

# **ΠΕΡΙΕΧΟΜΕΝΑ**

## **Κεφάλαιο 1 Εισαγωγή**

- 1.0. Επισκόπηση
- 1.1. Δίκτυα υπολογιστών
- 1.2. Πρωτόκολλα δικτύων υπολογιστών
- 1.3. Το πρόβλημα της διαχείρισης
- 1.4. Μοντέλα και πρότυπα διαχείρισης
- 1.5. Τρόποι διακίνησης πληροφοριών διαχείρισης
- 1.6. Βιβλιογραφία

## **Κεφάλαιο 2 Τα Πρωτόκολλα TCP/IP**

- 2.0. Εισαγωγή στα πρωτόκολλα TCP/IP και το INTERNET
- 2.1. Μέσα μετάδοσης, φυσικές διευθύνσεις
- 2.2. Το πρωτόκολλο IP
  - 2.2.1. Κατακερματισμός και επανασύνδεση (Fragmentation and reassembly)
  - 2.2.2. IP-διευθύνσεις
  - 2.2.3. Η διευθέτηση των υποδικτύων
  - 2.2.4. Το πρωτόκολλο ICMP
- 2.3. Δρομολόγηση
  - 2.3.1. Αυτόνομα Συστήματα
  - 2.3.2. Address Resolution Protocol (ARP) - Reverse Address Resolution Protocol (RARP)
  - 2.3.3. Interior Gateway Protocols (IGP): RIP - OSPF. Παραδείγματα
  - 2.3.4. Exterior Gateway Protocol (EGP)
  - 2.3.5. Αλγόριθμοι εύρεσης ελαχίστων δρόμων: Bellman-Ford, Dijkstra, Floyd-Warshall. Παραδείγματα
- 2.4. Πρωτόκολλα επιπέδου μεταφοράς
  - 2.4.1. Το πρωτόκολλο TCP
    - 2.4.1.1. Αξιόπιστη υπηρεσία μεταφοράς - stream
    - 2.4.1.2. Η λειτουργία του πρωτοκόλλου
  - 2.4.2. Το πρωτόκολλο UDP
- 2.5. Εφαρμογές TCP/IP
- 2.6. Προγραμματισμός με sockets στο Unix. Παραδείγματα
- 2.7. Ασκήσεις
- 2.8. Βιβλιογραφία

## **Κεφάλαιο 3 Το Μοντέλο Αναφοράς OSI**

- 3.0. Εισαγωγή
- 3.1. Το μοντέλο αναφοράς πρωτοκόλλων OSI του ISO

- 3.2. Εισαγωγή στα επίπεδα OSI
- 3.3. Περιγραφή των επιπέδων
  - 3.3.1. Επίπεδο Εφαρμογής
  - 3.3.2. Επίπεδο Παρουσίασης
  - 3.3.3. Επίπεδο Συνόδου
  - 3.3.4. Επίπεδο Μεταφοράς
  - 3.3.5. Επίπεδο Δικτύου
  - 3.3.6. Επίπεδο Σύνδεσης Δεδομένων
  - 3.3.7. Φυσικό Επίπεδο
- 3.4. Διαφορές με τα πρωτόκολλα TCP/IP
- 3.5. Ειδικά στοιχεία υπηρεσίας στο επίπεδο εφαρμογής: FTAM, MHS
- 3.6. Περιβάλλον ανάπτυξης εφαρμογών ISO/OSI και TCP/IP, ISODE
- 3.7. Ασκήσεις
- 3.8. Βιβλιογραφία

#### **Κεφάλαιο 4      Πρότυπα    και    Πρωτόκολλα    Διαχείρισης    Δικτύων TCP/IP**

- 4.0. Εισαγωγή
- 4.1. Μοντέλο διαχείρισης δικτύων
  - 4.1.1. Διαχειριζόμενοι κόμβοι (agents)
  - 4.1.2. Σταθμοί διαχείρισης δικτύων (managers)
  - 4.1.3. Πρωτόκολλα διαχείρισης δικτύων
  - 4.1.4. Διαχείριση με πληρεξούσιους κόμβους (proxy agents)
- 4.2. Διαχείριση δικτύων TCP/IP
- 4.3. Το πρωτόκολλο SNMP
- 4.4. Δομή και αποθήκευση της διαχειριζόμενης πληροφορίας: SMI και MIB
- 4.5. Δυνατές λειτουργίες στη διαχειριζόμενη πληροφορία. Μορφή και σημασία των ανταλλασσομένων μηνυμάτων
- 4.6. Σύνταξη και κωδικοποίηση πληροφορίας (ASN.1 και BER)
- 4.7. Υλοποίηση λειτουργιών διαχείρισης με βάση το SNMP
  - 4.7.1. Παρουσίαση της τοπολογίας του δικτύου
  - 4.7.2. Παρακολούθηση της απόδοσης του δικτύου
  - 4.7.3. Άλλες ενδείξεις
- 4.8. Διαχείριση Επιδόσεων (Performance Management)
  - 4.8.1. Υπολογισμός χρησιμοποίησης τμημάτων/συνδέσμων του δικτύου
  - 4.8.2. Διαχείριση καταστάσεων συμφόρησης
  - 4.8.3. Υπολογισμός ρυθμών και ποσοστών σφαλμάτων
  - 4.8.4. Αναγνώριση προτύπων για το φορτίο στο δίκτυο
- 4.9. Ασφάλεια: απειλές και μηχανισμοί
- 4.10. Remote Network Monitoring (RMON)
  - 4.10.1. Βασικές έννοιες
  - 4.10.2. RMON MIB
- 4.11. Το πρωτόκολλο SNMPv2 (SNMP-version 2)
  - 4.11.1. Εισαγωγή

- 4.11.2. Δομή της Πληροφορίας Διαχείρισης
- 4.11.3. Λειτουργίες του Πρωτοκόλλου
- 4.11.4. SNMPv2 MIB
- 4.11.5. Manager-to-Manager MIB
- 4.11.6. Συνύπαρξη με το SNMP
- 4.11.7. SNMPv2: Ασφάλεια
- 4.12. Ασκήσεις
- 4.13. Βιβλιογραφία

## **Κεφάλαιο 5 Διαχείριση OSI**

- 5.0. Εισαγωγή στη διαχείριση OSI
- 5.1. Διαχείριση δικτύων βασισμένη στο πρότυπο OSI
  - 5.1.1. Πρότυπα Διαχείρισης OSI
    - 5.2.1.1. Αρχιτεκτονική και Δομή
    - 5.2.1.2. Μεταφορά της Πληροφορίας Διαχείρισης
    - 5.2.1.3. Δομή της Πληροφορίας Διαχείρισης
    - 5.2.1.4. Λειτουργίες Διαχείρισης Συστημάτων
  - 5.1.2. Περιοχές Λειτουργιών Διαχείρισης OSI
    - 5.1.2.1. Διαχείριση Σφαλμάτων (Fault Management)
    - 5.1.2.2. Λογιστική Διαχείριση (Accounting Management)
    - 5.1.2.3. Διαχείριση Διάρθρωσης (Configuration Management)
    - 5.1.2.4. Διαχείριση Επιδόσεων (Performance Management)
    - 5.1.2.5. Διαχείριση Ασφάλειας (Security Management)
- 5.2. Διαχείριση Συστημάτων
  - 5.2.1. Μοντέλο Διαχείρισης Συστημάτων
    - 5.2.1.1. Θέματα Πληροφορίας
    - 5.2.1.2. Λειτουργικές Πλευρές
    - 5.2.1.3. Θέματα OSI Επικοινωνιών
    - 5.2.1.4. Οργανωτικές Πλευρές
- 5.3. Τα πρωτόκολλα CMIP/CMIS
  - 5.3.1. Common Management Information Service
  - 5.3.2. Common Management Information Protocol
- 5.4. Βάση Πληροφορίας Διαχείρισης OSI (OSI-MIB)
  - 5.4.1. Μοντέλο Πληροφορίας Διαχείρισης
  - 5.4.2. Ορισμός της Πληροφορίας Διαχείρισης
  - 5.4.3. GDMO (Guidelines for the Definition of Managed Objects)
  - 5.4.4. Πρακτικά ζητήματα
- 5.5. Περιγραφή των λειτουργιών διαχείρισης συστημάτων
- 5.6. Το όφελος από μια τυποποιημένη στοίβα πρωτοκόλλων
- 5.7. Σύγκριση μεταξύ των SNMP και CMIP
- 5.8. Το πρωτόκολλο CMOT
- 5.9. Ασκήσεις
- 5.10. Βιβλιογραφία

## **Κεφάλαιο 6 Διαχείριση Χαμηλών Επιπέδων**

- 6.0. Εισαγωγή
- 6.1. Διαχείριση φυσικού επιπέδου και επιπέδου MAC
  - 6.1.1. Συστήματα διαχείρισης γραμμών (Cable Management Systems)
  - 6.1.2. Μηχανές διάγνωσης προβλημάτων (Diagnostic Devices)
  - 6.1.3. Εφαρμογές των παραπάνω συστημάτων
  - 6.1.4. Δυνατότητες της MIB II για διαχείριση χαμηλών επιπέδων
- 6.2. Διαχείριση διαμορφωτών (τέστ βρόχου)
- 6.3. Θέματα διαχείρισης δημόσιων και ιδιωτικών δικτύων X.25
- 6.4. HELASPAC
- 6.5. Ευφυείς πολυπλέκτες. Ιδιωτικά λογικά δίκτυα: HELASCOM
- 6.6. ISDN - Διαχείριση του ISDN
- 6.7. Ασκήσεις
- 6.8. Βιβλιογραφία

## **Κεφάλαιο 7 Γέφυρες και Δρομολογητές**

- 7.0. Εισαγωγή
- 7.1. Τοπικά δίκτυα
- 7.2. Τα πρότυπα 802.X της IEEE
  - 7.2.1. Το επίπεδο ελέγχου προσπέλασης του μέσου (MAC)
  - 7.2.2. Το επίπεδο ελέγχου λογικών συνδέσεων (LLC)
- 7.3. Η ανάγκη για διασυνδεδεμένα τοπικά δίκτυα
- 7.4. Γέφυρες. Δρομολόγηση στο επίπεδο MAC
- 7.5. Διαφανείς γέφυρες
  - 7.5.1. Η λειτουργία ενημέρωσης της γέφυρας
  - 7.5.2. Ο αλγόριθμος επικαλύπτοντος δένδρου (Spanning Tree Algorithm)
  - 7.5.3. Συντονισμός της τοπολογίας
  - 7.5.4. Παράδειγμα
  - 7.5.5. Απομακρυσμένες γέφυρες (Remote Bridges)
- 7.6. Δρομολόγηση πηγής (Source Routing). Παράδειγμα
- 7.7. Σύγκριση μεταξύ διαφανών γεφυρών και γεφυρών δρομολόγησης πηγής
- 7.8. Η γεφύρωση διαφορετικών δικτύων
- 7.9. Θέματα διαχείρισης γεφυρών
- 7.10. Δρομολογητές (Routers), σύγκριση με τις γέφυρες
- 7.11. Ασκήσεις
- 7.12. Βιβλιογραφία

## **Κεφάλαιο 8 Αρχιτεκτονικές και Ανάπτυξη Συστημάτων Διαχείρισης**

- 8.0. Εισαγωγή
- 8.1. Γενικά χαρακτηριστικά ενός ΣΔΔ
- 8.2. Αρχιτεκτονικές ΣΔΔ
  - 8.2.1. Κεντροποιημένο ΣΔΔ

- 8.2.2. Κατανεμημένο ΣΔΔ
- 8.2.3. Ιεραρχικό ΣΔΔ
- 8.2.4. Δικτυωμένο ΣΔΔ
- 8.3. Συμμόρφωση με τα πρότυπα - Απαιτήσεις ενός ΣΔΔ
- 8.4. Αντικειμενοστραφής φιλοσοφία και διαχείριση δικτύων
  - 8.4.1. Τι είναι η αντικειμενοστραφής φιλοσοφία
  - 8.4.2. Αντικείμενα και MIB
    - 8.4.2.1. SNMP MIB
    - 8.4.2.2. CMIP MIB
  - 8.4.3. Εφαρμογή σε ΣΔΔ
  - 8.4.4. Πρότυπα - Τυποποιήσεις
- 8.5. Ανάπτυξη του λογισμικού ενός ΣΔΔ
- 8.6. Έμπειρα συστημάτων
  - 8.6.1. Έμπειρα συστήματα και διαχείριση δικτύων
  - 8.6.2. Τι κάνει ένα έμπειρο σύστημα
  - 8.6.3. Κατηγορίες έμπειρων συστημάτων
  - 8.6.4. Στοιχεία που αποτελούν το έμπειρο σύστημα
  - 8.6.5. Υλοποίηση εμπείρων συστημάτων
- 8.7. Ασκήσεις
- 8.8. Βιβλιογραφία

## **Κεφάλαιο 9      Ενοποιημένη Διαχείριση**

- 9.0. Εισαγωγή
- 9.1. Χαρακτηριστικά των συστημάτων ενοποιημένης διαχείρισης
- 9.2. Η αρχιτεκτονική Unified Network Management Architecture (UNMA)
  - 9.2.1. Μια γενική άποψη της αρχιτεκτονικής
  - 9.2.2. Το Network Management Protocol (NMP)
  - 9.2.3. Προσαρμοστικότητα της αρχιτεκτονικής UNMA
  - 9.2.4. Προϊόντα και υπηρεσίες βασιζόμενα στην UNMA αρχιτεκτονική
- 9.3. Η αρχιτεκτονική Enterprise Management Architecture (EMA)
  - 9.3.1. Περιγραφή της αρχιτεκτονικής EMA
  - 9.3.2. Διαχειριστικά προϊόντα της DEC
- 9.4. Η αρχιτεκτονική NetView
  - 9.4.1. Χαρακτηριστικά του NetView
  - 9.4.2. NetView Tools
- 9.5. Το περιβάλλον διαχείρισης OpenView
  - 9.5.1. Γενικά χαρακτηριστικά
  - 9.5.2. Η δομή του OpenView
  - 9.5.3. Κατανεμημένο περιβάλλον διαχείρισης (Distributed Management Environment)
- 9.6. Ολοκληρωμένη διαχείριση ετερογενών συστημάτων
- 9.7. Ασκήσεις
- 9.8. Βιβλιογραφία

**Κεφάλαιο 10 Δίκτυο Διαχείρισης Τηλεπικοινωνιών -  
Telecommunications Management Network (TMN)**

- 10.0. Εισαγωγή
- 10.1. Διαχείριση τηλεπικοινωνιακών δικτύων υψηλής ταχύτητας. Τα TMN πρότυπα στο Κοινοτικό Πρόγραμμα RACE
- 10.2. Λειτουργικό μοντέλο του TMN
- 10.3. Σημεία αναφοράς (Reference Points)
- 10.4. Φυσική Αρχιτεκτονική του TMN και σημεία διασύνδεσης
- 10.5. Λειτουργική δομή ενός Operations System
- 10.6. Βάση Πληροφορίας Διαχείρισης
- 10.7. Φυσική Αρχιτεκτονική του TMN
- 10.8. Βιβλιογραφία

**Κεφάλαιο 11 Παράδειγμα Διαχειριστικού Συστήματος: Το Σύστημα CNMS**

- 11.0. Εισαγωγή
- 11.1. Η δομή του διαχειριστικού συστήματος CNMS
- 11.2. Υλοποίηση της οντότητας διασύνδεσης με το δίκτυο
- 11.3. Συμπεράσματα
- 11.4. Παράδειγμα χρήσης : Το δίκτυο του ΕΜΠ
- 11.5. Βιβλιογραφία

**Γενική Βιβλιογραφία**

**Παράρτημα Α** Το Περιβάλλον Διαχείρισης του Carnegie Mellon University (CMU)

**Παράρτημα Β** MIB II (RFC 1213)

# Κεφάλαιο 1

## 1. Εισαγωγή

### Περιεχόμενα του Κεφαλαίου 1

- 1.0. Επισκόπηση
- 1.1. Λίκτυα υπολογιστών
- 1.2. Πρωτόκολλα δικτύων υπολογιστών
- 1.3. Το πρόβλημα της διαχείρισης
- 1.4. Μοντέλα και πρότυπα διαχείρισης
- 1.5. Τρόποι διακίνησης πληροφοριών διαχείρισης
- 1.6. Βιβλιογραφία

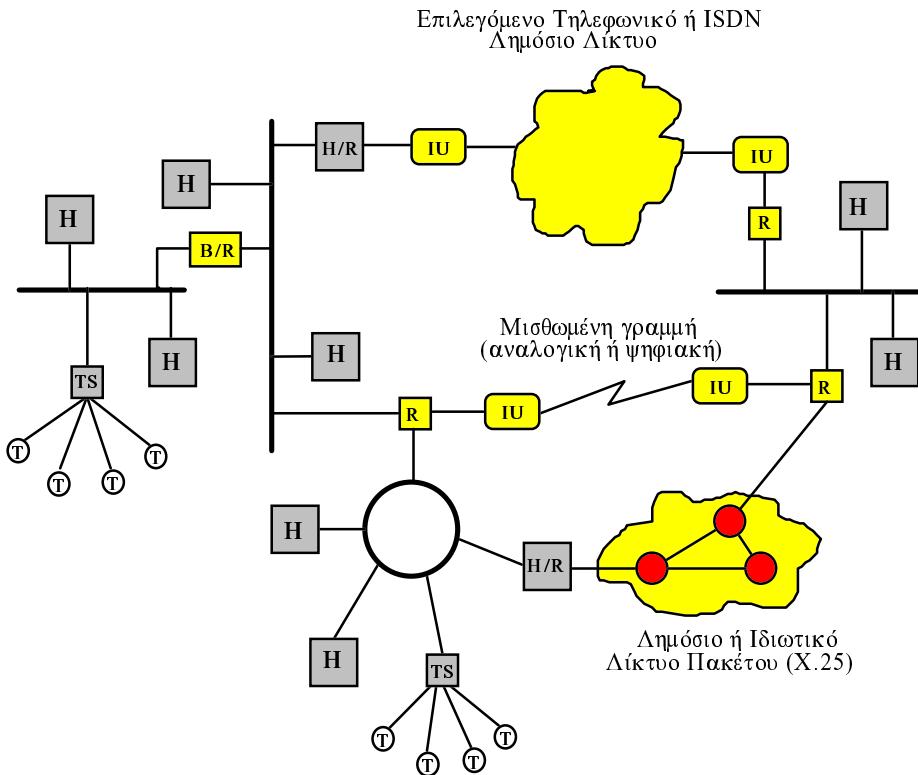
### 1.0. Επισκόπηση

Ονομάζουμε "Διαχείριση δικτύων" όλες τις ενέργειες που έχουν σκοπό τον έλεγχο λειτουργίας, ασφάλειας και απόδοσης, την αντιμετώπιση βλαβών και τη βελτιστοποίηση της λειτουργίας διασυνδεδεμένων υπολογιστικών συστημάτων. Στόχος των σημειώσεων αυτών είναι η εξέταση θεμάτων που αφορούν τη διαχείριση εκτεταμένων δικτύων υπολογιστών.

### 1.1. Λίκτυα υπολογιστών

Στο Σχήμα 1.1 παρουσιάζεται μια γενική μορφή δικτύου, στην οποία επιγραμματικά αναγνωρίζουμε επιμέρους συνιστώσες και σενάρια διασύνδεσης:

- **Υπολογιστές - σταθμούς εργασίας (Hosts)**, οι οποίοι συνδέονται σε ένα δίκτυο και προσφέρουν στους χρήστες υπηρεσίες επεξεργασίας, αποθήκευσης και ανταλλαγής δεδομένων. Η υπολογιστική τους ικανότητα μπορεί να ξεκινά από αυτή ενός προσωπικού υπολογιστή και να φθάνει αυτή ενός supercomputer.
- **Δρομολογητές (Routers) και τις πύλες (Gateways)**, οι οποίοι προσφέρουν υπηρεσίες διασύνδεσης τόσο τοπικών δικτύων μεταξύ τους όσο και με δίκτυα ευρείας περιοχής για την δημιουργία διασυνδεδεμένων δικτύων. Ταυτόχρονα παρέχουν υπηρεσίες δρομολόγησης αποφασίζοντας πιο μονοπάτι θα ακολουθήσει η ροή του φορτίου.
- **Γέφυρες (Bridges) και επαναλήπτες (Repeaters)**, που χρησιμοποιούνται και πάλι για διασύνδεση τοπικών δικτύων εκτελώντας όμως λιγότερες λειτουργίες από αυτές του δρομολογητή. Ειδικότερα η γέφυρα συνήθως λειτουργεί στο υποεπίπεδο ελέγχου πρόσβασης στο μέσο επικοινωνίας (MAC). Διαφέρει από τον επαναλήπτη για το λόγο ότι αποθηκεύει, επεξεργάζεται (στοιχειωδώς) και μετά προωθεί πλαίσια πληροφορίας, ενώ ο επαναλήπτης απλά αναγεννά τα σήματα στο φυσικό επίπεδο. Ετσι οι γέφυρες δεν αναμεταδίδουν λανθασμένα πλαίσια.



H : Υπολογιστής - Σταθμός εργασίας (Host)

H/R : Υπολογιστής - Δρομολογητής

R : Δρομολογητής (Router), Πύλη (Gateway)

B/R : Γέφυρα (Bridge) ή Επαναλήπτης (Repeater)

TS : Εξυπηρετητής Τερματικών Σταθμών(Terminal Server)

T : Τερματικό (Terminal)

IU : Μονάδα Διασύνδεσης (Interface Unit),

### Σχήμα 1.1 - Γενική Μορφή Δικτύου

- **Σταθμούς εξυπηρέτησης τερματικών (Terminal Servers) και τερματικά (Terminals).**
- **Δημόσια ή ιδιωτικά δίκτυα πακέτου (X.25).** Τα δίκτυα αυτά χρησιμοποιούνται για την διασύνδεση μεταξύ τοπικών δικτύων απομακρυσμένων μεταξύ τους. Θα αναφερθούμε λεπτομερώς στα διασυνδεδεμένα δίκτυα στη συνέχεια.
- **Επιλεγόμενο τηλεφωνικό ή ISDN (Integrated Services Digital Network) δημόσιο δίκτυο.** Στη σύγχρονη μορφή τους, τα δημόσια τηλεφωνικά δίκτυα εκτός από την μεταφορά φωνής, χρησιμοποιούνται για την μεταφορά δεδομένων και εικόνας. Στην ψηφιακή τηλεφωνία, η μεταγωγή και προώθηση εντός του τηλεφωνικού δικτύου γίνεται με ψηφιακό τρόπο. Η πρόσβαση των συνδρομητών στο δίκτυο γίνεται με τον κλασσικό αναλογικό τρόπο και το αναλογικό σήμα ψηφιοποιείται στα κέντρα μεταγωγής (ψηφιακά κέντρα).

Το ISDN αντιπροσωπεύει την εξέλιξη των ψηφιακών τηλεφωνικών δικτύων, όπου η ψηφιακή πρόσβαση επεκτείνεται μέχρι τον χρήστη, προσφέροντας μια σειρά

από βασικές και νέες τηλεπικοινωνιακές υπηρεσίες. Από τεχνικής άποψης το ISDN απαιτεί:

την παρουσία ενός δημόσιου ψηφιακού δικτύου κορμού (ψηφιακά τηλεφωνικά κέντρα),

τη δυνατότητα ψηφιακής πρόσβασης μέχρι τον χρήστη (digital local loop),

τη λειτουργία συστήματος κοινού καναλιού σηματοδοσίας στο ψηφιακό δίκτυο κορμού (το CSS 7).

Το ISDN ευρείας ζώνης (Broadband-ISDN) αποβλέπει στην εξυπηρέτηση συνδρομητών με μεγάλες απαιτήσεις για προηγμένες υπηρεσίες, όπως τηλεσυνδιάσκεψη, μετάδοση δεδομένων και εικόνων σε υψηλές ταχύτητες για εκδοτικούς σκοπούς, λειτουργίες CAD/CAM, ιατρικές εφαρμογές κ.α.

- **Μισθωμένες γραμμές (αναλογικές ή ψηφιακές).** Τις γραμμές αυτές μπορεί να προσφέρει ο οργανισμός τηλεπικοινωνιών για αποκλειστική χρησιμοποίηση από κάποιο χρήστη. Οι γραμμές αυτές μπορούν να χρησιμοποιηθούν είτε για μεταφορά φωνής είτε για μεταφορά δεδομένων. Σε περίπτωση αναλογικών γραμμών οι ταχύτητες κυμαίνονται από 2400-9600 Kbps, ενώ στην περίπτωση ψηφιακών γραμμών οι ταχύτητες αυξάνονται σημαντικά. Με κατάλληλη διευθέτηση των κυκλωμάτων μέσω πολυπλεκτών (cross connect) στα τηλεπικοινωνιακά κέντρα επιτυγχάνεται η αποκλειστική χρήση από ένα μόνο χρήστη και η παράκαμψη των συστημάτων μεταγωγής.

Τα δίκτυα υπολογιστών μπορούν να ταξινομηθούν με βάση διάφορα κριτήρια.

Οσον αφορά το εύρος της περιοχής που καλύπτει το δίκτυο, έχουμε τις εξής κατηγορίες:

1. **Τοπικά δίκτυα (Local Area Networks, LANs).** Σ' αυτά πολλοί χρήστες συνδέονται σε ένα κοινό επικοινωνιακό μέσο υψηλού ρυθμού μετάδοσης. Η διευθέτηση της κοινής προσπέλασης στηρίζεται συνήθως σε πρωτόκολλα που επιλύουν προβλήματα συγκρούσεων, πιθανά σε βάρος της βέλτιστης χρησιμοποίησης του μέσου και της μέγιστης απόστασης επικοινωνίας. Οι ταχύτητες κυμαίνονται από 256 Kbps μέχρι 2 Gbps, ενώ η γεωγραφική έκταση του δικτύου περιορίζεται από ένα έως μερικά κτιριακά συγκροτήματα.
2. **Μητροπολιτικά δίκτυα (Metropolitan Area Networks, MANs).** Στην κατηγορία αυτή ανήκουν νέες τεχνολογίες δικτύων υπολογιστών, που περιορίζονται σε μεσαίου μεγέθους γεωγραφικές περιοχές (επίπεδο μεγαλούπολης) και προσφέρουν υψηλές ταχύτητες μετάδοσης δεδομένων (100 Mbps έως 140 Mbps).
3. **Δίκτυα ευρείας περιοχής (Wide Area Networks, WANs).** Ονομάζονται και long haul networks. Συνδέουν κόμβους με οποιαδήποτε απόσταση μεταξύ τους. Τα δίκτυα WAN προσφέρουν τις χαμηλότερες ταχύτητες μετάδοσης οι οποίες σήμερα κυμαίνονται από 9,6 Kbps έως 2 Mbps. Συνήθως χρησιμοποιούν δικτυακές υποδομές δημοσίων τηλεπικοινωνιακών φορέων (π.χ OTE).

Οσον αφορά την αρχιτεκτονική εντοπίζουμε δύο κατηγορίες :

1. **Κεντροποιημένα Δίκτυα.** Τερματικοί κόμβοι εξυπηρετούνται από κάποιο μεγάλο υπολογιστικό "κέντρο". Σαν παράδειγμα αναφέρουμε τα δίκτυα που χρησιμοποιούν συχνά οι τράπεζες για την επίτευξη της επικοινωνίας μεταξύ υποκαταστημάτων και "μηχανογραφικού" κέντρου.

2. **Κατανεμημένα Δίκτυα.** Οι υπολογιστικές δυνατότητες είναι μοιρασμένες σε πολλά υπολογιστικά συστήματα.

