάτων ή την αποτίμηση της καταλληλότητας συστημάτων για την ανάπτυξη νέων εφαρμογών. Η μέθοδος που ακολουθείται κατά κανόνα βασίζεται στη χρήση *προγραμμάτων αναφοράς* (benchmark programs), δηλαδή προγραμμάτων που υποκαθιστούν το πραγματικό φορτίο και επιτρέπουν εκτίμηση της επίδοσης του συστήματος.[1] [2] Η ορθότητα της εκτίμησης χαρακτηρίζει την ποιότητα του προγράμματος αναφοράς ή —ισοδύναμα— την πιστότητα με την οποία το πρόγραμμα αναφοράς αντιπροσωπεύει τις πραγματικές εφαρμογές.

Εφόσον τα χαρακτηριστικά των εφαρμογών ποικίλλουν, έχει αναπτυχθεί αντίστοιχα ένα ευρύ φάσμα προγραμμάτων αναφοράς που διαφοροποιούνται ως προς την περιοχή εφαρμογής, το μέγεθος ή την πολυπλοκότητα. Σύμφωνα με μια υφιστάμενη κατηγοριοποίηση, διακρίνουμε δύο τύπους προγραμμάτων αναφοράς: αυτά που μετρούν την επίδοση μιας συνιστώσας ή ενός χαρακτηριστικού του συστήματος (micro-benchmarks) και αυτά που μετρούν την επίδοση του συστήματος ως ενιαίου συνόλου (macro-benchmarks). Τα προγράμματα αναφοράς μπορεί να είναι ολοκληρωμένες εφαρμογές ή απλά τμήματα κώδικα προγράμματος. Επίσης, μπορούμε να διακρίνουμε πραγματικά (φυσικά) προγράμματα, προερχόμενα από πραγματικές εφαρμογές, και συνθετικά (τεχνητά) προγράμματα, που σχεδιάζονται έτσι ώστε η συμπεριφορά τους να προσομοιάζει σε αυτήν των πραγματικών εφαρμογών.

8.3.1 Τύποι Προγραμμάτων Αναφοράς

Διάφοροι τύποι φορτίων έχουν χρησιμοποιηθεί κατά καιρούς για τη μελέτη και σύγκριση υπολογιστικών συστημάτων. Τα χαρακτηριστικά των φορτίων αυτών σχετίζονται στενά με την εξέλιξη του υλικού και του λογισμικού των συστημάτων.

(i) *Εντολή πρόσθεσης*

Τα πρώτα χρόνια της ιστορίας των υπολογιστών, όταν ο επεξεργαστής ήταν το πιο σημαντικό συστατικό του συστήματος, ο χρόνος εκτέλεσης μιας πρόσθεσης μπορούσε να θεωρηθεί ενδεικτικός της ταχύτητας του επεξεργαστή.

[1] Ο όρος "benchmark" προέρχεται από την τοπογραφία και δηλώνει ένα σημάδι σμιλευμένο σε σταθερή λίθινη κατασκευή, το οποίο δείχνει το υψόμετρο στη θέση αυτή και χρησιμεύει ως *σημείο αναφοράς* για μεταγενέστερες μετρήσεις.

[2] Σημειώνεται ότι για την απόδοση του όρου "benchmark" στα ελληνικά χρησιμοποιείται και ο όρος «μετροπρόγραμμα.»

(ii) *Μίγματα εντολών* (Instruction mixes)

Καθώς αυξανόταν η πολυπλοκότητα των εντολών ήταν απαραίτητη μία λεπτομερέστερη περιγραφή του φορτίου. Ένα μίγμα εντολών αποτελείται από μία ομάδα εντολών που συνοδεύονται από τη σχετική συχνότητα χρήσης τους, όπως έχει μετρηθεί σε πραγματικά συστήματα. Δεδομένου του χρόνου εκτέλεσης των διαφόρων εντολών σε ένα σύστημα, μπορεί να υπολογιστεί ο μέσος χρόνος εκτέλεσης εντολής, σταθμισμένος με βάση τις συχνότητες. Από τα διάφορα μίγματα εντολών που αναπτύχθηκαν, πιο συχνά αναφέρεται το μίγμα του Gibson (1959) που περιελάμβανε 13 είδη εντολών.

Τα μίγματα εντολών δεν επαρκούν για να χαρακτηρίσουν τη λειτουργία των σημερινών επεξεργαστών που στηρίζονται σε πολύπλοκους μηχανισμούς και τεχνικές. Επιπλέον, τα μίγματα εντολών εκφράζουν μόνο την ταχύτητα του επεξεργαστή και όχι τη συνολική επίδοση ενός συστήματος. Παρ΄ όλα αυτά, παρέχουν έναν απλό δείκτη επίδοσης και μπορεί να θεωρηθούν χρήσιμα σε ορισμένες περιπτώσεις. Το αντίστροφο του μέσου χρόνου εκτέλεσης εντολής αναφέρεται συνήθως ως η ταχύτητα του επεξεργαστή σε MIPS (Millions of Instructions Per Second) ή MFLOPS (Millions of Floating-Point Operations Per Second).

(iii) *Συνθετικά προγράμματα*

Τα συνθετικά προγράμματα είναι τεχνητά προγράμματα που περιέχουν επαναλήψεις εντολών ή κλήσεων υπηρεσιών του λειτουργικού συστήματος. Το βάρος των διαφόρων κλήσεων στο πρόγραμμα ρυθμίζεται εύκολα με τη βοήθεια παραμέτρων που ελέγχουν τον αριθμό των επαναλήψεων. Συνεπώς, θα μπορούσαν να θεωρηθούν επέκταση ή συμπλήρωμα των μιγμάτων εντολών. Τα συνθετικά προγράμματα αναπτύσσονται γρήγορα και μπορούν να περιέχουν και μηχανισμούς μέτρησης. Το μειονέκτημά τους είναι ότι δεν κάνουν αντιπροσωπευτική χρήση της μνήμης και των δίσκων, αφού δεν λαμβάνουν υπόψη τις αλληλεπιδράσεις που οφείλονται στη διάταξη των εντολών.

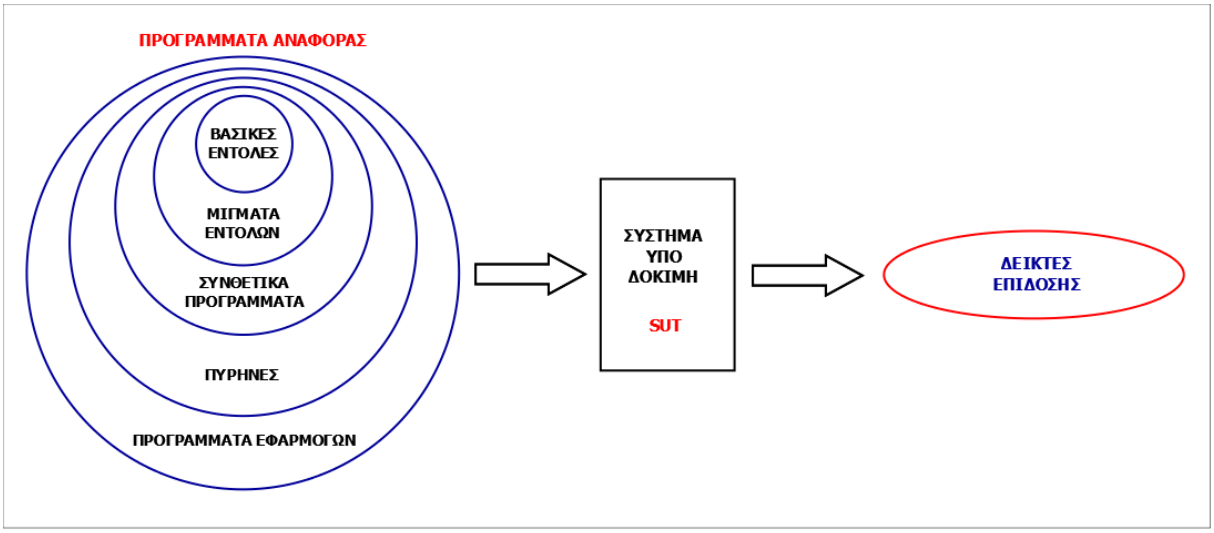
(iv) *Πυρήνες* (Kernels)

Σε αντίθεση με τα μίγματα εντολών και τα συνθετικά προγράμματα, μία ομάδα εντολών που συγκροτεί μία λειτουργία του συστήματος μπορεί να θεωρηθεί περισσότερο αντιπροσωπευτική του φορτίου. Μία τέτοια ρουτίνα ονομάζεται πυρήνας και συνήθως αποτελεί τμήμα μιας ευρύτερης εφαρμογής. Συνηθισμένοι πυρήνες είναι η αντιστροφή πίνακα, διαδικασίες ταξινόμησης και αναζήτησης, το Κόσκινο του Ερατοσθένους, η συνάρτηση του Ackermann κλπ. Οι πυρήνες είναι δημοφιλή προγράμματα που καθιερώθηκαν για τη σύγκριση συστημάτων, αν και η εμβέλειά τους είναι περιορισμένη. Συνήθως, ελέγχουν μόνο την επίδοση του επεξεργαστή, καθόσον δεν περιλαμβάνουν εντολές εισόδου/εξόδου ή κλήσεις του λειτουργικού συστήματος.

(v) *Προγράμματα αναφοράς εφαρμογών* (Application benchmarks)

Όταν το υπό μελέτη υπολογιστικό σύστημα πρόκειται να χρησιμοποιηθεί για ένα συγκεκριμένο είδος εφαρμογών, το φορτίο μπορεί να συγκροτηθεί από ένα αντιπροσωπευτικό υποσύνολο λειτουργιών της εφαρμογής. Τέτοιου είδους προγράμματα αναφοράς, τα οποία προέρχονται από πραγματικά φορτία, επιτρέπουν τη μελέτη της συμπεριφοράς όλων των πόρων του συστήματος. Αν και η αρχική έννοια του όρου 'benchmark' είναι η παραπάνω, στη βιβλιογραφία ο όρος χρησιμοποιείται συνήθως για να χαρακτηρίσει διάφορα είδη φορτίων που χρησιμοποιούνται για σύγκριση συστημάτων, συμπεριλαμβανομένων των πυρήνων και των συνθετικών προγραμμάτων.

Όπως αναφέρθηκε νωρίτερα, η επιλογή του κατάλληλου φορτίου για μια μελέτη επίδοσης μπορεί να βασιστεί στην ιεραρχική αντίληψη της δομής του συστήματος, με αντιστοιχία στις παρεχόμενες υπηρεσίες. Π.χ., ακολουθώντας τη σειρά των επιπέδων: εφαρμογή, λειτουργικό σύστημα, επεξεργαστής, αριθμητική-λογική μονάδα, τα κατάλληλα φορτία αναφοράς ανά επίπεδο θα ακολουθούσαν αντίστοιχα την ιεραρχία: benchmark εφαρμογής, συνθετικό πρόγραμμα, μίγμα εντολών, συχνότητα αριθμητικών εντολών (Σχ. 8.2).



Σχήμα 8.2: *Ιεραρχία προγραμμάτων αναφοράς.*

8.3.2 Συγκριτική Αξιολόγηση

Η αξιολόγηση της εκτέλεσης ενός προγράμματος αναφοράς σε ένα υπολογιστικό σύστημα βασίζεται κατά κύριο λόγο στη μέτρηση του αναγκαίου χρόνου. Υπάρχουν, όμως, και διαφορετικές προσεγγίσεις [8].

- Στη συνήθη περίπτωση, ο προσδιορισμός του δείκτη επίδοσης βασίζεται στη μέτρηση του *χρόνου για την εκτέλεση μιας σταθερής ποσότητας υπολογισμού*. Ο δείκτης αυτός αντιπροσωπεύει ένα μέγεθος *ταχύτητας* (ρυθμού απόδοσης), όπου η ποσότητα υπολογισμού εκφράζει τη φύση των υπηρεσιών που παρέχει το σύστημα, π.χ. MIPS ή transactions/sec.

- Μια εναλλακτική αντίληψη βασίζεται στη μέτρηση της *ποσότητας υπολογισμού που εκτελείται σε σταθερό χρόνο*. Για παράδειγμα, σε μια επιστημονική εφαρμογή, ο δείκτης επίδοσης μπορεί να είναι η ορθότητα (accuracy) υπολογισμού ενός μεγέθους, η οποία επιτυγχάνεται σε δεδομένο χρονικό διάστημα (υποθέτοντας ότι η ορθότητα του αποτελέσματος αυξάνει με την ποσότητα του εκτελούμενου υπολογισμού).

- Μια τρίτη προσέγγιση είναι ο *συνδυασμός* των παραπάνω, δηλαδή όταν ο δείκτης επίδοσης είναι συνάρτηση τόσο του χρόνου εκτέλεσης όσο και της ποσότητας υπολογισμού.

Όπως αναφέρθηκε παραπάνω, η πλέον διαδεδομένη στρατηγική είναι αυτή της μέτρησης του χρόνου για σταθερή ποσότητα υπολογισμού. Η έννοια της σταθερής ποσότητας υπολογισμού επιβάλλει ένα άνω φράγμα στη βελτίωση επίδοσης που μπορεί να επιτευχθεί αν αναβαθμιστεί ένα τμήμα του συστήματος. Η ιδέα διατυπώθηκε για πρώτη φορά το 1967 σε ένα σύντομο άρθρο του G. Amdahl [3].

Ο νόμος του Amdahl

Η θέση του Amdahl αφορούσε τη δυνατότητα βελτίωσης της επίδοσης λόγω της χρήσης πολλών επεξεργαστών, αλλά έχει γενικότερη εφαρμογή και καθιερώθηκε ως *νόμος*. Αυτό που ορίζει είναι ότι η συνολική βελτίωση επίδοσης (επιτάχυνση) της εκτέλεσης ενός προγράμματος περιορίζεται από το τμήμα του προγράμματος που δεν επηρεάζεται από τις αλλαγές που έγιναν στο υπολογιστικό σύστημα. Έστω T ο συνολικός χρόνος εκτέλεσης του προγράμματος στην αρχική διάταξη του συστήματος. Θα υποθέσουμε ότι επιτελείται μια αλλαγή στο σύστημα, η οποία μειώνει τον χρόνο εκτέλεσης ενός μέρους του προγράμματος κατά έναν παράγοντα p. Στην περίπτωση της παραλληλίας το p αντιπροσωπεύει τον αριθμό των επεξεργαστών, αν υποθέσουμε ότι επιτυγχάνεται η μέγιστη επιτάχυνση (ίση με p) για το τμήμα του κώδικα που μπορεί να παραλληλοποιηθεί. Θα υποθέσουμε ότι ένα κλάσμα α του συνολικού κώδικα προγράμματος δεν επηρεάζεται

από την αλλαγή του συστήματος, ενώ το υπόλοιπο κλάσμα $1-\alpha$ είναι αυτό που επηρεάζεται. Ο νέος συνολικός χρόνος θα αποτελείται από δύο συνιστώσες:

$$T' = \alpha T + (1-\alpha)T/p$$

και η επιτάχυνση λόγω αναβάθμισης θα είναι:

$$S = \frac{T}{T'} = \frac{1}{1/p + \alpha(1-1/p)} \tag{8.1}$$

Είναι εύκολο να διαπιστώσει κανείς ότι το όριο του S για $p \to \infty$ είναι ίσο με $1/\alpha$, δηλαδή όσο μεγάλη και αν είναι η βελτίωση, η συνολική επίδοση φράσσεται από το τμήμα του κώδικα που δεν επηρεάζεται από την αλλαγή (το τμήμα αυτό είναι το σειριακό μέρος του προγράμματος στην περίπτωση της παραλληλίας).

Σχετικά με την επίδοση των παράλληλων συστημάτων, έχει αναπτυχθεί παραλλαγή του νόμου του Amdahl, προσαρμοσμένη στην εναλλακτική της μεταβλητής ποσότητας υπολογισμού σε σταθερό χρόνο. Σύμφωνα με την παραλλαγή αυτή, η επίδοση του παράλληλου συστήματος δεν περιορίζεται από το σειριακό τμήμα του κώδικα.

8.3.3 Παραδείγματα Προγραμμάτων Αναφοράς

Μεταξύ των περισσότερο δημοφιλών προγραμμάτων αναφοράς, που έχουν χρησιμοποιηθεί κατά καιρούς, περιλαμβάνονται:

- Το Κόσκινο του Ερατοσθένους: για σύγκριση μικροεπεξεργαστών, προσωπικών υπολογιστών και γλωσσών προγραμματισμού.
- Η συνάρτηση του Ackermann:

$$A(m,n) = \begin{cases} n+1 & \text{αν } m = 0 \\ A(m-1,1) & \text{αν } n = 0 \\ A(m-1, A(m,n-1)) & \text{διαφορετικά} \end{cases}$$

  χρησιμοποιείται για τον έλεγχο του μηχανισμού κλήσεων υποπρογραμμάτων σε γλώσσες προγραμματισμού. Συνήθως αποτιμάται η $A(3,n)$, για n από 1 έως 6, η οποία δίνει αποτέλεσμα $2^{n+3}-3$. Ο αριθμός των αναδρομικών κλήσεων για τον υπολογισμό της $A(3,n)$ είναι $(512 \times 4^{n-1} - 15 \times 2^{n+3} + 9n + 37)/3$ και το μέγιστο βάθος των κλήσεων είναι $2^{n+3} - 4$.

- Whetstone Kernel (British Central Computer Agency, 1975): ομάδα προγραμμάτων που αντιπροσωπεύουν μικρές τεχνικές και επιστημονικές εφαρμογές.
- LINPACK Benchmark (1983): ομάδα προγραμμάτων για την επίλυση πυκνών γραμμικών συστημάτων (συνήθως 100×100).
- Dhrystone Kernel (Siemens, 1984): διατίθεται σε C, Pascal και Ada, και χαρακτηρίζει περιβάλλοντα προγραμματισμού συστήματος.
- Debit-Credit Benchmark: Πρόκειται για ένα αρκετά παλιό πρόγραμμα (1973) του οποίου οι νεότερες εκδόσεις (μετά το 1985) οδήγησαν στην καθιέρωση προτύπων για την αξιολόγηση συστημάτων επεξεργασίας συναλλαγών (transaction processing systems). Η επίδοση μετριέται σε TPS (Transactions Per Second). Το 1988 συστήθηκε από κατασκευαστές συστημάτων το Transaction Processing Performance Council (TPC), το οποίο τυποποίησε μία σειρά από προγράμματα αναφοράς βασισμένα στο debit-credit (TPC A, TPC B, TPC C, TPC D).
- SPEC Benchmarks: Αρχικά (1990) επρόκειτο για μία σειρά από προγράμματα αναφοράς που αναπτύχθηκαν από το System Performance Evaluation Cooperative (SPEC), το οποίο συστήθηκε από τους μεγαλύτερους κατασκευαστές συστημάτων προκειμένου να δημιουργηθεί ένα πρότυπο σύνολο

προγραμμάτων αναφοράς. Η σειρά περιελάμβανε 10 προγράμματα αναφοράς από διάφορες τεχνικές και επιστημονικές εφαρμογές. Για κάθε πρόγραμμα μετριέται ο ρυθμός απόδοσης του υπό μελέτη συστήματος ο οποίος κανονικοποιείται ως προς κάποιο υπολογιστικό σύστημα αναφοράς. Ο γεωμετρικός μέσος των ρυθμών απόδοσης για τα 10 benchmarks δίνει την επίδοση του συστήματος σε SPECmarks.

8.3.4 Οργανισμοί Πιστοποίησης

Το SPEC [1], όπως και το TPC [2], χαρακτηρίζουν την αρχή μιας τάσης στη βιομηχανία υπολογιστικών συστημάτων, σε διεθνή βάση, που έχει ως στόχο την ανάπτυξη πρότυπων προγραμμάτων αναφοράς και διαδικασιών για την αξιολόγηση και πιστοποίηση της επίδοσης συστημάτων και εφαρμογών.

8.3.4.1 SPEC

Ο οργανισμός System Performance Evaluation Cooperative, ο οποίος μετονομάστηκε σε *Standard Performance Evaluation Corporation (SPEC)*, ιδρύθηκε το 1988 από έναν μικρό αριθμό προμηθευτών σταθμών εργασίας προκειμένου να καλυφθεί η ανάγκη της αγοράς για ρεαλιστική και τυποποιημένη αξιολόγηση της επίδοσης. Πρόκειται για μη κερδοσκοπικό φορέα, ο οποίος δέχεται ως μέλη του εταιρείες και οργανισμούς που υποστηρίζουν τους σκοπούς του. Σήμερα (2015), αριθμεί περισσότερα από 60 μέλη και δημοσιεύει μεγάλο αριθμό αποτελεσμάτων επίδοσης.

Στόχος του SPEC είναι η εξασφάλιση ενός συνόλου αξιόπιστων δεικτών για συγκριτική αξιολόγηση συστημάτων. Η βασική προσέγγιση είναι η δημιουργία μιας τυποποιημένης σειράς προγραμμάτων βασισμένων σε υπάρχουσες εφαρμογές, οι οποίες μπορούν να εκτελεστούν στα υπό έλεγχο συστήματα. Το πηγαίο πρόγραμμα ματαγλωττίζεται σε κάθε σύστημα με ενδεχόμενη ρύθμιση του κώδικα για βέλτιστα αποτελέσματα. Με τον τρόπο αυτό, αφενός υπάρχει δικαιοσύνη και ομοιογένεια στη σύγκριση, αφετέρου παρέχεται ελευθερία αξιοποίησης των δυνατοτήτων κάθε συστήματος. Το SPEC περιλαμβανει τέσσερις διαφορετικές ομάδες εργασίας.

Open Systems Group (OSG) Η ομάδα αυτή προέρχεται από τον αρχικό πυρήνα του SPEC που εστίαζε τη δράση του σε δείκτες επίδοσης KME (SPECmarks). Η σημερινή δράση της OSG επικεντρώνεται σε προγράμματα για επιτραπέζια συστήματα, σταθμούς εργασίας και εξυπηρετητές. Περιλαμβάνει επιμέρους δραστηριότητες για διάφορα αντικείμενα, όπως CLOUD (εφαρμογές υπολογιστικού νέφους), KME, JAVA (εφαρμογές πελάτη/εξυπηρετητή), HANDHELD (συσκευές χειρός), POWER (ενεργειακή αποδοτικότητα), SFS (εξυπηρετητές αρχείων), WEB (εξυπηρετητές ιστού), κλπ.

Στο πλαίσιο κάθε επιμέρους δράσης αναπτύσσονται νέα προγράμματα αναφοράς, τα οποία αντικαθιστούν τα υπάρχοντα, όταν τα τελευταία παύουν να αντιστοιχούν επαρκώς στα τρέχοντα τεχνολογικά δεδομένα. Για την KME, ισχύει η σειρά προγραμμάτων αναφοράς CPU2006, που μπορεί να χρησιμοποιηθεί για τη σύγκριση συστημάτων σε φορτία με εντατικούς υπολογισμούς (compute-intensive). Περιλαμβάνει τα επιμέρους benchmarks CINT2006 (επίδοση ακεραίων) και CFP2006 (επίδοση κινητής υποδιαστολής). Τα προγράμματα αναφοράς για εξυπηρετητές ιστού προσομοιώνουν χρήστες οι οποίοι στέλνουν αιτήματα φυλλομετρητή στον εξυπηρετητή μέσω ευρυζωνικής σύνδεσης. Παρέχουν φορτία τραπεζικών συναλλαγών, ηλεκτρονικού εμπορίου κλπ.

High-Performance Group (HPG) Ασχολείται με αρχιτεκτονικές υψηλής επίδοσης, όπως συμμετρικούς πολυεπεξεργαστές, συστάδες σταθμών εργασίας (clusters), συστήματα μοιραζόμενης μνήμης, διανυσματικούς υπερυπολογιστές κλπ.

