ΤΕΙ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση του πρωτοκόλλου LLC (IEEE 802.2) σε διαδικτυακό περιβάλλον

Στέφανος Φ. Μιχόπουλος

Α.Μ. 2008031

Επιβλέπων καθηγητής: Δρ. Ιωάννης Α. Πικραμμένος

ΣΠΑΡΤΗ

ΝΟΕΜΒΡΙΟΣ 2015

« ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάση επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δε μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας. Όνομα και Επώνυμο Συγγραφέα (Με Κεφαλαία):
…… Υπογραφή (Ολογράφως, χωρίς μονογραφή):
…… Ημερομηνία (Ημέρα – Μήνας – Έτος):
…… »

Ευχαριστίες

Η παρούσα πτυχιακή εργασία σηματοδοτεί την ολοκλήρωση των σπουδών μου στο ΤΕΙ Πελοποννήσου, στο Τμήμα Μηχανικών Πληροφορικής Τ.Ε. και θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Ιωάννη Πικραμμένο, για την αμέριστη βοήθεια που μου προσέφερε για την εκπόνησή της.

ΠΕΡΙΕΧΟΜΕΝΑ

---

1 Θεωρία δικτύων

1.1 Δίκτυο υπολογιστών

Επικοινωνία αποκαλείται το μέσο έκφρασης ή μετάδοσης μιας πληροφορίας μεταξύ των ανθρώπων [1]. Για την επικοινωνία, όμως, απαιτείται μια γλώσσα ή ένας κώδικας καθώς και ένα μέσο για να μεταφερθεί το οποιοδήποτε μήνυμα ή η εκάστοτε πληροφορία. Αν δύο συνομιλητές βρίσκονται στον ίδιο χώρο, τότε η επικοινωνία γίνεται μέσω του προφορικού λόγου. Πολλές φορές όμως, η επικοινωνία γίνεται μεταξύ ανθρώπων που βρίσκονται όχι μόνο σε διαφορετικούς χώρους, αλλά και κτίρια, πόλεις ή ακόμα και χώρες. Στις περιπτώσεις αυτές, η επικοινωνία επιτυγχάνεται με χρήση κυρίως του γραπτού λόγου, μέσω ενός μηνύματος κειμένου (SMS), ηλεκτρονικού ταχυδρομείου (E-mail), τηλεομοιότυπου (Fax), τηλεδιάσκεψης, κ.α. για άμεση αποστολή κειμένου και εικόνας μεταξύ μεγάλων αποστάσεων. Οι προαναφερθείσες τεχνολογικές εφευρέσεις επιτρέπουν στους ανθρώπους όλου του κόσμου να επικοινωνούν μεταξύ τους, ανεξαρτήτως απόστασης και για το λόγο αυτό αποκαλούνται τηλεπικοινωνίες [2].

Το δίκτυο υπολογιστών αποτελεί ένα τηλεπικοινωνιακό μοντέλο που επιτρέπει τη μεταφορά δεδομένων μεταξύ δύο ή περισσότερων αυτόνομων ή μη αυτόνομων υπολογιστών και συσκευών που ενώνονται με διαύλους επικοινωνίας [3]. Αυτόνομοι θεωρούνται οι υπολογιστές που δεν είναι εφικτό να ελεγχθεί η λειτουργία του ενός υπολογιστή από κάποιον άλλο, ενώ μη αυτόνομοι είναι οι υπολογιστές οι οποίοι μπορούν να ανταλλάξουν μεταξύ τους πληροφορίες [4].

Τα δίκτυα αναπτύχθηκαν βάσει των αναγκών που προέκυπταν από την ευρεία διάδοση της χρήσης των υπολογιστών. Τα δίκτυα ηλεκτρονικών υπολογιστών μπορούν να χρησιμοποιηθούν για διάφορους σκοπούς, όπως [5]:

- ***Επικοινωνία***: Οι άνθρωποι στην καθημερινότητά τους μπορούν να επικοινωνούν εύκολα και αποτελεσματικά μέσω ενός δικτύου, για παράδειγμα, μέσω ηλεκτρονικού ταχυδρομείου, βίντεο κλήσεων, τηλεφωνικών κλήσεων, τηλεδιάσκεψης, κ.α.
- ***Κοινή χρήση υλικού***: Μέσω ενός δικτύου, κάθε συνδεδεμένος χρήστης μπορεί να έχει πρόσβαση και χρήση του υλικού και των συνδεδεμένων συσκευών στο δίκτυο. Παράδειγμα τέτοιας χρήσης αποτελεί η κοινή χρήση ενός εκτυπωτή από όλους τους υπολογιστές που είναι συνδεδεμένοι σε δίκτυο
- ***Κοινή χρήση λογισμικού***: Οι υπολογιστές που είναι συνδεδεμένοι σε ένα δίκτυο έχουν πρόσβαση σε όλα τα διαθέσιμα προγράμματα εφαρμογής του δικτύου

- ***Διανομή αρχείων, δεδομένων και πληροφοριών:*** Κάθε συνδεδεμένος χρήστης μπορεί να έχει πρόσβαση στις αποθηκευμένες πληροφορίες στο δίκτυο. Η δυνατότητα αυτή αποτελεί ένα βασικό χαρακτηριστικό των περισσότερων δικτύων

1.2 Ιστορική αναδρομή

Το ARPANET (Advanced Research Projects Agency Network) ήταν το πρώτο στον κόσμο πειραματικό δίκτυο μεταγωγής πακέτου, όπου τα προς μετάδοση δεδομένα κόβονται σε πακέτα και πολλοί χρήστες μπορούν να μοιραστούν την ίδια επικοινωνιακή γραμμή, ενώ έδωσε το έναυσμα για τη δημιουργία του παγκόσμιου Διαδικτύου (Internet) στη μορφή που έχει σήμερα [6]. Το πείραμα ξεκίνησε το 1969 εξαιτίας των φόβων των Η.Π.Α για μια πιθανή επίθεση των Ρώσων εναντίον τους. Χρηματοδοτήθηκε από το Γραφείο ερευνών Αμύνης (Defense Advanced Research Projects Agency (DARPA)) του τμήματος άμυνας των Ηνωμένων Πολιτειών, με σκοπό την τεχνολογική ανάπτυξη των στρατιωτικών δυνάμεων και τη δημιουργία ενός δικτύου επικοινωνίας, το οποίο θα μπορούσε να επιβιώσει σε μια ενδεχόμενη πυρηνική επίθεση [7].

Στις αρχές της δεκαετίας του 1970 ξεκινά ένα νέο ερευνητικό πρόγραμμα που ονομάζεται Internetting Project (Πρόγραμμα Διαδικτύωσης), προκειμένου να ενοποιηθούν οι διαφορετικοί τρόποι ανταλλαγής δεδομένων μεταξύ των δικτύων. Στόχος του προγράμματος είναι η διασύνδεση ανόμοιων δικτύων και η ομοιόμορφη διακίνηση δεδομένων μεταξύ αυτών. Αυτό έχει ως αποτέλεσμα τη γέννηση του Internet Protocol (IP), στο οποίο οφείλει το όνομά του το σημερινό Internet. Διαφορετικά δίκτυα που χρησιμοποιούν το κοινό πρωτόκολλο IP μπορούν να συνδέονται και να αποτελούν ένα διαδίκτυο. Επίσης, σχεδιάζεται μια άλλη τεχνική για τον έλεγχο της μετάδοσης των δεδομένων, το Transmission Control Protocol (TCP). Ορίζονται προδιαγραφές για τη μεταφορά αρχείων μεταξύ υπολογιστών (FTP) και για το ηλεκτρονικό ταχυδρομείο (E-mail). Σταδιακά το ARPANET εισχωρεί σε πανεπιστημιακά ιδρύματα άλλων χωρών, με το University College of London (Αγγλία) και το Royal Radar Establishment (Νορβηγία) να είναι τα πρώτα που συνδέονται στο δίκτυο.

Το 1983 το πρωτόκολλο TCP/IP αναγνωρίζεται ως πρότυπο από το αμερικανικό υπουργείο Άμυνας. Παίρνει την ονομασία του από τις συντομογραφίες των πρωτοκόλλων που αποτελούν βασικά συστατικά του, το TCP και το IP. Το σύστημα BSD (Berkeley Software Distribution) το οποίο αναφέρεται και ως Berkeley Unix, καθώς προέρχεται από το Unix, περιλαμβάνει το TCP/IP και συντελεί στη γρήγορη εξάπλωση της δικτύωσης των υπολογιστών. Εκατοντάδες πανεπιστήμια συνδέουν τους υπολογιστές τους στο ARPANET, έχοντας ως αποτέλεσμα την επιβάρυνσή του και τη διάσπασή του σε δύο επιμέρους τμήματα για διαφορετικούς σκοπούς. Το 1983 χωρίζεται σε MILNET (για στρατιωτικές επικοινωνίες) και στο νέο ARPANET (για χρήση αποκλειστικά από την πανεπιστημιακή κοινότητα και συνέχιση της έρευνας στη δικτύωση).

Το 1985 το National Science Foundation (NSF) δημιουργεί ένα δικό του γρήγορο δίκτυο, το NSFNET. Κάνοντας χρήση του πρωτοκόλλου TCP/IP, έχει ως στόχο την σύνδεση πέντε κέντρων υπερυπολογιστών μεταξύ τους και με την υπόλοιπη επιστημονική κοινότητα. Στα τέλη της δεκαετίας, το NSFNET βιώνει μαζική σύνδεση χωρών από όλο τον κόσμο, με τις Δανία, Γαλλία, Φινλανδία, Νορβηγία, Σουηδία, Ολλανδία, Ιαπωνία, Μεξικό να είναι μερικές από αυτές. Χιλιάδες πανεπιστήμια και οργανισμοί δημιουργούν τα δικά τους δίκτυα, τα οποία κατόπιν συνδέουν στο παγκόσμιο δίκτυο, το οποίο αρχίζει να γίνεται γνωστό ως Internet και να εξαπλώνεται με γρήγορους ρυθμούς σε ολόκληρο τον κόσμο. Η ραγδαία αυτή ανάπτυξη έχει ως αποτέλεσμα την κατάργηση του ARPANET.

Η σύνδεση νέων χωρών στο NSFNET συνεχίζεται με γοργούς ρυθμούς, μεταξύ των οποίων και η Ελλάδα το 1990. Το 1993 παρουσιάζεται το World Wide Web (Παγκόσμιος Ιστός - WWW) από το εργαστήριο CERN στην Ελβετία, ανάπτυξης του Tim Berners-Lee. Πρόκειται για ένα σύστημα διασύνδεσης πληροφοριών που βρίσκονται αποθηκευμένες σε εκατομμύρια δικτυωμένους υπολογιστές παγκοσμίως, και παρουσίασής τους μέσω ηλεκτρονικών σελίδων, στις οποίες μπορεί να περιηγηθεί οποιοσδήποτε χρησιμοποιώντας περιφερειακά μέσα. Το γραφικό περιβάλλον κάνει την εξερεύνηση του Internet προσιτή και διασκεδαστική στον απλό χρήστη. Παράλληλα, εμφανίζονται διάφορα εμπορικά δίκτυα που ανήκουν σε εταιρίες παροχής υπηρεσιών Διαδικτύου (Internet Service Providers - ISP) και προσφέρουν πρόσβαση σε όλους. Οποιοσδήποτε διαθέτει Η/Υ και modem μπορεί να συνδεθεί στο Διαδίκτυο μέσω του τηλεφωνικού δικτύου. Η κίνηση του NSFNET φτάνει να ξεπερνά το 10 τρισ. bytes/μήνα, παραδίδοντας το φορτίο του σε εμπορικά δίκτυα και μετατρέποντας το ίδιο ξανά σε ένα ερευνητικό δίκτυο. Το Διαδίκτυο δεν αποτελεί πλέον ένα μέσο επικοινωνίας και ανταλλαγής δεδομένων μεταξύ στρατιωτικών βάσεων και φοιτητών αποκλειστικά, καθώς νέοι χρήστες εισβάλουν σε αυτό καθημερινά, διευκολύνοντας τον τρόπο επικοινωνίας μεταξύ τους. Το μεγαλύτερο μέρος του πλανήτη είναι δικτυωμένο, έχοντας πρόσβαση σε υπηρεσίες ηλεκτρονικού εμπορίου, τηλεργασίας, τηλεκπαίδευσης, τηλεϊατρικής κ.α.

1.3 Πλεονεκτήματα & μειονεκτήματα δικτύων

Η εφαρμογή των δικτύων στην ανθρώπινη καθημερινότητα είναι συχνή. Πλήθος εργασιών καθημερινά μπορούν να διεκπεραιωθούν με τη χρήση κάποιου δικτύου υπολογιστών, όπως για παράδειγμα τραπεζικές συναλλαγές, κρατήσεις ξενοδοχείων, κρατήσεις εισιτηρίων κ.α.. Η χρήση των δικτύων έχει επίσης αλλάξει τον τρόπο με τον οποίο πραγματοποιούνται οι διεργασίες εντός των εταιριών όπως επίσης τον τρόπο με τον οποίο αποθηκεύεται το πλήθος των πληροφοριών που χρησιμοποιούνται στις διεργασίες αυτές. Οι διεργασίες αυτές περιέχουν ακόμα και λειτουργίες οι οποίες παλαιότερα ήταν αδιανόητο να πραγματοποιηθούν με το πάτημα ενός μόνο κουμπιού, γεγονός το οποίο δίνει τη δυνατότητα κέρδους τόσο σε χρόνο όσο και σε χρήμα [8]. Ωστόσο, παρά τη δημιουργία μιας σειράς θετικών πτυχών που απορρέουν από την εφαρμογή της υπολογιστικής δικτύωσης, εξακολουθούν να υπάρχουν ορισμένα

σημαντικά μειονεκτήματα. Το μυστικό της επιτυχημένης δικτύωσης είναι η ικανότητα ζυγοστάθμισης των πλεονεκτημάτων και μειονεκτημάτων αυτών.

Τα κυριότερα πλεονεκτήματα της υπολογιστικής δικτύωσης είναι τα εξής [9]:

- ***Επικοινωνία μεταξύ χρηστών και ανταλλαγή δεδομένων***: Μέσω των δικτύων οι χρήστες μπορούν να επεξεργαστούν όλα τα δεδομένα σε όλες τις φάσεις τους: από την συλλογή και καταχώρηση διαφόρων στοιχείων από όλους τους χρήστες, την επεξεργασία των στοιχείων αυτών από διαφορετικούς υπολογιστές, την αποθήκευσή τους αλλά και την διανομή-πρόσβασή τους από διαφορετικούς υπολογιστές
- ***Διαμοιρασμός προγραμμάτων και δεδομένων του δικτύου***: Τα προγράμματα, τα δεδομένα καθώς και οτιδήποτε είναι αποθηκευμένο στο δίκτυο είναι πάντα διαθέσιμα στους χρήστες που είναι συνδεδεμένοι
- ***Μικρό κόστος υλοποίησης και ευλυγισία χρήσης***: Η εγκατάσταση του κατάλληλου λογισμικού δικτύωσης δεν κοστίζει πάρα πολύ από τη στιγμή μάλιστα που η επιλογή του συνάδει με τον αποτελεσματικό διαμοιρασμό των πληροφοριών σε όλους τους συνδεδεμένους χρήστες. Η αλλαγή του λογισμικού δεν χρειάζεται να γίνεται τακτικά αφού μια απλή ενημέρωση σε καινούργιες εκδόσεις το καθιστούν ικανό να εξυπηρετήσει το δίκτυο για πολλά χρόνια. Η δυνατότητα αυτή ενισχύεται από την ευλυγισία χρήσης του καθώς η οποιαδήποτε αναβάθμιση δεν δημιουργεί προβλήματα στην λειτουργικότητά του

Όπως έχει ήδη όμως αναφερθεί, τα δίκτυα υπολογιστών δεν έχουν εξαλείψει κάποια πολύ σημαντικά προβλήματα, όπως [9]:

- ***Ασφάλεια***: Οι συνδεδεμένοι σε ένα δίκτυο υπολογιστές μπορεί να παραβιαστούν με σκοπό την υποκλοπή χρήσιμων πληροφοριών ή και την πρόκληση κάποιας λειτουργικής βλάβης. Για το λόγο αυτό, κάθε υπολογιστής συνδεδεμένος σε δίκτυο θα πρέπει να προστατεύεται με τη χρήση κωδικών πρόσβασης
- ***Παρουσία ιών και άλλων κακόβουλων λογισμικών***: Ένας ιός είναι εύκολο να μεταδοθεί από έναν υπολογιστή σε άλλον μέσω του δικτύου στο οποίο είναι κοινά συνδεδεμένοι. Αυτό μπορεί να δημιουργεί προβλήματα τόσο στο λογισμικό του δικτύου όσο και στα αποθηκευμένα δεδομένα. Η μοναδική λύση στο πρόβλημα των ιών και του κακόβουλου λογισμικού είναι ο συχνός έλεγχος από το διαχειριστή του δικτύου καθώς και η παρουσία firewall που είναι πιθανόν να αποτρέψουν την εγκατάσταση στο δίκτυο τέτοιων κινδύνων

1.4 Είδη Δικτύων

Τα δίκτυα υπολογιστών μπορούν να ταξινομηθούν σε πολλά είδη, ανάλογα με τον τρόπο που μελετώνται [10]. Παρακάτω, η ταξινόμηση αυτή γίνεται με βάση τη

γεωγραφική τους ανάπτυξη και το μέγεθος, τον τηλεπικοινωνιακό φορέα εξυπηρέτησης και την τεχνική προώθησης της πληροφορίας.

1.4.1 Κατηγοριοποίηση με βάση την γεωγραφική τους ανάπτυξη

Στην κατηγορία αυτή περιλαμβάνονται τα εξής δίκτυα:

1. ***Δίκτυα προσωπικής περιοχής (Personal Area Networks – PAN)***: Τα δίκτυα PAN ουσιαστικά περιλαμβάνουν τα οικιακά δίκτυα υπολογιστών. Γενικά συνδέουν οικιακούς υπολογιστές που μπορούν να διαμοιράζονται εκτυπωτές, σαρωτές, κτλ. Στα δίκτυα PAN χρησιμοποιείται ως επί το πλείστον η τεχνολογία Bluetooth

2. ***Δίκτυα μικρών αποστάσεων ή τοπικά δίκτυα (Local Area Networks - LAN)***: Τα δίκτυα LAN είναι ως επί το πλείστον ιδιωτικά δίκτυα τα οποία βρίσκονται εντός ενός μόνο κτιρίου ή κτιριακού συγκροτήματος. Χρησιμοποιούνται σε επιχειρήσεις ή οργανισμούς καλύπτοντας μικρές αποστάσεις και περιοχές με έκταση έως λίγα χιλιόμετρα. Κυριότερα πλεονεκτήματά τους αποτελούν το χαμηλό κόστος, καθώς επιτρέπεται η χρήση συσκευών και προγραμμάτων από διαφορετικούς χρήστες, η εύκολη και γρήγορη μετάδοση πληροφοριών ανάμεσα στους χρήστες, η επεκτασιμότητα, η συμβατότητα με διάφορες συσκευές κ.α. Στα δίκτυα LAN χρησιμοποιείται ως επί το πλείστον η τεχνολογία Ethernet

3. ***Αστικά Δίκτυα (Metropolitan Area Networks, MAN)***: Τα δίκτυα MAN καλύπτουν αποστάσεις εντός των συνόρων μίας πόλης. Είναι δίκτυα υψηλότερων ταχυτήτων από τα τοπικά δίκτυα και έχουν τη δυνατότητα αποδοτικότερης μετάδοσης εικόνας, φωνής και δεδομένων. Στα δίκτυα MAN συναντάται η χρήση πολλών τεχνολογιών όπως ATM (Asynchronous Transfer Mode), xDSL (οποιασδήποτε μορφής Digital Subscriber Line), ISDN (Integrated Services Digital Netwrok), καθώς επίσης και Ethernet ή οπτικές ίνες

4. ***Δίκτυα ευρείας περιοχής (Wide Area Networks – WAN)***: Τα δίκτυα WAN αποτελούν δίκτυα που καλύπτουν αποστάσεις λίγων χιλιομέτρων (άνω των 5 km) εντός της ίδιας πόλης, και μέχρι χιλιάδων χιλιομέτρων ανάμεσα σε διαφορετικές πόλεις, κράτη και ηπείρους. Τα δίκτυα αυτά περιλαμβάνουν συνδεδεμένους υπολογιστές, συνδεδεμένες συσκευές και γραμμές. Η πιο οικεία περίπτωση δικτύων WAN αποτελεί το διαδίκτυο καθώς συνδέει πλήθος δικτύων παγκοσμίως

1.4.2 Κατηγοριοποίηση με βάση τον τηλεπικοινωνιακό φορέα εξυπηρέτησης

Στην κατηγορία αυτή, τα δίκτυα υπολογιστών διακρίνονται σε ιδιωτικά και δημόσια δίκτυα. Πιο συγκεκριμένα:

1. ***Ιδιωτικά δίκτυα (Private Networks)***: Τα ιδιωτικά δίκτυα χρησιμοποιούνται αποκλειστικά από επιχειρήσεις και οργανισμούς και χρησιμοποιούν ιδιωτικές γραμμές επικοινωνίας ή γραμμές από δημόσιους τηλεπικοινωνιακούς φορείς (leased lines) τις οποίες και χρησιμοποιούν αποκλειστικά

2. Δημόσια δίκτυα (Public Networks): Τα δημόσια δίκτυα καλύπτουν τη σύνδεση απομακρυσμένων σημείων και χρησιμοποιούνται σε περιπτώσεις που η απόσταση καθιστά τη χρήση αποκλειστικών γραμμών αρκετά δαπανηρή καθώς επίσης και όταν ο φόρτος μεταφοράς δεδομένων δεν είναι υψηλός, επιτυγχάνοντας έτσι υψηλότερες ταχύτητες

1.4.3 Κατηγοριοποίηση με βάση την τεχνική προώθησης της πληροφορίας

Με βάση την τεχνική προώθησης της πληροφορίας, τα δίκτυα υπολογιστών μπορούν να διαχωριστούν σε δύο μεγάλες κατηγορίες: τα δίκτυα μεταγωγής και τα δίκτυα ακρόασης. Πιο συγκεκριμένα:

1. Δίκτυα μεταγωγής (Switching Networks): Τα δίκτυα μεταγωγής αποτελούν τα δίκτυα σημείου προς σημείο (point-to-point) που χρησιμοποιούν την τεχνική αποθήκευσης πακέτων μεταγωγής, καθώς επίσης και προώθησης. Σε ένα τέτοιο δίκτυο τα δεδομένα μεταφέρονται μέσω ενδιάμεσων κόμβων στους συνδεδεμένους χρήστες. Οι κόμβοι συνδέονται με τέτοιο τρόπο έτσι ώστε να υπάρχει πάντα εναλλακτικός δρόμος μεταξύ των χρηστών και των συσκευών.

Τα δίκτυα μεταγωγής μπορούν με τη σειρά τους να χωριστούν σε δύο μεγάλες κατηγορίες: τα δίκτυα στα οποία υπάρχει ένας πομπός και ένας δέκτης δεδομένων (unicast) και τα δίκτυα στα οποία υπάρχει ένας πομπός αλλά πολλοί δέκτες δεδομένων (multicast). Στα δίκτυα unicast μπορούν να περιλαμβάνονται εφαρμογές όπως HTTP, FTP, Telnet, VoIP, τηλεδιασκέψεις σημείου προς σημείο, κ.α. ενώ στα δίκτυα multicast αντίστοιχες εφαρμογές όπως audio και video streaming, κ.α.

2. Δίκτυα Ακρόασης (Broadcasting): Στα δίκτυα ακρόασης η πληροφορία εκπέμπεται σε ένα κοινό μέσο και όλοι οι χρήστες έχουν τη δυνατότητα να το ακούσουν. Στα συγκεκριμένα δίκτυα δεν υπάρχουν κόμβοι μεταγωγής. Τα δίκτυα ακρόασης λειτουργούν μέσω ειδικών τεχνικών προσπέλασης των πληροφοριών, όπως για παράδειγμα Reservation, Contention κ.α., και χρησιμοποιούνται από τηλεοπτικά δίκτυα, δορυφορικά δίκτυα κ.λπ.

1.5 Πρωτόκολλα επικοινωνίας

Ένα πρωτόκολλο περιγράφει τους κανόνες που διέπουν ένα δίκτυο και αφορούν τη μετάδοση, την αποθήκευση και τον διαμοιρασμό των δεδομένων μεταξύ κόμβων και χρηστών [5]. Τα πρωτόκολλα που συναντώνται κυρίως είναι τα Token Ring και Ethernet τα οποία και έχουν γίνει αποδεκτά από Διεθνείς Οργανισμούς Τυποποίησης (IEEE).

Η διαδικασία που ακολουθείται για τη μετάδοση δεδομένων σε ένα δίκτυο περιλαμβάνει τον υπολογιστή-αφετηρία, το πρωτόκολλο επικοινωνίας, τον μεταδότη, το καλώδιο μεταφοράς, το δέκτη και τέλος τον υπολογιστή-προορισμό:

- ***Υπολογιστής – αφετηρία***: Μπορεί να χρησιμοποιηθεί κάθε συνδεδεμένος στο δίκτυο υπολογιστής και θεωρείται ως ο υπολογιστής από τον οποίο ξεκινά η μεταφορά των δεδομένων
- ***Πρωτόκολλο επικοινωνίας***: Εμπεριέχει ολοκληρωμένα κυκλώματα και προγράμματα της κάρτας διασύνδεσης
- ***Μεταδότης***: Εκπέμπει ηλεκτρικά σήματα μέσω του καλωδίου μεταφοράς δεδομένων
- ***Δέκτης***: Δέχεται τα σήματα και τα αποκωδικοποιεί
- ***Υπολογιστής – προορισμός***: Μπορεί να χρησιμοποιηθεί κάθε συνδεδεμένος στο δίκτυο υπολογιστής και θεωρείται ως ο υπολογιστής στον οποίο καταλήγουν τα δεδομένα που μεταφέρονται

Σε γενικές γραμμές, ο μηχανισμός πρωτοκόλλου λαμβάνει τα δεδομένα (bits) από τον υπολογιστή – αφετηρία και δημιουργεί τα πλαίσια και τα πεδία δεδομένων, ελέγχου και διεύθυνσης που θα σταλούν. Στη συνέχεια, γίνεται η μετατροπή τους σε ηλεκτρικά σήματα, αποστέλλονται στον δέκτη, ανιχνεύονται τυχόν λάθη μετάδοσης μέσω του μηχανισμού πρωτοκόλλου του δέκτη, επιβεβαιώνεται η ορθή λήψη και τέλος μεταβιβάζονται στον υπολογιστή – προορισμό.

1.6 Λειτουργικό σύστημα δικτύου

Το λειτουργικό σύστημα διαχειρίζεται τους πόρους ενός υπολογιστή, όπως η μνήμη του υπολογιστή, η είσοδος/έξοδος σε περιφερειακές συσκευές, το σύστημα αρχείων, η εκτέλεση προγραμμάτων εφαρμογών στη μνήμη του υπολογιστή καθώς επίσης και ο χρονικός προγραμματισμός της CPU μεταξύ των εφαρμογών.

Το λειτουργικό σύστημα δικτύου (Network Operating System - NOS) αποτελείται από πόρους όπως εκτυπωτές, συσκευές επικοινωνίας κ.α. τους οποίους και διαχειρίζεται σε μεγάλη κλίμακα [11]. Το σύστημα NOS μπορεί να διαχειριστεί πόρους όπως τα συστήματα απομακρυσμένων αρχείων, την εκτέλεση κοινόχρηστων προγραμμάτων και εφαρμογών, τη μνήμη του υπολογιστή όπου το NOS εκτελείται, την είσοδο/έξοδο σε συσκευές δικτύου και τέλος τον χρονικό προγραμματισμό της CPU μεταξύ των διεργασιών.

