



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

ΔΙΚΤΥΑ ΔΗΜΟΣΙΑΣ ΧΡΗΣΗΣ ΚΑΙ ΔΙΑΣΥΝΔΕΣΗ ΔΙΚΤΥΩΝ

Ενότητα #12: Σχεδίαση πρωτοκόλλων

Καθηγητής Χρήστος Ι. Μπούρας

Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο
Πατρών

email: bouras@cti.gr, site: <http://ru6.cti.gr/ru6/bouras>

Σκοποί ενότητας

- Εξοικείωση με τη σχεδίαση πρωτοκόλλων



Περιεχόμενα ενότητας

- Εισαγωγή
- Στοιχεία πρωτοκόλλου
- Στόχοι
- Σχεδιαστικές αρχές
- Μηχανισμοί ελέγχου
- Προσδιορισμός προδιαγραφών (Specification)
- Ανάπτυξη



Σχεδίαση πρωτοκόλλων

Εισαγωγή

- Τα πρώτα δίκτυα υπολογιστών προορίζονταν για να προσφέρουν περιορισμένες υπηρεσίες και να επιτυγχάνουν συνδέσεις σε μικρή κλίμακα
- Σε αυτό συνέτεινε το μεγάλο κόστος των υπολογιστών και η ακριβή ανάπτυξη συστημάτων δικτύων
- Η έλλειψη κάποιων standards έκανε απαγορευτική την επέκταση τέτοιων συστημάτων προκειμένου να συνδεθούν με άλλα διαφορετικά συστήματα



Τι είναι πρωτόκολλο

- Προκειμένου να διασυνδέονται πολλά ετερογενή περιβάλλοντα πάνω από ένα κοινό μέσο, έγινε απαραίτητη η σύσταση standards που ορίζουν ακριβώς τον τρόπο σύνδεσης μεταξύ συνδρομητών και δικτύου
- Δηλαδή ένα σύνολο κανόνων που είναι συμφωνημένοι και από τα δυο μέρη και ελέγχει ή ενεργοποιεί τη σύνδεση, την επικοινωνία και τη μεταφορά δεδομένων μεταξύ τους
- Αυτό είναι ένα πρωτόκολλο επικοινωνίας και στην πιο απλή του μορφή, μπορεί να οριστεί σαν οι κανόνες που διέπουν το συντακτικό, τη σημασιολογία και το συγχρονισμό της επικοινωνίας



Πρωτόκολλο επικοινωνίας

- Ορίζει την ακριβή μορφοποίηση των μηνυμάτων που ανταλλάσσονται (συντακτικό)
- Ορίζει τους διαδικαστικούς κανόνες για την ανταλλαγή των δεδομένων (γραμματική)
- Ορίζει το λεξιλόγιο των μηνυμάτων που μπορούν να ανταλλαγούν και τη σημασία τους (σημασιολογικά)



Σχεδίαση πρωτοκόλλων

- Νέες εφαρμογές εμφανίζονται συνεχώς και όλο και περισσότερες από αυτές σχετίζονται με δίκτυα
- Δημιουργούνται συνεχώς νέες απαιτήσεις εξαιτίας της εξέλιξης της τεχνολογίας
 - Π.χ. ανάγκη για περισσότερες IP διευθύνσεις και η δημιουργία του IPv6 πρωτοκόλλου
- Η αποσύνθεση των εφαρμογών σε λειτουργικά τμήματα και η διανομή αυτών των τμημάτων κάνει τη σχεδίαση πρωτοκόλλων ένα αναπόσπαστο κομμάτι της σχεδίασης του συστήματος



Προϋποθέσεις Σχεδίασης Πρωτοκόλλων

- Προϋποθέσεις πριν τη σχεδίαση:
 - η κατανόηση του προβλήματος
 - η κατανόηση των απαιτήσεων (λειτουργικές και μη),
 - η κατανόηση των περιορισμών
 - η λήψη αποφάσεων σχετικά με τις θυσίες σε κάποιους τομείς για καλύτερη απόδοση σε άλλους
 - η προσπάθεια για γενίκευση του σχεδιασμού
 - η απόφαση της χρήσης της υπάρχουσας τεχνολογίας ή δημιουργία νέας από την αρχή



Στόχοι

- Στόχοι της σχεδίασης ενός πρωτοκόλλου είναι:
 - Ευελιξία
 - Επεκτασιμότητα
 - Αποτελεσματικότητα
 - Κάλυψη των αναγκών για τους οποίους σχεδιάστηκε



Παραδοχές

- Το πρωτόκολλο περιλαμβάνει παραδοχές και συμφωνίες για τις μεθόδους που θα χρησιμοποιηθούν για:
 - Έναρξη και λήξη της ανταλλαγής δεδομένων
 - Συγχρονισμό των αποστολών και των παραληπτών
 - Ανίχνευση και διόρθωση των λαθών μετάδοσης
 - Διαμόρφωση και κωδικοποίηση των δεδομένων

