Κεφάλαιο 5

ΤΟ ΠΡΩΤΟΚΟΛΛΟ TCP
(*TRANSMISSION CONTROL PROTOCOL*)

5.1 Εισαγωγή :

Όπως είδαμε, το πρωτόκολλο IP του 3ου στρώματος παρέχει υπηρεσίες αποστολής πακέτων δια μέσου ετερογενών φυσικών δικτύων χωρίς όμως να εξασφαλίζει την αξιοπιστία της ροής των δεδομένων. Ο λόγος είναι ότι η ανομοιογένεια και πανσπερμία των φυσικών δικτύων δεν θα επέτρεπε έτσι και αλλιώς εγγυημένη μετάδοση καθιστώντας αναγκαίο ένα πρωτόκολλο στρώματος μεταφοράς "βαρέως τύπου" που θα εξασφαλίσει την απαιτούμενη αξιοπιστία που δεν μπορεί να προσφέρει το IP. Πάντως δεν είναι μόνο το IP που δεν εξασφαλίζει αξιοπιστία, είναι συνήθης η παραδοχή ότι τα κατώτερα στρώματα ενός δικτύου υπολογιστών παρέχουν μη αξιόπιστη αποστολή πακέτων (παρά τις τεχνολογικές εξελίξεις και κυρίως την εισαγωγή οπτικών ινών), με την έννοια ότι τα πακέτα μπορούν να καταστραφούν από πολλές αιτίες π.χ. όταν να αλλοιωθούν αρκετά ψηφία στα δεδομένα από θορύβους, όταν αστοχήσουν συσκευές του δικτύου, ή όταν το δίκτυο δεν μπορεί να αντεπεξέλθει στο φορτίο που υπάρχει τη δεδομένη στιγμή. Τα δίκτυα που δρομολογούν δυναμικά τα πακέτα, μπορεί να τα αποστείλουν στο δέκτη με διαφορετική σειρά, από αυτή που τα έστειλε ο πομπός, ή να τα στείλουν με κάποια καθυστέρηση, ή ακόμη να στείλουν μερικά πακέτα δυο φορές.

Είναι λοιπόν αναγκαίες κάποιες πρόσθετες λειτουργίες για την πρόσδοση αξιοπιστίας όπως π.χ. ανίχνευση και διόρθωση λαθών με επαναποστολή λανθασμένων και μη αφιχθέντων πακέτων, συναρμολόγηση κτλ, ώστε οι εφαρμογές που τρέχουν στα ανώτερα επίπεδα και δεν ανέχονται πολλές φορές ούτε ένα λανθασμένο δυαδικό ψηφίο, να μπορούν να λειτουργήσουν. Φυσικά θα μπορούσε κάθε εφαρμογή να προβλέπει έλεγχο και αποκατάσταση λαθών. Η κατασκευή όμως προγραμμάτων που μπορούν να εγγυηθούν αξιοπιστία είναι δύσκολη και λίγοι προγραμματιστές εφαρμογών έχουν τις τεχνικές γνώσεις. Σαν αποτέλεσμα ένας βασικός στόχος στην έρευνα πρωτοκόλλων δικτύου ήταν η εύρεση μια γενικής λύσης στο πρόβλημα της αξιόπιστης αποστολής δεδομένων, η οποία θα έδινε την δυνατότητα να αναπτυχθεί από τους ειδικούς ένα πρωτόκολλο για την αποστολή των δεδομένων και το οποίο θα μπορούσε να χρησιμοποιηθεί από κάθε εφαρμογή χωρίς να χρειάζεται ο σχεδιαστής της να ξέρει λεπτομέρειες από δίκτυα όπως δεν χρειάζεται να σχεδιάσει το λειτουργικό σύστημα μαζί με την εφαρμογή (απόκρυψη πληροφορίας και στρωμάτωση λειτουργιών). Με την λύση της τυποποίησης πρωτοκόλλου μεταφοράς απηλλάγη ο κάθε συντάκτης προγραμμάτων εφαρμογής και ο χρήστης από την ανάγκη εξασφάλισης αξιόπιστης ροής δεδομένων και την λειτουργία αυτή ανέλαβε το ειδικό πρωτόκολλο μεταφοράς του δικτύου. Το πρωτόκολλο αυτό στην περίπτωση του Διαδικτύου είναι το *Transmission Control Protocol (TCP).* Είναι τόσο στενά συνυφασμένο με το IP που συνήθως τα συναντάμε μαζί σαν TCP/IP και είναι σχεδόν ταυτόσημα με το Διαδίκτυο.

Ωστόσο, το TCP είναι ένα ανεξάρτητο και γενικό πρωτόκολλο που μπορεί να χρησιμοποιηθεί και σε άλλα δίκτυα. Για παράδειγμα επειδή το πρωτόκολλο TCP θέτει πολύ λίγες προϋποθέσεις για το δίκτυο πάνω απ' το οποίο τρέχει, μπορεί να χρησιμοποιηθεί πάνω από ένα απλό δίκτυο Ethernet ή μία τηλεφωνική ζεύξη με modem το ίδιο καλά όπως και πάνω από ένα πολύπλοκο διεθνές δίκτυο. Είναι το πιο δημοφιλές πρωτόκολλο μεταφοράς και έχει αποτελέσει την βάση του πρωτοκόλλου TP4 που δημιούργησε ο Διεθνής Οργανισμός για την Τυποποίηση πρωτοκόλλων ανοιχτών συστημάτων (International Organization for Standardization's open system protocols).

Παρ' ότι το TCP και το IP βρίσκονται τόσο συχνά μαζί δεν πρέπει να ξεχνάμε ότι αναφέρονται σε διαφορετικά στρώματα δικτύου. Το IP ανήκει στο στρώμα δικτύου και τρέχει εκτός από τα τερματικά

και στους κόμβους ενώ το TCP ανήκει στο στρώμα μεταφοράς και υλοποιείται μόνο στα άκρα της σύνδεσης αλλά όχι και στους κόμβους δρομολόγησης.

Μία άλλη μεγάλη διαφορά είναι ότι το IP είναι πρωτόκολλο άνευ συνδέσεων (connectionless) ενώ το TCP χρησιμοποιεί συνδέσεις πράγμα απαραίτητο για τη δημιουργία κλειστού βρόχου ελέγχου λαθών και επαναποστολής λανθασμένων και χαμένων πακέτων. Το TCP είναι ένα πρωτόκολλο επικοινωνίας και όχι κάποιο λογισμικό (software) παρ' ότι συχνά χρησιμοποιούμε φράσεις όπως "φόρτωσα στον υπολογιστή τη στοίβα TCP/IP" δηλ. το λογισμικό που υλοποιεί τα πρωτόκολλα αυτά. Η διαφορά μεταξύ ενός πρωτοκόλλου και ενός προγράμματος που υλοποιεί το πρωτόκολλο, είναι ανάλογη με τη διαφορά που υπάρχει μεταξύ του ορισμού μιας γλώσσας προγραμματισμού και ενός μεταγλωττιστή (compiler). Όπως συμβαίνει συχνά στο κόσμο των γλωσσών προγραμματισμού, η διαφορά μεταξύ ορισμού και εφαρμογής γίνεται δύσκολη. Επειδή κάποιος ασχολείται καθημερινά πιο πολύ με το software που υλοποιεί το πρωτόκολλο TCP, παρά με τις προδιαγραφές που το καθορίζουν, η σύγχυση αυτή είναι συχνό φαινόμενο.

Αλλά τι ακριβώς καθορίζει το πρωτόκολλο TCP και πώς λειτουργεί; Το πρωτόκολλο καθορίζει τη φόρμα (format) δηλαδή τη μορφή αποστολής των δεδομένων και των μηνυμάτων ελέγχου του ιδίου του πρωτοκόλλου όπως π.χ. τα μηνύματα επιβεβαίωσης (acknowledgement), τα οποία ανταλλάσσουν οι δύο υπολογιστές ώστε να επιτύχουν την επιζητούμενη αξιόπιστη μεταφορά των δεδομένων. Επίσης καθορίζει τη διαδικασία που πρέπει να χρησιμοποιηθεί από τους υπολογιστές ώστε να εξασφαλίσουν ότι τα δεδομένα λαμβάνονται σωστά, πώς το λογισμικό υλοποίησης θα ξεχωρίζει τους διαφορετικούς προορισμούς των πακέτων που φθάνουν στον υπολογιστή προορισμού, και πως μπορούν να ανανήψει η επικοινωνία από διάφορα λάθη, όπως χαμένα ή διπλά πακέτα. Τέλος το πρωτόκολλο καθορίζει την διαδικασία που πρέπει να ακολουθήσουν δύο υπολογιστές ώστε να αρχικοποιήσουν μια μεταφορά δεδομένων να ορίσουν συνδέσεις και να συμφωνήσουν για το πότε ολοκληρώνεται.

Αν και το ίδιο το πρωτόκολλο TCP καθορίζει σε γενικές γραμμές τον τρόπο με τον οποίο θα το χρησιμοποιούν τα προγράμματα εφαρμογών του χρήστη, δεν περιγράφει με λεπτομέρεια την διεπαφή (interface) μεταξύ μιας εφαρμογής και του πρωτοκόλλου TCP. Δηλαδή το πρωτόκολλο περιγράφει μόνο τις λειτουργίες που αυτό εκτελεί και όχι το πως αυτές θα υλοποιηθούν. Ο λόγος που δεν καθορίζεται με λεπτομέρεια η διεπαφή μεταξύ των εφαρμογών και του TCP είναι η ευελιξία στην υλοποίησή του. Συγκεκριμένα επειδή οι προγραμματιστές του συστήματος υλοποιούν το πρωτόκολλο πάνω από το λειτουργικό σύστημα που χρησιμοποιεί ο υπολογιστής τους, αφήνονται να χρησιμοποιήσουν τη διεπαφή που υποστηρίζει το λειτουργικό σύστημα. Με το τρόπο αυτό μπορούν οι προδιαγραφές που θέτει το πρωτόκολλο να υλοποιηθούν διαφορετικά σε διάφορες μηχανές.

Τέλος επειδή το δίκτυο στο οποίο εφαρμόζεται το TCP χρειάζεται να ικανοποιεί λίγες απαιτήσεις, το πρωτόκολλο TCP μπορεί να χρησιμοποιηθεί από ποικιλία δικτύων. Το πρωτόκολλο αυτό μπορεί να εφαρμοσθεί πάνω από απλές τηλεφωνικές γραμμές, πάνω από τοπικά δίκτυα, πάνω από δίκτυα οπτικών ινών κ.α Η δυνατότητα αυτή είναι που έχει κάνει το πρωτόκολλο τόσο δημοφιλές.

Πέντε στοιχεία μπορούν να χαρακτηρίσουν το interface μεταξύ του πρωτοκόλλου TCP και της εφαρμογής του χρήστη.

- *Συνεχής ροή αδόμητων στοιχείων (Unstructured Stream Orientation).* Το πρωτόκολλο χειρίζεται τα δεδομένα που χρειάζεται να μεταφερθούν σαν μια συνεχή ροή (stream) από bits οργανωμένα φυσικά σε ομάδες των 8 bits (octets ή bytes). Η υπηρεσία αποστολής στην πλευρά του δέκτη στέλνει στην εφαρμογή ακριβώς την ίδια ακολουθία από bytes που έδωσε η εφαρμογή του ανταποκριτή στον πομπό. Εκτός του ότι χρησιμοποιείται συνεχής ροή το πρωτόκολλο δεν ενδιαφέρεται για την οποιαδήποτε δομή θα μπορούσαν να έχουν τα δεδομένα σε επίπεδο εφαρμογής. Δηλαδή η υπηρεσία αποστολής δεδομένων που παρέχει το πρωτόκολλο TCP δεν υποστηρίζει τις όποιες συγκεκριμένες δομές που ίσως χρησιμοποιούν οι εφαρμογές, στα δεδομένα που στέλνει. Έτσι, επί παραδείγματι, όταν μια εφαρμογή στέλνει τα δεδομένα ενός αρχείου μισθοδοσίας, το TCP θα τα στείλει σαν μία ροή

δεδομένων χωρίς π.χ. να μαρκάρει που αρχίζει το αρχείο του ενός υπαλλήλου και που τελειώνει. Η σημασία και η δομή των δεδομένων που ανταλλάσσονται από δύο εφαρμογές είναι ευθύνη των εφαρμογών και θα πρέπει να έχουν προσυμφωνηθεί αφού θα είναι κατανοητές μόνο από τις εφαρμογές.

- *Σύνδεση νοητού κυκλώματος (Virtual Circuit Connection).* Γίνεται χρήση της έννοιας του νοητού κυκλώματος όπως στο X.25 ή στην τηλεφωνία. Πριν δηλαδή αρχίσει η μεταφορά πρέπει τόσο ο πομπός όσο και ο δέκτης να αλληλεπιδράσουν με τα αντίστοιχα λειτουργικά συστήματα, πληροφορώντας τα, ότι επιθυμούν να έχουν μια αποστολή δεδομένων. Σαν συνέπεια πρέπει ένας από τους δύο να κάνει μία «κλήση» (call) την οποία πρέπει να αποδεχθεί ο άλλος. Τότε τα αντίστοιχα τμήματα των λειτουργικών συστημάτων, που είναι υπεύθυνα για την υλοποίηση του λογισμικού (software) του πρωτοκόλλου, θα επικοινωνήσουν, στέλνοντας μηνύματα μέσω του δικτύου, επαληθεύοντας ότι και οι δύο πλευρές είναι έτοιμες. Στην συνέχεια θα πληροφορήσουν τις εφαρμογές του πομπού και του δέκτη ότι έχει επιτευχθεί η σύνδεση και μπορεί να αρχίσει η αποστολή των δεδομένων. Κατά τη διάρκεια της αποστολής των δεδομένων το λογισμικό του πρωτοκόλλου του ενός υπολογιστή συνεχίζει να επικοινωνεί με αυτό του άλλου υπολογιστή και να πιστοποιούν ότι τα δεδομένα φτάνουν σωστά στον προορισμό τους. Αν κατά τη διάρκεια της μετάδοσης συμβεί κάποιο σφάλμα και η επικοινωνία χαθεί τότε και οι δύο υπολογιστές αντιλαμβάνονται ότι έχει διακοπεί η σύνδεση και το αναφέρουν στις αντίστοιχες εφαρμογές. Επομένως ο όρος «*νοητό κύκλωμα*» τονίζει το γεγονός ότι ενώ οι εφαρμογές θεωρούν την σύνδεση σαν ένα ξεχωριστό φυσικό κύκλωμα, στην πραγματικότητα αυτή επιτυγχάνεται μέσω του λογισμικού του πρωτοκόλλου.

- *Χρήση ταμιευτήρα (Buffered Transfer).* Κατά τη μεταφορά των δεδομένων κάθε εφαρμογή χρησιμοποιεί ό,τι μέγεθος πακέτου θεωρεί βέλτιστο, το οποίο μπορεί να είναι τόσο μικρό όσο και ένα byte αλλά συνήθως είναι πολλά bytes. Στην πλευρά του δέκτη το πρωτόκολλο στέλνει στην εφαρμογή, τα bytes ακριβώς με την ίδια σειρά με την οποία εστάλησαν, αμέσως μόλις φτάσουν και επιβεβαιωθεί ότι δεν υπάρχει λάθος. Το software του πρωτοκόλλου είναι ελεύθερο να ομαδοποιήσει τα δεδομένα σε πακέτα με μήκος που είναι ανεξάρτητο από το μέγεθος που χρησιμοποιεί η εφαρμογή για την μεταφορά των δεδομένων, προσπαθώντας κάθε φορά να χρησιμοποιήσει όσο καλύτερα μπορεί το δίκτυο. Για παράδειγμα αν εφαρμογή χρησιμοποιεί μικρό μέγεθος πακέτων, το πρωτόκολλο θα μαζέψει αρκετά δεδομένα από την εφαρμογή (σε ένα buffer), ώστε να γεμίσει ένα αρκετά μεγάλο δεδομενόγραμμα πριν αποστείλει τα δεδομένα στο δίκτυο. Με τον τρόπο αυτό ακόμα και αν η εφαρμογή έστελνε τα δεδομένα ένα byte κάθε φορά, θα είχαμε ικανοποιητική χρησιμοποίηση του δικτύου. Όμοια αν η εφαρμογή στέλνει τα δεδομένα σε υπερβολικά μεγάλο μέγεθος πακέτων, το πρωτόκολλο αναλαμβάνει να κατατμήσει τα δεδομένα σε πακέτα μικρότερου μεγέθους ώστε να έχουμε πάλι καλύτερη εκμετάλλευση του δικτύου.

Υπάρχουν όμως και εφαρμογές που απαιτούν τα δεδομένα να στέλνονται αμέσως μόλις λαμβάνονται. Για τις εφαρμογές αυτές το πρωτόκολλο παρέχει ένα μηχανισμό ώθησης που χρησιμοποιείται από την εφαρμογή για να αναγκάσει το πρωτόκολλο να αποστείλει τα δεδομένα ακόμα κι' αν δεν έχει γεμίσει πλήρως ο ταμιευτήρας. Παράδειγμα τέτοιο είναι οι εφαρμογές Telnet και rlogin που στέλνουν κάθε ένα χαρακτήρα που πληκτρολογείται (είτε κάθε γραμμή που πληκτρολογείται).

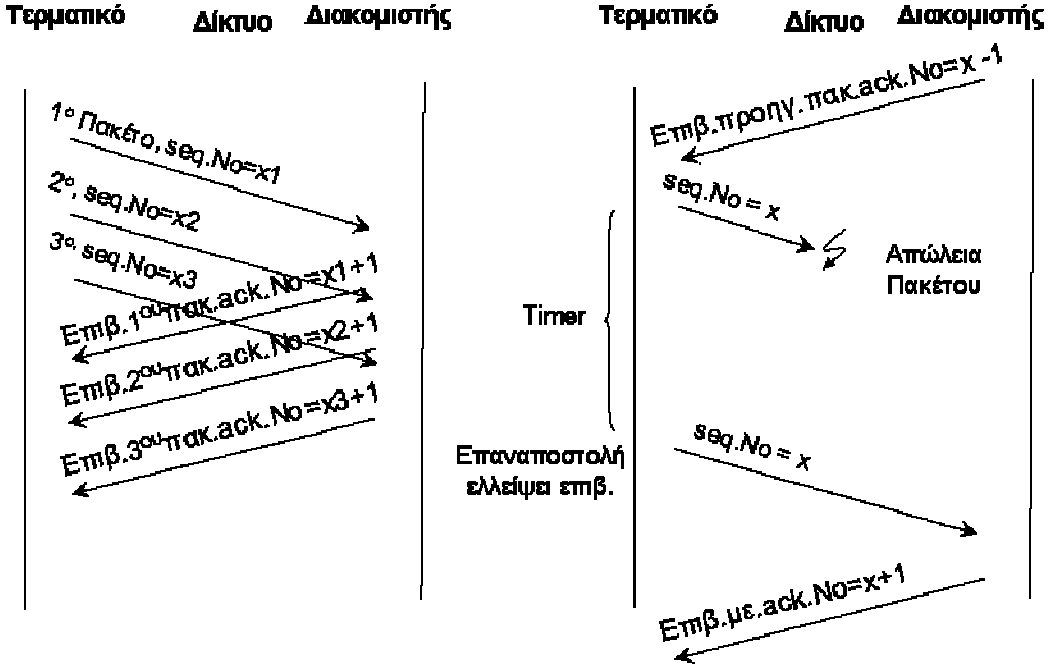
- *Αμφίδρομη σύνδεση (Full Duplex Connection).* Οι συνδέσεις που παρέχει το πρωτόκολλο TCP/IP χρησιμοποιεί ταυτόχρονη αποστολή δεδομένων και στις δύο κατευθύνσεις δηλαδή πλήρως αμφίδρομες (*full duplex) συνδέσεις.* Από την πλευρά της εφαρμογής του χρήστη μια αμφίδρομη σύνδεση αποτελείται από δύο ανεξάρτητες ροές δεδομένων σε αντίθετες κατευθύνσεις που είναι ανεξάρτητες παρ' ότι μπορούν να κουβαλούν πληροφορίες ελέγχου η κάθε μια για λογαριασμό της άλλης. Αυτό είναι και το πλεονέκτημα μιας αμφίδρομης σύνδεσης ότι δηλ. το πρωτόκολλο πάνω από

το οποίο τρέχει η εφαρμογή, μπορεί να χρησιμοποιήσει το πακέτο των δεδομένων που στέλνεται προς τη μία κατεύθυνση για να προσθέσει πληροφορίες ελέγχου που αφορούν τη ροή προς την αντίθετη κατεύθυνση. Με την τεχνική αυτή που ονομάζεται υπέρθεση (*piggybacking)* μειώνεται η κίνηση στο δίκτυο. Η υπηρεσία του πρωτοκόλλου TCP επιτρέπει ωστόσο τον τερματισμό της ροής σε μία κατεύθυνση, ενώ στην άλλη συνεχίζεται κανονικά, κάνοντας έτσι την σύνδεση ημιαμφίδρομη (*half duplex)*

5.2 Πώς επιτυγχάνεται η αξιόπιστη μεταφορά δεδομένων

Όπως έχουμε ήδη πει η υπηρεσία αξιόπιστης αποστολής δεδομένων εγγυάται την αποστολή μιας συνεχούς ροής δεδομένων από έναν υπολογιστή σε άλλον, χωρίς να χαθούν δεδομένα ή εμφανισθούν διπλά. Τα περισσότερα πρωτόκολλα που προσφέρουν αξιόπιστη μετάδοση δεδομένων χρησιμοποιούν μια απλή αλλά βασική ιδέα που είναι γνωστή σαν *positive acknowledgement with retransmission(επιβεβαίωση λήψης με αναμετάδοση).* Η τεχνική αυτή που χρησιμοποιείται και στο X.25, απαιτεί να επικοινωνεί ο δέκτης με την πηγή και να στέλνει σ' αυτήν ένα μήνυμα επιβεβαίωσης (*acknowledgement*) καθώς λαμβάνει τα δεδομένα. Ο πομπός κρατάει ένα αντίγραφο του πακέτου που έστειλε και περιμένει το μήνυμα επιβεβαίωσης από το δέκτη για να στείλει το επόμενο. Επίσης μόλις στείλει ένα πακέτο θέτει σε λειτουργία έναν χρονιστή (*timer*) και εάν μέσα σε κάποιο προσδιορισμένο χρονικό διάστημα δεν πάρει μήνυμα επιβεβαίωσης, ξαναστέλνει το πακέτο. Στο σχήμα 5.1.α φαίνεται η παραπάνω διαδικασία.

Στο σχήμα 5.1.β φαίνεται πώς αντιμετωπίζεται η περίπτωση που κάποιο πακέτο έχει χαθεί και ο πομπός δεν λάβει μέσα στο χρονικό διάστημα που έχει προσυμφωνηθεί ένα μήνυμα επιβεβαίωσης. Το τελευταίο πρόβλημα που πρέπει να λυθεί για να έχουμε αξιόπιστη μεταφορά δεδομένων είναι η περίπτωση διπλών (duplicate) πακέτων, η οποία μπορεί να συμβεί, όταν π.χ υπάρχει μεγάλη καθυστερήσει στο δίκτυο και επομένως ο πομπός αναμεταδίδει μετά την εκπνοή του χρονιστή τα ίδια πακέτα. Εδώ πρέπει να τονιστεί ότι μπορούμε να έχουμε διπλά πακέτα όχι μόνο δεδομένων αλλά και μηνυμάτων επιβεβαίωσης (να λάβει δηλαδή ο πομπός δύο μηνύματα επιβεβαίωσης για το ίδιο πακέτο). Συνήθως το πρόβλημα αντιμετωπίζεται αριθμώντας τα πακέτα και ζητώντας από τον δέκτη να «θυμάται» τους αριθμούς των πακέτων που έχει λάβει. Για να αντιμετωπιστεί το πρόβλημα διπλών μηνυμάτων επιβεβαίωσης (ack από τα πρώτα γράμματα της λέξης acknowledgement που σημαίνει επιβεβαίωση), ο δέκτης σε κάθε πακέτο που μεταφέρει μήνυμα επιβεβαίωσης στέλνει επίσης και τον αριθμό του πακέτου για το οποίο αντιστοιχεί το μήνυμα επιβεβαίωσης.



