

# Προγραμματισμός Δικτύων – Ε-01

## 8η Διάλεξη

Διδάσκων: Νίκος Ντάρμος

<ntarmos@cs.uoi.gr>  
[<http://www.cs.uoi.gr/~ntarmos/Courses/NetworkProgramming/>]

Τμήμα Πληροφορικής  
Πανεπιστήμιο Ιωαννίνων



- Διαχείριση λαθών.
- Πύλες και Σήραγγες.



## Διαχείριση λαθών



## Ανεξαρτήτως πρωτοκόλλου



## Διαχείριση λαθών

- Το Διαδίκτυο είναι κατά βάση μία υποδομή «best effort».
  - Συνδέσεις μπορεί να διακοπούν.
  - Κόμβοι μπορεί να καταρρεύσουν
  - Πελάτες/εξυπηρετητές μπορεί να αποχωρήσουν χωρίς προειδοποίηση.
  - Η διαδρομή ανάμεσα σε δύο οποιουσδήποτε απομακρυσμένους κόμβους μπορεί να αλλάξει ανά πάσα στιγμή.
  - Τα χαρακτηριστικά τις διαδρομής (εύρος ζώνης, φόρτος, κτλ.) αλλάζουν ανά πάσα στιγμή.
  - Υπάρχουν πάντα και εναλλακτικές διαδρομές...
- Το πρωτόκολλο UDP προσφέρει κι αυτό best effort υπηρεσίες επικοινωνίας.
- Τα πρωτόκολλα TCP και SCTP προσφέρουν «αξιόπιστη» επικοινωνία, αλλά και πάλι υπάρχει η πιθανότητα σφαλμάτων.
- Σφάλματα σε επίπεδο δικτύου αναφέρονται από το πρωτόκολλο ICMP.



## Διαχείριση λαθών

- Σε προγραμματιστικό επίπεδο αντιλαμβανόμαστε τα σφάλματα από:
  - Την **τιμή επιστροφής** των συναρτήσεων που χρησιμοποιούμε.
    - Επιστρέφεται -1 από τις συναρτήσεις και η καθολική μεταβλητή `errno` παρέχει λεπτομέρειες για το σφάλμα.
  - **Σήματα** που παραδίδονται στην διεργασία μας.
    - SIGPIPE: προσπάθεια E/E σε κλειστό socket descriptor.



### Λάθη σε τοπικό επίπεδο

- EBADF/ENOTSOCK: προσπάθεια E/E σε εσφαλμένο descriptor (ανύπαρκτο ή που δεν ανήκει σε socket).
- EINTR: η κλήση διακόπηκε (συνήθως λόγω παράδοσης κάποιου σήματος στη διεργασία).
- ENOBUFS: δεν υπάρχει χώρος στους buffers του πυρήνα για τα δεδομένα.
- EPIPE: προσπάθεια E/E σε κλειστό socket descriptor.
- EFAULT/EINVAL: λάθος δείκτης σε κάποιο από τα ορίσματα των συναρτήσεων.
- EMFILE/ENFILE: αδυναμία δημιουργίας νέου socket λόγω πλήρους πίνακα descriptors διεργασίας/συστήματος.
- EACCES/EPERM: αδυναμία ολοκλήρωσης λειτουργίας λόγω ανεπαρκών δικαιωμάτων χρήση.



### Λάθη ανεξάρτητα πρωτοκόλλου

- EHOSTUNREACH: αδυναμία πρόσβασης στον απομακρυσμένο κόμβο.
- ENETUNREACH: αδυναμία πρόσβασης στο απομακρυσμένο δίκτυο.
- EHOSTDOWN: ο απομακρυσμένος κόμβος είναι κλειστός.
- ENETDOWN: το απομακρυσμένο δίκτυο είναι κλειστό.
- EAFNOTSUPPORT: δεν υποστηρίζεται η ζητούμενη οικογένεια διευθύνσεων (IPv4, IPv6 κτλ.)
- EAGAIN: το socket είναι σε κατάσταση non-blocking E/E και η κλήση θα μπλόκαρε ή ο πυρήνας δε μπορούσε να ολοκληρώσει την λειτουργία την παρούσα στιγμή.
- EADDRINUSE: η διεύθυνση χρησιμοποιείται ήδη.
- EADDRNOTAVAIL: η ζητούμενη διεύθυνση δεν είναι διαθέσιμη στον τοπικό κόμβο.