Graphics and Workstation Performance Group (GWPG) Περιλαμβάνει επιμέρους ομάδες που αναπτύσσουν προγράμματα αναφοράς για γραφικά και σταθμούς εργασίας. Τα προγράμματα βασίζονται σε εφαρμογές δημιουργίας ψηφιακού περιεχομένου και οπτικοποίησης, καθώς και ευρέως χρησιμοποιούμενες εφαρμογές CAD/CAM και γραφικών.

SPEC Research Group (RG) Πρόκειται για νέα ομάδα του SPEC, η οποία προωθεί την έρευνα για νέες μεθοδολογίες και εργαλεία με στόχο την ανάλυση επίδοσης και την αξιολόγηση νέων τεχνολογιών, ενθαρρύνοντας τη συνεργασία και αλληλεπίδραση μεταξύ ακαδημαϊκής έρευνας και βιομηχανίας.

8.3.4.2 TPC

Ο οργανισμός Transaction Processing Performance Council (TPC) είναι μη κερδοσκοπικός φορέας που ιδρύθηκε με στόχο τον ορισμό προγραμμάτων εφαρμογής για βάσεις δεδομένων και επεξεργασία συναλλαγών, καθώς και για τον εφοδιασμό της αγοράς με αντικειμενικά και αξιόπιστα δεδομένα επίδοσης. Υπάρχουν (2015) 28 μέλη του του TPC, 22 πλήρη μέλη και 6 συνδεδεμένα μέλη, κυρίως κατασκευαστές συστημάτων.

Είναι χαρακτηριστικό ότι η έννοια της *συναλλαγής* αντιμετωπίζεται στο TPC κατ΄ αναλογία με την κοινή επιχειρηματική αντίληψη του όρου: μια εμπορική ανταλλαγή αγαθών, υπηρεσιών ή χρημάτων. Μια τυπική συναλλαγή, ανάλογα με την περίπτωση, θα περιλαμβάνει ενημέρωση της βάσης δεδομένων για θέματα όπως ο έλεγχος απογραφής, κρατήσεις θέσεων ή τραπεζικές εργασίες. Σε τέτοια περιβάλλοντα, ένας αριθμός πελατών πραγματοποιούν συναλλαγές συνδεόμενοι σε μια βάση δεδομένων μέσω τερματικών σταθμών. Το TPC παράγει προγράμματα αναφοράς, τα οποία μετρούν τον ρυθμό επεξεργασίας συναλλαγών και την επίδοση της βάσης δεδομένων βάσει του συνολικού ρυθμού των συναλλαγών που εκτελούνται από το υπολογιστικό σύστημα και τη βάση δεδομένων λαμβανόμενα ως ενιαίο σύνολο.

Όπως και το SPEC, το TPC αναπτύσσει διαρκώς νέα προγράμματα αναφοράς, τα οποία αντικαθιστούν τα παλιότερα. Σήμερα (2015), τα ενεργά προγράμματα αναφοράς μπορούν να ενταχθούν στις παρακάτω κατηγορίες.

Επεξεργασία συναλλαγών – On-Line Transaction Processing (OLTP) Τα προγράμματα αυτής της κατηγορίας προσομοιώνουν ένα πλήρες περιβάλλον μέσα στο οποίο ένας πληθυσμός χρηστών εκτελεί συναλλαγές σε μια βάση δεδομένων. Διακρίνουμε τα προγράμματα αναφοράς TPC-C και TPC-E. Το πρώτο εστιάζει στις κύριες δραστηριότητες ενός τέτοιου περιβάλλοντος, που περιλαμβάνουν παραγγελίες, πληρωμές, αποθήκες κλπ., και συνθέτουν το γενικό μοντέλο επιχείρησης που διαχειρίζεται, πουλάει ή διανέμει προϊόντα ή υπηρεσίες. Ο βασικός δείκτης επίδοσης που προκύπτει είναι ο ρυθμός συναλλαγών και μετριέται σε συναλλαγές ανά πρώτο λεπτό (tpmC). Το πρόγραμμα TPC-E προσομοιώνει τις συναλλαγές μιας χρηματιστηριακής εταιρείας, οι οποίες εκτελούνται για λογαριασμό των πελατών της εταιρείας και σχετίζονται με έρευνα αγοράς, χρηματιστηριακές πράξεις, διαχείριση λογαριασμών κλπ. Το αποτέλεσμα μετριέται σε συναλλαγές ανά δευτερόλεπτο (tps).

Υποστήριξη αποφάσεων – Decision Support Περιλαμβάνονται τα προγράμματα αναφοράς TPC-H, TPC-DS και TPC-DI, τα οποία προσομοιώνουν συστήματα υποστήριξης επιχειρηματικών αποφάσεων. Εκτελούνται ερωτήσεις υψηλής πολυπλοκότητας σε μεγάλες ποσότητες δεδομένων, προκειμένου να απαντηθούν κρίσιμες ερωτήσεις επιχειρηματικής φύσης. Ειδικότερα, τα TPC-DS και TPC-DI αναπτύχθηκαν πρόσφατα και ενσωματώνουν πληθώρα χαρακτηριστικών των σύγχρονων συστημάτων υποστήριξης αποφάσεων. Οι παρεχόμενοι δείκτες επιτρέπουν την αξιολόγηση λύσεων σε συστήματα που περιλαμβάνουν —μεταξύ άλλων— *μεγάλα δεδομένα* (big data). Το benchmark TPC-DI περιέχει και τη σημαντική διάσταση της ολοκλήρωσης δεδομένων (Data Integration), δηλαδή της ενοποιημένης αναπαράστασης και διαχείρισης δεδομένων προερχόμενων από διαφορετικές πηγές.

Εικονικοποίηση – Virtualization Το πρόγραμμα αναφοράς TPC-VMS (Virtual Measurement Single System Specification) εμπλουτίζει τα προγράμματα TPC-C, TPC-E, TPC-H και TPC-DS προσθέτοντας τη μεθοδολογία προσδιορισμού δεικτών επίδοσης για εικονικοποιημένες βάσεις δεδομένων. Ειδικότερα, στο υπό έλεγχο σύστημα δημιουργείται εικονικό περιβάλλον, το οποίο παριστάνει τρεις βάσεις δεδομένων, και σε καθεμία από τις βάσεις αυτές εκτελείται το φορτίο ενός προγράμματος αναφοράς, επιλεγμένου μεταξύ των προαναφερθέντων τεσσάρων. Ένα ειδικότερο benchmark της κατηγορίας αυτής είναι το TPCx-V, ένα πρόγραμμα εικονικής μηχανής για φορτία βάσεων δεδομένων. Το TPCx-V αποτιμά την επίδοση ενός εξυπηρετητή, ο οποίος εκτελεί φορτία σε εικονικοποιημένες βάσεις δεδομένων ακολουθώντας το σχήμα και τις συναλλαγές που προδιαγράφονται στο TPC-E.

Μεγάλα Δεδομένα — Big Data Το πρόγραμμα αναφοράς TPCx-HS παρέχει δείκτες επίδοσης για την αποτίμηση υλικού και λογισμικού σε εφαρμογές μεγάλων δεδομένων με βάση σύγχρονες τεχνολογίες, όπως η πλατφόρμα λογισμικού Apache Hadoop, η οποία είναι γραμμένη σε Java και πραγματοποιεί κατανεμημένη αποθήκευση και επεξεργασία μεγάλων συνόλων δεδομένων σε συστάδες (clusters) υπολογιστών.

Συμπληρωματικά, αναφέρουμε τις υποστηρικτικές οδηγίες TPC-Energy (μεθοδολογία μετρήσεων για την ενσωμάτωση δείκτη ενεργειακής κατανάλωσης στα προγράμματα αναφοράς του TPC) και TPC-Pricing (μεθοδολογία ενιαίας τιμολογιακής πολιτικής για την ευχερή επαλήθευση των τιμών που χρησιμοποιούνται στον υπολογισμό δεικτών επίδοσης).

8.4 Διαχείριση και Σχεδιασμός

Δύο βασικά ζητήματα που σχετίζονται με την αποδοτική λειτουργία ενός υπολογιστικού συστήματος είναι η *διαχείριση* και ο *σχεδιασμός παραγωγικής ικανότητας* (capacity management, capacity planning). Η διαχείριση έχει ως στόχο τη χρησιμοποίηση των διαθέσιμων πόρων με τρόπο που να εξασφαλίζει τη μέγιστη επίδοση. Ο σχεδιασμός πρέπει να εξασφαλίζει ότι επαρκείς πόροι θα είναι διαθέσιμοι για να καλύψουν τις μελλοντικές απαιτήσεις φορτίου, με οικονομικά σύμφορο τρόπο και ικανοποιώντας τους στόχους επίδοσης. Συνεπώς, η διαχείριση αναφέρεται στο παρόν, ενώ ο σχεδιασμός στο μέλλον. Σε κάθε περίπτωση, το πρώτο στάδιο μιας μελέτης θα αφορά την καταγραφή και αποτίμηση της τρέχουσας χρήσης του συστήματος. Ο χαρακτηρισμός του φορτίου επιτρέπει τη χρήση κατάλληλων μοντέλων για την εκτίμηση ή πρόβλεψη της επίδοσης.

Η έννοια της *παραγωγικής ικανότητας* σχετίζεται με το φορτίο και μπορεί να οριστεί ως συνάρτηση του μέγιστου ρυθμού απόδοσης ή του μέγιστου αριθμού χρηστών που μπορεί να εξυπηρετήσει το σύστημα —ανάλογα με την περίπτωση. Τα σύγχρονα συστήματα έχουν πληθώρα χαρακτηριστικών, με αποτέλεσμα να μην είναι εύκολη η ενιαία αντιμετώπιση της ικανότητάς τους. Επίσης, οι διάφοροι κατασκευαστές έχουν συχνά διαφορετική αντίληψη της παραγωγικής ικανότητας και ενσωματώνουν την αντίληψή τους σε εργαλεία διαχείρισης και σχεδιασμού. Στο θέμα αυτό, είναι πολύ σημαντική η συμβολή των οργανισμών τυποποίησης για την ανάπτυξη ανεξάρτητων φορτίων και διαδικασιών αποτίμησης. Θα πρέπει να σημειωθεί ότι η μελέτη επίδοσης θα πρέπει να συνοδεύεται από στοιχεία κόστους, τα οποία επίσης αποτελούν κρίσιμο παράγοντα στην υλοποίηση της διαχείρισης και του μοντέλου επίδοσης.

Η διαχείριση ασχολείται με τη ρύθμιση της χρησιμοποίησης των πόρων, την αναπροσαρμογή της διάταξης του συστήματος και την τροποποίηση των παραμέτρων με στόχο τη βελτιστοποίηση της επίδοσης. Η διαδικασία προσαρμογής των παραμέτρων για βελτιστοποίηση ονομάζεται *ρύθμιση επίδοσης* (performance tuning) και συνήθως βασίζεται σε λεπτομερή και εξειδικευμένα μοντέλα του συστήματος.

Ο σχεδιασμός αφορά την εξέταση εναλλακτικών λύσεων για την προμήθεια νέων υπολογιστικών πόρων. Η διαδικασία του σχεδιασμού περιλαμβάνει το χαρακτηρισμό φορτίου, την πρόβλεψη επίδοσης για διάφορες λύσεις και την επιλογή της λύσης με τη μέγιστη σχέση επίδοσης/κόστους. Η πρόβλεψη της επίδοσης για διαφορετικές διατάξεις συστήματος και μελλοντικά φορτία βασίζεται συνήθως σε γενικά μοντέλα, συχνά αναλυτικά μοντέλα δικτύων αναμονής.

Η γενική μεθοδολογία σχεδιασμού παραγωγικής ικανότητας μπορεί να διατυπωθεί με τη βοήθεια των παρακάτω βημάτων [9, 10]:

- κατανόηση του περιβάλλοντος,
- χαρακτηρισμός φορτίου,
- επικύρωση μοντέλου φορτίου,
- ανάπτυξη μοντέλου επίδοσης,
- επικύρωση μοντέλου επίδοσης,
- πρόβλεψη φορτίου,

- πρόβλεψη επίδοσης,
- ανάπτυξη μοντέλου κόστους,
- πρόβλεψη κόστους,
- ανάλυση κόστους/επίδοσης.

8.4.1 Υπηρεσίες Ιστού

Ο σχεδιασμός παραγωγικής ικανότητας για υπηρεσίες Ιστού (Web services) ακολουθεί τη γενική μεθοδολογία που περιγράφηκε παραπάνω. Πέραν των γενικών κατευθύνσεων, όμως, η μεθοδολογία πρέπει να πληροί ορισμένες ειδικές προϋποθέσεις, λόγω των ιδιαιτεροτήτων των συστημάτων που παρέχουν υπηρεσίες Ιστού. Πράγματι, όπως αναπτύχθηκε στο Κεφάλαιο 4 (4.4. Μοντελοποίηση Ιστού), στα συστήματα αυτά (α) συνήθως απαιτούνται υποδομές (υλικό, λογισμικό) μεγάλου εύρους και υψηλής πολυπλοκότητας (εξυπηρετητές Ιστού, εξυπηρετητές μεσολάβησης, μηχανισμοί λανθάνουσας μνήμης, αντικατοπτρισμός, δίκτυα κλπ.) και (β) εξυπηρετείται μεγάλος αριθμός χρηστών με τυχαία και δυναμική συμπεριφορά (γεωγραφική και χρονική διασπορά, εκρηκτικά φορτία, κατανομές μεγέθους αντικειμένων κλπ.).

Συχνά, η τρέχουσα παραγωγική ικανότητα ενός συστήματος δεν είναι γνωστή, με αποτέλεσμα ο σχεδιασμός να γίνεται χωρίς κατανόηση των αναγκών και χωρίς συγκεκριμένη μεθοδολογία. Η επάρκεια της παραγωγικής ικανότητας θα πρέπει, καταρχήν, να ικανοποιεί τις απαιτούμενες τιμές των δεικτών επίδοσης (χρόνος απόκρισης, ρυθμός απόδοσης, διαθεσιμότητα κλπ.) που ορίζονται από τη *Συμφωνία Επιπέδου Υπηρεσιών* (Service-Level Agreement – SLA), εφόσον υπάρχει. Η επίτευξη της ικανότητας που προδιαγράφεται από το SLA εξαρτάται από περιορισμούς κόστους, καθώς και από υφιστάμενους τεχνολογικούς περιορισμούς και επιλογές. Η μεθοδολογία σχεδιασμού βασίζεται σε τρία μοντέλα: το *μοντέλο φορτίου*, το *μοντέλο επίδοσης* και το *μοντέλο κόστους*.

Το μοντέλο φορτίου εκφράζει τις απαιτήσεις σε πόρους και τα χαρακτηριστικά της έντασης των διαφόρων συνιστωσών του φορτίου, όπως προσδιορίζονται από τη διαδικασία χαρακτηρισμού. Μια σημαντική διάσταση σχετίζεται με την *πρόβλεψη φορτίου* (workload forecasting). Εάν υπάρχουν διαθέσιμα ιστορικά δεδομένα, οι τάσεις και διακυμάνσεις που περιέχονται σ' αυτά μπορούν να αξιοποιηθούν με τις κατάλληλες τεχνικές. Οι πλέον χρησιμοποιούμενες τεχνικές πρόβλεψης είναι διάφοροι τύποι παλινδρόμησης (γραμμικής ή μη γραμμικής), καθώς και τεχνικές κινούμενης μέσης τιμής (απλής ή εκθετικής). Επίσης, η ανάλυση των επιχειρηματικών και στρατηγικών σχεδίων του οργανισμού συσχετίζεται με παραγωγικές διαδικασίες που έχουν αντίκτυπο στις συνιστώσες του φορτίου. Π.χ. προβλεπόμενη αύξηση του προσωπικού κατά 10% αναμένεται να προκαλέσει αύξηση 20% στη χρήση του e-mail και 15% στη χρήση του Διαδικτύου.

Το μοντέλο επίδοσης υπολογίζει εκτιμήσεις των δεικτών επίδοσης με βάση την περιγραφή του συστήματος και το μοντέλο του φορτίου. Συνήθως, οι απαιτήσεις επίδοσης συνοδεύονται από απαιτήσεις διαθεσιμότητας. Η διαθεσιμότητα μπορεί να εκτιμηθεί με τη βοήθεια *μοντέλων διαθεσιμότητας* (availability models). Συνήθως, πρόκειται για απλά (μαρκοβιανά) μοντέλα, τα οποία υπολογίζουν δείκτες διαθεσιμότητας λαμβάνοντας υπόψη ιδιότητες της υποστηρικτικής υποδομής των παρεχόμενων υπηρεσιών καθώς και την αξιοπιστία καθεμιάς συνιστώσας του συστήματος. Ένα κλασικό μοντέλο διαθεσιμότητας είναι το μοντέλο αναμονής $M/M/1/K/K$ (επισκευή μηχανών), το οποίο εξετάστηκε στο Κεφάλαιο 3. Ο συνδυασμός των αποτελεσμάτων της ανάλυσης επίδοσης και της ανάλυσης διαθεσιμότητας αναφέρεται συχνά με τον τεχνητό όρο *performability analysis*.

Για τον προσδιορισμό του μοντέλου κόστους πρέπει να εντοπιστούν οι πηγές κόστους και να καθοριστεί ο τρόπος μεταβολής του κόστους ως συνάρτηση του μεγέθους και άλλων ιδιοτήτων του συστήματος. Συνήθως, το κόστος διακρίνεται σε κόστος εγκατάστασης και κόστος (ετήσιας) λειτουργίας. Διαθέτοντας μοντέλο επίδοσης και μοντέλο κόστους, μπορούμε να πραγματοποιήσουμε ανάλυση κόστους/επίδοσης, εξετάζοντας διάφορα εναλλακτικά σενάρια και υπολογίζοντας για κάθε σενάριο τον σχετικό δείκτη επίδοσης και το αντίστοιχο κόστος.

8.5 Ερμηνεία και Παρουσίαση

Η ανάλυση των μετρήσεων απαιτεί εργαλεία και τεχνικές για τη συνόψιση των δεδομένων, την ερμηνεία των αποτελεσμάτων, καθώς και για τη σαφή και κατανοητή απεικόνισή τους [6].

Κατά την παρουσίαση των αποτελεσμάτων μιας μελέτης, υπάρχει πιθανότητα να χρησιμοποιηθούν τεχνάσματα που οδηγούν σε εσφαλμένα συμπεράσματα. Ιδιαίτερη προσοχή χρειάζεται όταν χρησιμοποιούνται λόγοι για συγκριτική αξιολόγηση των αποτελεσμάτων, καθόσον είναι εύκολο να γίνουν παραπλανητικές ερμηνείες (ratio games).

8.5.1 Λόγοι

Πολύ συχνά ένας δείκτης επίδοσης εκφράζεται ως λόγος δύο ποσοτήτων. Ο χειρισμός λόγων χρειάζεται ειδική προσοχή κυρίως όσον αφορά τη σύγκριση και τη χρήση μέσων τιμών. Ο παρονομαστής ονομάζεται και *βάση* του λόγου. Λόγοι με διαφορετικές βάσεις δεν είναι συγκρίσιμοι.

Μπορούμε να διατυπώσουμε τους ακόλουθους κανόνες σχετικά με τη συνόψιση n λόγων $x_i = a_i/b_i$:

(i) Αν το άθροισμα των αριθμητών και το άθροισμα των παρονομαστών έχουν φυσική σημασία, η μέση τιμή των λόγων είναι ο λόγος των μέσων τιμών:

$$\text{Μέση τιμή}\left(\frac{a_1}{b_1}, \frac{a_2}{b_2}, \ldots, \frac{a_n}{b_n}\right) = \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} = \frac{1/n \sum_{i=1}^n a_i}{1/n \sum_{i=1}^n b_i} = \frac{\overline{a}}{\overline{b}}$$

Παράδειγμα 8.1. *Μέτρηση του βαθμού χρησιμοποίησης μιας ΚΜΕ σε διαφορετικές χρονικές περιόδους:*

Διάρκεια μέτρησης	*Βαθμός χρησιμοποίησης*
20	0,40
10	0,47
8	0,55
5	0,36
32	0,20
92	0,25

Ο μέσος βαθμός χρησιμοποίησης δεν είναι ο αριθμητικός μέσος όρος $(0,40 + 0,47 + 0,55 + 0,36 + 0,20 + 0,25)/6 = 0,37$, όπως θα μπορούσε να φανταστεί κανείς εκ πρώτης όψεως. Αυτό συμβαίνει γιατί ο βαθμός χρησιμοποίησης σε κάθε διάστημα προκύπτει ως ο λόγος του χρόνου απασχόλησης προς τη διάρκεια της μέτρησης. Επειδή οι βάσεις είναι διαφορετικές, οι λόγοι δεν είναι συγκρίσιμοι. Ο μέσος βαθμός χρησιμοποίησης υπολογίζεται ως ο λόγος του συνολικού χρόνου απασχόλησης προς τον συνολικό χρόνο μέτρησης (φυσική σημασία αθροίσματος αριθμητών και αθροίσματος παρονομαστών):

$$\text{Μέσος βαθμός χρησιμοποίησης} = \frac{8 + 4,7 + 4,4 + 1,8 + 6,4 + 23}{20 + 10 + 8 + 5 + 32 + 92} = 0,29$$

□

Δύο ειδικές περιπτώσεις του παραπάνω κανόνα είναι οι εξής:

- Αν όλοι οι λόγοι έχουν την ίδια βάση και το άθροισμα των αριθμητών έχει φυσική σημασία, η μέση τιμή των λόγων είναι ο αριθμητικός τους μέσος:

$$\text{Μέση τιμή}\left(\frac{a_1}{b}, \frac{a_2}{b}, \ldots, \frac{a_n}{b}\right) = \frac{1/n \sum_{i=1}^n a_i}{b} = \frac{1}{n}\sum_{i=1}^n \frac{a_i}{b}$$

  Π.χ., αν κάθε μέτρηση του προηγούμενου παραδείγματος έχει γίνει στην ίδια χρονική διάρκεια μπορούμε να πάρουμε κατ' ευθείαν τον αριθμητικό μέσο των βαθμών χρησιμοποίησης.

- Αν όλοι οι αριθμητές είναι ίσοι και το άθροισμα των παρονομαστών έχει φυσική σημασία, η μέση τιμή των λόγων είναι ο αρμονικός τους μέσος:

$$\text{Μέση τιμή}\left(\frac{a_1}{b_1}, \frac{a_2}{b_2}, \ldots, \frac{a_n}{b_n}\right) = \frac{a}{1/n\sum_{i=1}^{n} b_i} = \frac{n}{\sum_{i=1}^{n} b_i/a}$$

  Π.χ., αν διαθέτουμε n μετρήσεις για τον χρόνο εκτέλεσης ενός προγράμματος αναφοράς σε έναν επεξεργαστή, ο ρυθμός MIPS του i πειράματος θα είναι a/b_i, όπου a το μέγεθος του προγράμματος σε εκατομμύρια εντολές και b_i ο αντίστοιχος χρόνος εκτέλεσης. Άρα ο μέσος ρυθμός MIPS θα είναι ο γεωμετρικός μέσος των ρυθμών των πειραμάτων.

(ii) Αν ο αριθμητής και ο παρονομαστής συνδέονται με πολλαπλασιαστική σχέση του τύπου $a_i = cb_i$, όπου c είναι κατά προσέγγιση σταθερά, τότε το c μπορεί να εκτιμηθεί από το γεωμετρικό μέσο των a_i/b_i:

$$\text{Μέση τιμή}\left(\frac{a_1}{b_1}, \frac{a_2}{b_2}, \ldots, \frac{a_n}{b_n}\right) = \left(\prod_{i=1}^{n} \frac{a_i}{b_i}\right)^{1/n}$$

Π.χ., έστω ότι το μέγεθος του κώδικα n προγραμμάτων μειώνεται με τη βοήθεια ενός βελτιστοποιητή προγραμμάτων (program optimizer). Αν a_i το μέγεθος του κώδικα μετά τη βελτιστοποίηση και b_i το μέγεθος πριν από τη βελτιστοποίηση, αναμένεται να ισχύει η σχέση $a_i = cb_i$, όπου c εκφράζει την επίδραση της βελτιστοποίησης και αναμένεται να είναι ανεξάρτητο από το μέγεθος των προγραμμάτων. Άρα το c μπορεί να εκτιμηθεί ως ο γεωμετρικός μέσος των λόγων a_i/b_i.