Σχήμα 5.1 α. (αριστερά) Η τεχνική *positive acknowledgement and retransmission,* σύμφωνα με την οποία ο πομπός περιμένει να λάβει μήνυμα *acknowledgement (*ACK) για κάθε πακέτο που στέλνει. **Σχήμα 5.1.β (δεξιά)** Η διαδικασία επαναποστολής όταν κάποιο πακέτο χαθεί και ο πομπός δεν λάβει μέσα στο καθορισμένο χρονικό διάστημα το ACK

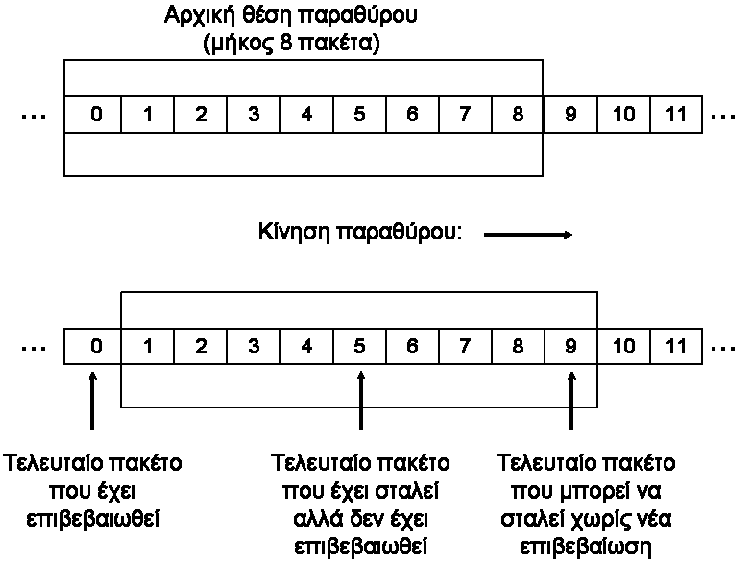
5.3 Η έννοια του παραθύρου ολίσθησης (SLIDING WINDOW)

Πριν εξεταστεί η υπηρεσία που προσφέρει το πρωτόκολλο TCP για την ροή το δεδομένων θα ξαναδούμε την έννοια του «παραθύρου ολίσθησης» (*sliding window), γνωστή και από το επίπεδο ζεύξης (HDLC) και το δίκτυο X.25.* Για να κατανοήσουμε το κίνητρο για τη χρησιμοποίηση του παραθύρου ολίσθησης, πρέπει να ξαναδούμε πιο προσεχτικά, τη ροή των δεδομένων στο σχήμα 5.1. Όπως φαίνεται για να επιτύχουμε την αξιόπιστη μεταφορά δεδομένων ο πομπός μόλις στείλει ένα πακέτο, περιμένει να πάρει επιβεβαίωση σωστής λήψης από το δέκτη πριν στείλει το επόμενο. Η τεχνική αυτή (που στα πρωτόκολλα ζεύξης τη γνωρίσαμε με το όνομα STOP-AND-WAIT) είναι πολύ συντηρητική αν και αποτελεσματική δεν είναι αποδοτική. Σε κάθε χρονική στιγμή η μετάδοση δεδομένων γίνεται μόνο σε μία κατεύθυνση, παρόλο που μπορεί το δίκτυο να υποστηρίζει ταυτόχρονη μετάδοση δεδομένων σε δύο αντίθετες κατευθύνσεις. Το δίκτυο λοιπόν παραμένει ανενεργό στα διαστήματα που οι υπολογιστές εκτελούν τους απαραίτητους υπολογισμούς για τον έλεγχο λαθών, τη δρομολόγηση των πακέτων, κ.τ.λ. Σ' ένα δίκτυο που έχει αρκετά μεγάλη καθυστέρηση μετάδοσης, το πρόβλημα που προκύπτει είναι φανερό.

Ένα πρωτόκολλο που χρησιμοποιεί την απλή τεχνική θετικής επιβεβαίωσης positive acknowledgment, σπαταλά ένα υπολογίσιμο μέρος του εύρους του δικτύου, γιατί καθυστερεί τη μετάδοση νέου πακέτου μέχρι να λάβει επιβεβαίωση για το προηγούμενο.

Η τεχνική του κινούμενου παραθύρου είναι μια πιο πολύπλοκη μορφή της απλής τεχνικής positive acknowledgement and retransmission, που αναφέρθηκε νωρίτερα. Με αυτό τον τρόπο χρησιμοποιείται αποδοτικότερα το εύρος ζώνης του δικτύου, αφού επιτρέπεται στον πομπό να στείλει περισσότερα πακέτα πριν αρχίσει να περιμένει για επιβεβαίωση.

Το παράδειγμα του σχήματος 5.2 βοηθά να κατανοήσουμε καλύτερα την τεχνική του παραθύρου ολίσθησης. Έστω ότι ο πομπός θέλει να στείλει την ακολουθία των πακέτων που φαίνονται στο σχήμα. Το πρωτόκολλο τοποθετεί ένα μικρό παράθυρο στην αρχή της σειρά των πακέτων και μεταδίδει όλα τα πακέτα που βρίσκονται μέσα στο παράθυρο, χωρίς να περιμένει επιβεβαίωση.



Σχήμα 5.2 Η έννοια του παραθύρου ολίσθησης

Ένα πακέτο που στάλθηκε και δεν έχει επιβεβαιωθεί λέγεται *unacknowledged* (ανεπιβεβαίωτο). Ο αριθμός των πακέτων που μπορεί να είναι ανεπιβεβαίωτα κάθε χρονική στιγμή περιορίζεται από το εύρος του παραθύρου που προσπαθούμε να το κάνουμε να περιλαμβάνει τόσα πακέτα όσα χωράνε και στην πραγματική σύνδεση. Αν για παράδειγμα το μήκος του παραθύρου είναι 8 ο πομπός μπορεί να στείλει 8 πακέτα πριν αρχίσει να περιμένει για επιβεβαιώσεις.

Όπως φαίνεται στο σχήμα 5.2 κάτω, μόλις ο πομπός λάβει μία επιβεβαίωση ack για το πρώτο πακέτο, το παράθυρο ολισθαίνει κατά μία θέση και μπορεί πλέον να στείλει το επόμενο πακέτο που πλέον καλύπτεται από το παράθυρο. Στη συνέχεια το παράθυρο εξακολουθεί να ολισθαίνει συνεχώς καθώς φτάνουν από το δέκτη μηνύματα επιβεβαίωσης. Κάθε φορά ο πομπός στέλνει όσα πακέτα βρίσκονται μέσα στα όρια του παραθύρου.

Η απόδοση του πρωτοκόλλου παραθύρου ολίσθησης εξαρτάται από το μέγεθος του. Επιστρέφοντας στο παράδειγμα του σχήματος 5.1.α μπορούμε να πούμε ότι η ανταλλαγή πακέτων έγινε με μήκος παραθύρου ίσο με 3 πακέτα. Όπως παρατηρούμε ο πομπός στέλνει και τα τρία πακέτα πριν λάβει ακόμα την πρώτη επιβεβαίωση).

Για μήκος παραθύρου ίσο με 1 το πρωτόκολλο παραθύρου ολίσθησης εκφυλίζεται σε απλό πρωτόκολλο θετικής επιβεβαίωσης που αναφέρεται επίσης σαν *stop and wait* (στάση και αναμονή). Αυξάνοντας το μέγεθος του παραθύρου είναι πιθανό να εξαλείψουμε τελείως το χρόνο που το δίκτυο παραμένει ανενεργό. Αυτό μπορεί να συμβεί όταν ο πομπός μπορεί να στέλνει πακέτα με τον ίδιο ρυθμό που το δίκτυο μπορεί να τα μεταδώσει. Το βασικό σημείο είναι:

Επειδή ένα καλά υπολογισμένο πρωτόκολλο παραθύρου ολίσθησης συνεχώς τροφοδοτεί το δίκτυο με πακέτα, τελικά παρατηρείται μεγαλύτερη παροχέτευση απ' ότι σε ένα απλό πρωτόκολλο θετικής επιβεβαίωσης.

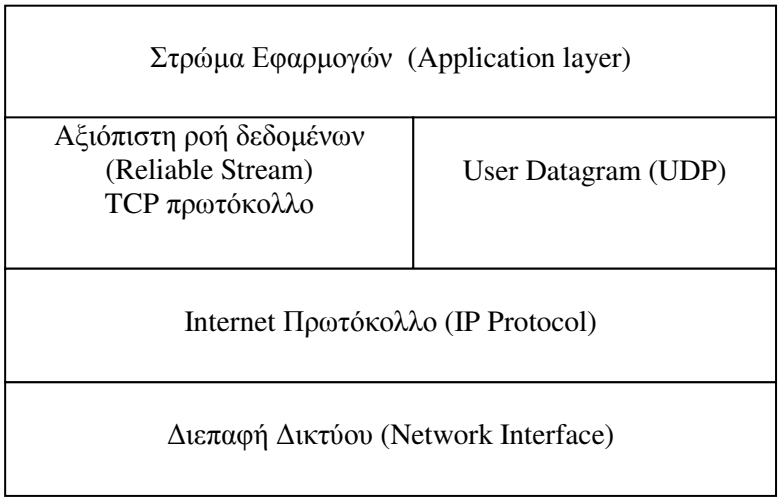
Το πρωτόκολλο παραθύρου ολίσθησης ανά πάσα στιγμή θυμάται ποια πακέτα έχουν επιβεβαιωθεί από το δέκτη, και κρατάει έναν διαφορετικό χρονιστή για κάθε ανεπιβεβαίωτο πακέτο. Αν κάποιο πακέτο χαθεί, τότε ο πομπός δεν θα λάβει μέσα στο χρονικό όριο το ACK για το πακέτο αυτό και θα το ξαναστείλει. Όταν ο πομπός κινεί το παράθυρο, προσπερνά όλα τα πακέτα που έχουν επιβεβαιωθεί. Στην πλευρά του δέκτη το πρωτόκολλο κρατάει ένα ανάλογο παράθυρο και δέχεται και επιβεβαιώνει τα πακέτα καθώς αυτά έρχονται. Ο μηχανισμός κινουμένου παραθύρου χωρίζει λοιπόν τα πακέτα σε τρία σύνολα: στα πακέτα που βρίσκονται αριστερά του παραθύρου και έχουν μεταδοθεί και επιβεβαιωθεί, στα πακέτα που βρίσκονται δεξιά του παραθύρου και τα οποία δεν έχουν μεταδοθεί ακόμα, και τέλος στα πακέτα που βρίσκονται μέσα στο παράθυρο και τα οποία έχουν μεταδοθεί αλλά δεν έχουν επιβεβαιωθεί από το δέκτη και γι' αυτό κόπιες τους βρίσκονται ακόμη στον ταμιευτήρα. Κάθε φορά το μικρότερο αριθμημένο πακέτο στο παράθυρο είναι το πρώτο πακέτο στη σειρά που δεν έχει επιβεβαιωθεί.

5.4 Θύρες (ports), Συνδέσεις (connections) και σημεία τερματισμού (end-points)

Όπως φαίνεται στο παρακάτω σχήμα στη ιεραρχία επιπέδων, το πρωτόκολλο TCP βρίσκεται πάνω απο το πρωτόκολλο IP. Το TCP επιτρέπει σε πολλές εφαρμογές να επικοινωνούν ταυτόχρονα, αναλαμβάνοντας να αποπλέκει (demultiplex) τα πακέτα TCP και να τα στέλνει στις κατάλληλες εφαρμογές. Το TCP χρησιμοποεί *αριθμούς θυρών πρωτοκόλλου (protocol port numbers)*, για να αναγνωρίζει τον τελικό προρισμό κάθε πακέτου μέσα σε κάποια μηχανή. Σε κάθε θύρα αντιστοιχίζεται ένας μικρός ακέραιος ο οποίος χρησιμοποιείται σαν ταυτότητα για τη θύρα αυτή.

Κάθε θύρα έχει μια ουρά στην οποία το λογισμικό του πρωτοκόλλου τοποθετεί τα αφικνούμενα πακέτα δηλ. τα datagrams. Το πρωτόκολλο TCP χρησιμοποιεί τους αριθμούς θυρών για να μπορεί να αναγνωρίζει τον τελικό προορισμό των δεδομένων εντός μιας μηχανής. Το TCP έχει δομηθεί πάνω σην έννοια των συνδέσεων νοητών κυκλωμάτων (virtual circuit connections). Οι συνδέσεις χαρακτηρίζονται από ένα ζεύγος σημείων τερματισμού (**end-points**). Μια συγκεκριμένη θύρα μπορεί να χρησιμοποιείται από πολλές συνδέσεις νοητών κυκλωμάτων, έτσι για να ξεχωρίζει το TCP τις διάφορες συνδέσεις χρησιμοποιεί το ζεύγος των σημείων τερματισμού για να χαρακτηρίζει κάθε σύνδεση.

ΙΕΡΑΡΧΙΑ ΠΡΩΤΟΚΟΛΛΩΝ



Σχήμα 5.3

Εφόσον μια σύνδεση είναι ένα νοητό (ή εικονικό) κύκλωμα μεταξύ δύο εφαρμογών θα ήταν λογικό να υποθέσουμε πως οι εφαρμογές χρησιμοποιούνται σαν σημεία τερματισμού από το πρωτόκολλο TCP. Αντίθετα όμως το TCP χρησιμοποιεί σαν endpoints ένα ζεύγος ακεραίων της μορφής (host,port) όπου "host" είναι η IP διεύθυνση μια μηχανής και "port" είναι μια θύρα TCP για τη μηχανή αυτή. Για παράδειγμα το σημείο τερματισμού (128.10.2.3,25) καθορίζει την θύρα 25 του TCP στη μηχανή που έχει IP διεύθυνση 128.10.2.3.

Έτσι αν έχουμε ορίσει τα σημεία τερματισμού είναι εύκολο να περιγράψουμε την σύνδεση μέσω του ζεύγους των τερματικών σημείων που ανταλλάσουν τις πληροφορίες που αυτή μεταφέρει. Ετσι 2 σημεία τερματισμού π.χ. (147.26.0.36,69) και (147.102.2.3,25) περιγράφουν μια σύνδεση μεταξύ ενος υπολογιστή με IP διεύθυνση 147.26.0.36 και ενός υπολογιστή με IP διεύθυνση 147.102.2.3. Στην ίδια μηχανή είναι δυνατό να έχουμε και άλλη σύνδεση για παράδειγμα από τη μηχανή με IP διεύθυνση 147.9.0.32 που έστω περιγράφεται από το ζεύγος σημείων τερματισμού: (147.9.0.32,84) και (147.102.2.3,53). Προχωρώντας ακόμα περισσότερο είναι δυνατό δύο συνδέσεις να μοιράζονται και την ίδια θύρα TCP. Θα μπορούσε δηλαδή να υπάρχει και άλλη σύνδεση στη θύρα 53 της μηχανής με IP διεύθυνση 147.102.2.3. π.χ από μιά μηχανη με σημείο τερματισμού (147.2.254.39,84). Κάτι τέτοιο δεν δημιουργεί σύγχυση, αφού το TCP χρησιμοποιεί και τα δύο σημεία τερματισμού για να προσδιορίσει σε ποιά σύνδεση αντιστοιχούν τα εισερχόμενα πακέτα. Επομένως το πρωτόκολλο TCP επιτρέπει πολλαπλές συνδέσεις στην ίδια μηχανή να μοιράζονται την ίδια θύρα TCP. Αυτή η δυνατότητα δίνει μεγάλη ευελιξία στους προγραμματιστές εφαρμογών αφού μπορούν πολλοί χρήστες να μοιράζονται μια εφαρμογή. Για παράδειγμα τα προγράμματα που διαβάζουν τα ηλεκτρονικά ταχυδρομεία (electronic mail) χρησιμοποιούν μία μόνο τοπική θύρα (την 25) για να δέχονται ταυτόχρονα τα μηνύματα απ' όλες τις συνδέσεις.. Το FTP χρησιμοποιεί την πόρτα 21 κτλ.

Επειδή το πρωτόκολλο TCP είναι προσανατολισμένο σε συνδέσεις (connection oriented) απαιτεί να συμφωνήσουν τόσο ο πομπός όσο και ο δέκτης, πριν αποκατασταθεί η σύνδεση. Για να γίνει αυτό, το πρόγραμμα εφαρμογής στην μία άκρη εκτελεί μια συνάρτηση «*παθητικού ανοίγματος» (passive open)* επικοινωνώντας με το λειτουργικό σύστημα, δηλώνοντας ότι αποδέχεται μια εισερχόμενη κλήση. Εκείνη τη στιγμή το λειτουργικό διαθέτει μια θύρα TCP την οποία θα χρησιμοποιεί το σημείο

τερματισμού (end-point) αυτής της σύνδεσης. Το πρόγραμμα εφαρμογής από την άλλη πλευρά επικοινωνεί με το δικό του λειτουργικό σύστημα χρησιμοποιώντας μια αίτηση *«ενεργού ανοίγματος» (active open)* για να εγκαταστήσει μια σύνδεση. Στη συνέχεια τα δύο τμήματα των λειτουργικών συστημάτων, που είναι υπεύθυνα για την υλοποίηση του πρωτοκόλλου TCP, επικοινωνούν και επιβεβαιώνουν την εγκατάσταση της σύνδεσης. Αφού εγκατασταθεί η σύνδεση τα δύο προγράμματα μπορούν ν' αρχίσουν την μεταφορά δεδομένων, ενώ το πρωτόκολλο TCP και στις δύο πλευρές ανταλλάσσει μηνύματα που εγγυώνται την αξιόπιστη μεταφορά των δεδομένων.

5.5 Τμήματα (segments), ροές (streams), και sequence numbers

Παρ'ότι το πρωτόκολλο TCP αντιμετωπίζει τα δεδομένα σαν μια συνεχή ροή από bytes ή οκτέτα (octets), δεν μπορεί φυσικά να μπεί το σύνολο των οκτέτων που θα ζητήσει η εφαρμογή να μεταδοθούν στη διάρκεια μιας σύνδεσης σε ένα τεράστιο πακέτο η μετάδοση του οποίου μπορεί να διαρκέσει ενα αυθαίρετο χρονικό διάστημα. Ετσι πρέπει το TCP να δημιουργήσει από την ροή αυτή τμήματα (segments) χωρίζοντας ένα αριθμό από οκτέτα ώστε να μπορεί να γίνει επί του τμήματος αυτού αρίθμηση, CRC, έλεγχος διπλο-μεταδόσεων κτλ.. Συνήθως κάθε τμήμα μεταδίδεται μέσω του Διαδικτύου σε ένα δεδομενόγραμμα του IP.

Το TCP χρησιμοποιεί έναν ειδικό μηχανισμό παραθύρου ολίσθησης (sliding window) που βασίζεται στην αρχή που περιγράψαμε για να πετύχει αποτελεσματική μετάδοση και έλεγχο ροής των δεδομένων. Μέσω του μηχανισμού αυτού το TCP επιτρέπει την αποστολή πολλών τμημάτων χωρίς να έχουν επιβεβαιωθεί από το δέκτη. Επίσης με το μηχανισμό αυτό λύνεται και το πρόβλημα του ελέγχου ροής από άκρη σε άκρη (end to end). Αν δηλαδή ο δέκτης δεν έχει αρκετό χώρο στον ταμιευτήρα ώστε να αποθηκεύσει προσωρινά τα νέα δεδομένα που έρχονται, μπορεί να εμποδίσει τη μετάδοση, μέχρι να ελευθερώσει αρκετό χώρο στον ταμιευτήρα, ώστε να μπορεί να διαχειριστεί νέα δεδομένα.

Θα περιγραφεί τώρα η συγκεκριμένη μορφή του μηχανισμού παραθύρου ολισθήσεως του TCP με τον τρόπο που υλοποιείται από το λογισμικό. Ο μηχανισμός αυτός βασίζεται σε αρίθμηση των bytes και όχι των τμημάτων (segment) ή πακέτων όπως στο X.25. Ο πομπός διατηρεί για κάθε ενεργή σύνδεση 3 δείκτες. που χρησιμοποιούνται για την περιγραφή του κινούμενου παραθύρου όπως φαίνεται στο κάτω μέρος του σχήματος 5.2. Ο πρώτος δείκτης τοποθετείται στα αριστερά του παραθύρου χωρίζοντας τα bytes που έχουν σταλεί και επιβεβαιωθεί από το δέκτη, από αυτά που μένουν να σταλούν ή να επιβεβαιωθούν. Ο δεύτερος δείκτης τοποθετείται στο τέλος του παραθύρου και δείχνει το μεγαλύτερο αριθμό byte που μπορεί να σταλεί, πριν αρχίσει ο πομπός να περιμένει για μηνύματα επιβεβαίωσης. Τέλος ο τρίτος δείκτης βρίσκεται μέσα στα όρια του παραθύρου και χωρίζει τα bytes που έχουν ήδη σταλέι απ'αυτά που μένουν να σταλούν. Επειδή τα bytes μέσα στα όρια του παραθύρου στέλνονται αμέσως χωρίς την αναμονή μηνυμάτων επιβεβαίωσης, ο τρίτος δείκτης ολισθαίνει γρήγορα απ' την αρχή του παραθύρου μέχρι το τέλος του. Η διαφορά του τελευταίου δείκτη από τον πρώτο ισούται με το μέγεθος του παραθύρου κάθε φορά και δεν μπορεί να υπερβεί το ορισθέν μέγιστο μέγεθος παραθύρου. Με τιμή παραθύρου ίση με μηδέν η εκπομπή αναστέλλεται. Δηλαδή, σε σχέση με τον απλό μηχανισμό παραθύρου ολίσθησης του HDLC/LAPB, ο αντίστοιχος μηχανισμός που χρησιμοποιείται από το TCP διαφέρει στο γεγονός ότι επιτρέπει το μήκος του παραθύρου να μεταβάλλεται με το χρόνο. Χρήση αυτού γίνεται για τον έλεγχο ροής αλλά και για τον έλεγχο συμφόρησης που θα μελετηθούν σε επόμενο κεφάλαιο.

Στην πλευρά του δέκτη υπάρχει ένας ανάλογος μηχανισμός που ελέγχει και επιβεβαιώνει τα πακέτα που στέλνει ο πομπός. Επειδή το πρωτόκολλο TCP υποστηρίζει αμφίδρομη σύνδεση full-duplex ο δέκτης μπορεί να είναι ταυτόχρονα και πομπός. Στην γενική περίπτωση λοιπόν, σε κάθε πλευρά υπάρχουν δύο παράθυρα (άρα συνολικά τέσσερα), ένα για να ελέγχει τα bytes που στέλνει στην άλλη πλευρά και ένα για να ελέγχει τα δεδομένα που λαμβάνει απ'την άλλη πλευρά.

5.6 Η φόρμα TCP

Στο σημείο αυτό ας εξετάσουμε τη φόρμα του τμήματος (segment format) του TCP που φαίνεται στο πιο κάτω σχήμα 5.4.

0 4 10 16 24 31

SOURCE PORT			DESTINATION PORT	
SEQUENCE NUMBER				
ACKNOWLEDGEMENT NUMBER				
HLEN	RESERVED	CODE BITS	WINDOW ADVERTISEMENT	
CHECKSUM			URGENT POINTER	
OPTIONS (IF ANY)				PADDING
ΔΕΔΟΜΕΝΑ (DATA)				
. . .				

Σχήμα 5.4. Η φόρμα της επικεφαλίδας TCP

Κάθε τμήμα χωρίζεται σε δύο μέρη, την επικεφαλίδα (**header)** και τα δεδομένα (**data)** Η επικεφαλίδα (*TCP Header)* μεταφέρει πληροφορίες ελέγχου (*control) και* αναγνώρισης (*identification*). Τα πεδία *SOURCE PORT (θύρα πηγής)* και *DESTINATION PORT (θύρα προορισμού)* περιέχουν τους αντίστοιχους αριθμούς των θυρών TCP και οι οποίοι χρησιμοποιούνται στα σημεία τερματισμού μέσα στο λογισμικό των τερματικών για να προσδιορίσουν μια σύνδεση απ' άκρου εις άκρο. Κάθε πόρτα αντιστοιχίζεται με κάποια διαφορετική εφαρμογή μέσα στον υπολογιστή. Το πεδίο *SEQUENCE NUMBER* είναι ένας αριθμός που μετρά συνεχώς τα bytes που αποστέλλονται και έτσι προσδιορίζει την θέση των δεδομένων του τμήματος, στην συνολική ακολουθία των bytes των δεδομένων του πομπού. (Όπως έχουμε πει το TCP θεωρεί τα δεδομένα που ανταλλάσσουν δύο εφαρμογές σαν μια συνεχή ροή από οκτάδες). Το πεδίο ACKNOWLEDGEMENT NUMBER (αριθμός επιβεβαίωσης) προσδιορίζει μέχρι ποιό αριθμό έχουν ληφθεί σωστά τα δεδομένα, δηλ. επιβεβαιώνει στον πομπό μέχρι ποιό σημείο μπορεί πλέον να σβύσει τα δεδομένα από τον ταμιευτήρα του διότι βρίσκονται ήδη στον ταμιευτήρα του δέκτη. (Αυτός είναι και ο σκοπός της αξιόπιστης μεταφοράς). Η σύμβαση είναι να τοποθετείται ο αριθμός του επόμενου byte που περιμένει να του στείλει η άλλη πλευρά. Πρέπει να γίνει κατανοητό ότι στο ίδιο πακέτο ενώ το πεδίο SEQUENCE NUMBER αναφέρεται στα δεδομένα που στέλνονται στη μια κατεύθυνση το πεδίο ACKNOWLEDGEMENT NUMBER αναφέρεται στα δεδομένα που αποστέλλονται από την άλλη κατεύθυνση και φυσικά μπορεί να διαφέρουν πολύ αφού δεν έχουν καμία σχέση μεταξύ τους. Αυτά που είναι σχετικά είναι το SEQUENCE NUMBER στα πακέτο που στέλνει κανείς με το ACKNOWLEDGEMENT NUMBER στα πακέτα που λαμβάνει (και όχι σε αυτά που στέλνει που σχετίζονται με την αρίθμηση της απέναντι πλευράς). Σημειωτέον ότι οι επιβεβαιώσεις έχουν συσωρευτικό χαρακτήρα (δηλ. επιβεβαιώνουν και όλα τα προηγούμενα δεδομένα που έχουν ληφθεί).

Το πεδίο HLEN περιέχει το μήκος του header γιατί αυτό μπορεί να διαφέρει ανάλογα με τις επιλογές που περιέχονται στο segment και που περιγράφονται στο πεδίο OPTIONS. Το πεδίο RESERVED περιέχει 6 bits που έχουν δεσμευθεί για πιθανή μελλοντική χρήση.

Το πεδίο CHECKSUM χρησιμοποιείται για τον έλεγχο λαθών στο τμήμα (επικεφαλίδα και δεδομένα). Το πεδίο CODE BITS των 6-bit χρησιμοποιείται από το TCP για να αναγνωρίζει το είδος του τμήματος.