Τα πιο ευρέως γνωστά λειτουργικά συστήματα δικτύων είναι τα εξής:

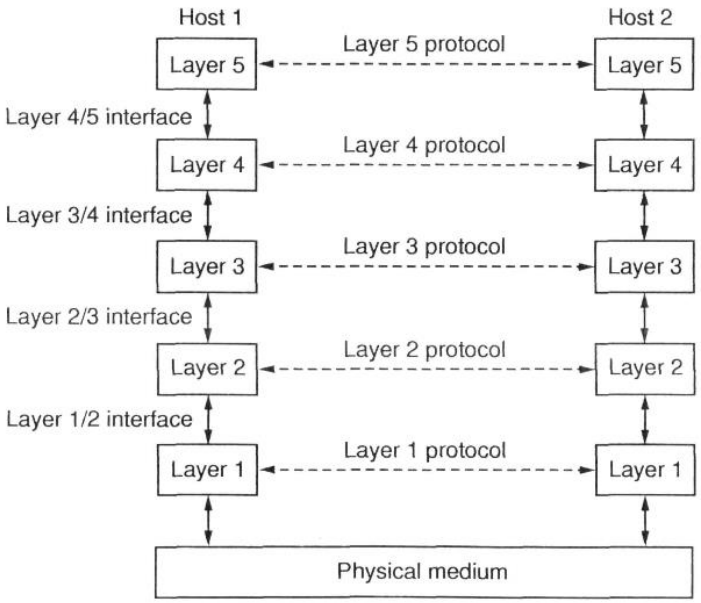
1. SNA της IBM, peer-to-peer, με πρωτόκολλα APPC /APPN
2. UNIX, peer-to-peer, με πρωτόκολλα TCP / IP
3. Novell Netware, dedicated server, με πρωτόκολλα επικοινωνίας IPX /SPX
4. Windows NT και Windows for Workgroups, peer-to-peer, με πρωτόκολλα επικοινωνίας NetBIOS / NetBEUI ή TCP/IP.

1.7 Αρχιτεκτονική Δικτύων

Λόγω των συνεχώς αυξανόμενων απαιτήσεων της τεχνολογίας και των προγραμμάτων, τα δίκτυα μεταβάλλονται και αναπτύσσονται ταχύτατα. Η αρχιτεκτονική δικτύων αναπτύχθηκε με σκοπό να καλύψει το έργο αυτό μέσω γενικών προδιαγραφών προγραμματισμού για τον σχεδιασμό καθώς επίσης και την υλοποίηση των δικτύων [4].

Για να απλουστευθεί η σχεδίαση των δικτύων τα περισσότερα οργανώνονται σε στρώματα (layers) ή επίπεδα (levels). Κάθε δίκτυο αποτελείται από διαφορετικό αριθμό επιπέδων και το όνομα καθώς επίσης και το περιεχόμενό τους διαφέρουν από το ένα δίκτυο στο άλλο. Σκοπός του κάθε επιπέδου είναι να παρέχει συγκεκριμένες υπηρεσίες στα ανώτερα επίπεδα.

Το επίπεδο ενός υπολογιστή ανταλλάσει πληροφορίες με το επίπεδο ενός άλλου χρησιμοποιώντας κανόνες οι οποίοι ονομάζονται πρωτόκολλα στρώματος. Ένα πρωτόκολλο στρώματος, επομένως, αποτελεί τον τρόπο με τον οποίο θα πραγματοποιηθεί η επικοινωνία μεταξύ δύο επιπέδων.



Εικόνα 1: Στρώματα, Πρωτόκολλα και Διεπαφές

Στην εικόνα 1 παρουσιάζονται τα στρώματα (layers), τα πρωτόκολλα (protocols) και οι διεπαφές (interfaces) της γενικής αρχιτεκτονικής ενός δικτύου. Τα στρώματα της γενικής αυτής αρχιτεκτονικής είναι κατά σειρά:

- Στρώμα (layer) 1: Φυσικό Στρώμα (Physical layer)
- Στρώμα (layer) 2: Στρώμα Ζεύξης (Data Link layer)
- Στρώμα (layer) 3: Στρώμα Δικτύου Δεδομένων (Network layer)
- Στρώμα (layer) 4: Στρώμα Μεταφοράς (Transport layer)
- Στρώμα (layer) 5: Στρώμα Εφαρμογής (Application layer)

Ως πρωτόκολλο ορίζεται μία σειρά κανόνων βάσει των οποίων ελέγχεται η μεταφορά των δεδομένων μέσα σε ένα σύστημα επικοινωνίας. Για την μετάδοση των δεδομένων ανάμεσα σε υπολογιστές πραγματοποιείται ο έλεγχος διαθεσιμότητας του υπολογιστή δέκτη, ο συγχρονισμός των υπολογιστών, ο τρόπος τεμαχισμού και συναρμολόγησης της πληροφορίας και τέλος, η σηματοδοσία. Η σειρά αυτή αποτελεί ένα πρωτόκολλο επικοινωνίας.

1.8 Μοντέλα αναφοράς

Η ευρεία χρήση των δικτύων καθώς επίσης και η τεχνολογική εξέλιξη συνετέλεσαν στην ανάπτυξη μοντέλων αναφοράς, δηλαδή στην δημιουργία νέων γενικών αρχών για την σχεδίαση των δικτύων [12]. Τα μοντέλα αναφοράς βασίζονται στα κοινά στοιχεία όσον αφορά τα επίπεδα και τον τρόπο παράδοσης δεδομένων στο ανώτερο επίπεδο.

Στη συνέχεια της ενότητας αυτής αναλύονται δύο από τις βασικότερες αρχιτεκτονικές δικτύων, το μοντέλο αναφοράς OSI και το μοντέλο αναφοράς TCP/IP. Το μοντέλο OSI είναι γενικό και έγκυρο, ενώ περιλαμβάνει σημαντικές λειτουργίες σε κάθε επίπεδο του μοντέλου. Τα συγκεκριμένα πρωτόκολλα δεν χρησιμοποιούνται συχνά πλέον. Αντιθέτως, το μοντέλο TCP/IP δεν είναι ιδιαίτερα χρήσιμο, αλλά τα πρωτόκολλά του χρησιμοποιούνται ευρέως.

1.8.1 Μοντέλο Αναφοράς OSI

Το μοντέλο ISO OSI (ISO OSI Reference Model), όπου OSI σημαίνει Διασύνδεση Ανοικτών Συστημάτων (Open Systems Interconnection), αποτελεί το πρώτο βήμα για τη διεθνή τυποποίηση των πρωτοκόλλων που χρησιμοποιούνται σε διάφορα επίπεδα δικτύων και αφορά συστήματα ανοικτά στην επικοινωνία με άλλα συστήματα. Το μοντέλο OSI αναπτύχθηκε από τον Διεθνή Οργανισμό Τυποποίησης (International Standards Organization ή ISO) [13]. Διαθέτει επτά στρώματα, καθένα από τα οποία εκτελεί συγκεκριμένες λειτουργίες και επικοινωνεί με τα επίπεδα που είναι ακριβώς από πάνω και κάτω του. Τα ανώτερα επίπεδα ασχολούνται κυρίως με τις υπηρεσίες, εφαρμογές και δραστηριότητες χρηστών και τα κατώτερα με την μετάδοση δεδομένων.

Το μοντέλο αναφοράς OSI είναι γνωστό και ως μοντέλο των επτά επιπέδων και αποτελεί μία περιγραφή για τη σχεδίαση τηλεπικοινωνιακών και δικτυακών πρωτοκόλλων (Εικ.2). Ο έλεγχος της μετάδοσης των μηνυμάτων μέσα στο δίκτυο πραγματοποιείται από τα τρία χαμηλότερα επίπεδα, ενώ από τα τέσσερα ανώτερα επίπεδα επιτυγχάνεται η αξιόπιστη μεταβίβαση των δεδομένων μεταξύ των τελικών χρηστών. Έτσι, και τα επτά επίπεδα υλοποιούνται μόνο στους υπολογιστές που λειτουργούν ως τερματικοί σταθμοί (hosts).

Η δημιουργία των επιπέδων βασίστηκε στις ακόλουθες γενικές αρχές [4]:

1. Η δημιουργία ενός επιπέδου πρέπει να πραγματοποιείται εκεί που χρειάζεται διαφορετικός βαθμός αφαίρεσης

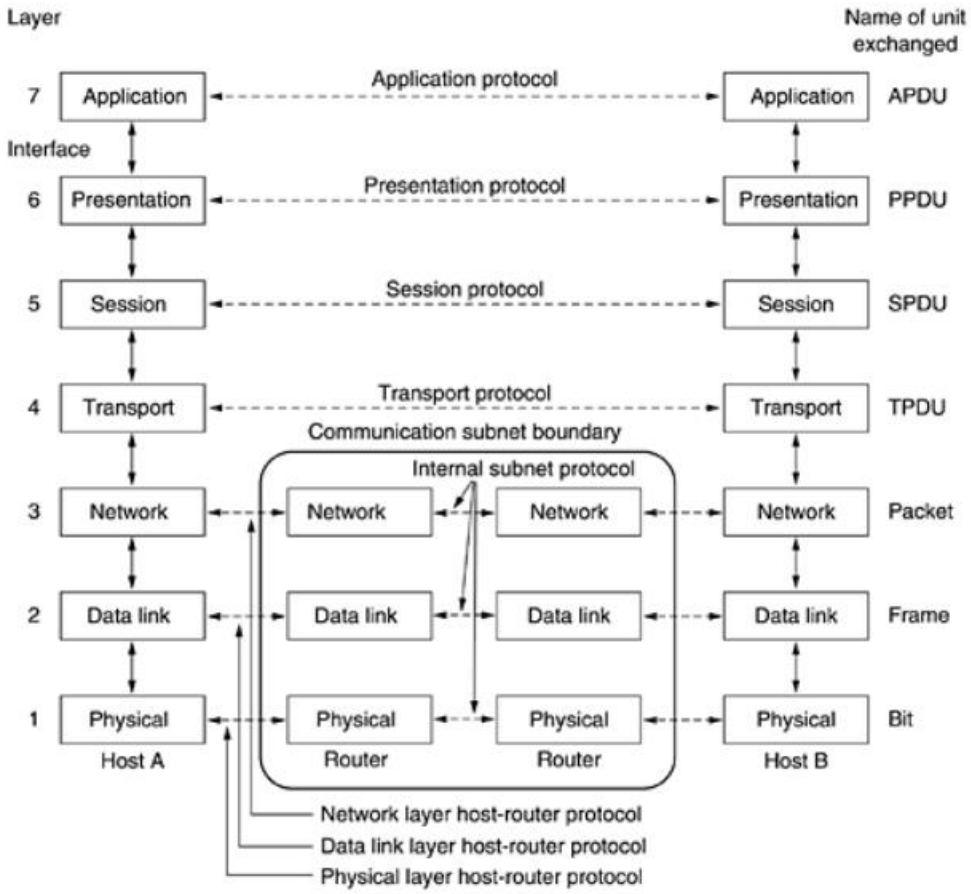
2. Κάθε επίπεδο πρέπει να επιτελεί μία αυστηρά προσδιορισμένη λειτουργία

3. Η λειτουργία κάθε επιπέδου πρέπει να επιλέγεται με βάση τα καθορισμένα διεθνώς (τυποποιημένα) πρωτόκολλα

4. Η επιλογή των ορίων των επιπέδων πρέπει να γίνεται με σκοπό την ελαχιστοποίηση της ροής των πληροφοριών μέσω των διεπαφών

5. Ο αριθμός των επιπέδων πρέπει να είναι αρκετά μεγάλος, ώστε διαφορετικές λειτουργίες να μη χρειάζεται να τοποθετηθούν μαζί στο ίδιο επίπεδο, χωρίς να υπάρχει απόλυτη ανάγκη

6. Ο αριθμός των επιπέδων πρέπει να είναι αρκετά μικρός, ώστε η αρχιτεκτονική των επιπέδων να μη γίνεται πολύπλοκη.



Εικόνα 2: Μοντέλο Αναφοράς OSI

I) Επίπεδο 1: Φυσικό Επίπεδο (Layer 1: Physical layer)

Βασική λειτουργία του φυσικού επιπέδου είναι η μετάδοση ανεπεξέργαστων (raw) δυαδικών ψηφίων (bits) μέσω ενός καναλιού επικοινωνίας. Τα θέματα σχεδίασης ικανοποιούν τη συνθήκη εξασφάλισης της ακεραιότητας των δεδομένων που αποστέλλονται. Αυτό σημαίνει ότι αν ένας host στέλνει το bit 1, αυτό λαμβάνεται από τον host που βρίσκεται στην άλλη μεριά της τηλεπικοινωνιακής γραμμής ως bit 1 και όχι ως bit 0.

II) Επίπεδο 2: Επίπεδο Σύνδεσης Δεδομένων (Layer 2: Data Link layer)

Το επίπεδο σύνδεσης δεδομένων επιτυγχάνει την αξιόπιστη μεταφορά των δεδομένων μέσω των φυσικών μέσων. Τα δεδομένα που εισέρχονται στο επίπεδο αυτό από το υψηλότερο επίπεδο δικτύου οργανώνονται σε πλαίσια (frames), όπου ενσωματώνονται οι πληροφορίες ελέγχου αυτού του επιπέδου, με τη μορφή επικεφαλίδας (overhead) και «ουράς». Οι πληροφορίες που περιέχονται στα πλαίσια, συνήθως χρησιμοποιούνται για τον έλεγχο ροής (flow control) και για τον προσδιορισμό της διεύθυνσης του φυσικού μέσου.

Όταν διαπιστωθεί σφάλμα μεταφοράς κατά τον έλεγχο ενός πλαισίου στον παραλήπτη, τότε είτε ζητείται η αποστολή εκ νέου του λανθασμένου πλαισίου, είτε απλώς ενημερώνεται το ανώτερο επίπεδο μέσω ενός σχετικού μηνύματος ειδοποίησης. Το επίπεδο σύνδεσης δεδομένων ασχολείται επίσης και με τον έλεγχο της ροής δεδομένων μεταξύ δύο κόμβων, ώστε να στέλνονται μόνο όσα δεδομένα μπορεί να δεχτεί ο κόμβος προορισμού.

Τέλος, στο επίπεδο σύνδεσης δεδομένων πραγματοποιείται και η διευθυνσιοδότηση, εάν το φυσικό μέσο μετάδοσης την υποστηρίζει. Έτσι, στην επικεφαλίδα του πλαισίου δεδομένων θα πρέπει να καθορίζεται η «φυσική» διεύθυνση του κόμβου προορισμού του, δηλαδή η διεύθυνση της αντίστοιχης μονάδας προσπέλασης του φυσικού μέσου μετάδοσης πάνω από το οποίο υλοποιείται το δίκτυο.

III) Επίπεδο 3: Επίπεδο Δικτύου (Layer 3: Network layer)

Στο επίπεδο δικτύου προσδιορίζεται ο τρόπος με τον οποίο δρομολογούνται τα πακέτα από τον αποστολέα στον παραλήπτη και ο έλεγχος συμφόρησης του δικτύου. Ως πακέτα ορίζονται οι μονάδες δεδομένων που ανταλλάσσουν οι ομότιμες διεργασίες στο επίπεδο δικτύου, ενώ ως συμφόρηση η κατάσταση του δικτύου όπου η κυκλοφορία είναι μεγαλύτερη από αυτή που μπορεί να εξυπηρετήσει το δίκτυο.

Ο αλγόριθμος δρομολόγησης των πακέτων μπορεί να είναι είτε στατικός, είτε δυναμικός. Οι δυναμικοί αλγόριθμοι δρομολόγησης έχουν ως στόχο την εξάλειψη των περιστατικών συμφόρησης στο δίκτυο. Κάθε κόμβος που ανήκει σε ένα δίκτυο χαρακτηρίζεται μοναδικά από τη διεύθυνση δικτύου. Η δρομολόγηση των πακέτων γίνεται με βάση τη διεύθυνση δικτύου του παραλήπτη κόμβου. Κατά την επιλογή της διαδρομής διοχέτευσης της κυκλοφορίας μιας κλήσης λαμβάνεται υπόψη και ο φόρτος του δικτύου.

Τέλος, όλοι οι κόμβοι που ανήκουν σε ένα δίκτυο χαρακτηρίζονται μοναδικά από τη διεύθυνση δικτύου. Η διεύθυνση δικτύου ορίζεται στο λογισμικό μέρος του. Η δρομολόγηση των πακέτων γίνεται με βάση τη διεύθυνση δικτύου του παραλήπτη.

IV) Επίπεδο 4: Επίπεδο Μεταφοράς (Layer 4: Transport layer)

Το επίπεδο μεταφοράς είναι υπεύθυνο για την εγκαθίδρυση, τη συντήρηση και τον τερματισμό των καναλιών επικοινωνίας μεταξύ των τερματικών κόμβων. Αυτά μπορεί να είναι είτε νοητά κυκλώματα ή να υλοποιούνται με αυτοδύναμα πακέτα.

Στο επίπεδο μεταφοράς δημιουργείται το κανάλι επικοινωνίας ανάμεσα στους τερματικούς κόμβους. Τα μηνύματα που εισέρχονται στον αποστολέα κόμβο από το ανώτερο επίπεδο συνόδου συνήθως διασπώνται σε πακέτα. Στην συνέχεια, αυτά τα πακέτα αριθμούνται και μεταδίδονται στο χαμηλότερο επίπεδο. Αντίστοιχα, ο παραλήπτης συνθέτει και πάλι τα αρχικά μηνύματα από τα εισερχόμενα πακέτα και τα προωθεί προς επεξεργασία στο επίπεδο συνόδου. Το ίδιο κανάλι επικοινωνίας μπορεί να χρησιμοποιηθεί από περισσότερα από ένα μηνύματα, ενώ σε άλλες περιπτώσεις ένα μήνυμα μπορεί να χρησιμοποιήσει περισσότερα από ένα κανάλια επικοινωνίας.

Τέλος, σε αυτό το επίπεδο πραγματοποιείται ο έλεγχος της ροής των δεδομένων μεταξύ των τερματικών κόμβων, ο οποίος είναι ξεχωριστός και ανεξάρτητος από τον έλεγχο ροής που διενεργείται στο επίπεδο σύνδεσης δεδομένων.

V) Επίπεδο 5: Επίπεδο Συνόδου (Layer 5: Session layer)

Πριν από την έναρξη της μετάδοσης δεδομένων θα πρέπει να συμφωνηθεί εάν η επικοινωνία θα είναι αμφίδρομη, εναλλακτικά αμφίδρομη ή μονόδρομη. Στην αμφίδρομη περίπτωση, τα δεδομένα μπορούν να μεταδίδονται και προς τις δύο κατευθύνσεις ταυτόχρονα. Στην εναλλακτικά αμφίδρομη, τα δεδομένα μπορούν να μεταδίδονται και προς τις δύο κατευθύνσεις αλλά όχι ταυτόχρονα. Τέλος, στην μονόδρομη, τα δεδομένα μεταδίδονται μόνο προς μία κατεύθυνση.

Το επίπεδο συνόδου προσφέρει και την υπηρεσία συγχρονισμού, δηλαδή, στην ακολουθία δεδομένων εισάγονται κάποια προσυμφωνημένα σημεία συγχρονισμού πριν από τη μετάδοσή τους.

VI) Επίπεδο 6: Επίπεδο Παρουσίασης (Layer 6: Presentation layer)

Η κύρια λειτουργία στο επίπεδο παρουσίασης είναι η εξασφάλιση της αναγνωσιμότητας των δεδομένων ακόμα και μεταξύ κόμβων που χρησιμοποιούν διαφορετικές μορφές αναπαράστασης της πληροφορίας. Επίσης, στο επίπεδο παρουσίασης συμφωνείται η τεχνική συμπίεσης δεδομένων και κρυπτογράφησης της πληροφορίας που θα ακολουθούν ο αποστολέας και ο παραλήπτης κόμβος.

VII) Επίπεδο 7: Επίπεδο Εφαρμογής (Layer 7: Application layer)

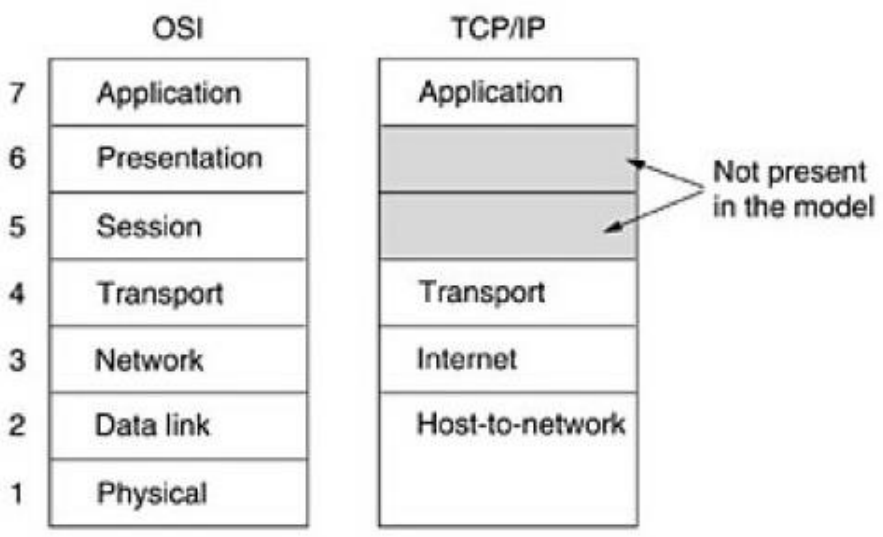
Το επίπεδο εφαρμογής παρέχει ένα σύνολο δικτυακών υπηρεσιών στις τελικές εφαρμογές των χρηστών (όπως, π.χ. το ηλεκτρονικό ταχυδρομείο, η μεταφορά αρχείων κ.ά.). Δε θα πρέπει, όμως, να συγχέεται με την τελική εφαρμογή, δηλαδή, ένα πρόγραμμα μεταφοράς αρχείου είναι μια τελική εφαρμογή χρήστη που βασίζεται στο πρωτόκολλο μεταφοράς αρχείου του επιπέδου εφαρμογής.

1.8.2 Μοντέλο αναφοράς TCP/IP

Το μοντέλο αναφοράς TCP/IP (TCP/IP Reference Model) χρησιμοποιήθηκε αρχικά στον πρόγονο όλων των δικτύων ευρείας περιοχής, το ARPANET, και μετέπειτα στον διάδοχό του, το παγκόσμιο Internet [14]. Έγινε ευρέως γνωστό λόγω της ικανότητάς του να συνδέει πολλά δίκτυα διαφορετικών πρωτοκόλλων με διαφανή τρόπο. Το μοντέλο αναφοράς TCP/IP ορίστηκε αρχικά το 1974 στο έγγραφο των Cerf και Kahn. Μια μεταγενέστερη προοπτική δίνεται το 1985 στο έγγραφο των Leiner και συνεργατών, ενώ η σχεδιαστική του φιλοσοφία παρουσιάζεται στο έγγραφο του Clark το 1988.

Το TCP/IP παρέχει συνδεσιμότητα end-to-end διευκρινίζοντας τον τρόπο με τον οποίο τα δεδομένα θα πρέπει να πακετάρονται, να διευθυσιοδοτούνται, να διαβιβάζονται, να δρομολογούνται και να λαμβάνονται από τον τελικό προορισμό. Αυτή η λειτουργία είναι οργανωμένη σε τέσσερα επίπεδα που χρησιμοποιούνται για την ταξινόμηση όλων των σχετικών πρωτοκόλλων σύμφωνα με το πεδίο εφαρμογής της δικτύωσης που εμπλέκονται. [4] Από το χαμηλότερο στο υψηλότερο, τα στρώματα αυτά είναι τα εξής (Εικ. 3):

- ***Στρώμα ζεύξης (Link layer)***: περιέχει τεχνολογίες επικοινωνίας για ένα μεμονωμένο τμήμα του δικτύου (link)
- ***Στρώμα διαδικτύου (Internet layer)***: συνδέει τους host που μπορεί να ανήκουν ακόμα και σε ανεξάρτητα δίκτυα, δημιουργώντας με αυτό τον τρόπο τη διαδικτύωση (internetworking)
- ***Στρώμα μεταφοράς (Transport layer)***: διαχειρίζεται την επικοινωνία μεταξύ των host
- ***Στρώμα εφαρμογής (Application layer)***: παρέχει δυνατότητα ανταλλαγής δεδομένων εφαρμογής μεταξύ των διεργασιών



Εικόνα 3: Σύγκριση μοντέλων αναφοράς OSI & TCP/IP

I) Επίπεδο ζεύξης (Link layer)

Κάθε host πρέπει να συνδέεται με το δίκτυο, κάνοντας χρήση κάποιου πρωτοκόλλου, ώστε να είναι σε θέση να στείλει πακέτα IP. Το πρωτόκολλο αυτό δεν προσδιορίζεται και παρουσιάζει διαφορές από υπολογιστή σε υπολογιστή και από δίκτυο σε δίκτυο.

II) Επίπεδο διαδικτύου (Internet layer)

Στο επίπεδο διαδικτύου έχουμε ένα δίκτυο μεταγωγής πακέτων που βασίζεται σε ένα ασυνδεσμικό επίπεδο διαδικτύου. Σκοπός του είναι να επιτρέπει στους υπολογιστές την εισαγωγή των πακέτων τους σε οποιοδήποτε δίκτυο κι αυτά να ταξιδεύουν προς τον προορισμό τους. Δεν παρουσιάζεται κανένα πρόβλημα ακόμα και αν τα πακέτα φτάσουν με διαφορετική σειρά από αυτήν της αποστολής, καθώς η αναδιάταξή τους αποτελεί δουλειά των ανώτερων επιπέδων. Τα πακέτα έχουν μια επίσημη μορφή, την οποία το επίπεδο διαδικτύου ορίζει και ονομάζει Πρωτόκολλο Διαδικτύου ή IP (Internet Protocol).

III) Επίπεδο μεταφοράς (Transport layer)

Το επίπεδο μεταφοράς σχεδιάστηκε για να επιτρέπει στις ομότιμες οντότητες στους υπολογιστές υπηρεσίας προέλευσης και προορισμού να επικοινωνούν μεταξύ τους. Αυτό επιτυγχάνεται με την χρήση δύο πρωτοκόλλων, του Πρωτοκόλλου Ελέγχου Μετάδοσης (Transmission Control Protocol - TCP) και του Πρωτοκόλλου Αυτοδύναμων Πακέτων Χρήστη (User Datagram Protocol - UDP). Το TCP είναι ένα αξιόπιστο συνδεσμοστρεφές πρωτόκολλο, που επιτρέπει σε μια ροή byte που προέρχεται από μια μηχανή, την παράδοση χωρίς σφάλματα σε κάποια άλλη μηχανή του διαδικτύου. Το UDP είναι ένα αναξιόπιστο ασυνδεσμικό πρωτόκολλο, που προορίζεται για εφαρμογές που δεν έχουν σαν στόχο την παράδοση των πακέτων σε σωστή σειρά ή τον έλεγχο ροής του TCP, καθώς επιθυμούν την παροχή δικών τους μηχανισμών.