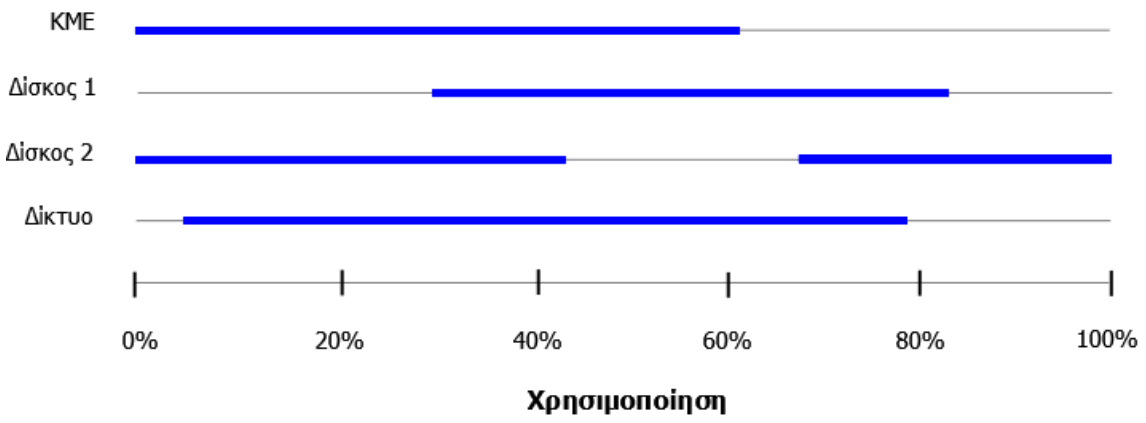
Οι λόγοι παρέχουν συχνά τη δυνατότητα εφαρμογής τεχνασμάτων. Με κατάλληλο συνδυασμό λόγων που έχουν διαφορετικές βάσεις μπορεί κανείς να αλλοιώσει τη σημασία των αποτελεσμάτων μιας μελέτης. Μία συνηθισμένη περίπτωση αφορά την σύγκριση της επίδοσης δύο ή περισσότερων συστημάτων σε διάφορα φορτία.

Παράδειγμα 8.2. *Ο χρόνος εκτέλεσης των φορτίων X και Y στα συστήματα A και B συνοψίζεται με τρεις διαφορετικούς τρόπους: (α) υπολογίζοντας τη μέση επίδοση κάθε συστήματος χωριστά και παίρνοντας τον λόγο, (β) κανονικοποιώντας την επίδοση κάθε συστήματος σε κάθε φορτίο με βάση την επίδοση του συστήματος A και παίρνοντας τη μέση τιμή των λόγων, και (γ) όπως το (β) αλλά κανονικοποιώντας με βάση το σύστημα B.*

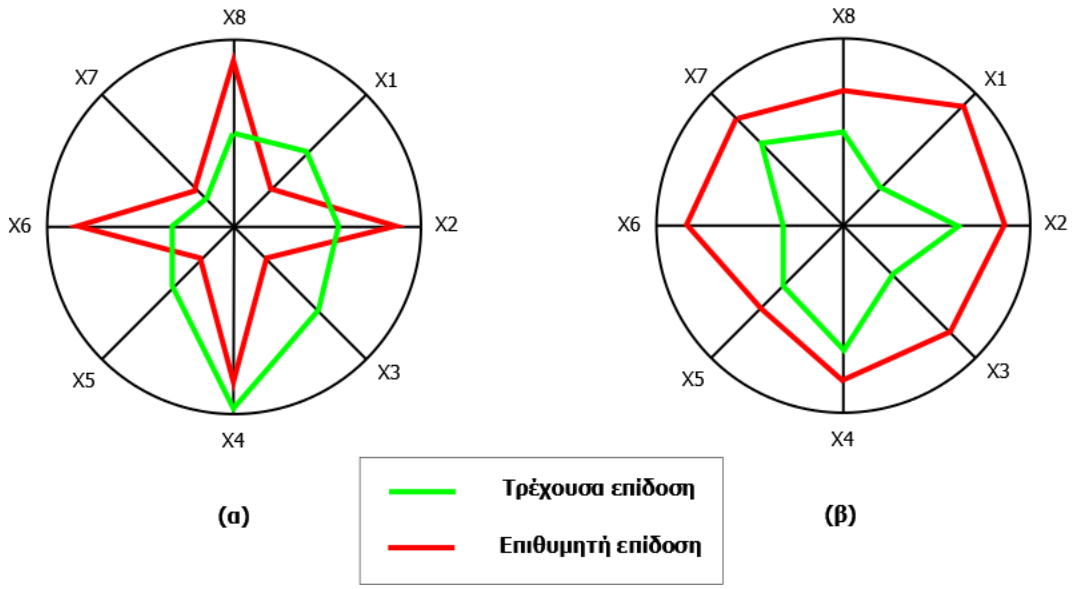
	Μετρήσεις		*Με βάση το A*		*Με βάση το B*	
Φορτίο	*A*	*B*	*A*	*B*	*A*	*B*
X	65, 42	82, 14	1, 00	1, 26	0, 80	1, 00
Y	73, 54	59, 03	1, 00	0, 80	1, 25	1, 00
Μέση τιμή	69, 48	70, 59	1, 00	1, 03	1, 03	1, 00

Οι τρεις τρόποι ανάλυσης οδηγούν σε τρία διαφορετικά συμπεράσματα: (α) το σύστημα B είναι χειρότερο (χρειάζεται 1,6% περισσότερο χρόνο από το A), (β) το σύστημα B είναι χειρότερο (χρειάζεται 3% περισσότερο χρόνο από το A), και (γ) το σύστημα A είναι χειρότερο (χρειάζεται 3% περισσότερο χρόνο από το B).

Η κύρια αιτία των αντιφατικών αποτελεσμάτων του παραδείγματος είναι η λανθασμένη προσέγγιση υπολογισμού της μέσης τιμής λόγων. Η σωστή μέθοδος για την ανάλυση τέτοιων δεδομένων βασίζεται στη χρήση τεχνικών σχεδίασης πειραμάτων. Η επίδοση γενικά επηρεάζεται από διάφορους παράγοντες, όπως το σύστημα και το φορτίο στο παράδειγμα, καθώς και από πειραματικά σφάλματα. Συνεπώς, οποιοδήποτε συμπέρασμα (π.χ. η σχετική ισχύς των δύο συστημάτων) μπορεί να εξαχθεί μόνο αφού αναπτυχθεί ένα μοντέλο που εκφράζει τις επιδράσεις των παραγόντων και των σφαλμάτων, καθώς και τις αλληλεπιδράσεις των παραγόντων. □



Σχήμα 8.3: *Διάγραμμα Gantt.*



Σχήμα 8.4: *Διαγράμματα Kiviat.*

8.5.2 Παρουσίαση Αποτελεσμάτων

Η παρουσίαση των αποτελεσμάτων είναι μία σημαντική φάση σε κάθε μελέτη επίδοσης και απαιτεί σωστή χρήση λέξεων, εικόνων και γραφικών παραστάσεων. Η σαφής και παραστατική απεικόνιση των αποτελεσμάτων της ανάλυσης διευκολύνει την κατανόηση και τη λήψη αποφάσεων. Γενικά, οι μεταβλητές που απεικονίζονται διακρίνονται σε *ποιοτικές ή κατηγορικές* και σε *ποσοτικές*. Οι ποιοτικές μπορούν περαιτέρω να διακριθούν σε διατεταγμένες και μη διατεταγμένες, ενώ οι ποσοτικές σε διακριτές και συνεχείς.

Υπάρχουν διάφοροι κανόνες που θα πρέπει να ακολουθούνται για τη σωστή γραφική απεικόνιση, ώστε να μεγιστοποιείται η χρήσιμη πληροφορία, να ελαχιστοποιείται η περιττή πληροφορία και να αποφεύγεται η δημιουργία εσφαλμένων εντυπώσεων. Εκτός από τους συνήθεις τύπους γραφικών παραστάσεων (καμπύλες, ιστογράμματα, bar charts, pie charts) υπάρχουν και τεχνικές γραφικής απεικόνισης ειδικά προσαρμοσμένες στην ανάλυση επίδοσης, όπως είναι τα διαγράμματα Gantt και τα διαγράμματα Kiviat.

Ένα βασικό μέλημα στη διαχείριση ενός υπολογιστικού συστήματος είναι η βέλτιστη χρήση των πόρων. Ο υψηλός βαθμός χρησιμοποίησης ενός πόρου συνεπάγεται ότι ο συγκεκριμένος πόρος αποτελεί στένωση (bottleneck) για την επίδοση του συστήματος. Αντίθετα, ο χαμηλός βαθμός χρησιμοποίησης συνεπάγεται αδυναμία του συστήματος να αξιοποιήσει τους πόρους του. Τα *διαγράμματα Gantt*, τα οποία γενικά παριστάνουν τη σχετική διάρκεια της ισχύος ενός αριθμού λογικών συνθηκών, μπορούν εύκολα να χρησιμοποιηθούν για να απεικονίσουν τον βαθμό χρησιμοποίησης και την επικάλυψη στη χρήση των πόρων ενός

συστήματος. Το διάγραμμα Gantt του Σχ. 8.3 παριστάνει τον βαθμό χρησιμοποίησης τεσσάρων πόρων (ΚΜΕ, δίσκοι, δίκτυο). Τα σχετικά μήκη και οι σχετικές θέσεις των ευθύγραμμων τμημάτων υποδηλώνουν τη χρονική επικάλυψη της δραστηριότητας των μονάδων.

Τα *διαγράμματα Kiviat* επιτρέπουν τη γρήγορη αναγνώριση προβλημάτων επίδοσης. Αναφέρονται, επίσης, ως διαγράμματα radar, διαγράμματα αστέρα, αραχνοειδή διαγράμματα ή πολικά διαγράμματα. Πρόκειται για κυκλικό γράφημα, οι ακτίνες του οποίου παριστάνουν δείκτες επίδοσης του υπό μελέτη συστήματος. Έστω ότι διαθέτουμε τους δείκτες $X1, X2, \ldots, Xn$ – όχι υποχρεωτικά ανεξάρτητους μεταξύ τους. (Συνήθως χρησιμοποιείται άρτιος αριθμός δεικτών.) Το σχετικό μέγεθος της τιμής κάθε δείκτη μετριέται πάνω στην ακτίνα από το κέντρο του κύκλου (μηδέν) ως την περιφέρεια (μέγιστο). Αν τα σημεία που προκύπτουν ενωθούν, σχηματίζεται ένα πολύγωνο που χαρακτηρίζει την επίδοση του συστήματος. Σύμφωνα με μία προσέγγιση, οι δείκτες επιλέγονται ώστε οι μισοί να έχουν υψηλές τιμές για καλή επίδοση (Higher Better – HB) και οι άλλοι μισοί χαμηλές (Lower Better – LB). Οι δύο αυτές κατηγορίες δεικτών τοποθετούνται εναλλάξ στις ακτίνες του κύκλου, έτσι ώστε, για ένα ιδανικό σύστημα, το διάγραμμα Kiviat να έχει τη μορφή αστέρα. Παρεκκλίσεις από τη μορφή αυτή βοηθούν στην επισήμανση προβλημάτων του συστήματος. Διαφορετικά, αν υποτεθεί ότι όλοι οι δείκτες είναι HB, το ιδανικό διάγραμμα θα έχει μορφή κανονικού πολυγώνου. Στη γενική περίπτωση, ανάλογα με τον τύπο των δεικτών, διαμορφώνεται ένα αντίστοιχο πολύγωνο που αποτελεί την επιθυμητή κατάσταση επίδοσης του συστήματος. Το «άνοιγμα» ανάμεσα στην επιθυμητή και την τρέχουσα κατάσταση, η οποία περιλαμβάνεται στο ίδιο γράφημα, προσδιορίζει τους κατάλληλους στόχους για τη ρύθμιση της λειτουργίας του συστήματος. Στο Σχ. 8.4 φαίνονται δύο περιπτώσεις διαγραμμάτων Kiviat: (α) μορφής αστέρα και (β) γενικής μορφής.

Βιβλιογραφία

[1] Standard Performance Evaluation Corporation, https://www.spec.org

[2] Transaction Processing Performance Council, https://www.tpc.org

[3] Amdahl, G., *Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities*, AFIPS Conference Proceedings, Spring Joint Computer Conference, April 1967, Atlantic City, NJ, pp.483–485.

[4] Fortier, P.J., and Michel, H.E., *Computer Systems Performance Evaluation and Prediction*, Elsevier Science, 2003.

[5] Gunther, N.J., *The Practical Performance Analyst*, Authors Choice Press, 2000.

[6] Jain, R., *The Art of Computer Systems Performance Analysis - Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, 1991.

[7] Leung, C.H.C., *Quantitative Analysis of Computer Systems*, John Wiley & Sons, 1988.

[8] Lilja, D.J., *Measuring Computer Performance: A Practitioner's Guide*, Cambridge University Press, 2000.

[9] Menasce, D.A., and Almeida, V.A.F., *Capacity Planning for Web Services: Metrics, Models and Methods*, Prentice-Hall, 2002.

[10] Menasce, D.A., Almeida, V.A.F., and Dowdy, L.W., *Performance by Design, Computer Capacity Planning by Example*, Prentice-Hall PTR, 2004.

[11] Obaidat, M.S., and Boudriga, N.A., *Performance Evaluation of Computer and Telecommunication Systems*, Wiley, 2010.

Κεφάλαιο 9

Σχεδίαση και Ανάλυση Πειραμάτων

Σύνοψη

Παρουσιάζονται οι βασικές αρχές σχεδίασης και ανάλυσης πειραματικών μετρήσεων. Κατ΄ αρχάς εξετάζονται θέματα ακρίβειας και πηγών σφάλματος, μοντέλα σφαλμάτων, σύγκριση εναλλακτικών λύσεων, διαστήματα εμπιστοσύνης, μοντέλα γραμμικής και μη γραμμικής παλινδρόμησης, ανάλυση διασποράς. Εν συνεχεία παρουσιάζονται οι κυριότεροι τύποι πειραμάτων και αναλύονται τα αντίστοιχα μοντέλα παλινδρόμησης με υπολογισμό των επιδράσεων και αλληλεπιδράσεων των παραγόντων και της κατανομής της μεταβλητότητας. Περιλαμβάνονται πειράματα ενός, δύο ή περισσότερων παραγόντων, πλήρη παραγοντικά πειράματα, κλασματικά παραγοντικά πειράματα και σύγχυση επιδράσεων, πειράματα με δύο στάθμες ανά παράγοντα (2^m), η μέθοδος του πίνακα προσήμων, πειράματα με επαναλήψεις, εκτίμηση πειραματικών σφαλμάτων.

Η μελέτη της επίδοσης ενός συστήματος με τη βοήθεια μετρήσεων στοχεύει στην κατανόηση της συμπεριφοράς του συστήματος αξιοποιώντας πληροφορίες από τον πραγματικό κόσμο. Το πρόβλημα είναι ότι παράγοντες του πραγματικού κόσμου εισάγουν αβεβαιότητα στην πληροφορία των μετρήσεων. Η αβεβαιότητα αυτή, που αναφέρεται ως *σφάλμα* ή *θόρυβος*, πρέπει να προσδιοριστεί ποσοτικά, προκειμένου να γνωρίζουμε πόσο αξιόπιστα είναι τα συμπεράσματά μας.

Γενικά, η αποτελεσματικότητα των μετρήσεων εξαρτάται από την επιλογή των κατάλληλων συνθηκών εισόδου (φορτίων), την επιτυχή οργάνωση των πειραμάτων και τη σωστή ανάλυση και ερμηνεία των αποτελεσμάτων [6, 4, 3, 2, 7, 8, 1, 9].

9.1 Σφάλματα

Η ποιότητα των μετρήσεων εκφράζεται συνήθως μέσω τριών χαρακτηριστικών [5]:

- Η *ορθότητα* (accuracy) είναι η απόλυτη διαφορά ανάμεσα στην τιμή μιας μέτρησης (ή στη μέση τιμή πολλών μετρήσεων) και στην (άγνωστη) «αληθινή» τιμή του μεγέθους που επιθυμούμε να μετρήσουμε.

- Η *ακρίβεια* (precision) εκφράζει την επαναληψιμότητα των μετρήσεων. Μπορεί να περιγραφεί ποσοτικά με χρήση της διασποράς των μετρήσεων, όταν πραγματοποιούμε πολλαπλές μετρήσεις ενός μεγέθους. Είναι φανερό ότι μπορεί να σημειωθεί χαμηλή ορθότητα σε μετρήσεις υψηλής ακρίβειας (χαμηλής διασποράς) και αντίστροφα.

- Η *ανάλυση* (resolution) είναι η ελάχιστη διαφορά τιμών που μπορεί να διαπιστωθεί, σύμφωνα με τη μέθοδο ή το εργαλείο μέτρησης.

Όταν λαμβάνουμε το αποτέλεσμα ενός συνόλου μετρήσεων, δεν είναι εύκολο να εξειδικεύσουμε τη συνεισφορά καθενός από τα παραπάνω τρία χαρακτηριστικά στη διαμόρφωση του σφάλματος της μέτρησης. Η ποσοτικοποίηση της ορθότητας είναι σχετικά δύσκολη. Αυτό που γίνεται συνήθως είναι ο προσδιορισμός ενός *διαστήματος εμπιστοσύνης* (confidence interval) για τη μέση τιμή των μετρήσεων.

Η ορθότητα, η ακρίβεια και η ανάλυση είναι χαρακτηριστικά της μεθόδου ή του εργαλείου μέτρησης. Πέραν αυτών, όμως, πολλές άλλες πηγές σφάλματος επηρεάζουν τη διαδικασία της μέτρησης και τα τελικά αποτελέσματα. Ως παράδειγμα, μπορούμε να αναφέρουμε διάφορες επιβαρύνσεις που οφείλονται στο ίδιο το υλικό ή/και το λογισμικό μέτρησης, επεξεργασία γεγονότων που προέρχονται από διεπαφή με άλλα συστήματα και αλληλεπίδραση με τους χρήστες κλπ. Επίσης, διαταραχές μπορεί να προκληθούν από άλλα μη-ντετερμινιστικά γεγονότα, όπως αστοχία λανθάνουσας μνήμης, έγερση εξαιρέσεων κλπ. Αποτέλεσμα είναι ότι οι τιμές των μετρήσεων μπορεί να διαφοροποιηθούν από μέτρηση σε μέτρηση. Στην πραγματικότητα, όλες οι μετρήσεις αποτελούν εκτιμήσεις της αληθινής τιμής που μετράμε.

Μπορούμε να κατατάξουμε τα σφάλματα σε δύο κατηγορίες: *συστηματικά σφάλματα* και *τυχαία σφάλματα*, ανάλογα με την πηγή τους. Τα συστηματικά σφάλματα οφείλονται σε κάποιο λάθος εκτέλεσης ή σε κάποια διαδικασία που πολώνει τις μετρήσεις. Η συνεισφορά τους είναι συνήθως σταθερή και επηρεάζει την ορθότητα των μετρήσεων. Αντίθετα, τα τυχαία σφάλματα είναι απρόβλεπτα και δεν πολώνουν τις μετρήσεις προς μια κατεύθυνση. Μπορεί να οφείλονται στη διαδικασία της μέτρησης ή σε άλλες τυχαίες διεργασίες στο εσωτερικό του συστήματος. Όπως είναι φανερό, επηρεάζουν την ακρίβεια των μετρήσεων.

Με την κατάλληλη σχεδίαση των πειραμάτων είναι συχνά εφικτή η μείωση της επίδρασης των συστηματικών σφαλμάτων. Εξάλλου, σημαντικό ρόλο παίζει η κατανόηση του μηχανισμού δημιουργίας αυτών των σφαλμάτων και της πόλωσης που προκαλούν στα αποτελέσματα.Όσον αφορά τα τυχαία σφάλματα, εφόσον δεν είναι εφικτός ο ακριβής προσδιορισμός τους, μπορούν να περιγραφούν μέσω στατιστικών μοντέλων. Η μοντελοποίηση αυτή επιτρέπει τη χρήση στατιστικών μεθόδων για την ποσοτικοποίηση της ακρίβειας.

Γενικά, υποθέτουμε ότι τα τυχαία πειραματικά σφάλματα ακολουθούν κανονική κατανομή. Συνεπώς, εάν πραγματοποιηθούν πολλαπλές μετρήσεις της ίδιας τιμής, οι μετρήσεις αυτές θα είναι κατανεμημένες σύμφωνα με κανονική κατανομή που θα έχει ως κέντρο την πραγματική μέση τιμή των μετρήσεων. Η τεχνική των διαστημάτων εμπιστοσύνης βασίζεται στο μοντέλο της κανονικής κατανομής για τα τυχαία σφάλματα .

9.2 Διαστήματα Εμπιστοσύνης

Τα χαρακτηριστικά των υπό μελέτη συστημάτων δεν μπορούν συνήθως να προσδιοριστούν με ντετερμινιστικό τρόπο. Αντίθετα, μπορεί να βρεθεί με πιθανοτικό τρόπο μια περιοχή τιμών μέσα στην οποία βρίσκεται η αληθινή τιμή του χαρακτηριστικού. Τα διαστήματα εμπιστοσύνης αποτελούν βασικό εργαλείο στη διαδικασία της ανάλυσης επίδοσης [5, 3].

9.2.1 Διάστημα Εμπιστοσύνης Μέσης Τιμής

Αν έχουμε ένα δείγμα $\{x_1, x_2, \ldots, x_n\}$ ενός πληθυσμού μπορούμε να πάρουμε μία εκτίμηση της μέσης τιμής μ του πληθυσμού από τη μέση τιμή του δείγματος:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{9.1}$$

Για την τυχαία μεταβλητή $\overline{x}$ μπορούμε να προσδιορίσουμε ένα διάστημα εμπιστοσύνης (c_1, c_2) που θα περιέχει την άγνωστη τιμή μ με καθορισμένη πιθανότητα:

$$Pr[c_1 \leq \mu \leq c_2] = 1 - \alpha$$

Η πιθανότητα $1 - \alpha$ (συνήθως εκφρασμένη ως ποσοστό) είναι ο βαθμός εμπιστοσύνης.

Σύμφωνα με γνωστά θεωρήματα, το διάστημα εμπιστοσύνης της μέσης τιμής θα είναι για σχετικά μεγάλο δείγμα (πρακτικά για $n \geq 30$):

$$(\overline{x} - z_{1-\alpha/2}s/\sqrt{n}, \overline{x} + z_{1-\alpha/2}s/\sqrt{n}) \tag{9.2}$$

όπου s^2 είναι η διασπορά του δείγματος

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 \tag{9.3}$$

και $z_{1-\alpha/2}$ είναι το $(1-\alpha/2)$ - ποσοστιαίο σημείο της μοναδιαίας κανονικής κατανομής. Από τους πίνακες βρίσκουμε ότι για βαθμό εμπιστοσύνης 90%, 95% και 99%, οι τιμές του $z_{1-\alpha/2}$ είναι αντίστοιχα 1,645, 1,96 και 2,576.

Το παραπάνω διάστημα εμπιστοσύνης ισχύει για σχετικά μεγάλο δείγμα. Για μικρότερο δείγμα, και με την προϋπόθεση ότι το δείγμα προέρχεται από κανονικό πληθυσμό, το διάστημα εμπιστοσύνης θα είναι:

$$(\overline{x} - t_{[n-1,1-\alpha/2]}s/\sqrt{n}, \overline{x} + t_{[n-1,1-\alpha/2]}s/\sqrt{n}) \tag{9.4}$$

όπου $t_{[n-1,1-\alpha/2]}$ είναι το $(1-\alpha/2)$ - ποσοστιαίο σημείο της κατανομής t του Student με $n-1$ βαθμούς ελευθερίας και βρίσκεται εύκολα από τους σχετικούς πίνακες.