Η σημασία των bits από αριστερά προς τα δεξιά είναι η ακόλουθη:

bit πεδίου CODE BITS (αριστερά προς τα δεξιά)	Σημασία του bit όταν είναι 1
URG	Το πεδίο URGENT POINTER είναι ενεργό
ACK	Το πεδίο AKNOWLEDGEMENT No είναι ενεργό
PSH	Το segment αυτό πρέπει να παραδοθεί άμεσα στην εφαρμογή χωρις να περιμένει να γεμίσει ο buffer (λειτουργία PUSH)
RST	επαναφορά (reset) της σύνδεσης (εκ νέου αρίθμηση)
SYN	Συγχρονισμός αριθμών ακολουθίας (Sequence No)
FIN	Ο πομπός έχει φτάσει στο τέλος των δεδομένων που έχει να στείλει (κλείσιμο σύνδεσης)

Δηλαδή, οι σημαίες SYN, ACK και URG υποδεικνύουν να εξετασθεί το περιεχόμενο αντιστοίχως των πεδίων SEQUENCE NUMBER, ACKNOWLEDGEMENT NUMBER και URGENT POINTER. Η λειτουργία αποστολής επειγόντων (urgent) δεδομένων δίνει ένα τρόπο να σταλεί ένα σήμα διακοπής (π.χ. από πάτημα πλήκτρου DEL ή ESC) στον δέκτη και ο URGENT POINTER δείχνει τη θέση των δεδομένων αυτών, ωστε να επιτευχθεί η διακοπή χωρίς να διαταραχθεί η ροή του TCP. Η σημαία RST υποδηλώνει ότι χάθηκε η σωστή αρίθμηση της ακολουθίας και ζητείtai να αρχίσει εκ νέου η αρίθμηση. Το πεδίο WINDOW ADVERTISEMENT χρησιμοποιείται για την αναγγελία παραθύρου δηλ για το μέγεθος του παραθύρου που θα χρησιμοποιεί ο πομπός για τη λειτουργία ελέγχου ροής. Τέλος το πεδίο PADDING (έρμα) περιέχει μηδενικά bits και μπαίνουν τόσα κάθε φορά όσα απαιτούνται για να είναι το συνολικό μήκος της επικεφαλίδας πολλαπλάσιο των 32 bits.

Στο πεδίο OPTIONS περιγράφονται οι προδιαγραφές που έχουν συμφωνήσει τα πρωτόκολλα TCP των δύο άκρων, για την σύνδεση. Μια σημαντική επιλογή είναι ο καθορισμός του μέγιστου τμήματος (segment) που μπορεί να μεταφερθεί, ώστε υπολογιστές που διαθέτουν μικρό ελεύθερο χώρο για την προσωρινή αποθήκευση των bytes, να μπορούν να επεξεργάζονται τα segment που στέλνει ο πομπός. Απ' την άλλη πλευρά η επιλογή του μέγιστου μήκους τμήματος είναι σημαντική και για την απόδοση του δικτύου. Πολύ μικρά η πολύ μεγάλα τμήματα δεν χρησιμοποιούν αποδοτικά το δίκτυο, με αποτέλεσμα να έχουμε μικρό ρυθμό μετάδοσης.

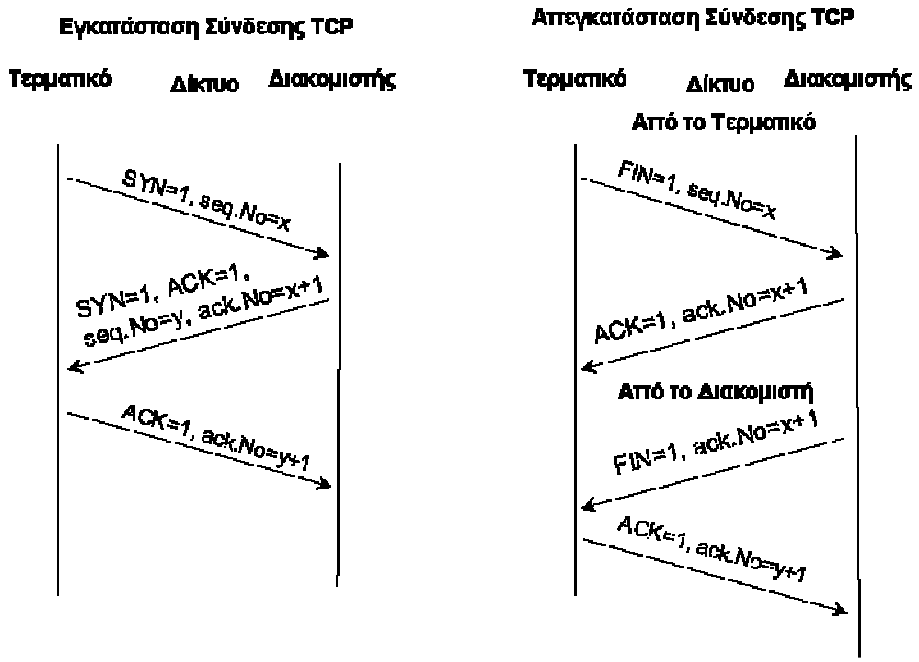
Θεωρητικά το βέλτιστο μέγεθος για το τμήμα (segment) είναι εκείνο που μπορεί να μεταφερθεί από όσο δεδομενόγραμμα IP μήκους ίσου με το μέγιστο υποστηριζόμενο, χωρίς δηλαδή να χρειαστεί σε κάποιο σημείο της διαδρομής από το πομπό στο δέκτη να κατατμηθεί σε μικρότερα δεδομενογράμματα. Ο υπολογισμός του βέλτιστου μεγέθους είναι αντικείμενο έρευνας και μέχρι σήμερα δεν υπάρχει κάποιο καθιερωμένο μέγιστο μήκος τμήματος.

Όπως είπαμε στο πεδίο ACKNOWLEDGEMENT NUMBER υπάρχει ο αριθμός του επόμενου byte που περιμένει να του στείλει η άλλη πλευρά. Αυτή είναι μια γενική σύμβαση του πρωτοκόλλου TCP. Ο πομπός αριθμεί τα bytes που θέλει να στείλει και την αρίθμηση αυτή (sequence number) χρησιμοποιεί ο δέκτης για να κατασκευάσει ακριβώς την ίδια ακολουθία από bytes που του έστειλε ο πομπός. Με τον τρόπο αυτό αντιμετωπίζονται περιπτώσεις όπου δεδομένα έχουν χαθεί, ή έχουν

σταλεί με διαφορετική σειρά. Κάθε φορά ο δέκτης επιβεβαιώνει το μέγιστο byte της ακολουθίας που έχει λάβει σωστά. Ωστόσο αυτό γίνεται *γράφοντας στο πεδίο ACKNOWLEDGEMENT* **το επόμενο byte που περιμένει να λάβει από τον πομπό**.

5.7 Εγκατάσταση και τερματισμός σύνδεσης TCP

Για την εγκατάσταση της επικοινωνίας, το πρωτόκολλο TCP χρησιμοποιεί τη μέθοδο της τρίοδης χειραψίας (three-way handshake) η οποία ολοκληρώνεται σε τρείς φάσεις. Στην απλούστερη περίπτωση η διαδικασία της χειραψίας είναι αυτή που φαίνεται στο σχήμα 5.5.α



Σχήμα 5.5.α (αριστερά): Η ακολουθία των μηνυμάτων (segment) που ανταλλάσσονται σε μια τρίοδη χειραψία (three-way handshake). **Σχήμα 5.5.β (δεξιά):** Η ακολουθία για τοκλείσιμο της σύνδεσης. (Τα SYN, ACK σημαίνουν ότι τα αντίστοιχα bits στο πεδίο CODE του segment είναι '1')

Το πρώτο τμήμα (segment) της χειραψίας ξεχωρίζει, γιατί έχει το στο πεδίο CODE το bit SYN '1' υποδεικνύοντας στον δέκτη να συντονίσει τις επιβεβαιώσεις του στην αρχική τιμή του seq.No=x. Η απάντηση έχει τόσο τη σημαία SYN όσο και την σημαία ACK ίση με '1', δείχνοντας την έναρξη της δικής του αρίθμησης από το πεδίο seq.No=y και ότι έχει αποδεχθεί το πρώτο τμήμα επιβεβαιώνοντας με το πεδίο ack.No=x+1. Το τρίτο τμήμα της χειραψίας έχει μόνο το bit ACK '1' που επιβεβαιώνει την αρίθμηση του διακομιστή από την τιμή y (θέτοντας ack.No=y+1) και πλέον έχουν και οι δύο πλευρές συμφωνήσει και έχει εγκατασταθεί η σύνδεση TCP μεταξύ τους.

Συνήθως το λογισμικό του TCP στον ένα υπολογιστή περιμένει για το πρώτο μήνυμα χειραψίας (handshake) ενώ η άλλη πλευρά αρχικοποιεί τη διαδικασία εγκατάστασης επικοινωνίας. Η διαδικασία χειραψίας όμως έχει σχεδιαστεί να δουλεύει σωστά ακόμα και εάν και οι δύο πλευρές δοκιμάσουν να κάνουν εγκατάσταση επικοινωνίας ταυτόχρονα. Έτσι η διαδικασία επικοινωνίας μπορεί να αρχίσει είτε από τη μία πλευρά, είτε απ' την άλλη, είτε τέλος και από τις δύο. Αφού εγκατασταθεί η επικοινωνία τα δεδομένα ανταλλάσσονται και στις δύο κατευθύνεις, δεν έχουμε δηλαδή master και slave.

Η επιλογή να γίνεται η χειραψία σε τρία στάδια έγινε ώστε να έχουμε σωστό συγχρονισμό μεταξύ των δύο πλευρών, εφόσον το πρωτόκολλο TCP βρίσκεται πάνω από μια υπηρεσία που δεν εγγυάται την αξιόπιστη μετάδοση πακέτων. Επομένως και τα πακέτα εγκατάστασης σύνδεσης μπορεί να χαθούν, να σταλούν σε λανθασμένη σειρά, η να σταλούν περισσότερο από μία φορά. Πρόβλημα λοιπόν κατά την εγκατάσταση της επικοινωνίας μπορεί να έχουμε όταν παραδείγματος χάριν, αιτήσεις εγκατάστασης που έχουν αναμεταδοθεί φτάσουν στο δέκτη ενώ έχει ήδη αρχίσει να εγκαθίσταται η επικοινωνία, η ακόμα, (αν υπάρχει αρκετή καθυστέρηση στο δίκτυο), φτάσουν όταν έχει ήδη εγκατασταθεί και τελειώσει η σύνδεση. Για να λύσει τα προβλήματα αυτά το πρωτόκολλο TCP εκτός από την τρίοδη χειραψία, χρησιμοποιεί και την εξής αρχή : *Αφού έχει εγκατασταθεί επικοινωνία μεταξύ δύο πλευρών αγνοούνται όλες τις πρόσθετες αιτήσεις για επικοινωνία που ίσως φτάσουν.*

Το πρωτόκολλο TCP λοιπόν εξασφαλίζει κατά την εγκατάσταση της επικοινωνίας δύο σημαντικές λειτουργίες. Εγγυάται ότι και οι δύο πλευρές είναι έτοιμες να στείλουν δεδομένα (και επιπλέον η κάθε πλευρά γνωρίζει ότι η άλλη πλευρά είναι έτοιμη) και επιτρέπει στις δύο πλευρές να συμφωνήσουν στην αρχική αρίθμηση της σειράς των bytes που θα ανταλλάξουν (initial sequence numbers). Η αρίθμηση αυτή επιλέγεται τυχαία και δεν μπορεί να ξεκινά πάντα από τον ίδιο αριθμό. Δηλαδή το πρωτόκολλο TCP δεν μπορεί για κάθε σύνδεση που κάνει να ξεκινά την αρίθμηση των bytes από 1. Είναι λοιπόν σημαντικό οι δύο πλευρές να συμφωνήσουν στην αρχική αρίθμηση των οκτάδων που θέλουν να στείλουν ώστε τα μηνύματα ACK να επιβεβαιώνουν το σωστό byte.

Σημειωτέον ότι το πρωτόκολλο TCP επιτρέπει στο τμήμα που αρχικοποιεί την διαδικασία εγκατάστασης να περιέχει και δεδομένα (δηλ. δεν πρέπει να ολοκληρωθεί η χειραψία για να σταλούν δεδομένα).

Τερματισμός σύνδεσης TCP

Ο τερματισμός μιας σύνδεσης TCP πρέπει να γίνει και από τις δύο πλευρές όπου η κάθε μια κλείνει την σύνδεση από τη δική της κατεύθυνση. Επειδή το πρωτόκολλο TCP επιτρέπει πλήρως αμφίδρομη επικοινωνία, όταν η μία πλευρά δεν έχει να στείλει άλλα δεδομένα πρέπει να κλείσει την επικοινωνία μόνο προς τη μία κατεύθυνση. Για να γίνει αυτό, αφού στείλει και τα τελευταία δεδομένα περιμένει να πάρει επιβεβαίωση γι' αυτά. Στη συνέχεια στέλνει ένα τμήμα με το bit FIN στο πεδίο CODE '1'. Το πρωτόκολλο TCP απ' την άλλη πλευρά αναγνωρίζει το segment FIN και πληροφορεί την εφαρμογή ότι δεν υπάρχουν άλλα δεδομένα να λάβει. Αφού κλείσει μια σύνδεση σε κάποια κατεύθυνση, το TCP αρνείται να λάβει άλλα δεδομένα προς αυτή την κατεύθυνση. Τα δεδομένα όμως προς την άλλη κατεύθυνση μεταδίδονται κανονικά μέχρι και η άλλη πλευρά να ζητήσει τερματισμό της σύνδεσης. Εδώ πρέπει να τονιστεί ότι αν και η σύνδεση προς τη μία κατεύθυνση έχει τερματιστεί τα τμήματα επιβεβαίωσης (ACK) στέλνονται κανονικά. Όταν η σύνδεση τερματιστεί και στις δύο κατευθύνσεις το TCP διαγράφει τα τερματικά σημεία (end-points) της σύνδεσης.

Στο σχήμα 5.5.β φαίνεται η διαδικασία τερματισμού μιας σύνδεσης TCP.

Η διαφορά μεταξύ της διαδικασίας της χειραψίας, κατά την εγκατάσταση και κατά τον τερματισμό μιας σύνδεσης βρίσκεται στο πρώτο τμήμα που λαμβάνει η άλλη πλευρά. Αντί το πρωτόκολλο TCP να ενημερώσει αμέσως την εφαρμογή για το τμήμα FIN που έλαβε, πρώτα στέλνει ένα ACK για το FIN και ύστερα ενημερώνει την εφαρμογή ότι η άλλη πλευρά δεν έχει άλλα δεδομένα να στείλει. Στην περίπτωση εγκατάστασης επικοινωνίας είναι απαραίτητο το TCP να περιμένει την εφαρμογή να αποδεχθεί την αίτηση κλήσης. Στην διαδικασία τερματισμού η άλλη πλευρά χρειάζεται απλώς να ενημερωθεί, για τον τερματισμό της σύνδεσης, επομένως το πρωτόκολλο TCP μπορεί να στείλει αμέσως ένα μήνυμα επιβεβαίωσης. Το να περίμενε την εφαρμογή να αποκριθεί στο τμήμα FIN όχι μόνο θα ήταν περιττό άλλα ίσως προκαλούσε την αναμετάδοση του τμήματος FIN απ' την άλλη πλευρά, αφού η εφαρμογή μπορεί να αργήσει να απαντήσει στο πρωτόκολλο TCP.

Κεφάλαιο 6

ΟΙ ΜΗΧΑΝΙΣΜΟΙ ΕΛΕΓΧΟΥ ΚΙΝΗΣΗΣ ΤΟΥ TCP

Το TCP όπως λέει και τ' όνομά του έχει σχεδιασθεί για να ελέγχει την εκπεμπόμενη προς το δίκτυο κίνηση (traffic) με σκοπό να πραγματοποιεί μια σειρά από λειτουργίες με κυριότερες τον έλεγχο ροής, διόρθωση λαθών και έλεγχο συμφόρησης του δικτύου. Για να επιτελέσει όλες αυτές τις λειτουργίες που του έχουν ανατεθεί, όπως όλα τα πρωτόκολλα διαθέτει μια σειρά από μηχανισμούς που βασίζονται σε αλγορίθμους ελέγχου κίνησης και υλοποιούνται με το λογισμικό που τρέχει στα τερματικά. Αυτοί οι μηχανισμοί που σαν βάση έχουν το παράθυρο ολίσθησης και χαρακτηρίζονται από ιδιαίτερα δυναμική συμπεριφορά, αφού λαμβάνουν υπ' όψη όλα όσα συμβαίνουν στο δίκτυο, θα αναλυθούν σε αυτό το κεφάλαιο. Είναι αρκετά πολύπλοκοι μηχανισμοί και κρύβουν πολυετή εμπειρία αφού είναι προϊόν συνεχούς εξέλιξης. Μέρος αυτής της ιστορίας θα παρουσιασθεί στην συνέχεια διότι έτσι γίνεται ευκολότερη η κατανόηση, αντί της κατ' ευθείαν εισαγωγής στα ισχύοντα σήμερα. Εξ άλλου ανά πάσα στιγμή βρίσκονται σε χρήση ένα μείγμα από παλαιές και νέες υλοποιήσεις, αφού το Διαδίκτυο είναι εξαιρετικά πολύπλοκο και κάθε βελτίωση παίρνει πολλά χρόνια να εκτοπίσει εντελώς τις παλιές λύσεις. Έτσι συνυπάρχουν συνήθως πολλές παραλλαγές του ιδίου μηχανισμού και προτού εκτοπισθεί οριστικά μια γενιά έχει ήδη αρχίσει να εισάγεται η νέα πράγμα που καθιστά το Διαδίκτυο ένα δυναμικά εξελισσόμενο και προσαρμοζόμενο οργανισμό επικοινωνίας.

Οι μηχανισμοί ελέγχου βασίζονται σε μετρήσεις χρόνων και ιδιαίτερα της μετ' επιστροφής διάρκειας ταξιδιού των πακέτων (Round trip times), γι' αυτό θα αρχίσουμε με τους χρονιστές του TCP.

6.1 Οι χρονιστές του TCP

Όπως σε όλα τα πρωτόκολλα μεγάλο ρόλο στην εκτέλεση και κατανόηση του TCP παίζουν οι χρονιστές που έχουν αποστολή, όχι μόνο να αποτρέπουν τα αδιέξοδα, αλλά και να βελτιστοποιούν την λειτουργικότητα. Αδιέξοδο π.χ. μπορεί να δημιουργηθεί όταν μετά από ένα τμήμα με αναγγελία παραθύρου 0 σταλεί ένα επόμενο τμήμα που επιχειρεί να ξανανοίξει το παράθυρο και αυτό χαθεί. Τότε και οι δύο πλευρές θα βρεθούν να περιμένουν επ' άπειρον η μία την άλλη. Το πρόβλημα αυτό λύνει ο χρονιστής εμμονής (persistence timer). Αλλά ας δούμε συνοπτικά τους κυριότερους χρονιστές.

6.1.1. Χρονιστής Εμμονής (persistence timer)

Αντιμετωπίζει το προαναφερθέν πρόβλημα στέλνοντας μικρά τμήματα του ενός οκτέτου κάθε 5 δευτερόλεπτα βολιδοσκοπώντας εάν ο δέκτης είναι έτοιμος να δεχθεί δεδομένα πάλι. Εάν δεν είναι, ο δέκτης στέλνει πάλι παράθυρο 0, αλλιώς μη μηδενικό παράθυρο που οδηγεί σε επανεκκίνηση των αποστολών δεδομένων.

6.1.2. Χρονιστής επαναποστολής (retransmission timer)

Αντιμετωπίζει το πρόβλημα της καθυστέρησης λήψης επιβεβαίωσης (ενδεχόμενη απώλεια του ACK) και με την εκπνοή του ξαναστέλνεται το πακέτο. Η τιμή του δεν είναι σταθερή διότι δεν είναι γνωστή η διαδρομή και ο χρόνος ταξιδιού μετ' επιστροφής (RTT). Έτσι αποφασίζεται διαφορετική τιμή για κάθε σύνδεση στη βάση δυναμικών μετρήσεων του RTT.

Επειδή η τιμή κάθε μετάβασης είναι διαφορετική κάθε φορά ακόμα και για το ίδιο προορισμό ανάλογα με το πόσο χρόνο μένουν τα πακέτα στους ταμιευτήρες, η τιμή που χρησιμοποιείται προκύπτει μετά από συνεχείς μετρήσεις και εκτίμηση ενός κινητού Μέσου Όρου που προκύπτει από τον αναδρομικό τύπο:

$RTT_k = aRTT_k + (1-a)RTT_{k-1}$ όπου ο συντελεστής a είναι τυπικά 0.1

Δηλαδή μετά την μέτρηση k βελτιώνεται ο μέχρι τότε μέσος όρος με χρήση κατά 10% της νέας τιμής και 90% του παλαιού Μ.Ο που είχε προκύψει μετά την k-1 μέτρηση.

Εκτός ωστόσο από το RTT, για την τιμή του χρονιστή λαμβάνονται υπ' όψη και η τυπική απόκλιση των μετρήσεων με σχετικά πολύπλοκους λογαριασμούς που ωστόσο αποφεύγουν τη χρήση δεκαδικών αριθμών και εκτελούνται με αριθμητική ακεραίων. Οι τιμές του RTT παίζουν σημαντικό ρόλο και στην εφαρμογή των αλγορίθμων αποφυγής συμφόρησης όπως θα δούμε πιο κάτω.

6.1.3. Χρονιστής Σιγής (quiet timer)

Πολλές φορές όταν λήγει μια σύνδεση TCP υπάρχουν ακόμα πακέτα με τους χρησιμοποιηθέντες αριθμούς θυρών (port numbers) που κυκλοφορούν για μερικά ακόμη δευτερόλεπτα. Για αποφυγή συγχύσεως, εάν επιχειρηθεί να ανοιχθεί σύνδεση με τους ίδιους αριθμούς θυρών το σύστημα τους κρατά δεσμευμένους μέχρι την εκπνοή του χρονιστή (30 δευτ.).

6.1.4. Χρονιστής «επιβίωσης» και χρονιστής αδράνειας (keep-alive and idle timer)

Πολλές φορές μια σύνδεση κλείνει από την μία πλευρά χωρίς ειδοποίησης της άλλης. Π.χ. ένας πελάτης (client) κλείνει τον φυλλομετρητή (browser) του χωρίς logout. Για να μη μείνει επ' αόριστο ημι-ανοικτή η σύνδεση από τη μεριά του εξυπηρετητή (server), αποστέλλεται ένα άδειο πακέτο κάθε φορά που εκπνέει ο χρονιστής ελέγχου ζωής και ταυτόχρονα τίθεται ο χρονιστής αδράνειας. Εάν υπάρξει απάντηση προτού εκπνεύσει ο χρονιστής αδράνειας, (6 λεπτά) η σύνδεση διατηρείται ζωντανή (επιβιώνει) για τουλάχιστον άλλο τόσο χρόνο, αλλιώς τερματίζεται (εξ ού και το αγγλικό όνομα «keep alive-κράτα ζωντανή».

ΤΥΠΙΚΕΣ ΤΙΜΕΣ ΧΡΟΝΙΣΤΩΝ

ΧΡΟΝΙΣΤΗΣ	*ΤΙΜΗ (δευτ/λεπτα)*
Χρονιστής Εμμονής (persistence timer)	5
Χρονιστής επαναποστολής (retransmission timer)	(κυμαινόμενη)
Χρονιστής Σιγής (quiet timer)	30
Χρονιστής ζωντάνιας (keep-alive timer)	45
Χρονιστής αδρανείας (idle timer)	360

6.2 Έλεγχος ροής μέσω του παραθύρου και των μηχανισμών του TCP

Σε σχέση με τον απλό μηχανισμό παραθύρου ολίσθησης του HDLC/LAPB, ο αντίστοιχος μηχανισμός που χρησιμοποιείται από το TCP διαφέρει στο γεγονός ότι επιτρέπει το μήκος του παραθύρου να μεταβάλλεται δυναμικά ώστε να επιτελεί πιο πολύπλοκες λειτουργίες υπό τον έλεγχο του λογισμικού. Σε κάθε επιβεβαίωση (ACK-acknowledgement) εκτός απ΄ τον αριθμό των οκτάδων που έχει λάβει σωστά, ο δέκτης στέλνει επίσης και μια αναγγελία παραθύρου (window advertisement), που καθορίζει τον αριθμό των οκτάδων που μπορεί να λάβει από το πομπό. Η αναγγελία παραθύρου είναι μια ένδειξη της κατάστασης του ταμιευτήρα (buffer) του δέκτη, δηλαδή το μέγεθος του ελεύθερου χώρου που διαθέτει. Σαν απάντηση μιας αύξησης στην αναγγελία, ο πομπός θα αυξήσει το μήκος του παραθύρου στέλνοντας περισσότερα bytes χωρίς να περιμένει επιβεβαίωση. Αντίστοιχα, αν ο δέκτης ζητήσει μείωση του παραθύρου με την αναγγελία, ο πομπός θα μειώσει το μήκος του παραθύρου και επομένως και τα bytes που θα στέλνει. Με τη διαδικασία αυτή, το πρωτόκολλο TCP πετυχαίνει αποδοτικό έλεγχο ροής. Στην ακραία περίπτωση που ο δέκτης δεν έχει καθόλου ελεύθερο χώρο για να αποθηκεύσει, νέα δεδομένα μπορεί να στείλει μηδέν στην αναγγελία με αποτέλεσμα ο πομπός να σταματήσει την αποστολή νέων δεδομένων μέχρι να λάβει νέα μη μηδενική αναγγελία. Αυτό θα γίνει μόλις ο δέκτης έχει επεξεργαστεί τα bytes που έχει αποθηκευμένα στον καταχωριστή και μπορεί να δεχθεί και άλλα..