## Διαχείριση λαθών

### Ανταλλαγή δεδομένων

- Συνήθως ο πυρήνας του λειτουργικού συστήματος διατηρεί buffers<sup>α</sup> για τα εισερχόμενα και εξερχόμενα δεδομένα για κάθε socket.
  - Οι κλήσεις αποστολής δεδομένων γράφουν δεδομένα **στους buffers εξόδου**.
  - Οι κλήσεις λήψης δεδομένων διαβάζουν δεδομένα **από τους buffers εισόδου**.
- Ο πυρήνας μπορεί να μην στείλει όλα τα δεδομένα από τους buffer εξόδου.
  - Επειδή είναι ενεργοποιημένος ο αλγόριθμος Nagle.
  - Επειδή υπάρχει περιορισμός υλικού στην κάρτα δικτύου.
  - Επειδή έπρεπε να διακόψει την αποστολή για να κάνει κάποια άλλη λειτουργία.
  - ...
- Αντίστοιχα, μπορεί να μην επιστρέψει όλα τα δεδομένα κατά την ανάγνωσή τους.
  - Επειδή δεν έχουν φτάσει ακόμα όλα τα ζητούμενα δεδομένα.
  - Επειδή βρίσκεται σε φάση ανασύνθεσης των εισερχόμενων πακέτων.
  - ...
- **Πρέπει να ελέγχουμε την τιμή επιστροφής των κλήσεων E/E και να τις επαναλαμβάνουμε αν χρειαστεί!**

<sup>α</sup>Υπενθύμιση: ρυθμίζουμε το μέγεθός τους με τις επιλογές SO\_SNDBUF και SO\_RCVBUF.



## Διαχείριση λαθών

### Ανταλλαγή δεδομένων

- **Διαφορετικά συστήματα έχουν διαφορετική αναπαράσταση δεδομένων.**
  - Διαφορετικά μεγέθη βασικών τύπων δεδομένων.
  - Διαφορετική σειρά αναπαράστασης (big/little-endian).
  - Διαφορετικές απαιτήσεις «ευθυγράμμισης».
  - ...
- **Πρέπει να φρονίζουμε τα δεδομένα να είναι μεταφέρσιμα!**
  - Με χρήση συναρτήσεων μετατροπής (`htons(3)`, `ntohs(3)`, ...)
  - Με μετατροπή τους προς/από συμβολοσειρές.
  - Με χρήση κάποιας «εξωτερικής αναπαράστασης», όπως το πρότυπο XDR<sup>α</sup>.
- **Πολλές φορές ενδέχεται να μεταφέρουμε πολλαπλά αντικείμενα (αρχεία, δομές δεδομένων, κτλ.) από την ίδια σύνδεση.**
- **Πρέπει να φρονίζουμε να διαχωρίζουμε τα δεδομένα!**
  - Μεταφορά τμημάτων σταθερού μεγέθους.
  - «Μαγικές ακολουθίες» δεδομένων για οριοθέτηση.
  - Πρωτόκολλα που προδιαγράφουν ή καθορίζουν το μέγεθος των δεδομένων.