Ομοίως με βάση την τοπολογία εντοπίζουμε τις εξής κατηγορίες :

1. **Τοπολογία Δένδρου.** Οι κόμβοι είναι διατεταγμένοι σε σχηματισμό δένδρου χωρίς δυνατότητες εναλλακτικής δρομολόγησης. Συνήθως μια τέτοια τοπολογία συνδυάζεται με κεντροποιημένη αρχιτεκτονική.
2. **Κατανεμημένη Τοπολογία (mesh topology).** Ο γράφος του δικτύου περιλαμβάνει βρόχους με δυνατότητα αξιόπιστης εναλλακτικής δρομολόγησης. Συνήθως μια τέτοια τοπολογία συνδυάζεται με κατανεμημένη αρχιτεκτονική.
3. **Τοπολογία Αρτηρίας (Bus) και Λακτυλίου (Ring).** Απαντώνται σε τοπικά και μητροπολιτικά δίκτυα και δίνουν δυνατότητες πολλαπλής πρόσβασης στο μέσο.

Σε μεγάλα δίκτυα υπολογιστών, διακρίνουμε ιεραρχίες παρόμοιες με αυτή του τηλεφωνικού δικτύου. Συνήθως η διάκριση γίνεται μεταξύ δικτύου κορμού (**backbone network**) με κατανεμημένη τοπολογία και επιπέδου τοπικής προσπέλασης των χρηστών (**local access network**).

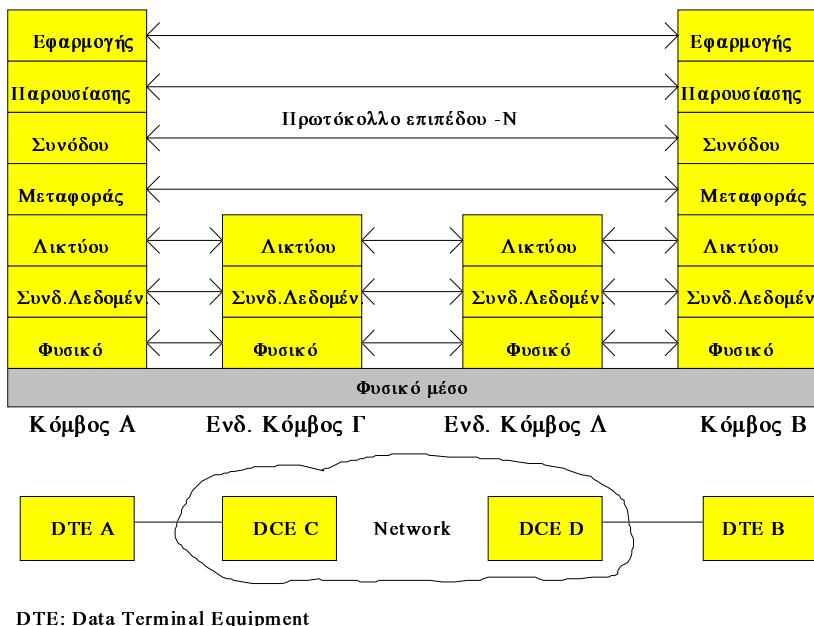
Η επικοινωνία στα δίκτυα υπολογιστών επιτυγχάνεται με την ανταλλαγή μηνυμάτων. Παρ' όλα αυτά υπάρχουν τρεις διαφορετικές τεχνικές για την μετάδοση των μηνυμάτων μέσα από το δίκτυο. Αυτές είναι οι παρακάτω:

- **Η μεταγωγή κυκλώματος (Circuit Switching).** Σύμφωνα με αυτήν, ένα πλήρες φυσικό κύκλωμα αποκαθίσταται μεταξύ των δύο χρηστών αποκλειστικά και στην συνέχεια όλη η πληροφορία μεταφέρεται μέσα από το κύκλωμα αυτό. Η τεχνική αυτή έχει ευρεία χρήση στα τηλεφωνικά δίκτυα. Το φανερό μειονέχτημά της είναι η χαμηλή χρησιμοποίηση του μέσου μεταφοράς, μια και η διαθέσιμη χωρητικότητα του εγκαταστημένου κυκλώματος μένει ανεκμετάλευτη ακόμη και όταν οι χρήστες δεν ανταλλάσουν μηνύματα.
- **Η μεταγωγή με μηνύματα.** Σύμφωνα με αυτήν, ένα μήνυμα μεταδίδεται από κόμβο σε κόμβο, μέχρι να φθάσει στον προορισμό του, σύμφωνα με οδηγίες που μεταφέρει. Κάθε ενδιάμεσος κόμβος, αποθηκεύει όλο το μήνυμα και το προωθεί σε επόμενο κόμβο (store and forward). Στην μεταγωγή με μηνύματα τα κανάλια επικοινωνίας δεν προορίζονται για αποκλειστική χρήση ενός χρήστη και έτσι έχουμε καλύτερη χρησιμοποίησή τους. Κλασσικό παράδειγμα αποτελεί η υπηρεσία ηλεκτρονικού ταχυδρομείου.
- **Η μεταγωγή με πακέτα.** Στην μεταγωγή με πακέτα η πληροφορία οργανώνεται σε πακέτα με στοιχεία αποστολέα και παραλήπτη. Τα πακέτα δρομολογούνται με διάφορους τρόπους (π.χ στατική δρομολόγηση, δυναμική δρομολόγηση datagram, εναλλακτική δρομολόγηση νοητού κυκλώματος virtual circuit). Οι αλγόριθμοι δρομολόγησης, με περιορισμούς στην επιτρεπτή πολυπλοκότητα και κόστος υλοποίησης, επιδιώκουν τη βελτιστοποίηση παραμέτρων επίδοσης του δικτύου όπως αποφυγή συμφόρησης, αξιοπιστία και ταχύτητα στη μετάδοση από άκρο σε άκρο. Η μεταγωγή πακέτου είναι εξέλιξη της μεταγωγής μηνύματος για περιβάλλοντα που απαιτούν μικρό χρόνο απόκρισης αλλά επιτρέπουν μικρές στατιστικές διαταραχές στο συγχρονισμό. Για απόλυτο συγχρονισμό (διαφάνεια) απαιτείται μεταγωγή κυκλώματος με τα συνεπαγόμενα μειονεκτήματα (κακή χρησιμοποίηση του δικτύου). Αξίζει να σημειωθεί οτι ακόμη και για εφαρμογές με απαιτήσεις συγχρονισμού (φωνή, video κ.λ.π) σε δίκτυα υψηλής ταχύτητας, η

τάση είναι να χρησιμοποιείται μεταγωγή πακέτου μικρού σταθερού μήκους (cell switching, Asynchronous Transfer Mode - ATM).

## 1.2. Πρωτόκολλα δικτύων υπολογιστών

Για να ελαττωθεί η πολυπλοκότητα της σχεδίασης, συνηθίζεται η αρχιτεκτονική του λογισμικού (αλλά και του υλικού) στα περισσότερα δίκτυα να σχεδιάζεται κατά επίπεδα (Layers). Σύμφωνα με την αρχιτεκτονική αυτή, όπως φαίνεται και στο Σχήμα 1.2, το επίπεδο-N κάποιου μηχανήματος επικοινωνεί με το επίπεδο-N ενός άλλου μηχανήματος. Οι κανόνες και οι συμβάσεις που ακολουθούνται κατά την διάρκεια της επικοινωνίας αυτής, είναι γνωστές ως το πρωτόκολλο του επιπέδου N. Μεταφορικά θα μπορούσαμε να πούμε ότι τα δύο ομότιμα επίπεδα μιλάνε την ίδια γλώσσα (π.χ διευθυντής προς διευθυντή). Στην πραγματικότητα, βέβαια, δεν μεταφέρονται απευθείας κάποια δεδομένα από το N επίπεδο κάποιου μηχανήματος στο N επίπεδο ενός άλλου. Αντίθετα, κάθε επίπεδο χρησιμοποιεί τις υπηρεσίες του αμέσως χαμηλότερου επιπέδου (π.χ του υφισταμένου του διευθυντή) έτσι ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα. Τελικά, θα χρησιμοποιηθούν οι υπηρεσίες του φυσικού επιπέδου, το οποίο θα προσφέρει την πραγματική (φυσική) επικοινωνία μεταξύ των δύο μηχανημάτων.



## Σχήμα 1.2 - Λιαστρωμάτωση στη σχεδίαση δικτύων

Ο Διεθνής Οργανισμός Τυποποίησης ISO (International Standards Organization) προτείνει ως μοντέλο αναφοράς πρωτοκόλλων το μοντέλο OSI (Open Systems Interconnection, OSI), που διέπει την ανοικτή διασύνδεση μεταξύ ετερογενών συστήματων. Αποτελείται από επτά καλά καθορισμένα επίπεδα. Στο Κεφάλαιο 2 θα εξετάσουμε συνοπτικά τα επίπεδα του OSI μοντέλου.

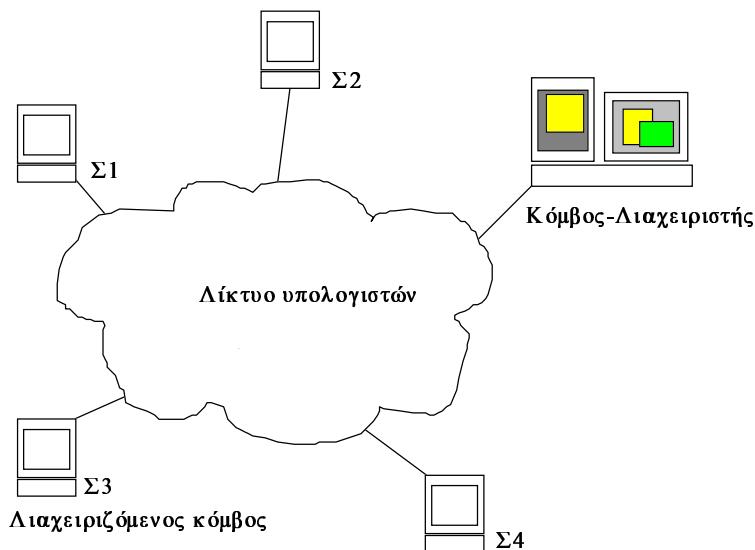
### 1.3. Το πρόβλημα της διαχείρισης

Η αύξηση του αριθμού, του μεγέθους και της πολυπλοκότητας ετερογενών δικτύων υπολογιστών, η μεγάλη διάδοση της χρήσης κατανεμημένων συστημάτων, η διασύνδεση των δικτύων σε διεθνές επίπεδο και οι ανάγκες για ποιότητα στην εξυπηρέτηση του χρήστη (Quality Of Service, QoS), αποδοτικότητα (Efficiency), και ασφάλεια (Security), είναι παράγοντες που εντείνουν την ανάγκη για αποτελεσματική διαχείριση δικτύων.

Συγκεκριμένα απαιτούνται διαχειριστικές πλατφόρμες με τα εξής χαρακτηριστικά: Φιλική παρουσίαση, αξιοπιστία, πολυεπίπεδη διαχείριση πρωτοκόλλων, ενοποίηση διαχειριστικών λειτουργιών υποδικτύων διαφορετικών τεχνολογιών, δυνατότητα χρήσης των πληροφοριών διαχείρισης για σχεδιαστικούς σκοπούς, συνδυασμός με εργαλεία διαχείρισης λειτουργικών συστημάτων, τεχνικο-οικονομική σκοπιμότητα (κόστος - λειτουργικότητα) και το σημαντικότερο, **ανοικτές** υλοποίησεις βασισμένες σε **διεθνή πρότυπα** (standards).

### 1.4. Μοντέλο και πρότυπα διαχείρισης

Η διαχείριση δικτύων μπορεί να περιγραφεί με βάση το εξής μοντέλο: Σε ένα δίκτυο υπάρχει ένας αριθμός από διαχειριζόμενους κόμβους. Καθένας από αυτούς, εκτός από τις εφαρμογές του χρήστη, εκτελεί ένα ειδικό πρόγραμμα που ονομάζουμε **αντιπρόσωπο (agent)**. Το πρόγραμμα αυτό παρακολουθεί και καταγράφει πληροφορίες σχετικές με τον κόμβο αυτό. Επιπλέον υπάρχει τουλάχιστον ένας **κόμβος-διαχειριστής (manager)** και ένα πρωτόκολλο επικοινωνίας μέσω του οποίου ανταλλάσσονται διαχειριστικές πληροφορίες μεταξύ του κόμβου-διαχειριστή και των διαχειριζόμενων κόμβων. Το μοντέλο αυτό απεικονίζεται στο Σχήμα 1.3.



**Σχήμα 1.3 - Μοντέλο διαχείρισης δικτύου**

Σύμφωνα με το παραπάνω μοντέλο επιτυγχάνεται οικονομία στην υπολογιστική ισχύ των διαχειριζόμενων κόμβων και λιτότητα στο λογισμικό που υλοποιεί τα πρωτόκολλα διαχείρισης. Ο υπολογιστικός φόρτος μεταφέρεται στο κόμβο-διαχειριστή.

Σε κάθε διαχειριζόμενο κόμβο, ο τοπικός agent διατηρεί και ενημερώνει ένα σύνολο μεταβλητών που δίνουν πληροφορίες διαχειριστικού περιεχομένου. Το σύνολο των μεταβλητών αυτών ονομάζεται Βάση Πληροφορίας Διαχείρισης (Management Information Base - MIB). Ο αριθμός των μεταβλητών που πρέπει να διατηρεί κάθε κόμβος, η ονομασία τους και η χρήση τους στη διαχείριση καθορίζεται με σαφήνεια από τον ορισμό του MIB. Η παρακολούθηση ενός κόμβου γίνεται με την πρόσβαση στις μεταβλητές που είναι καταχωρημένες στά MIB των κόμβων. Ο έλεγχος των κόμβων επιτυγχάνεται με την εγγραφή τιμών στις μεταβλητές αυτές. Η αναφορά ασυνήθιστων καταστάσεων από τις συσκευές του δικτύου στο διαχειριστή παραμένει στην υπεύθυνότητα του agent.

Τα σημαντικότερα πρότυπα διαχείρισης έχουν ορισθεί για δίκτυα TCP/IP (με αρχική προέλευση το Αμερικανικό Υπουργείο Εθνικής Αμυνας) και για δίκτυα του Διεθνούς Οργανισμού Τυποποίησης ISO/OSI .

Στο TCP/IP το πρωτόκολλο επικοινωνίας των διαχειριστικών πληροφοριών λέγεται SNMP (Simple Network Management Protocol). Αξίζει να σημειωθεί ότι εδώ δεν καθορίζονται άμεσα διαχειριστικές λειτουργίες σε αντίθεση με το πρότυπο OSI. Εμμεσα οι λειτουργίες περιορίζονται από τις πληροφορίες που συλλέγει και μεταδίδει ο agent. Στο OSI, το πρωτόκολλο επικοινωνίας ονομάζεται CMIP (Common Management Information Protocol) και οι διαχειριστικές λειτουργίες είναι σαφώς ορισμένες. Πιο ειδικά στο OSI ορίζεται ένα σύνολο από ομάδες λειτουργιών διαχείρισης, συχνά επικαλυπτόμενες, και ένα μοντέλο για τη συνολική δομή της διαχειριστικής πλατφόρμας.

Οι ομάδες των αναγκαίων διαχειριστικών λειτουργιών είναι:

- **Διαχείριση διάρθρωσης (Configuration Management)**, η οποία είναι υπεύθυνη για την απεικόνηση της δομής και της κατάστασης των διαφόρων λογικών και φυσικών στοιχείων του δικτύου.
- **Διαχείριση επιδόσεων (Performance Management)**, η οποία είναι υπεύθυνη για τον έλεγχο και την ανάλυση των ρυθμών απόδοσης και λαθών του δικτύου.
- **Διαχείριση βλαβών (Fault Management)**, η οποία είναι υπεύθυνη για τον εντοπισμό, απομόνωση και έλεγχο στοιχείων του δικτύου που παρουσιάζουν βλάβες.
- **Λογιστική διαχείριση (Accounting Management)**, η οποία είναι υπεύθυνη για την συλλογή και επεξεργασία στοιχείων σχετικών με τη χρήση των πόρων του δικτύου.
- **Διαχείριση ασφαλείας (Security Management)**, η οποία είναι υπεύθυνη για τον περιορισμό της πρόσβασης στους πόρους του δικτύου και την προστασία των δεδομένων.

Για την επίλυση των παραπάνω υποπροβλημάτων σύμφωνα με το OSI, το πρωτόκολλο διαχείρισης πρέπει να προσφέρει τρείς υπηρεσίες:

- **Παρακολούθηση (monitoring)** δηλαδή συλλογή των πληροφοριών διαχείρισης από το δίκτυο.
- **Έλεγχο (control)** δηλαδή χειρισμό των συσκευών του δικτύου.
- **Αναφορά (reporting)** δηλαδή μηχανισμό μέσω του οποίου οι συσκευές του δικτύου αναφέρουν προβλήματα κατά τη λειτουργία τους.

## 1.5. Τρόποι διακίνησης πληροφοριών διαχείρισης

Η διακίνηση των πληροφοριών διαχείρισης μπορεί να χρησιμοποιεί το υπό διαχείριση δίκτυο:

- **Λιαχειριστικά μηνύματα διακινούνται μαζί με το τηλεπικοινωνιακό φορτίο χρησιμοποιώντας τα ίδια κανάλια και τα ίδια πρωτόκολλα (In-channel, In-band).** Παράδειγμα αποτελεί η διαχείριση δικτύων TCP/IP όπου οι πληροφορίες μεταδίδονται από τους διαχειριζόμενους κόμβους στο διαχειριστή με πακέτα δεδομένων IP.

Οι πληροφορίες διαχείρισης έχουν ιδιαίτερη σημασία για τον έγκαιρο εντοπισμό προβλημάτων και την επιδιόρθωση βλαβών. Αρα, η επικοινωνία του διαχειριστή με τα στοιχεία του δικτύου πρέπει να είναι αξιόπιστη με πρόβλεψη εναλλακτικών τρόπων σε περιπτώσεις προβληματικής λειτουργίας (μεγάλες καθυστερήσεις, βλάβες). Αφ' ετέρου, δεν πρέπει το φορτίο διαχείρισης να δημιουργεί προβλήματα στην διακίνηση των πληροφοριών, όπως π.χ. με τη συχνή ανταλλαγή διαχειριστικών μηνυμάτων στο ίδιο κανάλι δεδομένων. Για τους λόγους αυτούς προτείνονται δύο άλλοι τρόποι διακίνησης, που όμως εισάγουν πρόσθετη πολυπλοκότητα και κόστος.

- **Μέσω παραπλεύρου καναλιού (Side-Channel) χωριστά από το τηλεπικοινωνιακό φορτίο χρησιμοποιώντας πολύπλεξη χρόνου ή συχνότητας στα ίδια κανάλια [TERP87].** Παράδειγμα αποτελεί η διακίνηση πληροφοριών για έλεγχους διαμορφωτών (modem loop-back tests, fall-back commands) και πολυπλεκτών (multiplexers).
- **Σε ξεχωριστό φυσικό δίκτυο.** Παράδειγμα αποτελούν: (1) Η χρήση του επιλεγόμενου δημοσίου τηλεφωνικού δικτύου μέσω dial-up modem (κυρίως για εναλλακτική επικοινωνία σε περιπτώσεις διακοπής του υπό διαχείριση δικτύου), (2) η χρήση αξιοπίστου δικτύου μεταγωγής πακέτου X.25 π.χ. για διαχείριση μεγάλων ενοποιημένων τηλεπικοινωνιακών δικτύων με διαχειριστικά συστήματα ISO/OSI.

## 1.6. Βιβλιογραφία

- [BRAG90] Braga G., Garaffini G., Liserve V., *Τηλεπικοινωνίες σήμερα και αύριο. Τάσεις και ευκαιρίες, απόδοση στα ελληνικά*: Β.Ξιάρχος, Δελτίο Ι.Ε.Δ.Μ.-Η. Σεπτ.1991.
- [COME91] Comer D.E., *Internetworking with TCP/IP Volume I; Principles, Protocols, and Architecture*, Second Edition, Prentice Hall, N.J., 1991.
- [HALS92] Halsall F., *Data Communications, Computer Networks and Open Systems*, 3rd Edition, Addison-Wesley, 1992.
- [ΠΙΡΩΤ88] Πρωτονοτάριος Ε.Ν., Συκάς Ε.Δ., Αναγνώστου Μ.Ε. και Πάσχος Σ.Θ., *Δίκτυα Υπολογιστών, Σημειώσεις*, Αθήνα 1988.
- [ΣΤΑΣ89] Στασινόπουλος Γ., *Ψηφιακά Συστήματα Επικοινωνιών*, Ε.Μ.Πολυτεχνείο, Τμήμα Ηλεκτρολόγων, Αθήνα 1989.
- [TANB91] Tanenbaum A.S., *Δίκτυα Υπολογιστών*, Δεύτερη Έκδοση, Prentice Hall, για την Ελληνική Έκδοση Παπασωτηρίου 1991.

[TERP87] K.Terplan, Communication Networks Management, Prentice Hall, 1987.

## Κεφάλαιο 2

### 2. Τα Πρωτόκολλα TCP/IP

#### Περιεχόμενα του Κεφαλαίου 2

- 2.0. Εισαγωγή στα πρωτόκολλα TCP/IP και το INTERNET
- 2.1. Μέσα μεταφοράς, φυσικές διευθύνσεις
- 2.2. Το πρωτόκολλο IP
  - 2.2.1. Κατακερματισμός και επανασύνδεση (Fragmentation and reassembly)
  - 2.2.2. IP-διευθύνσεις
  - 2.2.3. Η διευθέτηση των υποδικτύων
  - 2.2.2. Το πρωτόκολλο ICMP
- 2.3. Δρομολόγηση
  - 2.3.1. Αυτόνομα Συστήματα
  - 2.3.2. Address Resolution Protocol (ARP) - Reverse Address Resolution Protocol (RARP)
  - 2.3.3. Interior Gateway Protocol (IGP): RIP - OSPF. Παραδείγματα
  - 2.3.4. Exterior Gateway Protocol (EGP)
  - 2.3.5. Αλγόριθμοι εύρεσης ελαχίστων δρόμων: Bellman-Ford, Dijkstra, Floyd-Warshall. Παραδείγματα
- 2.4. Πρωτόκολλα επιπέδου μεταφοράς
  - 2.4.1. Το πρωτόκολλο TCP
    - 2.4.1.1. Αξιόπιστη υπηρεσία μεταφοράς - stream
    - 2.4.1.2. Η λειτουργία του πρωτοκόλλου
  - 2.4.2. Το πρωτόκολλο μεταφοράς UDP
- 2.5. Εφαρμογές TCP/IP
- 2.6. Προγραμματισμός με sockets στο Unix. Παραδείγματα
- 2.7. Ασκήσεις
- 2.8. Βιβλιογραφία

#### 2.0. Εισαγωγή στα πρωτόκολλα TCP/IP και το INTERNET

Τα πρωτόκολλα TCP/IP καθορίστηκαν σαν de-facto standards μετάδοσης δεδομένων σε τοπικά δίκτυα υπολογιστών και διασυνδεμένα δίκτυα μέσω **δρομολογητών, routers**. Αναπτύχθηκαν αρχικά με συγχρηματοδότηση του Υπουργείου Εθνικής Αμυνας των ΗΠΑ (US DOD - Department of Defense, Advanced Research Projects

Administration - ARPA) στη δεκαετία του 1970. Στη συνέχεια, νιοθετήθηκαν από τη βιομηχανία τηλεπληροφορικής (κατασκευαστές υπολογιστών) και τη κοινότητα ανάπτυξης λογισμικού δικτύων client-server γύρω από το Λειτουργικό Σύστημα **UNIX** (*Berkeley UNIX - BSD, AT&T - UNIX V*).

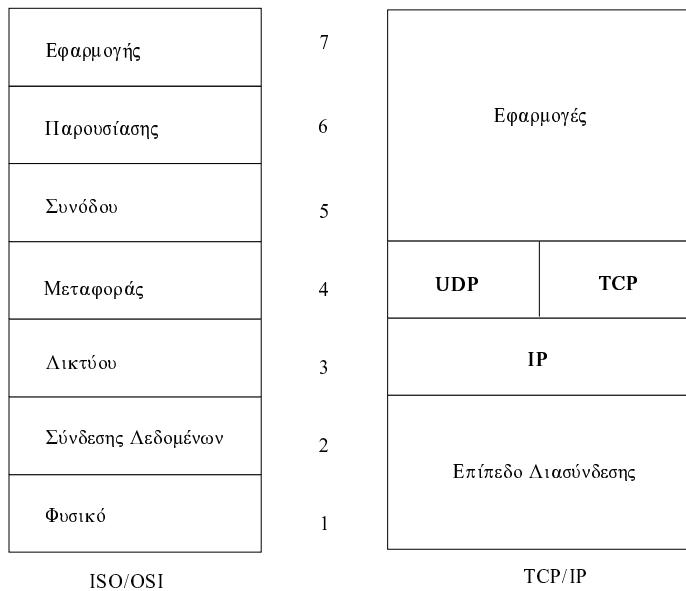
Από το 1970 αποτέλεσαν τα πρωτόκολλα μετάδοσης στο **ARPANET**, το πρώτο πειραματικό δίκτυο ευρείας αποστάσεως (Wide Area Network - WAN) με μεταγωγή πακέτου (Packet Switching). Σήμερα το ARPANET έχει αντικατασταθεί από το ταχύτερο και μεγαλύτερο **NSFNET** στις ΗΠΑ και επεκτείνεται στο γνωστό σε όλους διεθνές **INTERNET**. Το INTERNET διασυνδέει εκατομμύρια υπολογιστές στον κόσμο για **ερευνητικούς στόχους** και έχει φέρει επανάσταση στη κοινωνία της πληροφόρησης. Αναπτύσσεται ταχύτατα και μάλλον άναρχα, στη βάση συμφωνίας ενός Ακαδημαϊκού, Ερευνητικού, Κρατικού ή Βιομηχανικού φορέα με τη κεντρική αρχή διαχείρισης (Internet Network Information Center - InterNIC) ή πληρεξούσιους Εθνικούς ορείς (Local Internet Registries - IR). Κύρια κεντρική οικονομική υποχρέωση είναι η κάλυψη της μίσθωσης τηλεπικοινωνιακών γραμμών προς τους πλησιέστερους κόμβους άλλων φορέων. Το εσωτερικό δίκτυο του Ε.Μ.Π. είναι τμήμα του INTERNET σε συνενόηση με το ITE (Ηράκλειο Κρήτης, **FORTHNET**) και το ΕΚΕΦΕ-ΔΗΜΟΚΡΙΤΟΣ (**ARIADNE-T**).

Το INTERNET βασίζεται στη διασύνδεση Τοπικών Δικτύων (Local Area Networks - LAN) μέσω απλών μεταγωγέων πακέτου, **routers** (δρομολογητές) ή **gateways** (θύρες) που εκτελούν δρομολόγηση, καθορίζόμενη από τα σχετικά πρωτόκολλα **IP (Internet Protocols)**. Η αξιοπιστία της μετάδοσης πακέτων (**datagrams**) καθορίζεται από τους τελικούς (ακραίους) χρήστες με τα πρωτόκολλα **TCP** ή **UDP**. Οι τελικοί χρήστες (υπολογιστές UNIX ή PC) μπορεί να κτίσουν ανώτερες λειτουργίες βασισμένες στις υπηρεσίες TCP/IP: Ενοποίηση συστημάτων αρχείων σε τοπικό δίκτυο (**NFS**), μεταφορά αρχείων δια μέσου του INTERNET (**FTP**), πρόσβαση σε απομακρυσμένους υπολογιστές (**RLOGIN, TELNET**), ηλεκτρονικό ταχυδρομείο (**E-MAIL, SMTP**), διαχείριση κόμβων - υπολογιστών, routers και υποδικτύων (**SNMP**), πρόσβαση σε υπηρεσίες πληροφόρησης (**GOPHER, WORLD-WIDE-WEB, X-MOSAIC**) κλπ. Μερικές λεπτομέρειες για τα πρωτόκολλα TCP, UDP και IP αναφέρονται στις επόμενες παραγράφους.