Η ύπαρξη ενός μηχανισμού για έλεγχο ροής, είναι απαραίτητη για το περιβάλλον του Internet όπου είναι συνδεδεμένοι υπολογιστές διαφορετικής ταχύτητας και μεγέθους, οι οποίοι επικοινωνούν μέσω δικτύων που επίσης διαφέρουν στην ταχύτητα και διοχέτευση (throughput). Στην πραγματικότητα μπορούμε να διακρίνουμε δύο ανεξάρτητα προβλήματα στην ροή των δεδομένων που πρέπει να αντιμετωπίσει ένα πρωτόκολλο Internet. Κατ'αρχήν πρέπει να εξασφαλίζει έλεγχο ροής από άκρη σε άκρη (end to end). Για παράδειγμα όταν επικοινωνούν μέσω του Internet ένα μικροϋπολογιστής μ' έναν υπολογιστή μεγάλης ισχύος (mainframe), ο μικροϋπολογιστής, χρειάζεται να έχει έναν μηχανισμό για να ρυθμίζει τη ροή των δεδομένων που στέλνει ο μεγάλος υπολογιστής, διαφορετικά πολύ γρήγορα θα υπερφορτωθεί.

Όπως αναφέρθηκε, το πρωτόκολλο TCP ανταλλάσσει τα δεδομένα με τη μορφή υποσυνόλων από bytes που λέγονται τμήματα (segments). Το τμήμα είναι η βασική μονάδα ανταλλαγής των δεδομένων για το TCP, αν και οι επιβεβαιώσεις και οι αναγγελίες αναφέρουν bytes.

Τα μηνύματα επιβεβαίωσης μπορεί να δικαιολογούν την αποστολή αυτοτελούς τμήματος (που θα γίνει πακέτο) αλλά όταν υπάρχουν και δεδομένα προς αποστολή το πρωτόκολλο TCP χρησιμοποιεί την τεχνική της υπέρθεσης (piggybacking). όπως και το X.25. Δηλαδή στο τμήμα που στέλνει ένας υπολογιστής Α σε κάποιον Β περιέχονται εκτός των δεδομένων και τα πεδία ελέγχου με επιβεβαίωση για τα δεδομένα που έστειλε ο Β στον Α.

6.3 Οι μηχανισμοί ελέγχου συμφόρησης του TCP

Συμφόρηση (congestion) ονομάζεται η κατάσταση ενός δικτύου ή τμήματος δικτύου όταν δεν μπορεί να διαβιβάσει όλη την κίνηση που δέχεται, δηλ. η εισερχόμενη σε κάποιο υποτμήμα του δικτύου κίνηση είναι λιγότερη από αυτή που εξέρχεται συν την αύξηση της αποθήκευσης στο τμήμα αυτό σε κάποιο χρονικό διάστημα παρατήρησης. Εσωτερικά στο υποδίκτυο η συμφόρηση εκδηλώνεται σε κάποιες θύρες εξόδου οι οποίες δέχονται περισσότερη κίνηση απ' όση μπορούν να διαβιβάσουν με αποτέλεσμα να αυξάνει η πληρότητα των ταμιευτήρων προσωρινής αποθήκευσης που διαθέτουν. Εάν αυτή η ανισορροπία είναι προσωρινή και ανατραπεί προτού υπερχειλίσει ο ταμιευτήρας, έχουμε μια στιγμιαία υπερφόρτωση που είναι μια φυσιολογική κατάσταση στα δίκτυα πακέτων, αλλά εάν φθάσουμε σε υπερχείλιση, έχουμε εκδήλωση συμφόρησης.

Πολλοί μηχανισμοί έχουν προταθεί και δοκιμασθεί για τον έλεγχο συμφόρησης. Ένας άμεσος μηχανισμός είναι να θέτει ο κόμβος ένα bit αναγγελίας συμφόρησης το οποίο φθάνει στον προορισμό ο οποίος με τη σειρά του στέλνει πίσω στην πηγή την ειδοποίηση να ελαττώσει το ρυθμό. Ακόμα καλύτερα είναι να σταλεί η αναγγελία κατευθείαν πίσω στην πηγή σε ένα πακέτο που επιστρέφει. Η υλοποίηση όμως αυτής της λύσης είναι πολύπλοκη αφού ο κόμβος πρέπει να είναι σε θέση να παρακολουθεί ροές πακέτων δηλαδή να συσχετίζει προελεύσεις και προορισμούς. Αυτό είναι δύσκολο και χρονοβόρο και στην περίπτωση του IP αντίθετο με τη βασική φιλοσοφία δρομολόγησης που γίνεται βήμα-βήμα (hop-by-hop) χωρίς συνδέσεις.

Ο τρόπος που πραγματοποιεί **έλεγχο συμφόρησης** (congestion control) το TCP είναι από τα πιο χαρακτηριστικά και επιτυχή στοιχεία του πρωτοκόλλου. Είναι ένας μηχανισμός προσαρμοσμένος πλήρως στη φιλοσοφία σχεδίασης του Διαδικτύου όπου τα υποκείμενα φυσικά δίκτυα είναι ανομοιογενή και αντιμετωπίζονται σαν αναξιόπιστα μέσω του πρωτοκόλλου IP του στρώματος δικτύου. Έτσι πρέπει το TCP να αντιμετωπίσει τα αναπόφευκτα για κάθε δίκτυο προβλήματα συμφόρησης κατά τρόπο ανεξάρτητο των τυχόν μηχανισμών που διαθέτουν αυτά τα δίκτυα. Σε ομοιογενή δίκτυα μπορούν να χρησιμοποιηθούν σήματα επικείμενης ή προϊούσης συμφόρησης που εκδίδονται από τους κόμβους όπου εκδηλώνεται η συμφόρηση και σαν μέτρο της μπορεί να χρησιμοποιηθεί ο βαθμός πλήρωσης των ταμιευτήρων. Τέτοιο σήμα π.χ. είναι το ECN (explicit congestion notification δηλ. άμεση γνωστοποίηση συμφόρησης) του frame relay ή του ATM. Ωστόσο ένα τέτοιο σήμα δεν μπορεί να χρησιμοποιηθεί σε δίκτυα άλλου τύπου διότι λειτουργούν διαφορετικά οι κόμβοι τους.
Με άλλα λόγια το τίμημα της ανοχής της ανομοιογένειας είναι ότι το TCP πρέπει να κάνει τον έλεγχο χωρίς βοήθεια από τους ενδιάμεσους κόμβους έκαστος των οποίων μπορεί να χρησιμοποιεί διαφορετικές δικτυακές τεχνολογίες. Επελέγησαν λοιπόν έμμεσοι τρόποι ανίχνευσης της συμφόρησης που βασίζονται στην καθυστέρηση και στην απώλεια πακέτων (ένδειξη υπερχείλισης ταμιευτήρων), δεδομένου ότι οι δύο ακραίοι σταθμοί που εκτελούν το πρωτόκολλο δεν έχουν άμεσο τρόπο να γνωρίζουν τι γίνεται στους ταμιευτήρες των δρομολογητών. Όταν εντοπισθεί συμφόρηση, ο μηχανισμός ελέγχου αντιδρά με βάση ειδικούς αλγορίθμους που αξιοποιούν τις παρατηρήσεις και έμμεσες μετρήσεις που πραγματοποιούν.
Υπάρχει άλλος ένας λόγος που συνηγορεί υπέρ της έμμεσης μεθόδου ελέγχου συμφόρησης από τα άκρα (end-to-end): οι κόμβοι μπορούν να παραμένουν απλοί με όσο το δυνατόν λιγότερες λειτουργίες δεδομένου του τεράστιου αριθμού πακέτων που έχουν να επεξεργαστούν. Αντίθετα, οι τερματικοί σταθμοί έχουν συνήθως πλεονάζουσα ισχύ διότι εξυπηρετούν τις πιο πολλές φορές ένα ή λίγους χρήστες.

Ο πρώτος μηχανισμός που σχεδιάστηκε από τον Van Jacobson εισήχθη 8 χρόνια μετά την εισαγωγή του TCP/IP δηλ. γύρω στο 1988-89 όταν τα προβλήματα συμφόρησης είχαν αρχίσει να γίνονται έντονα. Μέχρι τότε όταν αρχίζανε οι απώλειες πακέτων, οι ξενιστές επιδείνωναν την κατάσταση αντί να την ανακουφίζουν, διότι μόλις εξέπνεε ο χρονιστής χωρίς να έχει έλθει επιβεβαίωση για κάποια πακέτα, απλά τα ξαναστέλνανε. Μετρήσεις που έγιναν διαπίστωσαν ότι κατά τη συμφόρηση, τα μισά περίπου πακέτα ήσαν επαναποστολές απολεσθέντων πακέτων.
Η κεντρική ιδέα είναι να χρησιμοποιεί ο αλγόριθμος του TCP την άφιξη κάθε επιβεβαίωσης (ACKs) σαν ένδειξη ότι το πακέτο διαβιβάστηκε επιτυχώς και άρα μπορεί να στείλει ένα νέο με τη βεβαιότητα ότι δεν προκαλεί συμφόρηση. Το TCP επιχειρεί περαιτέρω να εκτιμήσει από το ρυθμό των αφίξεων επιβεβαιώσεων

τη διαπερατότητα (throughput) που μπορεί να υποστηρίξει το δίκτυο, ούτως ώστε να εισάγει τα πακέτα μέχρι το ρυθμό που δεν προκαλεί συμφόρηση. Πρέπει για να σχηματίσουμε μια εικόνα του πόσο δυναμική και πολύπλοκη είναι η κατάσταση να σκεφθούμε πόσες ροές (συνδέσεις TCP) συναντώνται στους ταμιευτήρες κάθε πόρτες εξόδου και πως αυτές εγκαθίστανται και αποσύρονται ή απλά αλλάζουν το ρυθμό τους καθώς οι αντίστοιχοι χρήστες επισκέπτονται ιστοσελίδες ή ζητούν αρχεία, εικόνες βίντεο, κτλ.
Η προσπάθεια των αλγορίθμων ελέγχου συμφόρησης του TCP είναι να φέρουν την κάθε ροή και χρήστη να μοιράζεται δίκαια, δηλαδή ίσα τη διαθέσιμη χωρητικότητα του δικτύου. (Τα πράγματα γίνονται πιο πολύπλοκα με τις νέες υπηρεσίες όπου ο ίσος μερισμός παύει να θεωρείται δίκαιος διότι π.χ. με ένα χρήστη να έχει συνομιλία φωνής (VoIP-voice over IP) και ένα άλλο να παρακολουθεί βίντεο, ίση διαπερατότητα 16kbps επιτρέπει στον πρώτο να κάνει τη δουλειά του ενώ ο δεύτερος δεν βλέπει τίποτα. Ωστόσο για την εποχή της υπηρεσίας «κατά δύναμη» (best effort) ο ισομερισμός ήταν επαρκής για να αποδώσουμε την ιδιότητα της δικαιοσύνης (fairness) στην υπηρεσία του Διαδικτύου. Η ιδιότητα αυτή είναι μία από τις βασικές απαιτήσεις σχεδίασης ενός πρωτοκόλλου και στην γενική της μορφή επιβάλλει να μοιράζει δίκαια τους δικτυακούς πόρους (στην εξεταζόμενη περίπτωση το εύρος ζώνης και τις θέσεις ταμιευτήρα).
Προτού μπούμε στην εξέταση των μηχανισμών του TCP είναι σκόπιμο να τονίσουμε ότι οι μηχανισμοί αυτοί δεν είναι κατάλληλοι για υπηρεσίες πραγματικού χρόνου όπου οι αναμεταδόσεις δεν λύνουν το πρόβλημα, διότι δεν είναι επιτρεπτή η αναμονή. Κατά συνέπειαν, η εκδήλωση της συμφόρησης πρέπει να αποφευχθεί εντελώς με πρόβλεψη του φόρτου και αυτό απαιτεί μηχανισμούς με εκ των προτέρων έλεγχο της προσφερόμενης κίνησης που βασίζεται σε συμφωνία δικτύου και πηγής κίνησης (traffic contract). Τέτοιοι μηχανισμοί δεν προβλέπονται στο TCP/IP και γίνονται προσπάθειες εμπλουτισμού με νέα πρωτόκολλα. Από την άλλη, δεν απαιτείται συνήθως σχολαστική διόρθωση λαθών στις υπηρεσίες πραγματικού χρόνου και έτσι στο Διαδίκτυο προκειμένου για κίνηση πραγματικού χρόνου χρησιμοποιούνται «ελαφρά» πρωτόκολλα χωρία αναμεταδόσεις όπως το UDP ή το XTP.

Η παράμετρος η οποία αποτελεί το εργαλείο ποσοτικοποίησης των ενεργειών των αλγορίθμων αποφυγής συμφόρησης δεν είναι άλλη από το παράθυρο ολίσθησης. Προς αποφυγή συγχύσεως πρέπει να τονισθεί ότι ενώ για τον έλεγχο ροής ο δέκτης στέλνει μια αναγγελία παραθύρου, ταυτόχρονα οι αλγόριθμοι υπολογίζουν ένα συνιστώμενο παράθυρο αποφυγής συμφόρησης. (*CongestionWindow* που θα συμβολίζουμε στη συνέχεια με *cwnd)* Ποιο τελικά θα χρησιμοποιήσει ο πομπός; Μα προφανώς το μικρότερο από τα δύο. Δηλαδή για να ελαττώσει το ρυθμό αποστολής θα ελαττώσει το παράθυρο με δική του πλέον πρωτοβουλία ανεξάρτητα αν ο δέκτης έχει επιτρέψει μεγαλύτερο παράθυρο.

Πιο κάτω σκιαγραφούνται οι τρεις κυριότεροι μηχανισμοί που χρησιμοποιούνται σήμερα και που είναι γνωστοί με τα ονόματα Vegas, Tahoe και Reno (από τα ονόματα πόλεων της πολιτείας Νεβάδα). Πολλές άλλες παραλλαγές προτάθηκαν χωρίς ποτέ να φθάσουν στην εφαρμογή.

6.3.1 Η παραλλαγή Vegas

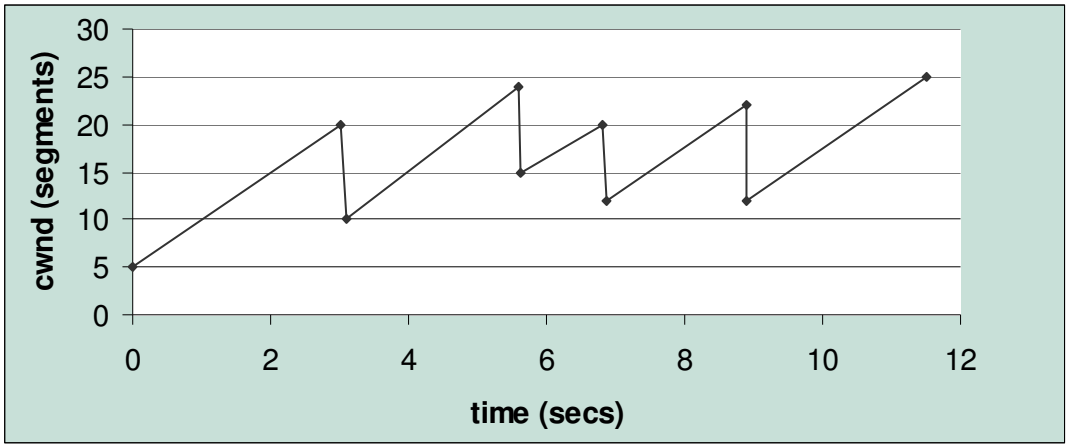
Ο μηχανισμός Vegas είναι ο παλαιότερος αλλά η ονομασία δόθηκε αργότερα όταν εισήχθη ο Tahoe για να διακρίνεται από αυτόν. Η ρύθμιση του παραθύρου δεν βασίζεται σε εκτίμηση απωλειών (αυτό έγινε αργότερα) αλλά στην εκτίμηση της καθυστέρησης (RTT- Round trip time). Το TCP διαθέτει μηχανισμούς συνεχούς εκτίμησης του RTT, αλλά η ακρίβεια των μετρήσεων δεν είναι και τόσο καλή διότι αυτό δεν είναι εύκολο να γίνει χωρίς ακριβά συστήματα ωρολογίων. Πάντως, με βάση αυτή την εκτίμηση, ο αποστολέας υπολογίζει χονδρικά τον αριθμό των πακέτων που είναι εγκλωβισμένα στον ταμιευτήρα στο σημείο της συμφόρησης και προσπαθεί να τα κρατήσει σε ένα χαμηλό αριθμό π.χ. 1-3. Ο αλγόριθμος ξεκινά από την παρατήρηση ότι ο αριθμός των πακέτων που βρίσκονται καθ' οδόν (που είναι «αποθηκευμένα» στο σωλήνα μεταξύ πηγής και δέκτη) είναι ίσος με το γινόμενο Rx(RTT) όπου R είναι ο ρυθμός εκπομπής. Το ελάχιστο RTT, έστω D παρατηρείται όταν οι ταμιευτήρες των ενδιάμεσων κόμβων είναι σχεδόν άδειοι. Οι μεγαλύτερες τιμές περιλαμβάνουν και την καθυστέρηση στους ταμιευτήρες (και κυρίως στο σημείο συμφόρησης). Ο στόχος είναι να είναι το παράθυρο W λίγο μεγαλύτερο από το RD:
Δηλαδή: W=RD+m, όπου m ένας μικρός αριθμός γύρω στο 2.

Ο αλγόριθμος λοιπόν στο Vegas λειτουργεί ως εξής:

αυξάνει το W_{k+1} εάν W_k-R_kD<=1
ενώ το μειώνει εάν W_k-R_kD>=3

Δηλαδή χρησιμοποιεί το W-RD σαν εκτίμηση των εγκλωβισμένων πακέτων στους ταμιευτήρες. Έτσι όταν ο αλγόριθμος συγκλίνει θα έχει πετύχει ο αριθμός των εγκλωβισμένων πακέτων όλων των (έστω N) συνδέσεων που περνούν από το σημείο της στένωσης (bottleneck) να είναι περίπου ο ίδιος (γύρω στο 2-3 με τις ανωτέρω τιμές). Όταν συμβαίνει αυτό ο αλγόριθμος επιτυγχάνει τον δίκαιο μερισμό των πόρων ανάμεσα σε όλες αυτές τις συνδέσεις με ρυθμό R=C/N για κάθε μια όπου C είναι η χωρητικότητα (δηλ. η συνολική ρυθμοδότηση) της στενωπού.
Αξίζει να θυμίσουμε για λόγους σύγκρισης, ότι και το στατικό παράθυρο ολίσθησης του HDLC/LAPB πρέπει να ορισθεί μεγαλύτερο από το συνολικό πλήθος των πλαισίων που χωρούν στη ζεύξη, αλλιώς η απόδοση της ζεύξης πέφτει.
Επίσης ας θυμίσουμε ως αυτονόητο ότι εάν ο δέκτης ζητήσει μικρότερο παράθυρο η τιμή αυτή υπερισχύει διότι ο έλεγχος ροής δεν πρέπει να παραβιαστεί. Μένοντας σε μικρότερο ρυθμό η σύνδεση αυτή παραχωρεί το μέρος του εύρους ζώνης (ρυθμοδότησης) που της αναλογεί στις υπόλοιπες συνδέσεις).
Τέλος αξίζει να παρατηρήσουμε ότι όταν ξεκινά η σύνδεση δεν υπάρχει εκτίμηση του RTT και έτσι το παράθυρο ξεκινά από την τιμή αναγγελίας, πράγμα που μπορεί να δημιουργήσει μεγάλες αρχικές αποκλίσεις από την τελικώς επιθυμητή τιμή σύγκλισης.

Ο αλγόριθμος Vegas ξεπεράστηκε από νέες παραλλαγές που χρησιμοποιούν για ανίχνευση συμφόρησης τις ενδείξεις απωλειών πακέτων (αφού δεν υπάρχει άμεση ειδοποίηση από τον δρομολογητή) που συνίστανται αφ' ενός στην καθυστέρηση της επιβεβαίωσης μέχρις εκπνοής του χρονιστή αναμετάδοσης και αφ 'ετέρου στην πολλαπλότητα των επιβεβαιώσεων (ACKs) δηλ. την εμφάνιση διπλών και τριπλών επιβεβαιώσεων (με τον ίδιο αριθμό επιβεβαίωσης). Αυτό γίνεται διότι και τα πακέτα αλλά και οι επιβεβαιώσεις τους καθυστερούν καθώς εγκλωβίζονται στις μακρές ουρές των ταμιευτήρων. Στη συνέχεια και εφόσον η συμφόρηση επιδεινωθεί θα επισυμβεί εκπνοή του χρονιστή (που ξεκινά σε κάθε εκπομπή πακέτου) χωρίς να έχει φθάσει η επιβεβαίωση από το δέκτη. Ο υπολογιστής από την άλλη μεριά ξαναστέλνει επιβεβαίωση με τον ίδιο αριθμό ACK όταν εκπνεύσει ο δικός του χρονιστής χωρίς να έχει λάβει νέο πακέτο ή όταν λάβει πακέτο εκτός σειράς (ίσως διότι το ενδιάμεσο χάθηκε). Αυτό οδηγεί στην εμφάνιση διπλών και τριπλών ACK, που συνιστά μια άλλη εκδήλωση της συμφόρησης την οποία αξιοποιούν οι αλγόριθμοι.
Ο πομπός αντιδρά σε κάθε εκπνοή με ελάττωση της μεταβλητής *cwnd* (*CongestionWindow)* και μάλιστα με πολλαπλασιαστική μείωση, δηλαδή ρίχνει το παράθυρο συμφόρησης στο μισό. Όταν τα συμπτώματα υποχωρήσουν το παράθυρο ξανα-αυξάνει αλλά όχι πολλαπλασιαστικά αλλά απλά προσθετικά (γραμμικά) κατά ένα τμήμα τη φορά. Αποτέλεσμα είναι το παράθυρο να ακολουθεί την πριονωτή μορφή του κατωτέρω σχήματος 6.1 παραμένοντας γύρω από την επιθυμητή τιμή.
Δυο πολύ σημαντικές ιδιότητες έχουν αποδειχθεί για αυτή την επιτυχή πολιτική της προσθετικής αύξησης και πολλαπλασιαστικής μείωσης. Πρώτον, ότι αποτελεί αναγκαία και ικανή συνθήκη για την ευστάθεια του μηχανισμού ελέγχου συμφόρησης. Δεύτερον, ότι εάν η πολιτική αυτή ακολουθείται από όλες τις ροές, τότε το διατιθέμενο εύρος ζώνης στη στενωπό (bottleneck) της σύνδεσης μοιράζεται δίκαια σε όλες τις ροές.

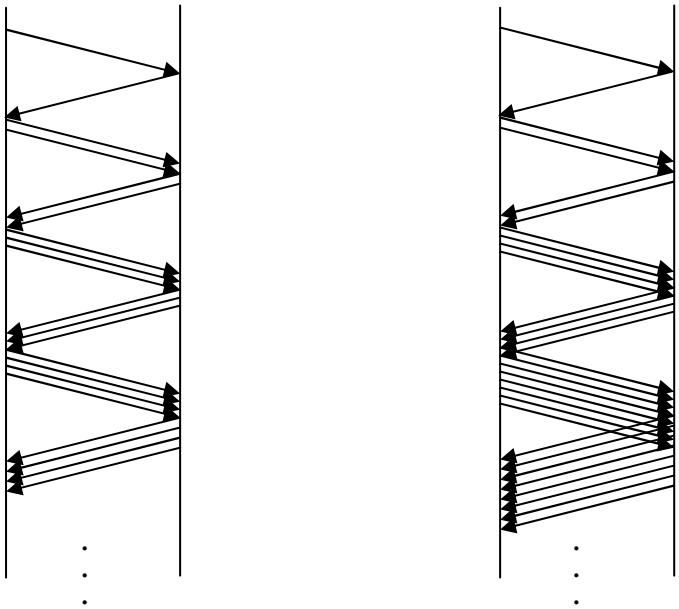


Σχήμα 6.1 Τυπική πορεία παραθύρου συμφόρησης

Η υλοποίηση της εκθετικής αύξησης γίνεται ως εξής: Με κάθε ACK η πηγή αυξάνει το παράθυρο κατά 1 και έτσι σε κάθε RTT που δέχεται w ACKs, το αυξάνει κατά w, δηλαδή το w διπλασιάζεται κάθε RTT δευτερόλεπτα και έτσι το παράθυρο αυξάνει εκθετικά με το χρόνο σύμφωνα περίπου με τη συνάρτηση: $w(t)=2^{t/RTT}$.
Ας σημειωθεί ότι μολονότι το TCP τηρεί όλες τις παραμέτρους σε οκτέτα (bytes), είναι πιο βολικό να περιγράφουμε τις ενέργειες με βάση το τμήμα. Έτσι κι αλλιώς οι πληροφορίες στέλνονται κατά τμήματα με κάθε τμήμα να μπαίνει συνήθως σε ένα πακέτο IP και το παράθυρο δεν επιτρέπεται να πέσει κάτω από το μέγεθος ενός τμήματος (στην ορολογία του TCP Maximum Segment Size-MSS). Αυτή την πρακτική δηλ. να εκφράζουμε το παράθυρο σε τμήματα, θα ακολουθούμε στη συνέχεια

6.3.2 Η παραλλαγή Tahoe

Ενώ η προσθετική αύξηση / πολλαπλασιαστική μείωση λειτουργούν θαυμάσια κοντά στη διαθέσιμη χωρητικότητα του δικτύου, αργούν πολύ να οδηγήσουν στην αποδοτική τιμή. Για να διορθωθεί αυτό εισήχθησαν δύο νέοι αλγόριθμοι (μηχανισμοί) στην επόμενη βαριάντα που είχε το όνομα Tahoe:η αργή εκκίνηση (slow start) και η αποφυγή συμφόρησης (congestion avoidance).
Η αργή εκκίνηση, σε αντίθεση με αυτό που υπονοεί το όνομά της, επιβάλλει κατά την αρχική εκκίνηση της σύνδεσης να αυξάνει εκθετικά και όχι γραμμικά (δηλ. προσθετικά). Ξεκινά με *cwnd* (*CongestionWindow)* ίσο με 1 τμήμα (πακέτο). Μόλις έλθει το ACK, αυξάνει το *cwnd σε* 2 και κάθε φορά που φθάνει μια επιβεβαίωση για προηγούμενα τμήματα, τότε το παράθυρο αυτό αυξάνει κατά 1 με αποτέλεσμα να διπλασιάζεται σε 2, 4, 8 κ.ο.κ., σε κάθε RTT. Εάν συγκρίνουμε τη συμπεριφορά με την αρχική μορφή του πρωτοκόλλου θα πάρουμε κάτι σαν αυτό που φαίνεται στο κατωτέρω σχήμα 6.2 όπου αριστερά φαίνεται το απλό Vegas και δεξιά το slow start του Tahoe.
Τι γίνεται στη συνέχεια; Η συνεχής αύξηση αναπόφευκτα θα οδηγήσει κάποια στιγμή σε κορεσμό κάποιας ζεύξης του Διαδικτύου κατά μήκος της διαδρομής της σύνδεσης και σε υπερχείλιση του προκείμενου ταμιευτήρα του δρομολογητή που την τροφοδοτεί (αφού την ίδια πολιτική αύξησης ακολουθούν όλες οι πηγές TCP του δικτύου). Η τιμή παραθύρου για την οποία εμφανίστηκε η απώλεια φυλάσσεται στην παράμετρο «κατωφλίου» (threshold) *Conthresh (CongestionThreshold)* και είναι βασική για τον επόμενο αλγόριθμο αποφυγής συμφόρησης.