<sup>α</sup>`xdr(3)`, RFCs 1832 & 4506



### Εργαλεία αποσφαλμάτωσης

- Μπορούμε να εξετάσουμε βήμα/βήμα και να αποσφαλματώσουμε τον κώδικά μας με χρήση του `gdb(1)` (Gnu Debugger).
- Μπορούμε να εξετάσουμε τις κλήσεις συστήματος που κάνει το πρόγραμμά μας με τα εργαλεία `truss(1)` / `kttrace(1)` / `strace(1)`.
- Αν έχουμε πρόσβαση `root`, μπορούμε να ελέγξουμε τα δεδομένα που μεταφέρονται στο δίκτυο με το εργαλείο `tcpdump(1)`.
- Μπορούμε να «μιλάμε» απευθείας με τις εφαρμογές μας με χρήση του προγράμματος `telnet(1)`.
- Μπορούμε να δημιουργήσουμε εύκολα δοκιμαστικούς πελάτες ή εξυπηρετητές με χρήση του εργαλείου `nc(1)` (netcat).
- Μπορούμε να εξετάσουμε όλα τα sockets του συστήματος με τις εντολές `netstat(1)` και `lsof(1)`.



## Ασυνδεσμική επικοινωνία



## Διαχείριση λαθών: Ασυνδεσμική επικοινωνία

### Γενικά

- Κυριότερα προβλήματα:
  - Απώλεια πακέτων.
  - Καθυστέρηση στην παράδοση.
  - Παράδοση διπλότυπων.
  - Έλλειψη ελέγχου ροής.
- Η υλοποίηση του πρωτοκόλλου στην διεπαφή των BSD sockets παρέχει έλεγχο λαθών **μόνο σε τοπικό επίπεδο**.
- Αν απαιτείται αξιοπιστία, η εφαρμογή πρέπει να υλοποιεί το **δικό της μηχανισμό ανίχνευσης σφαλμάτων και ανάνηψης**.
- Συνήθως απαιτείται να έχουμε καλέσει `connect (2)` για το UDP socket για να μπορούμε να λάβουμε τα μηνύματα λάθους ICMP.



## Διαχείριση λαθών: Ασυνδεσμική επικοινωνία

### Απώλεια πακέτων

- Σε επικοινωνία πάνω από δίκτυα ευρείας περιοχής ή υπερφορτωμένα τοπικά δίκτυα, υπάρχει πιθανότητα κάποιο πακέτο να μην παραδοθεί ποτέ.
  - Τι γίνεται αν αυτό είναι το μοναδικό πακέτο αίτησης/απάντησης;
  - Τι γίνεται αν είναι το τελευταίο και ο παραλήπτης περιμένει κάποιο προκαθορισμένο αριθμό πακέτων/«μαγική» αλληλουχία bytes;
- **Η εφαρμογή πρέπει να θέτει χρονικούς περιορισμούς στις κλήσεις E/E.**
  - Πώς ξέρουμε αν κάποιο εξερχόμενο πακέτο χάθηκε ή απλά έχει καθυστερήσει;
  - Πώς ξέρουμε αν κάποιο εισερχόμενο πακέτο χάθηκε, έχει καθυστερήσει ή αν δεν στάλθηκε ποτέ (κατάρρευση άλλου άκρου επικοινωνίας);



## Διαχείριση λαθών: Ασυνδεσμική επικοινωνία

### Αποστολή δεδομένων

- Ακόμα κι αν οι `send(2)` / `sendto(2)` / `sendmsg(2)` επιτύχουν, δεν υπάρχει εγγύηση παράδοσης των δεδομένων.
- Επιπλέον λάθη που μπορεί να συμβούν:
  - `EACCESS`: προσπάθεια αποστολής σε διεύθυνση broadcast χωρίς να έχει ενεργοποιηθεί η επιλογή `SO_BROADCAST`.
  - `EMSGSIZE`: προσπάθεια αποστολής μηνύματος μεγαλύτερου μεγέθους από το υποστηριζόμενο.
  - `ECONNREFUSED`: το τελευταίο μήνυμα πήρε ICMP απάντηση «destination unreachable» οπότε το απομακρυσμένο άκρο δεν δέχεται συνδέσεις/δεδομένα.