Μία συνηθισμένη εφαρμογή των διαστημάτων εμπιστοσύνης είναι ο έλεγχος της υπόθεσης ότι η μέση τιμή είναι σημαντικά διάφορη του μηδενός. Αν το μηδέν δεν περιλαμβάνεται στο διάστημα εμπιστοσύνης, τότε το αποτέλεσμα του ελέγχου είναι θετικό. Διαφορετικά είναι αρνητικό, για το δεδομένο βαθμό εμπιστοσύνης. Η ίδια διαδικασία εφαρμόζεται αν αντί για το μηδέν θεωρήσουμε οποιαδήποτε άλλη τιμή. Γενικά, η χρήση διαστημάτων εμπιστοσύνης για τον έλεγχο υποθέσεων αποτελεί μία αποτελεσματική προσέγγιση γιατί δεν υποδεικνύει απλά την αποδοχή ή την απόρριψη της υπόθεσης, αλλά παρέχει πληροφορίες και για το εύρος τιμών της παραμέτρου.

9.2.2 Διαστήματα Εμπιστοσύνης για Πιθανότητες

Για κατηγορικές μεταβλητές, τα δεδομένα έχουν συχνά τη μορφή πιθανοτήτων (ποσοστών) που σχετίζονται με τις τιμές της μεταβλητής. Αν k από τις n παρατηρήσεις ενός δείγματος αντιστοιχούν σε κάποια δεδομένη τιμή της κατηγορικής μεταβλητής, η εκτίμηση της πιθανότητας της τιμής αυτής θα είναι:

$$p = \frac{k}{n} \tag{9.5}$$

Οι μετρήσεις πιθανοτήτων ακολουθούν δυωνυμική κατανομή. Από τα χαρακτηριστικά της διωνυμικής κατανομής βρίσκουμε ότι η εκτίμηση της πιθανότητας είναι p με διασπορά $p(1-p)/n$. Το διάστημα εμπιστοσύνης για την πιθανότητα θα είναι:

$$\left(p - z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n}}, p + z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n}} \right) \tag{9.6}$$

Η παραπάνω σχέση βασίζεται στην προσέγγιση της διωνυμικής κατανομής με κανονική. Η προσέγγιση αυτή ισχύει για $np \geq 10$.

9.2.3 Σύγκριση Εναλλακτικών

9.2.3.1 Διάστημα Εμπιστοσύνης Διαφοράς Μέσων Τιμών

Το διάστημα εμπιστοσύνης της διαφοράς των μέσων τιμών μ_1 και μ_2 δύο πληθυσμών επιτρέπει τη σύγκριση μεταξύ εναλλακτικών επιλογών. Διακρίνουμε δύο περιπτώσεις ανάλογα με το αν οι τιμές των δειγμάτων εμφανίζονται κατά ζεύγη ή όχι.

Μετρήσεις κατά ζεύγη Αν εκτελέσουμε n πειράματα σε καθένα από δύο συστήματα, έτσι ώστε να υπάρχει αμφιμονοσήμαντη αντιστοιχία μεταξύ των αποτελεσμάτων των πειραμάτων στα δύο συστήματα, τότε έχουμε δύο δείγματα των οποίων οι τιμές σχετίζονται κατά ζεύγη. Η ανάλυση στην περίπτωση αυτή μπορεί να γίνει ακριβώς όπως προηγουμένως, αν λάβουμε τις διαφορές των αντίστοιχων τιμών των δύο δειγμάτων και θεωρήσουμε ότι αποτελούν ένα μοναδικό δείγμα. Αν το διάστημα εμπιστοσύνης που προκύπτει για τη μέση τιμή της διαφοράς περιλαμβάνει το μηδέν, τότε τα δύο συστήματα δεν διαφέρουν σημαντικά.

Μη αντιστοιχούσες μετρήσεις Αν οι παρατηρήσεις δεν εμφανίζονται κατά ζεύγη, έχουμε δύο ανεξάρτητα δείγματα με μεγέθη n_1 και n_2 που αφορούν τις δύο υπό μελέτη εναλλακτικές επιλογές. Αν x_1, x_2 και s_1, s_2 είναι οι μέσες τιμές και οι τυπικές αποκλίσεις των δειγμάτων αντίστοιχα, το διάστημα εμπιστοσύνης για τη διαφορά $\mu_1 - \mu_2$ θα είναι:

$$(\overline{x}_1 - \overline{x}_2 - t_{[\nu,1-\alpha/2]}s, \overline{x}_1 - \overline{x}_2 + t_{[\nu,1-\alpha/2]}s) \tag{9.7}$$

όπου

$$s = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \tag{9.8}$$

και οι βαθμοί ελευθερίας δίνονται από τη σχέση

$$\nu = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{\frac{1}{n_1 - 1}\left(\frac{s_1^2}{n_1}\right)^2 + \frac{1}{n_2 - 1}\left(\frac{s_2^2}{n_2}\right)^2} \tag{9.9}$$

9.2.3.2 Διάστημα Εμπιστοσύνης Διαφοράς Πιθανοτήτων

Προκειμένου να συγκρίνουμε δύο εναλλακτικές που εκφράζονται μέσω πιθανοτήτων, θεωρούμε τις εκτιμήσεις $p_1 = \frac{k_1}{n_1}$ και $p_2 = \frac{k_2}{n_2}$ σε δύο συστήματα, αντίστοιχα, όπου n_1, n_2 ο συνολικός αριθμός μετρήσεων και k_1, k_2 το πλήθος μετρήσεων που αντιστοιχούν στην τιμή ή στο γεγονός που μας ενδιαφέρει. Όπως παραπάνω, αν ισχύει $n_1 p_1 \geq 10$ και $n_2 p_2 \geq 10$, μπορούμε να προσεγγίσουμε τις κατανομές των πιθανοτήτων p_1, p_2 χρησιμοποιώντας κανονικές κατανομές με μέση τιμή p_1, p_2 και διασπορά $p_1(1-p_1)/n_1$, $p_2(1-p_2)/n_2$, αντίστοιχα. Σε αναλογία με τον υπολογισμό του διαστήματος εμπιστοσύνης της διαφοράς μέσων τιμών για μη αντιστοιχούσες μετρήσεις, βρίσκουμε το διάστημα εμπιστοσύνης:

$$(p - z_{1-\alpha/2}s, p + z_{1-\alpha/2}s) \tag{9.10}$$

όπου

$$p = p_1 - p_2 \tag{9.11}$$

και

$$s = \sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_b}} \tag{9.12}$$

9.2.4 Προσδιορισμός του Μεγέθους του Δείγματος

Το εύρος του διαστήματος εμπιστοσύνης προσδιορίζει την ακρίβεια της εκτίμησης. Για δεδομένο βαθμό εμπιστοσύνης, η ακρίβεια αυξάνεται όσο αυξάνεται το μέγεθος του δείγματος. Το συνηθισμένο πρόβλημα είναι να βρεθεί το ελάχιστο μέγεθος δείγματος που εξασφαλίζει έναν επιθυμητό βαθμό εμπιστοσύνης και ακρίβειας.

9.2.4.1 Εκτίμηση Μέσης Τιμής.

Έστω ότι θέλουμε να εκτιμήσουμε τη μέση τιμή ενός δείκτη επίδοσης με ακρίβεια $\pm r\%$ και βαθμό εμπιστοσύνης $100(1-\alpha)\%$. Για δείγμα μεγέθους n το διάστημα εμπιστοσύνης θα είναι $\overline{x} \mp zs/\sqrt{n}$. Σύμφωνα με την απαίτηση ακρίβειας το διάστημα εμπιστοσύνης θα πρέπει να είναι $(\overline{x}(1 - r/100), \overline{x}(1 + r/100))$. Εξισώνοντας τις δύο εκφράσεις έχουμε:

$$\overline{x} \mp zs/\sqrt{n} = \overline{x}(1 \mp r/100)$$

απ' όπου βρίσκουμε το απαιτούμενο μέγεθος δείγματος:

$$n = \left(\frac{100zs}{r\overline{x}}\right)^2 \tag{9.13}$$

Παράδειγμα 9.1. *Με βάση μία προκαταρκτική μέτρηση, η μέση τιμή του δείγματος για το χρόνο εγγραφής ενός αρχείου συγκεκριμένου μεγέθους στον δίσκο είναι* 12 sec *και η τυπική απόκλιση του δείγματος* $2,8$. *Ζητείται ο απαιτούμενος αριθμός επαναλήψεων για ακρίβεια* $0,3$ sec *και βαθμό εμπιστοσύνης* 90%.

Έχουμε $r = 0,3/12 = 2,5\%$, $\overline{x} = 12$, $s = 2,8$ και $z = 1,645$. Άρα

$$n = \left(\frac{100 \times 1,645 \times 2,8}{2,5 \times 12}\right)^2 = 235,72$$

δηλαδή, απαιτούνται τουλάχιστον 236 παρατηρήσεις. □

9.2.4.2 Εκτίμηση Πιθανότητας.

Όπως και προηγουμένως, θα έχουμε για ακρίβεια $\pm r\%$ και βαθμό εμπιστοσύνης $100(1-\alpha)\%$:

$$p \mp r = p \mp z\sqrt{\frac{p(1-p)}{n}}$$

απ' όπου βρίσκουμε το απαιτούμενο μέγεθος δείγματος:

$$n = z^2\frac{p(1-p)}{r^2} \tag{9.14}$$

Παράδειγμα 9.2. *Μία προκαταρκτική μέτρηση έδειξε ότι η πιθανότητα σφάλματος εκτύπωσης για έναν εκτυπωτή είναι* 1 *σελίδα στις* 8 000. *Ζητείται ο αριθμός σελίδων που πρέπει να παρατηρηθούν, ώστε η πιθανότητα σφάλματος να εκτιμηθεί με ακρίβεια* 1 *στις* 500 000 *και βαθμό εμπιστοσύνης* 90%.

Έχουμε $r = 1/500\,000 = 2 \times 10^{-6}$, $p = 1/8\,000 = 1,25 \times 10^{-4}$ και $z = 1,645$, οπότε βρίσκουμε ότι ο απαιτούμενος αριθμός σελίδων είναι 84 552 711. □

9.3 Μοντέλα Παλινδρόμησης

Τα μοντέλα *παλινδρόμησης* (regression) επιτρέπουν την εκτίμηση ή πρόβλεψη μιας (εξαρτημένης) τυχαίας μεταβλητής (*μεταβλητή απόκρισης*) συναρτήσει άλλων ανεξάρτητων μεταβλητών (*μεταβλητές πρόβλεψης ή παράγοντες*). Οι βασικές αρχές των μοντέλων παλινδρόμησης έχουν άμεση εφαρμογή στην ανάλυση πειραματικών δεδομένων. Με βάση ένα πεπερασμένο σύνολο μετρήσεων κατασκευάζεται ένα μαθηματικό μοντέλο το οποίο περιγράφει την απόκριση του συστήματος σε μια περιοχή τιμών εισόδου. Πέραν της εκτίμησης της μεταβλητής απόκρισης, το μοντέλο επιτρέπει την εκτίμηση της *βαρύτητας* των παραγόντων μέσω της συμμετοχής τους στη συνολική μεταβλητότητα των μετρήσεων. Μια γενικότερη τεχνική προσδιορισμού της σημαντικότητας των παραγόντων έναι η *Ανάλυση Διασποράς*.

9.3.1 Απλή Γραμμική Παλινδρόμηση

Η πλέον συνήθης χρήση της παλινδρόμησης αφορά γραμμικά μοντέλα. Η εκτίμηση γίνεται με την απαίτηση ικανοποίησης του κριτηρίου των ελαχίστων τετραγώνων. Θα εστιάσουμε στην *απλή γραμμική παλινδρόμηση*, η οποία γενικεύεται σχετικά εύκολα σε πιο σύνθετα μοντέλα, όπως η πολλαπλή γραμμική παλινδρόμηση, η πολυωνυμική παλινδρόμηση, η παλινδρόμηση με κατηγορικές μεταβλητές πρόβλεψης ή οι διάφορες περιπτώσεις καμπυλόγραμμης παλινδρόμησης.

Θεωρούμε γραμμικό μοντέλο με μία μεταβλητή πρόβλεψης:

$$\tilde{y} = a + bx \tag{9.15}$$

όπου $\tilde{y}$ είναι η προβλεπόμενη απόκριση όταν η μεταβλητή πρόβλεψης είναι x. Οι παράμετροι a και b προσδιορίζονται από τις παρατηρήσεις $\{(x_1, y_1), \ldots, (x_n, y_n)\}$. Το σφάλμα της πρόβλεψης $\tilde{y}_i = a + bx_i$ θα είναι $e_i = y_i - \tilde{y}_i$, $i = 1, \ldots, n$.

Σύμφωνα με τις απαιτήσεις του μοντέλου θα πρέπει να ελαχιστοποιείται το άθροισμα των τετραγώνων των σφαλμάτων (Sum of Squared Errors):

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2 \tag{9.16}$$

Θέτοντας τις παραγώγους του SSE ως προς a και b ίσες με το 0 λαμβάνουμε το σύστημα

$$na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \tag{9.17}$$

$$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i \tag{9.18}$$

Η εξίσωση (9.17) δηλώνει ότι το ολικό σφάλμα ισούται με 0. Από την επίλυση του συστήματος προκύπτουν οι τιμές των παραμέτρων:

$$b = \frac{\sum_{i=1}^{n} x_i y_i - n\overline{x}\,\overline{y}}{\sum_{i=1}^{n} x_i^2 - n\overline{x}^2} \tag{9.19}$$

$$a = \overline{y} - b\overline{x} \tag{9.20}$$

Η ανάπτυξη του μοντέλου στηρίζεται στις ακόλουθες παραδοχές:

- Υπάρχει γραμμική σχέση ανάμεσα στην απόκριση y και στη μεταβλητή πρόβλεψης x.
- Η μεταβλητή x δεν είναι στοχαστική και μετριέται χωρίς σφάλμα.
- Τα σφάλματα του μοντέλου είναι ανεξάρτητες και ισόνομες τυχαίες μεταβλητές με κατανομή $N(0, \sigma^2)$.

Κατανομή της μεταβλητότητας. Χωρίς το μοντέλο παλινδρόμησης, θα μπορούσε να χρησιμοποιηθεί απευθείας η μέση τιμή του y ως προβλεπόμενη απόκριση για όλες τις τιμές του x. Στην περίπτωση αυτή το άθροισμα των τετραγώνων των σφαλμάτων χωρίς παλινδρόμηση θα είναι:

$$SST = \sum_{i=1}^{n} (y_i - \overline{y})^2 \tag{9.21}$$

Το άθροισμα SST (Total Sum of Squares) εκφράζει τη *μεταβλητότητα* (variation) του y και μπορεί να γραφτεί:

$$SST = SSR + SSE \tag{9.22}$$

όπου SSR είναι το *επεξηγούμενο* από την παλινδρόμηση άθροισμα τετραγώνων και SSE το άθροισμα των τετραγώνων των σφαλμάτων με χρήση παλινδρόμησης. Το κλάσμα της μεταβλητότητας που επεξηγείται από την παλινδρόμηση είναι ο *συντελεστής προσαρμογής* (coefficient of determination) και χαρακτηρίζει την καταλληλότητα της παλινδρόμησης:

$$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST} \tag{9.23}$$

Παράδειγμα 9.3. *Τα ακόλουθα ζεύγη μετρήσεων (x_i, y_i) αφορούν τον απαιτούμενο χρόνο y ($\sigma\epsilon$ msec) για την εγγραφή αρχείων διαφόρων μεγεθών x ($\sigma\epsilon$ kbytes) στον δίσκο:*
$\{(17,\ 2,4), (48,\ 7,5), (207,\ 31,3), (443,\ 68,5), (739,\ 99,2), (980,\ 134,6)\}$.

Κατασκευάζουμε γραμμικό μοντέλο για την πρόβλεψη του χρόνου KME συναρτήσει του αριθμού προσπελάσεων στο δίσκο. Έχουμε $n = 6$, $\sum xy = 242\,442,2$, $\sum x^2 = 1\,748\,212$, $\overline{x} = 405,67$, $\overline{y} = 57,25$, απ' όπου βρίσκουμε $b = 0,1355$ και $a = 2,2817$. Άρα το ζητούμενο γραμμικό μοντέλο είναι:

$$y = 2,2817 + 0,1355x$$

Όσον αφορά την κατανομή της μεταβλητότητας βρίσκουμε: $SSE = 56,27$, $SST = 14\,026,38$, οπότε $SSR = 13\,970,11$ και $R^2 = 0,996$, δηλαδή η παλινδρόμηση επεξηγεί το 99,6% της μεταβλητότητας του χρόνου. □

Άλλα μοντέλα παλινδρόμησης. Η απλή γραμμική παλινδρόμηση υπόκειται σε περιορισμούς οι οποίοι δεν ικανοποιούνται πάντα στην πράξη. Σε τέτοιες περιπτώσεις, η τεχνική της παλινδρόμησης μπορεί να εφαρμοστεί με ανάλογες τροποποιήσεις.

Πολλαπλή γραμμική παλινδρόμηση Πρόκειται για άμεση γενίκευση της απλής γραμμικής παλινδρόμησης. Η μεταβλητή απόκρισης είναι γραμμική συνάρτηση πολλών (ανεξάρτητων) μεταβλητών εισόδου:

$$\tilde{y} = a + b_1 x_1 + b_2 x_2 + \ldots + b_k x_k \tag{9.24}$$

Όπως στο απλό μοντέλο, ελαχιστοποιείται το άθροισμα των τετραγώνων των σφαλμάτων. Από το γραμμικό σύστημα που προκύπτει, υπολογίζονται οι τιμές των παραμέτρων.

Μη γραμμική παλινδρόμηση Όταν η σχέση εισόδου-εξόδου είναι μη γραμμική, υπάρχει συχνά η δυνατότητα μετασχηματισμού σε γραμμική μορφή και εν συνεχεία εφαρμογής γραμμικού μοντέλου. Αν θεωρήσουμε, για παράδειγμα, την εκθετική σχέση

$$y = ax^b \tag{9.25}$$

μπορούμε να εφαρμόσουμε λογαριθμικό μετασχηματισμό:

$$\ln y = \ln a + b \ln x \tag{9.26}$$

Θέτοντας $y' = \ln y$ και $x' = \ln x$, καταλήγουμε σε γραμμικό μοντέλο. Η τεχνική της μετατροπής μιας μη γραμμικής σχέσης σε γραμμική και εν συνεχεία εφαρμογής γραμμικού μοντέλου αναφέρεται ως *καμπυλόγραμμη παλινδρόμηση* (curvilinear regression) και μπορεί να συνδυαστεί με ποικιλία μετασχηματισμών.

Κατηγορικές μεταβλητές Οι τεχνικές παλινδρόμησης μπορούν να χρησιμοποιηθούν και στην περίπτωση που υπάρχουν μία ή περισσότερες κατηγορικές (μη αριθμητικές) μεταβλητές. Ιδιαίτερα αποδοτική είναι η διαχείριση μεταβλητών που μπορούν να λάβουν δύο τιμές ή στάθμες. Μια τέτοια μεταβλητή A μπορεί να παρασταθεί με τη βοήθεια μιας δυαδικής μεταβλητής x_A με τιμές -1 και 1, ως εξής:

$$x_A = \begin{cases} -1 & \text{η μία τιμή της μεταβλητής } A \\ 1 & \text{η άλλη τιμή της μεταβλητής } A \end{cases} \tag{9.27}$$

Η χρήση της δυαδικής παράστασης κατηγορικών μεταβλητών θα εξεταστεί παρακάτω στο πλαίσιο της ανάλυσης πειραμάτων.

9.3.2 Ανάλυση Διασποράς

Η κατανομή της μεταβλητότητας, σύμφωνα με την οποία η μεταβλητότητα διαμερίζεται σε επεξηγούμενο και μη επεξηγούμενο μέρος, είναι μια απλή και διαισθητικά αποδεκτή τεχνική για την εκτίμηση της βαρύτητας παραγόντων και σφαλμάτων στην απόκριση του συστήματος. Στην ίδια κατεύθυνση, μια γενικότερη στατιστική μέθοδος είναι η *Ανάλυση Διασποράς* (Analysis of Variance — ANOVA), η οποία υπολογίζει τη σημαντικότητα της συνεισφοράς κάθε παράγοντα στη μεταβλητότητα.

Κάθε άθροισμα τετραγώνων συνδέεται με έναν αριθμό *βαθμών ελευθερίας* (degrees of freedom), ο οποίος αντιπροσωπεύει τις ανεξάρτητες τιμές που απαιτούνται για τον υπολογισμό του συγκεκριμένου αθροίσματος τετραγώνων. Π.χ., το SST έχει $n-1$ βαθμούς ελευθερίας, καθόσον προηγείται ο υπολογισμός της παραμέτρου $\overline{y}$, όπως προκύπτει από την εξίσωση (9.21). Αντίστοιχα, το SSE έχει $n-2$ βαθμούς ελευθερίας, εφόσον τα σφάλματα υπολογίζονται μετά τον προσδιορισμό δύο παραμέτρων από τα δεδομένα. Το SSR, που είναι η διαφορά των SST και SSE, έχει τον απομένοντα έναν βαθμό ελευθερίας. Ξαναγράφοντας την εξίσωση (9.22) διαπιστώνουμε ότι οι βαθμοί ελευθερίας αθροίζονται όπως ακριβώς τα αθροίσματα των τετραγώνων:

$$\begin{array}{rcccc} SST & = & SSR & + & SSE \\ n-1 & = & 1 & + & (n-2) \end{array}$$

Με την υπόθεση ότι τα σφάλματα είναι ανεξάρτητες και ισόνομες τυχαίες μεταβλητές που ακολουθούν κανονική κατανομή, και ότι οι είσοδοι x μετριούνται χωρίς σφάλμα, αποδεικνύεται ότι τα αθροίσματα τετραγώνων ακολουθούν κατανομές χ^2 με τους αντίστοιχους βαθμούς ελευθερίας. Αν θεωρήσουμε δύο αθροίσματα τετραγώνων SSi και SSk με ν_i και ν_k βαθμούς ελευθερίας αντίστοιχα, ο λόγος $(SSi/\nu_i)/(SSk/\nu_k)$ ακολουθεί κατανομή F με ν_i βαθμούς ελευθερίας στον αριθμητή και ν_k βαθμούς ελευθερίας στον παρονομαστή.

Η διαδικασία στατιστικού ελέγχου που προκύπτει ονομάζεται F-test και ελέγχει κατά πόσο δύο διασπορές είναι σημαντικά διάφορες μεταξύ τους. Η μεταβλητότητα (άθροισμα τετραγώνων) μιας συνιστώσας διαιρούμενη με τον αντίστοιχο αριθμό βαθμών ελευθερίας ονομάζεται *μέσο τετράγωνο* (mean square) της συνιστώσας και αποτελεί εκτίμηση της αντίστοιχης διασποράς. Συνοψίζοντας, ο λόγος F των μέσων τετραγώνων συγκρίνεται με την τιμή $F_{[1-\alpha;\nu_i,\nu_k]}$ από τους πίνακες της κατανομής F. Αν η τιμή που υπολογίστηκε είναι μεγαλύτερη από την κρίσιμη τιμή που προέρχεται από τους πίνακες, μπορούμε να πούμε ότι η μεταβλητότητα της συνιστώσας του αριθμητή είναι σημαντικά υψηλότερη από τη μεταβλητότητα της συνιστώσας του παρονομαστή, με επίπεδο σημαντικότητας α ή, ισοδύναμα, επίπεδο εμπιστοσύνης $1-\alpha$.