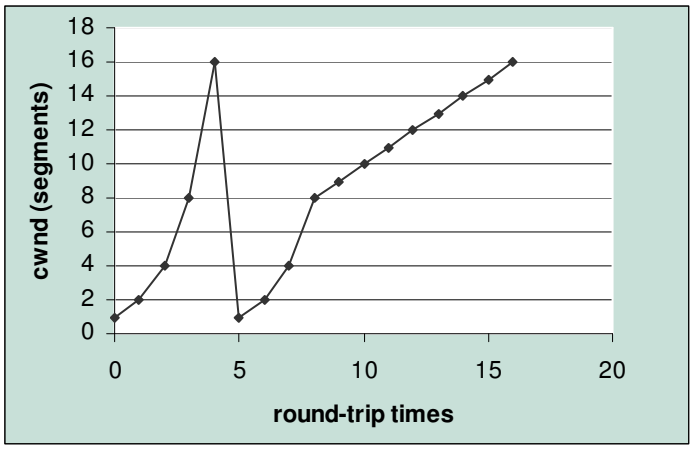
Σχήμα 6.2. Σύγκριση γραμμικής έναντι εκθετικής αύξησης ρυθμού

Όταν ανιχνευθεί η πρώτη απώλεια πακέτου, (με εκπνοή χρονιστή ή διπλό ACK) τότε το μισό της τιμής του *Cwnd* αλλά όχι λιγότερο από την τιμή δύο τμημάτων ή περισσότερο από την αναγγελία του δέκτη) φυλάσσεται στο *Conthresh.* Επιπροσθέτως, εάν είχαμε εκπνοή, τότε το *Cwnd* πέφτει στην τιμή 1 και αρχίζει πάλι αργή εκκίνηση αλλιώς συνεχίζει από την τιμή που είχε. Εάν νέα τμήματα επιβεβαιώνονται με άφιξη νέων ACK, τότε το *Cwnd* αυξάνει πάλι αλλά ο τρόπος αύξησης εξαρτάται από τη σύγκριση με το *Conthresh*, δηλαδή εάν το *Cwnd* είναι ίσο ή μικρότερο από το *Conthresh*, κάνουμε εκθετική αύξηση (αργή εκκίνηση), αλλιώς γραμμική αύξηση (αποφυγή συμφόρησης). Εάν πάμε σε αργή εκκίνηση, αυτή θα συνεχίσει μέχρι να φθάσει το παράθυρο την τιμή *Conthresh* που είναι η μισή της τιμής παραθύρου που προκάλεσε συμφόρηση. Από κει και πέρα συνεχίζει με γραμμική αποφυγή συμφόρησης. Δηλαδή, η μετάπτωση από την εκθετική αύξηση στον βασικό γραμμικό ρυθμό αύξησης συνιστά το δεύτερο μηχανισμό του Tahoe: την αποφυγή συμφόρησης (congestion avoidance).

Δηλαδή το Tahoe με το slow start επιχειρεί πολύ γρήγορα να βρει τον ρυθμό αποστολής που οδηγεί σε συμφόρηση και ακολούθως προσπαθεί να ισορροπήσει γύρω από τη σωστή τιμή. Με την εκθετική αύξηση δημιουργείται μια κατάσταση κατά την οποία χάνονται μαζί πολλά πακέτα (εάν φθάσουμε π.χ. σε μια αύξηση από 16 σε 32 τμήματα, στην χειρότερη περίπτωση οι ταμιευτήρες μπορεί μόλις να είχαν χωρέσει τα προηγούμενα πακέτα και έτσι να χαθούν και τα 32 νέα πακέτα ή εάν υπήρχαν κάποιες ακόμη θέσεις, να χαθούν τα μισά αλλά σε κάθε περίπτωση χάνονται πολλά μαζί με αποτέλεσμα να σταματά η ροή των ACK και να αδρανήσει η εκπομπή μέχρι να υπάρξει εκπνοή του χρονιστή.

Ας κάνουμε μια παρένθεση για ένα σχόλιο για την περίεργη ονομασία «αργή εκκίνηση». Όπως το παρουσιάσαμε η λέξη «αργή» φαίνεται παράταιρη για να εκφράσει μια τόσο πιο γρήγορη πολιτική σε σύγκριση με την γραμμική αύξηση. Για να γίνει αντιληπτή η ονομασία πρέπει να συγκριθεί με την αρχική συμπεριφορά του TCP. Τότε το TCP ξεκινούσε την αποστολή σε μια καινούργια σύνδεση με το παράθυρο που έστελνε ο δέκτης με την αναγγελία παραθύρου δημιουργώντας μια ριπή πακέτων που μπορεί να μην ήταν πρόβλημα στην πολύ αρχική λειτουργία των εντελών αφόρτιστων δικτύων, έγινε όμως αιτία μεγάλων προβλημάτων μόλις άρχισαν οι χρήστες να χρησιμοποιούν πιο πολύ το δίκτυο. Τότε μελετήθηκε το πρόβλημα και έγινε αντιληπτή η συμπεριφορά και ευστάθεια της γραμμικής αύξησης / πολλαπλασιαστικής μείωσης. Η αργή εκκίνηση πήρε αυτό το όνομα για να δηλώσει αυτή τη σταδιακή αύξηση έστω και εκθετική έναντι της αρχικής ανεξέλεγκτης συμπεριφοράς.

Πρέπει πλέον να έχει γίνει σαφής η συνολική στρατηγική του TCP όσον αφορά το χειρισμό της συμφόρησης. Σκεφθείτε ότι όταν ξεκινά το TCP δεν ξέρει εάν έχει μπροστά του μια ζεύξη 2,4Kbps 2,4Gbps, ούτε πόσες άλλες συνδέσεις τη μοιράζονται εκείνη τη στιγμή. Έτσι πρέπει να ψάξει για να βρει γρήγορα τι ρυθμό σηκώνει η ζεύξη. Αν γίνει αργά δεν έχουμε καλή απόδοση αφού χαραμίζουμε για πολύ χρόνο διαθέσιμη χωρητικότητα, εάν το ξεπερνάμε πάλι χάνουμε διότι οδηγούμεθα σε απώλειες πακέτων και πολλές αναμεταδόσεις.
Η φάση της αργής εκκίνησης φαίνεται παραστατικά στην αρχή του κατωτέρω σχήματος 6.3 όπου φαίνεται η εξέλιξη του αύξοντος αριθμού πακέτων (sequence number) με το χρόνο. Διακρίνεται η αρχική εκθετική φάση και ακολούθως τη χρονική στιγμή 4 ενώ το παράθυρο έχει φθάσει τιμή 16 επισυμβαίνει εκπνοή χρονιστή χωρίς επιβεβαίωση. Η τιμή πέφτει στο 1 και ακολουθεί πάλι αργή εκκίνηση (εκθετική). Αυτή τη δεύτερη φορά ωστόσο ξέρουμε κάτι περισσότερο για τις δυνατότητες του ενδιάμεσου δικτύου και έτσι μόλις φθάσουμε στο μισό του προηγούμενου κατωφλίου (8), μεταπίπτουμε σε γραμμική αύξηση του ρυθμού αποστολής.



Σχήμα 6.3 Αργή εκκίνηση και αποφυγή συμφόρησης

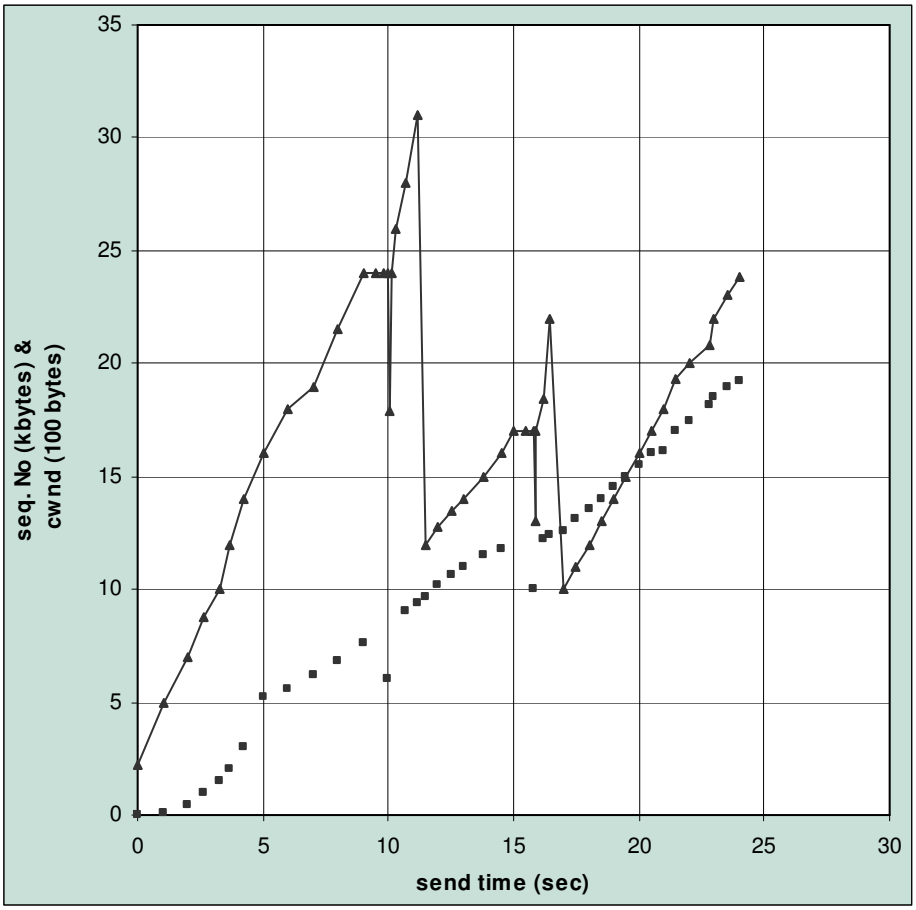
6.3.3 Η βαριάντα Reno

Οι σχετικοί αλγόριθμοι του TCP δέχθηκαν περαιτέρω βελτιώσεις που προτάθηκαν το 1990 από τον Jakobson. Συνίστανται σε δύο μηχανισμούς: την «ταχεία επανεκπομπή» (fast retransmit) και την «ταχεία ανάκαμψη» (fast recovery) που έτυχαν εκτεταμένης υλοποίησης και σήμερα κυριαρχούν. Με την προσθήκη της δεύτερης η υλοποίηση έλαβε το ανεπίσημο όνομα βαριάντα Reno. Το πρόβλημα που είχε προκύψει ήταν ότι λόγω της πολύ χονδρικής τιμής των εκτιμήσεων του χρόνου RTT και κατά συνέπεια των εκπνοών των χρονιστών, εμφανίζονταν μεγάλες περίοδοι αδράνειας του αποστολέα εν αναμονή εκπνοής χρονιστή. Η ταχεία επανεκπομπή προσπαθεί να αποφύγει την εκπνοή εάν απλά εμφανιστεί πολλαπλή επιβεβαίωση με την ελπίδα ότι δεν θα ανατραπεί ο ρυθμός εκπομπής από μια περιστασιακή απώλεια (εκτός του ότι πιθανόν να μην πρόκειται για απώλεια αλλά για καθυστέρηση του πακέτου ή αλλαγή της σειράς πακέτων σε ενδιάμεσους κόμβους, πράγματα όχι σπάνια). Φυσικά εάν παρ' ελπίδα πρόκειται για ισχυρή συμφόρηση, πολλές απώλειες θα ακολουθήσουν με αποτέλεσμα εκπνοή που θα ακολουθηθεί από μεγάλη μείωση του *cwnd* και συνεπώς και πλήρη ανατροπή του ρυθμού αποστολής όπως εξάλλου είναι απαραίτητο για να αντιμετωπισθεί η συμφόρηση. Τις πιο πολλές φορές όμως μια μικρότερη διόρθωση είναι αρκετή.

Έτσι λοιπόν η ταχεία επανεκπομπή προβλέπει ότι στην περίπτωση τριπλού ACK δηλ λήψη τριών με τον ίδιο αριθμό ACK, (υπενθυμίζεται ότι στον αριθμό αυτό ο δέκτης γράφει το αριθμό του επόμενου byte που αναμένει) τότε η πηγή ξαναστέλνει αμέσως το τμήμα που αρχίζει από τον αριθμό αυτό (που πιθανότατα έχει χαθεί) χωρίς να περιμένει την εκπνοή του χρονιστή. Θέτει το *Conthresh* στο μισό του *cwnd* (κίνηση προετοιμασίας) Κατόπιν η πηγή μπαίνει σε ταχεία ανάκαμψη δηλαδή ρίχνει το *cwnd στο Conthresh+3φορές*

το μέγεθος τμήματος) και εκτελεί αποφυγή συμφόρησης και όχι αργή εκκίνηση, δηλαδή συνεχίζει να αυξάνει το *cwnd* από την τιμή που βρίσκεται κατά 1/*cwnd* σε κάθε ACK. Στέλνει νέα τμήματα εάν επιτρέπεται από το παράθυρο που έχει εκείνη τη στιγμή, αλλιώς δεν στέλνει. Η ταχεία ανάκαμψη τερματίζεται με τη λήψη της επιβεβαίωσης του ελλείποντος τμήματος οπότε το *cwnd* παίρνει την τιμή που έχει το *Conthresh*. Έτσι εκμεταλλεύεται τα ACKs που βρίσκονται ακόμα καθ' οδόν στο «σωλήνα» για να καθορίσουν το τέμπο εκπομπής πακέτων.

Η φιλοσοφία της ταχείας επανεκπομπής και ταχείας ανάκαμψης είναι ότι το τριπλό ACK δίνει περισσότερες πληροφορίες για την κατάσταση του ενδιάμεσου δικτύου απ' ότι η εκπνοή του χρονιστή. Συγκεκριμένα δείχνει ότι κάποιο επόμενο τμήμα έχει φθάσει απέναντι (διότι μόνο τότε θα σταλούν πολλαπλά ACK) και άρα η ροή δεν έχει κοπεί πλήρως. Μπορεί λοιπόν η απώλεια να ήταν τυχαίο γεγονός και όχι συμφόρηση άρα δεν πρέπει να σπεύσουμε σε μεγάλη πτώση του ρυθμού χαλώντας την απόδοση. Αν βέβαια αυτή η υπόθεση δεν είναι σωστή, όπως ελέχθη, θα εμφανισθεί και άλλη απώλεια προκαλώντας εκπνοή του χρονιστή τελικά αναγκάζοντας την πηγή να πέσει σε αργή εκκίνηση.
Χάρις στην ταχεία επανεκπομπή αποφεύγονται περίπου οι μισές εκπνοές χρονιστών επιτυγχάνοντας αύξηση περίπου 20% στην διοχέτευση (throughput).



Σχήμα 6.4 Συνολικό Παράδειγμα αποφυγής συμφόρησης (▲=cnwd, ■=Seq.No)

Εδώ αξίζει να διευκρινισθεί ότι το διπλό ACK δεν μπορεί να χρησιμοποιηθεί κατά τον ίδιο τρόπο όπως το τριπλό, διότι πλειστάκις δεν οφείλεται σε απώλεια τμήματος αλλά απλή αναδιάταξη πακέτων που ακολούθησαν διαφορετικούς δρόμους και έτυχε το επόμενο να φθάσει πριν το προηγούμενο. Το TCP επιτάσσει άμα τη λήψει τμήματος εκτός σειράς, ο δέκτης να στείλει πάραυτα ένα ACK με αριθμό αυτόν του ελλείποντος τμήματος. (Αυτό το ACK δεν μπορεί να καθυστερήσει πράγμα που γίνεται συνήθως κατ'

οικονομίαν δεδομένου του αθροιστικού χαρακτήρα των ACK, ώστε με ένα επόμενο ACK να επιβεβαιωθούν και τα δύο). Υπενθυμίζεται ότι το TCP δεν διαθέτει NACK δηλαδή αρνητική επιβεβαίωση.
Στο σχήμα 6.4 δίδεται ένα τυπικό παράδειγμα της πορείας της παραμέτρου *cwnd (συνεχής γραμμή με τριγωνικά σημάδια τις στιγμές λήψης ACK)* σε συνάρτηση με την εξέλιξη της πορείας της παραμέτρου του αύξοντος αριθμού πακέτων -*sequence number (τετραγωνικά σημάδια τις στιγμές αποστολής πακέτου).* Και οι δύο παράμετροι δίδονται σε bytes για να μην ξεχνάμε ότι έτσι δουλεύει το TCP και να γίνει πιο ρεαλιστικό το παράδειγμα. Ωστόσο για να χωρούν και οι δύο καμπύλες στο γράφημα το *cwnd* είναι εκπεφρασμένο σε εκατοντάδες bytes και το δεύτερο σε χιλιάδες.
Το χρονικό των αποστολών και λήψεων κατά τη διάρκεια του ανωτέρω επεισοδίου fast retransmit / fast recovery (δηλ. ανάμεσα στα δευτ. 9 έως 12 περίπου) φαίνονται στον πιο κάτω πίνακα.
Παρατηρούμε την αρχική φάση αργής εκκίνησης μέχρι το δευτ. 5. Το δευτ. 10 έχουμε την πρώτη επαναποστολή λόγω λήψης τριπλού ACK (fast retransmit) Παρατηρείστε ότι το *cwnd*. παραμένει ως έχει (2400) μέχρι το 3^{ο} αφήνοντας οριζόντια την καμπύλη. Όταν έλθει το τρίτο ACK, το *conthresh* παίρνει τιμή ίση με το μισό *cwnd στρογγυλεμένο προς τα κάτω στο πολλαπλάσιο του μεγέθους τμήματος που είναι 256 ήτοι 1024.* Ταυτόχρονα το *cwnd* παίρνει τιμή ίση με το *conthresh* συν 3 φορές το μέγεθος τμήματος δηλ. 1024+3*256=1792, και αμέσως γίνεται η επαναποστολή του υπό υποψία απώλειας τμήματος. (Το τμήμα αυτό αντιστοιχεί στην τιμή του Seq No στο σχήμα που βρίσκεται κάτω από τις δύο προηγούμενες τιμές ενώ μέχρι τότε η καμπύλη ήταν μονοτόνως αύξουσα).
Ωστόσο με τα επόμενα πολλαπλά ACK (στο συγκεκριμένο παράδειγμα εστάλησαν τελικά 8 πολλαπλά ACK με ίδιο αριθμό επιβεβαίωσης) το *cwnd* συνεχίζει να αυξάνει αφού η συνεχής άφιξη των ACK έστω και διπλών δείχνει ότι δεν έχουμε πλήρη διακοπή επικοινωνίας. Αυτή είναι η συμπεριφορά του fast recovery και τερματίζεται μόλις έλθει ACK για νέο τμήμα (και όχι πλέον πολλαπλό). Ακολουθούν τα άλλα 5 πολλαπλά ACK και κάθε φορά το *cwnd* αυξάνει κατά το μέγεθος του τμήματος μέχρι που έρχεται το νέο που τερματίζει τη φάση fast recovery οπότε το *cwnd* παίρνει την τιμή του *conthresh* δηλ. 1024. Ακολουθεί πλέον κανονική αποφυγή συμφόρησης με γραμμική αύξηση μέχρι το νέο επεισόδιο πολλαπλών ACK γύρω στο 16^{ο} δευτ. όπου επαναλαμβάνεται περίπου η ίδια ιστορία. Το χρονικό των αποστολών και λήψεων κατά τη διάρκεια του ανωτέρω επεισοδίου fast retransmit / fast recovery (δηλ. ανάμεσα στα δευτ. 9 έως 12 περίπου) φαίνονται στον κατωτέρω πίνακα:

A/A (seq No)	Αποστολή (bytes)	Λήψη	*cwnd*	*conthresh*
8700	Νέο τμήμα (256)		2400	512
		1^{ο} πολ/λό ACK	2400	512
		2^{ο} πολ/λό ACK	2400	512
		3^{ο} πολ/λό ACK	1792	1024
6652	Επαναπ/λή (256)			
		4^{ο} πολ/λό ACK	2048	1024
		5^{ο} πολ/λό ACK	2304	1024
		6^{ο} πολ/λό ACK	2560	1024
8956	Νέο τμήμα (256)			
		7^{ο} πολ/λό ACK	2816	1024
9212	Νέο τμήμα (256)			
		8^{ο} πολ/λό ACK	3072	1024
9468	Νέο τμήμα (256)			
		Νέο ACK για 8956		

Παρατηρούμε ότι αποστέλλεται νέο τμήμα μετά το 6^{ο} και 7^{ο} πολλαπλό ACK αλλά όχι μετά το 4^{ο} και 5^{ο}. Ο λόγος είναι φυσικά η τιμή του παραθύρου που στις δύο τελευταίες αυτές περιπτώσεις έχει τιμή κατώτερη από το πλήθος μη επιβεβαιωθέντων bytes που είναι 2304. Ωστόσο, μόλις φθάσει το παράθυρο σε αυτή την τιμή

(υπό τον όρο ότι δεν έχουμε από το δέκτη τυχόν μικρότερη αναγγελία παραθύρου) μπορούν να αποσταλούν νέα τμήματα. Προσέξτε πως η τιμή του sequence No αυξάνει κατά το μέγεθος του τμήματος π.χ. 8700+256=8956+256=9212+256=9468.
Πρέπει να παρατηρήσουμε ότι στο παράδειγμα αυτό σε σχέση με το σχήμα 6.3 που δόθηκε στη βαριάντα Tahoe, δεν έχουμε εκπνοή του χρονιστή, (μόνο πολλαπλά ACK) έτσι πήγαμε σε ταχεία ανάκαμψη (fast recovery) και όχι σε αργή εκκίνηση, όπως θα πηγαίναμε εάν τελικά δεν αποφεύγαμε την εκπνοή. Αλλά ας μην ξεχνάμε ότι αυτό ακριβώς προσπαθούμε να αποφύγουμε με την ταχεία επανεκπομπή (fast retransmit).
Προτού εγκαταλείψουμε τη μελέτη του παραδείγματος, αξίζει να σχολιάσουμε τη συνολική εικόνα της εξέλιξης των δύο παραμέτρων στο γράφημα, δηλ. να δούμε πια το δάσος έχοντας μελετήσει πιο πάνω ενδελεχώς τα δέντρα.
Βλέπουμε λοιπόν ότι ενώ το παράθυρο συμφόρησης έχει σκαμπανεβάσματα αναζητώντας επιθετικά τη σωστή χωρητικότητα των διαθέσιμων «σωληνώσεων», ο αύξων αριθμός πακέτων (seq. No) παρά κάποιες μικροαποκλίσεις ακολουθεί μια περίπου γραμμική πορεία με κάπως σταθερή κλίση δηλαδή με περίπου σταθερό μακροπρόθεσμα ρυθμό εκπομπής. Δηλαδή μέσα από τους πολύπλοκους υπολογισμούς και ενδείξεις, το TCP βρίσκει ποιος είναι ο μακροπρόθεσμα εφικτός ρυθμός εκπομπής και επιτυγχάνει μια αξιοσημείωτα ομαλή αποστολή πληροφορίας αν σκεφθούμε τη δυναμικά μεταβαλλόμενη ανταγωνιστική κίνηση που έχουν να αντιμετωπίσουν τα πακέτα διασχίζοντας τα σημεία συμφόρησης (bottlenecks= στενωπούς) στη «ζούγκλα» του Διαδικτύου. Τα σημεία αυτά δημιουργούνται από τυχαίους στατιστικούς λόγους σε διάφορα σημεία του δικτύου. Εάν κάποιο σημείο αντιμετωπίζει συχνά συμφόρηση σημαίνει ότι ο χειριστής του δικτύου πρέπει να αυξήσει τη χωρητικότητα στο σημείο αυτό. Δουλειά του πρωτοκόλλου είναι να χειρίζεται σωστά τις δυναμικές, στατιστικές, στιγμιαίες συμφορήσεις προσφέροντας υψηλή απόδοση με ταυτόχρονη δίκαιη μοιρασιά του εύρους ζώνης.
Σήμερα πλέον οι πιο πολλοί υπολογιστές διαθέτουν την βαριάντα Reno. Τι συμβαίνει όμως όταν συναντώνται σε μία στενωπό συνδέσεις ενός παλαιού υπολογιστή που δεν έχει ανανεωθεί από την βαριάντα Vegas;. Η απάντηση είναι ότι οι αλγόριθμοι του Vegas προσπαθούν να κρατήσουν τα πακέτα στο σημείο συμφόρησης κοντά στο 2 ενώ οι αλγόριθμοι του Reno αυξάνουν τον ρυθμό μέχρι την υπερχείλιση και μετά υποχωρούν αλλά σε κάθε περίπτωση τα πακέτα που αφήνουν στο σημείο συμφόρησης είναι αρκετά πάνω από το 2. Δηλαδή η βαριάντα Vegas αδικείται από τον ανταγωνισμό με την υλοποίηση Reno.
Η ολοκλήρωση της μελέτης αυτού του θέματος, σίγουρα θα έχει επηρεάσει τον τρόπο που αντιμετωπίζουμε την πρόσβαση στο Διαδίκτυο, διότι κάθε φορά που με ένα κλικ θα ζητάμε κάποιες πληροφορίες ή αρχεία από κάποιο εξυπηρετητή δεν θα μπορούμε να μην αναλογισθούμε πόσο πολύπλοκοι μηχανισμοί αναλαμβάνουν δράση μέσα στο TCP για να μας εξυπηρετήσουν με ταχύτητα και αποδοτικότητα χωρίς να αδικήσουν ούτε εμάς ούτε τους άλλους χρήστες..