Για την συμβατή λειτουργία σε περίπτωση διασύνδεσης με το INTERNET απαιτείται διαδικασία έγκρισης των αναπτυσσομένων προτύπων με την έκδοση και συζήτηση (πάντα μέσω του δικτύου) των λεγομένων **Requests for Comments - RFC**. Την διαδικασία παρακολουθεί η κεντρική αρχή, το **Internet Architecture Board - IAB**. Ορισμένα RFC's ορίζονται μετά από συμφωνία σαν υποχρεωτικά standards τα οποία πρέπει να ακολουθούν προμηθευτές hardware - software και χρήστες του INTERNET.

Τα πρωτόκολλα TCP/IP (de-facto standards), όπως και τα πρωτόκολλα ISO/OSI (διεθνή standards) βασίζονται στην αρχιτεκτονική των στρωμάτων. Η μεθοδολογία της διαστρωμάτωσης είναι μια βασική τεχνική στην υλοποίηση των πρωτοκόλλων επικοινωνίας. Κατανέμει τη δυσκολία της μετάδοσης πληροφορίας σε περισσότερες από μία μονάδες λογισμικού-υλικού που ονομάζονται επίπεδα (ή στρώματα). Τα υψηλότερα επίπεδα μπορούν να χειρίζονται πιο πολύπλοκες μορφές πληροφορίας και πιο αφηρημένες έννοιες, ενώ τα χαμηλότερα απλούστερες και πιο κοντά στο επίπεδο μέσου. Κάθε ένα επίπεδο καθώς ανεβαίνουμε προς τα πάνω χρησιμοποιεί τις υπηρεσίες του αμέσως κατώτερου και προσφέρει υπηρεσίες πολυπλοκότερης μορφής στο αμέσως ανώτερό του.

Στο Σχήμα 2.1 φαίνονται τα διαδοχικά επίπεδα στις οικογένειες προτύπων OSI/ISO και TCP/IP και οι αντιστοιχίες των επιπέδων μεταξύ των δύο διαφορετικών οικογενειών προτύπων.



## Σχήμα 2.1 - Αντιστοιχία πρωτόκολλων ISO/OSI και TCP/IP

Στο Σχήμα 2.2 βλέπουμε αναλυτικότερα μια πιθανή στοίβα πρωτοκόλλων TCP/IP με τα οποία θα ασχοληθούμε αρχικά. Το IP (Internet Protocol) αντιστοιχεί στο επίπεδο δικτύου του μοντέλου αναφοράς OSI. Πρόκειται για μη αξιόπιστη υπηρεσία δικτύωσης χωρίς σύνδεση με κύρια ευθύνη τη δορυφολόγηση πακέτων δεδομένων (datagram data packets) δια μέσου των κόμβων ενός διαδικτύου: **Υπολογιστές - hosts, δρομολογητές- routers, θύρες - gateways**. Το TCP (Transmission Control Protocol) και το UDP (User Datagram Protocol) αντιστοιχούν στο επίπεδο μεταφοράς του OSI (δηλαδή είναι υπεύθυνα για την μεταφορά και απόδοση μηνυμάτων από την πηγή στον προορισμό). Το TCP προσφέρει αξιόπιστη υπηρεσία μεταφοράς μέσω σύνδεσης (reliable, connection-oriented service), ενώ το UDP προσφέρει μη αξιόπιστη μεταφορά χωρίς σύνδεση (unreliable connectionless service). Και τα δύο αυτά πρωτόκολλα μεταφοράς χρησιμοποιούν τις υπηρεσίες του IP για τον κατακερματισμό των μηνυμάτων σε πακέτα και την δρομολόγηση των πακέτων αυτών διαμέσου των κόμβων ενός δικτύου. Στην περίπτωση του UDP, μηνύματα μπορούν να χαθούν, ή να φτάσουν σε πολλαπλά αντίγραφα ή με λάθος σειρά, γι'αυτό η εφαρμογή που χρησιμοποιεί τις υπηρεσίες του UDP είναι υπεύθυνη να λύσει τα προβλήματα αυτά. Αντίθετα εφαρμογές που χρησιμοποιούν το TCP βασίζονται στις αυξημένες δυνατότητές του (έλεγχος ροής - flow-control, αλγόριθμοι για αποφυγή συμφόρησης, μηχανισμοί αναμετάδοσης σε περιπτώσεις μεγάλης καθυστέρησης, κλπ.), και αδιαφορούν για τον τρόπο που το μήνυμα θα μεταφερθεί στον προορισμό. Απλές εφαρμογές, με μικρό μέγεθος μηνύματος, ανοχές ως προς τα προαναφερθέντα προβλήματα, και ανάγκη "συνομιλίας" με πολλούς κόμβους συγχρόνως (όπως π.χ. το SNMP) προτιμούν το UDP ως πρωτόκολλο μεταφοράς γιατί η περιορισμένη του λειτουργικότητα ευνοεί την ταχύτητα μετάδοσης, ενώ εφαρμογές με μεγάλο όγκο μηνυμάτων και μηδενικές ή ελάχιστες ανοχές ως προς την αλλοιώση της πληροφορίας (όπως η μεταφορά αρχείων-FTP) έχουν ανάγκη τις υπηρεσίες του TCP.

		Άλλες υπηρεσίες				
Εφαρμογής	FTP, SMTP TELNET CMOT, ...	rlogin rep, ...	NFS, YP ...	TFTP DNS SNMP ...	PING	
Μεταφοράς	TCP		UDP			
Δικτύου	Internet Protocol (IP)			ICMP	ARP RARP	
Γραμμής Πρόσβασης και Φυσικό	( IEEE 802.2 Logical Link )					
	( IEEE 802.1 Bridging )					
IEEE 802.3	IEEE 802.4	IEEE 802.5	IEEE 802.6			
MAC	MAC	MAC	MAC			
Ethernet	Token Bus	Token Ring	MAN			

FTP: File Transfer Protocol

SMTP: Simple Mail Transfer Protocol

TELNET: Virtual Terminal

CMOT: CMIP over TCP/IP

rlogin: Remote Login (Unix application)

rep: Remote Copy (Unix application)

NFS: Network File System (Unix application)

YP: Yellow Pages (Unix application)

TFTP: Trivial File Transfer Protocol

DNS: Domain Name Transfer

SNMP: Simple Network Management Protocol

PING: Packet Internet Groper

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

IP: Internet Protocol

ICMP: Internet Control Message Protocol

ARP: Address Resolution Protocol

RARP: Reverse ARP

MAC: Media Access Control

MAN: Metropolitan Area Network

**ΣΗΜΕΙΩΣΕΙΣ:** Σε υλοποιήσεις TCP/IP πρωτοκόλλων πάνω από τοπικά δίκτυα Ethernet, χρησιμοποιείται αντί των επιπέδων IEEE802.3, IEEE802.2 το πρωτόκολλο ETHERNET II, που είναι παραλλαγή του IEEE802.3

Τα πρωτόκολλα IEEE802.1 ορίζουν τρόπους γεφύρωσης ομοίων υποδικτύων (πχ. Ethernet)

Τα πρωτόκολλα IEEE802.2 (Logical Link Control) επιτρέπουν τη γεφύρωση υποδικτύων διαφορετικών μεθόδων MAC.

## Σχήμα 2.2 - Μια τυπική στοίβα πρωτοκόλλων TCP/IP

Στο Σχήμα 2.2. παρατηρούμε ακόμα το πρωτόκολλο **ICMP (Internet Control Message Protocol)** το οποίο αποτελεί ένα ουσιαστικό κομμάτι του πρωτοκόλλου IP, το οποίο χειρίζεται μηνύματα λαθών και ελέγχου. Πιο συγκεκριμένα, δρομολογητές και σταθμοί εργασίας χρησιμοποιούν το πρωτόκολλο ICMP προκειμένου να αναφέρουν προβλήματα σχετικά με IP datagrams πίσω στην αρχική πηγή που έστειλε το datagram. Το ICMP επίσης χρησιμοποιεί μηνύματα echo request/reply προκειμένου να διαπιστώσει αν κάποιος κόμβος είναι τόσο σε λειτουργία, όσο και προσιτός. Η εφαρμογή **PING (Packet InterNet Groper)** είναι η μοναδική που έχει πρόσβαση κατ' ευθείαν στο επίπεδο δικτύου, χρησιμοποιώντας τα ICMP request μηνύματα. Άλλα πρωτόκολλα στο επίπεδο δικτύου είναι τα **ARP (Address Resolution Protocol)** και **RARP (Reverse Address Resolution Protocol)** τα οποία χρησιμοποιούνται για τη μετάφραση μεταξύ των φυσικών διευθύνσεων και των IP διευθύνσεων.

Τέλος κάτω από τη στοίβα πρωτοκόλλων TCP/IP παρατηρούμε διάφορα πρωτόκολλα Τοπικών δικτύων LAN (Ethernet IEEE 802.3, Token Bus IEEE 802.4, ...) τα οποία

χρησιμοποιούνται για τη πραγματική μεταφορά της πληροφορίας (επίπεδα 1 & 2 στο μοντέλο αναφοράς OSI). Αυτά τα εξετάσουμε στην επόμενη παράγραφο. Σε περίπτωση μεταφοράς μεταξύ απομακρυσμένων Τοπικών Δικτύων (LAN) χρησιμοποιούνται αντίστοιχα πρωτόκολλα WAN (HDLC/X.25, Point-to-Point Protocol - PPP, Serial Line IP - SLIP, ... ).

## 2.1. Μέσα μετάδοσης, φυσικές διευθύνσεις

### A. Μέσα Μετάδοσης

Θα ξεκινήσουμε την ανάπτυξη της παραγράφου αυτής, αναφέροντας κάποια πράγματα σχετικά με τα μέσα μετάδοσης, τα οποία χρησιμοποιούνται με σκοπό τη δικτύωση υπολογιστών. Πιο συγκεκριμένα θα εξετάσουμε πρότυπα που καθορίζουν τεχνολογίες, πάνω από τις οποίες μπορούν να τρέξουν τα πρωτόκολλα TCP/IP. Θα εξετάσουμε τεχνολογίες που προτείνονται από την IEEE, καθώς και το FDDI της ANSI.

#### 1. IEEE 802.3

Το πρότυπο αυτό ορίζει τον Carrier Sense Multiple Access / Collision Detection (CSMA/CD) τρόπο προσπέλασης του μέσου. Βασίζεται σε δουλειά που έγινε από την Xerox Corporation και ονομάζεται **Ethernet**.

Το IEEE 802.3 καθορίζει τόσο το φυσικό επίπεδο (Physical Layer) όσο και το επίπεδο προσπέλασης του μέσου (Medium Access Control layer). Το πρώτο πρότυπο εμφανίστηκε το 1985 και καθόριζε ως φυσικό μέσο  $50\Omega$  ομοαξονικό καλώδιο 10Mbps βασικής ζώνης, με μέγιστο μήκος τμήματος 500m. Από τότε ακολούθησαν πολλά άλλα συμπληρώματα στο αρχικό αυτό πρότυπο, όπως για παράδειγμα:

- 1BASE2 1Mbps baseband Thin Ethernet (Cheapnet)
- 1BASE5 1Mbps baseband Ethernet (StarLAN)
- 10BASE5 10Mbps baseband Ethernet
- 10BASEF 10Mbps Ethernet on Optical Fiber
- 10BASET 10Mbps baseband Ethernet on Twisted Pair
- 10BROAD36 10Mbps Broadband Ethernet

Η CSMA/CD μέθοδος προσπέλασης του μέσου χρησιμοποιείται κυρίως με δίκτυα τοπολογίας αρτηρίας (bus networks), όπως αυτά που καθορίζει το πρότυπο IEEE 802.3. Σύμφωνα με τη τοπολογία αυτή όλοι οι τερματικοί σταθμοί είναι απευθείας συνδεδεμένοι στο ίδιο καλώδιο. Όλα τα δεδομένα, που μεταδίδονται από κάποιο σταθμό εργασίας ενθηλακώνονται σε ένα πλαίσιο, στο οποίο τοποθετείται και η φυσική διεύθυνση του σταθμού προορισμού. Το πλαίσιο στην συνέχεια μεταδίδεται broadcast στο κοινό μέσο μεταφοράς. Όλοι οι άλλοι σταθμοί μπορούν και αντιλαμβάνονται το γεγονός, ότι κάποιος σταθμός μεταδίδει. Αν επιπλέον αναγνωρίσουν στη διεύθυνση προορισμού του πλαισίου τη διεύθυνσή τους, τότε συνεχίζουν να διαβάζουν και την υπόλοιπη πληροφορία που υπάρχει στο πλαίσιο. Στο πλαίσιο περιλαμβάνεται και η διεύθυνση του κόμβου γέννησης του μηνύματος, ώστε ο προορισμός να μπορεί να απαντήσει στο μήνυμα.

Με αυτόν τον τρόπο διευθέτησης του μέσου είναι δυνατόν δύο σταθμοί να προσπαθήσουν περίπου την ίδια στιγμή να μεταδόσουν στο δίκτυο. Στην περίπτωση αυτή η πληροφορία και των δύο θα καταστραφεί. Για να γίνει η πιθανότητα αυτή, όσο το δυνατόν μικρότερη, το CSMA/CD καθορίζει ότι πριν κάποιος σταθμός επιχειρήσει να μεταδώσει στο μέσο, ελέγχει αν κάποιος άλλος σταθμός μεταδίδει. Στην περίπτωση αυτή αναβάλει την μετάδοσή του, μέχρι την ολοκλήρωση της εξελισσόμενης μετάδοσης. Υπάρχει βέβαια η πιθανότητα, δύο σταθμοί ταυτόχρονα να διαπιστώσουν την αδράνεια του μέσου μεταφοράς και να επιχειρήσουν να μεταδόσουν. Στην περίπτωση αυτή η σύγκρουση είναι αναπόφευκτη, και είναι σημαντικό να γίνει αντιληπτή όσο το δυνατόν γρηγορότερα. Αυτό επιτυγχάνεται με έλεγχο από κάθε σταθμό του σήματος στο καλώδιο και σύγκριση αυτού με το σήμα που μεταδίδει. Αν τα σήματα αυτά είναι διαφορετικά τότε η σύγκρουση έχει ανιχνευθεί (collision detected). Στην περίπτωση σύγκρουσης η μετάδοση διακόπτεται αμέσως για εξοικονόμηση πόρων (π.χ. bandwidth) και επαναλαμβάνεται μετά από ένα κατάλληλα επιλεγμένο τυχαίο χρονικό διάστημα. Το Σχήμα 2.3 δείχνει την μορφή του 802.3 MAC πλαισίου.

PREAMBLE	SFD	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH	LLC DATA	PAD	FCS
7 octets	1 octets	2 or 6 octets	2 or 6 octets	2 octets			4 octets

### Σχήμα 2.3 - Η μορφή του 802.3 MAC πλαισίου

#### 2. IEEE 802.4

Το πρότυπο IEEE 802.4 καθορίζει το φυσικό επίπεδο και το επίπεδο ελέγχου προσπέλασης του μέσου για ένα σύστημα τοπολογίας αρτηρίας και ελέγχου προσπέλασης του μέσου με κουπόνι.

Το πρότυπο καθορίζει για το φυσικό μέσο τρία συστήματα κωδικοποίησης της πληροφορίας πάνω από ομοαξονικό καλώδιο  $75\Omega$ :

- Phase continuous Frequency Shift Keying (FSK) 1Mbps
- Phase coherent FSK 5,10Mbps
- Multi level duo-binary coding 5,10Mbps

Η μέθοδος ελέγχου της πρόσβασης στο μέσο μεταφοράς επιτυγχάνεται με τη χρησιμοποίηση ενός κουπονιού ελέγχου της πρόσβασης (control permission token). Αυτό το κουπόνι περνάει από σταθμό σε σταθμό σύμφωνα με κάποιο σύνολο από κανόνες, που αποδέχονται όλοι οι σταθμοί. Κάποιος σταθμός μπορεί να μεταδόσει ένα πλαίσιο, μόνο στην περίπτωση που έχει στην κατοχή του το κουπόνι, και μόλις ολοκληρώσει την μετάδοση του πλαισίου ελευθερώνει το κουπόνι, περνώντας το στον επόμενο σταθμό. Στην τεχνική αυτή εισάγεται η έννοια μιας λογικής σειράς στους σταθμούς, ώστε να γνωρίζει ο κάθε σταθμός τον επόμενό του, ειδικά στην περίπτωση δικτύων τοπολογίας αρτηρίας, όπου η λογική σειρά δεν είναι προφανής. Έτσι, κατά την αρχικοποίηση των συστημάτων εγκαθίσταται μια λογική συνδεσμολογία, και δημιουργείται ένα κουπόνι.

Λειτουργίες παρακολούθησης στους σταθμούς, που είναι συνδεδεμένοι στο κοινό μέσο παρέχουν υπηρεσίες αρχικοποίησης και επαναφοράς σε περίπτωση σφάλματος της λογικής συνδεσμολογίας. Αν και οι λειτουργίες επαναλαμβάνονται από όλους τους σταθμούς, κάθε χρονική στιγμή ένας σταθμός είναι υπεύθυνος για το δίκτυο. Το φυσικό μέσο δεν είναι απαραίτητο να έχει την τοπολογία δακτυλίου. Επιλέον υπάρχει και ένα μέγιστο δυνατό χρονικό διάστημα για το οποίο μπορεί να μεταδίδει ο σταθμός. Μετά το

πέρας αυτού ο σταθμός υποχρεωτικά ελευθερώνει το κουπόνι ακόμα και αν δεν έχει ολοκληρώσει τη μετάδοσή του. Προφανώς η μετάδοση αυτή θα συνεχιστεί όταν το κουπόνι επανέλθει στην κατοχή του συγκεκριμένου σταθμού. Η τεχνική ελέγχου προσπέλασης του μέσου με κουπόνι χρησιμοποιείται και σε δίκτυα τοπολογίας δακτυλίου. Στο Σχήμα 2.4 βλέπουμε την μορφή του MAC πλαισίου για το πρότυπο 802.4.

PREAMBLE	SD	FC	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH	DATA UNIT	FCS	ED
7 octets	1 octets each		2 or 6 octets	2 or 6 octets	2 octets	$\geq 0$ octets	4 octets	1 octet

### Σχήμα 2.4 - Η μορφή του 802.4 MAC πλαισίου

#### 3. IEEE 802.5

Το πρότυπο IEEE 802.5 καθορίζει το φυσικό επίπεδο καθώς και το επιπέδο ελέγχου προσπέλασης του μέσου για δίκτυα τοπολογίας δακτυλίου και ελέγχου προσπέλασης του μέσου με κουπόνι.

Το σύνολο των σταθμών που σχηματίζουν το token ring είναι σειριακά συνδεδεμένοι με το μέσο μεταφοράς, και η πληροφορία μεταφέρεται από τον ένα σταθμό στον άλλο.

Το IEEE 802.5 διαφέρει από το CSMA/CD και από το IEEE 802.4 από την άποψη, ότι τα bits σε κάθε πλαίσιο μεταδίδονται στο μέσο, ξεκινώντας από το πιο σημαντικό ψηφίο, πράγμα που δεν συμβαίνει στα δύο πρότυπα που εξετάσαμε προηγούμενα.

Το 802.5 καθορίζει σαν φυσικό μέσο, προστατευμένο ή απροστάτευτο ζευγάρι καλωδίων στο 1Mbps ή στα 4Mbps. Επιπλέον, η IBM έχει παρουσιάσει μια εξελιγμένη έκδοση του 802.5, η οποία χρησιμοποιεί προστατευμένα διπλά καλώδια με ρυθμό μετάδοσης 16Mbps. Στο Σχήμα 2.5 βλέπουμε την μορφή του MAC πλαισίου για το πρότυπο 802.5.

SD	AC	FC	DESTINATION ADDRESS	SOURCE ADDRESS	DATA UNIT	FCS	ED	FS
1 octets each			2 or 6 octets	2 or 6 octets	$\geq 0$ octets	4 octets	1 octet	1 octet

### Σχήμα 2.5 - Η μορφή του 802.5 MAC πλαισίου

#### 4. FDDI

Το πρότυπο αυτό είναι οργανωμένο σε τέσσερα μέρη, τα οποία είναι τα εξής:

- Medium Access Control (MAC)
- Physical Medium Dependent (PMD)
- Physical Layer Protocol (PHY)
- Station Management (SMT)

To Fiber Distributed Data Interface (FDDI) είναι ταχύτερο από τα μέσα μεταφοράς που εξετάσαμε παραπάνω, έχοντας μια πολύ μεγαλύτερη ικανότητα μεταφοράς πληροφορίας. Σ' αυτό βοηθάει και το γεγονός, ότι το πρότυπο αυτό επιτρέπει σε περισσότερα από ένα πλαίσια να βρίσκονται ταυτόχρονα σε μετάδοση. Υποστηρίζει 100Mbps μετάδοση πληροφορίας σε διπλό δακτύλιο από οπτική ίνα, που μπορεί να συνδέει μέχρι και 500 κόμβους τοποθετημένους σε αποστάσεις ακόμα και 2Km ο ένας από τον άλλο και για ένα συνολικό μήκος, μέχρι και 100Km.

Το FDDI μπορεί να χρησιμοποιηθεί σα δίκτυο κορμός για τη διασύνδεση άλλων τοπικών δικτύων, ή σαν τοπικό δίκτυο διασύνδεσης σταθμών εργασίας υψηλών ταχυτήτων με μεγάλες υπολογιστικές και άλλες ανάγκες. Επίσης χρησιμοποιείται για τη διασύνδεση back-end σταθμών εργασίας υψηλών προδιαγραφών για την υποστήριξη του συστήματος αρχείων.

## B. Φυσικές Λιευθύνσεις

Θα συνεχίσουμε διευκρινίζοντας κάποια θέματα σχετικά με τις διευθύνσεις κόμβων σ' ένα δίκτυο. Σαν κόμβο μπορούμε να θεωρήσουμε έναν μικροϋπολογιστή, έναν "έξυπνο" εκτυπωτή, έναν file server ή οποιοδήποτε άλλο μηχάνημα "τρέχει" κάποια υλοποίηση πρωτοκόλλων δικτύωσης υπολογιστών (π.χ. TCP/IP). Κάθε ένας από τους παραπάνω κόμβους έχει κάποια φυσική διεύθυνση για τις οντότητες πρωτοκόλλων (π.χ. κάρτα Ethernet που υλοποιεί το φυσικό επίπεδο και το επίπεδο προσπέλασης του μέσου) που τον συνδέουν με το φυσικό μέσο, και μπορεί να επικοινωνεί με άλλους κόμβους χρησιμοποιώντας τη διεύθυνση αυτή.

Οι φυσικές διευθύνσεις έχουν διαφορετική μορφή για κάθε δίκτυο και δίνονται σε κάποιο μηχάνημα με διαφορετικούς τρόπους. Για παράδειγμα, στα τοπικά δίκτυα Ethernet είναι αριθμητικές τιμές με μήκος 6 bytes, π.χ. 08-00-14-57-69-69 και δίνονται από τον κατασκευαστή της Ethernet κάρτας για τη σύνδεση ενός μηχανήματος με το δίκτυο (βλ. και παρακάτω σχέδιο). Αυτό σημαίνει ότι αν κάποιος κόμβος έχει περισσότερες από μία συνδέσεις σε Ethernet segments, θα έχει και περισσότερες από μία φυσικές διευθύνσεις. Η IEEE (Institute for Electrical and Electronic Engineers) διαχειρίζεται όλο το φάσμα των Ethernet φυσικών διευθύνσεων, αποδίδοντάς τες όπου αυτές είναι απαραίτητες. Παρακάτω δίνουμε τη συνήθη μορφή μιας Ethernet διεύθυνσης.

0x08 0x00 0x14 0x57 0x69 0x69	
^ ^ ^ ^ ^ ^	^ ^ ^ ^ ^ ^
manufacturer	board

Για ένα άλλο παράδειγμα φυσικής διεύθυνσης μπορεί να δει κανείς τα δίκτυα με X.25 συνδέσμους, όπου χρησιμοποιείται το πρότυπο X.121, όσο αναφορά τις φυσικές διευθύνσεις, και οι οποίες είναι αριθμοί 14 ψηφίων.

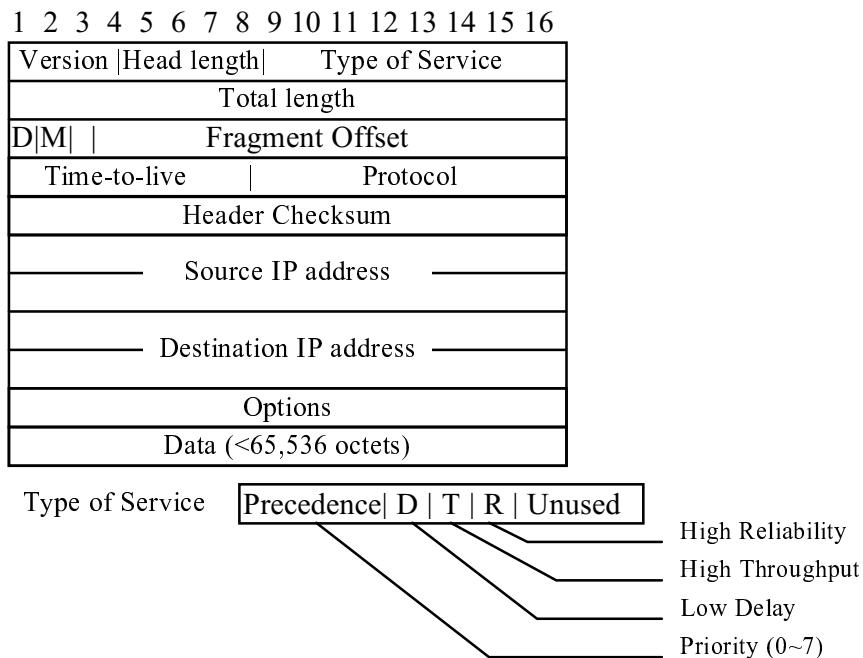
## 2.2. Το πρωτόκολλο IP

Στα TCP/IP δίκτυα όλη η πληροφορία μεταφέρεται από το μη αξιόπιστο πρωτόκολλο δικτύωσης χωρίς σύνδεση (connectionless), **IP (Internet Protocol)**. Η μονάδα δεδομένων του IP πρωτοκόλλου ονομάζεται IP-datatype. Για το πρωτόκολλο αυτό λέμε ότι είναι μη αξιόπιστο γιατί δεν εγγυάται την μεταφορά της πληροφορίας που αναλαμβάνει. Κάποιο IP-datatype μπορεί να φθάσει σε λάθος προορισμό, να διπλασιαστεί μέσα στο δίκτυο (μετά από καθυστέρηση και επαναμετάδοση) ή να χαθεί στο δρόμο προς το προορισμό του (μετά από υπερχείλιση κάποιας ουράς σε ενδιάμεσο κόμβο του δικτύου). Επίσης, λέμε ότι είναι χωρίς σύνδεση γιατί κάθε IP-datatype

μεταδίδεται ανεξάρτητα από όλα τα άλλα datagrams. Το ίδιο μεταφέρει όλες τις πληροφορίες που απαιτούνται (π.χ. διεύθυνση πηγής, διεύθυνση προορισμού, κ.ά.), σε αντίθεση για παράδειγμα με μια τηλεφωνική σύνδεση, όπου όλα τα δεδομένα (φωνή) ακολουθούν τον ίδιο δρόμο χωρίς να μεταφέρουν επιπρόσθετες πληροφορίες. Οι TCP/IP εφαρμογές (π.χ. μεταφορά αρχείου) μπορούν στην συνέχεια με κατάλληλους μηχανισμούς, είτε στο επίπεδο μεταφοράς, είτε στο επίπεδο εφαρμογής να προσφέρουν την ποιότητα της αξιοπιστίας που απαιτείται σε κάθε περίπτωση.