## Διαχείριση λαθών: Ασυνδεσμική επικοινωνία

### Λήψη δεδομένων

- Καθώς το UDP δεν προβλέπει κάποιο μηχανισμό ελέγχου ροής, υπάρχει πιθανότητα υπερχειίλισης των buffers στον παραλήπτη.
  - Και οι buffers του αποστολέα μπορεί να υπερχειλίσουν αλλά τότε θα επιστρέψει σφάλμα η `send(2)` / `sendto(2)` / `sendmsg(2)`.
- Μοναδική λύση: μεγαλύτεροι buffers. . .
  - Με τις επιλογές `SO_SNDBUF` / `SO_RCVBUF` επιπέδου `SOL_SOCKET`.
- **Αν δεχόμαστε δεδομένα από πολλούς αποστολείς, πρέπει επίσης να ελέγχουμε κάθε μήνυμα για τον αποστολέα του!**





## Συνδεομοστραφής επικοινωνία



## Διαχείριση λαθών : Συνδεσμοστραφής επικοινωνία

### Γενικά

- Κυριότερα προβλήματα :
  - Αδυναμία σύνδεσης.
  - Τερματισμός/κατάρρευση σύνδεσης κατά την επικοινωνία.
  - Έλεγχος δεδομένων.
- Το πρωτόκολλο εγγυάται την αξιόπιστη λειτουργία ακόμα και πάνω από αναξιόπιστα δίκτυα.
  - **Αυτό δε σημαίνει ότι πάντα θα εκτελούνται οι ζητούμενες λειτουργίες!**
  - **Σημαίνει όμως ότι το σύστημα θα μας ειδοποιεί σε περιπτώσεις σφαλμάτων!**



## Διαχείριση λαθών: Συνδεσμοστροφής επικοινωνία

### Αδυναμία σύνδεσης

- Κυριότερα προβλήματα:
  - EADDRINUSE: η διεύθυνση χρησιμοποιείται ήδη.
  - ENOTCONN: προσπάθεια E/E σε συνδεσμοστροφές socket για το οποίο δεν έχει κληθεί ακόμα η `connect (2)`.
  - EISCONN: προσπάθεια σύνδεσης ήδη συνδεδεμένου socket.
  - ETIMEDOUT: παρήλθε το ανώτατο χρονικό όριο πραγματοποίησης σύνδεσης.
  - ECONNREFUSED: αδυναμία σύνδεσης – το απομακρυσμένο άκρο δεν δέχεται συνδέσεις.
  - EINPROGRESS: το socket είναι σε κατάσταση non-blocking και η σύνδεση δε μπορεί να ολοκληρωθεί άμεσα.
  - EINTR: η κλήση σύνδεσης διακόπηκε από κάποιο σήμα. Η σύνδεση θα ολοκληρωθεί στο παρασκήνιο.
  - EALREADY: κάποια προηγούμενη προσπάθεια σύνδεσης δεν έχει ολοκληρωθεί ακόμα.
  - EAGAIN: επιλέχθηκε αυτόματη ανάθεση αριθμού τοπικής θύρας και δεν υπάρχει κάποια διαθέσιμη θύρα στον τοπικό κόμβο.
  - EACCES: προσπάθεια σύνδεσης σε διεύθυνση broadcast σε socket το οποίο δεν βρίσκεται στην κατάλληλη κατάσταση λειτουργίας.



## Διαχείριση λαθών: Συνδεσμολογία επικοινωνία

### Αδυναμία αποδοχής σύνδεσης

- Κυριότερα προβλήματα:
  - EWOULDBLOCK: το socket είναι σε κατάσταση non-blocking E/E και δεν υπάρχουν αιτήσεις σύνδεσης προς αποδοχή.
  - EINVAL: κλήση της `accept (2)` σε συνδεσμολογία socket για το οποίο δεν έχει κληθεί ακόμα η `listen (2)`.
  - ECONNABORTED: λήφθηκε νέα σύνδεση αλλά τερματίστηκε πριν αφαιρεθεί από την ουρά αναμονής (πριν κληθεί η `accept (2)`).