Επανερχόμενοι στην περίπτωση της απλής γραμμικής παλινδρόμησης, υπολογίζουμε τον λόγο

$$F = \frac{s_r^2}{s_e^2} = \frac{SSR/1}{SSE/(n-2)}$$

όπου s_r^2 και s_e^2 οι διασπορές (μέσα τετράγωνα) για την παλινδρόμηση και τα σφάλματα του μοντέλου, αντίστοιχα. Αν ισχύει $F > F_{[1-\alpha;1,n-2]}$, συμπεραίνουμε ότι το μοντέλο παλινδρόμησης επεξηγεί σημαντικά μεγαλύτερο μέρος της μεταβλητότητας σε σχέση με το μέρος της μεταβλητότητας που οφείλεται στα σφάλματα.

Η τεχνική ANOVA έχει άμεση εφαρμογή στην ανάλυση πειραμάτων, όπως θα δούμε στη συνέχεια.

9.4 Πειράματα

Η σχεδίαση πειραμάτων έχει ως στόχο την εξαγωγή της μέγιστης δυνατής πληροφορίας με τον ελάχιστο αριθμό πειραμάτων. Η ανάλυση των πειραμάτων επιτρέπει τον προσδιορισμό των επιδράσεων των διαφόρων παραγόντων που επηρεάζουν την επίδοση του συστήματος, καθώς και των αλληλεπιδράσεων των παραγόντων και της επίδρασης των πειραματικών σφαλμάτων [3, 5].

Το εξαγόμενο ενός πειράματος (η μετρούμενη επίδοση) είναι η *μεταβλητή απόκρισης* του πειράματος. Οι μεταβλητές που επηρεάζουν την απόκριση είναι οι *παράγοντες* του πειράματος και οι τιμές που μπορεί να λάβει ένας παράγων είναι οι *στάθμες* του παράγοντα. Οι παράγοντες των οποίων η επίδραση μελετάται ποσοτικά μέσω του πειράματος ονομάζονται πρωτεύοντες παράγοντες, ενώ οι υπόλοιποι ονομάζονται δευτερεύοντες. Η σχεδίαση αφορά τον καθορισμό του αριθμού των πειραμάτων, τον συνδυασμό των σταθμών των παραγόντων για κάθε πείραμα και τον αριθμό των επαναλήψεων για κάθε πείραμα.

Διακρίνουμε τρεις γενικές μεθόδους σχεδίασης πειραμάτων:

(i) *Απλή σχεδίαση.* Αρχίζουμε από έναν τυπικό συνδυασμό των σταθμών των παραγόντων, τον οποίο χρησιμοποιούμε ως βάση. Εν συνεχεία μεταβάλλουμε σε κάθε πείραμα τη στάθμη ενός παράγοντα και βλέπουμε πώς η μεταβολή αυτή επηρεάζει την απόκριση σε σχέση με τη βασική διάταξη. Αν έχουμε k παράγοντες και ο παράγων i έχει n_i στάθμες, ο απαιτούμενος αριθμός πειραμάτων θα είναι:

$$n = 1 + \sum_{i=1}^{k}(n_i - 1) \tag{9.28}$$

Η σχεδίαση αυτή είναι στατιστικά ανεπαρκής και δεν λαμβάνει υπόψη τις αλληλεπιδράσεις των παραγόντων, συνεπώς η χρήση της είναι περιορισμένη.

(ii) *Πλήρης παραγοντική σχεδίαση.* Εκτελούνται πειράματα για κάθε δυνατό συνδυασμό όλων των σταθμών όλων των παραγόντων. Ο απαιτούμενος αριθμός πειραμάτων θα είναι:

$$n = \prod_{i=1}^{k} n_i \tag{9.29}$$

Με την πλήρη παραγοντική σχεδίαση μπορούν να μελετηθούν οι επιδράσεις όλων των παραγόντων καθώς και οι αλληλεπιδράσεις τους. Δεδομένου, όμως, ότι ο μεγάλος αριθμός πειραμάτων που προκύπτει συνεπάγεται αυξημένο κόστος, συνήθως αναζητούνται τρόποι μείωσης του αριθμού των πειραμάτων. Δύο δυνατές λύσεις είναι να μειωθεί ο αριθμός των παραγόντων ή να μειωθεί ο αριθμός των σταθμών ανά παράγοντα. Η δεύτερη λύση είναι ιδιαίτερα αποτελεσματική. Σε πολλές περιπτώσεις, είναι δυνατό να εξεταστούν μόνο δύο στάθμες ανά παράγοντα, ώστε να προσδιοριστεί η σχετική σημασία των παραγόντων. (Η σχεδίαση αυτή περιλαμβάνει 2^k πειράματα αν οι παράγοντες είναι k και αναφέρεται ως σχεδίαση 2^k). Εν συνεχεία, αν αφαιρεθούν κάποιοι λιγότερο σημαντικοί παράγοντες, θα μπορούσαν να εξεταστούν περισσότερες στάθμες ανά παράγοντα. Μία τρίτη λύση είναι η κλασματική παραγοντική σχεδίαση.

(iii) *Κ*λασματική παραγοντική σχεδίαση. Εκτελείται μόνο ένα κλάσμα από το σύνολο των δυνατών πειραμάτων της πλήρους παραγοντικής σχεδίασης. Μία τεχνική επιλογής του κατάλληλου κλάσματος πειραμάτων θα περιγραφεί στη συνέχεια. Η κλασματική παραγοντική σχεδίαση έχει μειωμένο κόστος, αλλά συνεπάγεται και απώλεια πληροφορίας σε σχέση με την πλήρη σχεδίαση.

9.5 Πλήρη Παραγοντικά Πειράματα με Έναν και Δύο Παράγοντες

Η ανάλυση που ακολουθεί βασίζεται στην υπόθεση ότι τα σφάλματα είναι ανεξάρτητα και ισόνομα με κανονική κατανομή, σε αναλογία με την ανάλυση της απλής γραμμικής παλινδρόμησης. Επίσης, ειδικότερα στην περίπτωση των δύο παραγόντων, υποτίθεται ότι οι επιδράσεις των παραγόντων, οι αλληλεπιδράσεις και τα σφάλματα σχετίζονται με προσθετικό τρόπο. Συχνά, η τελευταία υπόθεση δεν ισχύει, π.χ. όταν οι επιδράσεις των δύο παραγόντων στην απόκριση σχετίζονται με πολλαπλασιαστικό τρόπο. Σε τέτοιες περιπτώσεις, χρησιμοποιούμε λογαριθμικό μετασχηματισμό της απόκρισης, οπότε προκύπτει προσθετικό μοντέλο το οποίο αναλύεται κανονικά. Μετά την ανάλυση, ο αντίστροφος μετασχηματισμός των προσθετικών επιδράσεων δίνει τις πολλαπλασιαστικές επιδράσεις. Άλλα κριτήρια που υποδεικνύουν την ανάγκη χρήσης πολλαπλασιαστικού μοντέλου είναι η μεγάλη κύμανση των τιμών της απόκρισης, οι μεγάλες τιμές σφάλματος ή η παρέκκλιση από την υπόθεση της κανονικής κατανομής των σφαλμάτων.

9.5.1 Πειράματα με Έναν Παράγοντα

Η σχεδίαση αυτή έχει ως στόχο τη σύγκριση εναλλακτικών τιμών μιας κατηγορικής μεταβλητής χωρίς περιορισμό στον αριθμό των σταθμών. Υποθέτουμε ότι ο παράγων έχει a στάθμες και ότι πραγματοποιούνται r επαναλήψεις του πειράματος για κάθε στάθμη. Το μοντέλο έχει τη μορφή

$$y_{ij} = \mu + \alpha_i + e_{ij} \tag{9.30}$$

όπου y_{ij} είναι η j παρατήρηση με τον παράγοντα στη στάθμη i, μ είναι η *μέση απόκριση*, α_i είναι η *επίδραση* (effect) της στάθμης i και e_{ij} είναι το πειραματικό σφάλμα. Οι επιδράσεις α_i υπολογίζονται με την απαίτηση ότι το άθροισμά τους είναι 0:

$$\sum_{i=1}^{a} \alpha_i = 0$$

Τα σφάλματα έχουν άθροισμα 0 για κάθε πείραμα (στάθμη):

$$\sum_{j=1}^{r} e_{ij} = 0, \ \forall i$$

Σύμφωνα με τις υποθέσεις του μοντέλου, οι παράμετροι δίνονται από τις σχέσεις:

$$\mu = \overline{y}_{..} \tag{9.31}$$

$$\alpha_i = \overline{y}_{i.} - \overline{y}_{..} \tag{9.32}$$

όπου $\overline{y}_{..}$ είναι η συνολική μέση τιμή όλων των παρατηρήσεων:

$$\overline{y}_{..} = \frac{1}{ar}\sum_{i=1}^{a}\sum_{j=1}^{r} y_{ij} \tag{9.33}$$

και $\overline{y}_{i.}$ είναι η μέση τιμή των παρατηρήσεων για τη στάθμη i:

$$\overline{y}_{i.} = \frac{1}{r}\sum_{j=1}^{r} y_{ij} \tag{9.34}$$

Μετά τον υπολογισμό των παραμέτρων, το μοντέλο μπορεί να χρησιμοποιηθεί για την εκτίμηση $\tilde{y}$ της απόκρισης για κάθε μία από τις a στάθμες:

$$\tilde{y}_i = \mu + \alpha_i = \overline{y}_{i.} \tag{9.35}$$

Η διαφορά ανάμεσα στην εκτίμηση $\tilde{y}_i$ και τη μέτρηση y_{ij} εκφράζει το πειραματικό σφάλμα: $e_{ij} = y_{ij} - \tilde{y}_i$.

Κατανομή της μεταβλητότητας. Η συνολική μεταβλητότητα του y θα είναι:

$$SST = \sum_{i=1}^{a}\sum_{j=1}^{r}(y_{ij} - \overline{y}_{..})^2 \tag{9.36}$$

και μπορεί να πάρει τη μορφή:

$$SST = r\sum_{i=1}^{a}\alpha_i^2 + \sum_{i=1}^{a}\sum_{j=1}^{r} e_{ij}^2 = SSA + SSE \tag{9.37}$$

όπου SSE είναι το μέρος της μεταβλητότητας που δεν επεξηγείται από το μοντέλο.Το άθροισμα τετραγώνων SST έχει $ar-1$ βαθμούς ελευθερίας από τους οποίους $a-1$ αντιστοιχούν στο SSA και $a(r-1)$ στο SSE (επιβεβαιώνοντας τη σχέση αντιστοιχίας μεταξύ αθροισμάτων τετραγώνων και βαθμών ελευθερίας).

Οι διασπορές που αντιστοιχούν στον παράγοντα A και στα σφάλματα θα είναι αντίστοιχα:

$$s_a^2 = \frac{SSA}{(a-1)} \tag{9.38}$$

$$s_e^2 = \frac{SSE}{a(r-1)} \tag{9.39}$$

και μπορούν να χρησιμοποιηθούν άμεσα για τον υπολογισμό του F-test κατά την εφαρμογή της ANOVA:

$$F = \frac{s_a^2}{s_e^2}$$

Αν $F > F_{[1-\alpha;(a-1),a(r-1)]}$, μπορούμε να πούμε ότι η μεταβλητότητα που οφείλεται στις διαφορές μεταξύ των σταθμών του παράγοντα A είναι σημαντικά υψηλότερη από αυτήν που οφείλεται στα σφάλματα.

9.5.2 Πειράματα με Δύο Παράγοντες

Η περίπτωση αυτή αποτελεί άμεση γενίκευση της προηγούμενης. Θεωρούμε δύο παράγοντες A και B με a και b στάθμες αντίστοιχα. Για καθένα από τα ab δυνατά πειράματα εκτελούνται r επαναλήψεις.

Το μοντέλο θα έχει τη μορφή:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + e_{ijk} \tag{9.40}$$

όπου y_{ijk} είναι η k παρατήρηση με τον παράγοντα A στη στάθμη i και τον παράγοντα B στη στάθμη j, μ είναι η μέση απόκριση, α_i είναι η επίδραση της στάθμης i του A, β_j είναι η επίδραση της στάθμης j του

B, γ_{ij} είναι η *αλληλεπίδραση* (interaction) των A και B στις στάθμες i και j αντίστοιχα, και e_{ijk} είναι το πειραματικό σφάλμα. Οι επιδράσεις α_i και β_j υπολογίζονται με την απαίτηση ότι το άθροισμά τους είναι 0:

$$\begin{aligned}\sum_{i=1}^{a} \alpha_i &= 0 \\ \sum_{j=1}^{b} \beta_j &= 0\end{aligned}$$

Οι αλληλεπιδράσεις υπολογίζονται με την απαίτηση ότι:

$$\begin{aligned}\sum_{i=1}^{a} \gamma_{ij} &= 0, \quad j = 1, \ldots, b \\ \sum_{j=1}^{b} \gamma_{ij} &= 0, \quad i = 1, \ldots, a\end{aligned}$$

Τα σφάλματα έχουν άθροισμα 0 για κάθε πείραμα (συνδυασμό σταθμών):

$$\sum_{k=1}^{r} e_{ijk} = 0, \ \forall i, j$$

Σύμφωνα με τις υποθέσεις του μοντέλου, οι παράμετροι δίνονται από τις σχέσεις:

$$\mu = \overline{y}_{...} \tag{9.41}$$
$$\alpha_i = \overline{y}_{i..} - \overline{y}_{...} \tag{9.42}$$
$$\beta_j = \overline{y}_{.j.} - \overline{y}_{...} \tag{9.43}$$
$$\gamma_{ij} = \overline{y}_{ij.} - \overline{y}_{i..} - \overline{y}_{.j.} + \overline{y}_{...} \tag{9.44}$$

όπου, όπως και σε προηγούμενες περιπτώσεις, ο συμβολισμός `.΄ σημαίνει μέση τιμή των παρατηρήσεων για όλες τις τιμές του αντίστοιχου δείκτη:

$$\overline{y}_{...} = \frac{1}{abr} \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{r} y_{ijk} \tag{9.45}$$
$$\overline{y}_{ij.} = \frac{1}{r} \sum_{k=1}^{r} y_{ijk} \tag{9.46}$$
$$\overline{y}_{i..} = \frac{1}{br} \sum_{j=1}^{b} \sum_{k=1}^{r} y_{ijk} \tag{9.47}$$
$$\overline{y}_{.j.} = \frac{1}{ar} \sum_{i=1}^{a} \sum_{k=1}^{r} y_{ijk} \tag{9.48}$$

Μετά τον υπολογισμό των παραμέτρων, το μοντέλο μπορεί να χρησιμοποιηθεί για την εκτίμηση $\tilde{y}$ της απόκρισης για κάθε συνδυασμό τιμών των παραγόντων:

$$\tilde{y}_{ij} = \mu + \alpha_i + \beta_j + \gamma_{ij} = \overline{y}_{ij.} \tag{9.49}$$

Η διαφορά ανάμεσα στην εκτίμηση $\tilde{y}_{ij}$ και τη μέτρηση y_{ijk} εκφράζει το πειραματικό σφάλμα: $e_{ijk} = y_{ijk} - \tilde{y}_{ij}$.

Κατανομή της μεταβλητότητας. Η συνολική μεταβλητότητα του y θα είναι:

$$SST = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{r}(y_{ijk} - \overline{y}_{...})^2 \tag{9.50}$$

και μπορεί να πάρει τη μορφή:

$$\begin{aligned} SST &= br\sum_{i=1}^{a}\alpha_i^2 + ar\sum_{j=1}^{b}\beta_j^2 + r\sum_{i=1}^{a}\sum_{j=1}^{b}\gamma_{ij}^2 + \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{r}e_{ijk}^2 \\ &= SSA + SSB + SSAB + SSE \end{aligned}$$

όπου SSE είναι το μέρος της μεταβλητότητας που δεν επεξηγείται από το μοντέλο. Το άθροισμα SST έχει $abr-1$ βαθμούς ελευθερίας, από τους οποίους $a-1$ αντιστοιχούν στο SSA, $b-1$ αντιστοιχούν στο SSB, $(a-1)(b-1)$ αντιστοιχούν στο $SSAB$ και $ab(r-1)$ στο SSE.

Σε αναλογία με την περίπτωση του ενός παράγοντα μπορούμε να υπολογίσουμε τις διασπορές που αντιστοιχούν στους παράγοντες A και B, και στην αλληλεπίδρασή τους, καθώς και στα πειραματικά σφάλματα. Εν συνεχεία μπορούμε να εφαρμόσουμε την ANOVA, υπολογίζοντας το F-test για κάθε διασπορά συγκρίνοντάς την με αυτήν που αντιστοιχεί στα σφάλματα.

Παράδειγμα 9.4. *Θέλουμε να μελετήσουμε την επίδραση του επεξεργαστή* (2, 4 *ή* 6 *πυρήνων*) *και της λανθάνουσας μνήμης* (8 MB *ή* 12 MB) *στον χρόνο απόκρισης* (sec) *ενός υπολογιστικού συστήματος σε δεδομένη εφαρμογή (benchmark). Στον πίνακα δίνονται τα αποτελέσματα των μετρήσεων (*2 *επαναλήψεις ανά πείραμα). Ζητούνται οι παράμετροι για τις κύριες επιδράσεις και αλληλεπιδράσεις των παραγόντων, καθώς και η αντίστοιχη κατανομή της μεταβλητότητας. Ποιο ποσοστό της μεταβλητότητας αποδίδεται στα πειραματικά σφάλματα;*

	Αριθμός πυρήνων		
Λανθάνουσα μνήμη	2	4	6
8 MB	(1,20, 1,32)	(0,68, 0,72)	(0,42, 0,38)
12 MB	(0,67, 0,63)	(0,41, 0,45)	(0,25, 0,27)

Οι εξεταζόμενοι παράγοντες είναι ο επεξεργαστής (A) με τρεις στάθμες ($a = 3$) και η λανθάνουσα μνήμη (B) με δύο στάθμες ($b = 2$). Εκτελούμε $r = 2$ επαναλήψεις ανά πείραμα.

Με εφαρμογή των τύπων, βρίσκουμε για τις παραμέτρους του μοντέλου:
$\mu = 0,617$
$\alpha_1 = 0,338$, $\alpha_2 = -0,052$, $\alpha_3 = -0,286$
$\beta_1 = 0,17$, $\beta_2 = -0,17$
$\gamma_{11} = 0,135$, $\gamma_{21} = -0,035$, $\gamma_{31} = -0,1$
$\gamma_{12} = -0,135$, $\gamma_{22} = 0,035$, $\gamma_{32} = 0,1$
Παρατηρούμε ότι ικανοποιούνται όλοι οι περιορισμοί.

Η κατανομή της μεταβλητότητας δίνεται από τη σχέση
$SST = SSA + SSB + SSAB + SSE$
όπου
$SSA = 0,7950$, $SSB = 0,3468$, $SSAB = 0,1178$, $SSE = 0,0106$, και $SST = 1,2702$.

Το ποσοστό της μεταβλητότητας που αποδίδεται στα πειραματικά σφάλματα είναι 0,83%. □

9.6 Παραγοντικά Πειράματα με Δυαδικούς Παράγοντες

Οι υποθέσεις που αναφέρθηκαν προηγουμένως στην περίπτωση των πλήρων παραγοντικών πειραμάτων, καθώς και η συζήτηση σχετικά με τη χρήση πολλαπλασιαστικών μοντέλων και μετασχηματισμών αφορούν και την ανάλυση που παρουσιάζεται στη συνέχεια για παραγοντικά πειράματα με δυαδικούς παράγοντες.

Πείραμα	I	A	B	AB	y
1	1	−1	−1	1	y_1
2	1	1	−1	−1	y_2
3	1	−1	1	−1	y_3
4	1	1	1	1	y_4

Πίνακας 9.1: Πίνακας προσήμων

9.6.1 Πειράματα 2^k

Σε ένα παραγοντικό πείραμα 2^k θεωρούμε δύο στάθμες για κάθε παράγοντα, συνήθως την ελάχιστη και τη μέγιστη. Η σχεδίαση αυτή είναι ιδιαίτερα αποτελεσματική, όταν η επίδραση ενός παράγοντα είναι μονότονη, δηλαδή η επίδοση του συστήματος είτε αυξάνει συνεχώς είτε μειώνεται συνεχώς καθώς η τιμή του παράγοντα αυξάνει από το ελάχιστο ως το μέγιστό της.

Θα εξετάσουμε την περίπτωση δύο παραγόντων A και B (πείραμα 2^2). Οι μεταβλητές y_1, y_2, y_3 και y_4 παριστάνουν τις αποκρίσεις των 4 δυνατών πειραμάτων, ενώ οι μεταβλητές x_A και x_B εκφράζουν τους παράγοντες και μπορούν να πάρουν τις τιμές -1 και 1 για τις δύο στάθμες αντίστοιχα.

Πείραμα	A	B	y
1	−1	−1	y_1
2	1	−1	y_2
3	−1	1	y_3
4	1	1	y_4

Η επίδραση των παραγόντων A και B στην απόκριση y μπορεί να παρασταθεί με το παρακάτω μοντέλο μη γραμμικής παλινδρόμησης:

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B \tag{9.51}$$

Αντικαθιστώντας τις τιμές των μεταβλητών, όπως φαίνονται στον πίνακα, για τα 4 πειράματα βρίσκουμε τις τιμές των παραμέτρων:

$$\begin{aligned} q_0 &= (y_1 + y_2 + y_3 + y_4)/4 \\ q_A &= (-y_1 + y_2 - y_3 + y_4)/4 \\ q_B &= (-y_1 - y_2 + y_3 + y_4)/4 \\ q_{AB} &= (y_1 - y_2 - y_3 + y_4)/4 \end{aligned}$$

Η παράμετρος q_0 είναι η *μέση* απόκριση $\overline{y}$. Οι παράμετροι q_A και q_B είναι οι *επιδράσεις* (effects) των παραγόντων A και B, ενώ η παράμετρος q_{AB} είναι η *αλληλεπίδραση* (interaction) των παραγόντων.

Παρατηρούμε ότι οι συντελεστές των αποκρίσεων y_i στις εκφράσεις για τα q_A και q_B ταυτίζονται με τις στάθμες των A και B όπως φαίνονται στις αντίστοιχες στήλες του πίνακα. Ομοίως, οι συντελεστές για το q_{AB} προκύπτουν πολλαπλασιάζοντας τις στήλες. Η παρατήρηση αυτή οδηγεί στη μέθοδο του *πίνακα προσήμων* που επιτρέπει τον εύκολο υπολογισμό των επιδράσεων (Πίνακας 9.1).

Για ένα πείραμα 2^2 θεωρούμε 4 στήλες προσήμων. Η πρώτη στήλη με την επιγραφή I περιέχει μόνο 1. Οι στήλες με επιγραφές A και B περιέχουν όλους τους δυνατούς συνδυασμούς των -1 και 1, ενώ η στήλη με επιγραφή AB είναι το γινόμενο των στηλών A και B. Εκτός από τη στήλη I, οι υπόλοιπες στήλες έχουν άθροισμα 0. Όλες οι στήλες είναι ορθογώνιες μεταξύ τους, δηλαδή το εσωτερικό γινόμενό τους ανά δύο είναι 0.

Οι παρατηρήσεις y_i γράφονται σε μία στήλη μετά τις στήλες των προσήμων. Οι τιμές των παραμέτρων q προκύπτουν αν πολλαπλασιάσουμε τα στοιχεία κάθε στήλης προσήμων με τα αντίστοιχα στοιχεία της στήλης y των παρατηρήσεων και διαιρέσουμε το άθροισμα με το πλήθος των πειραμάτων.

Κατανομή της μεταβλητότητας. Η σπουδαιότητα ενός παράγοντα προσδιορίζεται από το ποσοστό της συνολικής μεταβλητότητας της απόκρισης που επεξηγείται από τον παράγοντα. Η συνολική μεταβλητότητα (SST) του y είναι $\sum_{i=1}^{2^2}(y_i - \overline{y})^2$ και μετά από μερικές αλγεβρικές πράξεις μπορεί να πάρει τη μορφή:

$$SST = 2^2(q_A^2 + q_B^2 + q_{AB}^2) = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2 \tag{9.52}$$

Οι τρεις όροι του δεξιού μέλους εκφράζουν το μέρος της συνολικής μεταβλητότητας που επεξηγείται από την επίδραση του A, του B και από την αλληλεπίδραση AB αντίστοιχα, και συμβολίζονται:

$$SST = SSA + SSB + SSAB \tag{9.53}$$

Το κλάσμα της μεταβλητότητας που επεξηγείται από έναν παράγοντα, π.χ. SSA/SST, χαρακτηρίζει τη σπουδαιότητα του παράγοντα και συνεπώς αξίζει περισσότερο λεπτομερής εξέταση της επίδρασής του.