IV) Επίπεδο εφαρμογών (Application layer)

Στο επίπεδο εφαρμογών περιέχονται όλα τα πρωτόκολλα ανώτερου επιπέδου. Αρχικά, περιλαμβάνονταν το εικονικό τερματικό (TELNET), που επιτρέπει την απομακρυσμένη σύνδεση μιας μηχανής με κάποια άλλη, η μεταφορά αρχείων (FTP), όπου παρέχεται αποτελεσματική μεταφορά αρχείων από μια μηχανή σε μια άλλη, και το ηλεκτρονικό ταχυδρομείο, που χρησιμοποιήθηκε σαν ένα είδος μεταφοράς αρχείων, μέχρις ότου αναπτυχθεί ένα εξειδικευμένο πρωτόκολλο (SMTP) ειδικά για αυτό. Αργότερα, έγινε προσθήκη αρκετών ακόμα πρωτοκόλλων, με τα γνωστότερα να είναι το Σύστημα Ονομάτων Περιοχών (DNS), όπου αντιστοιχούνται τα ονόματα των υπολογιστών υπηρεσίας στις διευθύνσεις δικτύου τους, και το HTTP για την προσκόμιση σελίδων στον Παγκόσμιο Ιστό.

1.8.3 Σύγκριση των μοντέλων αναφοράς TCP/IP και OSI

Η σύγκριση ανάμεσα στα δύο μοντέλα αναφοράς OSI και TCP/IP αποφέρει σημαντικές ομοιότητες και διαφορές (Εικ. 3). Οι σημαντικότερες συνοψίζονται στην συνέχεια [4]:

Ι) Ομοιότητες

- Και τα δύο μοντέλα αναφοράς περιγράφονται υπό μορφή επιπέδων.
- Σε κάθε επίπεδο δρουν κάποια πρωτόκολλα, που αναφέρονται και ως πρωτόκολλα του αντίστοιχου επιπέδου.
- Κάθε επίπεδο περιλαμβάνει περισσότερα από ένα πρωτόκολλα. Το ποιο πρωτόκολλο θα χρησιμοποιηθεί εξαρτάται από τις απαιτήσεις των χρηστών και της εφαρμογής που επιλέγουν για να επικοινωνήσουν.
- Τα πρωτόκολλα των υψηλότερων επιπέδων από το επίπεδο μεταφοράς είναι ανεξάρτητα από το δίκτυο που χρησιμοποιείται για να επιτευχθεί η επικοινωνία.

ΙΙ) Διαφορές

- Η περιγραφή του OSI θεωρείται πληρέστερη από αυτήν του TCP/IP. Το μοντέλο OSI κάνει ένα σαφή διαχωρισμό ανάμεσα στις έννοιες της υπηρεσίας, της διεπαφής και του πρωτοκόλλου ενώ στο TCP/IP ο διαχωρισμός αυτός δεν είναι ευδιάκριτος.
- Το μοντέλο OSI πρώτα περιγράφηκε από τους ειδικούς και μετά γράφτηκαν τα πρωτόκολλα που αφορούσαν τη λειτουργία των υπηρεσιών που προσφέρονται. Αντίθετα, στην περίπτωση του TCP/IP πρώτα δημιουργήθηκαν τα πρωτόκολλα και μετά, με βάση τα υπάρχοντα πρωτόκολλα, δημιουργήθηκε το μοντέλο.
- Στην περίπτωση του TCP/IP δεν υπήρξε πρόβλημα συμφωνίας πρωτοκόλλων – μοντέλου.
- Στην περίπτωση του OSI η ανυπαρξία κάποιων έτοιμων πρωτοκόλλων τα οποία θα προσδιόριζαν με ακρίβεια τον ορισμό του μοντέλου είχε ως αποτέλεσμα το μοντέλο που δημιουργήθηκε να είναι αρκετά γενικό.
- Μια σημαντική διαφορά μεταξύ των δύο μοντέλων είναι αυτή του αριθμού των επιπέδων τους. Στο OSI έχουμε επτά, ενώ στο TCP/IP τέσσερα.
- Το επίπεδο συνόδου του OSI έχει στην πραγματικότητα πολύ μικρή εφαρμογή, ενώ το επίπεδο παρουσίασης απουσιάζει εντελώς από τις περισσότερες εφαρμογές.
- Το TCP/IP δεν κάνει σαφή διαχωρισμό μεταξύ του φυσικού επιπέδου και του επιπέδου γραμμής δεδομένων.
- Το TCP/IP έτυχε ευρύτερης αποδοχής από τον κόσμο των επικοινωνιών σε σύγκριση με το OSI. Οι αιτίες αυτής της αποδοχής είναι οι ακόλουθες:

- Το OSI προσπάθησε να αναπτυχθεί, όταν το TCP/IP χρησιμοποιούνταν ήδη από όλο σχεδόν το φάσμα του ακαδημαϊκού χώρου.
- Οι πρώτες εφαρμογές που γράφτηκαν στα πλαίσια του μοντέλου OSI ήταν πολύπλοκες και δύσχρηστες. Αντίθετα, τα πρωτόκολλα του TCP/IP ήταν καλογραμμένα και εύχρηστα.
- Οι εφαρμογές, οι βασισμένες στο μοντέλο TCP/IP, ήταν και είναι δωρεάν διαθέσιμες στους χρήστες υπολογιστών.
- Το OSI αποτελεί χρήσιμο εκπαιδευτικό εργαλείο για την εξερεύνηση και τη μελέτη των δικτύων ηλεκτρονικών υπολογιστών.

1.9 Σύνοψη

Η κατανόηση της βασικής θεωρίας των δικτύων ηλεκτρονικών υπολογιστών είναι σημαντική καθώς παρέχει όλα τα θεμελιώδη στοιχεία της πρακτικής υλοποίησης ενός διαδικτυακού περιβάλλοντος. Μια τέτοια υλοποίηση υποβοηθάται από την κατάτμηση των διαδικτυακών επικοινωνιών σε στρώματα, τα οποία είναι εφτά με βάση το μοντέλο αναφοράς OSI ή τέσσερα με βάση το μοντέλο αναφοράς TCP/IP. Τα δύο μοντέλα αναφοράς περιλαμβάνουν επίπεδα δικτύου, μεταφοράς και εφαρμογής, αλλά διαφοροποιούνται όσον αφορά τα υπόλοιπα στρώματα.

Τα δίκτυα υπολογιστών μπορούν να εφαρμοστούν για την υλοποίηση αναρίθμητων υπηρεσιών και για σκοπούς που εκτός από την επικοινωνία μεταξύ των υπολογιστών περιλαμβάνουν την κοινή χρήση υλικών και λογισμικού καθώς επίσης και το διαμοιρασμό αρχείων, δεδομένων και πληροφοριών.

Γενικά μιλώντας, τα δίκτυα υπολογιστών μπορούν να ταξινομηθούν σε πολλές κατηγορίες και με βάση πολλά κριτήρια. Όσον αφορά την εμβέλεια κάλυψης, τα δίκτυα μπορούν να χωριστούν σε PAN, LAN, WAN και MAN. Τα PAN καλύπτουν τα προσωπικά οικιακά δίκτυα. Τα LAN καλύπτουν την περιοχή ενός ολόκληρου κτιρίου και λειτουργούν σε υψηλότερες ταχύτητες. Τα MAN μπορούν να καλύψουν την περιοχή μιας ολόκληρης πόλης και χρησιμοποιούνται πλέον ως βάση πρόσβασης του διαδικτύου. Τα MAN καλύπτουν μια χώρα ή μια ήπειρο. Τα δίκτυα LAN και MAN δεν περιλαμβάνουν δρομολογητές για τη μεταφορά των δεδομένων, σε αντίθεση με τα WAN στα οποία η ύπαρξη δρομολογητών είναι απαραίτητη.

Το λογισμικό των δικτύων αποτελείται από πρωτόκολλα, τα οποία είναι κανόνες μέσω των οποίων επιτυγχάνεται η επικοινωνία μεταξύ των διαφόρων διεργασιών. Τα περισσότερα δίκτυα υποστηρίζουν μια ιεραρχία των πρωτοκόλλων, η οποία επιτρέπει σε κάθε στρώμα του μοντέλου αναφοράς να παρέχει υπηρεσίες προς τα ανωτέρω στρώματα, τις λεπτομέρειες των οποίων θα πρέπει να απομονώνει από τα στρώματα που βρίσκονται κάτω από αυτό.

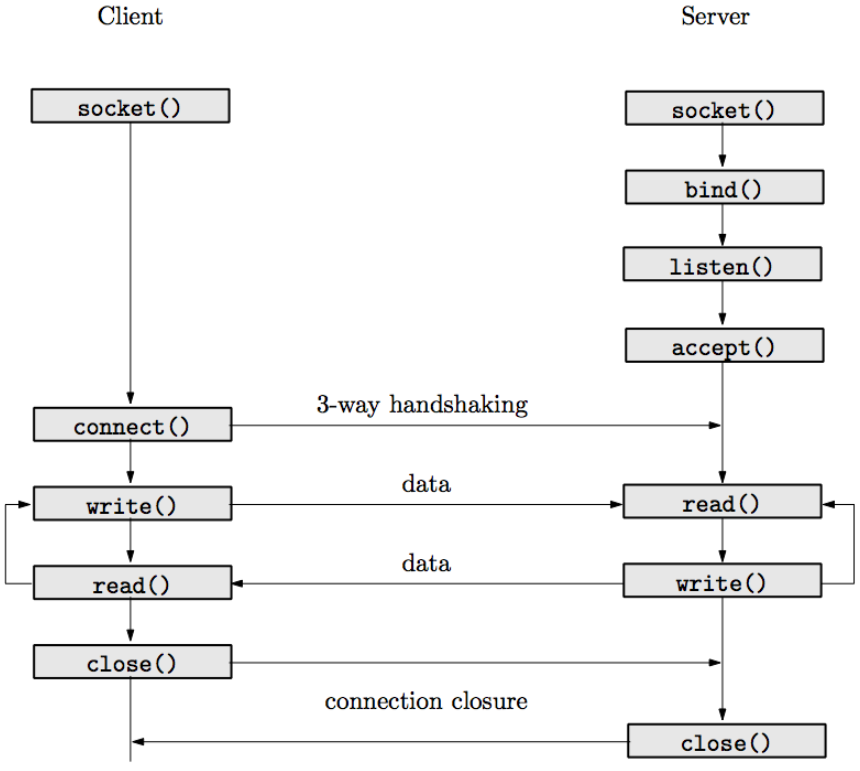
Τέλος, τα δίκτυα υπολογιστών ακολουθούν το μοντέλο πελάτη – διακομιστή (client / server model) για το οποίο θα γίνει αναλυτική αναφορά στο επόμενο κεφάλαιο.

2 Το μοντέλο πελάτη/διακομιστή

2.1 Αλληλεπίδραση μεταξύ πελάτη και διακομιστή

Σε ένα διαδικτυακό περιβάλλον ιδιαίτερο ενδιαφέρον παρουσιάζει ο τρόπος με τον οποίο γίνεται η ανταλλαγή πληροφοριών μεταξύ των εφαρμογών. Η πιο δημοφιλής μορφή αυτού του είδους της ανταλλαγής είναι η αλληλεπίδραση μεταξύ πελάτη και διακομιστή (client/server interaction). Το μοντέλο αυτό αποτελεί τη βάση των περισσότερων διαδικτυακών αρχιτεκτονικών και εφαρμογών [15].

Όπως φανερώνει και η ίδια η ονομασία του μοντέλου, ο διακομιστής (server) αναφέρεται ως μια οντότητα που παρέχει μια υπηρεσία για την οποία έχει αιτηθεί μια εφαρμογή ή ένας πελάτης (client). Οι διακομιστές βασικά διαχειρίζονται αιτήματα των πελατών, υλοποιούν τις υπηρεσίες και επιστρέφουν μια απάντηση υπό μορφή πακέτων. Την ίδια μορφή έχει και η αίτηση που αποστέλλεται από τον πελάτη προς τον διακομιστή. Ένα τυπικό παράδειγμα μιας τέτοιας αλληλεπίδρασης μεταξύ πελάτη και διακομιστή αποτελούν οι διακομιστές ιστού (web servers) που ανακτούν ιστοσελίδες σύμφωνα με τα αιτήματα των πελατών από τον Παγκόσμιο Ιστό (World Wide Web – WWW).



Εικόνα 4: Διάγραμμα αλληλεπίδρασης μεταξύ πελάτη και διακομιστή

Οι διακομιστές μπορούν επίσης να αποτελούν προγράμματα εφαρμογής, τα οποία εκτελούνται σε διαφορετικούς host από τις αντίστοιχες client εφαρμογές [16]. Στην περίπτωση αυτή, ο διακομιστής ξεκινάει πρώτος και περιμένει, ενώ ο πελάτης ξεκινάει δεύτερος και ενεργοποιεί τη σύνδεση με τον διακομιστή. Στο σημείο αυτό,

θα πρέπει να σημειωθεί ότι ο διακομιστής ουσιαστικά δεν ενδιαφέρεται με ποιον πελάτη θα επικοινωνήσει, ενώ αντίθετα ο πελάτης θα πρέπει να γνωρίζει με ποιον διακομιστή πρέπει να έρθει σε επαφή. Η επικοινωνία ξεκινάει με αίτημα που αποστέλλεται από τον πελάτη προς τον διακομιστή. Στη συνέχεια, η επικοινωνία πελάτη/διακομιστή διαρκεί μέχρι το πέρας της υπηρεσίας που αιτήθηκε ο πελάτης. Καθ' όλη τη διάρκεια της επικοινωνίας αυτής πελάτης και διακομιστής ανταλλάσσουν δεδομένα υπό μορφή πακέτων. Μετά το πέρας της επικοινωνίας, ο διακομιστής παραμένει ενεργός περιμένοντας νέο αίτημα για υπηρεσία, ενώ ο πελάτης μπορεί να ολοκληρώσει την επικοινωνία με τον διακομιστή ή να την συνεχίσει αποστέλλοντας νέο αίτημα.

Όπως παρουσιάζεται και στην εικόνα 4, η επικοινωνία μεταξύ πελάτη και διακομιστή στις διαδικτυακές εφαρμογές γίνεται μέσω υποδοχών (sockets), οι οποίες θα αναλυθούν σε επόμενη ενότητα.

2.2 Χαρακτηριστικά πελατών και διακομιστών

Αν και υπάρχουν αποκλίσεις, στις περισσότερες των περιπτώσεων, οι αλληλεπιδράσεις μεταξύ πελάτη και διακομιστή έχουν τα ίδια χαρακτηριστικά [16]. Γενικότερα τα χαρακτηριστικά του λογισμικού ενός πελάτη είναι τα εξής:

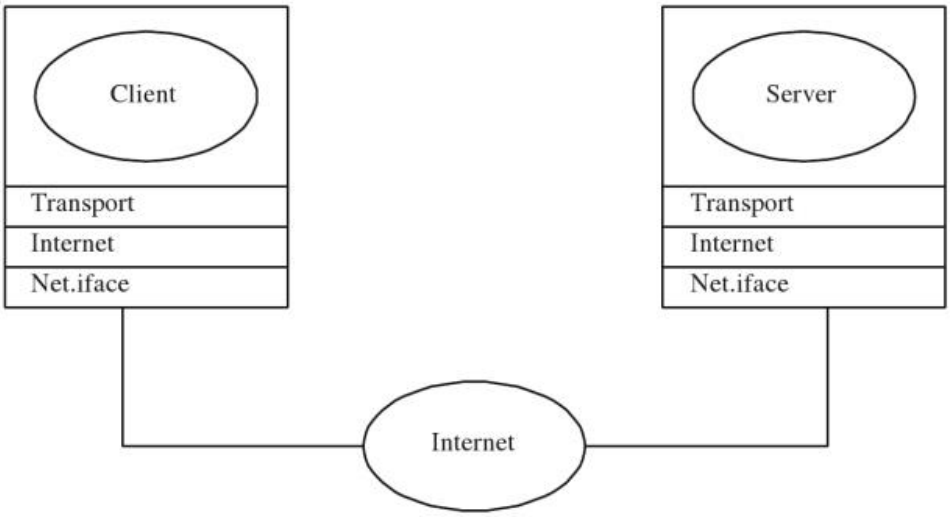
- Πρόκειται για οποιοδήποτε πρόγραμμα εφαρμογής, το οποίο γίνεται προσωρινά πελάτης στην περίπτωση που απαιτείται πρόσβαση σε κάποιο απομακρυσμένο πρόγραμμα, αλλά τοπικά πραγματοποιεί και άλλες διεργασίες.
- Καλείται άμεσα από κάποιο χρήστη και εκτελείται μόνο για μια σύνοδο.
- Εκτελείται τοπικά στον προσωπικό υπολογιστή του χρήστη.
- Ξεκινάει ενεργητικά την επαφή με ένα διακομιστή.
- Ανάλογα τις ανάγκες του, έχει τη δυνατότητα προσπέλασης πολλών υπηρεσιών, αλλά συνήθως έρχεται σε επαφή με μόνο έναν απομακρυσμένο διακομιστή την κάθε φορά.
- Δεν χρειάζεται κάποιο ειδικό υλικό, αλλά ούτε κάποιο προηγμένο λειτουργικό σύστημα.

Αντίθετα τα χαρακτηριστικά του λογισμικού ενός διακομιστή είναι τα εξής:

- Πρόκειται για ένα πρόγραμμα ειδικής χρήσης, που δραστηριοποιείται για την παροχή μίας μόνο υπηρεσίας, αλλά έχει τη δυνατότητα ταυτόχρονης εξυπηρέτησης πολλών απομακρυσμένων πελατών.
- Καλείται αυτόματα κατά την αρχική εκκίνηση ενός συστήματος και συνεχίζει να εκτελείται για πολλές συνόδους.
- Εκτελείται σε ένα μεριζόμενο υπολογιστή.
- Αναμένει παθητικά, μέχρι την πραγματοποίηση κάποιας επαφής από οποιουσδήποτε απομακρυσμένους πελάτες.
- Αποδέχεται την πραγματοποίηση επαφής από τους πελάτες, παρέχοντας μια μόνο υπηρεσία.

- Απαιτεί ισχυρό υλικό και προηγμένο λειτουργικό σύστημα.

Συνήθως, στα προγράμματα εφαρμογών χρησιμοποιείται ένα πρωτόκολλο μεταφοράς για την επικοινωνία μεταξύ του πελάτη και του διακομιστή [17]. Όπως μπορεί να φανεί και στην εικόνα 5, η επικοινωνία μεταξύ πελάτη και διακομιστή μπορεί να πραγματοποιηθεί με χρήση της στοίβας πρωτοκόλλων TCP/IP.



Εικόνα 5: Επικοινωνία πελάτη/διακομιστή με χρήση πρωτοκόλλων TCP/IP σε διαδικτυακό περιβάλλον

Η επικοινωνία και η αποστολή ή η αποδοχή πληροφοριών μεταξύ πελάτη ή διακομιστή γίνεται απευθείας με πρωτόκολλο του επιπέδου μεταφοράς. Με τον τρόπο αυτό, ένας υπολογιστής χρειάζεται να έχει μια πλήρη στοίβα πρωτοκόλλων για να λειτουργήσει είτε ως πελάτης είτε ως διακομιστής, καθώς χρησιμοποιεί πρωτόκολλα χαμηλότερων επιπέδων, τα οποία αποστέλλουν ή λαμβάνουν μεμονωμένα μηνύματα.

Τα πρωτόκολλα μεταφοράς υποστηρίζουν δύο βασικές μορφές επικοινωνίας, τη συνδεσμική (connection oriented communication) και την ασυνδεσμική (connectionless communication). Το πρωτόκολλο TCP παρέχει στις εφαρμογές συνδεσμική διασύνδεση, π.χ. όταν ζητηθεί η χρησιμοποίηση του TCP από μια εφαρμογή, πρέπει πρώτα το TCP να ανοίξει μια σύνδεση με μια άλλη εφαρμογή και στη συνέχεια να γίνει ανταλλαγή δεδομένων. Η σύνδεση πρέπει να κλείσει αφού ολοκληρωθεί η επικοινωνία των εφαρμογών. Με μια ασυνδεσμική διασύνδεση επιτρέπεται σε μια εφαρμογή η αποστολή ενός μηνύματος σε οποιοδήποτε προορισμό και σε οποιαδήποτε στιγμή. Για παράδειγμα, μια εφαρμογή που χρησιμοποιεί το πρωτόκολλο UDP, μπορεί να στείλει μηνύματα και το κάθε ένα να πηγαίνει σε διαφορετικό προορισμό. Οι πελάτες και οι διακομιστές μπορούν να χρησιμοποιούν συνδεσμικά ή ασυνδεσμικά πρωτόκολλα μεταφοράς για να επικοινωνούν.

2.3 Τοπολογίες μοντέλου πελάτη/διακομιστή

Ένας πελάτης ή διακομιστής μπορεί να αποτελείται από ένα πρόγραμμα εφαρμογής, ενώ ένας υπολογιστής μπορεί να εκτελεί πολλές εφαρμογές ταυτόχρονα

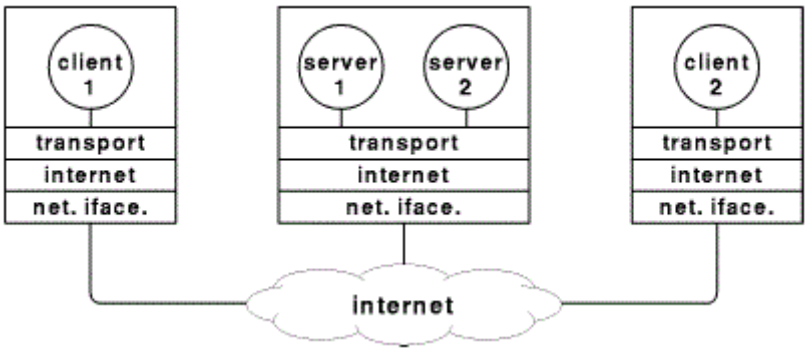
[16]. Ο συνδυασμός αυτών των δύο δεδομένων οδηγεί στο συμπέρασμα ότι ένας υπολογιστής μπορεί να εκτελεί:

- ένα μόνο διακομιστή
- ένα μόνο πελάτη
- πολλά αντίγραφα πελάτη που έρχονται σε επαφή με δεδομένο διακομιστή
- πολλούς πελάτες που ο καθένας τους έρχεται σε επαφή με συγκεκριμένο διακομιστή
- πολλούς διακομιστές, καθένας από τους οποίους παρέχει μια συγκεκριμένη υπηρεσία

Με βάση αυτό το συμπέρασμα προκύπτουν τρεις διαφορετικές τοπολογίες πελάτη/διακομιστή [18]:

- Ενός πελάτη – ενός διακομιστή
- Πολλών πελατών – ενός διακομιστή
- Πολλών πελατών - πολλών διακομιστών

Από τις τρεις αυτές τοπολογίες, αυτή που παρουσιάζει μεγαλύτερο ενδιαφέρον είναι η τοπολογία πολλών πελατών – πολλών διακομιστών, λόγω των πολλών πλεονεκτημάτων που παρουσιάζει (Εικ. 6). Αρχικά, για να είναι εφικτή η υλοποίηση αυτής της τοπολογίας, ο εκάστοτε υπολογιστής θα πρέπει να έχει επαρκείς πόρους υλικού και ένα τέτοιο λειτουργικό σύστημα που να επιτρέπει την ταυτόχρονη εκτέλεση πολλών προγραμμάτων εφαρμογών.



Εικόνα 6: Παράδειγμα υλοποίησης τοπολογίας δύο πελατών – δύο διακομιστών σε διαδικτυακό περιβάλλον

Η δυνατότητα να εξυπηρετεί ο ίδιος υπολογιστής πολλούς πελάτες είναι σημαντική, επειδή με τον τρόπο αυτό πολλές υπηρεσίες μπορούν να παρέχονται ταυτόχρονα. Κάθε εφαρμογή είναι ένας πελάτης που έρχεται σε επαφή με ένα συγκεκριμένο διακομιστή ανεξάρτητα. Στην πραγματικότητα, η τεχνολογία αυτή επιτρέπει σε κάθε χρήστη να έχει ανοιχτά τουλάχιστον δύο αντίγραφα μιας εφαρμογής, καθένα από τα οποία έρχεται σε επαφή με διαφορετικό ή τον ίδιο διακομιστή (για παράδειγμα δύο διαφορετικές σελίδες σε έναν περιηγητή ιστού).

Η δυνατότητα να εξυπηρετεί ο ίδιος υπολογιστής πολλούς διακομιστές είναι σημαντική, επειδή το υλικό μπορεί να διαμοιραστεί σε πολλές υπηρεσίες. Επίσης, η

συγκέντρωση διακομιστών σε ένα μεγάλο υπολογιστή βοηθά στη μείωση της επιβάρυνσης του συστήματος. Τέλος, η εκτέλεση πολλών διακομιστών σε έναν υπολογιστή ωφελεί, γιατί ένας διακομιστής δεν καταναλώνει υπολογιστικούς πόρους όταν περιμένει μια αίτηση.

2.4 Αναγνώριση διακομιστή

Τα διαδικτυακά πρωτόκολλα δίνουν τη δυνατότητα στους πελάτες να ξεχωρίζουν την αναγνώριση ενός διακομιστή σε δύο μέρη [16], [19]:

- την αναγνώριση του υπολογιστή στον οποίο εκτελείται ένας διακομιστής
- την αναγνώριση μιας συγκεκριμένης υπηρεσίας στον υπολογιστή

Ι) Αναγνώριση υπολογιστή

Σε κάθε υπολογιστή που συνδέεται στο διαδίκτυο αποδίδεται ένα μοναδικό αναγνωριστικό των 32 bit, που είναι γνωστό ως διεύθυνση IP (Internet Protocol address). Για να μπορέσει ένας πελάτης να έρθει σε επαφή με έναν διακομιστή θα πρέπει να καθορίσει τη διεύθυνση IP του συγκεκριμένου διακομιστή.

Η αναγνώριση των διακομιστών από τους ανθρώπους γίνεται μέσω του ονόματος που αποδίδεται σε κάθε υπολογιστή, ένα όνομα που μέσω του Συστήματος Ονομάτων Τομέων (Domain Name System – DNS) μεταφράζεται σε διεύθυνση. Με τον τρόπο αυτό ένας χρήστης βρίσκει έναν διακομιστή μέσω του ονόματός του (π.χ. www.cisco.com) και όχι μέσω κάποιας διεύθυνσης ακέραιου αριθμού.

ΙΙ) Αναγνώριση υπηρεσίας

Σε κάθε υπηρεσία που είναι διαθέσιμη στο διαδίκτυο αποδίδεται ένα μοναδικό αναγνωριστικό των 16 bit, γνωστό ως αριθμό θύρας (port number). Για παράδειγμα, στην υπηρεσία e-mail αποδίδεται το port number 25, ενώ στον ιστό αποδίδεται το port number 80.

Αυτό το αναγνωριστικό χρησιμοποιείται τόσο από τους πελάτες όσο και από τους διακομιστές. Όταν ένας διακομιστής αρχίζει να εκτελείται, καταχωρείται στο τοπικό σύστημα καθορίζοντας τον αριθμό θύρας της υπηρεσίας που παρέχει. Όταν ένας πελάτης έρχεται σε επαφή με έναν απομακρυσμένο διακομιστή για να αιτηθεί κάποια υπηρεσία, η αίτησή του περιέχει τον αριθμό θύρας της υπηρεσίας που επιθυμεί. Έτσι, όταν η αίτηση αυτή φτάσει στο διακομιστή, το λογισμικό του διακομιστή χρησιμοποιεί το αναγνωριστικό αυτό για να καθορίσει το ποια εφαρμογή του υπολογιστή θα πρέπει να διαχειριστεί το αίτημα αυτό. Ένα παράδειγμα αποτελεί το πρωτόκολλο TCP, το οποίο χρησιμοποιεί τον αριθμό θύρας πρωτοκόλλου που εμπεριέχεται σε ένα εισερχόμενο μήνυμα, για να ορίσει το διακομιστή που θα λάβει την αίτηση.