## Διαχείριση λαθών : Συνδεσμοστραφής επικοινωνία

### Ανταλλαγή δεδομένων

- Κάποιο από τα δύο άκρα επικοινωνίας ενδέχεται να αποχωρήσει ή να καταρρεύσει κατά την διάρκεια ανταλλαγής δεδομένων.
- Στην περίπτωση κατάρρευσης ή τερματισμού της απομακρυσμένης διεργασίας, ο απομακρυσμένος πυρήνας θα εκκινήσει τη διαδικασία αποσύνδεσης.
  - Θα στείλει μήνυμα FIN.
  - Ο τοπικός πυρήνας θα απαντήσει με μήνυμα ACK.
  - Τα δύο άκρα θα περάσουν σε κατάσταση FIN\_WAIT\_2 και CLOSE\_WAIT αντίστοιχα.
  - Το ίδιο όμως δεν γίνεται και στο `shut down (2)` ;
- Αν γίνει προσπάθεια επικοινωνίας σε αποσυνδεδεμένο socket οι κλήσεις E/E θα αποτυγχάνουν.
  - Αρχικά επιστρέφουν ECONNRESET: το απομακρυσμένο άκρο της επικοινωνίας έκλεισε.
  - Αν το λάθος αγνοηθεί, μετέπειτα κλήσεις θα επιστρέψουν EPIPE ή σήμα SIGPIPE.



## Διαχείριση λαθών: Συνδεσμοστραφής επικοινωνία

### Ανταλλαγή δεδομένων

- Στη περίπτωση κατάρρευσης του απομακρυσμένου κόμβου, δεν αποστέλλεται κάποιο μήνυμα.
  - Οι κλήσεις E/E του τοπικού κόμβου θα αποτυγχάνουν με λάθη ETIMEDOUT, EHOSTUNREACH ή ENETUNREACH.
  - Μπορούμε να θέσουμε χρονικά όρια για να αποφύγουμε μεγάλες καθυστερήσεις.
  - Μπορούμε να ενεργοποιήσουμε την επιλογή `SO_KEEPALIVE` για να ανιχνεύσει το επίπεδο των sockets την κατάρρευση της σύνδεσης.
- Για να αποφευχθούν τέτοια προβλήματα, καλό είναι οι πελάτες/εξυπηρετητές μας να «τερματίζουν με χάρη» (graceful termination).
  - Για παράδειγμα, κατά το κλείσιμο/επανεκκίνηση του τοπικού κόμβου.
- Συνήθως αυτό γίνεται με χειριστές για τα σήματα `SIGTERM` (στέλνεται κατά το κλείσιμο του υπολογιστή) και `SIGINT` (στέλνεται όταν πατήσουμε `Ctrl+c`).



## Πύλες και σήραγγες



# Σήραγγες





## Σήραγγες επιπέδου μεταφοράς

- Εκτός από τη γνωστή μας οικογένεια πρωτοκόλλων TCP/IP, υπάρχουν/έχουν υπάρξει και άλλες.
  - WAP.
  - Appletalk.
  - DECNET.
  - IBM SNA.
  - Novell IPX.
  - X.25.
  - ...
- **Πως μπορούμε να διασυνδέσουμε τέτοια διαφορετικά δίκτυα;**
  - Παράδειγμα: έστω ένα μικρό δίκτυο στο οποίο συνυπάρχουν:
    - Κάποιοι υπολογιστές Macintosh που χρησιμοποιούν Appletalk για να επικοινωνήσουν.
    - Κάποιοι προσωπικοί υπολογιστές (PCs) που χρησιμοποιούν TCP/IP.
    - Σύνδεση ISDN με τον πάροχο υπηρεσιών διαδικτύου η οποία χρησιμοποιεί X.25.