Το πρωτόκολλο IP προσφέρει ένα σύνολο από βασικές λειτουργίες οι οποίες είναι απαραίτητες για την επιτυχία της διασύνδεσης δικτύων υπολογιστών. Τέτοιες λειτουργίες είναι:

- Λειτουργίες **κατακερματισμού** των μηνυμάτων και **επανασύνδεσης** αυτών προκειμένου αυτά να περάσουν από υπο-δίκτυα που υποστηρίζουν διαφορετικού μεγέθους πεδίο δεδομένων στο πλαίσιο τους.
- Λειτουργίες **δρομολόγησης** των IP-datagrams διαμέσου των κόμβων του δικτύου, προκειμένου να φθάσουν στον προορισμό τους. Για να επιτευχθεί αυτό κάθε κόμβος πρέπει να γνωρίζει την IP-διεύθυνση του υπεύθυνου δρομολογητή προκειμένου να του αναθέτει τη δρομολόγηση ενός IP-telegram και κάθε δρομολογητής πρέπει να γνωρίζει τη διαδρομή που πρέπει να ακολουθηθεί από κάποιο datagram προκειμένου αυτό να φθάσει στον κόμβο προορισμό.
- Λειτουργίες **αναφοράς σφαλμάτων**. Κατά την εκτέλεση των δύο παραπάνω λειτουργιών είναι δυνατό να χαθούν datagrams. Αυτή η τρίτη λειτουργία σκοπό έχει την ενημέρωση του κόμβου-πηγή για τη σχετική απώλεια.

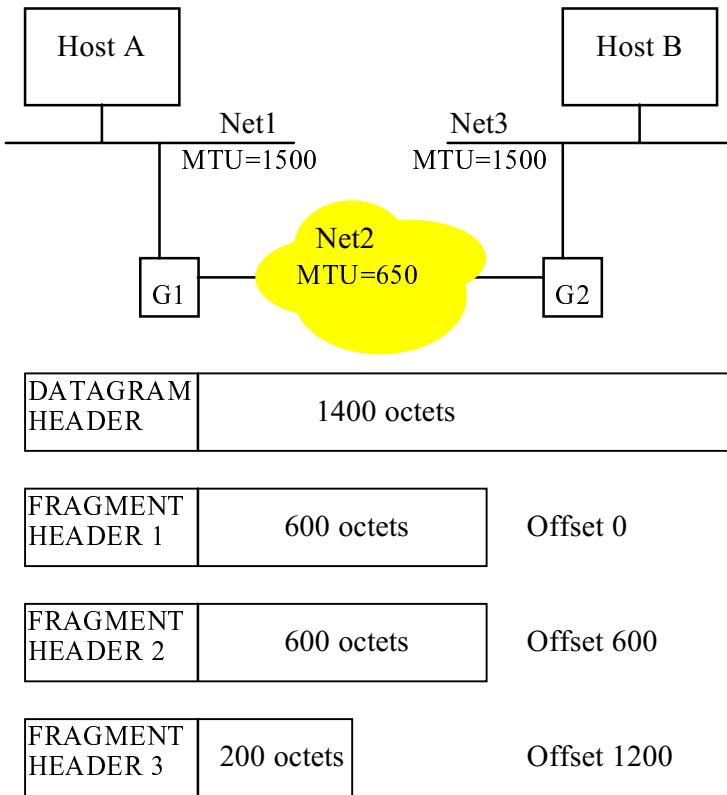


**Σχήμα 2.6. Ένα IP datagram**

Το IP πρωτόκολλο ορίζει την μορφή που πρέπει να πάρουν τα πακέτα και την επεξεργασία που πρέπει να δεχθούν προκειμένου να μεταδοθούν μέσα από το μέσο μεταφοράς. Η τελική μορφή ονομάζεται IP datagram. Ανάλογα και με άλλες μονάδες δεδομένων πρωτοκόλλων (π.χ. ένα Ethernet πλαίσιο), σε κάθε datagram ξεχωρίζουμε διάφορα πεδία, μεταξύ των οποίων τις IP διευθύνσεις πηγής και προορισμού, τον χρόνο ζωής του, το μήκος του και φυσικά τα δεδομένα, που μεταφέρει (βλ. και Σχήμα 2.6). Ένα IP datagram μεταφέρεται στο πεδίο δεδομένων του πλαισίου του επιπέδου σύνδεσης δεδομένων. Βέβαια το υλικό και λογισμικό υλοποίησης του επιπέδου σύνδεσης δεδομένων δεν αναγνωρίζουν, ούτε την μορφή ενός IP datagram, ούτε την IP διεύθυνση του προορισμού. Π.χ. ένα Ethernet πλαίσιο απλά μεταφέρει σαν κάποιο δεδομένο το IP datagram (η τεχνική αυτή είναι γνωστή σαν encapsulation, και είναι σύμφωνη με την OSI αρχιτεκτονική δικτύων υπολογιστών).

### 2.2.1. Κατακερματισμός και επανασύνδεση (Fragmentation and reassembly)

Σε αντίθεση με το μήκος του MAC πλαισίου που καθορίζεται από τα τεχνικά χαρακτηριστικά του μέσου μεταφοράς, το μήκος ενός IP datagram καθορίζεται από το λογισμικό υλοποίησης του IP πρωτοκόλλου, σύμφωνα δε με την επικεφαλίδα ενός IP datagram μπορεί να έχει μήκος μέχρι 65,535 octets (16 bit total length). Συνήθως όμως το λογισμικό αυτό καθορίζει το μήκος του IP datagram, έτσι ώστε να μπορεί να μεταφερθεί από ένα μοναδικό πλαίσιο, για λόγους κυρίως ικανότητας της επικοινωνίας. Αναφερόμαστε στο μέγεθος των δεδομένων που μπορεί να μεταφέρει κάποιο πλαίσιο με το όνομα **Maximum Transfer Unit (MTU)**. Σε περίπτωση που το IP datagram είναι μεγαλύτερο ή μικρότερο από το MTU, τότε ένα ή περισσότερα πλαίσια θα πρέπει να μεταφερθούν μη έχοντας γεμάτο το πεδίο δεδομένων. Επειδή επιπλέον η επικεφαλίδα του πλαισίου έχει σταθερό μήκος, εύκολα καταλαβαίνουμε ότι θα οδηγηθούμε σε χαμηλότερη χρησιμοποίηση του δικτύου.



### Σχήμα 2.7 - Λειτουργία Fragmentation / Reassembly

Το MTU δεν είναι το ίδιο για όλα τα είδη των μέσων μετάδοσης. Μπορεί να μεταβάλλεται από 128 bytes μέχρι 8,000 bytes σε σύγχρονα τοπικά δίκτυα. Αν λοιπόν κάποιο datagram στην πορεία του χρειαστεί να περάσει από κάποιο δίκτυο με MAC πλαισίο μικρότερου μήκους από αυτό του δικτύου που ξεκίνησε (βλ. και σχήμα 2.7), τότε ο υπεύθυνος δρομολογητής οφείλει να κατακερματίσει (fragmentation) το datagram σε μικρότερα τμήματα (fragments). Μετά την κατάτμηση ενός datagram και εφόσον τα τμήματα φθάσουν σε κάποιο δίκτυο με μεγαλύτερο MTU προσφέρονται δύο δυνατότητες. Είτε η επανασύνδεση (reassembly) των fragments, προκειμένου να γίνει η καλύτερη εκμετάλευση του επόμενου δικτύου, είτε η παραμονή του datagram στην ίδια κατάσταση μέχρι το κόμβο προορισμού. Σημειώνουμε ότι στην πράξη χρησιμοποιείται ο δεύτερος τρόπος, δηλ. από την στιγμή που κάποιο datagram κατακερματίστει δεν επανασυνδέεται παρά μόνο όταν φθάσει στον τελικό προορισμό του. Τα μειονεκτήματα της τεχνικής αυτής είναι τα εξής:

- Η χαμηλή χρησιμοποίηση των επόμενων στη σειρά δικτύων με μεγάλο MTU - για τους λόγους που εξηγήσαμε παραπάνω - και
- Η μεγαλύτερη πιθανότητα απώλειας κάποιου fragment και κατά συνέπεια απόρριψης του datagram, δεδομένου ότι το αρχικό datagram έχει διαιρεθεί σε περισσότερα τμήματα ενώ είναι αρκετή η απώλεια ενός από αυτά προκειμένου να χαθεί ολόκληρο το datagram.

Τα πλεονεκτήματα από την άλλη πλευρά είναι:

- Η τεχνική αυτή στην πράξη δουλεύει καλά, και

- Η τεχνική αυτή δεν επιφορτώνει ενδιάμεσους δρομολογητές με λειτουργίες αποθήκευσης και επανασύνδεσης των datagrams.

### 2.2.2. IP-διευθύνσεις

Μια διεύθυνση στο επίπεδο δικτύου IP (Internet Protocol) των TCP/IP πρωτοκόλλων, είναι αντίθετα με τις φυσικές διευθύνσεις που εξετάσαμε παραπάνω μια λογική διεύθυνση, η οποία είναι ανεξάρτητη από το μηχάνημα στο οποίο τρέχει η υλοποίηση των πρωτοκόλλων και εξαρτάται από το συγκεκριμένο δικτύο. Είναι μια αριθμητική τιμή μήκους 4 bytes (32-bit), η οποία καθορίζει τόσο το δίκτυο, όσο και το συγκεκριμένο κόμβο (υπολογιστή ή άλλο μηχάνημα) στο δίκτυο αυτό. Η 4-byte IP διεύθυνση συνήθως γράφεται (σε τεχνικά κείμενα) με δεκαδική μορφή. Π.χ. η 32-bit IP-διεύθυνση

10010011      01100110      0001101      00000001

γράφεται σαν

147.10.13.1

και είναι γνωστή σαν dotted decimal notation.

Οι κόμβοι που χρησιμοποιούν TCP/IP πρωτόκολλα μεταφράζουν την IP διεύθυνση προορισμού κάποιου πακέτου στην κατάλληλη φυσική διεύθυνση για τον κόμβο προορισμού. Κάθε εφαρμογή που επικοινωνεί με κάποια άλλη περιλαμβάνει τελικά στο πλαίσιο που θα σταλεί μέσα από το μέσο μεταφοράς και την IP διεύθυνση του μηχανήματος στο οποίο τρέχει, ώστε να είναι δυνατόν να της στείλει απάντηση η εφαρμογή-προορισμός. Επειδή η IP διεύθυνση είναι ανεξάρτητη του τύπου του δικτύου, μηχανήματα που βρίσκονται σε δύο διαφορετικά δίκτυα, που χρησιμοποιούν όμως την ίδια οικογένεια πρωτοκόλλων (π.χ. TCP/IP) μπορούν να επικοινωνήσουν. Σε κάθε ένα δίκτυο το TCP/IP λογισμικό θα αναλάβει να μεταφράσει τις IP-διευθύνσεις στις αντίστοιχες (διαφορετικές) φυσικές διευθύνσεις.

Από τα παραπάνω γίνεται φανερό ότι κάθε κόμβος σε ένα TCP/IP δίκτυο πρέπει να έχει μια IP-διεύθυνση προκειμένου να μπορεί να επικοινωνεί με τους υπόλοιπους κόμβους. Αντίθετα με τις φυσικές διευθύνσεις, οι οποίες είναι καλά καθορισμένες από τον κατασκευαστή της κάρτας, οι IP-διευθύνσεις μπορούν να καθοριστούν με ένα από τους παρακάτω τρόπους:

- Σε περίπτωση που οι χρήστες επιθυμούν την συμμετοχή του δικτύου τους στην κοινότητα INTERNET θα πρέπει να κάνουν αίτηση για επίσημη IP-διεύθυνση στο Internet Network Information Center (InterNIC) ή τον τοπικό πληρεξούσιο (Local Internet Registries). Οι μηχανισμοί αυτοί δίνουν μονάχα την IP-διεύθυνση του δικτύου σε κάποιο οργανισμό, ο οποίος έχει στην συνέχεια την ευθύνη διευθέτησης των IP-διευθύνσεων κάθε κόμβου.
- Σε αντίθεση περίπτωση, οι χρήστες καθορίζουν μόνοι τους τις IP διευθύνσεις των κόμβων του δικτύου τους, ακολουθώντας όμως τα παρακάτω κριτήρια:
  - Το μέρος της διεύθυνσης που αφορά το δίκτυο (βλ. και παρακάτω) να είναι το ίδιο σ' όλο το δίκτυο. Δηλ. αν αποφασιστεί ότι το δίκτυο θα έχει διεύθυνση 147.102 όλες οι διευθύνσεις των κόμβων θα πρέπει να ξεκινούν από 147.102.

- Η IP διεύθυνση για κάθε διαφορετικό κόμβο του δικτύου θα πρέπει να είναι μοναδική σε ολόκληρο το δικτύο.

Σίγουρα η τελευταία τακτική δεν είναι η καλύτερη, αφού απαγορεύει μελλοντική ένωση του δικτύου με το Internet (ειδικά σε περίπτωση που έχουν χρησιμοποιηθεί δεσμευμένες IP διευθύνσεις) και δυσκολεύει την ανταλλαγή λογισμικού.

### 2.2.3. Η διευθέτηση των υποδικτύων

#### A. Τάξεις IP-διευθύνσεων

Η διεύθυνση IP αποτελεί ένα μοναδικό αναγνωριστικό σ' ολόκληρο το Internet. Τρεις διαφορετικοί τύποι IP διευθύνσεων χρησιμοποιούνται, προκειμένου να εξυπηρετήσουν διαφορετικού μεγέθους δίκτυα. Κάθε τύπος είναι γνωστός σαν τάξη διεύθυνσης (address class). Ένα internet μπορεί να περιλαμβάνει διευθύνσεις όλων των τάξεων. Οι τρεις κύριες τάξεις IP διευθύνσεων είναι οι A, B, C, κάθε μία από τις οποίες προορίζεται να χρησιμοποιείται σε διαφορετικού μεγέθους δίκτυο (πάντοτε σε ότι αφορά τον αριθμό των κόμβων του δικτύου). Η κλάση στην οποία ανήκει κάποιο δίκτυο μπορεί να αναγνωριστεί από τη θέση του πρώτου μηδενικού στα τέσσερα πρώτα bits της IP-διεύθυνσης. Τα bits που υπολειπονται καθορίζουν δύο υπο-πεδία: ένα **αναγνωριστικό δικτύου (netid)** και ένα **αναγνωριστικό κόμβου (hostid)**, (βλ. και παρακάτω).

Αναλυτικότερα:

- Για την τάξη A, το netid έχει μήκος 1-byte, ενώ το hostid έχει μήκος 3-bytes. Στο netid το πιο σημαντικό bit είναι πάντα 0. Έτσι σε κάθε δίκτυο internet μπορούμε να έχουμε μέχρι 126 υποδίκτυα τάξης A, με περισσότερους από 16 εκατομμύρια κόμβους σε κάθε ένα από αυτά (σημειώνουμε ότι οι αριθμοί 0 και 127 είναι δεσμευμένοι). Συμπληρώνουμε, προς αποφυγή παρεξήγησης, ότι προκειμένου να χαρακτηριστεί κάποιο υποδίκτυο σαν τάξης A και να δεσμεύσει την αντίστοιχη IP διεύθυνση, πρέπει προφανώς να υπάρχει ανάγκη για κάτι τέτοιο (τεράστιος αριθμός κόμβων). Παράδειγμα δικτύου τάξης A αποτελεί το ARPANET.
- Για την τάξη B, το netid έχει μήκος 2-bytes, ενώ το hostid έχει επίσης μήκος 2-bytes. Στο netid τα δύο πιο σημαντικά bits είναι πάντα 10. Έτσι σε κάθε δίκτυο μπορούμε να έχουμε περίπου 16 χιλιάδες υποδίκτυα τάξης B, με περισσότερους από 65 χιλιάδες κόμβους σε κάθε ένα από αυτά. Το υποδίκτυο του E.M.II. είναι τάξης B με netid 147.102 (βλέπε ενδεικτικό σχήμα στο Παράρτημα B).
- Τέλος, για την τάξη C, το netid έχει μήκος 3-bytes, ενώ το hostid έχει μήκος 1-byte. Στο netid τα τρία πιο σημαντικά bits είναι πάντα 110. Έτσι σε κάθε δίκτυο internet μπορούμε να έχουμε περίπου 2 εκατομμύρια υποδίκτυα τάξης B, με 254 κόμβους σε κάθε ένα από αυτά.

1	8	16	24	32	
0	netid		hostid		= Class A (0~127)

1 0	netid		Hostid		= Class B (128~191)
-----	-------	--	--------	--	---------------------

1 1 0	netid		Hostid		= Class C (192~223)
-------	-------	--	--------	--	---------------------

1 1 1 0	multicast
1 1 1 1 1	reserved

Όσο αφορά τις δεσμευμένες IP διευθύνσεις συμπληρώνουμε τα εξής:

- **Λιεύθυνση δικτύου:** Δεν υπάρχει κόμβος σε ένα δίκτυο, που να έχει hostid όλο μηδενικά. Η διεύθυνση αυτή είναι η διεύθυνση του δικτύου (π.χ. 129.47.0.0 είναι η διεύθυνση ενός δικτύου τάξης B).
- **Λιεύθυνση προς όλους τους κόμβους (Broadcast Address):** Είναι η IP διεύθυνση που έχει hostid όλο 1. Βέβαια κανείς κόμβος δεν μπορεί να έχει μια τέτοιου είδους IP διεύθυνση.
- **Loopback Addresses:** Οι IP διευθύνσεις 127.0.0.0 και 127.0.0.1 είναι επίσης δεσμευμένες. Η διεύθυνση 127.0.0.0 χρησιμοποιείται για δοκιμές και επικοινωνία μεταξύ διεργασιών στο ίδιο μηχάνημα. Έτσι ενώ οι διεργασίες τρέχουν δεν εμφανίζεται καθόλου φορτίο στο δίκτυο.
- Τέλος, οι διευθύνσεις που το κομμάτι που αφορά το δίκτυο είναι όλο 1 ή όλο 0 είναι επίσης δεσμευμένες. Σαν κανόνας απομνημόνευσης 1 σημαίνει "όλοι" (δίκτυα ή κόμβοι), ενώ 0 σημαίνει "αυτό" (δίκτυο ή κόμβος).

Δεδομένου ότι μια IP-διεύθυνση κωδικοποιεί τόσο το δίκτυο, όσο και τον συγκεκριμένο κόμβο ουσιαστικά δεν αποτελεί αναγνωριστικό μιας μοναδικής μηχανής, αλλά μιας μοναδικής σύνδεσης σε ένα δίκτυο. Δηλ. ένας δρομολογητής ο οποίος συνδέει ν διαφορετικά δίκτυα, έχει ν διαφορετικές IP-διευθύνσεις (multi-homed). Επίσης αυτό σημαίνει ότι αν κάποιος χρήστης βγάλει το φορητό υπολογιστή του από κάποιο internet και τον μεταφέρει σε κάποιο άλλο internet θα πρέπει να αλλάξει την IP-διεύθυνση του μηχανήματός του.

Άλλο μειονέκτημα των IP-διευθύνσεων είναι ότι δεν προσφέρουν τη δυνατότητα μιας σταδιακής ανάπτυξης ενός δικτύου. Αυτό σημαίνει ότι αν κάποιος διαχειριστής διαχειρίζεται κάποιο δίκτυο τάξης C και το δίκτυο αυτό θα ξεπεράσει σύντομα τις 255 μηχανές, τότε ο διαχειριστής θα πρέπει να ζητήσει IP-διεύθυνση τάξης B, να διακόψει τη λειτουργία του δικτύου του, να αλλάξει τις IP-διευθύνσεις σε όλους τους κόμβους του δικτύου και να θέσει και πάλι το δίκτυο σε λειτουργία.

## B. Χωρισμός σε υποδίκτυα

Ένα TCP/IP δίκτυο μπορεί να διαιρεθεί σε ένα ή περισσότερα υποδίκτυα. Λόγοι που μπορούν να οδηγήσουν τον διαχειριστή ενός δικτύου σε μια τέτοια απόφαση είναι οι εξής:

- Η χρήση περισσοτέρων του ενός μέσων μεταφοράς. Όταν είναι αδύνατο, μηβολικό ή δαπανηρό, το να συνδεθούν όλοι οι απαραίτητοι κόμβοι σε ένα μοναδικό καλώδιο, πρέπει ο διαχειριστής να αποφασίσει διαιρέση του αρχικού δικτύου.

- Η μείωση του φορτίου (της συμφόρησης). Αν μετά από παρατήρηση αποφασιστεί ότι κάποιοι κόμβοι επικοινωνούν περισσότερο με κάποιους άλλους (πιθανώς σε συγκεκριμένη γεωγραφική περιοχή) (βλ. και διαχείριση επιδόσεων στο Κεφάλαιο 4) είναι σωστό να διαχωριστούν οι κόμβοι αυτοί από τους υπόλοιπους του δικτύου. Ο τρόπος είναι σχηματίζοντας υποδίκτυο με τους κόμβους που έχουν συχνότερη επικοινωνία μεταξύ τους.
- Η μείωση της χρονιμοποιήσης των επεξεργαστών στους κόμβων. Ανάλογα με τον παραπάνω λόγο, χωρίζοντας τους κόμβους σε υποδίκτυα μειώνεται η επεξεργασία σε ορισμένους από αυτούς (π.χ. λόγω broadcast μηνυμάτων).
- Η απομόνωση μεταξύ δικτύων. Με την απομόνωση επιτυγχάνεται η λειτουργία μέρους του δικτύου, σε περίπτωση προβλήματος που περιορίζεται σε κάποιο από τα υποδίκτυα.
- Τέλος, για λόγους ασφάλειας. Για παράδειγμα σε ένα μέσο μεταφοράς, όπως το Ethernet, κάθε κόμβος έχει πρόσβαση σε όλα τα πακέτα που μεταδίδονται. Έτσι ο μόνος τρόπος να προστατέψει ένας διαχειριστής απόρρητο φορτίο, είναι να το περιορίσει σε ξεχωριστό υποδίκτυο.

Οι τρόποι με τους οποίους μπορεί να διαιρέσει κανείς ένα TCP/IP δίκτυο σε υποδίκτυα είναι οι επόμενοι:

1. Σε περίπτωση που ο διαχειριστής δίνει από μόνος του τις IP διευθύνσεις τα πράγματα είναι απλούστατα, αφού μπορεί να δώσει νέες IP διευθύνσεις για το διαφορετικό υποδίκτυο.
2. Σε περίπτωση που το δίκτυο είναι συνδεδεμένο στο Internet (ή σε άλλο ανάλογο δίκτυο), ο διαχειριστής μπορεί να ζητήσει νέα νούμερα από την υπεύθυνη αρχή.
3. Τέλος, αν πρόκειται για κάποιο δίκτυο τάξης A ή B είναι ευκολότερο, αντί να ζητηθούν νέες διευθύνσεις δικτύων, ο διαχειριστής να χωρίσει το ήδη υφιστάμενο δίκτυο σε υποδίκτυα με κατάλληλο χειρισμό της IP διεύθυνσης του δικτύου.

Κάθε υποδίκτυο λειτουργεί ακριβώς σαν να ήταν ένα internet δίκτυο με τη διαφορά ότι τα υπόλοιπα δίκτυα βλέπουν τα υποδίκτυα στα οποία είναι διαιρεμένο κάποιο δίκτυο σαν ένα. Αυτό σημαίνει ότι αρκεί ένα netid για όλα τα υποδίκτυα. Μέσα τώρα στο δίκτυο αυτό, η επικοινωνία μεταξύ κόμβων που ανήκουν σε διαφορετικά υποδίκτυα γίνεται, όπως θα γινόταν αν βρίσκονταν σε διαφορετικά δίκτυα. Το πρωτόκολλο IP αναγνωρίζει την διεύθυνση του κόμβου προορισμού και προωθεί το datagram στον κατάλληλο δρομολογητή.

Αυτό που αξίζει να σημειώσουμε είναι ότι οι IP διευθύνσεις ακολουθούν τώρα ένα διαφορετικό διαχωρισμό. Δηλ. έχουμε:

<IP address> = <netid><subnetid><hostid>

Δηλ. όταν ένα δίκτυο διαιρείται σε υποδίκτυα το hostid διαιρείται σε δύο μέρη το subnetid και το hostid. Το hostid που μέχρι τώρα γνωρίσαμε καθορίζει τόσο το υποδίκτυο, όσο και τον κόμβο στο υποδίκτυο. Για παράδειγμα, ας θεωρήσουμε ένα δίκτυο τάξης B με netid 147.102. Οι διαχειριστές του δικτύου αυτού μπορούν να το διαιρέσουν σε υποδίκτυα με πολλούς τρόπους. Π.χ. αν το subnetid καθοριζόταν από 4 bits, τότε θα είχαμε 16 υποδίκτυα με 4094 κόμβους για το καθένα. Αν το subnetid καθοριζόταν από 8 bits, τότε θα είχαμε 256 υποδίκτυα με 254 κόμβους για το καθένα.

Μια μάσκα υποδικτύου (subnet mask) υποδεικνύει πως το αρχικό hostid διαιρείται σε subnetid και hostid. Γενικότερα μια network mask είναι ένας 32-bit αριθμός με 1 για τα bit του netid και του subnetid και 0 για τα bit του hostid.

## 2.2.4. Το πρωτόκολλο ICMP

Ένα άλλο πρωτόκολλο της οικογένειας TCP/IP είναι το Internet Control Message Protocol (ICMP). Τα ICMP μηνύματα μεταφέρουν πληροφορία σχετικά με διάφορες δυσλειτουργίες καθώς και λειτουργίες ελέγχου του δικτύου. Τέτοιες λειτουργίες είναι:

- αναφορά σφαλμάτων,
- δοκιμή δυνατότητας πρόσβασης σε κόμβο,
- έλεγχος συμφόρησης,
- ειδοποίηση αλλαγής διαδρομής,
- μέτρηση επιδόσεων, και
- διευθυνσιοδότηση υποδικτύων.

Το λογισμικό υλοποίησης των TCP/IP πρωτοκόλλων επεξηγεί τα ICMP μηνύματα και εκτελεί τις απαραίτητες ενέργειες, ανάλογα με το περιεχόμενο του μηνύματος. Δηλ. τα μηνύματα ICMP δεν στέλνονται από κάποια εφαρμογή (με εξαίρεση την εφαρμογή Ping που χρησιμοποιεί το ICMP για τα αποτελέσματά της, βλ. και Σχήμα 2.2), αλλά χρησιμοποιούνται από το λογισμικό υλοποίησης των TCP/IP πρωτοκόλλων σε ειδικές περιπτώσεις (για την αποστολή μηνυμάτων ελέγχου). Έτσι, δρομολογητές ή σταθμοί εργασίας μπορούν να αναφέρουν διάφορα προβλήματα που αφορούν IP datagrams, στην αρχική πηγή τους, χρησιμοποιώντας το ICMP πρωτόκολλο. Όλα τα ICMP μηνύματα φαίνονται στον Πίνακα 2.1.

Κάθε ένα από τα παραπάνω μηνύματα με την βοήθεια κατάλληλων κωδικών μπορεί να αναφέρει περισσότερες από μία περιπτώσεις. Για παράδειγμα στο ICMP μήνυμα που πληροφορεί για την απόρριψη ενός datagram, λόγω λήξης του χρόνου ζωής του, (Time Exceeded for a Datagram) μπορεί να υπάρχει πρόσθετη ερμηνεία ότι έληξε ο χρόνος ζωής του ή ότι έληξε ο χρόνος επανασύνδεσης ενός κερματισμένου datagram.

Τέλος, επειδή κάποιο ICMP μήνυμα συχνά χρειάζεται να περάσει μέσα από διαφορετικά δίκτυα, χρησιμοποιεί για την μεταφορά του το πεδίο δεδομένων του IP πακέτου.