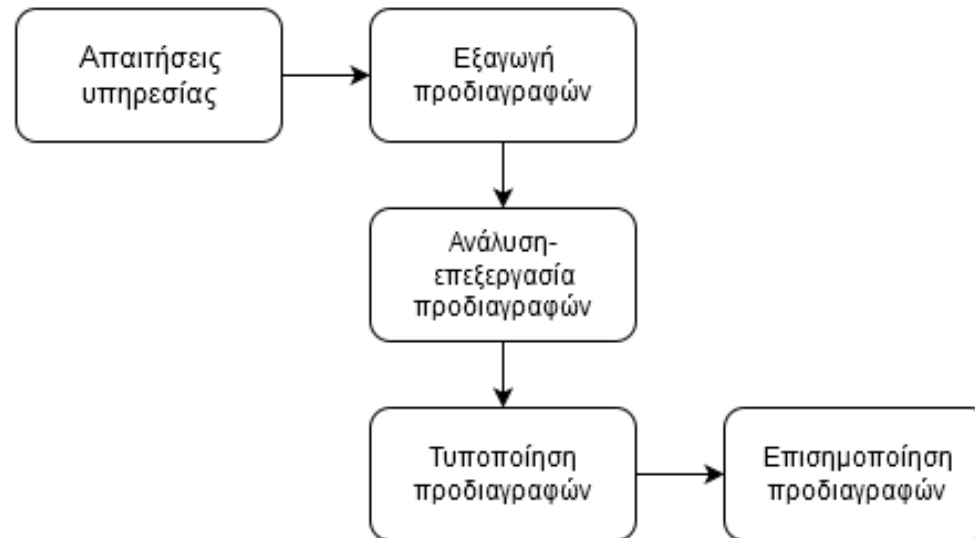


Στοιχεία πρωτοκόλλου

- Προδιαγραφή της υπηρεσίας που θα παρέχεται από το πρωτόκολλο
- Οι παραδοχές / υποθέσεις για το περιβάλλον λειτουργίας
- Το λεξιλόγιο των μηνυμάτων που χρησιμοποιούνται για την υλοποίηση του πρωτοκόλλου
- Η κωδικοποίηση κάθε μηνύματος
- Οι διαδικαστικοί κανόνες για τη διατήρηση της συνέπειας της ανταλλαγής μηνυμάτων
 - Συνιστούν το δυσκολότερο τμήμα ως προς τη σχεδίαση και την επαλήθευση



Στάδια δημιουργίας προδιαγραφών



Στάδια δημιουργίας προδιαγραφών

Παράδειγμα (1/3)

- Προδιαγραφή Υπηρεσίας
 - Μεταφορά ASCII αρχείων σαν σειρά χαρακτήρων
 - Προστασία για λάθη μετάδοσης
 - Αμφίδρομη επικοινωνία
 - Ύπαρξη θετικών και αρνητικών επιβεβαιώσεων (ACK, NACK)
- Υποθέσεις για το περιβάλλον λειτουργίας
 - Δύο χρήστες. Ζητάνε μεταφορά και περιμένουν να ολοκληρωθεί
 - Κανάλι επικοινωνίας. Μπορεί να καταστρέψει μηνύματα αλλά δε χάνει, δεν δημιουργεί αντίγραφα, ή αλλάζει τη σειρά τους



Παράδειγμα (2/3)

- Λεξιλόγιο
 - Τρεις τύποι μηνυμάτων:
 - *ack* μήνυμα με θετική αναγνώριση
 - *nack* μήνυμα με αρνητική αναγνώριση
 - *err* μήνυμα με λάθος μετάδοσης
- Μορφοποίηση Μηνυμάτων
 - *enum* control {ack, nack, err};
 - *struct* message {
 - enum control tag;
 - unsigned char data;
 - };



Παράδειγμα (3/3)

- Διαδικαστικοί κανόνες
 - Αν η προηγούμενη λήψη χωρίς λάθη \implies το επόμενο μήνυμα με ack
 - Αν η προηγούμενη λήψη με λάθος \implies το επόμενο μήνυμα με nack
 - Αν η προηγούμενη λήψη με nack \implies ξαναστείλε το παλιό μήνυμα
 - Αν η προηγούμενη λήψη με ack \implies στείλε ένα καινούργιο μήνυμα



Υπηρεσίες και Περιβάλλον

- Επίπεδο (Layer): ομαδοποίηση παρόμοιων διαδικασιών
- Διασύνδεση (Interface): σύνολο από service access points (σημεία προσπέλασης υπηρεσίας)
- Υπηρεσία: προσφέρεται στο πιο πάνω επίπεδο
- Υποθέσεις: σχετικά με τις υπηρεσίες των κατώτερων επιπέδων



Λεξιλόγιο και Μορφοποίηση (1/2)

- Μέθοδοι μορφοποίησης πρωτοκόλλων:
 - Bit oriented: σειρές από bits
 - 01100110|00110010010100110|10001010
 - Character oriented: πολλαπλά bits όπου αντιπροσωπεύουν έναν χαρακτήρα
 - Byte count oriented: Μετρά τον αριθμό των bytes προς μετάδοση αντί να χρησιμοποιεί χαρακτήρες ελέγχου



Λεξιλόγιο και Μορφοποίηση (2/2)

format = {header, data, trailer}

header = { type, destination, seq no, count}

trailer = {checksum, return address}



Παράδειγμα - Λεξιλόγιο & Μορφοποίηση



Βασικές σχεδιαστικές αρχές

- Αξιοπιστία (Reliability)
- Ανθεκτικότητα (Robustness)
- Ασφάλεια (Security)
- Εξελιξιμότητα (Evolvability)
- Απλότητα (Simplicity)
- Modularity
- Σωστή μορφοποίηση (Well-formed)
- Συνέπεια (Consistency)