## Σήραγγες επιπέδου μεταφοράς

- Την «μετάφραση» αναλαμβάνουν ειδικά ρυθμισμένοι δρομολογητές, οι οποίοι δημιουργούν «σήραγγες».
- Ουσιαστικά, χειριζόμαστε τα πακέτα του επιπέδου μεταφοράς ως πακέτα υψηλότερων επιπέδου.
  - Κάθε επίπεδο στη στοίβα OSI προσθέτει τη δική του κεφαλίδα, «ενθυλακώνοντας» έτσι τα πακέτα υψηλότερων επιπέδων.
  - Στις σήραγγες χρησιμοποιούμε πακέτα του τελικού πρωτοκόλλου επιπέδου μεταφοράς για να «ενθυλακώσουμε» πακέτα του αρχικού πρωτοκόλλου επιπέδου μεταφοράς.
  - Ο δρομολογητής-παραλήπτης πρέπει να είναι κι αυτός κατάλληλα ρυθμισμένος για να ανασυνθέσει τα αρχικά πακέτα.



## Σήραγγες επιπέδου εφαρμογών

- Η τεχνική αυτή μπορεί επίσης να χρησιμοποιηθεί όταν οι δρομολογητές δε μπορούν να ρυθμιστούν για να δημιουργούν σήραγγες για πρωτόκολλα επιπέδου μεταφοράς.
  - Παράδειγμα: επικοινωνία με πρωτόκολλο IPX ανάμεσα σε κόμβους που χρησιμοποιούν TCP/IP.
  - Ο χρήστης μπορεί να δημιουργήσει μία δική του βιβλιοθήκη η οποία μεταφράζει τις κλήσεις συστήματος της οικογένειας πρωτοκόλλων IPX σε αντίστοιχες κλήσεις TCP/IP.
- Η ίδια τεχνική μπορεί να εφαρμοστεί και στην περίπτωση που δύο απομακρυσμένες εφαρμογές θέλουν να επικοινωνήσουν αλλά δεν μπορούν να συνδεθούν απευθείας η μία στην άλλη ή ακόμα και για λόγους ασφάλειας, ελέγχου, κτλ.
  - Παράδειγμα: κρυπτογραφημένη σύνδεση στον εξυπηρετητή SMTP.
    - Για να στείλουμε κάποιο e-mail ο πελάτης μας πρέπει να συνδεθεί με τον εξυπηρετητή SMTP zeus.cs.uoi.gr στη θύρα 25.
    - Ωστόσο ο εξυπηρετητής δε διαθέτει υπηρεσίες κρυπτογράφησης για τη θύρα αυτή.
    - Δημιουργούμε μία σήραγγα μέσω του πρωτοκόλλου SSH από τη θύρα 2000 στον τοπικό υπολογιστή στην θύρα 25 του εξυπηρετητή:  
`me@mypc:~$ ssh -L 2000:zeus.cs.uoi.gr:25 scylla.cs.uoi.gr`
    - Τώρα τα δεδομένα που γράφω στη θύρα 2000 του τοπικού υπολογιστή μετατρέπονται σε δεδομένα SSH, μεταφέρονται στον scylla.cs.uoi.gr και από εκεί στη θύρα 25 του zeus.cs.uoi.gr.
  - Κάτι αντίστοιχο συμβαίνει και όταν συνδεόμαστε σε κάποιον εξυπηρετητή παγκόσμιου ιστού μέσω SSL.