ICMP μηνύματα
Echo Reply
Destination Unreachable
Source Quench
Redirect (change a route)
Echo Request
Time Exceeded for a Datagram
Parameter Problem on a Datagram
Timestamp Request
Timestamp Reply
Information Request (obsolete)
Information Reply (obsolete)

Address Mask Request
Address Mask Reply

### Πίνακας 2.1 - Όλα τα δυνατά ICMP μηνύματα

#### 2.3. Δρομολόγηση

Στην συνέχεια θα ασχοληθούμε αναλυτικά με τη **δρομολόγηση (routing)** στα TCP/IP δίκτυα. Ο όρος δρομολόγηση αναφέρεται στην μεταφορά ενός IP datagram από ένα κόμβο σε έναν άλλο, του ίδιου ή διαφορετικού δικτύου. Ο όρος αναφέρεται επίσης στον δρόμο που θα ακολουθήσει το IP datagram προκειμένου να φθάσει στον προορισμό του, και ο οποίος, όπως θα δούμε, βασίζεται στην IP διεύθυνση του δικτύου προορισμού. Θα ασχοληθούμε με δύο περιπτώσεις δρομολόγησης. Την άμεση και την έμμεση.

**Άμεση δρομολόγηση (direct)** έχουμε όταν, κάποιος κόμβος στέλνει IP datagrams σε κόμβο του ίδιου υπο-δικτύου (π.χ. του ίδιου Ethernet segment). Ότε με κατάλληλα μηνύματα (ARP) μπορεί να πληροφορηθεί την φυσική διεύθυνση του άλλου κόμβου, να τοποθετήσει το datagram σε ένα MAC πλαίσιο με τη φυσική διεύθυνση αυτή, και να το μεταδώσει.

Στην **έμμεση δρομολόγηση (indirect)**, κάποιος κόμβος στέλνει IP datagrams σε κόμβο διαφορετικού δικτύου χρησιμοποιώντας κατάλληλους ενδιάμεσους κόμβους, οι οποίοι ονομάζονται **δρομολογητές (routers)**. Όταν κάποιος κόμβος αναγνωρίσει ότι ένα IP datagram κατευθύνεται σε κόμβο διαφορετικού δικτύου, τότε μέσα από το μικρό πίνακα δρομολόγησης, που διαθέτει, επιλέγει τον κατάλληλο δρομολογητή. Με ένα ARP μήνυμα μαθαίνει την φυσική διεύθυνση του δρομολογητή αυτού, και του στέλνει το IP datagram με ένα MAC πλαίσιο. Σε περίπτωση που ο δρομολογητής είναι συνδεδεμένος και στο δίκτυο προορισμού, τότε πληροφορείται με παρόμοιο τρόπο την φυσική διεύθυνση του κόμβου προορισμού και του στέλνει το IP datagram. Σε αντίθετη περίπτωση, βρίσκεται με τη σειρά του ένα δεύτερο δρομολογητή στην φυσική διεύθυνση του οποίου στέλνεται το datagram, ο οποίος με την σειρά του θα εκτελέσει τις ίδιες λειτουργίες. Οι δρομολογητές, παίρνουν αποφάσεις με βάση το δίκτυο προορισμού και όχι με βάση τον σταθμό προορισμού. Αυτό σημαίνει ότι εξετάζουν αν είναι συνδεδεμένοι με δίκτυο το οποίο έχει το ίδιο netid με το κόμβο προορισμού, διαφορετικά στέλνουν το datagram σε άλλο δρομολογητή, ο οποίος θα καθορίσει τη συνέχεια της διαδρομής. Με λίγα λόγια, οι δρομολογητές σε ένα TCP/IP δίκτυο αποτελούν ένα συνεργαζόμενο διασυνδεδεμένο σύνολο, όπου datagrams περνούν από δρομολογητή σε δρομολογητή, έως ότου φθάσουν σ' εκείνο το δρομολογητή όπου η άμεση δρομολόγηση θα είναι δυνατή.

Κάθε αλγόριθμος δρομολόγησης χρησιμοποιεί ένα **πίνακα δρομολόγησης (routing table)**, όπου αποθηκεύονται πληροφορίες για τις διαδρομές που πρέπει να ακολουθήσει κάποιο datagram προκειμένου να φθάσει στον κόμβο προορισμού του. Σε ένα πίνακα δρομολόγησης αρκούν εγγραφές του τύπου (N, R), όπου N η IP διεύθυνση των δικτύων προορισμού, και R η IP διεύθυνση του επόμενου δρομολογητή στη διαδρομή, προκειμένου να προσεγγιστούν τα δίκτυα αυτά. Επειδή τόσο οι σταθμοί εργασίας, όσο και οι δρομολογητές παίρνουν αποφάσεις δρομολόγησης, ο πίνακας δρομολόγησης είναι απαραίτητος και στα δύο είδη κόμβων. Αν σκεφτούμε τον αριθμό των κόμβων σε ένα δίκτυο, όπως το Internet, εύκολα καταλαβαίνουμε ότι υπάρχει ανάγκη καθορισμού της (αυστηρά) απαραίτητης πληροφορίας, που πρέπει να υπάρχει σε ένα πίνακα δρομολόγησης. Αυτό σημαίνει ότι δεν μπορεί κάθε κόμβος ή δρομολογητής να γνωρίζει τη διαδρομή για οποιοδήποτε άλλο κόμβο του δικτύου.

Ο τρόπος επιλογής της διαδρομής που θα ακολουθηθεί από ένα datagram έχει σημαντικές συνέπειες στις επιδόσεις του δικτύου. Εντοπίζουμε ορισμένα προβλήματα που μπορεί να προκύψουν.

- Σε περίπτωση που ορίζεται μια **στατική διαδρομή** για κάποιο δίκτυο προορισμού, όλο το φορτίο για το δίκτυο αυτό (δηλ. για όλους τους κόμβους του δικτύου αυτού) θα ακολουθήσει την ίδια διαδρομή. Έτσι, ακόμα και στην περίπτωση που υπάρχουν πολλαπλές εναλλακτικές λύσεις για την δρομολόγηση προς το δίκτυο αυτό, αυτές δεν χρησιμοποιούνται. Επίσης, εξαιτίας του τρόπου δρομολόγησης παραμέτροι, όπως καθυστέρηση, φορτίο, ταχύτητα για κάθε διαδρομή δεν λαμβάνονται υπ' όψη.
- Οι πληροφορίες που καθορίζουν τη δρομολόγηση αφορούν τους δρομολογητές και όχι τους τελικούς κόμβους (υπολογιστές). Μόνο ο τελευταίος δρομολογητής, ο οποίος θα εκτελέσει την άμεση δρομολόγηση και θα δοκιμάσει να επικοινωνήσει με τον κόμβο προορισμό, είναι αυτός που πρώτος θα διαπιστώσει αν ο κόμβος αυτός λειτουργεί ή όχι. Στο ενδιάμεσο διάστημα ένα σημαντικό φορτίο μπορεί να έχει πλημμυρίσει το δίκτυο, ακόμα και αν ο κόμβος προορισμού είναι εκτός λειτουργίας.
- Κάθε δρομολογητής εκτελεί την λειτουργία της δρομολόγησης **κατανεμημένα**, ανεξάρτητα από τους υπόλοιπους δρομολογητές. Μπορεί να προκύψουν αστάθειες στον αλγόριθμο δρομολόγησης, δρόμοι με βρόγχους (loops) και διαφορές στη δρομολόγηση μεταξύ δύο κόμβων προς τις δύο κατευθύνσεις. Εποι, υπάρχει η περίπτωση το φορτίο από κάποιο κόμβο A προς κάποιο κόμβο B να ακολουθεί διαφορετική διαδρομή από αυτή που ακολουθεί το φορτίο από τον κόμβο B προς τον κόμβο A, και η μία κατεύθυνση να είναι προσωρινά αδύνατη.

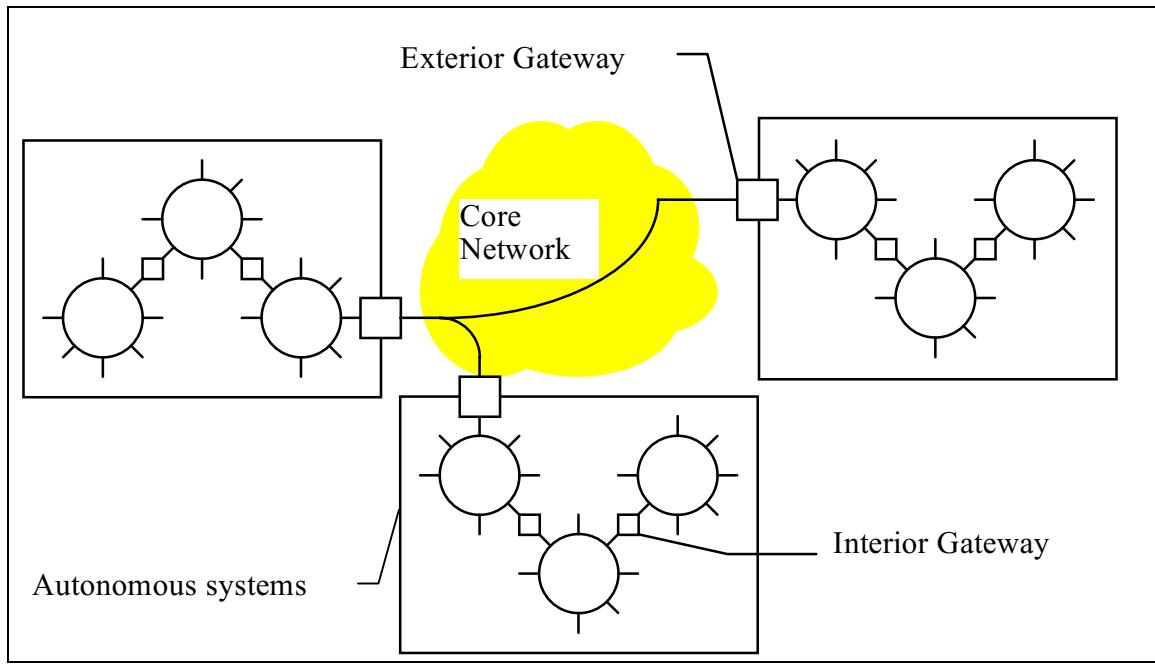
Οι συνήθεις αλγόριθμοι δρομολόγησης λαμβάνουν υπόψη και προκαθορισμένες διαδρομές (default), στην περίπτωση που δεν υπάρχει άλλη εναλλακτική διαδρομή. Η επιλογή αυτή είναι ιδιαίτερα χρήσιμη για διασυνδεδεμένα τοπικά δίκτυα με μία έξοδο προς το Internet. Επιπλέον, οι συνήθεις αλγόριθμοι δρομολόγησης μπορούν να λάβουν υπ' όψη τους και διαδρομές για κόμβους προορισμού (αντί για δίκτυα προορισμού), δίνοντας κατ' αυτό τον τρόπο στον διαχειριστή του δικτύου δυνατότητες για καλύτερο έλεγχο του δικτύου.

Τέλος σημειώνουμε ότι η πληροφορία στα IP datagrams, που αφορά τις διευθύνσεις προορισμού και πηγής δεν αλλάζει στη πορεία της δρομολόγησης. Αυτό που καθορίζεται στους πίνακες δρομολόγησης των ενδιαμέσων δρομολογητών είναι η διεύθυνση του επόμενου βήματος (next hop) προς τον τελικό προορισμό. Η φυσική όδευση προς το επόμενο βήμα σε ένα περιβάλλον LAN καθορίζεται σε χαμηλότερο επίπεδο (MAC) μέσω των πρωτοκόλλων ARP.

### 2.3.1. Αυτόνομα Συστήματα (Autonomous Systems)

Πριν προχωρήσουμε στην εξέταση των διαφόρων πρωτοκόλλων δρομολόγησης που χρησιμοποιούνται στο Internet, θα βοηθούσε να δίναμε μια σύντομη περιγραφή της γενικής αρχιτεκτονικής που ακολουθεί το Internet, καθώς και της ορολογίας που συνήθως χρησιμοποιείται.

To Internet αποτελείται από ένα σύνολο ανεξάρτητων διασυνδεδεμένων δικτύων (internets), και συνήθως το παρουσιάζουμε σαν ένα σύνολο **Αυτόνομων Συστημάτων (Autonomous Systems)**, κάθε ένα από τα οποία έχει την δική του διαχειριστική αρχή και χρησιμοποιεί εσωτερικά τα δικά του πρωτόκολλα δρομολόγησης. Όλα τα παραπάνω Αυτόνομα Συστήματα θεωρούνται προσαρτημένα στο δίκτυο κορμού του Internet (core backbone Network) (βλ. και παρακάτω Σχήμα 2.8 [HALS92]).



### Σχήμα 2.8 - Γενικές έννοιες στο Internet

Προκειμένου να διαχωρίσουμε τους δρομολογητές, οι οποίοι χρησιμοποιούνται μέσα σε κάποιο Αυτόνομο Σύστημα από αυτούς που χρησιμοποιούνται για την προσάρτηση του Αυτόνομου Συστήματος στο δίκτυο κορμού, χρησιμοποιούνται οι όροι *interior gateway* (εσωτερικός δρομολογητής-πύλη) και *exterior gateway* (εξωτερικός δρομολογητής-πύλη). Τα αντίστοιχα πρωτόκολλα δρομολόγησης, τα οποία χρησιμοποιούνται είναι τα: *Interior Gateway Protocol (IGP)* και *Exterior Gateway Protocol (EGP)*. Ενώ μπορεί να υπάρξουν πολλά IGP πρωτόκολλα δρομολόγησης (διαφορετικά για κάθε Αυτόνομο Σύστημα, ανάλογα με την επιλογή της κάθε διαχειριστικής αρχής), πρέπει να υπάρχει ένα μοναδικό EGP πρωτόκολλο δρομολόγησης, το οποίο θα ακολουθεί όλο το Internet.

Ένα αυτόνομο σύστημα μπορεί να αποτελείται από πολλά διασυνδεδεμένα δίκτυα υπολογιστών ή να είναι απλά ένα μοναδικό δίκτυο υπολογιστών, με ιδιαίτερη διαχειριστική αρχή. Στην πρώτη περίπτωση, αν τα διασυνδεδεμένα αυτά δίκτυα αποτελούνται από διάφορα υπο-δίκτυα (subnets), τότε εισάγουμε ένα ακόμα επίπεδο δρομολόγησης κάτω από τα exterior και interior gateways.

Παρακάτω θα εξετάσουμε πρωτόκολλα δρομολόγησης IGP και EGP. Αναπόσπαστο τμήμα των αλγορίθμων δρομολόγησης αποτελεί και το address resolution protocol (ARP), το οποίο χρησιμοποιείται μέσα σε ένα τοπικό δίκτυο για τη μετάφραση των IP διευθύνσεων σε φυσικές διευθύνσεις. Θα ξεκινήσουμε με τη περιγραφή του.

### 2.3.2. Address Resolution Protocol (ARP)

Η χρησιμοποίηση ενός δρομολογητή σε κάθε περίπτωση επικοινωνίας μεταξύ δύο σταθμών εργασίας φορτώνει υπερβολικά κάποιο υποδίκτυο. Σε περιπτώσεις όπου οι δύο σταθμοί εργασίας ανήκουν στο ίδιο υποδίκτυο (π.χ. στο ίδιο Ethernet segment), αρκεί ο σταθμός πηγή να γνωρίζει τη φυσική διεύθυνση του σταθμού προορισμού προκειμένου να του προωθήσει το IP datagram (άμεση δρομολόγηση). Το πρωτόκολλο το οποίο εκτελεί

την παραπάνω λειτουργία ονομάζεται ***Address Resolution Protocol (ARP)***. Το πρωτόκολλο ARP αποτελεί ουσιαστικό μέρος του IP σε κάθε κόμβο.

Η λειτουργία του πρωτοκόλλου αυτού έχει ως εξής:

- (1) Μόλις η λειτουργία κατάτμησης έχει ένα datagram έτοιμο για προώθηση, περνά ένα δείκτη στην μνήμη όπου βρίσκεται το datagram αυτό στο πρωτόκολλο ARP. Το τελευταίο διατηρεί ένα τοπικό πίνακα δρομολόγησης, ο οποίος περιέχει ζευγάρια IP διεύθυνσεων και φυσικών διευθύνσεων κόμβων του υποδικτύου.
- (2α) Εάν η διεύθυνση προορισμού βρίσκεται στον πίνακα αυτό, τότε το πρωτόκολλο ARP περνά το δείκτη στη μνήμη όπου βρίσκεται το datagram, μαζί με τη φυσική διεύθυνση στο υποκείμενο πρωτόκολλο για τη μετάδοση.
- (2β) Εάν η διεύθυνση προορισμού δεν βρίσκεται στον πίνακα αυτό, τότε το πρωτόκολλο ARP θα προσπαθήσει να την ανακαλύψει, στέλνοντας ένα ARP μήνυμα. Το μήνυμα αυτό θα περιέχει την IP και τη φυσική διεύθυνση της πηγής και την IP διεύθυνση του προορισμού, της οποίας την αντίστοιχη φυσική αναζητεί. Το μήνυμα αυτό μπορεί να σταλεί είτε broadcast, είτε σε συγκεκριμένο σταθμό εργασίας (π.χ. το δρομολογητή). Στην περίπτωση broadcast μετάδοσης ο σταθμός ο οποίος αναζητείται, θα αναγνωρίσει την IP διεύθυνσή του στο ARP μήνυμα και θα προχωρήσει στην επεξεργασία του μηνύματος. Στην περίπτωση χρήσης τρίτου σταθμού ο σταθμός αυτός θα εκτελέσει τις ίδιες λειτουργίες.
- (3) Κατ' αρχήν θα ενημερώσει το δικό του τοπικό πίνακα δρομολόγησης με το ζευγάρι της IP και της φυσικής διεύθυνσης του σταθμού που έστειλε το μήνυμα, εφόσον αυτές δεν υπάρχουν ήδη.
- (4) Στην συνέχεια στέλνει απάντηση στο σταθμό με ένα δεύτερο ARP μήνυμα, το οποίο περιέχει τη μέχρι τώρα άγνωστη φυσική διεύθυνση μαζί με όλα τα άλλα γνωστά στοιχεία.
- (5) Λαμβάνοντας την απάντηση ο πρώτος σταθμός (έχει επιτύχει το binding των δύο διευθύνσεων), αρχικά ενημερώνει τον τοπικό του πίνακα δρομολόγησης με το ζευγάρι διευθύνσεων που έλαβε, και συνεχίζει όπως στο (2α).

Σημειώνουμε ακόμα τα εξής:

- Κάθε σταθμός που λαμβάνει ένα πλαίσιο ενημερώνει τον τοπικό του πίνακα δρομολόγησης με το ζευγάρι IP και φυσικής διεύθυνσης του σταθμού πηγής, εφόσον η πληροφορία αυτή δεν υπάρχει. Αυτό γιατί η πληροφορία αυτή θα είναι πιθανότατα απαραίτητη σε μετέπειτα απάντηση των υψηλότερων πρωτοκόλλων στο συγκεκριμένο datagram.
- Τα ζευγάρια αυτά των διευθύνσεων διατηρούνται στη μνήμη του υπολογιστή για κάποιο συγκεκριμένο χρονικό διάστημα και όχι για πάντα, δεδομένου ότι οι αντιστοιχίες φυσικών - IP διεύθυνσεων μπορεί κάθε στιγμή να αλλάξουν σε κάποιο δίκτυο.

Συμπερασματικά το ARP είναι ένα χαμηλού επιπέδου πρωτόκολλο, το οποίο κρύβοντας την μορφή των φυσικών διεύθυνσεων, μας επιτρέπει να χρησιμοποιούμε τις IP διεύθυνσεις με όποιο τρόπο θέλουμε.

Η IP διεύθυνση κάποιου σταθμού διατηρείται σε κάποιο δίσκο, ώστε να είναι δυνατή η εύρεσή της κάθε φορά που το μηχάνημα ξεκινά. Στην περίπτωση σταθμών χωρίς δίσκο,

το παραπάνω δεν είναι δυνατό, οπότε και χρησιμοποιείται ένα κατάλληλο πρωτόκολλο για τη λύση του προβλήματος, το **Reverse Address Resolution Protocol (RARP)**.

Προκειμένου να λειτουργήσει το πρωτόκολλο RARP χρησιμοποιείται ένας server, ο οποίος είναι υπεύθυνος για μια σειρά από σταθμούς χωρίς δίσκο, διατηρώντας τα ζευγάρια IP - φυσικών διευθύνσεων, όλων των σταθμών αυτών στο δίσκο του. Μόλις κάποιος από τους σταθμούς αυτούς ξεκινήσει, στέλνει ένα broadcast RARP μήνυμα, το οποίο περιέχει μονάχα τη φυσική του διεύθυνση. Λάμβανοντας το μήνυμα αυτό ο κατάλληλος server απαντά με δεύτερο μήνυμα, το οποίο περιέχει την IP διεύθυνση του σταθμού χωρίς δίσκο, καθώς και το ζευγάρι IP - φυσικής διεύθυνσης του server.

### 2.3.3. Interior Gateway Protocol (IGP): RIP - OSPF. Παράδειγμα

Για τη δρομολόγηση σε ένα αυτόνομο σύστημα υπάρχουν ουσιαστικά δύο τρόποι. Η στατική και η δυναμική δρομολόγηση. Στην περίπτωση μικρών δικτύων, των οποίων η διάρθρωση δεν αλλάζει πολύ συχνά, οι διαχειριστές έχουν τη δυνατότητα να καθορίζουν και να μεταβάλλουν τις διαδρομές στους δρομολογητές με λειτουργίες διαχείρισης (στατικός τρόπος). Ο στατικός αυτός τρόπος δρομολόγησης δεν μπορεί να ανταποκριθεί στις γρήγορες μεταβολές ενός δικτύου, όπως π.χ. το Internet, όπου δρομολογητές και δίκτυα ανεβοκατεβαίνουν με ιδιαίτερα υψηλή συχνότητα. Σε μια τέτοια περίπτωση, ο καθορισμός των διαδρομών πρέπει να ανατεθεί σε αυτόματες διαδικασίες (πρωτόκολλα δρομολόγησης). Τα πρωτόκολλα δρομολόγησης μέσα σε ένα αυτόνομο σύστημα ονομάζονται *Interior Gateway Protocols (IGP)*.

Το πρωτόκολλο IGP μπορεί να είναι διαφορετικό από αυτόνομο σύστημα σε αυτόνομο σύστημα, και εξαρτάται από την επιλογή που θα κάνει η εκάστοτε διαχειριστική αρχή. Το IGP πρωτόκολλο, το οποίο χρησιμοποιείται συχνότερα είναι το **IP Routing Information Protocol (RIP)**, γνωστό και σαν **routed** από το πρόγραμμα (daemon) που υλοποιεί το πρωτόκολλο αυτό το λειτουργικό UNIX. Είναι ένα κατανεμημένο πρωτόκολλο δρομολόγησης, το οποίο βασίζεται σε μια τεχνική γνωστή σαν distance-vector algorithm. Η ιδέα της vector distance δρομολόγησης είναι πολύ κοντά στον αλγόριθμο εύρεσης ελάχιστου δρόμου Bellman-Ford (βλ. και παρακάτω). Το κύριο μειονέκτημα ενός τέτοιου αλγορίθμου είναι ότι εκτελεί ένα κατανεμημένο υπολογισμό του ελάχιστου μονοπατιού, ο οποίος μπορεί να μην συγκλίνει. Ένα άλλο μειονέκτημα είναι ότι τα μηνύματα που ανταλλάσσονται μεγαλώνουν, όσο αυξάνεται το μέγεθος των δικτύων.

Η κύρια εναλλακτική λύση είναι μια κλάση αλγορίθμων γνωστών σαν link-state Open Shortest Path First (OSPF). Οι αλγόριθμοι αυτοί, απαιτούν από κάθε δρομολογητή να έχει πλήρη εικόνα του δικτύου, δηλ. των υπολοίπων δρομολογητών και των δικτύων μεταξύ αυτών. Η εικόνα αυτή παρουσιάζει τους δρομολογητές σαν κόμβους ενός γράφου και τα δίκτυα σαν τους συνδέσμους του γράφου αυτού.

Σύμφωνα με τον αλγόριθμο αυτό, ένας δρομολογητής ελέγχει την κατάσταση των γειτόνων του και στην συνέχεια μεταδίδει την πληροφορία αυτή και σ' όλους τους άλλους δρομολογητές. Οποτεδήποτε κάτι αλλάζει στην κατάσταση του γράφου, χρησιμοποιείται ο αλγόριθμος ελαχίστου δρόμου του Dijkstra (βλ. και παρακάτω) για τον επαναυπολογισμό των ελαχίστων μονοπατιών. Με τον τρόπο αυτό πολλά από τα προβλήματα των vector distance αλγορίθμων ξεπερνιούνται [HALS92], [COME91].

Παρακάτω θα εξετάσουμε αναλυτικότερα το πρωτόκολλο RIP.

Ο όρος distance χρησιμοποιείται σαν ένα μέτρο αξιολόγησης της διαδρομής μεταξύ δύο δρομολογητών. Για παράδειγμα, το μέτρο δρομολόγησης μπορεί να είναι:

α) **Hops**, οπότε η απόσταση είναι ίση με τον αριθμό των δικτύων που υπάρχουν μεταξύ των δύο δρομολογητών,

β) **Mean transit delay**, οπότε η απόσταση είναι ίση με τη καθυστέρηση αυτή.

Οποιοδήποτε και αν είναι το μέτρο δρομολόγησης, το πρωτόκολλο RIP χρησιμοποιεί ένα κατανεμημένο αλγόριθμο, προκειμένου κάθε *interior* δρομολογητής σε ένα αυτόνομο σύστημα να σχηματίσει ένα πίνακα δρομολόγησης που θα περιέχει τις αποστάσεις μεταξύ αυτού και όλων των άλλων δικτύων στο σύστημα αυτό.

Αρχικά, κάθε δρομολογητής έχει στο πίνακα δρομολόγησης μονάχα τα *netids* των δικτύων στα οποία είναι προσαρτημένος. Επιπλέον γνωρίζει τις διευθύνσεις των υπολοίπων δρομολογητών που είναι προσαρτημένοι στα ίδια δίκτυα. Οι πληροφορίες αυτές μπορεί να έχουν εισαχθεί με λειτουργίες διαχείρισης της διάρθρωσης του δικτύου (Configuration Management). Περιοδικά (π.χ. κάθε 30 sec) οι δρομολογητές στέλνουν τα περιεχόμενα του πίνακα δρομολόγησης που διατηρούν, σε κάθε ένα από τους γείτονές τους. Παρόμοια και οι ίδιοι λαμβάνουν τους πίνακες δρομολόγησης των γειτόνων τους. Βάσει αυτών, θα προχωρήσουν στην επιβεβαίωση ή την αναθεώρηση του δικού τους πίνακα δρομολόγησης. Κάθε φορά που παρουσιάζεται κάποια διαδρομή προς ένα δίκτυο με μικρότερο κόστος από τη διαδρομή που ήδη χρησιμοποιείται, τότε η νέα διαδρομή θα αντικαταστήσει τη προηγουμένη. Το παράδειγμα που ακολουθεί διασαφηνίζει περισσότερο το πρωτόκολλο δρομολόγησης RIP.

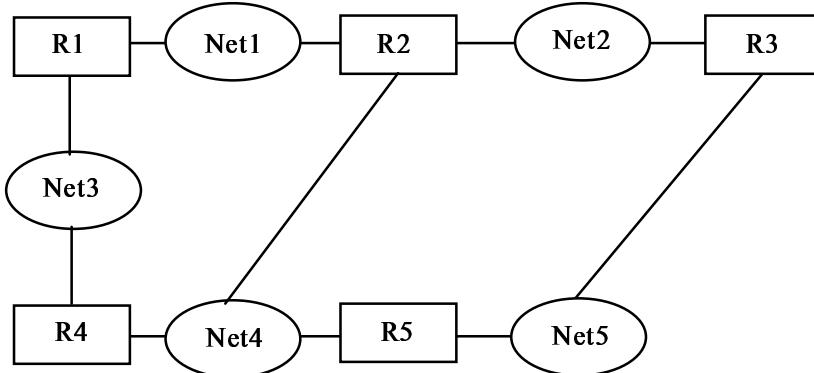
Προσθέτουμε, ότι κάθε εγγραφή στο πίνακα δρομολόγησης διατηρεί ένα *timer*. Αν η εγγραφή δεν ανανεωθεί ή επιβεβαιωθεί για κάποιο χρονικό διάστημα (π.χ. 180 sec), τότε η εγγραφή διαγράφεται. Τέλος άλλα προβλήματα τα οποία τίθονται σαν άσκηση είναι:

α) Η ταχύτητα σύγκλισης του αλγόριθμου σε κάποια σταθερή κατανομή.