Αξιοπιστία

- Ο βασικός στόχος ενός πρωτοκόλλου είναι ο συγχρονισμός (synchronization) της κατάστασης μεταξύ δύο ή περισσότερων κόμβων
- Για να είναι αξιόπιστος ο συγχρονισμός, πρέπει να μπορεί να επιτευχθεί με τον ελάχιστο αριθμό ανταλλαγής μηνυμάτων



Πιθανά προβλήματα αξιοπιστίας

- Πρόβλημα στην επικοινωνία από την επίδραση ενός συνδέσμου (λάθη στα bits, απώλειες πλαισίων, καθυστερήσεις κ.ά.)
- Πρόβλημα σε ένα router (όπως απώλεια πακέτων, καταστροφή πακέτων, καθυστέρηση ενός πακέτου κ.ά)
- Σφάλματα στο δίκτυο (συγκρούσεις, συμφόρηση, routing loops, μη διαθεσιμότητα δικτύου κ.ά)
- Σφάλματα στον παραλήπτη (απώλεια πακέτων εξαιτίας buffer overflow, αποτυχία κάποιας εφαρμογής, δυσλειτουργίες κ.ά).



Συμβιβασμοί

- Πιθανώς να πρέπει να ληφθούν κάποιες αποφάσεις που θα αυξήσουν την αξιοπιστία, αλλά θα μειώσουν την απόδοση σε άλλους τομείς
- Η κυριότερη είναι η επιβάρυνση στην καθυστέρηση προκειμένου να διευθετηθούν παραπάνω διεργασίες που θα απαιτήσουν μεγαλύτερο φόρτο επεξεργασίας
- Οι πιο σημαντικές είναι η κωδικοποίηση και αποκωδικοποίηση της αρχικής πληροφορίας για τον έλεγχο των λαθών και έλεγχοι που επιβάλλουν κάποιες τεχνικές ανίχνευσης ή διόρθωσης λαθών
- Επιπλέον, προκειμένου να γίνει έλεγχος λαθών μεταφέρεται μεγαλύτερο μέγεθος πληροφορίας, επομένως απαιτείται περισσότερος χρόνος για τη μετάδοση



Μηχανισμοί ελέγχου αξιοπιστίας

- Ο μηχανισμός ελέγχου αξιοπιστίας επιλέγεται σε σχέση με
 - την εφαρμογή και τη σημασία της
 - το περιβάλλον λειτουργίας (π.χ. συχνότητα σφαλμάτων)
 - την εγκατάσταση της επικοινωνίας (για παράδειγμα ο αριθμός των παραληπτών)
- Χωρίζονται σε:
 - Μηχανισμούς εύρεσης λαθών (error detection)
 - Μηχανισμούς διόρθωσης λαθών (error correction)



Μηχανισμοί εύρεσης λαθών

- Για να βρούμε λάθη σε επίπεδο bit ή σε επίπεδο πλαισίου (frame) σε ένα πακέτο, οι γνωστότεροι μηχανισμοί χρησιμοποιούν:
 - Checksums,
 - CRCs (Cyclic Redundancy Check Code)
 - MACs (Medium Access Control)
- Για την ανακάλυψη μη ληφθέντων πακέτων χρησιμοποιούνται:
 - Sequence numbers (SN)



Μηχανισμοί διόρθωσης λαθών

- Απλό Lock-Step Πρωτόκολλο (simple Lock-Step protocol)
- Αθροιστικό ACK με Go-back-N (Cumulative ACK with Go-back-N)
- Επιλεκτικά ACKs (Selective Acknowledgements)
- Απλό NACK Πρωτόκολλο (Simple Negative ACK Protocol)
- Forward Error Correction (FEC)



Απλό Lock-Step Πρωτόκολλο

- Τα δεδομένα αποστέλλονται και περιμένουμε για επιβεβαίωση (acknowledgment ή ACK)
- Η κάθε πλευρά λειτουργεί μόνο όταν η άλλη πλευρά έχει ολοκληρώσει τη δράση της
- Εάν ο αποστολέας δεν λάβει βεβαίωση μέσα σε ένα ορισμένο χρονικό διάστημα (timeout) αναμεταδίδει το τελευταίο πακέτο (retransmission)
- Απλό αλλά και πολύ περιοριστικό
- Παράδειγμα εφαρμογής: Trivial File Transfer Protocol (TFTP)



Αθροιστικό ACK με Go-back-N (1/2)

- Μηχανισμός βασισμένος σε παράθυρα που επιτρέπει τη μετάδοση πολλών πλαισίων
- Είναι μια ειδική περίπτωση του Sliding Window Protocol με το μέγεθος του παραθύρου μετάδοσης να είναι N και του παραθύρου λήψης 1
- Ο αποστολέας συνεχίζει να στέλνει έναν αριθμό πλαισίων που καθορίζονται από το μέγεθος του παραθύρου, ακόμη και χωρίς να λάβει επιβεβαίωση (ACK) πακέτου από το δέκτη



Αθροιστικό ACK με Go-back-N (2/2)