2.5 Ταυτόχρονοι διακομιστές

Η εξυπηρέτηση ενός πελάτη τη φορά από ένα διακομιστή αποτελεί μια σειριακή προσέγγιση του μοντέλου πελάτη/διακομιστή και χρησιμοποιείται σε τετριμμένες περιπτώσεις [20]. Σε ένα διαδικτυακό περιβάλλον οι περισσότεροι διακομιστές είναι ταυτόχρονοι (concurrent servers), κάτι που σημαίνει ότι για να διαχειριστούν ταυτόχρονα πολλούς πελάτες χρησιμοποιούν περισσότερα από ένα νήματα ελέγχου (threads of control).

Η ταυτόχρονη εξυπηρέτηση είναι σημαντική, καθώς αποτρέπει τα φαινόμενα αναμονής των πελατών που θα εμφανίζονταν αν πολλοί πελάτες έστελναν αιτήματα σε κάποιον διακομιστή και αυτός θα έπρεπε να τους εξυπηρετήσει σειριακά, δηλαδή έναν τη φορά. Ο ταυτόχρονος διακομιστής αρχείων μπορεί να εξυπηρετήσει πολλούς πελάτες την ίδια στιγμή, αναθέτοντας μια πιθανή αίτηση σε ένα νήμα ελέγχου, το οποίο μπορεί να εκτελείται ταυτόχρονα με άλλα υπάρχοντα νήματα.

Οι λεπτομέρειες της ταυτόχρονης εξυπηρέτησης εξαρτώνται από το λειτουργικό σύστημα που χρησιμοποιείται, αλλά η γενική ιδέα είναι ότι ο κώδικας ενός ταυτόχρονου διακομιστή χωρίζεται σε δύο μέρη: το κύριο πρόγραμμα (νήμα) και το μέρος εξυπηρέτησης (handler). Η μόνη λειτουργία του κυρίου νήματος είναι η αποδοχή της επαφής με τον πελάτη και η δημιουργία ενός νήματος ελέγχου. Κάθε νήμα ελέγχου αλληλεπιδρά με ένα μόνο πελάτη και εκτελεί τον κώδικα εξυπηρέτησης. Μετά το χειρισμό του πελάτη, το νήμα τερματίζεται. Μετά τη δημιουργία ενός νήματος για το χειρισμό ενός αιτήματος, το κύριο νήμα περιμένει την άφιξη νέου αιτήματος.

2.6 Διαδικτυακός προγραμματισμός και η υποδοχή API

Η διασύνδεση μεταξύ ενός προγράμματος εφαρμογής και των πρωτοκόλλων επικοινωνίας ενός λειτουργικού συστήματος είναι γνωστή ως Διεπαφή Προγραμματισμού Εφαρμογών (Application Program Interface - API) [16]. Αν και οι ακριβείς λεπτομέρειες μιας API εξαρτώνται από το λειτουργικό σύστημα, η API αποτελεί ένα ντε φάκτο πρότυπο των επικοινωνιών μεταξύ των λογισμικών μέσω του διαδικτύου. Γνωστή και ως υποδοχή API, ή απλά υποδοχή (socket), η API είναι διαθέσιμη σε πολλά λειτουργικά συστήματα. Κάποια εξ αυτών χρησιμοποιούνται σε προσωπικούς υπολογιστές (π.χ. Windows) και άλλα σε συστήματα UNIX (π.χ. Solaris), συμπεριλαμβανομένου και του LINUX.

Η API καθορίζει τις διαθέσιμες λειτουργικές δυνατότητες επειδή ορίζει ένα σύνολο λειτουργιών που μπορεί μια εφαρμογή να πραγματοποιεί όταν αλληλεπιδρά με το λογισμικό πρωτοκόλλων [21]. Μια API μπορεί να περιέχει μια διαδικασία που χρησιμοποιείται για επικοινωνία καθώς και ακόμα μια διαδικασία για αποστολή δεδομένων. Αυτό συνεπάγεται στο ότι η API περιέχει μια ξεχωριστή διαδικασία για κάθε λογική πράξη.

2.7 Υποδοχές, περιγραφείς και είσοδος/έξοδος δικτύου

Λόγω του γεγονότος ότι αρχικά αναπτύχθηκε ως μέρος του λειτουργικού συστήματος UNIX, η υποδοχή API περιέχει ενσωματωμένες εισόδους και εξόδους (I/O) [15]. Συγκεκριμένα, όταν μια εφαρμογή δημιουργεί μια υποδοχή για να την χρησιμοποιήσει για επικοινωνία στο Διαδίκτυο, το λειτουργικό σύστημα επιστρέφει ένα μικρό ακέραιο περιγραφέα (descriptor) που προσδιορίζει την υποδοχή. Η εφαρμογή περνά στη συνέχεια τον περιγραφέα ως όρισμα (argument) όταν καλεί συναρτήσεις για να εκτελέσει μια λειτουργία στην υποδοχή (π.χ., για τη μεταφορά δεδομένων μέσω του δικτύου ή για τη λήψη των εισερχομένων δεδομένων).

Σε πολλά λειτουργικά συστήματα, οι περιγραφείς των υποδοχών είναι ενσωματωμένοι με άλλους περιγραφείς I/O [16]. Αν ένα σύστημα χρησιμοποιεί τον ίδιο χώρο τιμών περιγραφέων για τις υποδοχές και για τις άλλες λειτουργίες I/O, η ίδια εφαρμογή μπορεί να χρησιμοποιείται για δικτυακή επικοινωνία αλλά και για τοπικές μεταφορές δεδομένων. Το κύριο πλεονέκτημα ενός τέτοιου δικτύου είναι η ευελιξία του, καθώς με τη χρήση μιας μόνο εφαρμογής παρέχεται η δυνατότητα μεταφοράς δεδομένων προς οποιονδήποτε προορισμό.

Για παράδειγμα στο UNIX οι υποδοχές είναι ενοποιημένες με τις άλλες λειτουργίες I/O, καθώς παρέχεται μόνο ένα σύνολο περιγραφέων. Συνεπώς, διαδικασίες όπως η ανάγνωση (read) και η εγγραφή (write) είναι πολύ γενικές, καθώς μια εφαρμογή μπορεί να κάνει χρήση της ίδιας διαδικασίας για την αποστολή δεδομένων σε άλλο πρόγραμμα, σε ένα αρχείο, ή μέσω δικτύου.

2.8 Παράμετροι και η API υποδοχών

Ο προγραμματισμός υποδοχών διαφέρει από τις συμβατικές I/O, επειδή στην περίπτωση αυτή η εφαρμογή πρέπει να καθορίζει πολλές λεπτομέρειες για να κάνει χρήση μιας υποδοχής, όπως τη διεύθυνση ενός απομακρυσμένου υπολογιστή, τον αριθμό θύρας και το εάν η εφαρμογή θα ενεργήσει ως πελάτης ή ως διακομιστής (δηλαδή, εάν θα ξεκινήσει μια σύνδεση ή όχι) [16].

Για να αποφευχθεί μια μεμονωμένη λειτουργία υποδοχών με πολλές παραμέτρους, οι σχεδιαστές της API υποδοχών προτίμησαν να ορίσουν πολλές συναρτήσεις και μια εφαρμογή έχει τη δυνατότητα παροχής τιμών για κάθε μια από αυτές. Επομένως, η αναπαράσταση των επιλογών και των παραμέτρων σε μια διασύνδεση προγραμματισμού εφαρμογών γίνεται με τον ορισμό πολλών συναρτήσεων. Στην ουσία, η εφαρμογή δημιουργεί μια υποδοχή και στη συνέχεια ενεργοποιεί τις λειτουργίες για να καθορίσει τις λεπτομέρειες. Το πλεονέκτημα της προσέγγισης των υποδοχών είναι ότι οι περισσότερες λειτουργίες έχουν λιγότερες από τρεις παραμέτρους, αλλά ο εκάστοτε προγραμματιστής θα πρέπει να μπαίνει στη διαδικασία να καλεί πολλαπλές συναρτήσεις, όταν χρησιμοποιεί υποδοχές.

Στον πίνακα 1 συνοψίζονται οι βασικές λειτουργίες της API υποδοχών.

Πίνακας 1: Οι βασικές λειτουργίες της API υποδοχών

Όνομα	Χρήση από	Περιγραφή
accept	διακομιστή	αποδοχή εισερχόμενης σύνδεσης
bind	διακομιστή	καθορισμός διεύθυνσης IP & θύρας πρωτοκόλλου
close	και τους δύο	τερματισμός επικοινωνίας
connect	πελάτη	σύνδεση σε απομακρυσμένη εφαρμογή
getpeername	διακομιστή	απόκτηση διεύθυνσης IP πελάτη
getsockopt	διακομιστή	απόκτηση τρεχουσών επιλογών υποδοχής
listen	διακομιστή	προετοιμασία υποδοχής για χρήση
recv	και τους δύο	λήψη εισερχομένων δεδομένων ή μηνυμάτων
recvmsg	και τους δύο	λήψη δεδομένων (μηνυμάτων)
recvfrom	και τους δύο	λήψη μηνύματος και διεύθυνσης αποστολέα
send (write)	και τους δύο	αποστολή εξερχομένων δεδομένων ή μηνυμάτων
sendmsg	και τους δύο	αποστολή εξερχομένων μηνυμάτων
sendto	και τους δύο	αποστολή μηνυμάτων (μεταβλητή της sendmsg)
setsockopt	και τους δύο	αλλαγή επιλογών υποδοχής
shutdown	και τους δύο	τερματισμός σύνδεσης
socket	και τους δύο	δημιουργία υποδοχής

2.9 Ακολουθία κλήσεων διαδικασιών υποδοχών

Στην εικόνα 4 απεικονίζεται η αλληλουχία των κλήσεων των υποδοχών που γίνονται από σε ένα τυπικό μοντέλο πελάτη/διακομιστή που χρησιμοποιούν μια stream σύνδεση [22]. Στην εικόνα αυτή, ο πελάτης στέλνει πρώτος τα δεδομένα (write) και ο διακομιστής περιμένει να λάβει (read) τα δεδομένα αυτά. Στην πράξη, ορισμένες εφαρμογές μεριμνούν έτσι ώστε ο διακομιστής να στέλνει πρώτος δεδομένα (δηλαδή, οι write και read καλούνται με την αντίστροφη σειρά).

Όπως μπορούμε να διαπιστώσουμε από την εικόνα 4, ο διακομιστής καλεί επτά διαδικασίες υποδοχών, ενώ αντίστοιχα ο πελάτης καλεί έξι. Αρχικά ο διακομιστής καλεί τη socket για να δημιουργήσει μια υποδοχή. Μετά τη δημιουργία υποδοχής, καλεί τη bind για τον καθορισμό μια τοπικής θύρας πρωτοκόλλου υποδοχής, και τη listen για να θέσει την υποδοχή σε παθητική κατάσταση. Στη συνέχεια, ο πελάτης καλεί τη socket για να δημιουργήσει μια υποδοχή και την connect για να συνδέσει την υποδοχή με ένα διακομιστή και ο διακομιστής καλεί την accept για να δεχθεί την επόμενη εισερχόμενη αίτηση σύνδεσης. Η ανταλλαγή δεδομένων επιτυγχάνεται μέσω

των συνεχόμενων κλήσεων των write και read, τόσο από τον πελάτη όσο και από τον διακομιστή. Μετά τη λήψη όλων των δεδομένων, καλείται η close από τον πελάτη για το κλείσιμο της υποδοχής, αλλά και από τον διακομιστή για το κλείσιμο της σύνδεσης. Αυτή η διαδικασία επαναλαμβάνεται σε περίπτωση αποδοχής της επόμενης εισερχόμενης σύνδεσης.

2.10 Διαδικασίες που υλοποιούν την API υποδοχών

Η υλοποίηση της API υποδοχών γίνεται με κλίση κάποιων από τις διαδικασίες που παρουσιάστηκαν στον Πίνακα 1. Όπως γίνεται φανερά κατανοητό από τον πίνακα αυτόν, οι περισσότερες χρησιμοποιούνται τόσο από τους πελάτες όσο και από τους διακομιστές. Υπάρχουν όμως και μερικές που χρησιμοποιούνται μόνο από τους πελάτες ή μόνο από τους διακομιστές. Στη συνέχεια της ενότητας θα παρουσιαστούν κάποιες από τις διαδικασίες αυτές, ταξινομημένες ανάλογα με τη χρήση τους από πελάτες και διακομιστές ή όχι [16].

2.10.1 Διαδικασίες που χρησιμοποιούνται από πελάτες και διακομιστές

I) Η διαδικασία socket

Η διαδικασία *socket* δημιουργεί μια υποδοχή και επιστρέφει έναν ακέραιο περιγραφέα:

descriptor = socket(protofamily, type, protocol)

Το όρισμα *protofamily* καθορίζει την οικογένεια πρωτοκόλλων που θα χρησιμοποιείται για την υποδοχή αυτή. Η πιο συνηθισμένη επιλογή είναι το *PF_INET*, το οποίο καθορίζει την οικογένεια πρωτοκόλλων TCP/IP που χρησιμοποιείται στο Διαδίκτυο.

Το όρισμα *type* καθορίζει τον τύπο επικοινωνίας που θα χρησιμοποιεί η υποδοχή. Οι δύο πιο συνηθισμένοι τύποι είναι η συνδεσμική μεταφορά ρεύματος (*SOCK_STREAM*), και η ασυνδεσμική μεταφορά μηνυμάτων (*SOCK_DGRAM*).

Το όρισμα *protocol* καθορίζει ένα συγκεκριμένο πρωτόκολλο μεταφοράς, το οποίο χρησιμοποιείται για την υποδοχή αυτή. Η ύπαρξη των ορισμάτων *protocol* και *type*, δίνει τη δυνατότητα σε μια οικογένεια πρωτοκόλλων, να συμπεριλάβει δύο ή περισσότερα πρωτόκολλα με τις ίδιες υπηρεσίες. Οι τιμές που μπορούν να χρησιμοποιηθούν για το όρισμα *protocol* εξαρτάται από την οικογένεια πρωτοκόλλων.

II) Η διαδικασία send

Η διαδικασία *send* χρησιμοποιείται από πελάτες και διακομιστές για την αποστολή δεδομένων. Τυπικά ένας πελάτης αποστέλλει αιτήματα, ενώ ένας διακομιστής αποστέλλει απαντήσεις. Η διαδικασία *send* περιέχει τέσσερα ορίσματα:

send(socket, data, length, flags)

Το όρισμα *socket* είναι ένας περιγραφέας της υποδοχής που χρησιμοποιείται. Το όρισμα *data* είναι η διεύθυνση της μνήμης των δεδομένων που αποστέλλονται. Το όρισμα *length* είναι ένας ακέραιος που φανερώνει τον αριθμό των byte των δεδομένων που αποστέλλονται. Τέλος, το όρισμα *flags* περιέχει bit που αιτούνται ειδικές επιλογές [16].

III) Η διαδικασία recv

Η διαδικασία *recv* χρησιμοποιείται από πελάτες και διακομιστές για τη λήψη δεδομένων που αποστέλλονται αμφίδρομα. Η διαδικασία έχει τη μορφή:

recv(socket, buffer, length, flags)

Το όρισμα *socket* είναι ο περιγραφέας της υποδοχής από την οποία θα ληφθούν τα δεδομένα. Το όρισμα *buffer* καθορίζει τη διεύθυνση μνήμης, όπου θα αποθηκευθεί το εισερχόμενο μήνυμα. Το όρισμα *length* καθορίζει το μέγεθος του χώρου προσωρινής αποθήκευσης (buffer). Τέλος, το όρισμα *flags* επιτρέπει στον καλούντα να ελέγχει διάφορες λεπτομέρειες (επιτρέπει, για παράδειγμα, σε μια εφαρμογή να βγάλει ένα αντίγραφο του εισερχόμενου μηνύματος χωρίς να αφαιρέσει το μήνυμα από την υποδοχή).

Η διαδικασία *recv* μπλοκάρει μέχρι την άφιξη των δεδομένων και στη συνέχεια τοποθετεί τόσα byte των λαμβανόμενων δεδομένων στο χώρο προσωρινής αποθήκευσης ανάλογα με τον ακέραιο αριθμό *length* (η επιστρεφόμενη τιμή της κλήσης της διεργασίας καθορίζει τον αριθμό των byte που αποθηκεύονται).

IV) Η διαδικασία close

Η διαδικασία *close* καθοδηγεί το λειτουργικό σύστημα, ώστε να τερματίσει τη χρήση μίας υποδοχής. Έχει τη μορφή:

close (socket)

Το όρισμα *socket* είναι ο περιγραφέας της υποδοχής που θα κλείσει. Η *close* τερματίζει τη σύνδεση πριν κλείσει την υποδοχή (δηλαδή ενημερώνει την άλλη μεριά). Το κλείσιμο μιας υποδοχής τερματίζει άμεσα τη χρήση της και ο περιγραφέας αποδεσμεύεται, γεγονός που αποτρέπει την εφαρμογή από την αποστολή ή λήψη περισσότερων δεδομένων.

2.10.2 Διαδικασίες που χρησιμοποιούνται από διακομιστές

I) Η διαδικασία bind

Μετά τη δημιουργία της, μια υποδοχή δεν περιέχει καμία πληροφορία όσον αφορά την τοπική ή την απομακρυσμένη διεύθυνση αλλά και τον αριθμό θύρας πρωτοκόλλου. Η διαδικασία *bind* παρέχει έναν αριθμό θύρας πρωτοκόλλου, στην οποία ο διακομιστής θα περιμένει για την πραγματοποίηση της επαφής. Η *bind* δέχεται τρία ορίσματα:

bind(socket, localaddr, addrlen)

Το όρισμα *socket* είναι ο περιγραφέας της υποδοχής που έχει δημιουργηθεί. Το όρισμα *localaddr* καθορίζει την τοπική διεύθυνση που θα δοθεί στην υποδοχή. Το όρισμα *addrlen* είναι ο ακέραιος αριθμός που καθορίζει το μήκος της διεύθυνσης αυτής.

Λόγω του ότι οι υποδοχές μπορούν να χρησιμοποιηθούν σε οποιοδήποτε πρωτόκολλο, η μορφή της διεύθυνσης εξαρτάται από το πρωτόκολλο που χρησιμοποιείται κάθε φορά. Η API υποδοχών καθορίζει μια γενικότερη μορφή της διεύθυνσης αυτής, η ακριβή μορφή της οποίας απαιτείται να καθοριστεί από την εκάστοτε οικογένεια πρωτοκόλλων. Η γενική μορφή της διεύθυνσης καθορίζεται από τη δομή *sockaddr*, η οποία συνήθως περιλαμβάνει τρία πεδία:

```
struct sockaddr {
        u_char sa_len; /* total length of the address */
        u_char sa_family; /* family of the address */
        char sa_data[14]; /* the address itself */
};
```

Κάθε οικογένεια πρωτοκόλλων ορίζει την ακριβή μορφή των διευθύνσεων που χρησιμοποιείται στο πεδίο *sa_data* της δομής *sockaddr*. Στο παρακάτω παράδειγμα απεικονίζονται πρωτόκολλα TCP/IP να χρησιμοποιούν τη δομή *sockaddr_in* για τον ορισμό μιας διεύθυνσης:

```
struct sockaddr_in {
        u_char sin_len; /* total length of the address */
        u_char sin_family; /* family of the address */
        u_short sin_port; /* protocol port number */
        struct in_addr sin_addr;/* IP address of computer */
        char sin_zero[8]; /* not used (set to zero) */
};
```

Κάθε διεύθυνση προσδιορίζει έναν υπολογιστή αλλά και μια συγκεκριμένη εφαρμογή στον ίδιο υπολογιστή. Το πεδίο *sin_addr* περιέχει τη διεύθυνση IP του υπολογιστή, ενώ το πεδίο *sin_port* περιέχει τον αριθμό θύρας πρωτοκόλλου μιας εφαρμογής. Επίσης, αν και το TCP/IP χρειάζεται έξι οκτάδες για την αποθήκευση μιας πλήρους διεύθυνσης, ενώ η γενική δομή *sockaddr* δεσμεύει δεκατέσσερις οκτάδες. Επομένως, η δομή *sockaddr*_in ορίζει τη μορφή που χρησιμοποιεί το TCP/IP για την αναπαράσταση μιας διεύθυνσης. Η δομή αυτή περιέχει πεδία για μια διεύθυνση IP αλλά και για έναν αριθμό θύρας πρωτοκόλλου. Παρ' όλα αυτά η API υποδοχών περιλαμβάνει μία συμβολική σταθερά κι έτσι επιτρέπεται στο διακομιστή να καθορίζει μια θύρα πρωτοκόλλου για οποιαδήποτε από τις διευθύνσεις IP του υπολογιστή.

II) Η διαδικασία listen

Μετά τη χρήση της διαδικασίας *bind* για τον καθορισμό της θύρας πρωτοκόλλου, ο διακομιστής καλεί τη διαδικασία *listen* με σκοπό να υποδείξει στο λειτουργικό σύστημα να θέσει μια υποδοχή σε παθητική κατάσταση, έτσι ώστε να μπορέσει να χρησιμοποιηθεί για αναμονή πραγματοποίησης επαφής με πελάτες. Η διαδικασία *listen* δέχεται δύο ορίσματα:

listen(socket, queuesize)

Το όρισμα *socket* είναι ο περιγραφέας μίας υποδοχής που έχει δημιουργηθεί και υπάρχει αντιστοιχία αυτής με μία τοπική διεύθυνση. Το όρισμα *queuesize* καθορίζει το μήκος ουράς αιτήσεων της υποδοχής.

Ένα λειτουργικό σύστημα δημιουργεί ξεχωριστές ουρές αιτήσεων για κάθε υποδοχή. Αρχικά, η ουρά είναι άδεια. Κάθε ένα από τα αιτήματα που φτάνουν από τους πελάτες, τοποθετούνται στην ουρά αυτή. Όταν ο διακομιστής ζητά να ανακτήσει μια εισερχόμενη αίτηση από την υποδοχή, το σύστημα εξάγει το επόμενο αίτημα από την ουρά. Το μήκος της ουράς είναι σημαντικό. Αν η ουρά είναι πλήρης όταν φτάνει ένα αίτημα, το σύστημα απορρίπτει το αίτημα αυτό. Η χρήση της ουράς αιτήσεων είναι σημαντική καθώς επιτρέπει στο σύστημα να έχει σε αναμονή τις νέες αιτήσεις που έρχονται, ενώ ο διακομιστής είναι απασχολημένος σε διεκπεραίωση υπάρχουσας αίτησης. Τέλος, επιτρέπεται σε κάθε διακομιστή να επιλέξει ένα μέγιστο μήκος ουράς και να το προσαρμόσει στην αντίστοιχη υπηρεσία.

III) Η διαδικασία accept

Η διαδικασία *accept* χρησιμοποιείται από ένα διακομιστή που λειτουργεί με συνδεσμική μεταφορά, έτσι ώστε να μπορέσει να δεχτεί την επόμενη αίτηση σύνδεσης. Αν υπάρχει αίτηση στην ουρά η *accept* επιστρέφει αμέσως, ενώ σε διαφορετική περίπτωση το σύστημα μπλοκάρει το διακομιστή μέχρι να εμφανιστεί πελάτης. Μετά την αποδοχή της σύνδεσης, ο διακομιστής χρησιμοποιεί τη σύνδεση αυτή για να αλληλεπιδράσει με τον πελάτη. Μετά την ολοκλήρωση της επικοινωνίας, ο διακομιστής κλείνει τη σύνδεση.

Η διαδικασία *accept* έχει την ακόλουθη μορφή:

newsock = accept(socket, caddress, caddresslen)

Το όρισμα *socket* είναι ο περιγραφέας μιας υποδοχής που έχει δημιουργήσει ο διακομιστής και την έχει αντιστοιχίσει σε μια συγκεκριμένη θύρα πρωτοκόλλου. Το όρισμα *caddress* είναι η διεύθυνση μιας δομής του τύπου *sockaddr*. Το όρισμα *caddresslen* είναι ένας δείκτης σε έναν ακέραιο αριθμό. Η *accept* θέτει στο όρισμα *caddress* τη διεύθυνση του πελάτη που δημιούργησε τη σύνδεση και στο όρισμα *caddresslen* το μήκος της διεύθυνσης.

Η *accept* δημιουργεί μια νέα υποδοχή για τη σύνδεση και επιστρέφει στο διακομιστή που την κάλεσε, τον περιγραφέα της. Στη συνέχεια ο διακομιστής για την επικοινωνία με τον πελάτη χρησιμοποιεί την νέα υποδοχή, και ύστερα κλείνει τη

σύνδεση. Τέλος, χρησιμοποιεί την αρχική υποδοχή για να δεχτεί την επόμενη σύνδεση με τον πελάτη, αφού στο μεσοδιάστημα αυτή έχει μείνει αμετάβλητη.

2.10.3 Διαδικασίες που χρησιμοποιούνται από πελάτες

Η διαδικασία connect

Η διαδικασία *connect* χρησιμοποιείται από τους πελάτες που θέλουν να πραγματοποιήσουν σύνδεση με ένα συγκεκριμένο διακομιστή. Η μορφή της διεργασίας είναι η εξής:

connect(socket, saddress, saddresslen)

Το όρισμα *socket* είναι ο περιγραφέας της υποδοχής που χρησιμοποιείται για τη σύνδεση του πελάτη με το διακομιστή. Το όρισμα *saddress* έχει τη δομή του τύπου *sockaddr* και καθορίζει τη διεύθυνση του διακομιστή και τον αριθμό της θύρας πρωτοκόλλου. Το όρισμα *saddresslen* καθορίζει το μήκος της διεύθυνσης του διακομιστή, μετρούμενη σε byte.

Η διαδικασία *connect* η οποία πραγματοποιείται από τους πελάτες, έχει τις εξής χρησιμότητες:

Σε περίπτωση χρησιμοποίησης για συνδεσμική μεταφορά, η *connect* πραγματοποιεί μία σύνδεση μεταφοράς με ένα καθορισμένο διακομιστή

Σε περίπτωση χρησιμοποίησης για ασυνδεσμική μεταφορά, καταγράφει τη διεύθυνση του διακομιστή στην υποδοχή, για να μπορεί ο πελάτης να στέλνει πολλά μηνύματα στον ίδιο διακομιστή χωρίς να υπάρχει απαίτηση καθορισμού της διεύθυνσης καθορισμού σε κάθε μήνυμα

2.11 Υποδοχές, νήματα και κληρονομικότητα

Η API υποδοχών έχει σχεδιαστεί για να λειτουργεί καλύτερα με ταυτόχρονους διακομιστές [16]. Έτσι λοιπόν, κάθε νέο νήμα που δημιουργείται κληρονομεί ένα αντίγραφο όλων των ανοιχτών υποδοχών από το νήμα που το δημιούργησε.

Οι υποδοχές χρησιμοποιούν ένα μηχανισμό καταμέτρησης αναφορών. Έτσι λοιπόν, όταν μια υποδοχή δημιουργείται, το σύστημα θέτει στο μετρητή την τιμή 1. Όταν δημιουργείται ένα νέο νήμα, το σύστημα δίνει στο νήμα μια λίστα όλων των υποδοχών του προγράμματος και αυξάνει το μετρητή αναφορών ανά υποδοχή κατά 1. Όταν ένα νήμα χρησιμοποιεί την *close*, τότε μειώνεται ο μετρητή αναφορών της υποδοχής κατά 1 και την καταργεί από τη λίστα του νήματος.