β) Η δημιουργία βρόγχων κατά τη λειτουργία του πρωτοκόλλου.

### Παράδειγμα

Στο δίκτυο του ακόλουθου σχήματος εφαρμόζουμε τον αλγόριθμο δρομολόγησης RIP.



Αρχικά οι πίνακες δρομολόγησης στους routers έχουν ως εξής:

R1	R2	R3	R4	R5
Net	Net	Net	Net	Net
1   D,R 0,1	1   D,R 0,2	2   D,R 0,3	3   D,R 0,4	4   D,R 0,5

3		0,1	2		0,2	5		0,3	4		0,4	5		0,5
			4		0,2									

Όπου **D**: το κόστος και **R**: ο router που πρέπει να χρησιμοποιηθεί.

Μετά από κάποιο αριθμό ανταλλαγών των πινάκων δρομολόγησης μεταξύ των δρομολογητών, οι πίνακες δρομολόγησης των παραπάνω δρομολογητών θα έχουν τη παρακάτω μορφή.

R1		R2		R3		R4		R5	
Net	D,R								
1	0,1	1	0,2	2	0,3	3	0,4	4	0,5
3	0,1	2	0,2	5	0,3	4	0,4	5	0,5
2	1,2	4	0,2	1	1,2	1	1,1	2	1,3
4	1,4	3	1,1	4	1,2	2	1,2	3	1,4
5	2,2	4	1,4	3	2,2	5	1,5	1	2,3

### 2.3.4. Exterior Gateway Protocol (EGP)

Η παράγραφος αυτή περιγράφει το πρωτόκολλο, το οποίο χρησιμοποιείται προκειμένου δίκτυα τα οποία ανήκουν σε διαφορετικά αυτόνομα συστήματα, να μπορούν να επικοινωνήσουν μεταξύ τους.

Η διαχειριστική αρχή κάθε αυτόνομου συστήματος επιλέγει ένα ή περισσότερους δρομολογητές προκειμένου να λειτουργήσουν σαν Exterior Gateway, για το συγκεκριμένο αυτόνομο σύστημα. Οι παραπάνω δρομολογητές επικοινωνούν με τους υπόλοιπους δρομολογητές του αυτόνομου συστήματος, χρησιμοποιώντας το IGP πρωτόκολλο, το οποίο έχει επιλεχθεί, και εξετάσαμε παραπάνω. Με το τρόπο αυτό, οι Exterior Gateways, μαθαίνουν τα netids των δικτύων, που ανήκουν στο συγκεκριμένο αυτόνομο σύστημα, καθώς και τις σχετικές αποστάσεις.

Το πρωτόκολλο EGP έχει τρία κύρια χαρακτηριστικά [COME91]:

1. Υποστηρίζει ένα μηχανισμό neighbor acquisition, ο οποίος επιτρέπει σε κάποιο δρομολογητή να ζητήσει από κάποιο άλλο, τη δυνατότητα της επικοινωνίας μεταξύ τους για την ανταλλαγή πληροφορίας προσιτότητας. Τότε λέμε ότι ο δρομολογητής αποκτά ένα ομότιμο δρομολογητή EGP ή ένα γείτονα EGP. Οι ομότιμοι δρομολογητές EGP είναι γείτονες μονάχα από την άποψη ότι θα ανταλλάσσουν πληροφορία δρομολόγησης και όχι ανάλογα με τη γεωγραφική τους θέση.
2. Επιτρέπει στους δρομολογητές EGP να δοκιμάζουν, κατά πόσο οι γείτονές τους εξακολουθούν να αποκρίνονται.
3. Υποστηρίζει ένα μηχανισμό ανταλλαγής πληροφορίας προσιτότητας, μέσω της ανταλλαγής μηνυμάτων routing update.

Οι μηχανισμοί αυτοί υλοποιούνται με την ανταλλαγή κατάλληλων μηνυμάτων. Τα μηνύματα EGP και η σημασία τους δίνεται παρακάτω.

Function	EGP message	Meaning
Neighbour acquisition	Acquisition request	Requests a gateway to become a neighbour

	Acquisition confirm	Gateway agrees to become a neighbour
	Acquisition refuse	Gateway refuses
	Cease request	Requests termination of a neighbour relationship
	Cease confirm	Confirms break-up of relationship
Neighbour reachability	Hello	Requests neighbour to confirm a previously established relationship
	I-heard-you	Confirms relationship
Routing update	Poll request	Requests network reachability update
	Routing Update	Network reachability information
Error response	Error	Response to any incorrect request message

Ενδιαφέρον παρουσιάζει ο τρίτος μηχανισμός, ο οποίος αφορά την πληροφορία για τη δρομολόγηση. Ένας Exterior δρομολογητής στέλνει ένα μήνυμα routing update προκειμένου να μεταφέρει πληροφορία σ' ένα γείτονα EGP δρομολογητή, σχετικά με τα δίκτυα που είναι προσιτά σ' αυτόν. Ο τελευταίος έχει συλλέξει τη σχετική πληροφορία και την ανακοινώνει στο γείτονά του. Το μήνυμα routing update περιέχει μια λίστα από δίκτυα (netids) τα οποία είναι προσιτά από κάθε δρομολογητή του αυτόνομου συστήματος τοποθετημένα με σειρά ανάλογα με την απόσταση από τον δρομολογητή EGP που αποκρίνεται.

Ένας δρομολογητής που τρέχει το πρωτόκολλο EGP, έχει τη δυνατότητα να αναφέρει δύο είδη προσιτότητας: (1) Δίκτυα προορισμού τα οποία βρίσκονται στο δικό του αυτόνομο σύστημα, και (2) Δίκτυα προορισμού για τα οποία έχει μάθει, αλλά τα οποία βρίσκονται πέρα από το σύνορο του αυτόνομου συστήματός του. Τη δεύτερη δυνατότητα μπορεί να την εκμεταλευτούν μόνο οι **δρομολογητές κορμού (core gateways)**, οι οποίοι συντηρούν πληροφορίες προς προορισμούς σε μη γειτονικά αυτόνομα συστήματα (autonomous system).

Αν ένα Πανεπιστημιακό συγκρότημα (campus) αποτελεί ένα αυτόνομο σύστημα, τότε ο exterior δρομολογητής του μπορεί να συλλέγει πληροφορίες για τα δίκτυα του campus και να την ανακοινώνει στο core Internet. Δεν θα είναι δική του ευθύνη να ανακοινώνει διαδρομές προς δίκτυα που ανήκουν σε διαφορετικά campus. Δεν συμβαίνει το ίδιο και για τα core gateways του Internet.

Το EGP ελέγχεται από το πρωτόκολλο διαχείρισης SNMP, όπως θα δούμε παρακάτω.

### 2.3.5. Αλγόριθμοι εύρεσης ελαχίστων δρόμων: Bellman-Ford, Dijkstra, Floyd-Warshall.

Θα παρουσιάσουμε παρακάτω τρεις διαφορετικούς αλγόριθμους δρομολόγησης για την εύρεση του ελαχίστου μονοπατιού μεταξύ δύο κόμβων ενός δικτύου, και συγκεκριμένα τους αλγόριθμους Bellman-Ford, Dijkstra και Floyd-Warshall [BERT87]. Οι δύο πρώτοι αλγόριθμοι βρίσκουν το ελάχιστο μονοπάτι από ένα δεδομένο κόμβο πηγή προς όλους τους άλλους κόμβους (ή ισοδύναμα από όλους τους κόμβους προς ένα δεδομένο κόμβο προορισμό). Ο τρίτος αλγόριθμος βρίσκει τα ελάχιστα μονοπάτια από όλους τους κόμβους προς όλους τους άλλους κόμβους του δικτύου.

### α) Αλγόριθμος Bellman-Ford

Ο αλγόριθμος θεωρεί γνωστή την τοπολογία του δικτύου και ότι κάποιος κόμβος  $1$  είναι ο κόμβος πηγής. Έστω  $n$  μη αρνητικός ακέραιος αριθμός, τότε ορίζουμε ως  $D_i^{(n)}$  το μήκος της ελάχιστης διαδρομής από τον κόμβο  $1$  στον κόμβο  $i$  με την προϋπόθεση ότι το σχετικό μονοπάτι έχει το πολύ  $n$  το πλήθος συνδέσμους και  $d_{ij}$  το κόστος του μονοπατιού από τον κόμβο  $i$  στον κόμβο  $j$ .

Αρχικά έχουμε:  $D_1^{(h)} = 0 \quad \forall h$ ,  $D_i^{(0)} = \infty \quad \forall i \neq 1$

Για κάθε διαδοχικό  $h \geq 0$ :  $D_i^{(h+1)} = \min_j [D_j^{(h)} + d_{ij}] \quad \forall i \neq 1$

Αν:  $\forall i \neq 1 \quad D_i^{(n+1)} = D_i^{(n)}$  για δύο διαδοχικά  $h$  τότε σταματάμε τον αλγόριθμο, ή διαφορετικά έπειτα από  $N$  επαναλήψεις.

Στην χειρότερη περίπτωση ο αλγόριθμος πρέπει να επαναληφθεί  $N-1$  φορές για  $N-1$  κόμβους και με  $N-1$  εναλλακτικές λύσεις. Επομένως, πρόκειται για αλγόριθμο πολυπλοκότητας  $O(N^3)$ .

Παράδειγμα: INITIAL LABELS:  $L(1)=L(2)=\dots=L(5)=\infty$ ,  $L(6)=0$

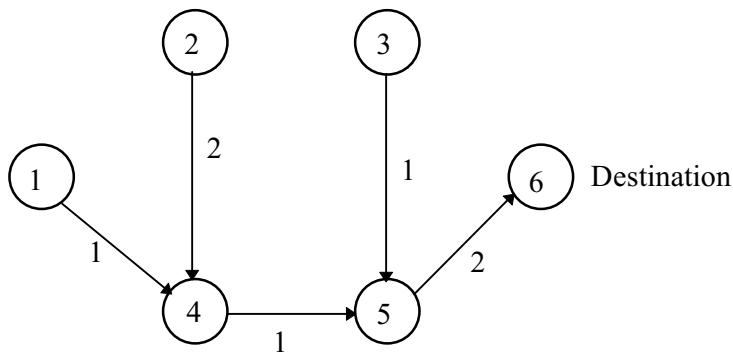
UPDATE ORDER 5,4,3,2,1

Iteration Number	Labels L(n), Current Predecessor Node P(n)				
	L(5), P(5)	L(4), P(4)	L(3), P(3)	L(2), P(2)	L(1), P(1)
1	2 6	3 5	3 5	5 4	4 4
2	2 6	3 5	3 5	5 4	4 4

UPDATE ORDER 1,2,3,4,5

Iteration Number	Labels L(n), Current Predecessor Node P(n)				
	L(1), P(1)	L(2), P(2)	L(3), P(3)	L(4), P(4)	L(5), P(5)
1	$\infty$ -	$\infty$ -	5 6	$\infty$ -	2 6
2	$\infty$ -	8 3	3 5	3 5	2 6
3	4 4	5 4	3 5	3 5	2 6
4	4 4	5 4	3 5	3 5	2 6

### SHORTEST PATH TREE



### β) Αλγόριθμος Dijkstra

Ο αλγόριθμος προϋποθέτει ότι όλοι οι σύνδεσμοι πρέπει να έχουν θετικά μήκη, γεγονός που ισχύει για εφαρμογές δικτύων υπολογιστών. Έστω  $D_i$  το μήκος της ελάχιστης

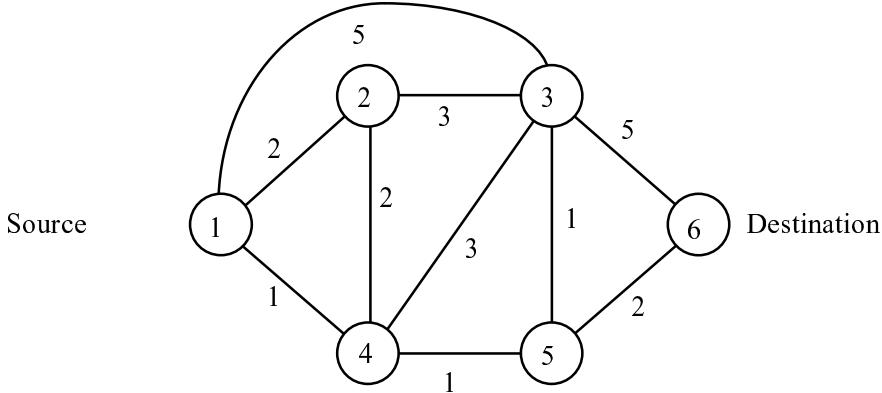
διαδρομής από τον κόμβο πηγή προς τον κόμβο  $j$  με την προϋπόθεση ότι στο συγκεκριμένο μονοπάτι όλοι οι κόμβοι πλην του τελευταίου κόμβου ανήκουν στο σύνολο  $P$ .

Αρχικά έχουμε:  $P=\{1\}$ ,  $D_1=0$ ,  $D_j=d_{1j} \forall j \neq 1$

Βήμα 1°: Βρες  $i \notin P$  τέτοιο ώστε:  $D_i = \min_{j \notin P} D_j$  και κάνε:  $P = P \cup \{i\}$  Εάν το  $P$  περιλαμβάνει όλους τους κόμβους τότε ο αλγόριθμος σταματά, αλλιώς:

Βήμα 2°: Για όλα  $j \notin P$   $D_j = \min[D_j, D_i + d_{ij}]$  και ξανά στο 1° Βήμα

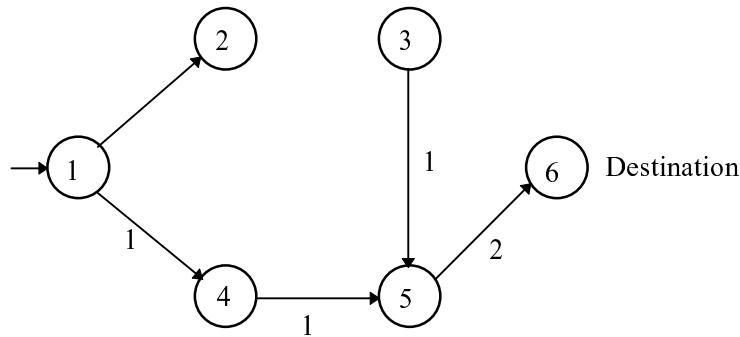
Σε κάθε βήμα ο αλγόριθμος απαιτεί έναν αριθμό πράξεων ανάλογο του  $N$  και έχουμε  $N-1$  βήματα. Δηλ. είναι αλγόριθμος πολυπλοκότητας  $O(N^2)$ .



INITIAL LABELS:  $L(1)=0$ ,  $L(2)=L(3)=\dots=L(6)=\infty$

Iteration Number	Permanently Labeled Nodes	Labels $L(n)$ , Current Predecessor Node $P(n)$				
		$L(2), P(2)$	$L(3), P(3)$	$L(4), P(4)$	$L(5), P(5)$	$L(6), P(6)$
1	1	2 1	5 1	1 1	$\infty$ -	$\infty$ -
2	1,4	2 1	4 4	-	2 4	$\infty$ -
3	1,4,2	-	4 4	-	2 4	$\infty$ -
4	1,4,2,5	-	3 5	-	-	4 5
5	1,4,2,5,3	-	-	-	-	4 5

### SHORTEST PATH TREE



γ) Αλγόριθμος Floyd-Warshall

Έστω  $D_{ij}^{(n)}$  είναι το μήκος της ελάχιστης διαδρομής που συνδέει τον κόμβο  $i$  με τον κόμβο  $j$  και με δεδομένο ότι οι ενδιάμεσοι κόμβοι σ' αυτήν την διαδρομή είναι οι  $1, 2, \dots, n$ .

Αρχικά:  $D_{ij}^{(0)} = d_{ij} \quad \forall i, j \text{ με } i \neq j$

Για:  $n = 0, 1, \dots, N-1$   $D_{ij}^{(n+1)} = \min [D_{ij}^{(n)}, D_{i(n+1)}^{(n)} + D_{(n+1)j}^{(n)}] \quad \forall i, j \text{ με } i \neq j$

Είναι αλγόριθμος πολυπλοκότητας  $O(N^3)$ , αλλά υπολογίζει τα ελάχιστα μονοπάτια μεταξύ όλων των ζευγαριών κόμβων.

## 2.4. Πρωτόκολλα Επιπέδου Μεταφοράς

Το επίπεδο μεταφοράς της ομάδας των TCP/IP πρωτοκόλλων περιλαμβάνει το **Transmission Control Protocol (TCP)** και το **User Datagram Protocol (UDP)**. Το πρώτο είναι μια αξιόπιστη υπηρεσία μεταφοράς με σύνδεση, ενώ το δεύτερο είναι μια μη αξιόπιστη υπηρεσία μεταφοράς χωρίς σύνδεση.

Είδαμε ότι με τη βοήθεια του πρωτοκόλλου IP μπορούν να μεταφέρονται δεδομένα μεταξύ υπολογιστών σε ένα μεγάλο διαδίκτυο, όπως είναι το Internet, το οποίο αποτελείται από πλήθος διασυνδεδεμένων δικτύων, με βάση την IP-διεύθυνση του προορισμού. Γεννιέται τώρα το ερώτημα, πώς θα αναγνωρίστεί η κατάλληλη εφαρμογή στον υπολογιστή-προορισμό στην οποία πρέπει να οδηγηθεί το datagram; Δηλ. αν στο συγκεκριμένο υπολογιστή τρέχουν δύο ή περισσότερα προγράμματα μεταφοράς αρχείων από δύο ή περισσότερους χρήστες, πώς το πρωτόκολλο IP θα επιλέξει σε ποιο από τα προγράμματα αυτά και σε ποιο συγκεκριμένο χρήστη του υπολογιστή θα παραδόσει το datagram;

Το λειτουργικό σύστημα σε πολλά υπολογιστικά μηχανήματα επιτρέπει την πολυεπεξεργασία, το οποίο σημαίνει ότι επιτρέπει σε πολλά προγράμματα να τρέχουν ταυτόχρονα. Ονομάζουμε κάθε ένα από τα προγράμματα αυτά: process, task, application program, user level process, κ.ά. Είναι φανερό ότι ο τελικός προορισμός των δεδομένων σε κάποια επικοινωνία θα είναι κάποιο από τα προγράμματα αυτά. Το να καθορίσει όμως κάποιος, ένα process σαν τον τελικό προορισμό των δεδομένων δεν είναι και τόσο απλό, για διάφορους λόγους, όπως:

- επειδή οι διεργασίες γεννιούνται και πεθαίνουν δυναμικά,
- συχνά οι διεργασίες αντικαθίστανται (π.χ. με κάποια επανεκκίνηση του υπολογιστή),
- συχνά μας ενδιαφέρουν μονάχα οι υπηρεσίες κάποιας διεργασίας και όχι αυτή καθαυτή η διεργασία (π.χ. μας ενδιαφέρει απλά κάποιος file-server).

Αντί λοιπόν να θεωρούμε κάποια διεργασία σαν το τελικό προορισμό των δεδομένων μας, μπορούμε να φανταστούμε ότι σε κάθε μηχάνημα υπάρχει ένα σύνολο από νοητά σημεία προορισμού, τα οποία ονομάζουμε **protocol ports**, και κάθε ένα από τα οποία αναγνωρίζεται από ένα θετικό ακέραιο αριθμό. Το λειτουργικό σύστημα του μηχανήματος έχει την ευθύνη της ανάθεσης των protocol ports σε διεργασίες, μέσω ενός interface, ώστε να είναι δυνατή η πρόσβαση στις διεργασίες αυτές.

Η πρόσβαση σε κάποια από τα protocol ports συνήθως είναι:

- **synchronous** (σύγχρονη), δηλ. το λειτουργικό σύστημα σταματά το process μέχρι την άφιξη δεδομένων,
- **buffered**, δηλ. δεδομένα που φθάνουν πριν η διεργασία να είναι έτοιμη να τα δεχθεί δεν χάνονται.

Συμπερασματικά, προκειμένου να επικοινωνήσει κανείς με κάποιο από τα ports που περιγράφαμε, θα πρέπει να γνωρίζει τόσο την IP-διεύθυνση του μηχανήματος προορισμού, όσο και το protocol port που τον ενδιαφέρει. Επιπρόσθετα τα μηνύματα θα μεταφέρουν, πέρα από το protocol port του προορισμού, και το protocol port της διεργασίας-πηγής των μηνυμάτων, το οποίο θα είναι χρήσιμο σε περίπτωση απόκρισης στο αρχικό μήνυμα.

## 2.4.1. Το πρωτόκολλο TCP

Για εφαρμογές που θέλουν να μετακινήσουν μεγάλες ποσότητες φορτίου μια μη αξιόπιστη υπηρεσία μεταφοράς πακέτων δεν είναι η καλύτερη λύση. Οι προγραμματιστές που υλοποιούν τη συγκεκριμένη εφαρμογή θα πρέπει να αναπτύξουν υψηλής ποιότητας ρουτίνες, προκειμένου να ελέγχουν λεπτομερώς τη μεταφορά του φορτίου πακέτο-πακέτο, μη επιτρέποντας να συμβεί το παραμικρό λάθος. Προκειμένου να αποφευχθεί ο συνεχής προγραμματισμός τέτοιων ρουτινών κάθε φορά που αναπτύσσεται κάποια εφαρμογή, στα πρωτόκολλα TCP/IP είναι αναγκαία μια υπηρεσία αξιόπιστης μεταφοράς φορτίου με σύνδεση (connection oriented reliable stream delivery services) και την οποία μπορεί να προσφέρει το πρωτόκολλο TCP.

### 2.4.1.1. Αξιόπιστη υπηρεσία μεταφοράς - stream

Όταν δύο διαδικασίες-εφαρμογής ανταλλάσσουν μεγάλες ποσότητες δεδομένων, θεωρούμε τα δεδομένα αυτά σαν μια σειρά από octets. Μια αξιόπιστη υπηρεσία μεταφοράς - stream, όπως είναι το TCP περνά στον προορισμό τα δεδομένα με την ίδια σειρά με την οποία τα έστειλε η διαδικασία πηγή.

Προκειμένου να το επιτύχει αυτό, το TCP εγκαθιστά ένα νοητό κύκλωμα μεταξύ των δύο σταθμών εργασίας και στην συνέχεια κάθε stream από octets που ξεκινάει από το ένα μηχάνημα φθάνει στο άλλο με την ίδια ακριβώς σειρά. Οι εφαρμογές που χρησιμοποιούν το TCP πρωτόκολλο, λαμβάνουν από το λειτουργικό σύστημα δύο αριθμούς προκειμένου η επικοινωνία τους να καθορίζεται καλά, σε σχέση με άλλες επικοινωνίες διεργασιών μεταξύ των ίδιων μηχανημάτων. Οι αριθμοί που λαμβάνουν ονομάζονται **TCP port numbers**. Παραδείγματα δεσμευμένων port numbers υπάρχουν στον Πίνακα 2.2 σε αντιστοιχία με τις υπηρεσίες.

Port	Περιγραφή
0	Reserved
1	TCP Multiplexor
5	Remote Job Entry
7	Echo
9	Discard
11	Active Users
13	Daytime
15	Network Status Program

17	Quote of the Day
19	Character Generator
20	File Transfer Protocol (data)
21	File Transfer Protocol
23	Terminal Connection
25	Simple Mail Transport Protocol
37	Time
42	Host Name Server
43	Who Is
53	Domain Name Server
77	any private RJE service
79	Finger
93	Device Control Protocol
95	SUPDUP Protocol
101	NIC Host Name Server
102	ISO - TSAP
103	X.400 Mail Server
104	X.400 Mail Sending
111	SUN Remote Procedure Call
113	Authentication Service
117	UUCP Path Service
119	USENET News Transfer Protocol
129	Password Generator Protocol
139	NETBIOS Session Service

**Πίνακας 2.2 - Δεσμευμένα TCP port numbers**

Πιο συγκεκριμένα η μια πλευρά που επιθυμεί να επικοινωνήσει εκτελεί μια **passive open** λειτουργία, ενημερώνοντας το λειτουργικό σύστημα του υπολογιστή, ότι στην συνέχεια θα δέχεται συνδέσεις στο συγκεκριμένο TCP port number. Η άλλη πλευρά, η οποία γνωρίζει το παραπάνω port number, εκτελεί μια **active open** λειτουργία και λαμβάνει από το λειτουργικό σύστημα ένα TCP port number προκειμένου να επικοινωνήσει με την άλλη πλευρά. Στη συνέχεια τα δύο TCP software modules επικοινωνούν μεταξύ τους, προκειμένου να εγκαταστήσουν και να επιβεβαιώσουν τη σύνδεση. Από τη στιγμή που η σύνδεση έχει εγκατασταθεί, οι διαδικασίες εφαρμογής μπορούν να ανταλλάσσουν δεδομένα, ενώ τα TCP software modules στα δύο άκρα εξασφαλίζουν την αξιόπιστη παράδοση των δεδομένων αυτών.

#### 2.4.1.2. Η λειτουργία του πρωτοκόλλου

Η μεταφορά δεδομένων με το TCP είναι full-duplex δηλ. και οι δύο διεργασίες μπορούν να στέλνουν και να λαμβάνουν ταυτόχρονα δεδομένα, δηλ. μπορούμε να έχουμε ταυτόχρονη μεταφορά δεδομένων και προς τις δύο διευθύνσεις.

Η μονάδα πληροφορίας πρωτοκόλλου στο TCP ονομάζεται **TCP segment**. Τα TCP segments μεταφέρονται στο πεδίο δεδομένων ενός IP datagram. Το λογισμικό του TCP αποθηκεύει τα bytes που πρέπει να μεταφέρει μέχρι να μπορεί να "γεμίσει" ένα IP datagram. Το stream των bytes δεν έχει μια συγκεκριμένη δομή. Δηλ. για έναν εξωτερικό παρατηρητή, ακόμα και αν μεταφέρεται ένα αρχείο με συγκεκριμένες εγγραφές (π.χ. λογαριασμούς πελατών κάποιας τράπεζας), ένα TCP segment δεν είναι

τίποτα περισσότερο από μια σειρά από 0 και 1. Οι δύο εφαρμογές που επικοινωνούν πρέπει να έχουν συμφωνήσει πάνω στην δομή των μεταδιδόμενων δεδομένων, (Βλ. και στο κεφάλαιο 4: Σύνταξη και κωδικοποίηση πληροφορίας (ASN.1 και BER)), πριν ξεκινήσουν την επικοινωνία τους.

Κάθε TCP segment έχει σε πεδίο με πληροφορίες ελέγχου έναν μοναδικό αριθμό, που το αναγνωρίζει σε σχέση με τα υπόλοιπα segments. Με τον τρόπο αυτό η διεργασία προορισμός πάντα ξέρει ποιο είναι το επόμενο segment που πρέπει να αποδεχθεί. Μόλις το λάβει στέλνει μια θετική επιβεβαίωση στη διεργασία πηγή, η οποία συνεχίζει την αποστολή των segments. Αν η διεργασία πηγή δεν δεχθεί μια θετική επιβεβαίωση για κάποιο segment εντός ενός προκαθορισμένου χρονικού διαστήματος, τότε επαναμεταδίδει το μη επιβεβαιωμένο segment. Αν η διεργασία προορισμός δεχθεί κάποιο segment δεύτερη φορά (π.χ. όταν έχει καθυστερήσει στο δίκτυο, και έχει επαναμεταδοθεί) μπορεί να το απορρίψει, έτσι όλα τα μειονεκτήματα του μη αξιόπιστου πρωτοκόλλου IP στο επιπέδο δικτύου ξεπερνιούνται. Ας δούμε, όμως πως λειτουργεί ο μηχανισμός αυτός της θετικής επιβεβαίωσης.