- Τα πλαίσια είναι αριθμημένα
- Όταν ο δέκτης αναγνωρίζει ότι έλαβε σωστά ένα πλαίσιο με κάποιο αριθμό, αυτόματα σημαίνει ότι και όλα τα πλαίσια με μικρότερο αριθμό ελήφθησαν σωστά
- Μειονέκτημα: Δεν επιτρέπει το acknowledgment οποιουδήποτε $n+k$ πακέτου, αν το n δεν έφτασε σωστά
- Εφαρμογή: TCP



Επιλεκτικά ACKs

- Βασισμένη σε παράθυρο αλλά με ρητή επιβεβαίωση των ληφθέντων πακέτων
- Ο αποστολέας αναφέρει ρητά για ποια πακέτα πρέπει να επιβεβαιώσει ο δέκτης με acknowledgment (είτε αρνητικά είτε θετικά)
- Εφαρμογή: η προσέγγιση με επιλεκτικά θετικά ACKs μπορεί να επιλεχθεί στο TCP και είναι χρήσιμο στη διαδικτυακή πρόσβαση μέσω δορυφόρου



Απλό NACK Πρωτόκολλο

- Αισιόδοξη υπόθεση: τα πακέτα θα φθάσουν, άρα αναφέρουμε μόνο τις αποτυχίες
- Ο αποστολέας στέλνει επιβεβαίωση (acknowledgment) μόνο για πακέτα που ελήφθησαν λανθασμένα (Negative ACK)
- Εφαρμογή: Προτιμάται σε reliable multicast πρωτόκολλα



Forward Error Correction - FEC (1/2)

- Βασική υπόθεση: θα συμβούν λάθη
- Στο FEC ο αποστολέας κωδικοποιεί το μήνυμα με πλεονάζουσα πληροφορία (overhead) χρησιμοποιώντας κώδικα διόρθωσης σφαλμάτων
- Η πληροφορία επιτρέπει στον δέκτη να ανιχνεύσει έναν περιορισμένο αριθμό λαθών στο μήνυμα, και συχνά να διορθώσει αυτά τα λάθη, χωρίς αναμετάδοση



Forward Error Correction - FEC (2/2)

- Μειονέκτημα: Αυξάνονται οι απαιτήσεις σε bandwidth λόγω αποστολής πλεονασματικής πληροφορίας (αύξηση overhead)
- Πλεονέκτημα: Μειώνεται σημαντικά η καθυστέρηση αφού δε χρειάζεται να περιμένουμε για NACK ή timeout
- Εφαρμογή: σε μεταδόσεις multicast ή σε εφαρμογές όπου η καθυστέρηση είναι κρίσιμος παράγοντας



Χαλάρωση των απαιτήσεων

- Προκειμένου να μειωθούν οι απαιτήσεις αξιοπιστίας προς όφελος της απόδοσης σε άλλους τομείς, πιθανές ενέργειες περιλαμβάνουν τη χαλάρωση των απαιτήσεων σε μη κρίσιμους τομείς, όπως:
 - Δεν είναι απαραίτητο για όλους τους κόμβους να παίρνουν όλες τις πληροφορίες
 - Παροχή CRC και/ή FEC μόνο στα σημαντικά τμήματα του πακέτου
 - Χαλάρωση σε θέματα που αφορούν την ακολουθιακή μετάδοση των δεδομένων και στις συγκρούσεις



Ανθεκτικότητα

- Το πρωτόκολλο πρέπει να εξασφαλίζει τη σωστή εκτέλεσή του κάτω από όλες τις πιθανές συνθήκες
- Ο σχεδιασμός του οφείλει να γίνει:
 - Θεωρώντας τις ελάχιστες γνώσεις σχετικά με το περιβάλλον
 - Αποφεύγοντας εξαρτήσεις από άλλα στοιχεία του πρωτοκόλλου, παραμέτρους συστήματος κλπ.



Evolvability

- Εξελικτικότητα είναι η ικανότητα να εξελίσσεσαι εύκολα και χωρίς σημαντικά προβλήματα είτε προσαρμοζόμενος στο εξελισσόμενο περιβάλλον, είτε αφομοιώνοντας μη προβλέψιμες απαιτήσεις
- Η σχεδίαση πρωτοκόλλων είναι καταδικασμένη σε αρκετά λάθη αφού η βελτιστοποίηση γίνεται με βάση το άγνωστο μέλλον και τα πρωτόκολλα χρειάζονται μερικά χρόνια μέχρι να φθάσουν στην αγορά
 - Παράδειγμα: το IP multicast στο IP, απέτυχε μεν σε παγκόσμιο επίπεδο ανάπτυξης, δουλεύει δε σε εταιρικά δίκτυα ή σε ειδικά περιβάλλοντα (π.χ. ακαδημαϊκό)



Απλότητα

- Ένα καλά δομημένο πρωτόκολλο πρέπει να μπορεί να κατασκευαστεί από ένα μικρό αριθμό καλά σχεδιασμένων και καλά ορισμένων μικρότερων μερών
- Για την κατανόηση του πρωτοκόλλου θα πρέπει να αρκεί η κατανόηση των μερών από τα οποία είναι κατασκευασμένο και του τρόπου με τον οποίο αυτά αλληλεπιδρούν
- Τα παραπάνω ενσωματώνουν και τον σχεδιαστικό κανόνα για modularity