Συνοψίζοντας, η υποδοχή που χρησιμοποιεί ένας ταυτόχρονος διακομιστής για την αποδοχή συνδέσεων υφίσταται για όσο διάστημα εκτελείται το κύριο νήμα του διακομιστή. Μια υποδοχή που χρησιμοποιείται για μια συγκεκριμένη σύνδεση υφίσταται για όσο διάστημα υπάρχει και το νήμα που δημιουργήθηκε για να τη χειριστεί.

2.12 Σύνοψη

Το βασικό μοντέλο επικοινωνίας που χρησιμοποιούν οι εφαρμογές σε διαδικτυακά περιβάλλοντα είναι γνωστό ως μοντέλο πελάτη/διακομιστή. Στο μοντέλο αυτό ο διακομιστής είναι ένα πρόγραμμα που περιμένει παθητικά για επαφή, ενώ ο πελάτης είναι ένα πρόγραμμα που ξεκινά ενεργά επαφή με ένα διακομιστή.

Σε κάθε υπολογιστή του διαδικτυακού περιβάλλοντος αντιστοιχεί μια μοναδική διεύθυνση, ενώ σε κάθε υπηρεσία, όπως το ηλεκτρονικό ταχυδρομείο ή πρόσβαση στο διαδίκτυο, εκχωρείται ένα μοναδικό αναγνωριστικό γνωστό ως αριθμός θύρας πρωτοκόλλου. Κατά την εκκίνησή του, ένας διακομιστής καθορίζει έναν αριθμό θύρας πρωτοκόλλου. Όταν ένας πελάτης έρχεται σε επαφή με ένα διακομιστή, καθορίζει τη διεύθυνση του υπολογιστή στον οποίο εκτελείται ο διακομιστής, καθώς και τον αριθμό θύρας πρωτοκόλλου που χρησιμοποιεί ο διακομιστής.

Ένας πελάτης μπορεί να έχει πρόσβαση σε περισσότερες από μία υπηρεσία ή μπορεί να έχει πρόσβαση σε πολλούς διακομιστές. Ένας διακομιστής μιας υπηρεσίας μπορεί να γίνει και πελάτης σε άλλες υπηρεσίες.

Μια Διεπαφή Προγραμματισμού Εφαρμογών (API) καθορίζει τις λεπτομέρειες για το πώς ένα πρόγραμμα εφαρμογής αλληλεπιδρά με το λογισμικό του πρωτοκόλλου. Αν και οι λεπτομέρειες εξαρτώνται από το λειτουργικό σύστημα, η API υποδοχών είναι ένα ντε φάκτο πρότυπο. Ένα πρόγραμμα δημιουργεί μια υποδοχή και στη συνέχεια επικαλείται μια σειρά από διαδικασίες για να χρησιμοποιήσει την υποδοχή αυτή. Ένας διακομιστής που χρησιμοποιεί σύνδεση stream καλεί τις διεργασίες *socket*, *bind*, *listen*, *accept*, *recv*, *send* και *close*. Στην ίδια σύνδεση ένας πελάτης καλεί τις διεργασίες *socket*, *connect*, *send*, *recv* και *close*.

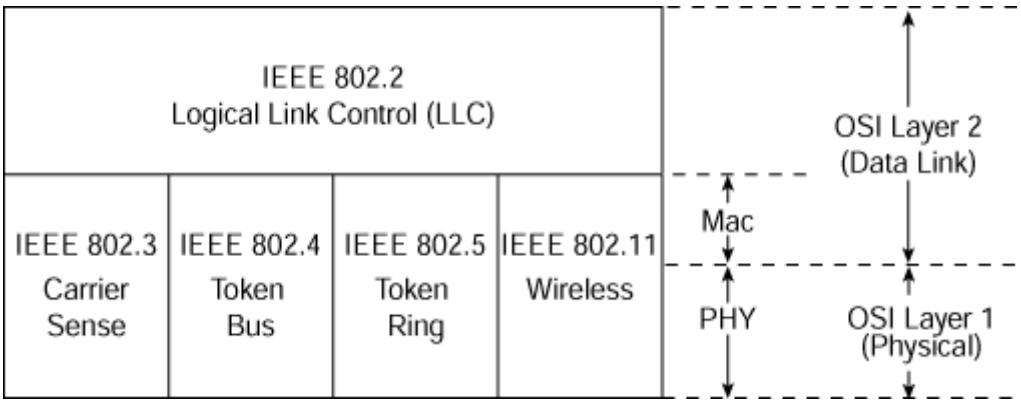
Λόγω του γεγονότος ότι πολλοί διακομιστές είναι ταυτόχρονοι, οι υποδοχές σχεδιάζονται για να λειτουργούν με ταυτόχρονες εφαρμογές. Κατά τη δημιουργία ενός νέου νήματος, κληροδοτούνται οι προσβάσεις σε όλες τις υποδοχές που κατείχε το αρχικό νήμα.

3 Το πρωτόκολλο IEEE 802.2 – Logical Link Control

3.1 Η οικογένεια πρωτοκόλλων IEEE 802

Ο Διεθνής Οργανισμός Τυποποίησης (IEEE) παρήγαγε πρότυπα για την λειτουργία των δικτύων LAN και MAN. Τα πρότυπα αυτά είναι στο σύνολό τους γνωστά ως IEEE 802 και έχουν γίνει αποδεκτά από την ISO ως διεθνή πρότυπα, γνωστά και ως ISO 8802 [23].

Όπως παρουσιάζεται και στην εικόνα 7, η οικογένεια IEEE 802 περιλαμβάνει μια σειρά από πρωτόκολλα. Το φυσικό επίπεδο (physical layer) και το επίπεδο Ελέγχου Πρόσβασης Μέσου (Medium Access Control – MAC layer) των πρωτοκόλλων 802 είναι οργανωμένα σε μια ξεχωριστή σειρά πρωτοκόλλων από το επίπεδο Ελέγχου Λογικής Ζεύξης (Logical Link Control – LLC layer) για λόγους ανεξαρτησίας του ελέγχου πρόσβασης του μέσου, από το ίδιο το μέσο και την τοπολογία του δικτύου.



Εικόνα 7: Η οικογένεια πρωτοκόλλων IEEE 802 ταυτίζεται με τα επίπεδα 1 και 2 του μοντέλου αναφοράς OSI

3.2 Σύνοψη του πρωτοκόλλου IEEE 802.2 – LLC

Το φυσικό επίπεδο δεν μπορεί να θεωρηθεί ότι παρέχει αξιόπιστη μετάδοση πληροφοριών. Το επίπεδο MAC των δικτύων LAN, μπορεί μεν να επιλύει το πρόβλημα της πολλαπλής πρόσβασης, αλλά δεν προσφέρει καμία βελτίωση στο θέμα της αξιοπιστίας. Η κατάσταση από πλευράς ποιότητας ζεύξης παραμένει η ίδια τόσο στα άνω του MAC επίπεδα, όσο και στα άνω του φυσικού επιπέδου των ζεύξεων σημείου προς σημείο. Για την διόρθωση των σφαλμάτων και των απωλειών πακέτων λόγω συμφόρησης ή κακής δρομολόγησης στο δίκτυο LAN απαιτείται η ύπαρξη ενός πρωτοκόλλου που να διαθέτει τους κατάλληλους μηχανισμούς ανίχνευσης λαθών και επαναποστολής των δεδομένων.

Για το σκοπό αυτό, εισήχθη το πρωτόκολλο LLC, το οποίο είναι το υψηλότερο στρώμα του μοντέλου αναφοράς IEEE 802 και παρέχει λειτουργίες παρόμοιες με το

παραδοσιακό πρωτόκολλο ελέγχου ζεύξης δεδομένων HDLC (High-Level Data Link Control). Πιο συγκεκριμένα, στις 7 Μάη 1998, καθορίστηκε από το πρότυπο ISO / IEC 8802-2 (ANSI / IEEE 802.2), ως το ανώτερο μέρος του Επιπέδου Ζεύξης Δεδομένων του μοντέλου αναφοράς OSI [24]. Σκοπός του LLC είναι η ανταλλαγή δεδομένων μεταξύ των τελικών χρηστών σε ένα LAN με χρήση μιας 802-based ελεγχόμενης MAC ζεύξης, με άλλα λόγια παρέχει έναν τρόπο επικοινωνίας μεταξύ των ανωτέρων επιπέδων του μοντέλου αναφοράς OSI με οποιονδήποτε τύπο επιπέδου MAC (για παράδειγμα Ethernet - IEEE 802.3 CSMA/CD ή Token Ring IEEE 802.5 Token Passing) (Εικ.8). Παρέχει διευθυνσιοδότηση και έλεγχο δεδομένων ζεύξης, διαθέτει τον ίδιο μηχανισμό αρίθμησης πλαισίων, αρίθμησης αναγνωρίσεων (ACK), παραθύρου, ελέγχου ροής, κ.τ.λ., και είναι ανεξάρτητο από την τοπολογία του δικτύου, το μέσο μετάδοσης αλλά και την τεχνική ελέγχου πρόσβασης στο μέσο που επιλέγεται.

<table>
<tr><td>Network</td><td>Unix IP
SAP: 80</td><td>IBM Netbios
SAP: F0</td><td>Novell IPX
SAP: E0</td></tr>
<tr><td rowspan="2">Data Link</td><td colspan="3">IEEE 802.2 Logical Link Control Layer (LLC)</td></tr>
<tr><td colspan="3">IEEE 802.3 CSMA/CD
Medium Access Control Layer</td></tr>
<tr><td>Physical</td><td>802.3 - 10Base5</td><td>802.3a - 10Base2</td><td>802.3i - 10BaseT</td></tr>
</table>

Εικόνα 8: Το επίπεδο LLC

Υψηλότερα επίπεδα, όπως το TCP/IP, περνούν τα δεδομένα του χρήστη μέχρι το LLC με εξασφαλισμένη μια μετάδοση χωρίς σφάλματα σε όλο το δίκτυο. Το LLC με τη σειρά του προσθέτει μια κεφαλίδα ελέγχου, δημιουργώντας μια μονάδα δεδομένων πρωτοκόλλου LLC (Protocol Data Unit - PDU) και χρησιμοποιεί την πληροφορία ελέγχου στη λειτουργία του πρωτοκόλλου LLC.

Πριν τη μεταφορά των δεδομένων, η PDU του πρωτοκόλλου LLC μεταφέρεται στο επίπεδο MAC μέσω του Σημείου Πρόσβασης Υπηρεσίας MAC (Service Access Point - SAP), το οποίο προσθέτει πληροφορίες ελέγχου στην αρχή και στο τέλος του πακέτου, σχηματίζοντας ένα πλαίσιο MAC. Οι πληροφορίες ελέγχου στο πλαίσιο είναι αναγκαίες για τη λειτουργία του πρωτοκόλλου MAC. Η PDU του πρωτοκόλλου LLC έχει τη γενική μορφή που απεικονίζεται στην εικόνα 9.

<table>
<tr><th colspan="3">802.2 LLC Header</th><th rowspan="2">Information</th></tr>
<tr><th>DSAP address</th><th>SSAP address</th><th>Control</th></tr>
<tr><td>8 bits</td><td>8 bits</td><td>8 or 16 bits</td><td>multiple of 8 bits</td></tr>
</table>

Εικόνα 9: Γενική μορφή LLC PDU

Οι λειτουργίες του πρωτοκόλλου LLC μπορούν να συνοψισθούν στις εξής [25]:

- Διαχείριση των επικοινωνιών του επιπέδου ζεύξης δεδομένων
- Διευθυνσιοδότηση ζεύξης
- Ορισμός των σημείων πρόσβασης υπηρεσίας (SAP)
- Έλεγχος δεδομένων ζεύξης

Πίνακας 2: Διευθύνσεις SAP & αντίστοιχες λειτουργίες και πρωτόκολλα

Διεύθυνση	Λειτουργίες & πρωτόκολλα
00	Null LSAP
02	Individual LLC Sublayer Management Function
03	Group LLC Sublayer Management Function
04	IBM SNA Path Control (individual)
05	IBM SNA Path Control (group)
06	ARPANET Internet Protocol (IP)
08	SNA
0C	SNA
0E	PROWAY (IEC955) Network Management & Initialization
18	Texas Instruments
42	IEEE 802.1 Bridge Spannning Tree Protocol
4E	EIA RS-511 Manufacturing Message Service
7E	ISO 8208 (X.25 over IEEE 802.2 Type 2 LLC)
80	Xerox Network Systems (XNS)
86	Nestar
8E	PROWAY (IEC 955) Active Station List Maintenance
98	ARPANET Address Resolution Protocol (ARP)
AA	SubNetwork Access Protocol (SNAP)
BC	Banyan VINES
E0	Novell NetWare
F0	IBM NetBIOS
F4	IBM LAN Management (individual)
F5	IBM LAN Management (group)
F8	IBM Remote Program Load (RPL)
FA	Ungermann-Bass
FE	ISO Network Layer Protocol
FF	Global LSAP

3.3 Σημεία πρόσβασης υπηρεσίας

Τα σημεία πρόσβασης υπηρεσίας (SAP) αποτελούν ετικέτες αναγνώρισης των τελικών σημείων του δικτύου που χρησιμοποιούνται στη δικτύωση διασύνδεσης ανοικτών συστημάτων (OSI). Ένα SAP είναι μια εννοιολογική θέση (λογική πύλη) στην οποία ένα επίπεδο OSI μπορεί να ζητήσει τις υπηρεσίες από ένα άλλο επίπεδο OSI [26].

Στην περίπτωση του πρωτοκόλλου LLC, τα SAP είναι μια πύλη (λογική ζεύξη) με το πρωτόκολλο του επιπέδου δικτύου. Επομένως, τα SAP αποτελούν τη μέθοδο την οποία χρησιμοποιεί το LLC για να διαπιστώσει το ποια πρωτόκολλα επικοινωνούν μεταξύ τους και συνδυάζουν τα εισερχόμενα και εξερχόμενα πλαίσια MAC, τα οποία και οδηγούν προς την κατάλληλη εφαρμογή ή το πρωτόκολλο ανωτέρου επιπέδου.

Στα δίκτυα LAN, όπου λειτουργούν πολλαπλά πρωτόκολλα, κάθε πρωτόκολλο επιπέδου δικτύου έχει το δικό του SAP. Για παράδειγμα, το TCP/IP του Unix, το SPX/IPX του Novell και το Netbios της IBM έχουν διαφορετικό SAP, για να προσδιορίζονται πιο εύκολα. Το NetBios χρησιμοποιεί τη διεύθυνση F0, το IP την διεύθυνση 06, το SNA τη διεύθυνση 04 και η Διαχείριση Δικτύου τη διεύθυνση F4. Η διεύθυνση 00 δεν διατίθεται σε κανένα πρωτόκολλο. Στον πίνακα 2, παρουσιάζονται κάποιες από τις διαδικτυακές λειτουργίες και πρωτόκολλα στα οποία έχουν αποδοθεί συγκεκριμένες διευθύνσεις SAP.

3.4 Υπηρεσίες IEEE 802.2 LLC

Το LLC παρέχει τις ακόλουθες τρεις υπηρεσίες για ένα πρωτόκολλο επιπέδου δικτύου [28]:

- Μη επιβεβαιωμένη ασυνδεσμική υπηρεσία (Unacknowledged connectionless service)
- Συνδεσμική υπηρεσία (Connection-oriented service)
- Επιβεβαιωμένη ασυνδεσμική υπηρεσία (Acknowledged connectionless service)

Οι υπηρεσίες αυτές ισχύουν για την επικοινωνία μεταξύ των επιπέδων LLC που βρίσκονται στον υπολογιστή – αφετηρία και στον υπολογιστή – προορισμό ή στον πελάτη και στον διακομιστή, αντίστοιχα, στο μοντέλο πελάτη/διακομιστή. Συνήθως, οι υπηρεσίες είναι προαιρετικές.

Και τα τρία πρωτόκολλα LLC χρησιμοποιούν την ίδια μορφή PDU που αναλύεται στην επόμενη ενότητα.

3.4.1 Μη επιβεβαιωμένη ασυνδεσμική υπηρεσία

Η μη επιβεβαιωμένη ασυνδεσμική υπηρεσία (unacknowledged connectionless service) ή υπηρεσία LLC τύπου 1 (LLC1) είναι μια υπηρεσία τύπου αυτοδύναμων πακέτων (datagrams) που δεν περιλαμβάνει μηχανισμούς ελέγχου σφάλματος ή ελέγχου ροής.

Η υπηρεσία LLC1 αποστέλλει και λαμβάνει Μονάδες Δεδομένων Υπηρεσίας Ζεύξης (Link Service Data Unit – LSDU) χωρίς να υπάρχει ανάγκη επιβεβαίωσης μετάδοσης. Η υπηρεσία αυτή δεν συνεπάγεται τη δημιουργία μιας σύνδεσης επιπέδων ζεύξης δεδομένων, υποστηρίζει ατομική (individual), πολλαπλή (multicast) και ακροαματική (broadcast) επικοινωνία και είναι κατάλληλη για πρωτόκολλα ανώτερων επιπέδων που πραγματοποιούν αλληλουχία, διευθυνσιοδότηση, δρομολόγηση και ανάκτηση δεδομένων, όπως τα IPX, TCP/IP, Vines, XNS, AppleTalk κτλ.

Η μη επιβεβαιωμένη ασυνδεσμική υπηρεσία έχει μικρότερη ανάγκη για δικτυακούς πόρους σε σχέση με τους άλλους δύο τύπους υπηρεσιών LLC καθώς απλά στέλνει και λαμβάνει PDU του πρωτοκόλλου LLC χωρίς επιβεβαιωμένη παραλαβή. Επειδή η παροχή των δεδομένων δεν είναι εγγυημένη, ένα υψηλότερο επίπεδο, όπως το TCP, πρέπει να ασχοληθεί με τα ζητήματα αξιοπιστίας και ακεραιότητας των δεδομένων.

Η μη επιβεβαιωμένη ασυνδεσμική υπηρεσία παρουσιάζει πλεονεκτήματα στις ακόλουθες περιπτώσεις:

- Στην περίπτωση που ανώτερα στρώματα της στοίβας πρωτοκόλλων παρέχουν τους απαραίτητους μηχανισμούς αξιοπιστίας και ελέγχου ροής, όπως για παράδειγμα τα πρωτόκολλα μεταφοράς TCP ή ISO, η εκ νέου παρουσία τους στο πρωτόκολλο LLC θα ήταν αναποτελεσματική. Σε αυτή την περίπτωση, είναι σκόπιμη η χρήση μη επιβεβαιωμένης ασυνδεσμικής υπηρεσίας.
- Η ύπαρξη ανατροφοδότησης σχετικά με την επιτυχή παράδοση των πληροφοριών, δεν είναι πάντα απαραίτητη. Το overhead της δημιουργίας και διατήρησης μιας σύνδεσης μπορεί να είναι αναποτελεσματικό σε εφαρμογές που αφορούν την περιοδική δειγματοληψία των πηγών δεδομένων, όπως για παράδειγμα οι αισθητήρες παρακολούθησης. Η μη επιβεβαιωμένη ασυνδεσμική υπηρεσία μπορεί να ικανοποιήσει καλύτερα τις απαιτήσεις αυτές.

3.4.2 Συνδεσμική υπηρεσία

Η συνδεσμική υπηρεσία (connection oriented service) ή υπηρεσία LLC τύπου 2 (LLC2) δημιουργεί μια λογική σύνδεση μεταξύ δύο Σημείων Πρόσβασης Υπηρεσίας Ζεύξης (Link Service Access Points – LSAP) που χρειάζεται να ανταλλάξουν δεδομένα. Η υπηρεσία LLC2 παρέχει έλεγχο ροής, αλληλουχία των πλαισίων και έλεγχο σφάλματος. Η υπηρεσία αυτή συνεπάγεται τη δημιουργία μιας σύνδεσης μεταξύ των πρωτοκόλλων LLC πραγματοποιώντας δημιουργία σύνδεσης, μεταφορά δεδομένων και λειτουργίες τερματισμού σύνδεσης. Η συνδεσμική υπηρεσία μπορεί να συνδέσει μόνο δύο τερματικά και ως εκ τούτου, δεν υποστηρίζει πολλαπλές συνδέσεις (multicast) ή ακρόαση. Παρουσιάζει πολλά πλεονεκτήματα, κυρίως εάν τα ανώτερα στρώματα της στοίβας πρωτοκόλλων δεν παρέχουν την απαραίτητη αξιοπιστία και τους μηχανισμούς ελέγχου ροής, γεγονός το οποίο συμβαίνει κατά

κανόνα στους ελεγκτές τερματικών και σε περιβάλλοντα που τρέχουν πρωτόκολλα όπως το NetBios ή το SNA.

Ο έλεγχος ροής είναι ένα χαρακτηριστικό των πρωτοκόλλων που εξασφαλίζει ότι ένα τερματικό εκπομπής δεν θα κατακλύσει ένα τερματικό λήψης με δεδομένα. Με τον έλεγχο ροής, κάθε τερματικό διαθέτει ένα πεπερασμένο ποσό πόρων μνήμης για να αποθηκεύει τις απεσταλμένες και λαμβανόμενες PDU.

Ένα από τα μεγαλύτερα ζητήματα των δικτύων είναι η ύπαρξη διαφόρων παραγόντων που μπορούν να οδηγήσουν σε σφάλματα μετάδοσης. Για την προστασία από τα σφάλματα μετάδοσης, η συνδεσμική υπηρεσία και η επιβεβαιωμένη ασυνδεσμική υπηρεσία LLC χρησιμοποιούν μηχανισμούς ελέγχου σφαλμάτων που εντοπίζουν και διορθώνουν τα σφάλματα που προκύπτουν κατά τη μετάδοση των PDU. Ένας από τους μηχανισμούς αυτούς, ο μηχανισμός Αυτόματης Επανάληψης Αιτήματος (Automatic Repeat Request – ARQ) του πρωτοκόλλου LLC αναγνωρίζει τους παρακάτω δύο τύπους σφαλμάτων:

- *Απώλεια PDU*: Στην περίπτωση αυτή, μια PDU αδυνατεί να φτάσει στο άλλο άκρο
- *Καταστροφή PDU*: Στην περίπτωση αυτή, η PDU φτάνει στο άλλο άκρο, αλλά μερικά bit είναι αλλοιωμένα

Όταν ένα πλαίσιο φθάσει σε ένα τερματικό λήψης, γίνεται έλεγχος για παρουσία τυχόν λαθών με τη χρήση του Κυκλικού Ελέγχου Πλεονασμού (Cyclic Redundancy Check - CRC), ενός αλγορίθμου ανίχνευσης σφαλμάτων. Σε γενικές γραμμές, το τερματικό λήψεως στέλνει μήνυμα θετικής ή αρνητικής αναγνώρισης, ανάλογα με το αποτέλεσμα της διαδικασίας ανίχνευσης σφάλματος. Σε περίπτωση που το μήνυμα αναγνώρισης χαθεί στη διαδρομή προς το τερματικό αποστολής, αυτό με τη σειρά του θα μεταδώσει το πλαίσιο μετά από ένα ορισμένο χρονικό διάστημα. Η διαδικασία αυτή συχνά αναφέρεται ως ARQ.

Σε γενικές γραμμές, η ARQ είναι καλύτερη για τη διόρθωση των σφαλμάτων ριπής (burst errors) επειδή αυτός ο τύπος της απομείωσης εμφανίζεται σε ένα μικρό ποσοστό πλαισίων με αποτέλεσμα να μην απαιτούνται πολλές αναμεταδόσεις. Λόγω της ανατροφοδότησης που συνδέεται με τα πρωτόκολλα ARQ, οι ζεύξεις μετάδοσης πρέπει να υποστηρίζουν εκ περιδρομής αμφίδρομες (half-duplex) ή πλήρως αμφίδρομες (full-duplex) μεταδόσεις. Εάν, για λόγους σκοπιμότητας, είναι διαθέσιμες απλού τύπου συνδέσεις, τότε η χρήση της τεχνικής ARQ είναι αδύνατη επειδή ο παραλήπτης δεν θα είναι σε θέση να ενημερώσει τον αποστολέα για την ύπαρξη σφαλμάτων στα πλαίσια δεδομένων που απεστάλησαν.

Τύποι ARQ:

I) Συνεχής ARQ

Σε αυτόν τον τύπο της ARQ, που συχνά αποκαλείται και ως Πρωτόκολλο Συρόμενου Παραθύρου (Sliding Window Protocol - SWP), το τερματικό αποστολής μεταδίδει πλαίσια συνεχώς μέχρι το τερματικό λήψης να ανιχνεύσει κάποιο σφάλμα. Το τερματικό αποστολής είναι συνήθως σε θέση να μεταδίδει ένα συγκεκριμένο αριθμό πλαισίων και διατηρεί έναν πίνακα που δείχνει ποια πλαίσια έχουν αποσταλεί.

Ο σχεδιαστής του διαδικτυακού συστήματος μπορεί να καθορίσει τον αριθμό των πλαισίων που αποστέλλονται πριν από τη διακοπή, μέσω των παραμέτρων διαμόρφωσης των δικτυακών συσκευών. Εάν ένας δέκτης ανιχνεύσει ένα πλαίσιο που περιέχει σφάλματα, στέλνει μια αρνητική επιβεβαίωση στο τερματικό αποστολής ζητώντας την επαναποστολή του συγκεκριμένου πλαισίου. Τη στιγμή που το τερματικό αποστολής λάβει το αίτημα επαναποστολής του πλαισίου, υπάρχει περίπτωση να έχει ήδη αποστείλει αρκετά επόμενα πλαίσια, λόγω της ύπαρξης καθυστερήσεων διάδοσης μεταξύ του αποστολέα και του παραλήπτη. Ως εκ τούτου, το τερματικό αποστολής θα πρέπει να γυρίσει πίσω και να αναμεταδώσει το πλαίσιο δεδομένων που περιείχε σφάλμα.

Ένα τερματικό αποστολής μπορεί να στείλει εκ νέου πλαίσια με χρήση της συνεχούς ARQ, με δύο τρόπους. Μια μέθοδος είναι η πηγή να ανακτήσει το πλαίσιο που περιέχει σφάλματα από την προσωρινή μνήμη (buffer) εκπομπής και να το αποστείλει μαζί με όλα τα πλαίσια που το ακολουθούν. Η τεχνική αυτή ονομάζεται Go-Back-N (GBN). Η μέθοδος αυτή όμως γίνεται αναποτελεσματική όταν το N (ο αριθμός των πλαισίων που αποστέλλει το τερματικό αποστολής μετά το πλαίσιο που περιέχει το σφάλμα, συν ένα) γίνεται μεγάλο. Αυτό συμβαίνει επειδή η αναμετάδοση, έστω και ενός μόνο πλαισίου, σημαίνει και την αναμετάδοση ενός μεγάλου αριθμού πλαισίων τα οποία είχαν ήδη αποσταλεί χωρίς σφάλματα, κάτι που μειώνει την απόδοση (throughput) του δικτύου.

Η τεχνική GBN είναι χρήσιμη σε εφαρμογές για τις οποίες η χωρητικότητα του buffer του τερματικού λήψης είναι περιορισμένη. Λόγω του ότι τα πλαίσια πρέπει να παραδίδονται με τη σειρά, η χωρητικότητα αυτή πρέπει να είναι τόση ώστε να χωράει τουλάχιστον ένα πλαίσιο. Όταν ο κόμβος λήψης απορρίπτει ένα πλαίσιο με σφάλματα (στέλνει μια αρνητική επιβεβαίωση), δεν χρειάζεται να αποθηκεύσει τυχόν μεταγενέστερα πλαίσια για ενδεχόμενη αναδιάταξη ενώ είναι σε αναμονή για την αναμετάδοση, επειδή όλα τα επόμενα πλαίσια θα επαναποσταλούν εκ νέου.