Το TCP είναι μια αξιόπιστη υπηρεσία μεταφοράς. Αυτό σημαίνει ότι οι εφαρμογές που το χρησιμοποιούν δεν χρειάζεται να ανησυχούν για χαμένα, καταστραμένα ή διπλά πακέτα μεταφοράς δεδομένων. Όλα αυτά είναι ευθύνη του πρωτοκόλλου. Προκειμένου να προσφέρει τις αξιόπιστες υπηρεσίες του το πρωτόκολλο TCP χρησιμοποιεί την τεχνική του συρόμενου παραθύρου (Sliding Window Technique). Δηλαδή επιτρέπει σε κάθε χρήστη του δικτύου να μεταδώσει ένα αριθμό πακέτων (Window Size) πριν αρχίσει να περιμένει (χωρίς να μεταδίδει) για τις επιβεβαιώσεις των πακέτων που έστειλε. Μ' αυτήν την τεχνική επιτυγχάνονται δύο δεδομένα:

Η αξιόπιστη υπηρεσία μεταφοράς αφού κάθε δεδομένο που μεταδίδεται ελέγχεται για το αν έφθασε στον σωστό προορισμό του, για το αν έφθασε σωστό, για το αν έφθασε δύο φορές κτλ.

Γίνεται σωστή χρήση του δικτύου αντίθετα με την περίπτωση που ένα πακέτο μεταδιδόταν και στην συνέχεια έπρεπε να έρθει η επιβεβαίωση γι' αυτό πριν μεταδοθεί το δεύτερο.

Πέρα όμως από το πρόβλημα της σωστής μεταφοράς δεδομένων και της ικανοποιητικής χρησιμοποίησης του δικτύου το πρωτόκολλο TCP με το μηχανισμό του συρόμενου παραθύρου λύνει και ένα δεύτερο πρόβλημα. Τον έλεγχο ροής του δικτύου από άκρη σε άκρη (End-to-End Flow Control), δίνοντας την δυνατότητα στον προορισμό να περιορίσει τον ρυθμό μετάδοσης μέχρι να αδειάσουν οι καταχωρητές του.

Ο μηχανισμός του συρόμενου παραθύρου που χρησιμοποιεί το πρωτόκολλο TCP είναι κάπως πιο περίπλοκος. Κατ' αρχήν, αναφέρουμε ότι και ο δέκτης των δεδομένων διατηρεί ένα ίδιο παράθυρο, προκειμένου να τοποθετεί τα δεδομένα με την ίδια σειρά και να τα παραδίδει μόλις ολοκληρωθεί η λήψη τους στην κατάλληλη εφαρμογή. Επίσης σε κάθε πακέτο επιβεβαίωσης ο δέκτης βάζει μια ένδειξη του διαθέσιμου χώρου που υπάρχει στους καταχωρητές του (Window Advertisement). Μ' αυτόν τον τρόπο ο πομπός ενημερώνεται και έτσι μπορεί να μεταβάλει το μέγεθος του δικού του παραθύρου, είτε προς τα πάνω, είτε προς τα κάτω. Τον τρόπο θα τον δούμε παρακάτω.

Το πλεονέκτημα του μεταβλητού μήκους παραθύρου είναι αυτό που επιτυγχάνει τον σωστό έλεγχο ροής από άκρη σε άκρη. Στέλνοντας ένα μηδενικό διαθέσιμο χώρο ο δέκτης μπορεί να διακόψει όλες τις μεταδόσεις, έτσι ώστε το δίκτυο ή η σύνδεση να μπορέσει να επανέρθει μετά από μια άσχημη κατάσταση.

Για έναν ουσιαστικό όμως έλεγχο ροής ο μηχανισμός του παραθύρου δεν είναι η λύση όλων των προβλημάτων. Το TCP χρησιμοποιεί επιπλέον ένα έξυπνο τρόπο αντιμετώπισης των προβλημάτων του χρόνου αναμονής των επιβεβαιώσεων (Timeout)

και των επαναμεταδόσεων. Ας θυμηθούμε κατ' αρχήν ότι το TCP είναι προορισμένο για δίκτυα που δεν μπορούν να προσφέρουν αξιόπιστες υπηρεσίες μεταφοράς, όπως το Internet και τα οποία αποτελούνται από πολλά διαφορετικά υποδίκτυα.

Προκειμένου λοιπόν να αντιμετωπίσει όλες τις πιθανές καθυστερήσεις που θα συναντήσει κάποιο πακέτο, χρησιμοποιεί ένα προσαρμόζομενο αλγόριθμο επαναμετάδοσης που παρακολουθεί τις καθυστερήσεις που παρακολουθεί τις καθυστερήσεις σε όλες τις συνδέσεις και καθορίζει τις παραμέτρους λήξης των χρονομέτρων ανάλογα.

Κάθε φορά που το TCP έχει μια καινούργια μέτρηση για το χρόνο καθυστέρησης (Sample Round Trip Time) ρυθμίζει με βάση την νέα τιμή και την παλιά εκτίμηση, το τι θα θεωρεί στο εξής σαν χρόνο καθυστέρησης. Στην συνέχεια ο χρόνος αναμονής υπολογίζεται σαν μια συνάρτηση του χρόνου καθυστέρησης.

Προβλήματα εισάγονται ως προς το τι θεωρείται ακριβή μέτρηση του round trip time. Αν ένα πακέτο επαναμεταδοθεί, τότε είναι συνολική καθυστέρηση ο χρόνος μεταξύ της αρχικής μετάδοσης και της λήψης της επιβεβαίωσης, ή μεταξύ της επαναμετάδοσης και της επιβεβαίωσης; Και οι δύο μετρήσεις παρουσιάζουν προβλήματα αν χρησιμοποιηθούν χωρίς να αλληλοχρησιμοποιούνται.

Το πρόβλημα λύθηκε με τον αλγόριθμο του Karn, σύμφωνα με τον οποίο όταν υπολογίζεται μια εκτίμηση του round trip time αγνοούνται πακέτα που επαναμεταδόθηκαν. Στην περίπτωση όμως που δεν φθάνει μια επιβεβαίωση και ένα πακέτο πρέπει να επαναμεταδοθεί, χρησιμοποιείται μια μέθοδος backoff. Με την μέθοδο αυτή η αρχική εκτίμηση του round trip time επαναπροσδιορίζεται (συνήθως διπλασιάζεται) μέχρι να επιτευχθεί η μετάδοση του πακέτου. Μόλις ένα πακέτο μεταδοθεί σωστά με την πρώτη προσπάθεια, η εκτίμηση του round trip time ξαναϋπολογίζεται από την αρχή.

Ένα πρωτόκολλο μονάχα με τους μηχανισμούς αυτούς δεν μπορεί να κάνει ουσιαστικό έλεγχο ροής και δεν μπορεί να αποφύγει κάποια κατάσταση συμφόρησης. Μια και η συμφόρηση οφείλεται συνήθως σε υπερχείλιση των καταχωρητών των ενδιάμεσων μεταγωγικών κόμβων ενός δικτύου, οι παραπάνω μηχανισμοί επαναμεταδίδοντας τα πακέτα που καθυστερούν απλά χειροτερεύουν την κατάσταση του δικτύου. Ο μηχανισμός που λύνει αυτά τα προβλήματα είναι ο μηχανισμός του παραθύρου που αναφέραμε και παραπάνω.

Για την αποφυγή καταστάσεων συμφόρησης το πρωτόκολλο TCP προτείνει δύο μηχανισμούς που πλαισιώνουν τον μηχανισμό του συρόμενου παραθύρου. Οι μηχανισμοί αυτοί είναι:

Ο μηχανισμός της πολλαπλασιαστικής μείωσης (Multiplicative Decrease Congestion Avoidance) σύμφωνα με τον οποίο σε περίπτωση χασίματος ενός πακέτου, το παράθυρο συμφόρησης μειώνεται στο μισό (μέχρι ένα ελάχιστο ίσο με ένα πακέτο). Αφού μειωθεί το παράθυρο στο μισό για τα πακέτα που παραμένουν σ' αυτό αυξάνεται ο χρονιστής επαναμετάδοσης εκθετικά (exponential backoff).

Αργού ξεκινήματος (Slow Start Recovery). Όταν η κυκλοφορία ξεκινά σε μια καινούργια σύνδεση, ή αυξάνεται μετά από μια κατάσταση συμφόρησης, το μέγεθος του παραθύρου συμφόρησης τίθεται να είναι ένα πακέτο. Με κάθε άφιξη επιβεβαίωσης (εννοείται χωρίς επαναμετάδοση) αυξάνεται κατά ένα. Στο σημείο αυτό θα προσθέσουμε την σχέση:

$$\text{Allowed\_Window} = \text{MIN}(\text{Receiver\_Advertisement}, \text{Congestion\_Window})$$

Το Allowed\_Window είναι το μέγεθος παραθύρου που χρησιμοποιείται. Τα άλλα δύο μεγέθη παραθύρων αναφέραμε ήδη πως καθορίζονται. Πιστεύουμε ότι με όλα τα παραπάνω η διεξαγωγή του ελέγχου ροής, αλλά και του ελέγχου συμφόρησης στα δίκτυα TCP/IP είναι ξεκάθαρη.

## 2.4.2. Το πρωτόκολλο UDP

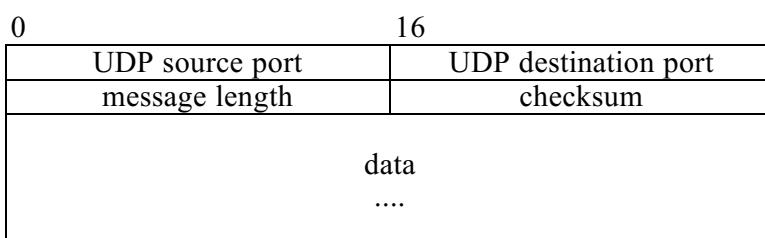
Το **User Datagram Protocol (UDP)** επιτρέπει στις TCP/IP εφαρμογές να ανταλλάσσουν μονοσήμαντα ανεξάρτητα μηνύματα πληροφορίας πάνω από ένα δίκτυο σε ένα περιβάλλον πολυεπεξεργασίας (βλ. για παράδειγμα τα SNMP μηνύματα, τα οποία μεταφέρουν πληροφορία που αφορά τη διαχείριση του δικτύου).

Το UDP πρωτόκολλο προσφέρει μια μη αξιόπιστη υπηρεσία μεταφοράς χωρίς σύνδεση, χρησιμοποιώντας το πρωτόκολλο IP για τη μεταφορά των μηνυμάτων μεταξύ μηχανών. Δεν χρησιμοποιεί επιβεβαιώσεις, δεν αριθμεί τα μηνύματα και δεν ελέγχει τη ροή τους. Έτσι ένα UDP μήνυμα μπορεί να χαθεί ή να φθάσει σε δύο αντίγραφα, ενώ UDP μηνύματα μπορεί να φθάσουν με διαφορετική σειρά από αυτήν που ξεκίνησαν. Επιπλέον τα UDP μηνύματα μπορεί να φθάσουν σε μία διεργασία συχνότερα απ' ότι αυτή μπορεί να τα επεξεργαστεί.

Όλα τα παραπάνω σημαίνουν ότι μια εφαρμογή που χρησιμοποιεί το UDP, θα πρέπει να λύσει το πρόβλημα της αξιοπιστίας, ελέγχοντας για χαμένα, διπλά, καθυστερημένα, και με αλλαγμένη σειρά μηνύματα. Επίσης ανώμαλος τερματισμός της σύνδεσης είναι πιθανός, ειδικά σε μεγάλα διαδίκτυα. Σε μια τέτοια περίπτωση η άλλη πλευρά μπορεί να συνεχίζει να φορτίζει το δίκτυο με UDP datagrams.

Αξίζει να τονίσουμε, ότι ο έλεγχος της TCP/IP εφαρμογής, που χρησιμοποιεί το UDP πρωτόκολλο μεταφοράς, και η επιτυχής της λειτουργία σε ένα περιορισμένο περιβάλλον ενός αξιόπιστου τοπικού δικτύου υψηλής ταχύτητας δεν εγγυάται την λειτουργία της εφαρμογής και σε ένα περιβάλλον όπως το Internet.

Κάθε μονάδα πληροφορίας του UDP πρωτοκόλλου ονομάζεται επίσης datagram. Οπως στο TCP έτσι και στο UDP ζευγάρια αριθμών 16 bit χρησιμοποιούνται για να ξεχωρίζουν τις διεργασίες που επικοινωνούν. Δηλ. κάθε UDP μήνυμα μαζί με τα δεδομένα μεταφέρει και τα port numbers για την πηγή και τον προορισμό του μηνύματος. Το παρακάτω σχήμα παρουσιάζει τη μορφή ενός UDP datagram, όπου το length μας δίνει σε octets το συνολικό μήκος του UDP datagram, ενώ η χρήση του checksum είναι προαιρετική. Παραδείγματα δεσμευμένων UDP port numbers υπάρχουν στον Πίνακα 2.3 καθώς και οι αντίστοιχες υπηρεσίες.



Port	Περιγραφή
7	Echo
9	Discard
11	Active Users
13	Daytime

15	Who is up or NETSTAT
17	quote of the day
19	character generator
37	time
42	hostname server
43	whois
53	domain name server
67	Bootstrap protocol server
68	Bootstrap protocol client
69	Trivial File Transfer
111	SUN Microsystems RPC
123	network time protocol
161	snmp net monitor
162	snmp traps
512	UNIX comsat
513	UNIX rwho daemon
514	system log
515	time daemon

**Πίνακας 2.3 - Λεσμενόντα UDP port numbers**

## 2.5. Εφαρμογές TCP/IP

Στην παράγραφο αυτή, θα εξετάσουμε υψηλού επιπέδου υπηρεσίες του Internet και τα πρωτόκολλα που τις υποστηρίζουν. Οι υπηρεσίες αυτές χρησιμοποιούνται ουσιαστικά από τους ίδιους τους χρήστες ή από προγράμματα αυτών.

### 1. Η εφαρμογή FTP

Η μεταφορά αρχείων είναι από τις πιο συχνά χρησιμοποιούμενες TCP/IP εφαρμογές και δημιουργεί σημαντικό μέρος του φορτίου, το οποίο τρέχει σε κάποιο δίκτυο. Το πιο σημαντικό από τα πρότυπα (Standards) πρωτόκολλα μεταφοράς αρχείων είναι το File Transfer Protocol (FTP).

Μερικά από τα πλεονεκτήματα που έχει το FTP, σαν τρόπος μεταφοράς αρχείων, είναι τα παρακάτω:

- α) Διαλογική επικοινωνία του χρήστη με τον απομακρυσμένο server, με σκοπό την σωστή πληροφόρηση του χρήστη (π.χ. λίστα αρχείων στο απομακρυσμένο directory).
- β) Καθορισμός από τον χρήστη του είδους και της μορφής που θα έχουν τα δεδομένα που θα μεταφερθούν.
- γ) Έλεγχος από τον server για το αν ο χρήστης έχει δικαιώματα πρόσβασης στο απομακρυσμένο μηχάνημα και μεταφοράς αρχειών (π.χ. login, password). Επίσης ζεπερνάει τα προβλήματα που δημιουργούνται με τους παραπάνω περιορισμούς με τα περιορισμένα δικαιώματα πρόσβασης που προσφέρει το login anonymous, quest.

Ο FTP server υλοποιείται όπως και οι υπόλοιπές εφαρμογές. Υπάρχει μια διαδικασία η οποία περιμένει παθητικά για την δημιουργία συνδέσεων, και η οποία δημιουργεί μια διαδικασία παιδί για την εξυπηρέτηση καθεμιάς σύνδεσης. Ένα FTP session περιλαμβάνει κατ' αρχήν μια σύνδεση ελέγχου και κάποιο αριθμό συνδέσεων μεταφοράς

δεδομένων. Η διαδικασία παιδί για την οποία μιλήσαμε παραπάνω, εξυπηρετεί μόνο την σύνδεση ελέγχου. Χρησιμοποιεί άλλες διαδικασίες για την εξυπηρέτηση των συνδέσεων μεταφοράς δεδομένων.

Οι συνδέσεις μεταφοράς δεδομένων και οι διαδικασίες μεταφοράς δεδομένων που τις χρησιμοποιούν, μπορούν να δημιουργηθούν δυναμικά, όποτε αυτό είναι απαραίτητο, ενώ η σύνδεση ελέγχου παραμένει καθ' όλη την διάρκεια του session. Το RFC 959 περιέχει ολόκληρο τον ορισμό του πρωτοκόλλου.

## 2. Η εφαρμογή TFTP

Αν και το FTP είναι το πιο γενικό πρωτοκόλλο μεταφοράς αρχείων στην TCP/IP κοινότητα, είναι επίσης το πιο πολύπλοκο και δύσκολο να υλοποιηθεί. Πολλές εφαρμογές δεν χρειάζονται την πλήρη λειτουργικότητα που προσφέρει το FTP, ούτε την μεγάλη πολυπλοκότητά του. Για παράδειγμα, το FTP απαιτεί από client και servers την διαχείριση πολλαπλών παράλληλων TCP συνδέσεων, το οποίο είναι αδύνατο για κάποιο προσωπικό υπολογιστή (PC). με ένα κοινό λειτουργικό σύστημα.

Έτσι τα TCP/IP πρωτόκολλα περιλαμβάνουν ένα δεύτερο πρωτόκολλο μεταφοράς αρχείων, το οποίο παρέχει φθηνές και μη περίπλοκες υπηρεσίες, το Trivial File Transfer Protocol (TFTP), σε εφαρμογές που δεν χρειάζονται πολύπλοκες συναλλαγές.

Το μικρό μέγεθος του TFTP επιτρέπει στους κατασκευαστές να το κωδικοποιούν σε read-only memory (ROM) προκειμένου σταθμοί εργασίας χωρίς δίσκο, να μπορούν να λαμβάνουν από κάποιο εξυπηρετητή κάποιες σελίδες μνήμης κατά την αρχικοποίησή τους. Το πρόγραμμα αυτό σε ROM ονομάζεται system bootstrap.

Αντίθετα με το FTP, το TFTP δεν χρησιμοποιεί αξιόπιστη υπηρεσία μεταφοράς, αλλά μη αξιόπιστη, όπως για παράδειγμα το UDP πρωτόκολλο. Βέβαια, χρησιμοποιεί το ίδιο χρονιστές και επαναμεταδόσεις για την διασφάλιση της μεταφοράς των δεδομένων. Η πλευρά που στέλνει τα δεδομένα χρησιμοποιεί blocks σταθερού μήκους 512 bytes και περιμένει επιβεβαίωση πριν στείλει κάθε φορά το επόμενο block.

## 3. Η εφαρμογή SMTP

Τα TCP/IP πρωτόκολλα καθορίζουν επίσης κάποιο πρότυπο για την ανταλλαγή ταχυδρομείου μεταξύ δύο μηχανημάτων. Το πρότυπο αυτό καθορίζει την ακριβή μορφή των ανταλλασσόμενων μηνυμάτων μεταξύ της διεργασίας πελάτη στην μία μηχανή, που στέλνει το ταχυδρομείο και της διεργασίας εξυπηρετητή στην άλλη μηχανή, που λαμβάνει το ταχυδρομείο. Το πρότυπο αυτό ονομάζεται Simple Mail Transfer Protocol (SMTP). Το SMTP πρωτόκολλο καθορίζει πώς το σύστημα ταχυδρομείου στέλνει μηνύματα από ένα μηχάνημα σε ένα άλλο. Δεν καθορίζει πώς το σύστημα δέχεται το ταχυδρομείο από τον χρήστη ή πως του παρουσιάζει το εισερχόμενο ταχυδρομείο.

Το πρωτόκολλο SMTP έχει επίσης απλή μορφή. Η επικοινωνία μεταξύ πελάτη και εξυπηρετητή χρησιμοποιεί κάποιο transcript, το οποίο μπορεί να διαβαστεί. Κάθε μήνυμα ξεκινάει με μια τριψήφια εντολή, η οποία ακολουθείται από κάποιο κείμενο.

## 4. Η εφαρμογή TELNET

Το TELNET είναι ένα απλό πρωτόκολλο απομακρυσμένου τερματικού. Επιτρέπει σε κάποιο χρήστη να εγκαταστήσει μια σύνδεση TCP με έναν εξυπηρετητή σε μια απομακρυσμένη μηχανή, και στην συνέχεια να μπορεί να πληκτρολογεί σαν να ήταν

χρήστης τερματικού της μηχανής αυτής. Την ίδια χρονική στιγμή βέβαια, το TELNET επιστρέφει την έξοδο στην οθόνη του τερματικού του χρήστη.

Οι δύο διεργασίες (πελάτης και εξυπηρετητής) επικοινωνούν μεταξύ τους χρησιμοποιώντας κατάλληλες εντολές - οι οποίες είναι πολύ απλοί χαρακτήρες ή σειρές από χαρακτήρες - και κωδικοποιούνται σε μια πρότυπη μορφή, γνωστή σαν network virtual terminal (NVT). Το σύνολο χαρακτήρων που χρησιμοποιούνται για τις εντολές είναι το ASCII, ενώ σε περίπτωση που κάποιο από τα δύο μηχανήματα δεν χρησιμοποιεί αυτό το σύνολο χαρακτήρων, η τοπική διεργασία TELNET αναλαμβάνει την απεικόνιση, δηλ. παρέχει και υπηρεσίες επιπέδου παρουσιάσης κατά το μοντέλο αναφοράς πρωτοκόλλων ISO/OSI.

## 2.6. Προγραμματισμός με sockets στο Unix\*. Παραδείγματα

Τα πρωτόκολλα TCP/IP παρέχουν στον προγραμματιστή την πλατφόρμα για υλοποίηση εφαρμογών κατανεμημένης επεξεργασίας. Η επικοινωνία εφαρμογών σε περιβάλλοντα UNIX (και σήμερα MS-WINDOWS) μπορεί να γίνει με μηχανισμούς sockets, που αποτελούν το πιο δημοφιλές API (Application Programming Interface) μεταξύ TCP/IP και εφαρμογών. Η γνώση προγραμματισμού με sockets είναι απαραίτητη για την ανάπτυξη εφαρμογών Διαχείρισης όπως θα φανεί στο 4ο Κεφάλαιο (Προτυπα και Πρωτόκολλα Διαχείρισης TCP/IP).

Σε ένα σύστημα που υποστηρίζει την πολυεπεξεργασία όπως το UNIX, είναι απαραίτητη η επικοινωνία μεταξύ διεργασιών που τρέχουν παράλληλα, έτσι ώστε να επιτυγχάνεται συγχρονισμός, καθώς και ανταλλαγή πληροφοριών όταν αυτό είναι απαραίτητο. Γνωστοί τρόποι με τους οποίους επιτυγχάνεται αυτό είναι οι σηματοφορείς (semaphores) και τα pipes μηνυμάτων. Οι παραπάνω τρόποι καθιστούν δυνατή την επικοινωνία μεταξύ διεργασιών με ένα όμως σοβαρό περιορισμό: οι διεργασίες που επικοινωνούν πρέπει να τρέχουν στο ίδιο μηχάνημα και επιπλέον να έχουν κάποιο κοινό πρόγονο (π.χ. να είναι διεργασίες-παιδιά της ίδιας διεργασίας-πατέρα). Τι γίνεται όμως στην περίπτωση που θέλουμε δύο εντελώς ανεξάρτητες διεργασίες που τρέχουν σε διαφορετικές μηχανές να επικοινωνήσουν; Για την επικοινωνία αυτή πρέπει να χρησιμοποιηθεί το επικοινωνιακό υποδίκτυο που ενώνει τις δύο μηχανές. Συνεπώς ενώ οι δύο εφαρμογές τρέχουν στο επίπεδο εφαρμογών πρέπει για την επικοινωνία τους να εμπλακούν και τα χαμηλότερα επίπεδα και πιο ειδικά τα επίπεδα μεταφοράς και δικτύου (όσο αναφορά τα TCP/IP πρωτόκολλα). Την επικοινωνία αυτή καθιστούν δυνατή τα sockets, που πρωτοεμφανίστηκαν στην έκδοση 4.2BSD του Berkley Unix. Με δύο λόγια τα sockets αποτελούν μια δυνατή διασύνδεση\* του λογισμικού υλοποίησης των εφαρμογών με το λογισμικό υλοποίησης των χαμηλοτέρων επιπέδων του ISO/OSI μοντέλου αναφοράς πρωτοκόλλων. Παρακάτω θα εξετάσουμε αναλυτικότερα την διασύνδεση με την βοήθεια των sockets.

### 1. Μερικές εισαγωγικές έννοιες.

Ας θεωρήσουμε την απλή περίπτωση όπου μια διεργασία A χρειάζεται κάποιες πληροφορίες από κάποια άλλη διεργασία B. Τότε για να πάρει τις πληροφορίες αυτές πρέπει να επικοινωνήσει με τη διεργασία B και να τις ζητήσει. Η διεργασία B με τη σειρά της αφού δεί ότι κάποιος ζητάει τις πληροφορίες που κατέχει, κοιτάει ποιός είναι αυτός και του τις στέλνει. Με βάση αυτό το απλό μοντέλο μπορεί να περιγραφεί κάθε άλλη δυνατή περίπτωση επικοινωνίας ανάμεσα σε δύο διεργασίες. Μια από αυτές

\* Η ενότητα που ακολουθεί βασίζεται σε υλικό που συγκέντρωσε ο B.Μωραΐτης, Επιστημονικός Συνεργάτης του Εργαστηρίου NETMODE στο Ε.Μ.Π.

\* Μια άλλη δυνατή διασύνδεση είναι το Transport Library Interface (TLI) του AT&T System V.

χρειάζεται μια υπηρεσία, που η άλλη μπορεί να προσφέρει. Το μοντέλο αυτό είναι γνωστό σαν **μοντέλο πελάτη - εξυπηρετητή** (client - server model), όπου η διεργασία που ζητά την υπηρεσία είναι ο πελάτης και η διεργασία που μπορεί να προσφέρει τη συγκεκριμένη υπηρεσία είναι ο εξυπηρετητής. Ανάλογα με τον τρόπο που χειρίζεται ο εξυπηρετητής τις αιτήσεις για παροχή υπηρεσιών, διακρίνουμε της εξής περιπτώσεις:

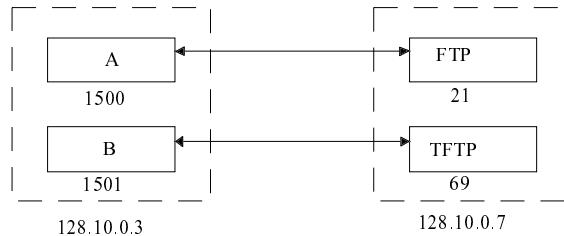
- (α). Κάθε φορά που ο εξυπηρετητής δέχεται μια αίτηση για παροχή υπηρεσίας δημιουργεί μια διεργασία παιδί που αναλαμβάνει την εξυπηρέτηση της αίτησης, ενώ ο ίδιος περιμένει να παραλάβει την επόμενη. Σε αυτή την περίπτωση έχουμε έναν **παράλληλο (concurrent) εξυπηρετητή**.
- (β). Όταν ο εξυπηρετητής δέχεται μια αίτηση τότε την εξυπηρετεί ο ίδιος και όταν τελειώσει τότε μπαίνει σε αναμονή για την επόμενη αίτηση. Έχουμε τότε έναν **συμπαγή (iterative) εξυπηρετητή**. Η ανάγκη για buffering και στις δύο περιπτώσεις εξυπηρετητή είναι προφανής, αλλά στην περίπτωση αυτή είναι πιο έντονη.