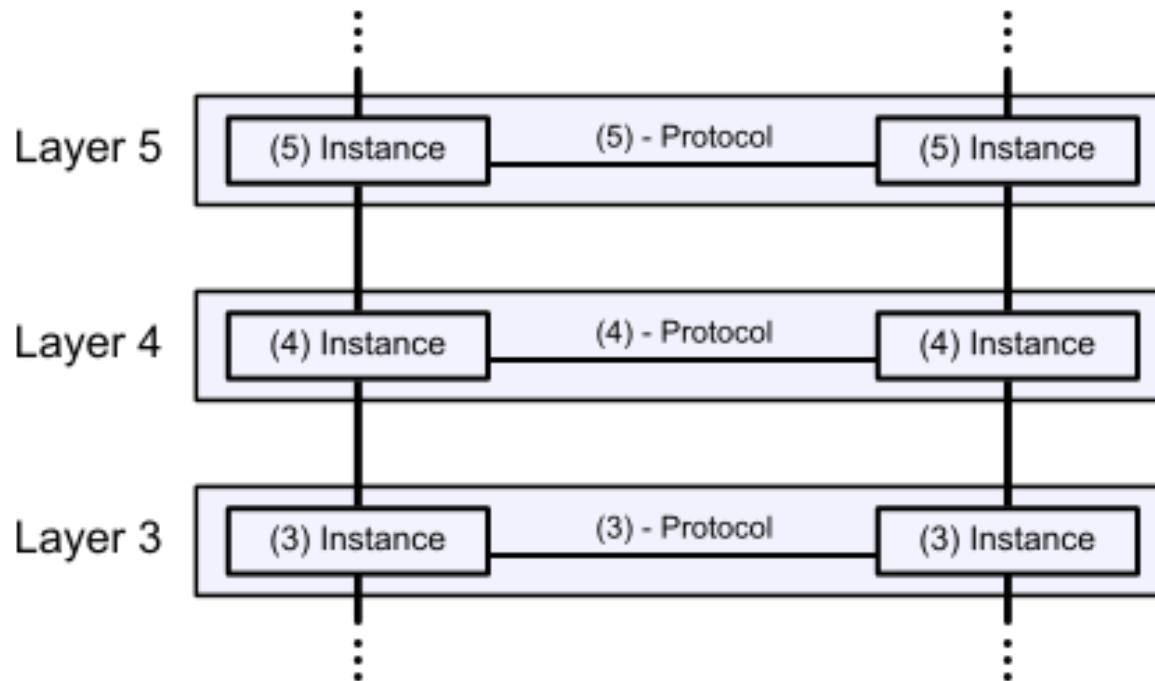


Modularity

- Modularity σημαίνει ότι ένα πρωτόκολλο που εκτελεί μια σύνθετη λειτουργία μπορεί να κατασκευαστεί από αλληλεπιδρώντα μικρότερα κομμάτια
- Κάθε κομμάτι μπορεί να θεωρηθεί ένα ελαφρύτερο πρωτόκολλο που μπορεί αυτόνομα να αναπτυχθεί, και να υλοποιηθεί, και να σχεδιαστεί σαν ανεξάρτητη οντότητα
- Το modularity υποδηλώνει και τη διαστρωμάτωση του πρωτοκόλλου η οποία:
 - Καθορίζει τα επίπεδα της αφαίρεσης
 - Κατηγοριοποιεί σχετικές μεταξύ τους λειτουργίες
 - Ορίζει μικρές και σαφώς καθορισμένες διεπαφές

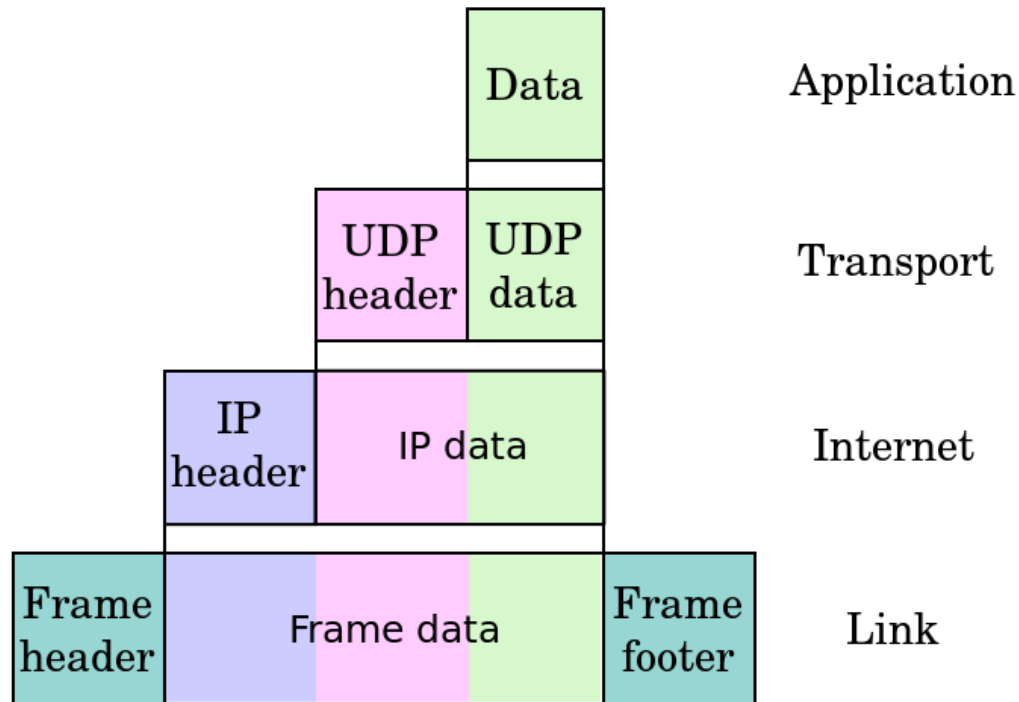


Επίπεδα διαστρωμάτωσης



Επικοινωνία μεταξύ επιπέδων διαστρωμάτωσης
(πηγή: https://en.wikipedia.org/wiki/OSI_model#/media/File:OSI-model-Communication.svg)