6.4 Αμεση Ειδοποίηση Συμφόρησης (ECN)-

Προτού κλείσουμε το κεφάλαιο, αξίζει να αναφερθεί ότι πολλοί ερευνητές νοιώθουν ότι η λύση της έμμεσης γνώστοποίηση συμφόρησης μέσω απωλειών πακέτων ήταν μια λύση ανάγκης όσο το Διαδίκτυο πατούσε πάνω στα διάφορα φυσικά δικτυα της εποχής. Τώρα που έχει κυριαρχήσει και το πρωτόκολλο IP είναι το συνηθέστερο πρωτόκολλο στρώματος δικτύου, αλλά και δεδομένου ότι απαιτητικές εφαρμογές ζωντανών υπηρεσιών δεν λειτουργούν καλά με απώλειες, έχουν προτείνει την εισαγωγή άμεσης ειδοποίησης συμφόρησης (ECN-Explicit Congestion Notification), που απαιτεί συνεργασία IP και TCP. Προς τούτο προτείνουν την χρήση των δύο περισσευόντων bits στα δεξιά του πεδίου DS (πρώην TOS) της κεφαλίδας του IP για να ειδοποιούνται οι πομποί για επικείμενη υπερχείλιση ταμιευτήρων ενός δρομολογητή πριν καν επισυμβεί αυτή. Η κωδικοιποίηση που προτείνεται στο RFC 3168 του 2001 είναι η εξής:

- 00: Non ECN-Capable Transport - Non-ECT
- 10: ECN Capable Transport - ECT(0)
- 01: ECN Capable Transport - ECT(1)
- 11: Congestion Encountered - CE

Δεν θα επεκταθούμε σε περαιτέρω λεπτομέρειες καθόσον το θέμα είναι ακόμη σε πρώϊμα στάδια και δεν έχει τύχει ευρείας εφαρμογής.

6.4 Αλγόριθμοι βελτίωσης απόδοσης

Πολλές βελτιώσεις για καλύτερη απόδοση έχουν εισαχθεί με τα χρόνια στο TCP. Δεν υπάρχει χώρος για να τους αναλύσουμε εδώ παρά να αναφερθούν ακροθιγώς. Για λεπτομέρειες μπορεί να βρεί ο αναγνώστης στο βιβλίο του Stevens "TCP/IP illustrated".

- Καθυστέρηση επιβεβαίωσης. Σε μερικές περιπτώσεις βελτιώνεται η αποδοτικότητα με το να καθυστερεί το ACK κατά 0.1 δευτ.
- Αποφυγή συνδρόμου ανοήτου παραθύρου (silly window syndrome). Σε μερικές περιπτώσεις η τιμή του παραθύρου είναι πολύ μικρή και το δίκτυο φορτώνεται με πολλά πακέτα ACK. Για αποφυγή του φαινομένου, ο αποστολέας στέλνει πακέτα μόνο εάν το παράθυρο είναι επαρκώς μεγάλο.
- Αλγόριθμος του Nagle. Προσπαθεί να αποφύγει την αποστολή μικρών τμημάτων TCP.

Κεφάλαιο 7

ΑΛΛΑ ΠΡΩΤΟΚΟΛΛΑ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ

7.1 IP version 6

Η έκδοση 4 του IP είναι εκείνη που γνώρισε τη μεγαλύτερη εξάπλωση αλλά αυτή ακριβώς ήταν η αιτία που γέννησε την ανάγκη για αντικατάστασή της. Ο λόγος ήταν η εξάντληση του χώρου διευθύνσεων λόγω της ραγδαίας διάδοσης IP, όπως εξηγήθηκε στο υποκεφάλαιο περί αταξικής δρομολόγησης η οποία και αποτελεί προσωρινή λύση αυτού του προβλήματος. Η μόνιμη λύση μαζί με πολλές άλλες βελτιώσεις δίδεται με τη νέα έκδοση την 6.

Οι βελτιώσεις που εισάγει το IPv6 είναι συνοπτικά τα εξής:

- Τεράστιο πλήθος διευθύνσεων με το πεδίο διευθύνσεων να έχει μήκος 128 bits (16 bytes)
- Ελάττωση των πεδίων της επικεφαλίδας του IPv6 που αποκτά σταθερό μήκος 40 bytes (εξ ών 32 μόνο για τις διευθύνσεις προέλευσης και προορισμού) προς διευκόλυνσης κατασκευής ολοκληρωμένων κυκλωμάτων γρήγορης δρομολόγησης. Τα πρόσθετα πεδία που απαιτούνται για καινούργιες λειτουργίες μπαίνουν σε νέες προαιρετικές επικεφαλίδες (extension headers) που ακολουθούν την βασική επικεφαλίδα.
- Ο τεμαχισμός πακέτων στους ενδιάμεσους κόμβους καταργείται (μόνο ο αρχικός κόμβος μπορεί να το κάνει). Η πηγή πρέπει να στείλει ένα πακέτο ανακάλυψης της μικρότερης MTU κατά μήκος της διαδρομής εάν θέλει να στείλει μεγάλα πακέτα.
- Εισάγεται ο προσδιορισμός ροών κίνησης (traffic flows) για να μπορούν να προσδιορίζονται ειδικές τεχνικές απόρριψης ανάλογα με τη ροή ώστε να μπορούν να υποστηριχθούν διαφορετικά υπηρεσίες πραγματικού χρόνου (π.χ. φωνή, βίντεο).
- Βελτιώνεται η υποστήριξη της πολλαπλής εκπομπής (Multicasting), όταν προορισμός είναι όλοι οι κόμβοι ενός σετ, και εισάγεται η έννοια του anycasting (δηλ. όταν προορισμός είναι οποιοσδήποτε από ένα σετ κόμβων).
- Εισάγονται εργαλεία αυθεντικοποίησης και κρυπτογράφησης (authentication and encryption) για υποστήριξη ασφάλειας στο επίπεδο δικτύου. (Σήμερα αυτό γίνεται μόνο στο επίπεδο εφαρμογής). Έτσι θα μπορούν να δημιουργηθούν νοητά ασφαλή υποδίκτυα μέσα στο Διαδίκτυο.

Επίσης έχει γίνει μεγάλη προσπάθεια απλοποίησης της επεξεργασίας που πρέπει να κάνουν οι δρομολογητές ώστε να μπορούν πολλές λειτουργίες να γίνονται με υλισμικό (ολοκληρωμένα κυκλώματα) και όχι με λογισμικό.

Οι διευθύνσεις αποδίδονται στις διεπαφές και όχι στους κόμβους. Οι μεγάλες διευθύνσεις του IPv6 σε συνδυασμό με τις διαφορετικές διευθύνσεις σε κάθε διεπαφή του δρομολογητή επιτρέπουν μικρότερους πίνακες στους δρομολογητές. Αυτή η φαινομενική αντίφαση εξηγείται διότι επιτρέπει την ομαδοποίηση διευθύνσεων σε ιεραρχημένες δομές κατά γεωγραφική ή διοικητική (δηλ. ανά πάροχο υπηρεσίας). Έτσι ομάδες διευθύνσεων αντιπροσωπεύονται από μία καταχώρηση στον πίνακα δρομολόγησης. Ομοίως η ύπαρξη διαφορετικής διεύθυνσης σε κάθε διεπαφή επιτρέπει να ομαδοποιηθεί ο κόμβος κάτω από την ομαδική διεύθυνση του κάθε παρόχου υπηρεσιών αλλιώς θα έπρεπε να καταχωρισθεί χωριστά η διεύθυνση του κάθε παρόχου.
Στο κατωτέρω σχήμα φαίνεται ένα πακέτο που έχει όλες τις προαιρετικές επικεφαλίδες με τη σωστή σειρά που απαιτεί το πρότυπο. Κάθε κεφαλίδα έχει το πεδίο next header όπου δηλώνει τι κεφαλίδα ακολουθεί ή ότι δεν ακολουθεί άλλη (η λεπτομέρεια αυτή ωστόσο δεν φαίνεται στο σχήμα).
Ο ρόλος των πιο πολλών πεδίων είναι αυτονόητος, μερικά που δεν είναι προφανή είναι τα εξής:
Το πεδίο hop limit είναι το γνωστό Time to Live το οποίο στην πράξη είχε εκφυλιστεί από χρονικό όριο σε Hop count πράγμα που επισημοποιείται στο IPv6.

0	4	8	16	24	31
Version	Priority	FLOW LABEL			
Payload Length			Next Header	Hop Limit	
Source address 4x4 bytes=16 bytes					
Source address (continued)					
Source address (continued)					
Source address (continued)					
Destination address 4x4 bytes=16 bytes					
Destination address (continued)					
Destination address (continued)					
Destination address (continued)					
Hop-by-hop options header (variable length)					
Routing header (variable length)					
Fragment header 2x4 bytes					
Fragment header (continued)					
Authentication header (variable)					
Encapsulation security payload header (variable)					
Destination options header					
TCP header (20bytes (optional variable part)					
Application Data (variable up to 64kbytes)					
Application Data (continued)					
. . .					
Application Data					

΄Σχήμα 7.1 Φόρμα του πακέτου IPv6 με όλες τις επικεφαλίδες επέκτασης

Το πεδίο flow label είναι ένας αριθμός που μπαίνει από την πηγή για να αναγνωρίζονται τα πακέτα που ανήκουν στη ίδια ροή. Αυτό θα επιτρέψει στο μέλλον να υποστηρίζονται με ειδικό τρόπο όλα τα πακέτα μιας ροής. Για να γίνει αυτό θα πρέπει να έχει προηγηθεί η αποστολή ή η συμφωνία για αυτή την ειδική υποστήριξη (χρήση περισσότερου χώρου ταμιευτήρα ή παροχή ελάχιστου εύρους ζώνης), μέσω διαπραγματεύσεων της πηγής διότι δεν περιέχεται σχετική πληροφορία στην κεφαλίδα. Αυτό σημαίνει ότι οι δρομολογητές πρέπει να τηρούν σχετικές πληροφορίες σε βάση δεδομένων (state information) κάτι που είχε αποφευχθεί μέχρι τώρα στο Διαδίκτυο. Οι πηγές που δεν υποστηρίζουν το πεδίο αυτό θέτουν 0. Οι δρομολογητές που δεν το υποστηρίζουν το διαβιβάζουν αναλλοίωτο.

Μετά το πρώτο πεδίο βερσιόν, περιέχεται ένα νέο πεδίο το πεδίο *προτεραιότητος*. Το πεδίο αυτό περιέχει ένα bit που χαρακτηρίζει τη ροή σαν υποκείμενη σε έλεγχο συμφόρησης ή όχι. Τα άλλα τρία bit κατατάσσουν το πακέτο σε μία από 8 στάθμες προτεραιότητος. Οι προτεραιότητες της υποκείμενης σε έλεγχο συμφόρησης κίνησης (όπως π.χ. είναι η κίνηση που βασίζεται στο TCP) είναι οι εξής, κινούμενοι από την υψηλή στην χαμηλή προτεραιότητα:

- Κίνηση ελέγχου του Διαδικτύου, π.χ. μηνύματα OSPF ή BGP τα οποία πρέπει να περάσουν οπωσδήποτε διότι ενημερώνουν τους δρομολογητές με την κατάσταση συμφόρησης και είναι απαραίτητα για την

αντιμετώπισή της, (κατ' αναλογίαν, είναι σαν να αφήνουμε να περνούν τα περιπολικά της τροχαίας σε περίπτωση μποτιλιαρίσματος). Το ίδιο προβλέπεται για τα μηνύματα διαχείρισης (SNMP).

- Διαδραστική κίνηση. Ζωντανές υπηρεσίες όπου η γρήγορη απόκριση είναι βασικό στοιχείο ποιότητος υπηρεσίας όπως VoIP και διαδραστικά παιχνίδια
- Μαζικές μεταφορές με παρακολούθηση του χρήστη (π.χ. FTP ή HTTP όπου ο χρήστης περιμένει γρήγορο αποτέλεσμα έστω και όχι ακαριαίο)
- Μαζικές μεταφορές χωρίς επιμέλεια του χρήστη (π.χ. ηλεκτρονικό ταχυδρομείο)
- Κίνηση «Παραγέμισματος» (Filler traffic) δηλ. μηνύματα που στέλνονται όταν υπάρχει ευκαιρία (π.χ USENET messages). Αυτή η κίνηση έχει και τη χαμηλότερη προτεραιότητα και διαβιβάζεται όταν δεν αναμένεται άλλη κίνηση.

Η επικεφαλίδα Hop-by-hop option περιέχει πληροφορία που πρέπει να εξετασθεί από κάθε δρομολογητή.
Η επικεφαλίδα routing header είναι το ισοδύναμο του source routing στο IPv4 και περιέχει τις διευθύνσεις όλων των ενδιάμεσων δρομολογητών που πρέπει να διασχισθούν.

7.2 CIDR Classless Interdomain Routing (Αταξική Δρομολόγηση μεταξύ Επικρατειών)

Ενώ στο ξεκίνημα του Διαδικτύου ορίσθηκαν οι 4 κλάσεις διευθύνσεων που παρουσιάστηκαν στο αντίστοιχο κεφάλαιο, η εκρηκτική αύξηση των χρηστών του Διαδικτύου προκάλεσε εξάντληση των διευθύνσεων. Βέβαια τα 32 bits δίνουν περίπου 4 δις διευθύνσεων και ακόμα δεν έχουμε φτάσει ούτε στο 1 δις χρηστών, ωστόσο οι διευθύνσεις δεν μπορεί να μοιράζονται με συνεχή αρίθμηση σε κάθε υποδίκτυο αλλά πρέπει να δίδονται και πολλές πλεονάζουσες για μελλοντική επέκταση. Για να γίνει αντιληπτό ας συγκρίνουμε με τις τηλεφωνικές διευθύνσεις με τις οποίες είμαστε πιο εξοικειωμένοι. Αν υποθέσουμε ότι δεν επαρκούν οι τηλεφωνικοί αριθμοί και άρχισαν να εξαντλούνται π.χ. στην Δανία λόγω μεγάλης διάδοσης. Δεν μπορούμε να διαθέσουμε διευθύνσεις του υποδικτύου της Ζιμπάμπουε διότι το πρόθεμα το οποίο ανήκει στη χώρα αυτή έχει δεσμεύσει όλους τους αριθμούς ανεξάρτητα εάν εκεί θα χρησιμοποιηθούν ποτέ. Προσωρινά εάν βρίσκαμε τρόπο να μπορούμε να το κάνουμε αυτό θα ανακουφιζότανε το πρόβλημα. Η μόνιμη λύση είναι να αυξηθεί το μήκος των αριθμών και αυτό έγινε με το IPv6 όπου το μήκος των διευθύνσεων έγινε 128 bits!.
Ωστόσο μέχρι να διαδοθεί το IPv6 θα περάσουν 10-15 χρόνια. Μέχρι τότε υιοθετήθηκε η προσωρινή λύση με την παραβίαση των ορίων των κλάσεων. Το πρόβλημα κυρίως εκδηλώθηκε πρώτα με την εξάντληση των διευθύνσεων κλάσεως Β. Τα σχετικά δίκτυα μπορούν πλέον να πάρουν πολλές διευθύνσεις κλάσεως C και να τις χρησιμοποιούν σαν μία μικτή κλάση. Αυτή η κατάργηση των κλάσεων δημιουργεί ωστόσο ένα νέο σοβαρό πρόβλημα: κάθε υποδίκτυο πρέπει να έχει μια δική του καταχώρηση σε κάθε δρομολογητή του κόσμου. Αυτό μεγεθύνει επικίνδυνα τους πίνακες δρομολόγησης και επιβραδύνει δραματικά τη λειτουργία της δρομολόγησης με μακρές αναζητήσεις. Η απάντηση σε αυτό το πρόβλημα είναι το CIDR που επιτρέπει τη συνόψιση πολλών διευθύνσεων σε μια καταχώρηση του πίνακα στη βάση των κοινών bit ανώτερης τάξης.
Έτσι εάν υποδίκτυο πάρει π.χ. 16 διευθύνσεις κλάσεως C σε τρόπο ώστε τα ανώτερης τάξεως bits να είναι τα ίδια, τότε οι διευθύνσεις τους μπορούν να συνοψισθούν σε μια καταχώρηση αρκεί να συνοδεύονται με μια μάσκα που να δείχνει πόσα είναι τα κοινά bits (τα 20 από αριστερά στο παράδειγμα). Αυτό γίνεται με τη μάσκα να έχει 1 στη θέση όλων των κοινών bits ανώτερης τάξεως και 0 στη θέση των μη κοινών κατώτερης τάξεως που προσδιορίζουν τον ξενιστή, π.χ. 1111 1111 1111 1111 1111 0000 0000 0000. Δηλαδή η ουσιαστική πληροφορία της μάσκας είναι σε ποιο σημείο σταματούν οι άσσοι και αρχίζουν τα μηδενικά. (Αυτή η πληροφορία ήταν σταθερή στην κάθε κλάση και φαινόταν στα πρώτα bits της κάθε διεύθυνσης, αλλά τώρα αυτές οι 3 τυποποιημένες κλάσεις δεν αρκούν).
Για να υλοποιηθεί η αταξική δρομολόγηση χρειάζονταν δύο ακόμη στοιχεία: Πρώτον, να εμπλουτισθούν οι δρομολογητές ώστε να αναζητούν διευθύνσεις στους πίνακες με βάση όχι μόνο τη διεύθυνση των 32 ψηφίων αλλά και τη μάσκα. Η αναζήτηση μάλιστα βελτιώθηκε ώστε να μπορεί να γίνεται με βάση το μακρύτερο ταίριασμα ψηφίων από αριστερά της διεύθυνσης (longest address match), ώστε όσο πιο πολλά ψηφία ταιριάζουν με την καταχώριση του πίνακα, τόσο το καλύτερο. Δεύτερον, να εμπλουτισθούν τα πρωτόκολλα ανταλλαγής πληροφοριών δρομολόγησης μεταξύ των δρομολογητών ώστε να μεταφέρουν και τις μάσκες εκτός από τις διευθύνσεις. Αυτό έγινε με το OSFP, το RIP-2 και το BGP version 4. Έχοντας εφοδιασθεί με

όλα αυτά τα στοιχεία μπορεί το IP v4 να συνεχίσει για μερικά ακόμη χρόνια να χρησιμοποιεί τις διευθύνσεις 32 bits.
Ας δούμε ένα παράδειγμα. Έστω ένας πάροχος (ISP) που πήρε 16 διαδοχικές διευθύνσεις class C με το μπλοκ 194.0.16.0 έως 194.0.31.255. Η καταχώριση που αντιστοιχεί στους πίνακες θα είναι μόνο η: 194.0.16.0 με μάσκα 255.255.240.0. Εάν αναζητείται η διεύθυνση του ξενιστή 194.0.22.44, θα αναζητηθεί το ταίριασμα των αριστερών ψηφίων και θα ευρεθεί η καταχώριση: 194.0.16.0 που οδηγεί στη σωστή πόρτα.
Για να τονισθεί καλύτερα η αξία της συνόψισης θα αναφερθεί ότι υπάρχει η σύσταση οι νέες δευθύνσεις κλάσεως C στην Ευρώπη να είναι στην περιοχή 194.0.0.0 έως 194.255.255.255. Στους δρομολογητές στις άλλες περιοχές του κόσμου θεωρητικά θα αρκεί μία καταχώριση με τιμή 194.0.0.0 και μάσκα 254.0.0.0. Φυσικά εντός της Ευρώπης θα απαιτείται πιο λεπτομερής εύρεση της διαδρομής κάθε υποδικτύου και θα χρησιμοποιούνται πιο πολλές καταχωρίσεις και ταίριασμα περισσότερων ψηφίων, αλλά αυτό είναι αναπόφευκτο αφού οι δρόμοι των πακέτων χωρίζουν και οι πόρτες που πρέπει να αντιστοιχισθούν είναι περισσότερες.
Η ίδια λογική που περιγράψαμε πιο πάνω μπορεί να χρησιμοποιηθεί και εσωτερικά σε ένα υποδίκτυο κάποιου οργανισμού όταν δίνει διευθύνσεις στα εσωτερικά τμήματα κατά μπλοκ που ορίζονται από τα πιο σημαντικά ψηφία της εσωτερικής διευθυνσιοδότησης. Φυσικά τά πιο αριστερά bits μένουν πάντα σταθερά διότι ανήκουν στην ταυτότητα του υποδικτύου, ωστόσο τα επόμενα bits διατίθενται με την ίδια λογική στα υποτμήματα αφήνοντας τα τελευταία bits ώς συνήθως για τους ξενιστές. Η διαδικασία αυτή ονομάζεται subnetting (υποδικτύωση)

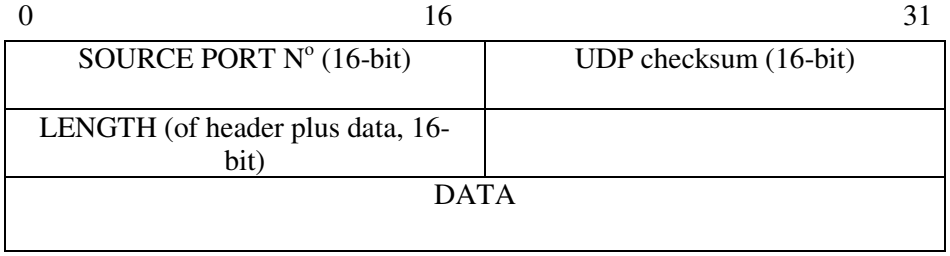
7.3 NAT- Network Address Translator (Μεταφραστής Διευθύνσεων Δικτύου)

Άλλη μια πρόχειρη λύση που επινοήθηκε για να αντιμετωπισθεί η έλλειψη διευθύνσεων διαδικτύου ήταν και η μετάφραση διευθύνσεων. Θα εξηγηθεί με ένα παράδειγμα από τη χρήση του σε σπίτι ή μικρή επιχείρηση, παρότι μπορεί να χρησιμοποιηθεί και από μεγαλύτερους οργανισμούς (π.χ. χρησιμοποιείται στο ΤΕΙ Πειραιά αλλά κυρίως για λόγους ασφάλειας. Εστω λοιπόν ένα σπιτικό modem/router εφοδιασμένο με DHCP (Dynamic Host Configuration Protocol) και NAT. Στο modem ανατίθεται από τον πάροχο μια και μόνη διεύθυνση π.χ. η 156.132.210.4 την οποία και θα χρησιμοποιήσουν όλοι οι υπολογιστές του εσωτερικού δικτύου LAN ή WLAN. Το DHCP έχει αναθέσει τις εσωτερικές διευθύνσεις 192.168.0.0 έως 192.168.0.6 στους 5 υπολογιστές του σπιτιού κρατώντας την 192.168.0.0 για τον εαυτό του. Για το πρώτο πακέτο που ξεκινά από τον κάθε εσωτερικό υπολογιστή, ο NAT δημιουργεί μια καταχώριση σε ένα δικό του πίνακα όπου αντιστοιχίζει την εσωτερική IP και τον αριθμό πόρτας, στην κοινή εξωτερική διεύθυνση 156.132.210.4 αλλά πάντα σε διαφορετική πόρτα. Συνήθως αφήνει τις πρώτες 4096 πόρτες που έχουν καταχωρισθεί σε υπηρεσίες (π.χ. η πόρτα 80 στο HTTP), και χρησιμοποιεί τις υπόλοιπες μέχρι το 65326 που είναι άφθονες. Όταν γυρίσει η απάντηση, το NAT αλλάζει την διεύθυνση στην εσωτερική διεύθυνση και πόρτα που βρίσκει στον πίνακα του. Αυτό φυσικά βασίζεται στον αριθμό πόρτας του εισερχόμενου πακέτου δεδομένου ότι η διεύθυνση IP παρότι ήταν απαραίτητη προκειμένου να δρομολογηθεί το πακέτο ως το σπίτι, δεν μας λέει και πολλά για μέσα στο σπίτι, αλλά ο αριθμός πόρτας είναι μοναδικός. Με την εσωτερική IP πλέον και την πόρτα εφαρμογής, το πακέτο θα δρομολογηθεί άνετα στον σωστό υπολογιστή και στην σωστή εφαρμογή. Σημειώστε ότι πακέτα browsing από τον κάθε εσωτερικό υπολογιστή μπορεί να έχει τον γνωστό αριθμό πόρτας (δηλ. 80) σε όλους τους υπολογιστές, απλά δεν θα είναι αυτός ο αριθμός πόρτας με τον οποίο θα ταξιδεύσει στο Διαδίκτυο, πράγμα που δεν αποτελεί πρόβλημα. Η λύση του NAT αν και ανακουφίζει το πρόβλημα των διευθύνσεων και αποτελεί ένα ακόμα εμπόδιο για τους hackers, δεν είναι αρεστή σε πολλούς διότι παραβιάζει κάποιες αρχές των δικτύων. Δεν θα επεκταθούμε σε αυτά, ωστόσο αξίζει να σημειώσουμε ότι το NAT δεν μπορεί να λειτουργήσει το ίδιο απλά εάν η εκκίνηση της επικοινωνίας γίνει εξωτερικά του δικτύου π.χ. εάν βάλουμε ένα server στο σπίτι και κάποιος απέξω επιχειρήσει να τον προσπελάσει, ο NAT δεν θα βρει αντιστοίχιση στον πίνακά του όπως όταν η εκκίνηση γίνει από εσωτερικό υπολογιστή. Το ίδιο εάν κάποιος προσπαθήσει να εκκινήσει μια σύνδεση VoIP εξωτερικά του σπιτιού (εάν την ξεκινήσει εσωτερικός υπολογιστής, θα προκαλέσει εγγραφή αντιστοίχισης στον πίνακα του NAT και έτσι δεν θα υπάρχει πρόβλημα). Αυτά βέβαια τα προβλήματα αντιμετωπίζονται με διάφορες επίσης εμβαλωματικές λύσεις, όπως το ίδιο το NAT, αλλά τις αναφέρουμε σαν χαρακτηριστικές του ανορθόδοξου χαρακτήρα του

NAT. Σε κάθε περίπτωση μένει να δούμε εάν η επικράτηση του IPv6 θα εξαφανίσει το NAT, ή εάν θα επαληθευθεί και στην περίπτωση αυτή η ρήση «ουδέν μονιμότερον του προσωρινού».