Μια εναλλακτική λύση της συνεχούς GBN τεχνικής είναι η επιλεκτική αναμετάδοση μόνο του πλαισίου που περιείχε σφάλματα και στη συνέχεια η επαναφορά στην κανονική μετάδοση από το σημείο ακριβώς πριν την κοινοποίηση της ύπαρξης σφάλματος στο πλαίσιο αυτό. Αυτή η προσέγγιση ονομάζεται επιλεκτική επανάληψη (selective repeat). Είναι προφανώς καλύτερη από τη συνεχή GBN, όσον αφορά την απόδοση του δικτύου, επειδή αναμεταδίδεται μόνο το πλαίσιο που

περιείχε σφάλματα. Με την τεχνική αυτή, ωστόσο, το τερματικό λήψης θα πρέπει να είναι ικανό να αποθηκεύει έναν αριθμό πλαισίων, εάν πρόκειται να υποβληθούν σε επεξεργασία με τη σειρά, αφού η όποια επεξεργασία του συνόλου των δεδομένων θα πρέπει να ξεκινήσει μετά την επαναποστολή του πλαισίου που περιείχε σφάλματα.

II) Stop & Wait ARQ

Με τη μέθοδο αυτή, το τερματικό αποστολής μεταδίδει ένα πλαίσιο, στη συνέχεια σταματά και περιμένει για κάποιο είδος αναγνώριση από το δέκτη σχετικά με το αν το συγκεκριμένο πλαίσιο ήταν αποδεκτό ή όχι. Εάν το τερματικό λήψης στείλει μια αρνητική επιβεβαίωση, το πλαίσιο θα πρέπει να σταλεί ξανά. Το επόμενο πλαίσιο θα αποσταλεί μόνο σε περίπτωση θετικής επιβεβαίωσης από το τερματικό λήψης.

Ένα πλεονέκτημα της stop-and-wait ARQ είναι ότι δεν απαιτεί πολύ χώρο στα buffer των τερματικών αποστολής ή λήψης. Το τερματικό αποστολής πρέπει να αποθηκεύσει μόνο το τρέχον μεταδιδόμενο πλαίσιο. Ωστόσο, η stop-and-wait ARQ καθίσταται αναποτελεσματική, καθώς η καθυστέρηση μετάδοσης δεδομένων γίνεται μεγάλη. Το πρόβλημα είναι ότι με μεγαλύτερα πλαίσια, η πιθανότητα ύπαρξης σφάλματος είναι μεγαλύτερη. Έτσι, η αναμετάδοση θα συμβαίνει συχνότερα, γεγονός που μειώνει την προκύπτουσα απόδοση.

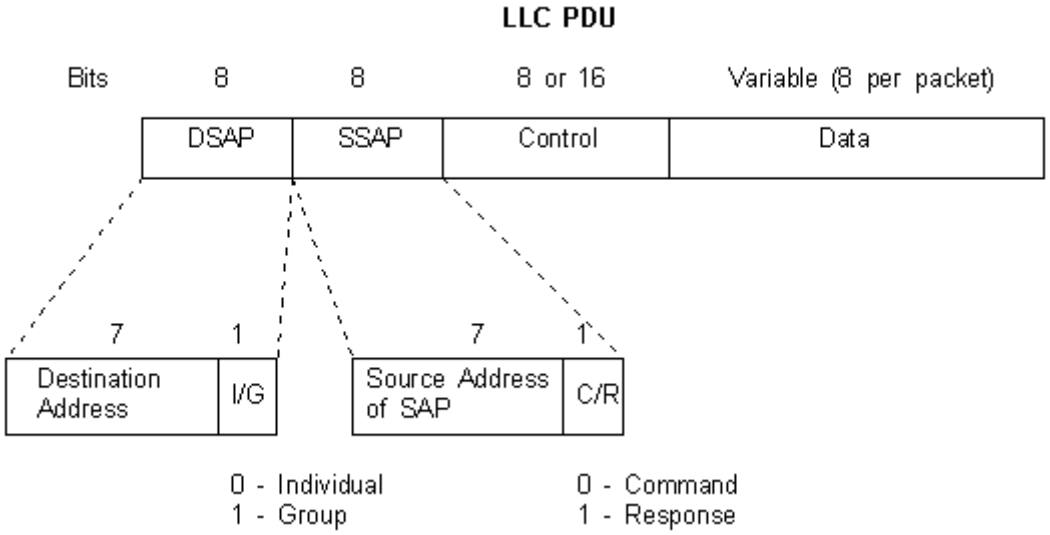
3.4.3 Επιβεβαιωμένη ασυνδεσμική υπηρεσία

Όπως συμβαίνει και με την μη επιβεβαιωμένη ασυνδεσμική υπηρεσία, η επιβεβαιωμένη ασυνδεσμική υπηρεσία (acknowledged connectionless service) ή υπηρεσία τύπου 3 (LLC3) δεν συνεπάγεται τη δημιουργία μιας λογικής σύνδεσης μεταξύ των τερματικών ενός δικτύου. Η διαφορά των δύο υπηρεσιών έγκειται στην επιβεβαίωση της επιτυχούς παράδοσης των πακέτων δεδομένων, που παρέχει η επιβεβαιωμένη ασυνδεσμική υπηρεσία. Ο έλεγχος ροής και σφάλματος, στην υπηρεσία αυτή αντιμετωπίζεται με τη χρήση της μεθόδου stop-and-wait ARQ.

Η επιβεβαιωμένη ασυνδεσμική υπηρεσία είναι χρήσιμη σε διάφορες εφαρμογές. Η συνδεσμική υπηρεσία πρέπει να διατηρεί έναν πίνακα για κάθε ενεργή σύνδεση με σκοπό την παρακολούθηση της κατάστασής της. Σε περίπτωση όμως που μια εφαρμογή απαιτεί εγγυημένη παράδοση, αλλά ο αριθμός των προορισμών που χρειάζεται να λάβουν τα δεδομένα είναι μεγάλος, τότε η συνδεσμική υπηρεσία μπορεί να μην είναι πρακτική λόγω του μεγάλου αριθμού των πινάκων που απαιτούνται. Παραδείγματα που ταιριάζουν με αυτό το σενάριο περιλαμβάνουν τον έλεγχο μιας διαδικασίας καθώς και αυτοματοποιημένα περιβάλλοντα εργοστασίων που απαιτούν την επικοινωνία μιας κεντρικής θέσης με έναν μεγάλο αριθμό επεξεργαστών και προγραμματιζόμενων ελεγκτών. Επιπλέον, η διαχείριση σημαντικών και χρονικά κρίσιμων σημάτων συναγερμού ή ελέγχου έκτακτης ανάγκης σε ένα εργοστάσιο, μπορεί να ταιριάξει επίσης στο σενάριο αυτό. Σε όλα αυτά τα παραδείγματα, το τερματικό αποστολής χρειάζεται επιβεβαίωση για να εξασφαλίσει την επιτυχή παράδοση των δεδομένων. Ωστόσο, στις περιπτώσεις αυτές, μια επείγουσα μετάδοση δεν μπορεί να περιμένει τη δημιουργία μιας σύνδεσης.

3.5 Μονάδα δεδομένων πρωτοκόλλου LLC

Μια μονάδα δεδομένων πρωτοκόλλου (PDU) αποτελείται από μια αλληλουχία συνεχόμενων οκτάδων που παραδίδονται ως μία μονάδα στο κατώτερο γειτονικό επίπεδο ή λαμβάνονται ως μία μονάδα από το κατώτερο γειτονικό επίπεδο [29]. Η PDU του πρωτοκόλλου LLC αποτελείται από τέσσερα πεδία, όπως ήδη αναφέρθηκε σε προηγούμενη ενότητα και παρουσιάζεται στην εικόνα 9. Στην εικόνα 10, παρουσιάζεται μια πιο λεπτομερής μορφή της PDU του πρωτοκόλλου LLC.



Εικόνα 10: Λεπτομερής μορφή LLC PDU

Η PDU του πρωτοκόλλου LLC περιέχει τα εξής πεδία:

- Σημείου Πρόσβασης Υπηρεσίας Προορισμού (Destination Service Access Point – DSAP)
- Σημείου Πρόσβασης Υπηρεσίας Πηγής (Source Service Access Point – SSAP)
- Ελέγχου (Control)
- Πληροφοριών (Information)

3.5.1 Πεδίο σημείου πρόσβασης υπηρεσίας προορισμού

Το πεδίο Σημείου Πρόσβασης Υπηρεσίας Προορισμού (Destination Service Access Point – DSAP) προσδιορίζει το σημείο πρόσβασης υπηρεσίας στο οποίο πρόκειται να αποσταλεί το πεδίο πληροφοριών του LLC. Αποτελείται από 6 bit διεύθυνσης, ένα bit χρήστη (User - U) και ένα bit ατόμου/ομάδας (Individual/Group – I/G) (Εικ.10).

Το bit χρήστη (U) ορίζει το στοιχείο LLC διαχείρισης ζεύξης και δεν πρέπει να χρησιμοποιείται για τον προσδιορισμό κανενός σημείου πρόσβασης υπηρεσίας. Με πιο απλά λόγια, υποδεικνύει εάν η διεύθυνση καθορίζεται από την IEEE (σε αυτή την περίπτωση παίρνει την τιμή 1) ή καθορίζεται από το χρήστη (σε αυτή την περίπτωση παίρνει την τιμή 0).

Το bit (I/G) δείχνει αν η SAP είναι μια διεύθυνση ομάδας (σε αυτή την περίπτωση παίρνει την τιμή 1) ή είναι ατομική διεύθυνση (σε αυτή την περίπτωση παίρνει την τιμή 0).

3.5.2 Πεδίο σημείου πρόσβασης υπηρεσίας πηγής

Το πεδίο Σημείου Πρόσβασης Υπηρεσίας Πηγής (Source Service Access Point – SSAP) προσδιορίζει το σημείο πρόσβασης υπηρεσίας από το οποίο προέρχεται το πεδίο πληροφοριών του LLC. Αποτελείται από 6 bit διεύθυνσης, ένα bit χρήστη (User - U) και ένα bit εντολής/απόκρισης (Command/Response – C/R) (Εικ.10).

Το bit χρήστη (U) ορίζει το στοιχείο LLC διαχείρισης ζεύξης και δεν πρέπει να χρησιμοποιείται για τον προσδιορισμό κανενός σημείου πρόσβασης υπηρεσίας. Με πιο απλά λόγια, υποδεικνύει εάν η διεύθυνση καθορίζεται από την IEEE (σε αυτή την περίπτωση παίρνει την τιμή 1) ή καθορίζεται από το χρήστη (σε αυτή την περίπτωση παίρνει την τιμή 0).

Το bit (C/R) δείχνει αν η PDU αποτελεί εντολή ή απόκριση (παίρνει την τιμή 0 για εντολή και την τιμή 1 για απόκριση). Στην περίπτωση λήψης μιας PDU, το bit C/R δεν θεωρείται μέρος του πεδίου SSAP και για το λόγο αυτό, ως επί το πλείστον, το SSAP θεωρείται ότι αποτελείται από 7 bit.

3.5.3 Πεδίο ελέγχου

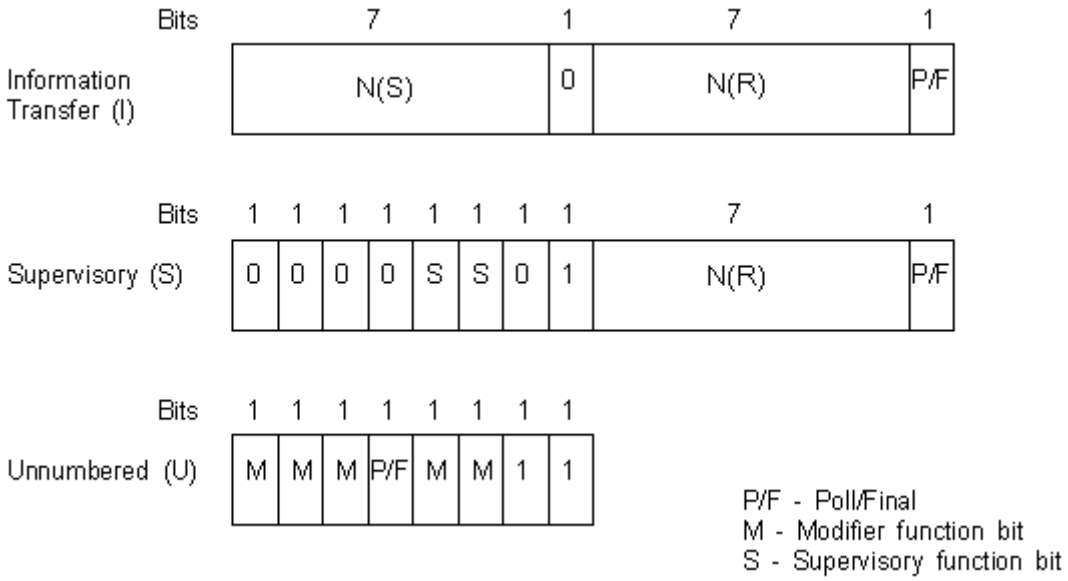
Η μορφή του πεδίου ελέγχου (Control) είναι ταυτόσημη με εκείνη του HDLC και υπάρχει μόνο στις PDU που περιέχουν αριθμούς ακολουθίας, στις θέσεις που παρουσιάζονται στην εικόνα 11.

Το πεδίο ελέγχου διαιρείται πάντοτε σε δύο υποπεδία :

- ένα υποπεδίο 8-bit που περιέχει τον απεσταλμένο αριθμό ακολουθίας N(S) που κωδικοποιείται στα πιο σημαντικά bit και χρησιμοποιείται με το πρωτόκολλο συρόμενου παραθύρου και για επιβεβαίωση των PDU
- ένα υποπεδίο 8-bit που περιέχει τον λαμβανόμενο αριθμό ακολουθίας N(R) που κωδικοποιείται στα λιγότερο σημαντικά bit και χρησιμοποιείται με το πρωτόκολλο συρόμενου παραθύρου και για επιβεβαίωση των PDU

Το πεδίο ελέγχου της PDU περιέχει bits που δείχνουν αν το πλαίσιο είναι ένας από τους ακόλουθους τύπους (Εικ.11) [29]:

- *Πληροφοριών (Information – I-format PDU)*: Χρησιμοποιείται για τη μεταφορά δεδομένων χρήστη σε λειτουργία υπηρεσίας LLC2
- *Εποπτικό (Supervisory – S-format PDU)*: Χρησιμοποιείται για τον έλεγχο της ροής και τον έλεγχο σφαλμάτων σε λειτουργία υπηρεσίας LLC2
- *Μη αριθμημένο (Unnumbered – U-format PDU)*: Χρησιμοποιείται για την εκτέλεση διαφόρων PDU ελέγχου πρωτοκόλλου σε λειτουργία υπηρεσίας LLC1



Εικόνα 11: Τύποι πεδίου ελέγχου πρωτοκόλλου LLC

Ι) Πεδίο ελέγχου PDU τύπου (I)

Το πεδίο ελέγχου της PDU τύπου (I) χρησιμοποιείται για τη μεταφορά πληροφοριών ή δεδομένων μεταξύ πηγής και προορισμού σε λειτουργία τύπου 2 (LLC2 - συνδεσμική υπηρεσία). Είναι η μόνη PDU του πρωτοκόλλου LLC που επιτρέπεται να μεταφέρει πληροφορίες σε μια λειτουργία τύπου 2.

Το τελευταίο bit του υποπεδίου N(S) παίρνει την τιμή 0, για να ορίσει ότι η PDU έχει πεδίο ελέγχου τύπου (I).

Το τελευταίο bit του υποπεδίου N(R) αποτελεί το P/F (Poll/Final) bit, το οποίο χρησιμοποιείται από την πηγή για τη ζήτηση μιας απόκρισης από τον προορισμό και από τον προορισμό για την απόκριση στην ζήτηση της πηγής.

Τα N(R) bit είναι γνωστά και με την ορολογία "Επιβεβαίωση Piggyback", επειδή η απόκριση επιβεβαιώνεται μαζί με τη μεταφορά των δεδομένων.

Στο πεδίο ελέγχου PDU τύπου (Information), δεν υπάρχουν εντολές και αποκρίσεις, αλλά μόνο μεταφορά δεδομένων.

ΙΙ) Πεδίο ελέγχου PDU τύπου (S)

Το πεδίο ελέγχου της PDU τύπου (S) χρησιμοποιείται για λειτουργίες εποπτικού ελέγχου του επιπέδου δεδομένων (handshaking) καθώς επίσης και για την επιβεβαίωση των πεδίων ελέγχου PDU τύπου (I) που ζητούν επαναποστολή ή προσωρινή παύση αποστολής δεδομένων.

Τα δύο τελευταία bit του υποπεδίου N(S) παίρνουν τις τιμές 01, για να ορίσουν ότι η PDU έχει πεδίο ελέγχου τύπου (S).

Τα δύο bit του υποπεδίου N(S) που σημειώνονται ως S στην εικόνα 11 είναι λειτουργικά bit και καθορίζουν τον σκοπό του πεδίου ελέγχου.

Τα πρώτα τέσσερα bit του υποπεδίου N(S) είναι δεσμευμένα και παίρνουν πάντα την τιμή 0.

Το τελευταίο bit του υποπεδίου N(R) αποτελεί το P/F (Poll/Final) bit, το οποίο χρησιμοποιείται από την πηγή για τη ζήτηση μιας απόκρισης από τον προορισμό και από τον προορισμό για την απόκριση στην ζήτηση της πηγής.

Πίνακας 3: Εντολές & αποκρίσεις των PDU του LLC

Υπηρεσία	Εντολή	Απόκριση	Τύπος πεδίου ελέγχου PDU	Τιμή πεδίου ελέγχου (HEX)
LLC1	Unnumbered Information (UI)		Unnumbered (U)	03
	Exchange Identification (XI)	Exchange Identification (XI)	Unnumbered (U)	AF, BF
	Test (TEST)	Test (TEST)	Unnumbered (U)	E, F3
LLC2	Information (I)	Information (I)	Information (I)	00 00 ως FE FF
	Receiver Ready (RR)	Receiver Ready (RR)	Supervisory (S)	01 00 ως 01 FF
	Receiver Not Ready (RNR)	Receiver Not Ready (RNR)	Supervisory (S)	05 00 ως 05 FF
	Reject (REJ)	Reject (REJ)	Supervisory (S)	09 00 ως 09 FF
	Set Asynchronous Balance Mode Extended (SABME)	Unnumbered Acknowledgement (UA)	Unnumbered (U)	6F, 7F (SABME) και 63, 73 (UA)
	Disconnect (DISC)	Disconnected Mode (DM)	Unnumbered (U)	43, 53 (DISC) και 0F, 1F (DM)
		Frame Reject (FRMR)	Unnumbered (U)	87, 97
LLC3	Ack Connectionless, seq 0 (AC0)	Ack Connectionless, seq 0 (AC0)	Unnumbered (U)	67, F7
	Ack Connectionless, seq 0 (AC1)	Ack Connectionless, seq 0 (AC1)	Unnumbered (U)	E7, F7

Στο πεδίο ελέγχου PDU τύπου (S), υπάρχουν οι ακόλουθες 3 εντολές και αποκρίσεις. Οι εντολές αυτές δεν περιέχουν κάποιο πεδίο πληροφοριών και για το λόγο αυτό δεν αυξάνουν τον αριθμό της ακολουθίας δεδομένων που αποστέλλεται [30]:

- *RR (Receiver Ready)*: Το πλαίσιο RR χρησιμοποιείται όταν ένα τερματικό αποστολής επιβεβαιώνει την αποστολή προηγούμενων δεδομένων και δεν διαθέτει νέα δεδομένα για αποστολή. Στην περίπτωση αυτή το υποπεδίο N(R) συνοδεύεται από την εντολή RR η οποία παίρνει την τιμή που φαίνεται στον πίνακα 3.
- *RNR (Receive Not Ready)*: Εάν ένα τερματικό αποστολής στείλει την εντολή RNR, σημαίνει ότι βρίσκεται σε μία προσωρινά απασχολημένη κατάσταση (busy condition) και δεν είναι σε θέση να δεχθεί I πλαίσια. Μετά από ένα σύντομο χρονικό διάστημα (μετά την αποστολή της εντολής RNR), εάν το τερματικό καταστεί έτοιμο να δεχτεί πλαίσια, θα αποστείλει την εντολή RR. Η εντολή RNR παίρνει την τιμή που φαίνεται στον πίνακα 3.
- *REJ (Reject)*: Μια εντολή REJ χρησιμοποιείται για να ζητηθεί η επαναποστολή ενός ή περισσότερων πλαισίων με αρχή το πλαίσιο που αναφέρεται στην εντολή REJ. Η εντολή REJ αποτελεί ένα ευγενικό αίτημα αναμετάδοσης δεδομένων που είχαν χαθεί κατά τη μεταφορά και παίρνει την τιμή που φαίνεται στον πίνακα 3.

III) Πεδίο ελέγχου PDU τύπου (U)

Στην περίπτωση που το πεδίο ελέγχου της PDU είναι τύπου (U), αποτελείται μόνο από το υποπεδίο N(S), τα τελευταία δύο bit του οποίου παίρνουν τις τιμές 11.

Το bit P/F (Poll/Final) χρησιμοποιείται από την πηγή για τη ζήτηση μιας απόκρισης από τον προορισμό και από τον προορισμό για την απόκριση στην ζήτηση της πηγής.

Τα υπόλοιπα bit, στην περίπτωση αυτή ορίζονται ως M (Modifier bits) και παίρνουν τιμές ανάλογα με τον τύπο λειτουργίας: Εντολή, Απόκριση ή Δεδομένα.

Το μη αριθμημένο πεδίο ελέγχου LLC χρησιμοποιείται ως επί το πλείστον σε ασυνδεσμική υπηρεσία. Οι PDU δεν είναι αριθμημένες, αποστέλλονται και αν όλα πάνε καλά ενδέχεται να φθάσουν στον προορισμό τους. Το πεδίο ελέγχου της PDU τύπου (U) μπορεί να είναι εντολές, αποκρίσεις ή δεδομένα και περιέχει μόνο 8 bit. Στο πεδίο ελέγχου PDU τύπου (U), υπάρχουν οι ακόλουθες 8 εντολές και αποκρίσεις:

- *UI (Unnumbered information)*: Η εντολή UI στέλνει πληροφορίες χωρίς καμία αναμενόμενη επιβεβαίωση. Αποτελεί ασυνδεσμική μορφή πλαισίου και ως εκ τούτου τα δεδομένα που περιέχονται σε ένα πλαίσιο UI μπορεί να χαθούν αν προκύψει κάποιο σφάλμα μετάδοσης. Η εντολή UI παίρνει την τιμή που φαίνεται στον πίνακα 3.
- *DISC (Disconnect)*: Όταν ένα τερματικό θέλει να τερματίσει μια σύνδεση, στέλνει μια εντολή DISC. Η εντολή ενημερώνει τον προορισμό ότι η πηγή αναστέλλει τη λειτουργία της σύνδεσης ζεύξης δεδομένων. Ο προορισμός θα πρέπει στη συνέχεια να μπει σε κατάσταση αποσύνδεσης. Στην περίπτωση λήψης μιας εντολής DISC, οι τυχόν εκκρεμείς επιβεβαιώσεις

ακυρώνονται αν υποτεθεί ότι το τερματικό αποστολής ανέστειλε τη λειτουργία του για δικούς του λόγους. Η εντολή DISC παίρνει την τιμή που φαίνεται στον πίνακα 3.

- *SABME (Set Asynchronous Balanced Mode Extended)*: Η εντολή αυτή χρησιμοποιείται για να καθοριστεί η σχέση μεταξύ δύο τερματικών LLC και παρουσιάζεται πάντα κατά την έναρξη μιας σύνδεσης τύπου LLC2. Σε περίπτωση λήψης μιας εντολής SABME από ένα τερματικό στο οποίο οφείλονται επιβεβαιώσεις (από μια προηγούμενη σύνδεση), οι εκκρεμείς επιβεβαιώσεις ακυρώνονται υποθέτοντας ότι το τερματικό αποστολής αντιμετώπισε κάποια κατάσταση που απαιτούσε τη δημιουργία μιας εντελώς νέας σύνδεσης. Η εντολή SABME παίρνει την τιμή που φαίνεται στον πίνακα 3.
- *XID (Exchange IDs)*: Η εντολή XID μεταφέρει τα είδη των υπηρεσιών LLC που υποστηρίζονται καθώς και το μέγεθος του παραθύρου λήψης. Η λήψη μιας εντολής XID προκαλεί μια αντίστοιχη απόκριση στο μικρότερο δυνατό χρόνο. Η μορφή της XID αποτελείται από ένα αναγνωριστικό των 8 bit που ακολουθείται από ένα πεδίο παραμέτρων των 16 bit. Ο IEEE έχει ορίσει μια «Βασική μορφή XID», η οποία καθορίζεται όταν το αναγνωριστικό παίρνει την τιμή 81 (hex). Στην περίπτωση αυτή, το πεδίο των παραμέτρων αποτελείται από ένα byte που ορίζεται ως το αναγνωριστικό του τύπου (LLC1 ή LLC2) και από ένα byte που περιέχει την τιμή του μεγέθους του παραθύρου λήψης. Το μέγεθος του παραθύρου λήψης καθορίζει τον μέγιστο αριθμό των byte που μπορούν να αποσταλούν σε ένα τερματικό χωρίς να λάβει κάποια επιβεβαίωση. Εάν γίνεται χρήση της βασικής μορφής XID και υπάρξει υπέρβαση στο όριο του μεγέθους του παραθύρου, τότε, το τερματικό στέλνει μια απόκριση FRMR. Η εντολή XID παίρνει την τιμή που φαίνεται στον πίνακα 3.
- *TEST (Έλεγχος της ζεύξης)*: Εάν το λογισμικό του συστήματος θέλει να ελέγξει τη ζεύξη LLC μπορεί να προκαλέσει την αποστολή μιας εντολής TEST. Ένα τερματικό που λαμβάνει μια εντολή TEST αναμένεται να στείλει απόκριση TEST στην πηγή το συντομότερο δυνατό. Στην απόκριση TEST συμπεριλαμβάνονται τυχόν δεδομένα ελέγχου της ζεύξης. Η εντολή TEST παίρνει την τιμή που φαίνεται στον πίνακα 3.
- *UA (Unnumbered Acknowledgement)*: Η εντολή UA αποτελεί μια μη αριθμημένη επιβεβαίωση και παίρνει την τιμή που φαίνεται στο πίνακα 3.
- *DM (Disconnect Mode)*: Η απόκριση DM χρησιμοποιείται για την αναφορά της κατάστασης ενός τερματικού που έχει εισέλθει σε λειτουργία αποσύνδεσης. Συνήθως εμφανίζεται μετά τη λήψη μιας εντολής DISC. Η απόκριση DM παίρνει την τιμή που φαίνεται στον πίνακα 3.
- *FRMR (Frame Reject)*: Η FRMR είναι μία από τις πιο παρεξηγημένες εντολές. Οι προδιαγραφές του 802.2 αναφέρουν ότι το πλαίσιο αυτό αποστέλλεται όταν υπάρχει μια κατάσταση η οποία δεν μπορεί να διορθωθεί με την εκ νέου αποστολή προηγουμένως απεσταλμένων

πλαισίων. Στην πραγματικότητα, μία FRMR δείχνει την ύπαρξη μιας καταστροφικής αποτυχίας. Η παρουσία FRMR πλαισίων, υπονοεί ενδεχόμενα σφάλματα στο λογισμικό του προγράμματος οδήγησης μιας συσκευής ή πιθανή βλάβη υλικού. Απολεσθέντα πλαίσια σε ένα δίκτυο δεν προκαλούν FRMR, αλλά μόνο REJ.