Ενθυλάκωση



Η ενθυλάκωση είναι συνέπεια της διαστρωμάτωσης
(πηγή:
[https://commons.wikimedia.org/wiki/File:UDP_encapsulation.s
vg](https://commons.wikimedia.org/wiki/File:UDP_encapsulation.svg))



Σωστή μορφοποίηση

- Ένα καλοσχηματισμένο πρωτόκολλο δεν είναι υπερβολικά ή ελλιπώς προσδιορισμένο, δηλαδή δεν περιέχει αχρησιμοποίητο ή ελλιπή κώδικα αντίστοιχα
- Έχει καλώς ορισμένα όρια και δεν μπορεί να ξεπεράσει τα όρια του συστήματος, όπως περιορισμένη χωρητικότητα στις ουρές μηνυμάτων
- Έχει διαδικασίες για αυτό-σταθεροποίηση. Εάν ένα παροδικό σφάλμα αλλάξει αυθαίρετα την κατάσταση του πρωτοκόλλου, αυτό επιστρέφει σε μια επιθυμητή κατάσταση μέσα σε ένα πεπερασμένο αριθμό μεταβάσεων
- Είναι προσαρμόσιμο. Μπορεί για παράδειγμα να προσαρμόσει τον ρυθμό που τα δεδομένα αποστέλλονται στο ρυθμό με τον οποίο ο δέκτης μπορεί να τα αξιοποιεί



ΣΥΝΕΠΕΙΑ

- Είναι η ικανότητα του πρωτοκόλλου απέναντι σε αποτυχίες όπως:
 - Αδιέξοδα: καταστάσεις στις οποίες καμία περαιτέρω εκτέλεση δεν είναι δυνατή
 - Π.χ. επειδή οι διαδικασίες περιμένουν συνθήκες που δεν μπορούν ποτέ να εκπληρωθούν
 - Livelocks: ακολουθίες που μπορεί να επαναλαμβάνονται επ' αόριστον χωρίς ουσιαστική πρόοδο
 - Άστοχοι τερματισμοί: η ολοκλήρωση της εκτέλεσης του πρωτοκόλλου χωρίς να πληρούνται οι κατάλληλες συνθήκες τερματισμού



Specification

- Ο προσδιορισμός των προδιαγραφών (specification) γίνεται κυρίως με τους ακόλουθους τρόπους:
 - State machines
 - Specification and Description Languages

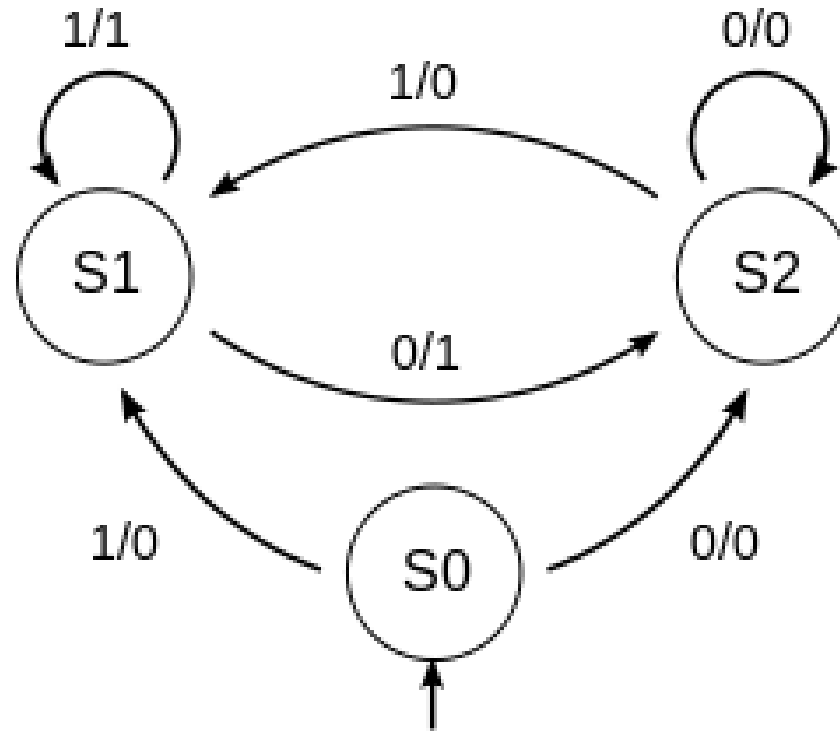


State machines

- Ένα πρωτόκολλο αλληλεπιδρά με το περιβάλλον
 - Ενεργοποιείται από τα γεγονότα
 - Αποκρίνεται με ενέργειες
 - Η συμπεριφορά εξαρτάται από το παρελθόν
- Χρήση Finite State Machines (FSM)
 - Moore μηχανές (η έξοδος καθορίζεται μόνο από την τρέχουσα κατάσταση)
 - Mealy μηχανές (η έξοδος καθορίζεται και από την είσοδο)