Οι τεχνικές που συζητήθηκαν ως τώρα για τη σχεδίαση πειραμάτων 2^2 γενικεύονται άμεσα στη σχεδίαση 2^k. Στην περίπτωση αυτή έχουμε 2^k παραμέτρους, από τις οποίες k είναι κύριες επιδράσεις, $\begin{pmatrix} k \\ 2 \end{pmatrix}$ είναι αλληλεπιδράσεις παραγόντων ανά 2, $\begin{pmatrix} k \\ 3 \end{pmatrix}$ είναι αλληλεπιδράσεις παραγόντων ανά 3, κ.ο.κ. Η μέθοδος του πίνακα προσήμων εφαρμόζεται με τον ίδιο ακριβώς τρόπο, θεωρώντας 2^k γραμμές και 2^k στήλες προσήμων. Η έκφραση για τη συνολική μεταβλητότητα SST στη γενική της μορφή αποτελείται από το άθροισμα των τετραγώνων όλων των επιδράσεων και αλληλεπιδράσεων πολλαπλασιασμένο επί 2^k.

9.6.2 Πειράματα $2^k r$

Αν καθένα από τα 2^k πειράματα επαναλαμβάνεται r φορές, έχουμε τη σχεδίαση $2^k r$, η οποία περιλαμβάνει και την εκτίμηση πειραματικών σφαλμάτων. Εξετάζουμε πάλι την περίπτωση δύο δυαδικών παραγόντων A και B, η οποία γενικεύεται εύκολα για k παράγοντες. Το μοντέλο παλινδρόμησης τροποποιείται με την προσθήκη ενός όρου για το σφάλμα:

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B + e \tag{9.54}$$

Η ανάλυση γίνεται ακριβώς όπως στην περίπτωση 2^2, αν θέσουμε στη στήλη y του πίνακα προσήμων τη μέση τιμή των r παρατηρήσεων κάθε πειράματος.

Μετά τον υπολογισμό των επιδράσεων q, το μοντέλο μπορεί να χρησιμοποιηθεί για την εκτίμηση $\tilde{y}$ της απόκρισης, όταν δίνονται οι τιμές x_{Ai} και x_{Bi} των παραγόντων:

$$\tilde{y}_i = q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{Ai} x_{Bi} \tag{9.55}$$

Η διαφορά ανάμεσα στην εκτίμηση $\tilde{y}_i$ και τη μέτρηση y_{ij} που προκύπτει από την j επανάληψη του i πειράματος εκφράζει το πειραματικό σφάλμα: $e_{ij} = y_{ij} - \tilde{y}_i$. Το άθροισμα των σφαλμάτων είναι 0 για κάθε i.

Κατανομή της μεταβλητότητας. Η συνολική μεταβλητότητα του y θα είναι:

$$SST = \sum_{i=1}^{2^2}\sum_{j=1}^{r}(y_{ij} - \overline{y}_{..})^2 \tag{9.56}$$

όπου $\overline{y}_{..}$ είναι η μέση τιμή των παρατηρήσεων από όλες τις επαναλήψεις όλων των πειραμάτων (ίση με q_0). Η μεταβλητότητα SST μπορεί να πάρει τη μορφή:

$$\begin{aligned} SST &= 2^2 r q_A^2 + 2^2 r q_B^2 + 2^2 r q_{AB}^2 + \sum_{i=1}^{2^2}\sum_{j=1}^{r} e_{ij}^2 \\ &= SSA + SSB + SSAB + SSE \end{aligned}$$

όπου SSE είναι το μέρος της μεταβλητότητας που δεν επεξηγείται από το μοντέλο και αποδίδεται στα πειραματικά σφάλματα.

Το άθροισμα SSE μπορεί να χρησιμοποιηθεί για την εκτίμηση της διασποράς των σφαλμάτων:

$$s_e^2 = \frac{SSE}{2^2(r-1)}$$

όπου οι βαθμοί ελευθερίας στον παρονομαστή προκύπτουν από το γεγονός ότι για κάθε πείραμα τα σφάλματα των r επαναλήψεων έχουν άθροισμα 0.

Παράδειγμα 9.5. *Κατά τη σχεδίαση ενός συστήματος, μελετώνται με τη βοήθεια ενός benchmark τρεις παράγοντες: η ταχύτητα του επεξεργαστή* (3,4 GHz *ή* 4,8 GHz)*, το μέγεθος της κύριας μνήμης* (4 GB *ή* 8 GB) *και το μέγεθος της μνήμης της κάρτας γραφικών* (512 MB *ή* 1 GB)*. Για κάθε συνδυασμό των παραγόντων πραγματοποιήθηκαν δύο μετρήσεις (κανονικοποιημένος ρυθμός απόδοσης), όπως φαίνεται και στον πίνακα.*

Ζητείται ο προσδιορισμός των παραμέτρων για τις κύριες επιδράσεις και αλληλεπιδράσεις των παραγόντων, καθώς και η αντίστοιχη κατανομή της μεταβλητότητας. Ποιο ποσοστό της μεταβλητότητας αποδίδεται στα πειραματικά σφάλματα;

	4 GB		8 GB	
	512 MB	1 GB	512 MB	1 GB
3,4 GHz	(5,3, 5,9)	(12,7, 14,1)	(8,5, 9,0)	(20,1, 19,2)
4,8 GHz	(7,3, 8,1)	(18,2, 16,9)	(10,7, 11,5)	(25,6, 23,7)

Θα παραστήσουμε με A, B και C την ταχύτητα του επεξεργαστή, το μέγεθος της κύριας μνήμης και το μέγεθος της μνήμης της κάρτας γραφικών, αντίστοιχα. Για κάθε παράγοντα, συμβολίζουμε με -1 και 1 τη χαμηλή και την υψηλή στάθμη αντίστοιχα. Εκτελούμε $r = 2$ επαναλήψεις ανά πείραμα.

Καταρχάς, κατασκευάζουμε τον πίνακα προσήμων, στον οποίο έχουμε περιλάβει τις μετρήσεις κάθε πειράματος και τη μέση τιμή τους:

Πείραμα	I	A	B	C	AB	BC	CA	ABC	y	$\overline{y}$
1	1	−1	−1	−1	1	1	1	−1	(5,3, 5,9)	5,6
2	1	1	−1	−1	−1	1	−1	1	(7,3, 8,1)	7,7
3	1	−1	1	−1	−1	−1	1	1	(8,5, 9,0)	8,75
4	1	1	1	−1	1	−1	−1	−1	(10,7, 11,5)	11,1
5	1	−1	−1	1	1	−1	−1	1	(12,7, 14,1)	13,4
6	1	1	−1	1	−1	−1	1	−1	(18,2, 16,9)	17,55
7	1	−1	1	1	−1	1	−1	−1	(20,1, 19,2)	19,65
8	1	1	1	1	1	1	1	1	(25,6, 23,7)	24,65

Υπολογίζουμε τις επιδράσεις και αλληλεπιδράσεις των παραγόντων $q_0 = 13,55$, $q_A = 1,7$, $q_B = 2,488$, $q_C = 5,263$, $q_{AB} = 0,138$, $q_{BC} = 0,85$, $q_{CA} = 0,588$, $q_{ABC} = 0,075$, οπότε έχουμε:

$$\begin{aligned}\tilde{y} &= 13,55 + 1,7x_A + 2,488x_B + 5,263x_C + \\ &\quad 0,138x_Ax_B + 0,85x_Bx_C + 0,588x_Cx_A + 0,075x_Ax_Bx_C\end{aligned}$$

Η κατανομή της μεταβλητότητας δίνεται από τη σχέση
$SST = SSA + SSB + SSC + SSAB + SSBC + SSCA + SSABC + SSE$
όπου $SSA = 46,24$, $SSB = 99,00$, $SSC = 443,10$, $SSAB = 0,3025$, $SSBC = 11,56$, $SSCA = 5,5225$, $SSABC = 0,09$, $SSE = 4,98$, και $SST = 610,795$.

Το ποσοστό της μεταβλητότητας που δεν επεξηγείται και αποδίδεται στα πειραματικά σφάλματα είναι 0,82%. □

9.7 Κλασματικά Παραγοντικά Πειράματα 2^{k-p}

Η κλασματική παραγοντική σχεδίαση 2^{k-p} επιτρέπει την ανάλυση της επίδρασης k παραγόντων με δύο στάθμες πραγματοποιώντας μόνο 2^{k-p} πειράματα. Η βασική τεχνική στηρίζεται στην κατασκευή ενός πίνακα με 2^{k-p} στήλες προσήμων, οι οποίες αντιστοιχούν σε κύριες επιδράσεις ή αλληλεπιδράσεις παραγόντων. Είναι προφανές ότι θα απουσιάζουν $2^k - 2^{k-p}$ από το σύνολο των δυνατών επιδράσεων. Όπως θα δούμε, η συμβολή των επιδράσεων αυτών περιλαμβάνεται έμμεσα στο μοντέλο καθόσον 'συγχέεται' με τη συμβολή των υπολοίπων επιδράσεων.

Το πρώτο βήμα της μεθόδου είναι η κατασκευή ενός πίνακα με ορθογώνιες στήλες:
Επιλέγουμε $k-p$ παράγοντες από τους k και κατασκευάζουμε πίνακα προσήμων για πλήρη σχεδίαση με $k-p$ παράγοντες. Η πρώτη στήλη επιγράφεται I και περιέχει μόνο 1. Οι επόμενες $k-p$ στήλες αποδίδονται στους $k-p$ παράγοντες που επιλέχθηκαν. Οι υπόλοιπες $2^{k-p} - (k-p) - 1$ στήλες είναι γινόμενα των παραγόντων αυτών. Από τις στήλες αυτές επιλέγουμε p και τις αναθέτουμε στους p παράγοντες που δεν επιλέχθηκαν αρχικά.

Π.χ., για να σχεδιάσουμε ένα πείραμα 2^{4-1} με παράγοντες A, B, C και D, αρχίζουμε με έναν πίνακα προσήμων 2^3, ο οποίος περιλαμβάνει τις στήλες I, A, B, C, AB, AC, BC και ABC. Από τις τέσσερις τελευταίες στήλες επιλέγουμε την τελευταία και την αναθέτουμε στον παράγοντα D. Η σχεδίαση που προκύπτει μας επιτρέπει να υπολογίσουμε τις κύριες επιδράσεις q_A, q_B, q_C και q_D, καθώς και τις αλληλεπιδράσεις q_{AB}, q_{AC} και q_{BC}, χρησιμοποιώντας ακριβώς τις ίδιες σχέσεις που θα χρησιμοποιούσαμε σε ένα πλήρες παραγοντικό πείραμα 2^3.

Πείραμα	I	A	B	C	AB	BC	AC	D
1	1	−1	−1	−1	1	1	1	−1
2	1	1	−1	−1	−1	1	−1	1
3	1	−1	1	−1	−1	−1	1	1
4	1	1	1	−1	1	−1	−1	−1
5	1	−1	−1	1	1	−1	−1	1
6	1	1	−1	1	−1	−1	1	−1
7	1	−1	1	1	−1	1	−1	−1
8	1	1	1	1	1	1	1	1

Το πρόβλημα με την κλασματική παραγοντική σχεδίαση είναι ότι δεν είναι δυνατόν να προσδιοριστούν όλες οι επιδράσεις. Στην πραγματικότητα κάθε στήλη παριστάνει τη συνισταμένη δύο ή περισσότερων επιδράσεων. Το φαινόμενο αυτό αναφέρεται ως σύγχυση (confounding) και οι επιδράσεις των οποίων η συμβολή δεν μπορεί να διαχωριστεί ονομάζονται συγκεχυμένες. Προφανώς η σχεδίαση ενός κλασματικού παραγοντικού πειράματος δεν είναι μοναδική, άρα μπορούν να προκύψουν διάφοροι συνδυασμοί συγκεχυμένων επιδράσεων.

Σε ένα πείραμα 2^{4-1}, όπως αυτό του παραδείγματος, μόνο 8 παράμετροι από τις 16 μπορούν να υπολογιστούν, επομένως κάθε παράμετρος αντιπροσωπεύει 2 επιδράσεις. Συνεπώς, η παράμετρος q' που υπολογίζεται με βάση την τελευταία στήλη του πίνακα θα είναι το άθροισμα των συγκεχυμένων επιδράσεων:

$$q' = q_{ABC} + q_D$$

Η σύγχυση δύο επιδράσεων μπορεί να παρασταθεί με έναν συμβολισμό αλγεβρικής ισότητας, π.χ. $D = ABC$ στο παράδειγμα, όπου ο πολλαπλασιασμός δηλώνει αλληλεπίδραση παραγόντων. Δεδομένης της σύγχυσης δύο επιδράσεων, είναι δυνατόν να προσδιοριστούν όλες οι άλλες συγχύσεις πολλαπλασιάζοντας τα δύο μέλη του αλγεβρικού συμβολισμού με κάποια επίδραση και χρησιμοποιώντας δύο απλούς κανόνες που βασίζονται στις ιδιότητες του πίνακα προσήμων: (α) το I συμπεριφέρεται ως μονάδα (ουδέτερο στοιχείο), π.χ. $IA = A$, και (β) κάθε όρος που υψώνεται στο τετράγωνο απαλείφεται, π.χ. $A^2BC = BC$. Σύμφωνα με τα παραπάνω, για το παράδειγμα προκύπτουν οι ακόλουθες συγχύσεις: $D = ABC$, $AD = BC$, $C = ABD$, $AB = CD$, $AC = BD$, $A = BCD$, $B = ACD$ και $I = ABCD$. Η σχέση $I = ABCD$ χρησιμοποιείται ως «γεννήτρια» όλων των υπολοίπων και χαρακτηρίζει τη σχεδίαση. Γενικά, σε μία σχεδίαση 2^{k-p} η γεννήτρια σχέση παριστάνει την ισότητα (σύγχυση) 2^p όρων (επιδράσεων).

Η *τάξη* μιας σύγχυσης είναι $i + j$ όταν συγχέονται μία επίδραση τάξης i και μία επίδραση τάξης j, π.χ. η σύγχυση $A = BCD$ είναι τάξης 4. Η ελάχιστη τάξη όλων των συγχύσεων μιας κλασματικής παραγοντικής σχεδίασης καθορίζει την *ανάλυση* (resolution) της σχεδίασης, π.χ. η σχεδίαση του παραδείγματος έχει ανάλυση 4. Δεδομένου ότι συνήθως οι αλληλεπιδράσεις υψηλής τάξης είναι πολύ μικρότερες από τις επιδράσεις χαμηλής τάξης, θα πρέπει να επιδιώκεται η σχεδίαση πειραμάτων υψηλής ανάλυσης.

Παράδειγμα 9.6. *Κατά τη σχεδίαση ενός συστήματος μελετώνται με τη βοήθεια ενός benchmark τέσσερις παράγοντες: η ταχύτητα του επεξεργαστή* (2,4 GHz *ή* 3,6 GHz)*, το μέγεθος της κύριας μνήμης* (1 GB *ή* 4 GB)*, η ταχύτητα του σκληρού δίσκου* (7.200 RPM *ή* 10.000 RPM) *και το λειτουργικό σύστημα* (Windows *ή* Unix)*. Στον πίνακα δίνεται κλασματικό πείραμα* 2^{4-1} *με τα αποτελέσματα των μετρήσεων (κανονικοποιημένος ρυθμός απόδοσης). Να προσδιοριστούν οι συγχύσεις και η ανάλυση της σχεδίασης, καθώς και οι παράμετροι για τις κύριες επιδράσεις και αλληλεπιδράσεις των παραγόντων.*

Πείραμα	Επεξεργαστής	Κύρια μνήμη	Δίσκος	Λειτουργικό	y
1	2,4 GHz	1 GB	7.200 RPM	Windows	34
2	3,6 GHz	1 GB	7.200 RPM	Windows	43
3	2,4 GHz	4 GB	7.200 RPM	Unix	65
4	3,6 GHz	4 GB	7.200 RPM	Unix	87
5	2,4 GHz	1 GB	10.000 RPM	Unix	40
6	3,6 GHz	1 GB	10.000 RPM	Unix	49
7	2,4 GHz	4 GB	10.000 RPM	Windows	80
8	3,6 GHz	4 GB	10.000 RPM	Windows	102

Θα παραστήσουμε με A, B, C και D την ταχύτητα του επεξεργαστή, το μέγεθος της κύριας μνήμης, την ταχύτητα του δίσκου και το λειτουργικό σύστημα αντίστοιχα. Για κάθε παράγοντα, συμβολίζουμε με -1 και 1 τις δύο στάθμες του αντίστοιχα. Κατασκευάζουμε πίνακα προσήμων 2^3, στον οποίο τοποθετούμε αρχικά τους παράγοντες A, B, C και τις αλληλεπιδράσεις τους. Με βάση τα δεδομένα του παραπάνω πίνακα διαπιστώνουμε ότι ο παράγων D έχει τοποθετηθεί στη στήλη BC, συνεπώς έχουμε τη σύγχυση $D = BC$:

Πείραμα	I	A	B	C	AB	BC	CA	ABC	D	y
1	1	−1	−1	−1	1	1	1	−1	1	34
2	1	1	−1	−1	−1	1	−1	1	1	43
3	1	−1	1	−1	−1	−1	1	1	−1	65
4	1	1	1	−1	1	−1	−1	−1	−1	87
5	1	−1	−1	1	1	−1	−1	1	−1	40
6	1	1	−1	1	−1	−1	1	−1	−1	49
7	1	−1	1	1	−1	1	−1	−1	1	80
8	1	1	1	1	1	1	1	1	1	102

Οι υπόλοιπες συγχύσεις είναι: $I = BCD$, $AD = ABC$, $A = ABCD$, $B = CD$, $C = BD$, $AC = ABD$, $AB = ACD$. Η ανάλυση της σχεδίασης είναι 3, δηλαδή είναι σχετικά χαμηλή. Αυτό είναι αναμενόμενο, αφού οι κύριες επιδράσεις συγχέονται με αλληλεπιδράσεις χαμηλής τάξης.

Υπολογίζουμε τις επιδράσεις και αλληλεπιδράσεις των παραγόντων λαμβάνοντας υπόψη τις συγχύσεις:

$$q_0 + q_{BCD} = 62,5 \qquad q_A + q_{ABCD} = 7,75$$
$$q_B + q_{CD} = 21 \qquad q_C + q_{BD} = 5,25$$
$$q_{AB} + q_{ACD} = 3,25 \qquad q_{BC} + q_D = 2,25$$
$$q_{CA} + q_{ABD} = 0 \qquad q_{ABC} + q_{AD} = 0$$

□

Βιβλιογραφία

[1] Fortier, P.J., and Michel, H.E., *Computer Systems Performance Evaluation and Prediction*, Elsevier Science, 2003.

[2] Gunther, N.J., *The Practical Performance Analyst*, Authors Choice Press, 2000.

[3] Jain, R., *The Art of Computer Systems Performance Analysis - Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, 1991.

[4] Leung, C.H.C., *Quantitative Analysis of Computer Systems*, John Wiley & Sons, 1988.

[5] Lilja, D.J., *Measuring Computer Performance: A Practitioner's Guide*, Cambridge University Press, 2000.

[6] MacNair, E.A. and Sauer, C.H., *Elements of Practical Performance Modeling*, Prentice-Hall, 1985.

[7] Menasce, D.A., and Almeida, V.A.F., *Capacity Planning for Web Services: Metrics, Models and Methods*, Prentice-Hall, 2002.

[8] Menasce, D.A., Almeida, V.A.F., and Dowdy, L.W., *Performance by Design, Computer Capacity Planning by Example*, Prentice-Hall PTR, 2004.

[9] Obaidat, M.S., and Boudriga, N.A., *Performance Evaluation of Computer and Telecommunication Systems*, Wiley, 2010.

Κεφάλαιο 10

Εργαλεία και Διαδικασίες

Σύνοψη
Στο τελευταίο αυτό κεφάλαιο, εν είδει συμπεράσματος, γίνεται αναφορά στη χρήση εργαλείων λογισμικού και στη μεθοδολογία σχεδίασης και υλοποίησης μιας μελέτης για την ανάλυση επίδοσης. Εξετάζονται τα κυριότερα χαρακτηριστικά τέτοιων εργαλείων, με έμφαση σε εργαλεία μοντελοποίησης που βασίζονται σε δίκτυα αναμονής και στη μέθοδο της προσομοίωσης. Γίνεται περιγραφή ορισμένων εργαλείων και δίνονται παραδείγματα, όσον αφορά τη δομή των λειτουργιών και τη διεπαφή με τον χρήστη. Εν συνεχεία, συζητούνται κριτήρια και διαδικασίες βάσει των οποίων γίνεται η εφαρμογή των κατάλληλων μεθόδων ή εργαλείων στο πλαίσιο ενός έργου μελέτης της επίδοσης, με έμφαση στα χαρακτηριστικά των συστημάτων που βασίζονται στον Ιστό.

10.1 Εργαλεία Λογισμικού

Η αυξημένη πολυπλοκότητα των σύγχρονων συστημάτων έχει καταστήσει σαφή την ανάγκη για εργαλεία ανάλυσης της επίδοσης. Σε περιπτώσεις δυσκολίας ή αδυναμίας διεξαγωγής πραγματικών μετρήσεων, η μοντελοποίηση είναι πολύ αποτελεσματική επιλογή. Διάφορα εργαλεία λογισμικού έχουν αναπτυχθεί, καθένα από τα οποία περιλαμβάνει μία ή περισσότερες μεθόδους για την ανάπτυξη και επίλυση μοντέλων επίδοσης. Γενικά, πέραν της ποικιλίας των ενσωματωμένων μεθόδων μοντελοποίησης, τα εργαλεία μπορούν να χαρακτηριστούν και ως προς διάφορες άλλες ιδιότητες [7, 5, 14].

10.1.1 Χαρακτηριστικά

Οι κυριότερες μέθοδοι που παρέχονται από τα εργαλεία λογισμικού είναι οι ακόλουθες:

- Δίκτυα αναμονής μορφής γινομένου (ακριβής ή προσεγγιστική λύση),
- Προσεγγιστικές τεχνικές για δίκτυα αναμονής που δεν επιδέχονται λύση μορφής γινομένου,
- Αλυσίδες Markov συνεχούς ή διακριτού χρόνου,
- Προσομοίωση,
- Συνδυασμός των παραπάνω μεθόδων (υβριδικά/ιεραρχικά μοντέλα),
- Τεχνικές ανάλυσης δεδομένων και χαρακτηρισμού φορτίου.

Τα μοντέλα της θεωρίας αναμονής επιτρέπουν την αναπαράσταση του συστήματος με απλό και συνεκτικό τρόπο, με σχετικά περιορισμένες δυνατότητες, όμως, ως προς τον βαθμό λεπτομέρειας και την ακρίβεια της περιγραφής. Οι αποδοτικοί αλγόριθμοι επίλυσης προϋποθέτουν ικανοποίηση των απαιτήσεων για λύση σε μορφή γινομένου, που συχνά δεν ισχύει στην πράξη (π.χ. ύπαρξη παραλληλίας, αποκλεισμού κλπ.). Στις περιπτώσεις αυτές, προσφεύγουμε σε προσεγγιστικές τεχνικές. Όταν οι τελευταίες εμφανίζουν μειωμένη

ακρίβεια ή αδυναμία απεικόνισης του προβλήματος, χρησιμοποιούμε γενικά μοντέλα αλυσίδων Markov ή προσομοίωση.