Τα γεγονότα που μπορεί να προκαλέσουν την μετάδοση ενός πλαισίου LLC FRMR είναι τα εξής:

- Η λήψη μιας άκυρης ή ανυλοποίητης εντολής ή απόκρισης, όπως μιας εποπτικής ή μη αριθμημένης εντολής με (I) πεδίο που δεν επιτρέπεται ή μιας απρόσμενης απόκρισης UA (όταν δεν έχει αποσταλεί κάποια εντολή)
- Το μήκος του πεδίου πληροφοριών υπερβαίνει το προηγούμενα καθορισμένο (με μια XID) μέγιστο μήκος για την εν λόγω σύνδεση
- Ένα τερματικό στέλνει μια επιβεβαίωση για ένα πλαίσιο που δεν έχει διαβιβαστεί

3.5.4 Πεδίο πληροφοριών

Το πεδίο πληροφοριών περιέχει τις πληροφορίες από τα πρωτόκολλα ανωτέρων επιπέδων που το LLC μεταφέρει στον προορισμό. Αποτελείται από έναν ακέραιο αριθμό οκτάδων που έχει ως ανώτατο όριο τη Μέγιστη Μονάδα Πληροφοριών (Maximum Information Unit – MIU). Το πεδίο πληροφοριών μπορεί όμως να είναι ακόμα και κενό.

Στις τεχνικές προδιαγραφές του πρωτοκόλλου LLC προσδιορίζεται, για κάθε PDU, αν το πεδίο των πληροφοριών υπάρχει και το περιεχόμενό του [29].

3.6 Στοιχεία υπηρεσίας επιπέδων LLC/MAC

Στην αρχιτεκτονική 802 τα επίπεδα επικοινωνούν μεταξύ τους μέσω στοιχείων υπηρεσίας (service primitives) που έχουν τις ακόλουθες μορφές [23]:

- *Αίτημα (Request)*: Ένα επίπεδο χρησιμοποιεί αυτού του είδους το στοιχείο για να ζητήσει από ένα άλλο επίπεδο να εκτελέσει μια συγκεκριμένη υπηρεσία.
- *Επιβεβαίωση (Confirm)*: Ένα επίπεδο χρησιμοποιεί αυτού του είδους το στοιχείο για να μεταφέρει τα αποτελέσματα ενός προηγούμενου αιτήματος στοιχείου υπηρεσίας.
- *Ένδειξη (Indication)*: Ένα επίπεδο χρησιμοποιεί αυτού του είδους το στοιχείο για να δείξει σε άλλο επίπεδο ότι έχει συμβεί κάποιο σημαντικό γεγονός. Το στοιχείο αυτό μπορεί να προκύψει από ένα αίτημα υπηρεσίας ή από κάποιο εσωτερικά παραγόμενο γεγονός.
- *Απόκριση (Response)*: Ένα επίπεδο χρησιμοποιεί αυτού του είδους το στοιχείο για να ολοκληρώσει μια διαδικασία που ξεκίνησε από ένα στοιχείο ένδειξης.

Τα στοιχεία αυτά είναι ένας αφηρημένος τρόπος καθορισμού του πρωτοκόλλου και δεν προϋποθέτουν μια συγκεκριμένη μέθοδο φυσικής υλοποίησης. Κάθε επίπεδο εντός του μοντέλου 802 χρησιμοποιεί αντίστοιχα ειδικά στοιχεία.

Το επίπεδο LLC επικοινωνεί με το αντίστοιχό του MAC επίπεδο μέσω των ακόλουθων ειδικών συνόλων στοιχείων υπηρεσίας:

- *MA-UNITDATA.request*: Το επίπεδο LLC στέλνει το στοιχείο αυτό στο επίπεδο MAC για να ζητήσει τη μεταφορά ενός πλαισίου δεδομένων από μια τοπική οντότητα LLC σε μια συγκεκριμένη οντότητα ομότιμου (peer) LLC ή ομάδας ομότιμων οντοτήτων σε διαφορετικά τερματικά. Το πλαίσιο δεδομένων μπορεί να είναι ένα πλαίσιο πληροφοριών που περιέχει δεδομένα από ένα ανώτερο επίπεδο ή ένα πλαίσιο ελέγχου (όπως ένα εποπτικό ή μη αριθμημένο πλαίσιο) που το LLC δημιουργεί εσωτερικά για να επικοινωνεί με ομότιμα LLC.
- *MA-UNITDATA.indication*: Το επίπεδο MAC στέλνει το στοιχείο αυτό στο στρώμα LLC να μεταφέρει ένα πλαίσιο δεδομένων από το επίπεδο MAC στο LLC. Αυτό συμβαίνει μόνο όταν το MAC έχει βρει ότι ένα πλαίσιο που λαμβάνει από το φυσικό επίπεδο είναι έγκυρο, δεν έχει σφάλματα και η διεύθυνση προορισμού υποδεικνύει τη σωστή διεύθυνση MAC του τερματικού.
- *MA-UNITDATA-STATUS.indication*: Το επίπεδο MAC στέλνει το στοιχείο αυτό στο επίπεδο LLC για την παροχή πληροφοριών σχετικά με την κατάσταση της παρεχόμενης υπηρεσίας για ένα προηγούμενο MA-UNITDATA.request στοιχείο.

3.7 Πρωτόκολλο πρόσβασης υποδικτύου

Το Πρωτόκολλο Πρόσβασης Υποδικτύου (Subnet Access Protocol – SNAP) είναι μια επέκταση του IEEE 802.2 LLC με σκοπό τη διάκριση πολύ περισσότερων πρωτοκόλλων του ανώτερου επιπέδου από ότι η χρήση των 8-bit πεδίων SAP που υπάρχουν στην κεφαλίδα του IEEE802.2 [24]. Τα πεδία SNAP και SAP προστίθενται στα πακέτα στον κόμβο μετάδοσης προκειμένου να επιτραπεί στον κόμβο λήψης να περάσει κάθε λαμβανόμενο πλαίσιο στο κατάλληλο πρόγραμμα οδήγησης συσκευής που κατανοεί το αντίστοιχο δεδομένο πρωτόκολλο.

Υπάρχει η ειδική περίπτωση του SNAP, όπου τα DSAP και SSAP παίρνουν τις τιμές AA και το πεδίο ελέγχου έχει την τιμή 03. Στην περίπτωση αυτή, αμέσως μετά από το πεδίο ελέγχου και πριν από το πεδίο των πληροφοριών, υπάρχουν τρία byte που προορίζονται για το Οργανωτικά Μοναδικό Αναγνωριστικό (Organisational Unique Identifier - OUI) και στη συνέχεια άλλα δύο byte που προορίζονται για το Αναγνωριστικό Πρωτοκόλλου (Protocol Identifier - PID) (Εικ.12) [31].

802.2 LLC Header			SNAP extension		Upper layer data
DSAP	SSAP	Control	OUI	Protocol ID	
8 bits	8 bits	8 or 16 bits	24 bits	16 bits	multiple of 8 bits

Εικόνα 12: Μορφή PDU του LLC με επέκταση SNAP

Το SAP χρησιμοποιείται γενικά για τα 802.x συμβατά πρωτόκολλα, ωστόσο, η ιδέα του SNAP είναι να επιτραπεί σε μη συμμορφούμενα πρωτόκολλα IEEE να γίνουν ψευδο-συμμορφούμενα χωρίς την ανάγκη για εκ νέου εγγραφή του κώδικα των οδηγών (drivers) του δικτύου. Για παράδειγμα το αναγνωριστικό OUI με τιμή 00-00-00 δηλώνει ότι το πλαίσιο είναι ένα πλαίσιο Ethernet και όχι ένα πλαίσιο 802.3 που δεν έχει εκχωρηθεί σε κάποιο συγκεκριμένο προμηθευτή. Η δυνατότητα αυτή είναι πολύ χρήσιμη ιδιαίτερα στις περιπτώσεις που τα πλαίσια δεδομένων περνάνε από πολλά και διαφορετικά μέσα, καθώς, στη συγκεκριμένη περίπτωση, το πλαίσιο Ethernet, μετά το πέρασμά του από τα διάφορα μέσα, θα ανακατασκευαστεί ως πλαίσιο Ethernet και όχι ως κάτι άλλο.

3.8 Σύνοψη

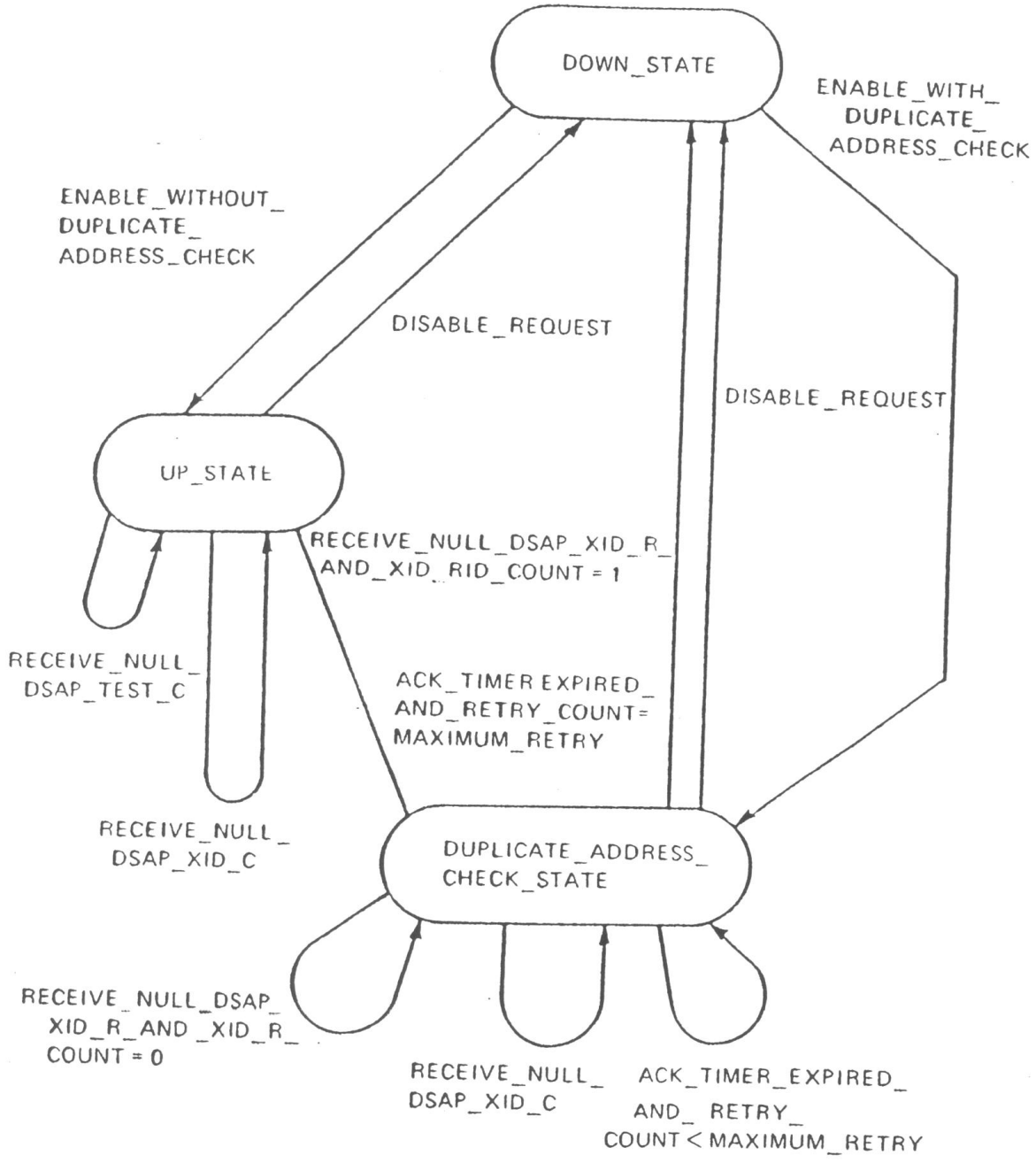
Το πρωτόκολλο IEEE 802.2 Ελέγχου Λογικής Ζεύξης (Logical Link Control – LLC) αποτελεί το ανώτερο στρώμα του επιπέδου ζεύξης δεδομένων του μοντέλου αναφοράς OSI. Σκοπός του είναι να παρέχει έναν τρόπο επικοινωνίας μεταξύ των ανωτέρων επιπέδων του μοντέλου αναφοράς OSI με οποιονδήποτε τύπο επιπέδου MAC και να ελέγχει τα δεδομένα ζεύξης, διαθέτοντας μηχανισμούς αρίθμησης πλαισίων, αρίθμησης αναγνωρίσεων (ACK), παραθύρου, ελέγχου ροής, κ.τ.λ.

Το LLC παρέχει τις ακόλουθες τρεις υπηρεσίες: συνδεσμική, επιβεβαιωμένη ασυνδεσμική και μη επιβεβαιωμένη ασυνδεσμική υπηρεσία. Υψηλότερα επίπεδα περνούν τα δεδομένα του χρήστη μέχρι το LLC, έχοντας εξασφαλισμένη μια μετάδοση χωρίς σφάλματα σε όλο το δίκτυο. Το LLC με τη σειρά του προσθέτει μια κεφαλίδα ελέγχου, δημιουργώντας μια μονάδα δεδομένων πρωτοκόλλου LLC, την PDU, και χρησιμοποιεί την πληροφορία ελέγχου στη λειτουργία του πρωτοκόλλου LLC.

Η PDU περιλαμβάνει τέσσερα πεδία: τα πεδία Σημείου Πρόσβασης Υπηρεσίας Προορισμού (DSAP), Σημείου Πρόσβασης Υπηρεσίας Πηγής (SSAP), Ελέγχου και Πληροφοριών. Τα πεδία DSAP και SSAP επιτρέπουν την πολυπλεξία με το LLC, διαφόρων πρωτοκόλλων ανωτέρου επιπέδου. Ωστόσο, πολλά πρωτόκολλα χρησιμοποιούν την επέκταση Πρωτοκόλλου Πρόσβασης Υποδικτύου (SNAP), η οποία επιτρέπει τη χρήση των τιμών EtherType για τον καθορισμό των πρωτοκόλλων που μεταφέρονται προς την κορυφή του IEEE 802.2.

4 Γενική περιγραφή της εφαρμογής

Η εφαρμογή είναι μια εφαρμογή πελάτη-εξυπηρετητή, η οποία εκτελείται σε έναν υπολογιστή.



Πρέπει να επισημανθεί ότι η γενική μορφή του κώδικα μοιάζει σε μια μηχανή κατάστασης, εφόσον υπάρχουν τρεις διαφορετικές καταστάσεις:

- Κατάσταση *«DOWN_STATE»:* είναι η κατάσταση λειτουργίας όταν υπάρχει κάποιο σφάλμα απροσδιόριστης προέλευσης.

```
DOWN_STATE:
    if(ENABLE_WITH_DUPLICATE_ADDRESS_CHECK) {
    xid.uh.dsap        = NULL_DSAP;
    xid.uh.ssap        = NULL_SSAP_C;
    xid.uh.command = XID_com;
    xid.xi.format      = IEEE_Basic;
    xid.xi.type  = LLC_type_I;
    xid.xi.window      = WINDOW_size
    if(sendto(s,xid,sizeof(struct
xid),0,&newaddr_in,sizeof(struct sockaddr_in)==-1) {
            fprintf(sdterr, "unable to send request\n");
            exit(1);
    }
    signal(SIGALRM, handler);
    alarm(5);
    RETRY_COUNT=0;
    XID_R_COUNT=0;
    goto DUPLICATE_ADDRESS_CHECK_STATE;
    }
    if(ENABLE_WITHOUT_DUPLICATE_ADDRESS_CHECK){
    REPORT_STATUS(STATION_UP);
    goto UP_STATE;
    }
```

- Κατάσταση *«UP_STATE»:* είναι η κατάσταση της ορθής λειτουργίας της εφαρμογής.

```
UP_STATE:
    if(DISABLE_REQUEST){
    REPORT_STATUS(STATION_DOWN);
    goto DOWN_STATE;
    }
```

```
        if(recv(s,in_UI,sizeof(struct test),0)==-1) {
        fprintf(stderr, "unable to receive response");
        exit(1);
        }       else   {
        if(what_came=2) {
                xid.uh.dsap         = in_UI.uh.ssap;
                xid.uh.ssap         = NULL_SSAP_R;
                xid.uh.command = XID_com;
                xid.xi.format       = IEEE_Basic;
                xid.xi.type  = LLC_type_I;
                xid.xi.window       = WINDOW_size;
                if(sendto(s,xid,sizeof(struct
xid),0,&newaddr_in,sizeof(struct sockaddr_in)==-1) {
                        fprintf(sdterr, "unable to send request\n");
                        exit(1);
                goto UP_STATE;
        }
        }
        if(what_came=3) {
        test.uh.dsap                = in_UI.uh.ssap;
        test.uh.ssap                = NULL_SSAP_R;
        test.uh.command         = XID_com;
        test.info[]       = in_UI.info[];
        if(sendto(s,test,sizeof(struct
test),0,&newaddr_in,sizeof(struct sockaddr_in)==-1) {
                fprintf(sdterr, "unable to send request\n");
                exit(1);
        goto UP_STATE;
        }
```

- Κατάσταση *«DUPLICATE_ADDRESS_CHECK_STATE».*

```
    DUPLICATE_ADDRESS_CHECK_STATE:
        if(recv(s,in_UI,sizeof(struct test),0)==-1) {
        fprintf(stderr, "unable to receive response");
        exit(1);
        }       else   {
        if(what_came=2 && XID_R_COUNT=0) {
                XID_R_COUNT = XID_R_COUNT + 1;
                goto DUPLICATE_ADDRESS_CHECK_STATE;
        }
        if(what_came=2 && XID_R_COUNT=1) {
                REPORT_STATUS(DUPLICATE_ADDRESS_FOUND);
                goto DOWN_STATE;
        }
        if(what_came=1) {       /*
        RECEIVE_NULL_DSAP_XID_C */
        xid.uh.dsap         = NULL_DSAP;
        xid.uh.ssap         = NULL_SSAP_R;
        xid.uh.command = XID_com;
        xid.xi.format       = IEEE_Basic;
        xid.xi.type  = LLC_type_I;
        xid.xi.window       = WINDOW_size;
                if(sendto(s,xid,sizeof(struct
xid),0,&newaddr_in,sizeof(struct sockaddr_in)==-1) {
                        fprintf(sdterr, "unable to send request\n");
                        exit(1);
                goto DUPLICATE_ADDRESS_CHECK_STATE;
        }
        if(errno==EINTR && RETRY_COUNT<MAXIMUM_RETRY)
        {
```

```
            if(sendto(s,xid,sizeof(struct
xid),0,&newaddr_in,sizeof(struct sockaddr_in)==-1){
                  fprintf(sdterr, "unable to send request\n");
                  exit(1);
            }
            signal(SIGALRM, handler);
            alarm(5);

            RETRY_COUNT=RETRY_COUNT+1;
            XID_R_COUNT=0;
            goto DUPLICATE_ADDRESS_CHECK_STATE;
      }
      if(errno==EINTR && RETRY_COUNT=MAXIMUM_RETRY)
{
            REPORT_STATUS(STATION_UP);
            goto STATION_UP;
      }
      if(DISABLE_REQUEST){
            REPORT_STATUS(STATION_DOWN);
            goto DOWN_STATE;
      }
      }
```

4.1 Κύρια ρουτίνα

η κύρια ρουτίνα εγκαθιστά την επικοινωνία μεταξύ των απόμακρων μερών και διαχειρίζεται την λειτουργία της μηχανής κατάστασης με χρήση ατέρμονα κύκλου. Σε περίπτωση σφάλματος, οπότε και θα σπάσει ο κύκλος, δημιουργείται alarm. Στη ρουτίνα ορίζονται οι τοπικές μεταβλητές και λαμβάνει χώρα η δέσμευση μνήμης

```
   main()
   int RETRY_COUNT;
   int XID_R_COUNT;
   struct test in_UI;
```

```
    {
    memset ((char *)&myaddr_in,0,sizeof(struct
sockaddr_in));
    memset ((char *)&newaddr_in,0,sizeof(struct
sockaddr_in));
    myaddr_in.sin_family = AF_INET;
    myaddr_in.sin_port = 0;
    myaddr_in.sin_addr.s_addr = INADDR_ANY;
    newaddr_in = myaddr_in;
    s = socket(AF_INET,SOCK_DGRAM,0);
    if(s==-1) {
        fprintf(sdterr,"unable to create socket\n");
        exit(1);
    }
    if(bind(s,&myaddr_in,sizeof(struct sockaddr_in))==-1) {
        fprintf(sdterr,"unable to bindsocket\n");
        exit(1);
    }
    …
        _ΕΔΩ_ΠΕΡΙΛΑΜΒΑΝΕΤΑΙ_Η_ΜΗΧΑΝΗ_ΚΑΤΑΣΤΑΣΗΣ_
    …
        alarm(0);
    }
```

4.1.1 Πλαίσια πρωτοκόλλου

Η λειτουργία του πρωτοκόλλου βασίζεται σε πλαίσια. Η δομή των πλαισίων ορίστηκε ως ακολούθως:

```
    struct UI_HEADER {
        unsigned short int dsap;
        unsigned short int ssap;
        unsigned short int command;
```

```
    };

struct XID_INFO {
    unsigned short int format;
    unsigned short int type;
    unsigned short int window;
    };
union xid {
    struct UI_HEADER uh;
    struct XID_INFO xi;
    };
```

Η δομή των πλαισίων παρουσιάζεται σχηματικά ακολούθως:

```
/* XID format:                                                    *
*   <lsb (least significant bit)                                 *
*  I/G D D D D D D D          Individual(=0)/Group
(=1) DSAP              *
*  C/R S S S S S S S          Command(=0)
/Response(=1) SSAP            *
*   1 1 1 1 P 1 0 1       Poll(=0/1)    XID    command
[Resp.:Finall(=0/1)] *
*   X X X X X X X X       Format Identifier (10000001:
IEEE basic)      *
*   Y Y Y Y Y Z Z Z       Y: LLC Type            (10000:
class I LLC)      Z=0    *
*   Z W W W W W W W       Z=0       W:   Winndow
size                  *
*                    msb>
```

Παράδειγμα των δυνατών τιμών που μπορούν να έχουν τα πλαίσια είναι:

NULL_DSAP_XID_C = 00000000 00000000 1111P101
10000001 10000000 0wwwwwww

NULL_DSAP_XID_R = 00000000 10000000 1111P101
10000001 10000000 0wwwwwww

NULL_DSAP_TEST_R = 00000000 10000000 1100P111

4.2 Πίνακας Μετάβασης Καταστάσεων του Station Component

Current State	Event	Action(s)	Next State
DOWN_STATE	[ENABLE_WITH_DUPLICATE_ ADDRESS_CHECK]	SEND_NULL_DSAP_XID_C START_ACK_TIMER RETRY_COUNT:=0 XID_R_COUNT:=0	DUPLICATE_ ADDRESS_ CHECK_STATE
	[ENABLE_WITHOUT_ DUPLICATE_ ADDRESS_CHECK	REPORT_STATUS(STATION_UP)	UP_STATE
UP_STATE	DISABLE_REQUEST	REPORT_STATUS(STATION_DOWN)	DOWN_STATE
	RECEIVE_NULL_DSAP_XID_C	SEND_XID_R	UP_STATE
	RECEIVE_NULL_DSAP_TEST_C	SEND_TEST_R	UP_STATE
DUPLICATE_ ADDRESS_ CHECK_STATE (OPTIONAL)	[RECEIVE_NULL_DSAP_XID_ R_AND_XID_R_COUNT=0]	XID_R_COUNT:= XID_R_COUNT+1	DUPLICATE_ ADDRESS_ CHECK_STATE
	[RECEIVE_NULL_DSAP_XID_ R_AND_XID_R_COUNT=1	REPORT_STATUS(DUPLICATE_ADDRESS_FOUND)	DOWN_STATE
	[RECEIVE_NULL_DSAP_XID_C]	SEND_XID_R	DUPLICATE_ ADDRESS CHECK_STATE
	[ACK_TIMER_EXPIRED_AND_ RETRY_COUNT<MAXIMUM_RETRY]	SEND_NULL_DSAP_XID_C START_ACK_TIMER RETRY_COUNT:=RETRY_ COUNT+1 XID_R_COUNT:=0	DUPLICATE_ ADDRESS_ CHECK_STATE
	[ACK_TIMER_EXPIRED_AND_ RETRY_COUNT=MAXIMUM_RETRY	REPORT_STATUS(STATION_UP)	UP_STATE
	[DISABLE_REQUEST]	REPORT_STATUS(STATION_DOWN)	DOWN_STATE

4.2.1 Περιγραφή των Καταστάσεων του Station Component

DOWN _STATE : Το Station component είναι εκτός λειτουργίας ή απενεργοποιημένο.

DUPLICATE_ADDRESS_CHECK_STATE : Το Station component είναι σε διαδικασία ελέγχου για διπλές MAC διευθύνσεις στο LAN. Ο κύριος σκοπός αυτής της κατάστασης είναι να προσδιορίσει το LLC Station component ότι η MAC διεύθυνση του σταθμού είναι μοναδική στο τοπικό δίκτυο. Το Station component θα στείλει μια XID εντολή με ταυτόσημες DA και SA MAC διευθύνσεις και θα περιμένει για μια πιθανή XID απόκριση η οποία θα προσδιορίζει αν υπάρχει άλλος σταθμός με την ίδια MAC διεύθυνση.

UP_STATE : Το Station component είναι ενεργοποιημένο και λειτουργεί στο LAN. Το LLC επιτρέπει στα SAPs να ανταλλάσσουν LLC PDU's στο κανάλι.

4.2.2 Περιγραφή των Γεγονότων του Station Component

ENABLE_WITH_DUPLICATE_ADDRESS_CHECK : Ο χρήστης του Station Component το έχει ενεργοποιήσει και έχει κάνει αίτηση στο LLC να ελέγξει αν υπάρχει διπλή διεύθυνση του MAC SAP στο δίκτυο.

ENABLE_WITHOUT_DUPLICATE_ADDRESS_CHECK : Ο χρήστης του Station Component το έχει ενεργοποιήσει χωρίς να ελέγξει αν υπάρχει διπλή διεύθυνση του MAC SAP στο δίκτυο.

ACK_TIMER_EXPIRED_AND_RETRY_COUNT<MAXIMUM_RETRY : Το χρονόμετρο επιβεβαίωσης έχει λήξει και ο αριθμός αναμεταδόσεων είναι μικρότερος του μέγιστου ορίου.

ACK_TIMER_EXPIRED_AND_RETRY_COUNT=MAXIMUM_RETRY : Το χρονόμετρο επιβεβαίωσης έχει λήξει και ο αριθμός αναμεταδόσεων είναι ίσος του μέγιστου ορίου.

RECEIVE_NULL_DSAP_XID_C : Μια XID εντολή με NULL DSAP διεύθυνση έχει ληφθεί.