7.4 UDP (User Datagram Protocol)

Όπως έχει τονισθεί το IP είναι ένα πρωτόκολλο που δεν έχει προβλέψεις για να εξασφαλίσει την ακεραιότητα των δεδομένων από το ένα άκρο στο άλλο και δεν διαθέτει έλεγχο ροής, επιβεβαιώσεις, και αναμεταδόσεις. Κάθε δεδομενόγραμμα αντιμετωπίζεται μόνο του χωρίς συσχέτιση με άλλα που ανήκουν στην ίδια ροή ή σύνδεση (φυσική ή νοητή). Ωστόσο δεν μπορεί να υπάρξει σωστή επικοινωνία χωρίς παρακολούθηση των ροών και αναμεταδόσεις. Αυτές οι λειτουργίες αφήνονται στα ανώτερα πρωτόκολλα του στρώματος μεταφοράς. Είδαμε πως δρα το πρωτόκολλο TCP για να εξασφαλίσει τη διόρθωση λαθών, έλεγχο ροής και συμφόρησης.
Υπάρχουν ωστόσο εφαρμογές όπου η εξασφάλιση της ακεραιότητος των δεδομένων δεν μπορεί να επιτευχθεί μέσω αναμετάδοσης διότι η επικαιρότητα τους εκπνέει πριν την ανταλλαγή επιβεβαιώσεων και αναμεταδόσεων. Σε τέτοιες εφαρμογές, που απαιτούν υψηλή ταχύτητα π.χ. κατανεμημένα συστήματα αρχείων, VoIP (φωνή μέσω IP), ζωντανό βίντεο, κτλ. η ποιότητα πρέπει να εξασφαλιστεί με άλλες μεθόδους. Οι υπηρεσίες αυτού του τύπου ανέχονται περισσότερα σφάλματα και τα υπόλοιπα πρέπει να διορθώνονται με κώδικες διόρθωσης και όχι ανίχνευσης (τεχνική που ονομάζεται FEC-Forward error correction). Όσον αφορά τις απώλειες πακέτων λόγω υπερχείλισης ταμιευτήρων, πρέπει να αποτρέπονται με προληπτικό έλεγχο συμφόρησης δηλαδή έλεγχο κατά την πρόσβαση ώστε η κίνηση να μην υπερβαίνει τις δυνατότητες του δικτύου, (όπως λειτουργεί και το τηλεφωνικό δίκτυο που μπλοκάρει δίνοντας σήμα κατειλημμένου όταν οι πόροι του δικτύου εξαντλούνται).
Για τις υπηρεσίες αυτές το TCP είναι ακατάλληλο και απαιτείται ένα ελαφρό πρωτόκολλο, λιτό με λίγες βασικές λειτουργίες. Για το σκοπό αυτό σχεδιάστηκε το πρωτόκολλο UDP το οποίο ανήκει επίσης στο στρώμα μεταφοράς και χρησιμοποιείται εναλλακτικά αντί για το TCP. Είναι πρωτόκολλο άνευ συνδέσεων (connectionless) και βασίζεται σε αριθμούς πόρτας για διευθυνσιοδότηση δηλαδή για να αναγνωρισθεί η ουρά στη οποία πρέπει να αποδοθεί το πακέτο μέσα στον υπολογιστή προορισμού όπου πιθανόν τρέχουν πολλές άλλες εφαρμογές κάθε μία με διαφορετική πόρτα. Στην διεπαφή του υπολογιστή φθάνει βάσει της διεύθυνσης IP.
Στο επόμενο σχήμα 7.1 φαίνεται η φόρμα (format) του πακέτου UDP. Περιέχει λιγότερα πεδία από το TCP, δεν διαθέτει πεδίο αρίθμησης και πεδίο επιβεβαίωσης ούτε παράθυρο ολίσθησης. Η επικεφαλίδα αποτελείται από 8 bytes και ακολουθούν τα δεδομένα που θεωρητικά μπορεί να είναι μέχρι 64k bytes αλλά πολλές εφαρμογές περιορίζουν το μέγεθος στα 8192 bytes δεδομένων. Τα δύο πρώτα bytes περιέχουν τον αριθμό πόρτας πηγής και τα άλλα δύο του προορισμού όπως και στο TCP. Ακολουθεί το μήκος πακέτου και το πεδίο ανίχνευσης λαθών όπου μπαίνει το σύνδρομο (υπόλοιπο) της διαίρεσης με το πολυώνυμο CRC που καλύπτει και την επικεφαλίδα και τα δεδομένα όπως στο TCP (υπενθυμίζεται ότι στο IP το checksum καλύπτει μόνο την επικεφαλίδα) αφήνοντας τα δεδομένα να προστατευθούν από άλλα στρώματα (ζεύξης και μεταφοράς).

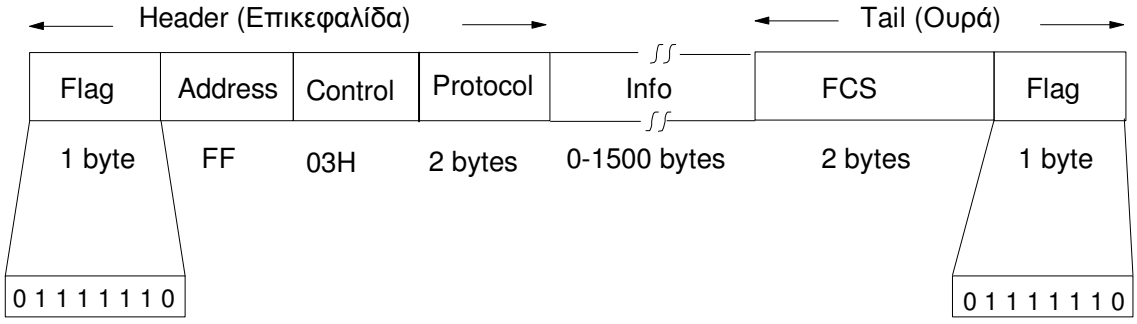


7.5 Το Πρωτόκολλο PPP

Για τη σύνδεση ξενιστών από το σπίτι όπου συνήθως δεν διατίθεται διεπαφή με δίκτυο δεδομένων όπως στις επιχειρήσεις, χρησιμοποιείται η μόνη διαθέσιμη γραμμή επικοινωνίας δηλαδή η τηλεφωνική. Η τηλεφωνική γραμμή δεν είναι φτιαγμένη για δεδομένα αλλά με την κατάλληλη διαμόρφωση τα δυαδικά ψηφία μεταβιβάζονται σαν ακουστικά σήματα. Ωστόσο αυτό δεν αρκεί καθώς απαιτούνται και οι λειτουργίες του στρώματος ζεύξης. Για το σκοπό αυτό χρησιμοποιούνται τα πρωτόκολλα SLIP (Serial Line Interface Protocol) και PPP (point-to-point protocol – σημείου-προς-σημείο). Θα πούμε δυο λόγια για το δεύτερο που έχει πλέον κυριαρχήσει αντικαθιστώντας το πρώτο. Το PPP που βασίζεται στο πρωτόκολλο HDLC (με κάποιες απλοποιήσεις) εκτελεί 3 βασικές λειτουργίες.

1. Ενθυλάκωση πακέτων IP σε πλαίσια PPP δηλαδή οριοθέτηση της αρχής και τέλους και παροχή φόρμας με τα απαραίτητα πεδία.
2. Ένα πρωτόκολλο ελέγχου ζεύξης (link control protocol – LCP), για την εγκατάσταση, διαχείριση και έλεγχο των συνδέσεων. Επιτρέπει σε κάθε άκρο να διαπραγματεύεται διαφορετικές επιλογές.
3. Μια οικογένεια πρωτοκόλλων ελέγχου δικτύου (network control protocols – NCPs) κάθενα από τα οποία αντιστοιχεί σε διαφορετικό πρωτόκολλο δικτύου.

Το επόμενο σχήμα 7.2 δείχνει ένα πλαίσιο PPP και τα διαφορετικά δεδομένα που μπορεί να μεταφέρει. Κάθε πλαίσιο αρχίζει και τελειώνει με μια σημαία (flag) ενός byte, ακολουθεί ένα πεδίο διεύθυνσης ενός byte που έχει πάντα την τιμή FFH, και ένα byte ελέγχου που πάντα έχει τιμή 03H (δηλ. αυτά τα δύο πεδία είναι ανενεργά). Στη συνέχεια έπεται ένα πεδίο πρωτοκόλλου, που έχει την ίδια σημασία με το πεδίο τύπου των πλαισίων Ethernet, δηλ. καθορίζει εάν τα δεδομένα της του πλαισίου είναι κάποιο πακέτο IP (τότε έχει τιμή 0021H), δεδομένα ελέγχου ζεύξης (τιμή C021H), ή δεδομένα ελέγχου δικτύου (τιμή 8021H). Τέλος υπάρχει και ένα πεδίο CRC για εντοπισμό λαθών στο πλαίσιο. Αξίζει να προσεχθούν οι ομοιότητες πλαισίου με το HDLC παρά τις λειτουργικές διαφορές ελλείψει ουσιαστικού πεδίου ελέγχου.



Σχήμα 7.2: Πλαίσιο PPP

Το PPP αντικατέστησε το παλαιότερο πρωτόκολλο SLIP, σε αργές σειριακές γραμμές προσφέροντας τα ακόλουθα πλεονεκτήματα συγκρινόμενο με το SLIP.

1. Υποστήριξη πολλαπλών πρωτοκόλλων σε μία σειριακή γραμμή.
2. Εντοπισμό λαθών στα πλαίσια.
3. Δυναμική διαπραγμάτευση της διεύθυνσης IP σε κάθε άκρο μέσω του πρωτοκόλλου ελέγχου δικτύου IP.
4. Συμπίεση των επικεφαλίδων TCP και IP.
5. Διαπραγμάτευση πολλών επιλογών για τη γραμμή μέσω του πρωτοκόλλου ελέγχου ζεύξης

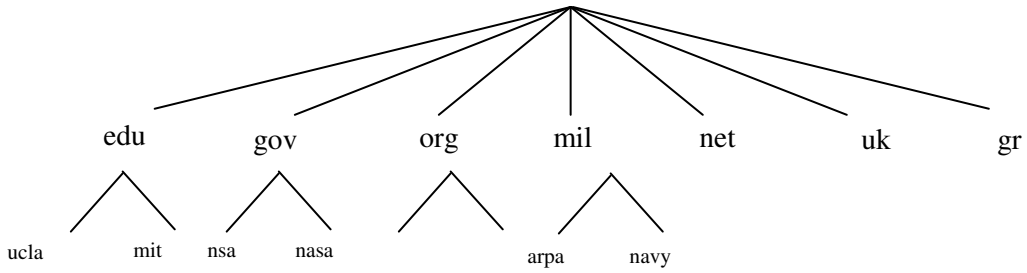
Το κόστος που προκύπτει από τα πλεονεκτήματα αυτά είναι ένα επιπλέον φορτίο 6 bytes, μερικά πλαίσια διαπραγμάτευσης κατά την εγκατάσταση της ζεύξης, και λίγο πιο πολύπλοκη υλοποίηση.

7.6 Το Domain Name System (DNS)

Όπως είναι γνωστό, το Internet Protocol (IP) χρησιμοποιεί τις μεγέθους 32 bit διευθύνσεις του Internet για την παράδοση των δεδομενογραμμάτων που περιέχουν τα δεδομένα των χρηστών. Όμως οι δυαδικές διευθύνσεις έστω και στην εστιγμένη δεκαδική μορφή τους δεν είναι ευκολομνημόνευτες από τους χρήστες καθώς δεν παρέχουν καμμία συσχέτιση με τα περιεχόμενα ή τους οργανισμούς που αντιπροσωπεύουν. Επίσης οι αριθμητικές διευθύνσεις IP συχνά αλλάζουν και θα πρέπει οι χρήστες να παρακολουθούν τις μεταβολές τους.
Σε αντίθεση όμως με την τηλεφωνία όπου επικράτησε η χρήση των αριθμητικών διευθύνσεων ελλείψει καλύτερης λύσης, οι υπολογιστές μπορούν να μας βοηθήσουν στην μετάφραση των αριθμητικών διευθύνσεων σε άλλες πιο ανθρωποκεντρικές και ευμνημόνευτες με βάσεις δεδομένων που αντιστοιχίζουν τα ονόματα ξενιστών με τις διευθύνσεις τους. Αλλά γιατί να μην καταργηθούν οι δυαδικές διευθύνσεις εντελώς; Μπορεί τα ονόματα να απομνημονεύονται πιο εύκολα από τους ανθρώπους αφού σχετίζονται με γνωστή θεματολογία αλλά οι δυαδικές διευθύνσεις είναι πολύ πιο αποδοτικές κατά την αναζήτηση τους στους πίνακες των δρομολογητών. Επιπλέον, τα ονόματα δεν περιέχουν πληροφορίες που θα βοηθήσουν το δίκτυο να εντοπίσει τον ξενιστή.
Εξαιτίας του μεγέθους του, το Internet έχει ένα καλά οργανωμένο σύστημα αντιστοίχισης ονομάτων – το *Domain Name System* (DNS). Να σημειώσουμε ότι το Internet δεν χρησιμοποιούσε πάντα το DNS. Όταν υπήρχαν μόνο λίγοι εκατοντάδες υπολογιστές στο Internet, μια κεντρική υπηρεσία που ονομαζόταν Network Information Center (NIC) διατηρούσε ένα αρχείο με αντιστοιχίες ονομάτων με διευθύνσεις. Αυτό το αρχείο ονομαζόταν *hosts.txt*. Όταν μια τοποθεσία ήθελε να εντάξει ένα νέο υπολογιστή στο Internet, ο υπεύθυνος της τοποθεσίας έστελνε μήνυμα στο NIC δίνοντας το όνομα και την αντίστοιχη διεύθυνση του ξενιστή. Τα στοιχεία αυτά στη συνέχεια τα εκχωρούσαν στο αρχείο .txt . Το τροποποιημένο αρχείο στελνόταν στις διάφορες τοποθεσίες κάθε λίγες μέρες, και ο υπεύθυνος της κάθε τοποθεσίας με τη σειρά του το έστελνε σε κάθε ξενιστή του. Σήμερα την ευθύνη της ανάθεσης ονομάτων έχει το ICANN (Intenet Corporation for Assigned Names and Numbers-www.icann.org).
Προφανώς το σύστημα αυτό δεν μπορούσε να παρακολουθήσει την κλίμακα μεγέθυνσης του Διαδικτύου και έπρεπε να εισαχθεί ένα κλιμακωτό και κατανεμημένο σύστημα που θα μπορούσε να μεγαλώνει με τον αριθμό ξενιστών. Οπότε, στα μέσα της δεκαετίας του `80 τέθηκε σε εφαρμογή το DNS. Αναπτύχθηκε από την εταιρεία Sun Microsystems και η εν λόγω πρόταση της Sun υιοθετήθηκε εν συνεχεία ως το σύστημα διευθυνσιοδότησης του διαδικτύου. Το σύστημα αυτό βασίζεται σε μία ιεραρχία από εξυπηρετητές ονομάτων. Πριν την αρχική αποστολή ενός πακέτου προς διεύθυνση της οποίας δεν είναι γνωστή η διεύθυνση IP αλλά μόνο το URL (Universal Resource Locator) ή η διεύθυνση e-mail, η εφαρμογή (π.χ. ο φυλλομετρητής – browser) θα στείλει πρώτα ένα πακέτο στον πλησιέστερο Name server (του οποίου την διεύθυνση IP έχουμε δώσει στον υπολογιστή μας όταν εισαγάγαμε τα στοιχεία του TCP (start, settings, control panel, network, TCP protocol, enable DNS). Όταν επιστρέψει η απάντηση με τη διεύθυνση IP, θα γίνει η αποστολή του πακέτου.

7.6.1 Ιεραρχία στο DNS

Το DNS χρησιμοποιεί έναν ιεραρχικό χώρο ονομάτων. Παρόλο που η επεξεργασία των ονομάτων στο DNS γίνεται από τα δεξιά προς τα αριστερά, οι χρήστες τα διαβάζουν από αριστερά στα δεξιά. Π.χ. Plato.telecom.teipir.gr Η ιεραρχία στο DNS μπορεί να απεικονιστεί με ένα δέντρο (ιδέ σχ. 7.3), όπου κάθε κόμβος αντιστοιχεί σε μία επικράτεια (domain), και τα φύλλα στο δέντρο αντιπροσωπεύουν τους υπολογιστές.
Οι επικράτειες που βρίσκονται στο ανώτερο επίπεδο της ιεραρχίας διατηρούν καταλόγους και διευθύνσεις των επικρατειών που βρίσκονται στο αμέσως κατώτερο επίπεδο από αυτά. Αυτές οι δευτερεύουσες επικράτειες έχουν παρόμοιες αρμοδιότητες για τις επικράτειες που βρίσκονται στο αμέσως κατώτερο επίπεδο κ.ο.κ. Κατ' αυτόν τον τρόπο, κάθε υπολογιστής του Δικτύου αποκτά μία διεύθυνση που αποτελείται από χαρακτήρες ASCII.

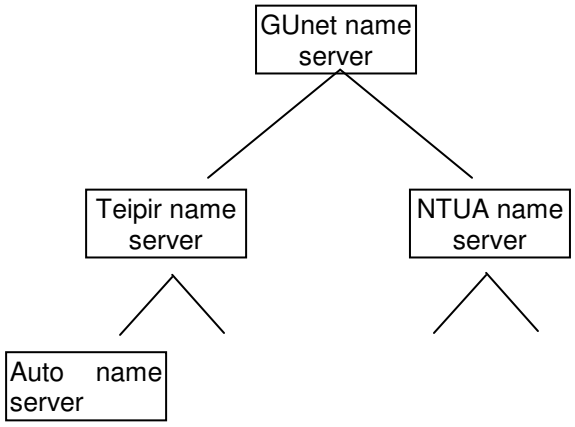


Σχήμα 7.3: Παράδειγμα ιεραρχίας στο DNS

Μία διεύθυνση ηλεκτρονικού ταχυδρομείου στο Internet αποτελείται από δύο κύρια τμήματα τα οποία διαχωρίζονται από το σύμβολο @ (at). Ας πάρουμε για παράδειγμα τη διεύθυνση user@teipir.gr. Το πρώτο τμήμα της διεύθυνσης – το οποίο βρίσκεται αριστερά του συμβόλου @ - είναι το όνομα του χρήστη, το οποίο συνήθως αναφέρεται στο πρόσωπο που διατηρεί ένα λογαριασμό (account) στο Internet και συνήθως είναι το όνομα που χρησιμοποιεί για login. Το δεύτερο τμήμα της διεύθυνσης που βρίσκεται δεξιά του συμβόλου @, δηλώνει το όνομα του host ή το επικράτεια name, το οποίο καθορίζει τον συγκεκριμένο υπολογιστή στον οποίο ο χρήστης διατηρεί το λογαριασμό του στο Internet.
Για να γίνει η μετάφραση των ονομάτων όσον το δυνατόν αποτελεσματικότερα, το Internet έχει οργανωθεί σε ορισμένες μεγάλες επικράτειες. Οι εν λόγω επικράτειες αναφέρονται με γράμματα στο τέλος της διεύθυνσης π.χ. .com. Οι συνηθέστερες επικράτειες (domains) στις ΗΠΑ είναι το .com για εμπορική χρήση, το .edu για την εκπαίδευση, το .gov για την κυβέρνηση, το .mil για τον στρατό, το .net για δίκτυα (το οποίο χρησιμοποιείται ευρέως από ISPs καθώς και από εταιρίες και ομάδες που σχετίζονται με την οργάνωση του Internet) και το .org για οργανισμούς. Το τελευταίο διάστημα έχουν προταθεί διάφορα σχέδια για την προσθήκη νέων επικρατειών όπως το .biz για επιχειρήσεις (business).name. (για προσωπικές τοποθεσίες ατόμων, .pro, .coop, .info κτλ. Στις υπόλοιπες χώρες του κόσμου πλην των ΗΠΑ χρησιμοποιούνται μόνο δύο γράμματα για τον καθορισμό των επικρατειών όπως για παράδειγμα το .gr για την Ελλάδα.

7.6.2 Servers Ονομάτων (Name Servers)

Το σύστημα DNS παρακολουθεί τις αλλαγές των διευθύνσεων IP έτσι ώστε αν υπάρξει μία αλλαγή στην διεύθυνση να εξακολουθεί να ισχύει η διεύθυνση URL και e-mail. Ορισμένοι υπολογιστές παίζουν το ρόλο εξυπηρετητών ονομάτων (name servers), δηλαδή είναι υπεύθυνοι για την αντιστοίχιση μεταξύ διευθύνσεων IP και URL / e-mail καθώς επίσης και των αλλαγών. Οι εξυπηρετητές ονομάτων είναι οργανωμένοι ιεραρχικά. Η ιεραρχία αυτή υλοποιείται σε *ζώνες*. Κάθε ζώνη συνήθως βασίζεται σε μια διοικητική αρχή η οποία είναι υπεύθυνη για το συγκεκριμένο μέρος του δικτύου. Το πρώτο επίπεδο της ιεραρχίας σχηματίζει μια ζώνη την οποία διαχειρίζεται το NIC (σήμερα έχει μετονομαστεί). Πολλά επίπεδα πιο κάτω βρίσκονται οι εξυπηρετητές ονομάτων της Ελλάδος και εκεί αυτός του Πανεπιστημιακού δικτύου Gunet (Greek University network). Από κάτω υπάρχει μια ζώνη που αντιστοιχεί π.χ. στο ΤΕΙ Πειραιά και μια π.χ. στο ΕΜΠ. Κάποια τμήματα παραμένουν στη ζώνη όλου του ΤΕΙ ενώ κάποια άλλα π.χ. του Αυτοματισμού μπορεί να επιλέξουν το δικό τους επίπεδο έχοντας την δικιά τους ζώνη.
Οι πληροφορίες που περιέχονται σε κάθε ζώνη περιέχονται σε δύο ή περισσότερους servers ονομάτων. Κάθε server είναι ένα πρόγραμμα στο οποίο μπορεί κανείς να έχει πρόσβαση μέσω του Internet. Οι πελάτες στέλνουν ερωτήματα στους server ονομάτων και αυτοί ανταποκρίνονται στέλνοντας τις κατάλληλες πληροφορίες. Μερικές φορές οι πληροφορίες μπορεί να έχουν την τελική απάντηση που ο πελάτης επιθυμεί ή να παραπέμπουν σε άλλον server που ο πελάτης μπορεί να ρωτήσει στην συνέχεια. Από μια άποψη, είναι καλύτερο να πούμε ότι το DNS παρουσιάζεται από μια ιεραρχία με servers ονομάτων παρά από ιεραρχία με domains, όπως απεικονίζεται στο σχήμα 7.4.



Σχήμα 7.4: Ιεραρχία των name server.