Η ανάπτυξη γενικών μαρκοβιανών μοντέλων επιτρέπει τη λεπτομερή αναπαράσταση του συστήματος, με τίμημα την εκθετική αύξηση του χώρου καταστάσεων. Πράγματι, η πολυλοκότητα είναι υψηλη ακόμη και για μικρά συστήματα, με αποτέλεσμα να είναι πρακτικά αδύνατη η επίλυση με το χέρι. Κατά συνέπεια, απαιτούνται εργαλεία λογισμικού, τα οποία δημιουργούν αυτομάτως τον χώρο καταστάσεων των αλυσίδων Markov και εφαρμόζουν κατάλληλες τεχνικές επίλυσης.

Η προσομοίωση παρέχει δυνατότητες για ακριβή περιγραφή της συμπεριφοράς του συστήματος, αλλά είναι απαιτητική όσον αφορά το υπολογιστικό κόστος και τη χρήση των πόρων. Τα μοντέλα προσομοίωσης μπορούν να υλοποιηθούν με γλώσσες προγραμματισμού γενικής χρήσης, γλώσσες προσομοίωσης ή πακέτα λογισμικού. Η χρήση κοινών γλωσσών προγραμματισμού συνεπάγεται υψηλό κόστος ανάπτυξης. Ενδείκνυται η χρήση αντικειμενοστρεφών γλωσσών, οι οποίες μπορούν να απεικονίσουν αποτελεσματικά τις οντότητες του πραγματικού κόσμου. Το κόστος ανάπτυξης μειώνεται με τη χρήση γλωσσών προσομοίωσης (Κεφ. 7), που ενσωματώνουν πολλά ειδικά χαρακτηριστικά, αλλά χαρακτηρίζονται από λιγότερη ευελιξία σε σχέση με τις γλώσσες γενικής χρήσης. Τα πακέτα προσομοίωσης συνήθως είναι εξειδικευμένα εργαλεία με μειωμένη ευελιξία, τα οποία απαιτούν μικρή προσπάθεια και κόστος χρήσης.

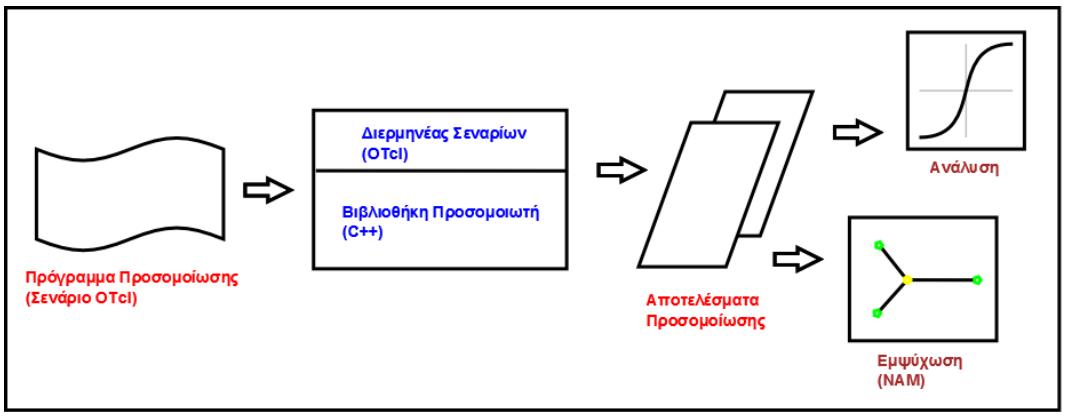
Όσον αφορά τις τεχνικές μετρήσεων, αυτές συνήθως υλοποιούνται ως εξειδικευμένα εργαλεία που σχετίζονται με συγκεκριμένους τύπους υλικού και λογισμικού και αναπτύσσονται από τους κατασκευαστές. Τα εργαλεία αυτά πραγματοποιούν μετρήσεις στο πλαίσιο λειτουργιών συντήρησης, ρύθμισης ή διάγνωσης προβλημάτων στο υλικό ή το λογισμικό του υπό μελέτη συστήματος. Σημειώνεται, πάντως, ότι, πέραν των εξειδικευμένων προϊόντων για λειτουργίες μετρήσεων, τα περισσότερα γενικά εργαλεία παρέχουν δυνατότητες στατιστικής ανάλυσης δεδομένων και αλγορίθμων χαρακτηρισμού φορτίου.

Όπως αναφέρθηκε παραπάνω, τα εργαλεία λογισμικού χαρακτηρίζονται και από άλλα γνωρίσματα πέραν των ενσωματωμένων αλγορίθμων. Καταρχάς, ένας σχετικός χαρακτηρισμός αφορά τον τύπο του μοντέλου (έννοιες και δομές) στο οποίο βασίζεται ο αλγόριθμος, π.χ. δίκτυο αναμονής, γράφος μεταβάσεων (μαρκοβιανής αλυσίδας) κλπ. Από τον τύπο αυτόν εξαρτώνται διάφορες λειτουργίες, όπως η αναπαράσταση του μοντέλου στην είσοδο και στην έξοδο, η δομή δεδομένων που υλοποιεί το μοντέλο (αποθήκευση, επεξεργασία), και άλλες.

Σημαντικό χαρακτηριστικό που λαμβάνεται υπόψη είναι η *διεπαφή με τον χρήστη* και οι δυνατότητες εισόδου/εξόδου. Μπορεί να υπάρχει γραφική διεπαφή χρήστη (graphical user interface — GUI) ή ειδική γλώσσα διεπαφής με εισαγωγή κειμένου (διαλογική ή batch) ή και τα δύο. Όσον αφορά την έξοδο, μπορεί να προβλέπονται διαφορετικές μορφές προβολής των αποτελεσμάτων, όπως γραφική απεικόνιση, αναφορά, εμψύχωση (animation). Η ευχρηστία της διεπαφής είναι σημαντική για την ταχεία ανάπτυξη μοντέλων χωρίς να απαιτείται γνώση των λεπτομερειών της υλοποίησης. Καθόσον οι περισσότερες λειτουργίες είναι προκαθορισμένες, περιορίζεται η ανάγκη συγγραφής κώδικα και μειώνεται η πιθανότητα σφαλμάτων.

Στη φάση της επιλογής κατάλληλου εργαλείου, μπορούν να συνεκτιμηθούν διάφορα ακόμη στοιχεία.

- Ορθότητα (accuracy) που παρέχεται από το εργαλείο,
- Καταλληλότητα του εργαλείου για το σύστημα υπό μελέτη και η ακρίβεια αναπαράστασης του συστήματος,
- Ταχύτητα εκτέλεσης του προγράμματος,
- Ύπαρξη συγκεκριμένων δυνατοτήτων ως προς τη διαχείριση των συστατικών και των ιδιοτήτων του μοντέλου,
- Δυνατότητα σύνδεσης με εξωτερικό κώδικα (συμπληρωματικές λειτουργίες, υβριδικά μοντέλα),
- Ευελιξία όσον αφορά τη δυνατότητα υποστήριξης εννοιών και δομών,
- Διαθεσιμότητα και φορητότητα,
- Υποστήριξη διόρθωσης σφαλμάτων και επίλυσης προβλημάτων (troubleshooting).



Σχήμα 10.1: *NS 2 — Άποψη της δομής του προσομοιωτή.*

Ο αριθμός των πακέτων λογισμικού, που έχουν αναπτυχθεί για τη μοντελοποίηση και ανάλυση υπολογιστικών συστημάτων, είναι μεγάλος. Άλλα εργαλεία είναι ακριβά εμπορικά προϊόντα και άλλα είναι ανοικτά προϊόντα που διατίθενται ελεύθερα. Αντίστοιχα, άλλα έχουν αναπτυχθεί από εταιρείες και άλλα από ακαδημαϊκούς φορείς. Τέλος, μπορούμε να αναφερθούμε σε γενικά (generic) και εξειδικευμένα εργαλεία. Τα πρώτα δηλώνουν εργαλεία που δεν προορίζονται αποκλειστικά για την ανάλυση υπολογιστικών συστημάτων και τηλεπικοινωνιακών δικτύων, ενώ τα δεύτερα έχουν ρυθμιστεί για αποδοτική χρήση στην αναπαράσταση τέτοιων συστημάτων.

10.1.2 Παραδείγματα

Θα περιγράψουμε εν συντομία τρία παραδείγματα εργαλείων λογισμικού, τα οποία έχουν αναπτυχθεί σε πανεπιστημιακά ιδρύματα και διατίθενται ελεύθερα. Πρόκειται για ένα εργαλείο προσομοίωσης και δύο εργαλεία–συλλογές μεθόδων όλων των τύπων.

10.1.2.1 Network Simulator 2 (NS 2)

Το πακέτο NS είναι ένα δημοφιλές εξειδικευμένο εργαλείο για την προσομοίωση δικτύων υπολογιστών [6, 8], το οποίο αναπτύχθηκε στο Πανεπιστήμιο του Berkeley. Περιλαμβάνει την υλοποίηση πρωτοκόλλων δικτύου, όπως το Transmission Control Protocol (TCP), το User Datagram Protocol (UDP) και το Hypertext Transfer Protocol (HTTP), καθώς επίσης και διάφορα χαρακτηριστικά της λειτουργίας του δικτύου, όπως οι εντολές ftp, telnet, τύπους κυκλοφορίας, όπως Constant Bit Rate (CBR) και Variable Bit Rate (VBR), μηχανισμούς διαχείρισης δρομολογητή, ουρές αναμονής, αλγορίθμους δρομολόγησης κλπ.

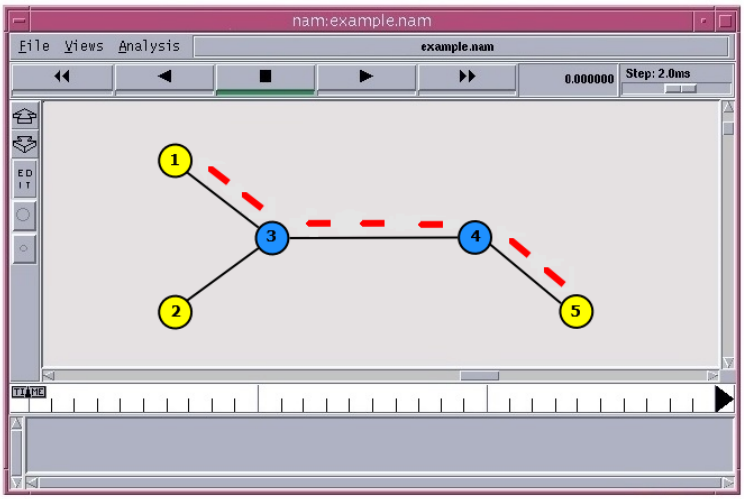
Η έκδοση 2 (NS 2) βασίζεται στην τεχνική της χρονοδρομολόγησης γεγονότων και έχει σχεδιαστεί σύμφωνα με τις αρχές του αντικειμενοστρεφούς προγραμματισμού. Η ανάπτυξη του NS 2 έγινε με χρήση της γλώσσας προγραμματισμού C++ και της γλώσσας σεναρίων (script language) OTcl. Η τελευταία δημιουργήθηκε στο MIT ως αντικειμενοστρεφής επέκταση της γλώσσας σεναρίων Tcl (Tool command language).

Οι βασικές συνιστώσες του προσομοιωτή NS 2 είναι οι ακόλουθες (Σχ. 10.1):

- Διερμηνέας σεναρίων (script interpreter). Το πρόγραμμα (σενάριο) προσομοίωσης, γραμμένο σε OTcl, εκτελείται από τον διερμηνέα OTcl ενεργοποιώντας τις επόμενες συνιστώσες. Γενικά, οι λειτουργίες ελέγχου της προσομοίωσης είναι γραμμένες σε OTcl.

- Βιβλιοθήκη προσομοιωτή.

  – Κόμβοι και συνδέσεις. Είναι αντικείμενα, γραμμένα σε C++, τα οποία ορίζουν την τοπολογία του δικτύου και διαχειρίζονται τη διακίνηση των πακέτων μέσα στο δίκτυο και τις ουρές αναμονής.



Σχήμα 10.2: *NS 2 — Διεπαφή του εμψυχωτή δικτύου.*

- Πηγές πακέτων. Οι πηγές (traffic sources) δημιουργούν πακέτα και τα διαβιβάζουν στο δίκτυο. Ένα γεγονός σχετίζεται με ένα χρονοδρομολογημένο πακέτο και τον κόμβο, ο οποίος θα χειριστεί το γεγονός (συνήθως είναι ο ίδιος κόμβος που δρομολόγησε το γεγονός).
- Χρονοδρομολογητής γεγονότων (event scheduler). Ο χρονοδρομολογητής γεγονότων παρακολουθεί τον χρόνο προσομοίωσης (ρολόι) και διαχειρίζεται τη λίστα γεγονότων. Όπως οι περισσότερες συνιστώσες του δικτύου, είναι γραμμένος σε C++.

- Έξοδος του προσομοιωτή. Με το πέρας της προσομοίωσης, παράγονται αποτελέσματα σε μορφή αρχείων κειμένου, τα οποία χρησιμοποιούνται για ανάλυση ή τροφοδοτούν ένα εργαλείο γραφικής απεικόνισης που ονομάζεται «εμψυχωτής δικτύου» (network animator — NAM).

Στο Σχήμα 10.2 παρουσιάζεται ένα στιγμιότυπο της διεπαφής του NAM.

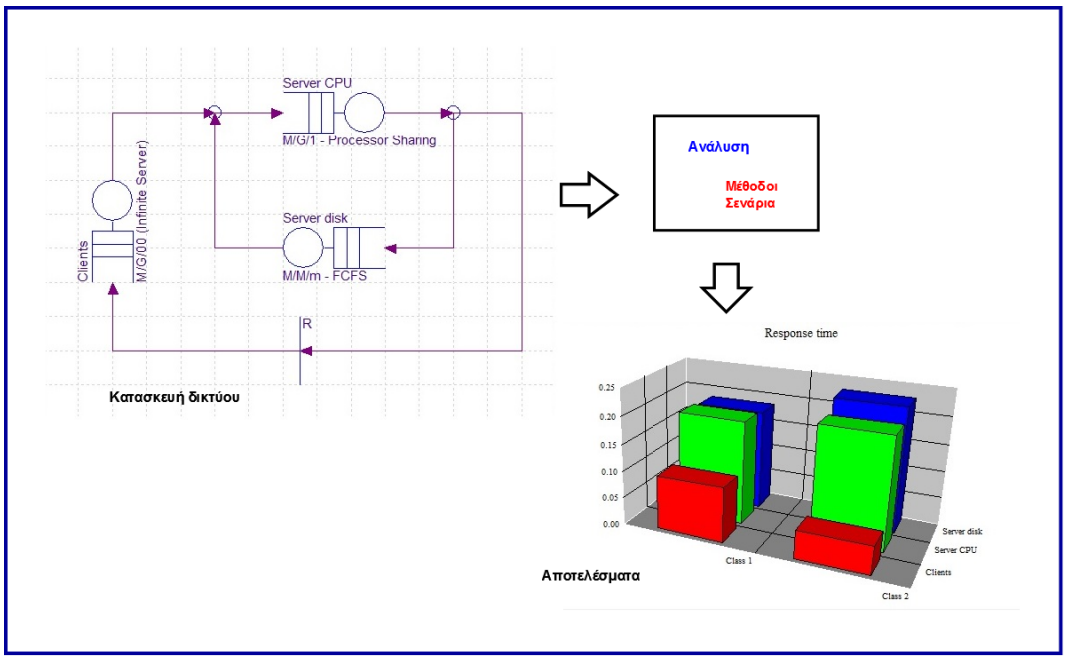
10.1.2.2 Performance Evaluation and Prediction System (PEPSY)

Το εργαλείο PEPSY δημιουργήθηκε στο Πανεπιστήμιο του Erlangen-Nürnberg [4] και επιτρέπει τον ορισμό και την επίλυση μοντέλων δικτύων αναμονής –τόσο μοντέλων που επιδέχονται λύση μορφής γινομένου όσο και μοντέλων που δεν διαθέτουν αυτή την ιδιότητα. Περιλαμβάνει περίπου 40 αλγορίθμους, ακριβείς ή προσεγγιστικούς, για ανοικτά, κλειστά και μικτά δίκτυα αναμονής. Επίσης, το πακέτο περιέχει μαρκοβιανή ανάλυση και προσομοίωση διακριτών γεγονότων. Η διεπαφή με τον χρήστη στην είσοδο γίνεται σε μορφή κειμένου ή σε γραφική μορφή, και περιλαμβάνει τον ορισμό του δικτύου (network editor) και τον ορισμό σεναρίων (scenario editor). Τα αποτελέσματα της ανάλυσης μπορούν να απεικονιστούν στην έξοδο σε μορφή πινάκων ή μέσω γραφικών παραστάσεων. Το PEPSY αναπτύχθηκε αρχικά για συστήματα UNIX. Η έκδοση για Windows, που αναπτύχθηκε εν συνεχεία, ονομάζεται WinPEPSY και έχει παρόμοια χαρακτηριστικά, με μόνη διαφορά ότι περιλαμβάνει λιγότερους αλγορίθμους [1].

Το Σχήμα 10.3 απεικονίζει τη διαδικασία ανάλυσης δικτύων στο περιβάλλον WinPEPSY.

Το εργαλείο WinPEPSY, όπως και το PEPSY, μπορεί να αναλύσει δίκτυα πολλών κατηγοριών. Η δρομολόγηση των εργασιών περιγράφεται από τον μέσο αριθμό επισκέψεων ή εναλλακτικά από τις πιθανότητες δρομολόγησης. Η διαδικασία των αφίξεων για τις ανοικτές κατηγορίες δίνεται από τον μέσο ρυθμό αφίξεων και το τετράγωνο του συντελεστή μεταβλητότητας (coefficient of variation) του χρόνου μεταξύ αφίξεων. Ομοίως, η κατανομή του χρόνου εξυπηρέτησης σε έναν κόμβο περιγράφεται από τον μέσο ρυθμό εξυπηρέτησης και το τετράγωνο του συντελεστή μεταβλητότητας του χρόνου εξυπηρέτησης.

Προβλέπονται οι παρακάτω τύποι σταθμών δικτύου. Χρησιμοποιούμε τον συμβολισμό του Kendall χωρίς προσδιορισμό της διαδικασίας αφίξεων. (Ο τύπος σταθμού καθορίζεται από την κατανομή του χρόνου εξυπηρέτησης και τον κανονισμό εξυπηρέτησης.)



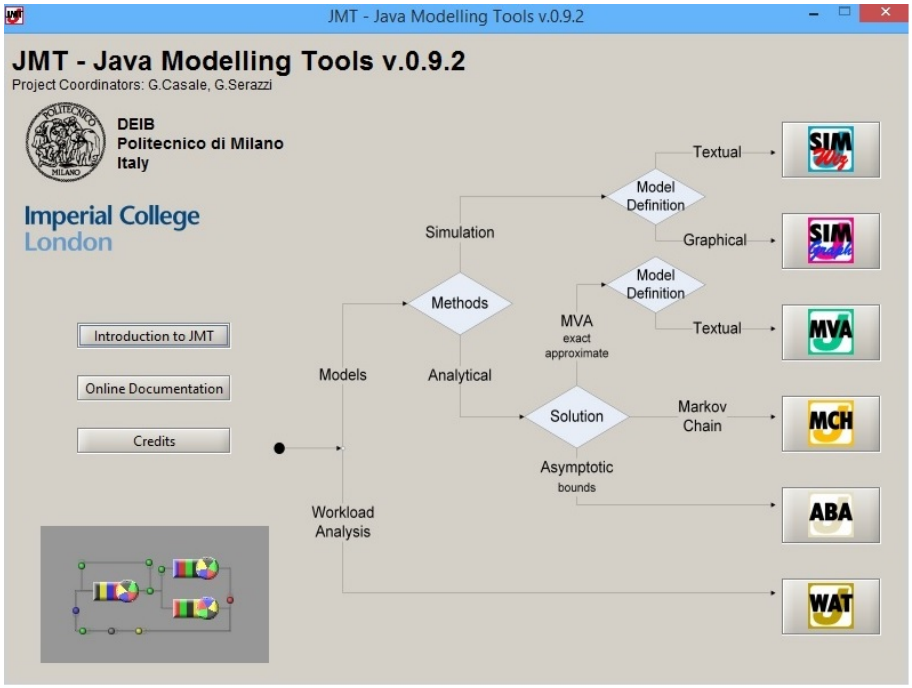
Σχήμα 10.3: *WinPEPSY — Μοντελοποίηση και ανάλυση.*

- $-/M/m$ – FCFS (First-Come-First-Served)
- $-/G/1$ – PS (Processor Sharing)
- $-/G/\infty$ – IS (Infinite Servers)
- $-/G/1$ – LCFS PR (Last-Come-First-Served Preemptive-Resume)
- $-/G/m$ – FCFS
- $-/M/m$ – FCFS asym (ασύμμετροι κόμβοι: στον ίδιο σταθμό μονάδες εξυπηρέτησης με διαφορετικές κατανομές χρόνου εξυπηρέτησης)
- $-/G/m$ – FCFS asym

Για την επίλυση των μοντέλων, το εργαλείο WinPEPSY διαθέτει τις ακόλουθες μεθόδους:

- Ανάλυση Μέσης Τιμής (MVA) για κλειστά δίκτυα μορφής γινομένου,
- Τεχνικές επίλυσης για ανοικτά δίκτυα μορφής γινομένου,
- Προσεγγίσεις για κλειστά δίκτυα που δεν επιδέχονται μορφή γινομένου (Μέθοδος του R. Marie [10, 11]),
- Μέθοδος της διάσπασης για ανοικτά δίκτυα που δεν επιδέχονται μορφή γινομένου,
- Μαρκοβιανή ανάλυση (επίλυση αλυσίδας Markov συνεχούς χρόνου στη μόνιμη κατάσταση),
- Προσομοίωση διακριτών γεγονότων.

Ο ορισμός σεναρίων επιτρέπει τη δημιουργία μιας σειράς εκτελέσεων της μεθόδου που εφαρμόζεται. Η σειρά αυτή προκύπτει από την επιλογή μιας έως δύο παραμέτρων του δικτύου (π.χ. αριθμός εργασιών μιας κλειστής κατηγορίας), για τις οποίες ορίζεται μια περιοχή τιμών. Η ανάλυση διεξάγεται αυτόματα για κάθε συνδυασμό τιμών των επιλεγμένων παραμέτρων. Τα αποτελέσματα της ανάλυσης (μέσες τιμές στη μόνιμη κατάσταση) είναι οι δείκτες επίδοσης για κάθε σταθμό και κατηγορία: χρόνος απόκρισης, χρόνος αναμονής, αριθμός εργασιών, αριθμός εργασιών σε αναμονή, βαθμός χρησιμοποίησης, ρυθμός απόδοσης σταθμών και δικτύου.