Mealy μηχανή



Παράδειγμα Mealy μηχανής

(πηγή: https://en.wikibooks.org/wiki/A-level_Computing/AQA/Problem_Solving,_Programming,_Data_Representation_and_Practical_Exercise/Problem_Solving/Finite_state_machines#/media/File:CPT-FSM-Mealy-01.svg)



Finite States Machines

- Ορισμός: $M = (K, \Sigma, g, s, F)$
 - K = πεπερασμένο σύνολο καταστάσεων
 - Σ = σύνολο σημάτων εισόδου
 - F (υποσύνολο του K) = τελικές καταστάσεις
 - s = Αρχική κατάσταση
 - g = συνάρτηση μετάβασης
- Αν $\sigma \in \Sigma$ & $q \in K \implies g(q, \sigma) \in K$



Παράδειγμα (1/3)

- 1. Αποστολέας: στέλνει πακέτο στον παραλήπτη, και κρατάει αντίγραφο
- 2. Παραλήπτης: ειδοποιεί το αποστολέα αν το πακέτο έφτασε σωστά ή όχι
- 3. Τα πακέτα αν έχουν καταστραφεί αγνοούνται
- 4. Ο αποστολέας χρησιμοποιεί timer

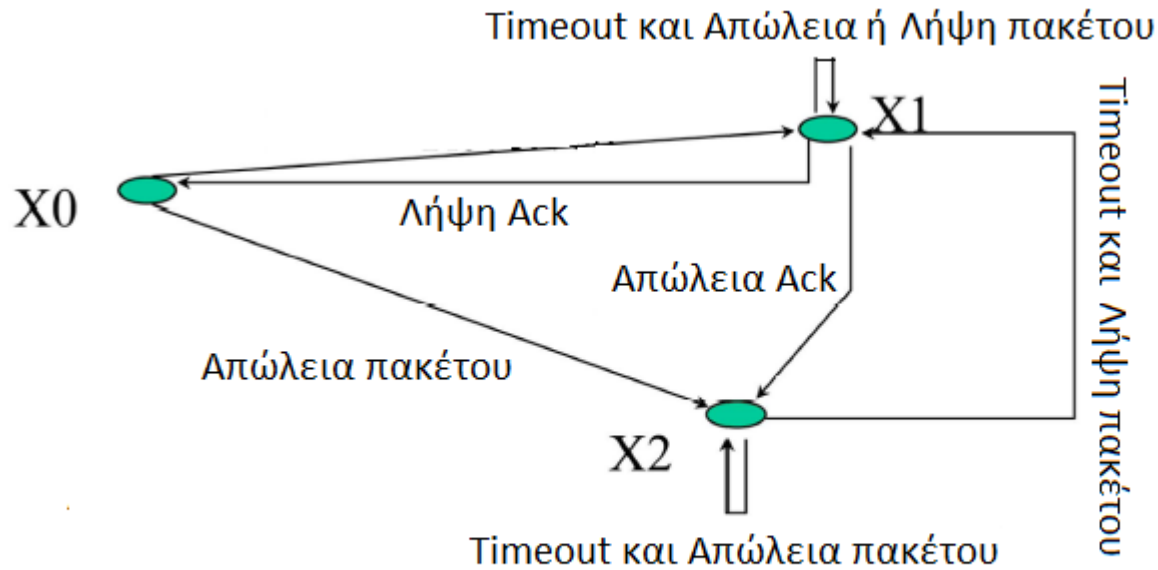


Παράδειγμα (2/3)

- Καταστάσεις διεργασιών αποστολέα:
 - AE = Ack Expected
 - NAE = No Ack Expected
- Καταστάσεις διεργασιών παραλήπτη
 - MA = Must ack
 - NA = No ack
- Καταστάσεις συστήματος: $\chi = (s, r)$
 - $X0 = \{NAE, NA\}$ <--- αρχική κατάσταση
 - $X1 = \{AE, MA\}$
 - $X2 = \{AE, NA\}$



Παράδειγμα (3/3)



source: <http://eclass.uoa.gr/modules/document/document.php?course=D210&openDir=/4c2b2de8icb8>

Μηχανή πεπερασμένων καταστάσεων



Περιορισμοί

- Περιορισμοί FSMs
 - Δεν υποστηρίζουν μεταβλητές
 - Πεπερασμένη μνήμη
 - Δεν έχουν τη δυνατότητα αφαίρεσης (abstraction)
- Εναλλακτική:
 - Extended FSMs
 - Description Languages



Specification and Description Language (SDL)

- SDL στοιχεία:
 - Οι SDLs περιγράφουν παράλληλες διεργασίες και την αλληλεπίδρασή τους επεκτείνοντας τις extended FSMs
 - Παρέχει σημειογραφία κειμένου και γραφικών
- Μια SDL προδιαγραφή ενός συστήματος περιγράφει:
 - τη Δομή
 - την Επικοινωνία
 - τη Συμπεριφορά
 - τα Δεδομένα