RECEIVE_NULL_DSAP_XID_R_AND_XID_R_COUNT=0 : Μια και μόνη XID εντολή με NULL DSAP διεύθυνση έχει ληφθεί.

RECEIVE_NULL_DSAP_XID_R_AND_XID_R_COUNT=1 : Μια δεύτερη XID εντολή με NULL DSAP διεύθυνση έχει ληφθεί.

RECEIVE_NULL_DSAP_TEST_C : Μια TEST εντολή με NULL DSAP διεύθυνση έχει ληφθεί.

DISABLE_REQUEST : Ο χρήστης έχει κάνει αίτηση απενεργοποίησης του Station Component.

4.2.3 Περιγραφή των Ενεργειών του Station Component

START_ACK_TIMER : Εκκίνηση του χρονομέτρου επιβεβαίωσης. Με αυτή τη διαδικασία ελέγχουμε αν έχει ληφθεί επιβεβαίωση από το απομακρυσμένο LLC σε μια συγκεκριμένη χρονική διάρκεια.

RETRY_COUNT:=0 : Αρχικοποίηση του αριθμού αναμεταδόσεων.

RETRY_COUNT:=RETRY_COUNT+1 : Αύξηση κατά ένα του αριθμού αναμεταδόσεων.

XID_R_COUNT:=0 : Αρχικοποίηση του μετρητή των XID αποκρίσεων.

XID_R_COUNT:=XID_R_COUNT+1 : Αύξηση κατά ένα του μετρητή των XID αποκρίσεων.

SEND_NULL_DSAP_XID_C : Το LLC θα στείλει μια XID εντολή με NULL SSAP και NULL DSAP διευθύνσεις και με ταυτόσημες DA και SA MAC διευθύνσεις.

SEND_XID_R : Το LLC θα στείλει μια XID απόκριση, χρησιμοποιώντας την SSAP διεύθυνση της XID εντολής ως την DSAP διεύθυνση της απόκρισης και χρησιμοποιώντας NULL SSAP διεύθυνση.

SEND_TEST_R : Το LLC θα στείλει μια TEST απόκριση, χρησιμοποιώντας την SSAP διεύθυνση της TEST εντολής ως την DSAP διεύθυνση της απόκρισης και χρησιμοποιώντας NULL SSAP διεύθυνση.

REPORT_STATUS : Το LLC είναι σε θέση να αναφέρει την κατάσταση της σύνδεσης. Τα έγκυρα αποτελέσματα είναι :

STATION_UP : Το LLC είναι σε λειτουργία.

STATION_DOWN : Το LLC δεν λειτουργεί.

DUPLICATE_ADDRESS_FOUND : Το LLC ανακάλυψε ένα άλλο LLC με ταυτόσημη MAC διεύθυνση με τη δική του.

4.3 SAP Component

Το SAP Component χειρίζεται όλη την κίνηση των LLC τύπου 1 PDUs για ένα συγκεκριμένο DSAP στο τοπικό Station Component. Ο τοπικός SAP χρήστης (user) είναι σε θέση να ενεργοποιεί και να απενεργοποιεί τη λειτουργία κάθε SAP στο Station Component. Από τη στιγμή που ενεργοποιηθεί το SAP Component μπορεί να επεξεργαστεί τύπου 1 LLC PDU's που προορίζονται για το DSAP και να μεταδίδει τύπου 1 LLC PDU's ως αποτέλεσμα αίτησης του SAP χρήστη ή ως αποτέλεσμα κάποιας ενέργειας του πρωτοκόλλου.

```
sap()
{
int i,lsn;
unsigned short int xid[5];
struct ldata lind,lreq;
struct mind mind;
struct mreq mreq;
struct mcon mcon;
```

```
    close(lsn);
    i = recv(sm,mind,sizeof(struct mind),0);
    if((i = -1) && (errno != EWOULDBLOCK))          exit(1);
    lind.l_addr  = mind.m_sdu[0];
    lind.r_addr  = mind.m_sdu[1];
    lind.s_class = mind.s_class;

    switch(mind.m_sdu[2]) {
    case UI_com:case UI_com_p:
       for(i=0;i<=(1496-3);i++)
lind.l_sdu[i]=mind.m_sdu[i+3];
       if(send(sn,lind,sizeof(struct ldata),0)==-1)     exit(1);
    case XID_com:case XID_com_p:
                                        /*     RECEIVE XID      */
       if(mind.m_sdu[1] & 1)  {
       lind.r_addr -= 1;
                          /*     XID RESPONSE*/
       for(i=0;i<=(5-3);i++)   lind.l_sdu[i]=mind.m_sdu[i+3];
       if(send(sn,lind,sizeof(struct ldata),0)==-1)     exit(1);
       }       else   {
       xid[0] = mind.m_sdu[1];
                                  /*     XID COMMAND    */
       if(mind.m_sdu[0] & 1) {              xid[1]                =
mind.m_sdu[0];                  /*     GROUP DSAP      */
       } else { xid[1] = mind.m_sdu[0]+1; }
       /*+1 to resp SSAP by inDIV DSAP */
       mreq.d_addr   = mind.s_addr;
       mreq.m_sdu[0] = mind.m_sdu[1];
       mreq.m_sdu[2] = mind.m_sdu[2];
       mreq.m_sdu[3] = IEEE_basic;
       mreq.m_sdu[4] = LLC_type_I;
       mreq.m_sdu[5] = WINDOW_size;
```

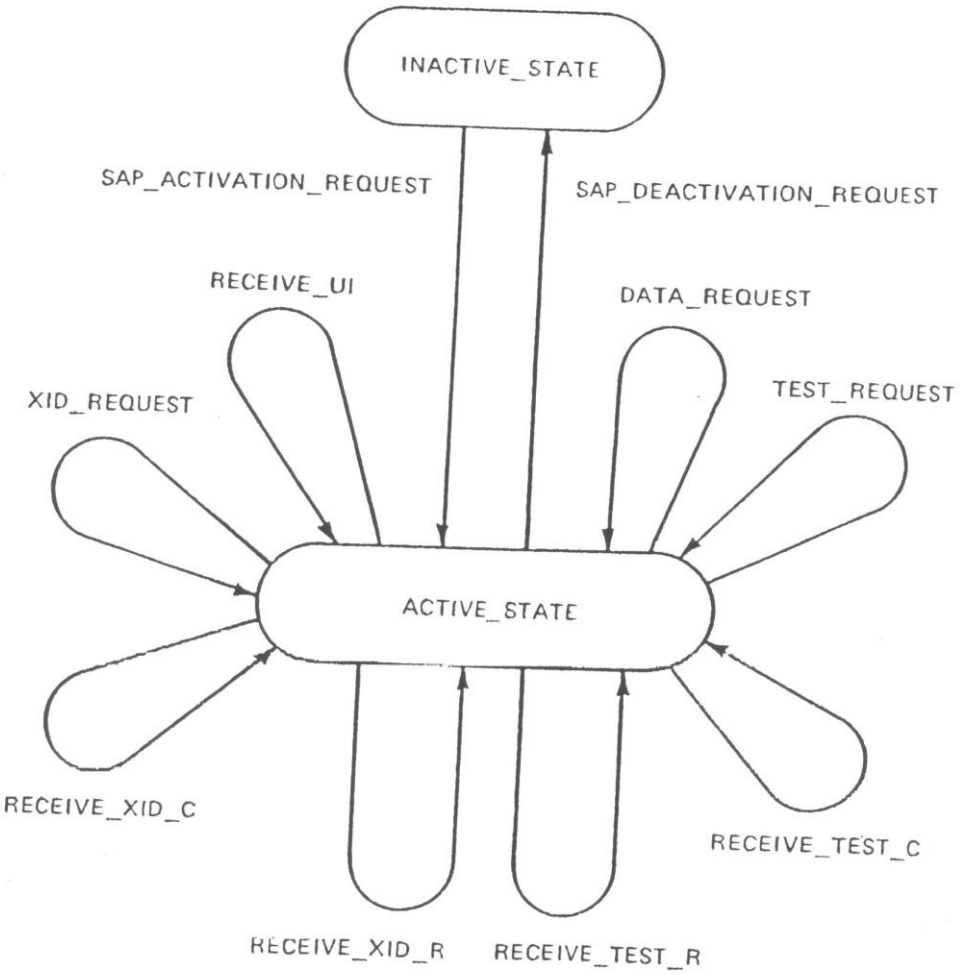
```
        for(i=0;i<=5;i++)        mreq.m_sdu[i]=xid[i];
        mreq.s_class  = mind.s_class;
        send(sm,mreq,sizeof(struct mreq),0);
        recv(sm,mcon,sizeof(struct mcon),0);
        }
    case TEST_com:case TEST_com_p:
                              /* RECEIVE TEST*/
        if(mind.m_sdu[1] & 1)  {
        lind.r_addr -= 1;
              /*TEST RESPONSE*/
        for(i=0;i<=(1496-3);i++)
lind.l_sdu[i]=mind.m_sdu[i+3];
        if(send(sn,lind,sizeof(struct ldata),0)==-1) exit(1);
        }       else   {
        mreq.d_addr  = mind.s_addr;
                        /* TEST COMMAND*/
        for(i=0;i<=1496;i++)
mreq.m_sdu[i]=mind.m_sdu[i];
        mreq.s_class = mind.s_class;
        send(sm,mreq,sizeof(struct mreq),0);
        recv(sm,mcon,sizeof(struct mcon),0);
        }
    }

    i = recv(sn,lreq,sizeof(struct ldata),0);
    if((i = -1) && (errno != EWOULDBLOCK))              exit(1);
/* !!!!???? close(stderr)   ???!!!*/
    if(lreq.l_sdu[0]=DATA_REQUEST) {
        mreq.d_addr   = LOCAL_MAC_SAP_ADDR;
        mreq.m_sdu[0] = lreq.r_addr;
        mreq.m_sdu[1] = lreq.l_addr;
        mreq.m_sdu[2] = UI_com;
```

```
        for(i=0;i<=(1496-3);i++)
mreq.m_sdu[i+3]=lreq.l_sdu[i];
        mreq.s_class  = lreq.s_class;
        send(sm,mreq,sizeof(struct mreq),0);
        recv(sm,mcon,sizeof(struct mcon),0);
    }
    if(lreq.l_sdu[0]=XID_REQUEST) {
        mreq.d_addr   = LOCAL_MAC_SAP_ADDR;
        mreq.m_sdu[0] = lreq.r_addr;
        mreq.m_sdu[1] = lreq.l_addr;
        mreq.m_sdu[2] = XID_com;
        mreq.m_sdu[3] = IEEE_basic;
        mreq.m_sdu[4] = LLC_type_I;
        mreq.m_sdu[5] = WINDOW_size;
        mreq.s_class  = lreq.s_class;
        send(sm,mreq,sizeof(struct mreq),0);
        recv(sm,mcon,sizeof(struct mcon),0);
    }
    if(lreq.l_sdu[0]=TEST_REQUEST) {
        mreq.d_addr   = LOCAL_MAC_SAP_ADDR;
        mreq.m_sdu[0] = lreq.r_addr;
        mreq.m_sdu[1] = lreq.l_addr;
        mreq.m_sdu[2] = TEST_com;
        for(i=0;i<=(1496-3);i++)
mreq.m_sdu[i+3]=lreq.l_sdu[i];
        mreq.s_class = lreq.s_class;
        send(sm,mreq,sizeof(struct mreq),0);
        recv(sm,mcon,sizeof(struct mcon),0);
    }
    if(lreq.l_sdu[0]=SAP_DEACTIVATION_REQUEST) {
        REPORT_STATUS(SAP_INACTIVE);
        close(sn);
```

```
} else {     sap(); }
}
```

4.3.1 Διάγραμμα Μετάβασης Καταστάσεων του SAP Component



4.3.2 Πίνακας Μετάβασης Καταστάσεων του SAP Component

Current State	Event	Action(s)	Next State
INACTIVE_ STATE	SAP_ACTIVAT :_REQUEST	REPORT_STATUS(SAP_ACTIVE)	ACTIVE_STATE
ACTIVE_STATE	RECEIVE_UI	DATA_INDICATE	ACTIVE_STATE
	DATA_REQUEST	SEND_UI	ACTIVE_STATE
	XID_REQUEST	SEND_XID_C	ACTIVE_STATE
	RECEIVE_XID_C	SEND_XID_R	ACTIVE_STATE
	RECEIVE_XID_R	XID_INDICATE	ACTIVE_STATE
	TEST_REQUEST	SEND_TEST_C	ACTIVE_STATE
	RECEIVE_TEST_C	SEND_TEST_R	ACTIVE_STATE
	RECEIVE_TEST_R	TEST_INDICATE	ACTIVE_STATE
	SAP_DEACTIVATION_REQUEST	REPORT_STATUS(SAP_INACTIVE)	INACTIVE_ STATE

4.3.3 Περιγραφή των Καταστάσεων του SAP Component

INACTIVE_STATE : Το SAP Component του LLC είναι απενεργοποιημένο.

ACTIVE_STATE : Το SAP Component του LLC είναι ενεργοποιημένο και μπορεί να στείλει και να λάβει PDUs.

4.3.4 Περιγραφή των Γεγονότων του SAP Component

SAP_ACTIVATION_REQUEST : Ο χρήστης του SAP έχει κάνει αίτηση το συγκεκριμένο SAP να ενεργοποιηθεί και να είναι σε θέση να παρέχει υπηρεσίες τύπου 1.

SAP_DEACTIVATION_REQUEST : Ο χρήστης του SAP έχει κάνει αίτηση το συγκεκριμένο SAP να απενεργοποιηθεί.

XID_REQUEST : Ο SAP χρήστης έχει κάνει αίτηση το SAP Component να στείλει μια XID εντολή σε ένα ή περισσότερα απομακρυσμένα SAPs.

TEST_REQUEST : Ο SAP χρήστης έχει κάνει αίτηση το SAP Component να στείλει μια TEST εντολή σε ένα ή περισσότερα απομακρυσμένα SAPs.

RECEIVE_UI : Το τοπικό SAP Component έχει λάβει μια UI PDU από το απομακρυσμένο SAP Component.

DATA_REQUEST : Ο SAP χρήστης έχει κάνει αίτηση μια μονάδα δεδομένων (Data Unit) να μεταδοθεί στο απομακρυσμένο SAP, ìέσω μιας UI PDU.

RECEIVE_XID_C : Το τοπικό SAP Component έχει λάβει μια XID εντολή από το απομακρυσμένο SAP Component.

RECEIVE_XID_R : Το τοπικό SAP Component έχει λάβει μια XID απόκριση από το απομακρυσμένο SAP Component.

RECEIVE_TEST_C : Το τοπικό SAP Component έχει λάβει μια TEST εντολή από το απομακρυσμένο SAP Component.

RECEIVE_TEST_R : Το τοπικό SAP Component έχει λάβει μια TEST απόκριση από το απομακρυσμένο SAP Component.

4.3.5 Περιγραφή των Ενεργειών του SAP Component

DATA_INDICATE : Το SAP Component έχει λάβει μια UI PDU από το απομακρυσμένο SAP. Η μονάδα δεδομένων της UI PDU μεταβιβάζεται στον SAP χρήστη.

SEND_UI : Μία UI PDU στέλνεται σε ένα ή περισσότερα απομακρυσμένα SAPs σε απόκριση μιας αίτησης του SAP χρήστη να στείλει μία SDU.

SEND_XID_C : Το SAP Component θα στείλει μία XID εντολή σε άλλα απομακρυσμένα SAPs σε απόκριση μιας αίτησης του SAP χρήστη να αναγνωρίσει άλλα SAP's.

SEND_XID_R : Το SAP Component θα στείλει μία XID απόκριση σε άλλα απομακρυσμένα SAP's σε απόκριση μιας ληφθείσας XID εντολής.

SEND_TEST_C : Το SAP Component θα στείλει μία TEST εντολή σε απόκριση μιας αίτησης του SAP χρήστη να ελέγξει τα απομακρυσμένα SAPs.

SEND_TEST_R : Το SAP Component θα στείλει μία TEST απόκριση σε απόκριση μιας ληφθείσας TEST εντολής.

REPORT_STATUS : Το SAP Component είναι σε θέση να αναφέρει την κατάσταση της σύνδεσης του συγκεκριμένου SAP. Τα έγκυρα αποτελέσματα είναι :

SAP_ACTIVE : Το SAP τέθηκε και είναι σε λειτουργία κατόπιν της σχετικής αίτησης.

SAP_INACTIVE : Το SAP είναι απενεργοποιημένο κατόπιν της σχετικής αίτησης.

XID_INDICATE : Το SAP Component έχει λάβει μια XID απόκριση από ένα απομακρυσμένο SAP. Μια ένδειξη (Indication) του γεγονότος αυτού μεταδίδεται στο SAP χρήστη.

TEST_INDICATE : Το SAP Component έχει λάβει μια TEST απόκριση από ένα απομακρυσμένο SAP. Μια ένδειξη (Indication) του γεγονότος αυτού μεταδίδεται στο SAP χρήστη.

Συνημμένες βιβλιοθήκες

Αρχεία που συμπεριλαμβάνονται με την βοήθεια της #INCLUDE, είναι τα ακόλουθα:

- errno.h: πρόκειται για το αρχείο που επιτρέπει στο πρόγραμμα να δώσει επακριβείς κώδικες σφαλμάτων, που επιτρέπουν στον προγραμματιστή να κάνει τις κατάλληλες διορθώσεις.
- memory.h: πρόκειται για κεφαλίδα που είναι απαραίτητη για την ασφαλή αρχικοποίηση μεταβλητών που πραγματοποιείται μέσω της εντολής memset.
- cstdlib.h: μια από τις βασικότερες βιβλιοθήκες της C++, χρησιμοποιείται μεταξύ άλλων για απλές κλήσεις συστήματος, όπως το exit(), που χρησιμοποιείται για έξοδο σε περίπτωση σφάλματος και εμφάνιση ενός κωδικού, κατά κανόνα διαφορετικού από το μηδέν, ο οποίος και θα ανιχνεύει το είδος του σφάλματος που παρουσιάστηκε.
- winsock2.h: πρόκειται για την δεύτερη έκδοση της βιβλιοθήκης η οποία συμπεριλαμβάνεται στο Visual C++ 2008 και μέσω της οποίας μπορεί να δρομολογήσει όλες τις βασικές πράξεις που αφορούν τον προγραμματισμό sockets: δημιουργία, binding, αποστολή(sendto), λήψη(recv) και άλλα.
- ws2tcpip.h: πρόκειται για ένα από τα δύο βασικότερα αρχεία κεφαλίδας μαζί με το προηγούμενο, τα οποία χρησιμεύουν στην διεκπεραίωση των βασικότερων λειτουργιών των sockets.

Αρχεία που συμπεριλαμβάνονται στο κύριο σώμα κώδικα:

- #pragma comment: αυτή η εντολή συνδέει την εφαρμογή με το αρχείο «Ws2_32.lib», κάτι το οποίο σύμφωνα με τις επίσημες οδηγίες της Microsoft για το Winsock, είναι απαραίτητο για μια εφαρμογή Winsock.
- Δημιουργία μιας διαδικτυακής διεύθυνσης(Internet address) που αντιστοιχεί στο λεγόμενο τοπικό τερματικό (local server): ***struct sockaddr_in myaddr_in;***
  Για την δημιουργία μιας τέτοιας διεύθυνσης υπάρχουν ορισμένες δομές που μπορούν να χρησιμοποιηθούν ως διαθέσιμες στα Windows. Οι συχνότερα χρησιμοποιούμενες είναι οι δομές *sockaddr* και *sockaddr_in*. Η δομή sockaddr θα χρησιμοποιηθεί σε κάθε περίπτωση ως όρισμα, οπότε θα πραγματοποιηθεί μετατροπή σε αυτήν. Εδώ αξιοποιούμε την δομή sockaddr_in. Αυτή η δομή είναι η πλουσιότερη, εφόσον διαθέτει τέσσερις μεταβλητές-μέλη συνολικά, μια εκ των οποίων είναι η θύρα σύνδεσης.
- Δημιουργία μιας δεύτερης διαδικτυακής διεύθυνσης, η οποία γίνεται μέσα στον κώδικα του τερματικού του πελάτη. Αυτή η εργασία μέσα στον κώδικα πραγματοποιείται με την εξής εντολή: **struct sockaddr_in newaddr_in;**
  Αυτή η διεύθυνση είναι αυτή που θα χρησιμοποιηθεί προκειμένου να αποσταλεί μήνυμα και να ληφθεί απάντηση από τον διακομιστή στον κώδικα του socket client.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η ανάπτυξη του προτύπου LLC αποτελεί μια πρόκληση μιας και ενσωματώνει ποικιλία μηχανισμών και τεχνικών. Η πραγματική πρόκληση είναι η μεθοδολογία υλοποίησης του σε κώδικα. Έχοντας σχεδιαστεί και υλοποιηθεί σε προγενέστερη περίοδο, λειτουργική περιγραφή του ακολουθούσε του κανόνες τις εποχής. Η απουσία αντικειμενοστραφούς προγραμματισμού έκανε τη χρήση τεχνικών σχεδόν πρωτόγονες. Παρόλα αυτά, αναπτύχθηκε ένα πρότυπο που ενσωματώνει τις λειτουργίες του.

Το σημαντικότερο χαρακτηριστικό του πρωτοκόλλου είναι ο αυτοματισμός του. Λόγω της σχεδίασης του, η μηχανή κατάστασης διατηρεί το πρωτόκολλο σε λειτουργία σε κάθε ενδεχόμενο. Το αποτέλεσμα είναι μια σειρά από μηνύματα που επιβεβαιώνουν την ανταλλαγή των μηνυμάτων. Δεν ήταν όμως εύκολο να αναπαρασταθεί αυτό γραφικά.

Για την εξακρίβωση της καλής λειτουργίας του πρωτοκόλλου αναπτύχτηκε επιπλέον κώδικας δοκιμών. Ο κώδικας αυτό περιλαμβάνει την υλοποίηση της επικοινωνίας με Socket και έλεγχο σφαλμάτων. Σκοπός του η εφαρμογή σεναρίων ελέγχου του πρωτοκόλλου αλλά και της αποτελεσματικότητας του κώδικα. Η παράθεση του κώδικα δοκιμών δεν κρίθηκε απαραίτητη να ενσωματωθεί μιας και δεν αποτελεί απαραίτητο στο πρωτόκολλο μέρος.

Σημαντικότερο πρόβλημα της ανάπτυξης ήταν η πολυπλοκότητα ενός πρωτοκόλλου και η χρήση νέων τεχνολογιών. Το πρωτόκολλο διαθέτει ποικιλία μηχανισμών και απαιτεί καλή γνώση των τεχνικών προγραμματισμού για να απεικονιστούν πλήρως οι λειτουργίες του. Οι νέες τεχνολογίες από την άλλη έχουν πολύ μεγάλο εύρος με αποτέλεσμα να είναι δύσκολο να αξιοποιηθούν πλήρως. Η καλύτερη γνώση και των δύο θα βοηθούσε στην απόλυτη υλοποίηση του μηχανισμού του πρωτοκόλλου.

Βιβλιογραφία

[1] Wikipedia “Communication”, https://en.wikipedia.org/wiki/Communication

[2] Wikipedia “Telecommunication”, https://en.wikipedia.org/wiki/Telecommunication

[3] Wikipedia “Computer network”, https://en.wikipedia.org/wiki/Computer_network

[4] A. S. Tanenbaum, D. J. Wetherall “Computer Networks”, Pearson; 5th edition, Oct. 2010, ISBN: 978-0-132-12695-3

[5] J. F. Kurose, K. W. Ross, “Computer Networking: A Top-Down Approach”, Addison-Wesley, 6th edition, June 2012, ASIN: B0092WUB4C

[6] Wikipedia “ARPANET”, https://en.wikipedia.org/wiki/ARPANET

[7] ΚΕΔ – ΠΘ “Εισαγωγή στο Internet”, http://www.uth.gr/main/help/help-desk/internet/internet3.html

[8] HubPages Inc. “The Pros and Cons of Computer Networks”, http://energyguild.hubpages.com/hub/The-Pros-and-Cons-of-Computer-Networks

[9] OccupyTheory.org “Advantages and Disadvantages of Computer Networking”, http://occupytheory.org/advantages-and-disadvantages-of-computer-networking/

[10] Y. Donoso, R. Fabregat “Multi-Objective Optimization in Computer Networks Using Metaheuristics”, CRC Press, 2007, ISBN 978-1-420-01362-7

[11] T. Dean “Network+ Guide to Networks”, Cengage Learning, 5th edition, 2009, ISBN 978-1-423-90245-4

[12] V.S.Bagad, I.A.Dhotre “Computer Network”, Technical Publications, 2009, ISBN 978-8-184-31703-9

[13] D. Reilly, M. Reilly “Java: Network Programming and Distributed Computing”, Addison-Wesley Professional, Apr. 2002, ISBN 978-0-201-71037-3

[14] R. Hauben “From the ARPANET to the Internet”, A Study of the ARPANET TCP/IP Digest and of the Role of Online Communication in the Transition from the ARPANET to the Internet, http://www.columbia.edu/~rh120/other/tcpdigest_paper.txt

[15] N. P. Gopalan, B. Siva Selvan “TCP/IP ILLUSTRATED”, PHI Learning Pvt. Ltd., 2008, ISBN 978-8-120-33283-6

[16] D. E. Comer “Computer Networks and Internets”, Pearson Education, 5th edition, 2011, ISBN 978-0-133-00210-2

[17] B. Singh “DATA COMMUNICATIONS AND COMPUTER NETWORKS”, PHI Learning Pvt. Ltd., 3rd edition, 2011, ISBN 978-8-120-34466-2

[18] S. C. Yadava “Introduction To Client Server Computing”, New Age International, 2009, ISBN 978-8-122-42689-2

[19] G. Held “Windows Networking Tools: The Complete Guide to Management, Troubleshooting, and Security”, CRC Press, 2013, ISBN 978-1-466-58887-5

[20] M. Kerrisk “The Linux Programming Interface”, No Starch Press, 2010, ISBN 978-1-593-27220-3

[21] E. R. Harold “Java Network Programming”, "O'Reilly Media, Inc.", 2013, ISBN 978-1-449-36596-7

[22] R. Williams “Computer Systems Architecture: A Networking Approach”, Pearson Education, 2006, ISBN 978-0-321-34079-5

[23] J. Geier “Wireless LANs”, Sams, Sams White Book Series, 2nd edition, July 2002, ISBN 978-0-672-32058-3

[24] Wikipedia “IEEE 802.2”, https://en.wikipedia.org/wiki/IEEE_802.2 / IEEE Standard for Information technology “Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements--Part 2: Logical Link Control”, New York: The Institute of Electrical and Electronics Engineers, Inc. May, 2008. ISBN 1-55937-959-6

[25] E. Blanchard “Introduction to Networking and Data Communications”, http://epq.com.co/softw_internet/nag1/c5083.htm

[26] S. Banzal “Data and Computer Network Communication”, Firewall Media, Jan. 2007, ISBN 978-8-131-80139-0

[27] Internet Assigned Numbers Authority “IEEE 802 Numbers”, http://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml

[28] Rhys Haden “SDLC, HDLC and LLC”, Data Network Resource, http://www.rhyshaden.com/hdlc.htm

[29] Cisco “Understanding Logical Link Control”, Document ID: 12247, Sep. 2005, http://www.cisco.com/c/en/us/support/docs/ibm-technologies/logical-link-control-llc/12247-45.html

[30] Savvius Inc. “LLC Overview”, http://www.wildpackets.com/resources/compendium/llc/overview

[31] Rhys Haden “IPX”, Data Network Resource, http://www.rhyshaden.com/ipx.htm