Σχήμα 10.4: *JMT — Κεντρική οθόνη.*

10.1.2.3 Java Modelling Tools (JMT)

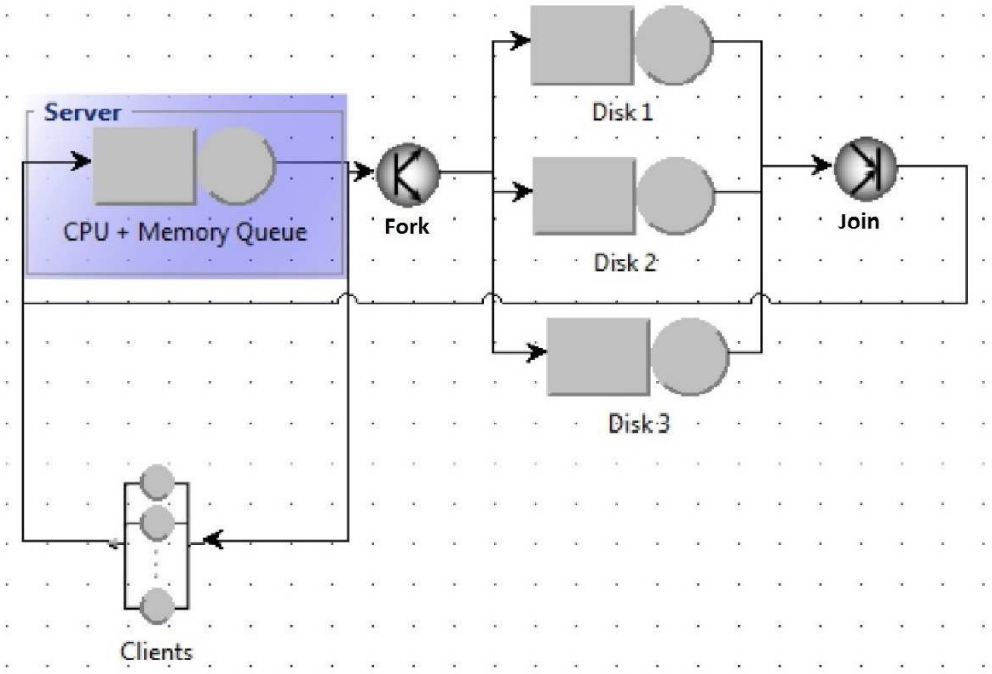
Η *σουίτα* JMT είναι ελεύθερο λογισμικό ανοικτού κώδικα και περιλαμβάνει έξι εργαλεία για τη μοντελοποίηση και ανάλυση υπολογιστικών και τηλεπικοινωνιακών συστημάτων [15, 2, 3]. Τα εργαλεία υλοποιούν διάφορους αλγορίθμους για την ακριβή ή προσεγγιστική επίλυση δικτύων αναμονής, είτε αυτά επιδέχονται λύση μορφής γινομένου είτε όχι. Επίσης, παρέχεται ασυμπτωτική ανάλυση, ανάλυση φορτίου και προσομοίωση διακριτών γεγονότων. Τα μοντέλα δικτύων ορίζονται μέσω γραφικής διεπαφής ή μέσω διαλογικής διεπαφής σε μορφή κειμένου (wizard).

Η ανάπτυξη του JMT άρχισε το 2002 στο Politecnico di Milano, χρησιμοποιώντας αρχικά ως γλώσσα προγραμματισμού τη *C* και εν συνεχεία τη *C++*. Από το 2002, ο κώδικας άρχισε να μετατρέπεται σε Java και το έργο έλαβε την τελική του μορφή και ονομασία. Τα τελευταία χρόνια, στην ομάδα ανάπτυξης του JMT (με συντονιστή τον G. Serazzi) συμμετέχει και το Imperial College London.

Η σουίτα περιλαμβάνει τα ακόλουθα εργαλεία. Η επιλογή γίνεται μέσα από την κεντρική γραφική διεπαφή της σουίτας (Σχ. 10.4).

- JSIM Ο προσομοιωτής διακριτών γεγονότων JSIM διατίθεται σε δύο εκδόσεις, την JSIMwiz και την JSIMgraph, που χρησιμοποιούν την ίδια μηχανή προσομοίωσης (JSIMengine) με διαφορετικές διεπαφές (διεπαφή κειμένου και γραφική διεπαφή, αντίστοιχα). Εκτός από τις συνήθεις δυνατότητες των γλωσσών και εργαλείων προσομοίωσης, η JSIM διαθέτει προηγμένα χαρακτηριστικά, όπως ρυθμούς εξυπηρέτησης εξαρτώμενους από το φορτίο, εκρηκτικά (bursty) φορτία, ειδικές στρατηγικές δρομολόγησης (π.χ. δρομολόγηση στον σταθμό με τον μικρότερο βαθμό χρησιμοποίησης), περιοχές του δικτύου με πεπερασμένη χωρητικότητα (blocking), παραλληλία (δομές fork-join), κατηγορίες με προτεραιότητες κλπ. Γίνεται αυτόματη ανίχνευση του μεταβατικού φαινομένου και αυτόματος τερματισμός όταν ικανοποιηθούν οι απαιτήσεις ακρίβειας. Είναι δυνατή η εκτέλεση σειράς πειραμάτων προσομοίωσης για διάφορες τιμές των παραμέτρων ελέγχου. Η ανάλυση των αποτελεσμάτων παρέχει εκτιμήσεις διαφόρων δεικτών επίδοσης σε επίπεδο σταθμών και σε επίπεδο δικτύου συνολικά, με αντίστοιχες δυνατότητες γραφικής απεικόνισης. Τα Σχήματα 10.5 και 10.6 απεικονίζουν ένα στιγμιότυπο γραφικής εισόδου και ένα δείγμα γραφικής παράστασης στην έξοδο, αντίστοιχα.

- JMVA Το εργαλείο αυτό παρέχει ακριβή ανάλυση ανοικτών, κλειστών και μικτών δικτύων μορφής



Σχήμα 10.5: *JMT — Γραφική διεπαφή κατασκευής μοντέλου.*

γινομένου μιας ή πολλών κατηγοριών, καθώς και προσεγγιστική ανάλυση με χρήση διαφόρων αλγορίθμων.

- JMCH Εφαρμόζει τεχνική προσομοίωσης για την επίλυση ορισμένων μοντέλων απλών συστημάτων αναμονής (γεννήσεων-θανάτων) και απεικονίζει τη δυναμική λειτουργία του μοντέλου (αλυσίδα Markov) με τη βοήθεια εμψύχωσης (animation).

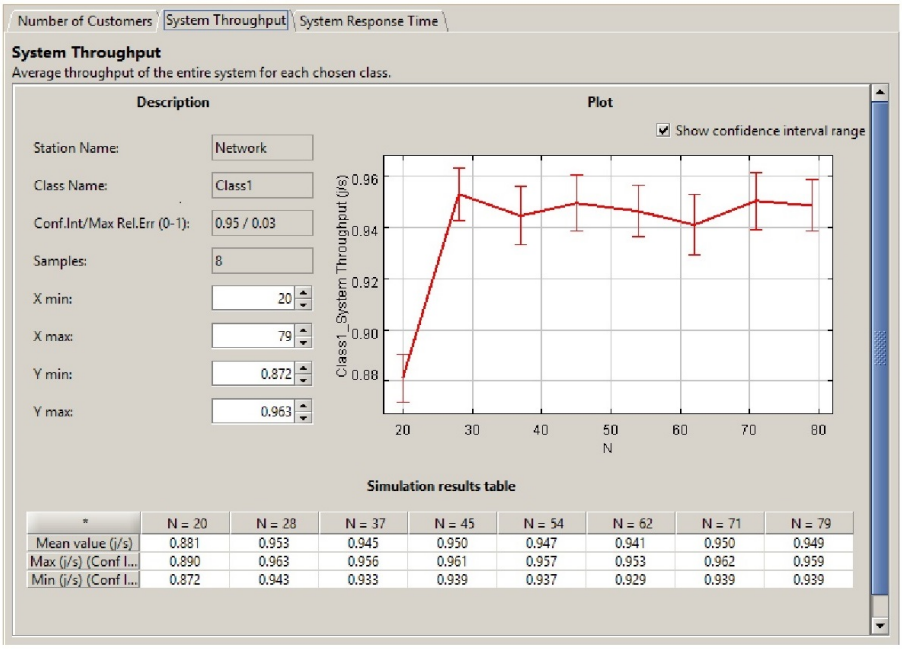
- JABA Πραγματοποιεί ασυμπτωτική ανάλυση στένωσης (bottleneck analysis) σε δίκτυα μορφής γινομένου πολλών κατηγοριών (μέχρι 3 κατηγορίες). Επίσης, εφαρμόζει τεχνικές βελτιστοποίησης των δεικτών επίδοσης ως προς τη σύνθεση του μίγματος των κατηγοριών στο σύστημα.

- JWAT Το εργαλείο αυτό παρέχει δυνατότητες χαρακτηρισμού του φορτίου. Υποστηρίζονται ορισμένοι τυποποιημένοι μορφότυποι αρχείων καταγραφής, καθώς και η δυνατότητα ορισμού μορφότυπων από τον χρήστη. Τα δεδομένα εισόδου μπορούν να αναλυθούν με εφαρμογή βασικών στατιστικών τεχνικών και μεθόδων ομαδοποίησης (clustering), για τον εντοπισμό ομάδων πελατών με παρόμοια χαρακτηριστικά. Οι μέσες τιμές των παραμέτρων των ομάδων χρησιμοποιούνται για τον χαρακτηρισμό του φορτίου.

Η αρχιτεκτονική του JMT αποτελείται από ένα σύνολο εφαρμογών διεπαφής σε χαλαρή σύζευξη μεταξύ τους, οι οποίες —μέσω ενός στρώματος XML— επικοινωνούν με ένα σύνολο υπολογιστικών εφαρμογών (μηχανή προσομοίωσης, αλγόριθμοι, μαθηματικές συναρτήσεις κλπ.). Ο διαχωρισμός της διεπαφής από το υπολογιστικό τμήμα επιτρέπει την απευθείας ενσωμάτωση των αλγορίθμων σε άλλες εφαρμογές με απλή χρήση ενός αρχείου εισόδου XML.

10.2 Μοντελοποίηση και Μελέτη Επίδοσης

Η διεξαγωγή μελέτης που αφορά την ανάλυση επίδοσης ενός υπολογιστικού συστήματος αφενός βασίζεται στην εμπειρία και γνώση του μελετητή και αφετέρου στην ορθή εφαρμογή μιας γενικής μεθοδολογίας. Σύμφωνα με όσα έχουν αναφερθεί ως τώρα, η ανάπτυξη μοντέλων αποτελεί μια απλή και αποδοτική επιλογή



Σχήμα 10.6: *JMT — Γραφική παράσταση αποτελεσμάτων.*

στις περισσότερες περιπτώσεις. Η επιτυχία της μοντελοποίησης οφείλεται κατά κύριο λόγο στο γεγονός ότι οι λεπτομέρειες χαμηλού επιπέδου ελάχιστα επηρεάζουν την επίδοση ενός συστήματος σε υψηλό επίπεδο. Για τον λόγο αυτό, είναι φυσικό να ακολουθήσει κανείς μια διαδικασία ανάλυσης από πάνω προς τα κάτω (top-down), δηλαδή να ξεκινήσει εντοπίζοντας τις βασικές συνιστώσες του συστήματος και να προσθέσει λεπτομέρειες στον βαθμό που αυτό θα κριθεί αναγακαίο. Η αντιμετώπιση αυτή σημαίνει ότι η μελέτη θα βασιστεί σε ένα σύνολο (κατά το δυνατό δικαιολογημένων) υποθέσεων [9, 12, 13, 7].

Η διατύπωση και επιβεβαίωση των κατάλληλων υποθέσεων είναι σημαντικός παράγων επιτυχίας μιας μελέτης. Η υιοθέτηση απλοποιητικών υποθέσεων προϋποθέτει ότι είναι δυνατός ο εντοπισμός χαρακτηριστικών του συστήματος των οποίων η επίδραση κρίνεται δευτερεύουσας σημασίας. Για παράδειγμα, έστω ότι το φορτίο ενός συστήματος αποτελείται από πολλές συνιστώσες, αλλά ενδιαφερόμαστε για μία μόνο από αυτές. Μπορούμε, στην περίπτωση αυτή, να θεωρήσουμε μοντέλο δύο κατηγοριών, περιλαμβάνοντας στη δεύτερη όλες τις λοιπές συνιστώσες που κρίνονται ως δευτερεύοντος ενδιαφέροντος. Συχνά, οι υποθέσεις είναι αναγκαίες προκειμένου να είναι εφικτή η κατασκευή και αποτίμηση του μοντέλου. Π.χ. προκειμένου να χρησιμοποιηθεί ένα αναλυτικό μοντέλο που βασίζεται στην υπόθεση εκθετικών χρόνων μεταξύ αφίξεων, θα πρέπει να αγνοηθεί τυχόν εκρηκτική συμπεριφορά των αφίξεων στο πραγματικό σύστημα. Εξάλλου, ακόμη και αν το μοντέλο διαθέτει την απαραίτητη δυνατότητα αναπαράστασης, είναι δυνατό να ελλείπουν οι τιμές ορισμένων παραμέτρων, διότι δεν είναι εφικτή η μέτρηση των καταλληλων ποσοτήτων για την εκτίμησή τους.

Η συνηθέστερη εφαρμογή της μοντελοποίησης αφορά την εκτίμηση της επίδρασης στην επίδοση, που προκαλείται από μεταβολές στη διάταξη ή στο φορτίο ενός (υπαρχοντος) συστήματος. Η κατασκευή του μοντέλου, στο οποίο στηρίζεται μια τέτοια μελέτη, είναι επαναληπτική διαδικασία εξαιτίας των εξαρτήσεων που υπάρχουν ανάμεσα στον ορισμό του μοντέλου, τις μετρήσεις που χρησιμοποιούνται για την παραμετροποίηση του μοντέλου, τις τεχνικές ανάλυσης και τους ειδικούς στόχους της μελέτης. Ο γενικός *κύκλος μοντελοποίησης* περιλαμβάνει 3 φάσεις:

- *Επικύρωση* (validation). Κατασκευάζεται ένα βασικό μοντέλο του υπάρχοντος συστήματος (αναπαράσταση συστατικών του συστήματος και του φορτίου) και επιβεβαιώνεται η επάρκειά του. Οι παράμετροι του μοντέλου προσδιορίζονται από υπάρχοντα δεδομένα μετρήσεων.

- *Προβολή* (projection). Το μοντέλο που ορίστηκε χρησιμοποιείται για την πρόβλεψη της επίδρασης των μεταβολών του συστήματος στην επίδοση.

- *Επαλήθευση* (verification). Η έξοδος (πρόβλεψη) του μοντέλου συγκρίνεται με την πραγματική επίδοση του συστήματος (μετρήσεις).

Τόσο η κάθε φάση χωριστά όσο και ο συνολικός κύκλος μοντελοποίησης μπορούν να υλοποιηθούν επαναληπτικά με στόχο να επιτευχθεί η επιθυμητή πιστότητα αναπαράστασης του πραγματικού συστήματος και του φορτίου από το μοντέλο. Μετά την ολοκλήρωση της κατασκευής του το μοντέλο μπορεί να εφαρμοστεί στην αποτίμηση νέων καταστάσεων για τις οποίες δεν υπάρχει δυνατότητα μετρήσεων (π.χ μετακίνηση του υπάρχοντος φορτίου σε νέο σύστημα ή εφαρμογή νέου φορτίου σε υπάρχον σύστημα). Ο χαρακτηρισμός φορτίου, αν και εισάγει ένα είδος ανακρίβειας και αβεβαιότητας στο μοντέλο, αποτελεί απαραίτητο στοιχείο της μοντελοποίησης, καθόσον παρέχει το φορτίο πάνω στο οποίο θα στηριχτεί η μελέτη επίδοσης.

10.3 Ποιότητα Υπηρεσιών Ιστού

Οι υπηρεσίες Ιστού λειτουργούν σε συστήματα μεγάλων διαστάσεων, αποτελούμενα από χιλιάδες συστατικά υλικού και λογισμικού που αλληλεπιδρούν. Ο Ιστός και το Διαδίκτυο είναι μεγάλα κατανεμημένα συστήματα, τα οποία χαρακτηρίζονται από ιδιαίτερη συμπεριφορά. Η ανάλυση, σχεδίαση και λειτουργία τέτοιων συστημάτων απαιτεί κατάλληλες ποσοτικές μεθόδους, καθόσον είναι διαρκής η ανάγκη για την αντιμετώπιση προβλημάτων παραγωγικής ικανότητας και αποτίμησης της επίδοσης. Κρίσιμο στοιχείο είναι η Ποιότητα Υπηρεσιών, η οποία είναι έννοια ευρύτερη από την επίδοση, εφόσον προϋποθέτει επιπλέον ορθότητα λειτουργίας, διαθεσιμότητα, αξιοπιστία, ασφάλεια και ευχέρεια κλιμάκωσης (επαρκές επίπεδο υπηρεσιών ακόμη και σε περιπτώσεις αιφνίδιας αύξησης του φορτίου) [12, 13].

Τα χαρακτηριστικά των συστημάτων που βασίζονται στον Ιστό πρέπει να λαμβάνονται υπόψη στη διαδικασία της μοντελοποίησης. Ως βασικές ιδιότητές του μπορούν να αναφερθούν:

- Μεγάλη κλίμακα, υψηλή πολυπλοκότητα, υψηλή μεταβλητότητα.
- Υψηλή ετερογένεια (μεγάλος αριθμός υποσυστημάτων και συστατικών διαφορετικών κατασκευαστών και τεχνολογιών).
- Κατανεμημένη λειτουργία και διαχείριση (διασύνδεση και αλληλεπίδραση πολλών «ανεξάρτητων» συστημάτων).
- Συνεχής αύξηση του αριθμού των χρηστών και της ζήτησης των υπηρεσιών.
- Μεγάλη ταχύτητα επέκτασης (δυσκολία διαμόρφωσης ακριβών προβλέψεων).
- Υποχρέωση ικανοποίησης των απαιτήσεων της Ποιότητας Υπηρεσιών (QoS).

Η ανάλυση των υπηρεσιών Ιστού πραγματοποιείται μέσω μελέτης του συστήματος, η οποία ακολουθεί τις γενικές κατευθύνσεις υλοποίησης. Οι κυριότερες τεχνικές βασίζονται σε μοντέλα του φορτίου και της συμπεριφοράς του συστήματος, τα οποία λαμβάνουν υπόψη τις ιδιαιτερότητες των συστημάτων Ιστού (εκρηκτικότητα φορτίου, κατανομές βαριάς ουράς, φορτίο πολλών κατηγοριών). Τα μοντέλα επίδοσης του συστήματος υπολογίζουν τον χρόνο απόκρισης (που περιλαμβάνει τον χρόνο εξυπηρέτησης), και τους λοιπούς δείκτες επίδοσης, συναρτήσει των παραμέτρων του φορτίου. Το φορτίο περιγράφεται με την ένταση φορτίου και τους χρόνους εξυπηρέτησης, οι οποίοι προκύπτουν από μετρήσεις ή υπολογίζονται συναρτήσει χαρακτηριστικών κατασκευής και λειτουργίας των αντίστοιχων συσκευών. Είναι σημαντικό ότι η διαδικασία αυτή γενικεύεται εύκολα, καθόσον μπορεί να ενσωματώσει νέα χαρακτηριστικά με κατάλληλη προσαρμογή των παραμέτρων του φορτίου.

Το μοντέλο του συστήματος μπορεί να μελετηθεί με χρήση προσομοίωσης ή αναλυτικής επίλυσης (με ακριβή ή προσεγγιστική μέθοδο). Όπως είδαμε, τα αναλυτικά μοντέλα αναμονής μπορεί να αναφέρονται σε επίπεδο συστήματος (απλές ουρές αναμονής) ή σε επίπεδο συνιστωσών (δίκτυα αναμονής). Τα μοντέλα δικτύων αναμονής έχουν τη δυνατότητα αναπαράστασης του φορτίου και της λειτουργίας των συστημάτων Ιστού. Τα ανοικτά δίκτυα προσφέρονται για τη μοντελοποίηση από την πλευρά του εξυπηρετητή (δημόσιοι ιστότοποι στο Διαδίκτυο), ενώ τα κλειστά μοντέλα μπορούν να περιγράψουν την οπτική γωνία των χρηστών

(πελατών) που συνδέονται με εξωτερικούς εξυπηρετητές ή ανήκουν σε εταιρικό δίκτυο. Με τη χρήση των μοντέλων αναμονής μπορούν να απαντηθούν ικανοποιητικά τα περισσότερα από τα συνήθη ερωτήματα που πρέπει να αντιμετωπίσει ο διαχειριστής των υπηρεσιών Ιστού προκειμένου να είναι συνεπής προς τη ζητούμενη Ποιότητα Υπηρεσιών και να τηρεί τη Συμφωνία Επιπέδου Υπηρεσιών απέναντι στους χρήστες.

Τα σύγχρονα υπολογιστικά συστήματα έχουν περάσει στην εποχή των πολύπλοκων δυναμικά εξελισσόμενων δικτύων, που περιλαμβάνουν εκατομμύρια μικρών και μεγάλων συσκευών και παροχή υπηρεσιών πολύ μεγάλης κλίμακας. Οι μεθοδολογίες της ανάλυσης επίδοσης θα πρέπει να ανταποκριθούν στην πρόκληση, ώστε να είναι σε θέση να υποστηρίξουν την εξασφάλιση της Ποιότητας Υπηρεσιών στις «υπηρεσίες Ιστού» του μέλλοντος. Βέβαια, ο Ιστός –στο παρόν και στο μέλλον– καλύπτει τεράστιο φάσμα συστημάτων και εφαρμογών που δεν έρχονται σε άμεση επαφή με τον χρήστη των υπηρεσιών αλλά η επίδοσή τους είναι κρίσιμης σημασίας (όπως πολυεπεξεργαστικά συστήματα υψηλής παραλληλίας, τηλεπικοινωνιακές δομές αιχμής ή εξειδικευμένα συστήματα παραγωγής). Σε κάθε περίπτωση, η ανάγκη επίλυσης δύσκολων προβλημάτων επίδοσης σε μικρή ή μεγάλη κλίμακα οδηγεί στην πολλαπλή αξιοποίηση της υπάρχουσας τεχνογνωσίας και στην αναζήτηση νέων δυνατοτήτων μέσα από τη θεωρία και την εφαρμογή.

Βιβλιογραφία

[1] Bazan, P., Bolch, G. and German, R., *WinPEPSY-QNS — Performance Evaluation and Prediction System for Queueing Networks*, Proceedings of the 11th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA'04), Magdeburg, Germany, (SCS-European Publishing House), pp.147–150, June 2004.

[2] Bertoli, M., Casale, G., and Serazzi, G., *JMT: performance engineering tools for system modeling*, ACM SIGMETRICS Performance Evaluation Review, Vol. 36, No. 4, pp. 10–15, New York, US, March 2009.

[3] Bertoli, M., Casale, G., and Serazzi, G., *User-Friendly Approach to Capacity Planning Studies with Java Modelling Tools*, International ICST Conference on Simulation Tools and Techniques, SIMUTools 2009, Rome, Italy, 2009.

[4] Bolch, G. and Kirschnick, M., *PEPSY-QNS — Performance Evaluation and Prediction System for Queueing Networks*, Technical Report TR-I4-21-92, Universität Erlangen-Nürnberg, Oct. 1992.

[5] Bolch, G., Greiner, S., De Meer, H., and Trivedi, K.S., *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, Wiley-Interscience, 2006.

[6] Chung, J. and Claypool, M., *NS by Example*, Worcester Polytechnic Institute, http://nile.wpi.edu/NS/, 2002.

[7] Fortier, P.J., and Michel, H.E., *Computer Systems Performance Evaluation and Prediction*, Elsevier Science, 2003.

[8] Issariyakul, T., and Hossain, E. *Introduction to Network Simulator NS2*, Second Edition, Springer, 2011.

[9] Lazowska, E.D., Zahorjan, J., Scott Graham, G. and Sevcik, K.C., *Quantitative System Performance - Computer System Analysis Using Queueing Network Models*, Prentice-Hall, 1984.

[10] Marie, R., *An Approximate Analytical Method for General Queueing Networks*, IEEE Transactions on Software Engineering, Vol. 5, No. 5, pp. 530–538, Sept. 1979.

[11] Marie, R., *Calculating Equilibrium Probabilities for* $\lambda(n)/C_k/1/N$ *Queues*, ACM Sigmetrics Performance Evaluation Review, Vol. 9, No. 2, pp.117–125, 1980.

[12] Menasce, D.A., and Almeida, V.A.F., *Capacity Planning for Web Services: Metrics, Models and Methods*, Prentice-Hall, 2002.

[13] Menasce, D.A., Almeida, V.A.F., and Dowdy, L.W., *Performance by Design, Computer Capacity Planning by Example*, Prentice-Hall PTR, 2004.

[14] Obaidat, M.S., and Boudriga, N.A., *Performance Evaluation of Computer and Telecommunication Systems*, Wiley, 2010.

[15] Serazzi, G., Ed., *Performance Evaluation Modelling with JMT: learning by examples*, Politecnico di Milano - DEI, TR 2008.09, June 2008.

Ευρετήριο