# Πύλες



## Πύλες επιπέδου εφαρμογών

- Πολλές φορές το πρόβλημα βρίσκεται στο επίπεδο εφαρμογών.
  - Ο υπολογιστής του πελάτη δεν διαθέτει το κατάλληλο λογισμικό.
  - Ο υπολογιστής του πελάτη δεν διαθέτει τις ελάχιστες απαιτήσεις εκτέλεσης κάποιου προγράμματος.
  - Η πολιτική ασφαλείας του συστήματος/δικτύου δεν επιτρέπουν την πρόσβαση σε κάποιες διευθύνσεις/θύρες.
- **Τι μπορούμε να κάνουμε σε αυτές τις περιπτώσεις;**
  - Παράδειγμα: σύνδεση σε υπηρεσία FTP μέσω φυλλομετρητή παγκόσμιου ιστού.
    - Ο φυλλομετρητής μου χρησιμοποιεί μόνο το πρωτόκολλο HTTP.
    - Θέλω μέσω αυτού να αποκτήσω πρόσβαση σε μία υπηρεσία FTP.



## Πύλες επιπέδου εφαρμογών

- Τη «μετάφραση» αναλαμβάνουν ειδικά προγράμματα λογισμικού τα οποία ονομάζονται «**πύλες επιπέδου εφαρμογών**».
  - Τα προγράμματα αυτά χρησιμοποιούν και τα δύο πρωτόκολλα ανάμεσα στα οποία μεταφράζουν.
  - Μεταφράζουν κάθε αίτημα του πελάτη στο πρωτόκολλο που χρησιμοποιεί ο εξυπηρετητής.
  - Μεταφράζουν κάθε απάντηση του εξυπηρετητή στο πρωτόκολλο που χρησιμοποιεί ο πελάτης.
- Υπάρχει περίπτωση η πύλη να χρησιμοποιεί το ίδιο πρωτόκολλο με τον πελάτη και τον εξυπηρετητή.
  - Προσφέρει απλά ένα ενδιάμεσο βήμα ανάμεσα στον πελάτη και τον εξυπηρετητή.
  - Γιατί χρειάζεται κάτι τέτοιο ;
    - Ασφάλεια (ο πελάτης δεν αποκτά πρόσβαση απευθείας στον εξυπηρετητή).
    - Απόδοση (παράδειγμα: web proxies).
    - Αξιοπιστία (παράδειγμα: διαμοιρασμός φόρτου σε πολλαπλούς εξυπηρετητές).
    - ...



## Σύγκριση



## Σήραγγες ή πύλες;

- Οι πύλες:

- + Δεν απαιτούν ρύθμιση του λειτουργικού συστήματος ή των δικτυακών συσκευών.
- + Δεν απαιτούν αλλαγές στα προγράμματα πελάτη και εξυπηρετητή.
- + Επιτρέπουν στα διάφορα συστήματα να συνεχίσουν να λειτουργούν όπως και πριν, χωρίς να απαιτεί από τους χρήστες ή τους διαχειριστές να κάνουν κάποια αλλαγή ή να μάθουν κάποια νέα τεχνολογία/διεπαφή.
- Απαιτούν μία νέα, διαφορετική πύλη για κάθε ζεύγος πρωτοκόλλων.
- Ίσως αυξήσουν τις απαιτήσεις σε υλικό, δικτυακούς πόρους, άδειες λογισμικού κτλ. για τον κόμβο που θα εκτελεί τον κώδικα της πύλης.

- Οι σήραγγες:

- + Είναι διαφανείς προς τις εφαρμογές, αφού λειτουργούν σε χαμηλότερο επίπεδο.
- + Δεν απαιτούν αλλαγές όταν χρειασθεί η προσθήκη κάποιας νέας υπηρεσίας (στο επίπεδο εφαρμογών).
- Απαιτούν την παρέμβαση στο λειτουργικό σύστημα και τις ρυθμίσεις των συσκευών του δικτύου που θα συμμετέχουν στη «σήραγγα».
- Σε περίπτωση αλλαγής στα υποστηριζόμενα πρωτόκολλα θα απαιτηθεί και αλλαγή στο λογισμικό των πελατών/εξυπηρετητών.





- Broadcast/multicast.
- SCTP multihoming/multistreaming.
- RAW sockets.
- Πρόσβαση στο επίπεδο διασύνδεσης (datalink).