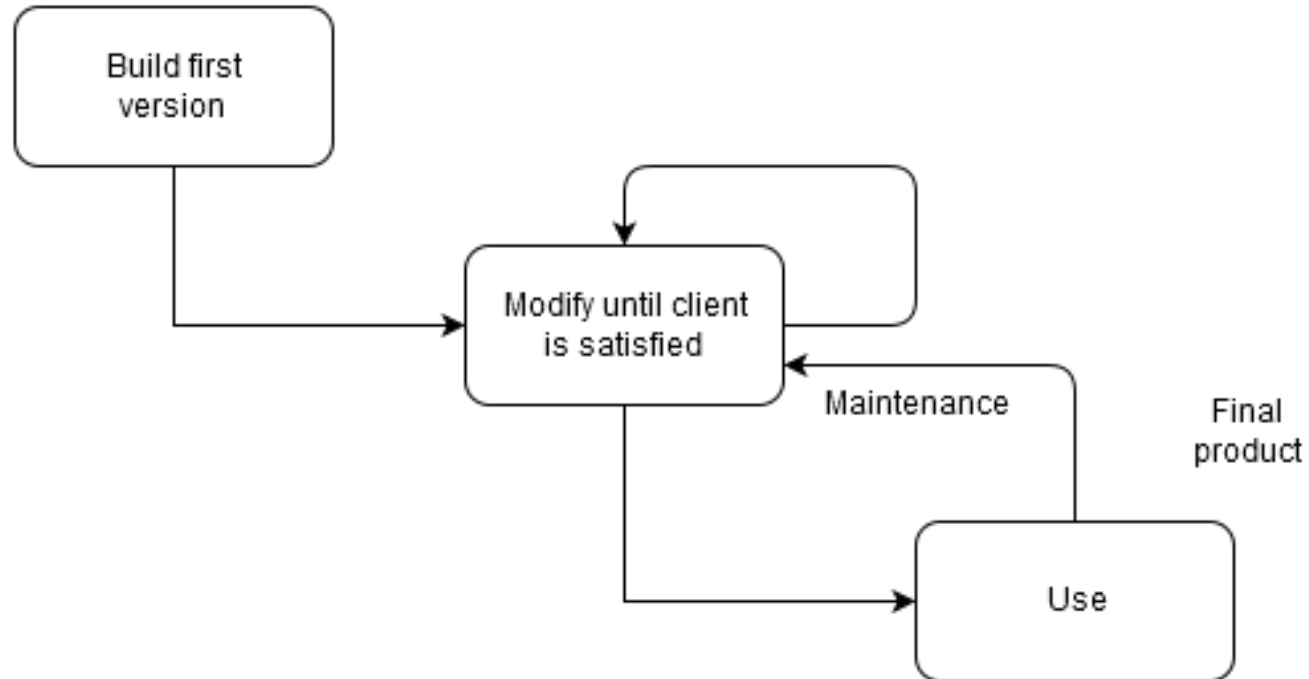


Ανάπτυξη

- Η διαδικασία ανάπτυξης οφείλει να:
 - Δομηθεί σε προσεχτικά προγραμματισμένα στάδια
 - Να εμπεριέχει καλώς ορισμένες μεταβάσεις
- Παραδείγματα μοντέλων ανάπτυξης:
 - Build-and-Fix Model
 - Waterfall Model
 - Boehm's Spiral Model
 - Κλπ.



Build-and-Fix Model



Build-and-Fix Model

Στάδια ανάπτυξης

- Ανάλυση απαιτήσεων
- Προδιαγραφές σχεδιασμού
- Επικύρωση (Validation)
- Υλοποίηση
- Έλεγχος και αξιολόγηση
- Εφαρμογή
- Συντήρηση



Σύντομη ανασκόπηση

- Εισαγωγή
- Στοιχεία πρωτοκόλλου
- Στόχοι
- Σχεδιαστικές αρχές
- Μηχανισμοί ελέγχου
- Προσδιορισμός προδιαγραφών (Specification)
- Ανάπτυξη



Βιβλιογραφία

- Σημειώσεις μαθήματος (Κεφάλαιο 11)
- Βιβλία:
 - Principles of Protocol Design, Robin Sharp
 - Elements of Network Protocol Design, Mohamed G. Gouda
- Links:
 - <http://ru6.cti.gr/ru6/bouras/undergraduate-courses/diktua-dhmosias-xrhshs-kai-diasundesh-diktuwn?language=el> Δικτ. τόπος μαθήματος
 - <http://pet.ece.iisc.ernet.in/course/E2223/Cha1.pdf> Basics of Communication Protocol Presentation
 - <http://archive.cone.informatik.uni-freiburg.de/teaching/lecture/protocol-design-s09/slides/01-Introduction.pdf> Presentation on Network Protocol Design and Evaluation



Ερωτήσεις



Τέλος Ενότητας



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση **2.0**.



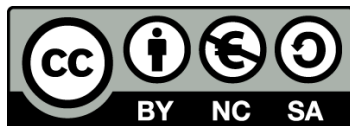
Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Χρήστος Μπούρας 2014. «Δίκτυα δημόσιας χρήσης και διασύνδεση δικτύων. Σχεδίαση πρωτοκόλλων». Έκδοση: 2.0. Πάτρα 2017. Διαθέσιμο από τη δικτυακή διεύθυνση:
<https://eclass.upatras.gr/courses/CEID1064/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.