7.7 Εφαρμογές Telnet και Rlogin

Μια από τις πρώτες δυνατότητες του Internet ήταν η πρόσβαση σε απομακρυσμένους υπολογιστής σε οποιοδήποτε σημείο πάνω στη γη από ένα απλό τερματικό. Τα πρώτα χρόνια του Διαδικτύου δεν υπήρχε η έννοια του προσωπικού υπολογιστή. Ένας ξενιστής συνήθως επέτρεπε σε πολλούς διαφορετικούς χρήστες να έχουν πρόσβαση στους πόρους του την ίδια στιγμή. Η πρόσβαση αυτή γινόταν με δύο απλά πρωτοκόλλα το Telnet και το Rlogin που παρέχουν τη δυνατότητα απομακρυσμένης σύνδεσης η οποία παραμένει από τις πιο δημοφιλής εφαρμογές του Internet αν και υπάρχει η τάση περιορισμού τους λόγω καταχρήσεων από Χακερς.
Το Telnet είναι μια εφαρμογή που προσφέρεται σχεδόν σε κάθε υλοποίηση TCP/IP και λειτουργεί μεταξύ υπολογιστών που χρησιμοποιούν διαφορετικά λειτουργικά συστήματα. Χρησιμοποιεί την διαδικασία διαπραγμάτευσης μεταξύ πελάτη και εξυπηρετητή (client-server) και προσδιορίζει ποιες δυνατότητες μπορούν να εκτελεσθούν από το κάθε τερματικό.
Στη σύνδεση μέσω Telnet, θα πρέπει να χρησιμοποιείται τερματική εξομοίωση, που στην ουσία σημαίνει ότι οι λειτουργίες του πληκτρολογίου και της οθόνης θα γίνονται όπως τις αναμένει ο ξενιστής (host). Η πιο συνηθισμένη τερματική εξομοίωση είναι ο VT-100, οπότε αν χρησιμοποιηθεί Telnet αυτός είναι ο πιο ασφαλής εξομοιωτής για χρήση.
Όταν ένας υπολογιστής έρχεται σε επαφή με κάποιον άλλο μέσω του telnet, οι δύο υπολογιστές συνεννοούνται για το πως θα επικοινωνούν αναμεταξύ τους και αποφασίζουν ποια τερματική εξομοίωση θα χρησιμοποιήσουν. Η τερματική εξομοίωση καθορίζει πως το πληκτρολόγιο θα μεταδίδει δεδομένα στον απομακρυσμένο υπολογιστή και πως αυτά τα δεδομένα θα εμφανίζονται στην οθόνη. Για παράδειγμα, καθορίζει πως θα δουλεύει το πλήκτρο back-space.
Το πρωτόκολλο telnet υποθέτει ότι οι δύο υπολογιστές είναι ένα Network Virtual Terminal (NVT). Κάθε NVT έχει έναν εικονικό εκτυπωτή και ένα εικονικό πληκτρολόγιο. Το πληκτρολόγιο στέλνει δεδομένα από το ένα NVT στο άλλο. Όταν πληκτρολογήσετε κάτι τότε χρησιμοποιείτε το πληκτρολόγιο NVT. Ο εκτυπωτής δεν υπάρχει στην πραγματικότητα και ότι λαμβάνει το εμφανίζει στην οθόνη του υπολογιστή.
Το κείμενο που πληκτρολογείτε αποθηκεύεται σε ένα buffer (μνήμη προσωρινής αποθήκευσης) στον υπολογιστή σας. Όταν μια ολοκληρωμένη γραμμή δεδομένων είναι έτοιμη για μετάδοση, ή όταν δώσετε εντολή για μετάδοση δεδομένων, αυτά στέλνονται στο Internet από το δικό σας πληκτρολόγιο NVT. Μαζί με τα δεδομένα βρίσκεται και η IP διεύθυνση του host, με την οποία εξασφαλίζουμε ότι το πακέτο δεδομένων θα φτάσει στο κατάλληλο σημείο.
Ο Telnet host δέχεται τα δεδομένα που έχετε στείλει. Τα επεξεργάζεται και επιστρέφει στον εκτυπωτή NVT, δηλαδή την οθόνη σας, τα αποτελέσματα ή εκτελεί τις εντολές σε ένα απομακρυσμένο υπολογιστή. Για

παράδειγμα, αν δώσετε την εντολή "DIR" και πατήσετε το ENTER, ο απομακρυσμένος υπολογιστής θα την εκτελέσει και θα σας αποστείλει το αποτέλεσμα.
Επειδή τα πακέτα φεύγουν προς κάθε κατεύθυνση μεταξύ του υπολογιστή σας και του host, ίσως υπάρξει μια χρονική καθυστέρηση μεταξύ της στιγμής που θα στείλετε τα δεδομένα μέχρι να εμφανιστούν τα αποτελέσματα στην οθόνη σας.
Το Rlogin είναι περισσότερο απλό από το Telnet και αναπτύχθηκε για να δουλεύει μόνο μεταξύ συστημάτων Unix, αλλά τα τελευταία χρόνια έχει εισαχθεί και στα Windows και σε άλλα λειτουργικά συστήματα. ¨όπως και το telnet προσφέρει σύνδεση από απόσταση από έναν πελάτη υπολογιστή σε ένα υπολογιστή εξυπηρετητή, ώστε να μας επιτραπεί να εκτελέσουμε προγράμματα στον εξυπηρετητή. Και οι δύο εφαρμογές χρησιμοποιούν την κατάσταση άμεσης προτεραιότητας (urgent mode) του TCP για να στείλουν από τον server στον πελάτη εντολές ακόμη και αν η ροή των δεδομένων σε αυτή τη κατεύθυνση έχει διακοπεί από τον έλεγχο ροής.

7.8 FTP : File Transfer Protocol - Πρωτόκολλο Μεταφοράς Αρχείου

Το FTP είναι μια ευρέως χρησιμοποιούμενη αυτοτελής εφαρμογή, από τις πρώτες του Διαδικτύου και περιγράφεται στο RFC959 της IETF. Σήμερα τείνει να ενσωματωθεί στούς φυλλομετρητές και να μην είναι άμεσα ορατή με τη δική της γραφική διεπαφή χρήστη (GUI- graphical user interface)[*]. Ο σκοπός του ftp είναι να εκτελεί αξιόπιστη μεταφορά αρχείων (file transfer) ανάμεσα σε δύο υπολογιστές ανεξαρτήτως απόστασης, διαφορετικού λειτουργικού συστήματος, διαφορετική δομή συστήματος αρχείου (π.χ. NFS, FAT16, FAT32), διαφορετικούς χαρακτήρες κτλ. Το αρχείο φυαικά παραμένει στον υπολογιστή προέλευσης δηλαδή πρόκειται για είναι μια πλήρης αντιγραφή ενός αρχείου από το ένα σύστημα αρχείων σε άλλο. Το FTP μπορεί να χειριστεί όλες τις διαφορές μεταξύ ανόμοιων συστημάτων χρησιμοποιώντας κάθε φορά μια διαφορετική προσέγγιση. Το FTP υποστηρίζει ένα ορισμένο αριθμό τύπων αρχείου (ASCII, δυαδικό, κτλ.) και δομών αρχείου. Συνήθως για να χρησιμοποιήσουμε το FTP χρειαζόμαστε έναν λογαριασμό για να έχουμε πρόσβαση στον εξυπηρετητή (server), ή το χρησιμοποιούμε σε έναν server που δέχεται *ανώνυμο* FTP.
Το FTP διαφέρει από άλλα πρωτόκολλα διότι χρησιμοποιεί δυο συνδέσεις TCP για να μεταφέρει ένα αρχείο.

1. Η *σύνδεση ελέγχου* (control connection) έχει να κάνει με την συνδιάλεξη πελάτη και server. Ο server ανοίγει τη γνωστή θύρα (port) 21 για το FTP και περιμένει να συνδεθεί ο πελάτης. Ο πελάτης ανοίγει την TCP θύρα 21 για να πραγματοποιήσει την σύνδεση ελέγχου. Η σύνδεση ελέγχου διαρκεί για όσο χρόνο επικοινωνεί ο πελάτης με τον server. Αυτή η σύνδεση χρησιμεύει για την μεταφορά των εντολών από τον πελάτη στον server και των απαντήσεων του server. Ο IP τύπος υπηρεσίας για την σύνδεση ελέγχου πρέπει να είναι η "*ελάχιστη καθυστέρηση*" αφού οι εντολές γράφονται από τον χρήστη.
2. Μια *σύνδεση δεδομένων* (data connection) δημιουργείται κάθε φορά που ένα αρχείο μεταφέρεται μεταξύ πελάτη και server. Ο IP τύπος υπηρεσίας για την σύνδεση δεδομένων πρέπει να είναι η "*μέγιστη απόδοση*" αφού αυτή η σύνδεση είναι για μεταφορά αρχείου.

Οι πιο συνήθεις επιλογές του πρωτοκόλλου FTP για τη διευθέτηση του τρόπου με τον οποίο θα μεταφέρεται και θα αποθηκεύεται ένα αρχείο είναι: Τύπος αρχείου (File Type) ASCII (Default) όπου το κείμενο μεταφέρεται δια μέσου της σύνδεσης δεδομένων σαν χαρακτήρες ASCII, ή Δυαδικός τύπος αρχείου όπου η πληροφορία στέλνεται σαν μια συνεχόμενη σειρά από bits.

7.8.1 Οι Εντολές του FTP

Οι εντολές και οι απαντήσεις που στέλνονται δια μέσου της σύνδεσης ελέγχου μεταξύ πελάτη και server είναι σε ASCII. Οι μόνες εντολές telnet που μπορεί να στείλει ο πελάτης στον server είναι εντολές διακοπής. Αυτές οι εντολές telnet χρησιμοποιούνται για να εγκαταλείψουμε την μεταφορά ενός αρχείου που βρίσκεται σε εξέλιξη, ή να στέλνουμε ερωτήματα στον server ενώ μεταφέρεται ένα αρχείο.

[*] Σημ. Η γραφική διεπαφή χρήστη είναι το πρόγραμμα που φέρνει στην οθόνη τα μενού και τα λοιπά στοιχεία μέσω των οποίων ο χρήστης δίνει εντολές στο πρόγραμμα και εισάγει στοιχεία. Δηλαδή αυτά που βλέπουμε στη οθόνη και αναγνωρίζουμε ποιο είναι το πρόγραμμα π.χ. Word, explorer, κτλ.

Οι εντολές είναι 3 ή 4 bytes από κεφαλαίους χαρακτήρες ASCII. Παραπάνω από 30 διαφορετικές FTP εντολές μπορεί να στείλει ο πελάτης στον server. Πάντως αυτό που αξιοσημείωτο και πρέπει να αντιληφθεί ο σύγχρονος χρήστης, είναι ότι αυτές οι παλιές εφαρμογές όπως το telnet και το ftp βασίζονται σε απλή ανταλλαγή εντολών σε μορφή ASCII strings που κάποτε έπρεπε να τις γνωρίζει και να τις συντάξει σωστά στο τερματικό του ο χρήστης για να πετύχει το σωστό αποτέλεσμα. Σήμερα οι ίδιες εντολές αποστέλλονται από τον ένα υπολογιστή στον άλλο χωρίς ωστόσο πια να χρειάζεται να τις γνωρίζει ο χρήστης. Το γραφικό περιβάλλον που διαθέτουμε μας επιτρέπει να κάνουμε χειρισμούς με το ποντίκι στην οθόνη και το λογισμικό της διεπαφής χρήστη τις μεταφράζει σε εντολές που αποστέλλονται στον απέναντι εξυπηρετητή. Π.χ. σύροντας το όνομα ενός αρχείου (drag and drop) μπορούμε να προκαλέσουμε το κατέβασμα (download) του αρχείου. Για να πάρει μια ιδέα ο αναγνώστης μερικές από τις εντολές είναι:

- RETR, (Retrieve). Κατεβάζει (download) το αρχείο του οποίου το όνομα ακολουθεί
- STOR, (Store). Ανεβάζει (upload), δηλ. στέλνει το αρχείο που ακολουθεί
- STOU Store Unique. Ανεβάζει το αρχείο ελέγχοντας τη μοναδικότητα του (δεν γράφει πάνω σε άλλο ομώνυμο αρχείο)
- APPE, Append. Ανεβάζει και γράφει σαν συνέχεια ενός αρχείου
- DELE, Delete. Διαγράφει το οριζόμενο αρχείο
- CWD, Change Working Directory. Αλλάζει κατάλογο
- MKD, Make Directory. Δημιουργεί νέο κατάλογο
- RMD, Remove Directory. Διαγράφει κατάλογο
- PWD, Print Working Directory. Ενημερώνει για τον κατάλογο στον οποίο βρισκόμαστε

Οι αποκρίσεις στις εντολές αυτές είναι τριψήφιοι αριθμοί σε κωδικούς ASCII, με ένα προαιρετικό μήνυμα να τους συνοδεύει. Το λογισμικό χρειάζεται μόνο να δει τον αριθμό για να αποφασίσει πώς να επεξεργαστεί την απάντηση, το μήνυμα είναι απλά ενημερωτικό για τους χρήστες. Ένας έμπειρος χρήστης θα μπορούσε να καταλάβει το νόημα μιας απάντησης διαβάζοντας μόνο το μήνυμα (χωρίς να χρειάζεται να μνημονεύσει το κωδικό κάθε απάντησης) αλλά οι σύγχρονες υλοποιήσεις δεν μας στέλνουν τίποτε από τα δυο, απλά βλέπουμε το απτό αποτέλεσμα που είναι και η ουσία.
Παρακάτω φαίνεται το νόημα μερικών κωδικών για να δοθεί μια γεύση:

- 1yz: θετική προκαταρκτική απάντηση. Η ενέργεια ξεκινάει αλλά αναμένεται άλλη μια απάντηση προτού σταλεί άλλη εντολή.
- 2yz: θετική ολοκληρωμένη απάντηση. Μια νέα εντολή μπορεί να αποσταλεί.
- 3yz: θετική ενδιάμεση απάντηση. Η εντολή έχει γίνει αποδεκτή αλλά πρέπει να σταλεί μια ακόμη εντολή για ολοκλήρωση της ενέργειας.

7.9. Ηλεκτρονικό Ταχυδρομείο

Το ηλεκτρονικό ταχυδρομείο (electronic mail) ή e-mail αποτελεί το συχνότερα χρησιμοποιούμενο χαρακτηριστικό του Internet. Μπορείτε να το χρησιμοποιείτε για να στέλνετε μηνύματα σε οποιονδήποτε είναι συνδεδεμένος στο διαδίκτυο ή σε ένα δίκτυο το οποίο διαθέτει μία σύνδεση στο Internet. Εκατομμύρια χρήστες στέλνουν και λαμβάνουν e-mail καθημερινά. Τα μηνύματα ηλεκτρονικού ταχυδρομείου στέλνονται με τον ίδιο τρόπο όπως και τα περισσότερα δεδομένα του Internet.
Στα μηνύματα ηλεκτρονικού ταχυδρομείου μπορείτε επίσης να επισυνάψετε δυαδικά αρχεία, όπως εικόνες, βίντεο, ήχους και εκτελέσιμα αρχεία. Επειδή το Internet δεν μπορεί να διαχειριστεί απ' ευθείας δυαδικά αρχεία στο ηλεκτρονικό ταχυδρομείο, το αρχείο πρέπει πρώτα να κωδικοποιηθεί με τη χρήση ενός σχήματος κωδικοποίησης. Τα πιο δημοφιλή σχήματα είναι το MIME και το uuencode. Ο χρήστης που λαμβάνει το επισυναπτόμενο δυαδικό αρχείο (το οποίο ονομάζεται attachment) πρέπει να αποκωδικοποιήσει το αρχείο με το ίδιο σχήμα που χρησιμοποιήθηκε για την κωδικοποίηση. Τα περισσότερα πακέτα ηλεκτρονικού ταχυδρομείου αναλαμβάνουν πλέον αυτή τη διαδικασία αυτόματα.
Η λίστα ταχυδρόμησης (mailing list) ή αλλιώς ανακλαστήρας (reflector) αποτελεί μια από τις πιο ενδιαφέρουσες χρήσεις του ηλεκτρονικού ταχυδρομείου και συνδέει μία ομάδα ανθρώπων που ενδιαφέρονται για το ίδιο αντικείμενο. Όταν κάποιος στέλνει ένα μήνυμα στο mailing list, το μήνυμα στέλνεται αυτόματα σε οποιονδήποτε περιλαμβάνεται στον κατάλογο.
Στο παρελθόν ήταν πολύ δύσκολο να βρεις την ηλεκτρονική διεύθυνση κάποιου αν ήξερες μόνο το όνομά του. Τώρα αυτό δεν είναι και πολύ δύσκολο. Στο Internet υπάρχει μία ποικιλία directories τα οποία σας επιτρέπουν να βρείτε εύκολα την ηλεκτρονική διεύθυνση κάποιου χρήστη. Τα εν λόγω sites χρησιμοποιούν κυρίως ένα πρότυπο ονόματι Lightweight Directory Access Protocol (LDAP) το οποίο σας επιτρέπει να βρείτε την ηλεκτρονική διεύθυνση κάποιου χωρίς να χρειαστεί να επισκεφθείτε κάποιο Web site. Χρησιμοποιώντας αυτό το πρωτόκολλο, μπορείτε να αναζητήσετε για μια ηλεκτρονική διεύθυνση στο Internet απ' ευθείας από το πρόγραμμα ηλεκτρονικού ταχυδρομείου που χρησιμοποιείτε.
Καθώς περιγράφονται όλες οι εφαρμογές σε αυτή την ενότητα, για να καταλάβουμε πώς λειτουργεί το email πρέπει να διαχωρίσουμε τη διεπαφή του χρήστη από το θεμελιώδες πρωτόκολλο μεταφοράς (στη περίπτωση μας το SMTP), και δεύτερον να κάνουμε το διαχωρισμό μεταξύ αυτού του πρωτοκόλλου μεταφοράς και ενός συνοδευτικού πρωτοκόλλου (RFC 822 και MIME) το οποίο ορίζει τη μορφή του μηνύματος που ανταλλάσσεται. Η ανταλλαγή mail στην οποία χρησιμοποιείται TCP εκτελείται από ένα μέσο μεταφοράς μηνύματος (message transfer agent /MTA). Οι χρήστες δεν έρχονται σε επαφή με το MTA το οποίο είναι υπευθυνότητα του διαχειριστή του συστήματος να εγκαταστήσει το τοπικό MTA.

7.9.1. Μορφή Μηνύματος

Το RFC 822 καθορίζει τα μηνύματα ώστε να έχουν δύο μέρη: μια επικεφαλίδα και ένα σώμα. Και τα δύο μέρη αναπαρίστανται σε κείμενο ASCII. Αρχικά, το σώμα θεωρήθηκε ότι θα είναι ένα απλό κείμενο. Αυτό συμβαίνει ακόμη και σήμερα, αν και το RFC 822 έχει επεκταθεί από το MIME ώστε να επιτρέπει στο σώμα του μηνύματος να μεταφέρει όλα τα είδη δεδομένων. Αυτά τα δεδομένα αναπαρίστανται ακόμη σαν ASCII κείμενο, αλλά επειδή μπορεί είναι μια κωδικοποιημένη έκδοσή του (π.χ. μια εικόνα JPEG) δεν είναι απαραίτητα αναγνώσιμο από ανθρώπους.
Η επικεφαλίδα του μηνύματος είναι μια σειρά από <CRLF> - τελικές γραμμές Η επικεφαλίδα χωρίζεται από το σώμα του μηνύματος. Κάθε γραμμή της επικεφαλίδας περιέχει ένα τύπο και τιμή που χωρίζονται με ένα σύμβολο. Για παράδειγμα, η επικεφαλίδα To: προσδιορίζει τον αποδέκτη του μηνύματος και η επικεφαλίδα Subject: αναφέρεται στον σκοπό του μηνύματος.
Οι άλλες επικεφαλίδες αντικαθίστανται από το σύστημα διανομής της βασικής αλληλογραφίας. Π.χ. Date: όταν το μήνυμα μεταδόθηκε, From: ποιος χρήστης στέλνει το μήνυμα, και Received: για κάθε server που διαχειρίστηκε το μήνυμα. Υπάρχουν φυσικά πολλές επικεφαλίδες, και ο ενδιαφερόμενος αναγνώστης παραπέμπεται στο RFC 822.
Το RFC 822 επεκτάθηκε το 1993 (και εκσυγχρονίστηκε ξανά το 1996) ώστε να επιτρέπει μέσω της αλληλογραφίας να μεταφέρονται και άλλοι τύποι δεδομένων: ήχος, βίντεο, εικόνες, κείμενα κ.λ.π. Το MIME

αποτελείται από τρία βασικά μέρη. Το πρώτο μέρος είναι μια συλλογή από επικεφαλίδες οι οποίες αυξάνουν το αρχικό τμήμα που ορίζεται από το RFC 822. Αυτές οι επικεφαλίδες περιγράφουν με διαφορετικούς τρόπους, τα δεδομένα που μεταφέρονται από το σώμα του μηνύματος. Περιλαμβάνουν, την MIME-Version: (η έκδοση του MIME που χρησιμοποιείται), την Content-Description (μια περιγραφή αναγνώσιμη από ανθρώπους για το τι είναι το μήνυμα, ανάλογα με το θέμα Subject), τον Content-Type: (ο τύπος των δεδομένων που περιέχονται στο μήνυμα), και το Content-Transfer-Encoding (για το πώς είναι κωδικοποιημένα τα δεδομένα μέσα στο μήνυμα).
Το δεύτερο μέρος είναι ορισμοί για μια ομάδα από τύπους που περιέχονται (και υποκατηγορίες). Για παράδειγμα, το MIME ορίζει δύο διαφορετικού τύπους εικόνας, δηλωμένη εικόνα/gif και εικόνα/jpeg, καθένας με προφανές νόημα. Σε άλλο παράδειγμα, το text/plain αναφέρεται σε ένα απλό κείμενο που μπορούμε να βρούμε στο στυλ ενός μηνύματος στο 822, ενώ το text/richtext σημαίνει ότι το μήνυμα περιέχει εμπλουτισμένο κείμενο. Σαν τρίτο παράδειγμα, το MIME ορίζει ένα τύπο εφαρμογής, όπου τα στοιχεία αναφέρονται στο νόημα των διαφορετικών προγραμμάτων εφαρμογής.
Το τρίτο μέρος είναι ένας τρόπος για να κωδικοποιήσουμε τους διάφορους τύπους δεδομένων έτσι ώστε να μπορεί να ενσωματωθεί σε ένα ηλ. Ταχυδρομείο που επιτρέπει μόνο χαρακτήρες ASCII. Το πρόβλημα είναι ότι για μερικούς τύπους δεδομένων (μια jpeg εικόνα, για παράδειγμα), κάθε byte μέσα στην εικόνα μπορεί να περιέχει μία από τις 256 διαφορετικές τιμές. Ωστόσο μόνο ένα υποσύνολο από αυτές τις τιμές είναι έγκυροι χαρακτήρες ASCII. Πρέπει να κωδικοποιηθούν όλα τα byte που δεν αντιστοιχούν σε ASCII ώστε να μπορούν να περάσουν μέσα από ενδιάμεσα συστήματα (πύλες ταχυδρομείου) τα οποία υποθέτουν ότι όλα τα email είναι ASCII και παραμορφώνουν το μήνυμα σε περίπτωση που δεν περιέχει χαρακτήρες ASCII. Για τον έλεγχο αυτού του ζητήματος, το MIME χρησιμοποιεί μια άμεση κωδικοποίηση δεκαδικών δεδομένων μέσα στο σύνολο των χαρακτήρων ASCII. Η κωδικοποίηση αυτή ονομάζεται base64. Το νόημα είναι ότι αντιστοιχίζονται κάθε τρία bytes του αρχικού δεκαδικού δεδομένου σε τέσσερις χαρακτήρες ASCII. Αυτό γίνεται με την ομαδοποίηση του δεκαδικού δεδομένου σε μονάδες των 24-bit και τεμαχίζοντας κάθε τέτοια μονάδα σε τέσσερα κομμάτια των 6-bit. Κάθε κομμάτι των 6-bit αντιστοιχίζεται σε ένα από τους 64 ισχύοντες χαρακτήρες ASCII. Για παράδειγμα, το 0 αντιστοιχεί στα A, το 1 στο B, κ.λ.π. αν προσέξουμε στο μήνυμα που έχει κωδικοποιηθεί από το σύστημα base64 θα παρατηρήσουμε μόνο τα 52 άνω και κάτω γράμματα, τα ψηφία από το 0 έως το 9 και τους ειδικούς χαρακτήρες + και /. Αυτές είναι οι 64 πρώτες τιμές του σετ του ASCII χαρακτήρα.
Έτσι λοιπόν κάνουμε όσο πιο ανώδυνη γίνεται την ανάγνωση αλληλογραφίας για τους περισσότερους από εμάς που χρησιμοποιούμε μόνο κείμενα καθώς ένα μήνυμα MIME το οποίο αποτελείται από ένα απλό κείμενο μπορεί να κωδικοποιηθεί χρησιμοποιώντας 7-bit ASCII. Συμπερασματικά, ένα μήνυμα το οποίο περιέχει μερικά απλά κείμενα, μια εικόνα jpeg, και ένα υστερόγραφο θα εμφανιστεί περίπου σαν το παρακάτω:

```
MIME-Version: 1.0
Content-Type:multipart/mixed; boundary=" -------417CA6E2DE4ABCAFBC5"
From: alica Smith Alica@cisco.com
To: bob@cs.Princeton.edu
Subject: promised material
Date: Mon, 07 Sep 1998 19:45:19 -0400

--------417CA6E2DE4ABCAFBC5
Content-Type : text/plain ; charset=us-ascii
Content-Transfer-Encoding : 7bit

Bob

Εδώ είναι η εικόνα jpeg και το σχέδιο
αναφοράς που σου υποσχέθηκα.

--Alice
```

```
--------417CA6E2DE4ABCAFBC5
Content-Type: image/jpeg
Content-Transfer-Encoding : base64
:
:μη αναγνώσιμη κωδικοποίηση μια απεικόνισης jpeg

--------417CA6E2DE4ABCAFBC5
Content-Type: application/postscript; name="draft.ps"
Content-Transfer-Encoding : 7bit
:
: αναγνώσιμη κωδικοποίηση του κειμένου του υστερόγραφου
```

Σε αυτό το παράδειγμα, η γραμμή του Content-Type στην επικεφαλίδα του μηνύματος αναφέρει ότι το μήνυμα περιέχει ποικίλα κομμάτια, κάθε ένα δηλωμένο από ένα χαρακτήρα – σύμβολο αλλά δεν εμφανίζεται στο ίδιο το μήνυμα. Κάθε τμήμα έχει τις δικές του γραμμές Content-type και Content-Transfer-Encoding.
Το πρωτόκολλο που χρησιμοποιείται για να μεταφέρει μηνύματα από τον ένα υπολογιστή στον άλλο είναι το SMTP (Simple Mail Transfer Protocol). Προτού κληθεί το SMTP οι χρήστες χρησιμοποιούν ένα πρόγραμμα-πελάτη που τρέχει στον υπολογιστή τους (όπως το Outlook express, Eudora κτλ.) για να συντάξουν το κείμενο (επίσης όταν αναζητούν και διαβάζουν το email τους). Υπάρχουν πολλά τέτοια προγράμματα όπως υπάρχουν πολλοί browsers Web. Σήμερα, υπάρχει η δυνατότητα ανάγνωσης email μέσω browser.