Τέλος, ανάλογα με το πρωτόκολλο που χρησιμοποιείται στο επίπεδο δικτύου, διακρίνουμε τις περιπτώσεις **connection oriented επικοινωνίας** (όταν το χρησιμοποιούμενο πρωτόκολλο αποκαθιστά επιοινωνία νοητού κυκλώματος - virtual circuit - ανάμεσα στις επικοινωνούσες διεργασίες) και **connectionless επικοινωνίας** (όταν το πρωτόκολλο χρησιμοποιεί datagrams για επικοινωνία).

## 2. Μια συσχέτιση (association) στο Unix.

Για να καθορίσουμε μια σύνδεση ανάμεσα σε δύο διεργασίες χρησιμοποιούμε την έννοια της συσχέτισης. Μια συσχέτιση καθορίζει ακριβώς τις διεργασίες οι οποίες επικοινωνούν και είναι μια πεντάδα της μορφής:

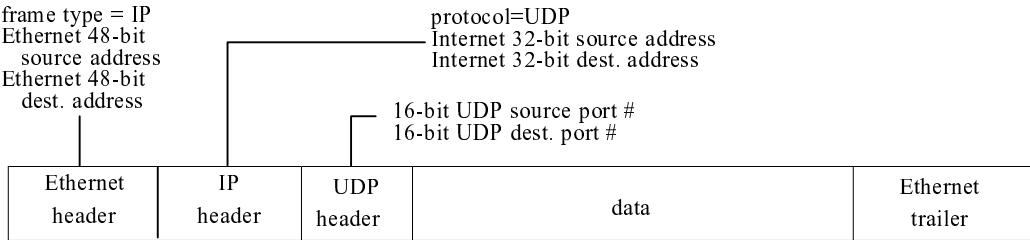
{πρωτόκολλο,τοπική-διευθ,τοπική-διεργ,ξένη-διευθ,ξένη-διεργ}



**Σχήμα 2.9 - Επικοινωνία διεργασιών**

όπου πρωτόκολλο είναι το πρωτόκολλο επικοινωνίας (TCP/UDP για τα πρωτόκολλα της Internet με τα οποία θα ασχοληθούμε εμείς), τοπική-διευθ και ξένη-διευθ είναι οι Internet διευθύνσεις (4 bytes) των μηχανών πάνω στις οποίες τρέχουν οι επικοινωνούσες διεργασίες, και τοπική-διεργ και ξένη-διεργ είναι δύο 16-μπιτοί αριθμοί, οι οποίοι καθορίζουν επακριβώς τις επικοινωνούσες διεργασίες πάνω στις μηχανές στις οποίες τρέχουν. Οι αριθμοί αυτοί ονομάζονται στα πρωτόκολλα της Internet port numbers και καθορίζουν μοναδικά την θύρα (port) μέσω της οποίας γίνεται η επικοινωνία κάθε διεργασίας με τον υπόλοιπο κόσμο που χρησιμοποιεί τα sockets. Για να διευκρινίσουμε, ας δούμε το Σχήμα 2.9. Στο Σχήμα 2.9 αυτό έχουμε δύο διεργασίες τις A και B στη μηχανή με (Internet) διεύθυνση 128.10.0.3 και δύο εξυπηρετητές στη μηχανή με διεύθυνση 128.10.0.7. Η διεργασία A ζητάει την υπηρεσία FTP και στέλνει την αίτηση της στη διεύθυνση 128.10.0.7. Στη διεύθυνση όμως αυτή υπάρχει και ο εξυπηρετητής του TFTP. Πώς θα γνωρίζει το σύστημα σε ποιόν από τους δύο εξυπηρετητές ανήκει η αίτηση; Η απάντηση είναι ότι η διεργασία A όταν στείλει

την αίτηση της θα καθορίσει και τον καθορισμένο αριθμό θύρας (well-known port number) του εξυπηρετητή από τον οποίο ζητά την υπηρεσία.



### Σχήμα 2.10 - Ενθυλάκωση πληροφορίας ψηλότερων επιπέδων σε πλαίσιο

Σε κάθε σύστημα Unix υπάρχει ένας αριθμός από δεσμευμένους (reserved) αριθμούς θυρών οι οποίοι χρησιμοποιούνται για αναφορά στις standard υπηρεσίες που προσφέρει κάθε σύστημα Unix, όπως τα FTP, TFTP, TELNET και SMTP. Οι υπόλοιποι 16-μπιτοι αριθμοί θυρών μπορούν να χρησιμοποιηθούν από εμάς για την υλοποίηση των δικών μας servers. Στο παράδειγμα του Σχήματος 2.9 η υπηρεσία FTP έχει δεσμευμένο τον καθορισμένο αριθμό θύρας 21 τον οποίο πρέπει να γνωρίζει η διεργασία A για να επικοινωνήσει με τον εξυπηρετητή FTP. Όταν ο εξυπηρετητής λάβει το μήνυμα από τη διεργασία A θα βρει μέσα στο πακέτο τον αριθμό θύρας και την διέυθυνση από την οποία προήλθε το πακέτο και έτσι θα μπορέσει να απαντήσει σωστά στην διεργασία A και όχι στη B η οποία επίσης τρέχει στο ίδιο μηχάνημα με την A και επικοινωνεί με τον TFTP εξυπηρετητή. Στο Σχήμα 2.10 φαίνεται ο τρόπος με τον οποίο η πληροφορία για τον αριθμό θύρας περνάει στο πακέτο που διακινείται μέσα σε ένα δίκτυο Ethernet.

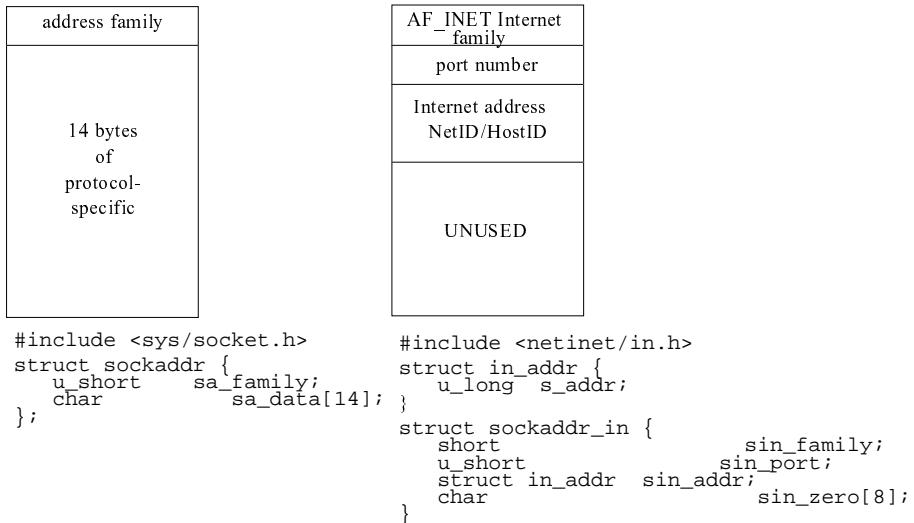
### 3. Χρήσιμες δομές δεδομένων - συναρτήσεις βιβλιοθήκης.

Στις επόμενες παραγράφους θα δούμε τις πρωτογενείς κλήσεις που υποστηρίζει το Unix για την χρήση των sockets. Σε πολλές από τις πρωτογενείς αυτές κλήσεις περνιούνται σαν παράμετροι δείκτες προς δομές τύπου sockaddr. Η δομή sockaddr είναι τελείως γενική και επιτρέπει στο Unix να υποστηρίζει πολλά πρωτόκολλα επικοινωνιών όπως το XNS της Xerox και τα Unix domain protocols. Στο Σχήμα 2.11 φαίνεται η δομή sockaddr και η δομή sockaddr\_in η οποία χρησιμοποιείται για τη διαχείριση πληροφοριών στα πρωτόκολλα της Internet. Ο τύπος u\_long είναι ορισμένος σαν unsigned long και ο τύπος u\_short σαν unsigned short μέσα στο αρχείο <sys/types.h>. Η σταθερά AF\_INET είναι ορισμένη στο αρχείο <sys/socket.h> και χρησιμοποιείται για την αρχικοποίηση ενός socket σε κάποιο από τα πρωτόκολλα επικοινωνίας της Internet. Όταν χρειάζεται να περαστεί σαν παράμετρος ένας δείκτης προς δομή τύπου sockaddr\_in εκεί που μια πρωτογενής κλήση περιμένει δείκτη τύπου sockaddr κάνουμε casting. Π.χ.

```

sockaddr_in host_addr;
sockaddr *addr;
.
.
.
addr=(sockaddr *) &host_addr;

```



### Σχήμα 2.11 - Οι δομές sockaddr και sockaddr\_in

Σε ένα άλλο θέμα τώρα. Ίσως έχει δημιουργηθεί η απορία πως βρίσκουμε τα port numbers για κάποιες συγκεκριμένες υπηρεσίες που προσφέρονται από το σύστημα, όπως π.χ. για το FTP. Πρέπει να θυμόμαστε δηλαδή ότι ο αριθμός θύρας για τον FTP server είναι 21; Πρέπει επίσης να θυμόμαστε την Internet διεύθυνση του μηχανήματος στο οποίο τρέχει ο FTP εξυπηρετητής; Όχι ακριβώς. Υπάρχουν δύο συναρτήσεις βιβλιοθήκης του Unix που μας βοηθούν να ξεπεράσουμε κάποια τέτοια προβλήματα. Τα πρωτότυπα τους ορίζονται στο αρχείο <netdb.h> και φαίνονται παρακάτω:

```

struct hostent *gethostbyname(char *hostname);
struct servent *getservbyname(char *servname, char
                             *proto)

```

Η πρώτη παίρνει σαν όρισμα ένα string που περιέχει το όνομα της μηχανής και επιστρέφει ένα δείκτη προς μια δομή τύπου hostent(ry). Η δεύτερη παίρνει σαν ορίσματα δύο string από τα οποία το πρώτο περιέχει το όνομα της υπηρεσίας (π.χ. FTP, TELNET) της οποίας ζητάμε τον αριθμό θύρας, ενώ το δεύτερο όρισμα μπορεί να είναι NULL ή να περιέχει το όνομα κάποιου πρωτοκόλλου (π.χ. UDP, TCP αφού κάποιες υπηρεσίες είναι διαθέσιμες είτε χρησιμοποιείται connection-oriented είτε connectionless πρωτόκολλο). Και στις δύο περιπτώσεις η συνάρτηση επιστρέφει ένα δείκτη προς δομή τύπου servent. Οι δομές hostent και servent είναι οι παρακάτω:

```

#include <netdb.h>
struct hostent {
    char *          h_name;
    char **        h_aliases;
    int            h_addr_type;
    int            h_length;
    char **        h_addr_list;
};

#define             h_addr           h_addr_list[0];

struct servent {
    char *          s_name;
    char **        s_aliases;
    int            s_port;
    char *          s_proto;
};

```

Στη δομή hostent, το πεδίο h\_name περιέχει το όνομα του μηχανήματος στο δίκτυο, το h\_addrtype έχει την τιμή AF\_INET για τα πρωτόκολλα της Internet και για τον ίδιο λόγο το h\_length έχει πάντα την τιμή 4. Αυτό όμως που μας ενδιαφέρει περισσότερο είναι το πεδίο h\_addr\_list το οποίο είναι ένας πίνακας από δείκτες όχι σε χαρακτήρες, αλλά σε δομές τύπου in\_addr (βλ. Σχήμα 2.11). Ο αριθμός των στοιχείων του πίνακα δεν είναι γνωστός, αλλά γνωρίζουμε ότι το τελευταίο του στοιχείο είναι ένας δείκτης NULL. Όπως θα έχει γίνει ήδη κατανοητό αυτές είναι οι διευθύνσεις Internet όλων των δικτύων στα οποία ίσως συμμετέχει το μηχάνημα. Επειδή στη γενικότερη περίπτωση η μηχανή θα ανήκει σε ένα μόνο δίκτυο έχει οριστεί ένα macro ώστε να είναι ευκολότερη η προσπέλαση του πρώτου (και μοναδικού στην περίπτωση αυτή) στοιχείου του πίνακα h\_addr\_list. Το μοναδικό σημείο που ίσως έμεινε αδιευκρίνιστο είναι το όνομα που περνίεται σαν παράμετρος στην gethostbyname(). Συνηθίζεται να δίνονται ονόματα στις δίαφορες μηχανές ενός δικτύου (π.χ. Πήγασος ή Δαιδαλος εδώ στο Πολυτεχνείο). Τα ονόματα αυτά υπάρχουν συνήθως σε μια βάση δεδομένων στοαρχείο /etc/hosts. Η gethostbyname() κοιτάζει τις καταχωρήσεις στη βάση και επιστρέφει τα ζητούμενα δεδομένα μέσω της δομής hostent.

Τα πράγματα είναι εντελώς ανάλογα στη δομή servent όπου το πεδίο s\_name είναι το όνομα της υπηρεσίας, το s\_port είναι το port\_number με το οποίο συνδέεται η υπηρεσία και το s\_proto είναι το πρωτόκολλο επικοινωνίας κάτω από το οποίο παρέχεται η συγκεκριμένη υπηρεσία. Οι πληροφορίες οι οποίες επιστρέφει η συνάρτηση getservbyname() είναι αποθηκευμένες στο αρχείο /etc/services.

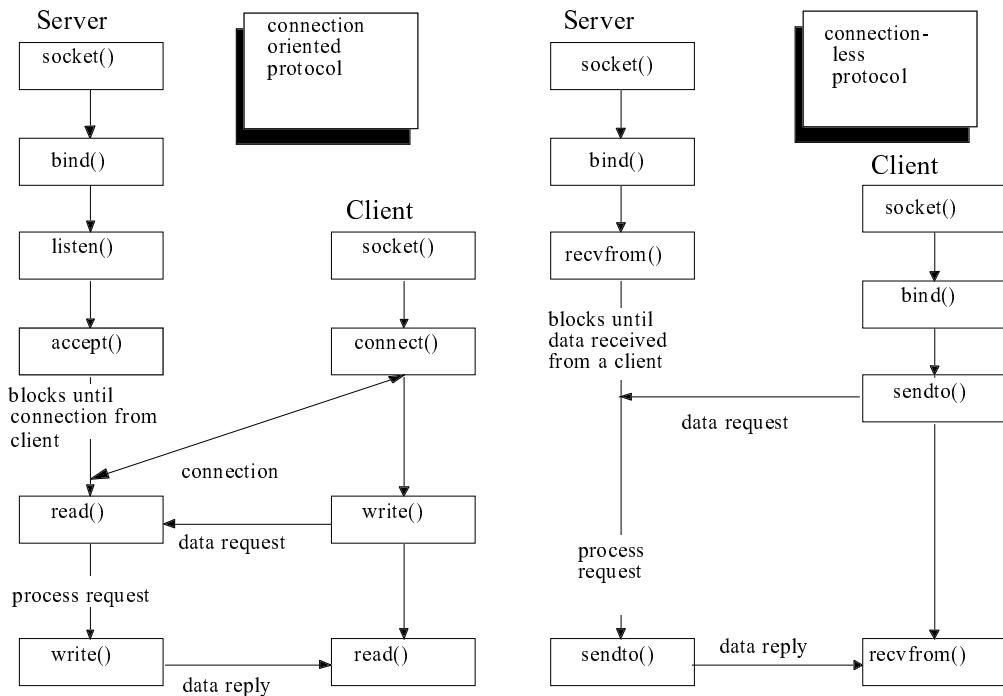
Όταν δημιουργούμε κάποια καινούρια υπηρεσία, η οποία δεν υπάρχει στο αρχείο /etc/services τότε πρέπει να γνωστοποιήσουμε τον αριθμό θύρας με τον οποίο ταυτοποιείται η παροχή της υπηρεσίας μας, αλλιώς δεν θα είναι σε θέση να προσπελασθεί από κανένα χρήστη.

#### 4. Οι πρωτογενείς κλήσεις για τα sockets.

Στη συνέχεια παρουσιάζουμε ένα σχήμα με τις περισσότερες πρωτογενείς κλήσεις του Unix για τα sockets στο Σχήμα 2.12, που περιλαμβάνει και τις δύο περιπτώσεις επικοινωνίας, με νοητό κύκλωμα και με datagrams. Θα εξετάσουμε τις κλήσεις αυτές μέσα στο πλάισιο της επικοινωνίας ενός εξυπηρετητή και ενός πελάτη και στους δύο δυνατούς τρόπους επικοινωνίας. Για οποιαδήποτε από τις παρακάτω κλήσεις δεν αναφέρονται κάποιες συγκεκριμένες οδηγίες #include εννοούνται οι:

```
#include <sys/types.h>
#include <sys/socket.h>
```

Κοινή η πρώτη ενέργεια για οποιαδήποτε διεργασία (πελάτη ή εξυπηρετητή) και για οποιοδήποτε τρόπο επικοινωνίας. Η κλήση



**Σχήμα 2.12 - Οι πρωτογενείς κλήσεις για τα sockets**

```
int socket(int family, int type, int protocol)
```

επιστρέφει σε περίπτωση επιτυχίας έναν ακέραιο ανάλογο με αυτό που οι ρουτίνες διαχείρισης αρχείων επιστρέφουν σαν περιγραφητή αρχείου (fd). Εδώ θα τον ονομάζουμε περιγραφητή `socket` (`sockfd`). Σε περίπτωση αποτυχίας επιστρέφεται αρνητικός αριθμός. Το όρισμα `family` για την οικογένεια πρωτοκόλλων Internet παίρνει την τιμή `AF_INET`. Το όρισμα `type` περιγράφει τον τύπο επικοινωνίας και παίρνει την τιμή κάποιας σταθεράς. Στη δική μας περίπτωση οι σταθερές είναι οι `SOCK_STREAM` και `SOCK_DGRAM`, που αντιστοιχούν στην `connection oriented` και `connectionless` επικοινωνία αντίστοιχα. Η παράμετρος `protocol` στις περισσότερες περιπτώσεις τίθεται ίση με 0, αναγκάζοντας έτσι το σύστημα να επιλέξει την σωστή τιμή για την παράμετρο αυτή. Η τιμή του `sockfd` που επιστρέφεται χρησιμοποιείται από τις επόμενες κλήσεις.

### Η κλήση

```
int bind(int sockfd, struct sockaddr *myaddr, int addrlen)
```

συνδέει το `sockfd` που έχει επιστραφεί από την προηγούμενη κλήση σε μια διεργασία, με μια τοπική διεύθυνση (αριθμό θύρας - port number), γνωστοποιούντας στο σύστημα ότι τα μηνύματα (πακέτα) που έρχονται στη θύρα αυτή απευθύνονται στη συγκεκριμένη διαδικασία και πρέπει να παραδίδονται σε αυτή. Αν θυμηθούμε τον ορισμό της συσχέτισης που δώσαμε στο μέρος 2, μπορούμε να παρατηρήσουμε ότι η κλήση `socket()` καθορίζει το πρώτο στοιχείο της πεντάδας (πρωτόκολλο) ενώ η κλήση `bind()` καθορίζει τα επόμενα δύο στοιχεία (τοπική διεύθ, τοπική διεργ). Όπως βλέπουμε και στο Σχήμα 2.12 η κλήση αυτή είναι επίσης απαραίτητη σε όλους εκτός από τον `connection-oriented` πελάτη. Το γιατί θα το δούμε παρακάτω. Τα πεδία της δομής `sockaddr` στην οποία δείχνει η παράμετρος `myaddr` πρέπει να αρχικοποιηθούν πρίν την κλήση της `bind()`, η οποία επιστρέφει σε περίπτωση αποτυχίας αρνητικό ακέραιο. Η παράμετρος `addrlen` περιέχει το μήκος (σε bytes) της δομής στην οποία δείχνει ο δείκτης `myaddr`.

Να μιλήσουμε τέλος για μια ειδική περίπτωση. Ας υποθέσουμε ότι έχουμε ένα μηχάνημα το οποίο συμμετέχει σε περισσότερα από ένα δίκτυα Internet, δηλαδή περιέχει περισσότερες από μία κάρτες διασύνδεσης με δίκτυο που υποστηρίζει τα πρωτόκολλα Internet. Για να γράψουμε ένα server ο οποίος να δέχεται μηνύματα από όλα τα δίκτυα στα οποία συμμετέχει, στο πεδίο `myaddr->sin_addr.s_addr` βάζουμε την τιμή της σταθεράς `INADDR_ANY` που είναι ορισμένη από το σύστημα και δεν χρειάζεται να κάνουμε `bind()` σε όλες τις διευθύνσεις με τις οποίες ταυτίζεται το μηχάνημα εξαιτίας των συνδέσεων του με πολλά δίκτυα.

## Η κλήση

```
int connect(int sockfd, struct sockaddr *servaddr, int addrlen)
```

συνδέει τον περιγραφητή socket που έχει επιστραφεί με την κλήση `socket()` με μια Internet διεύθυνση και ένα αριθμό θύρας με τον οποίο θέλει να συνδεθεί ο πελάτης (μπορεί να βρίσκεται στο ίδιο μηχάνημα ή σε οποιοδήποτε άλλο). Συνεπώς τα πεδία της δομής στην οποία δείχνει ο `servaddr` πρέπει να είναι συμπληρωμένα πριν την κλήση. Εντελώς αδιάφανα το σύστημα παραχωρεί και έναν αριθμό θύρας στο μηχάνημα στο οποίο τρέχει ο πελάτης και μέσω της κλήσης αυτής αποκαθίσταται ένα νοητό κύκλωμα ανάμεσα στον πελάτη και τον εξυπηρετητή, μέσω του οποίου γίνεται η επικοινωνία. Είναι φανερό ότι αφού η κλήση αποκαθιστά ένα νοητό κύκλωμα, η χρήση της είναι δόκιμη μόνο από connection oriented πελάτες. Σε περίπτωση αποτυχίας η κλήση επιστρέφει αρνητικό ακέραιο.

Υπάρχει η περίπτωση να εκτελέσει μια κλήση `connect()` και ένας connectionless πελάτης. Αυτό που συμβαίνει τότε είναι ότι καταγράφεται η διεύθυνση που υπάρχει μέσα στη δομή `servaddr` και οποιεσδήποτε μελλοντικές εγγραφές στο socket αυτό θα προωθηθούν προς τη διεύθυνση αυτή, ενώ οποιαδήποτε μηνύματα φτάνουν από τη διεύθυνση αυτή θα παραδίδονται σε αυτή τη διεργασία - πελάτη. Αυτό έχει το πλεονέκτημα ότι η εφρμογή δεν χρειάζεται να ορίζει για κάθε `datagram` που στέλνει στον εξυπηρετητή τη διεύθυνση του. Ούτε χρειάζεται να ορίσει τη διεύθυνση από την οποία περιμένει απάντηση.

## Η κλήση

```
int listen(int sockfd, int backlog)
```

χρησιμοποιείται από connection-oriented εξυπηρετητές για να ενημερωθεί το σύστημα ότι ο εξυπηρετητής είναι έτοιμος να λάβει μηνύματα στο socket με περιγραφητή `sockfd`. Το όρισμα `backlog` ορίζει το μέγιστο αριθμό αιτήσεων για σύνδεση που μπορούν να μπούν σε αναμονή για τον εξυπηρετητή, ενώσω αυτός είναι απασχολημένος εκτελώντας την κλήση `accept()`. Και εδώ σε περίπτωση αποτυχίας επιστρέφεται αρνητικός αριθμός.

## Η κλήση

```
int accept(sockfd, struct sockaddr *peer, int *addrlen)
```

εκτελείται επίσης από connection-oriented εξυπηρετητές μόνο. Η κλήση αυτή παίρνει την πρώτη αίτηση σύνδεσης από την ουρά και επιστρέφει είτε άρνητικό αριθμό σε περίπτωση αποτυχίας, ή ένα νέο περιγραφητή socket με τα ίδια χαρακτηριστικά του `sockfd`. Αν δεν υπάρχουν αιτήσεις σύνδεσης στην ουρά τότε η κλήση μπλοκάρει τη διεργασία μεχρι να φτάσει κάποια αίτηση. Στην δομή `sockaddr` που δείχνει ο `peer` γράφονται από το σύστημα τα στοιχεία της σύνδεσης που έγινε δεκτή, δηλαδή η

Internet διεύθυνση και το port number του πελάτη. Στον ακέραιο που δείχνει ο addrlen επιστρέφεται το μήκος της δομής sockaddr που γράφτηκε στον peer. Στα πρωτόκολλα TCP/IP (Internet) ο αριθμός αυτός είναι 16.

Όπως είδαμε η κλήση επιστρέφει ένα νέο περιγραφητή socket υποθέτωντας ότι χρησιμοποιείται ένας παράλληλος εξυπηρετητής σύμφωνα με το σενάριο του Σχήματος 2.13. Να τονίσουμε εδώ ότι η κλήση accept() ορίζει και τα 5 πεδία της συσχέτισης δημιουργώντας μια νέα συσχέτιση την οποία χρησιμοποιεί η διεργασία παιδί του εξυπηρετητή για να παρέχει τη ζητούμενη υπηρεσία. Ο νέος αυτός περιγραφητής που δημιουργείται επιτρέπει στην διεργασία πατέρα να εκτελέσει μια νέα κλήση accept() χωρίς να χρειαστεί να δηλώσει και να χρησιμοποιήσει ένα νέο socket. Το μοντέλο του παράλληλου εξυπηρετητή είναι πολύ χρήσιμο στις περιπτώσεις connection-oriented επικοινωνίας ενώ το μοντέλο του συμπαγούς εξυπηρετητή χρησιμοποιείται περισσότερο με τα connectionless πρωτόκολλα. Ένα παράδειγμα με περισσότερες λεπτομέρεις για κάθε μια από τις δύο περιπτώσεις μπορεί να βρεθεί στο [TOMA91].

```

int sockfd,newsockfd;
if ((sockfd=socket(...))<0)
    err_sys("socket error");
if (bind(sockfd,...)<0)
    err_sys("bind error");
if (listen(sockfd,...)<0)
    err_sys("listen error");
for (;;) {
    newsockfd=accept(sockfd,...);
    if (newsockfd<0)
        err_sys("accept error");
    if (fork()==0) { /* child process */
        close(sockfd);
        doit(newsockfd);
        /* process request */
        exit(0);
    }
    close(newsockfd);
    /* father */
}

```

### Σχήμα 2.13 - Σενάριο χρήσης παράλληλων εξυπηρετητών

Οι κλήσεις

```

int send(int sockfd, char *buff, int nbytes, int nflags)
int sendto(int sockfd, char *buff, int nbytes, int
           nflags, struct sockaddr *to, int addrlen)
int recv(int sockfd, char *buff, int nbytes, int nflags)
int recvfrom(int sockfd, char *buff, int nbytes, int
             nflags, struct sockaddr *from, int *addrlen)

```

χρησιμοποιούνται για να μεταφερθούν τελικά οι πληροφορίες από τον πελάτη στον εξυπηρετητή και αντίστροφα. Τα πρώτα τρία ορίσματα σε κάθε μιά από αυτές τις κλήσεις είναι αντίστοιχα με τα ορίσματα των πρωτογενών κλήσεων read(), write() για διάβασμα/εγγραφή σε/από αρχείο αντίστοιχα. Τα nflags τίθενται ίσα με 0 ή χρησιμοποιούνται για κάποιες εξειδικευμένες λειτουργίες. Στην κλήση sendto() πρέπει να συμπληρωθούν τα πεδία της δομής sockaddr στην οποία δείχνει ο δείκτης το και το μέγεθος της δομής στην παράμετρο addrlen ώστε να σταλούν τα δεδομένα στον επιθυμητό παραλήπτη. Στην κλήση recvform() οι μεταβλητές στις οποίες δείχνουν οι

δείκτες from και addrlen συμπληρώνονται από το σύστημα όταν φθάσει κάποια αίτηση σύνδεσης. Είναι προφανές ότι οι κλήσεις recvfrom() και sendto() είναι δόκιμες όταν χρησιμοποιούνται datagrams ενώ οι send() και recv() όταν χρησιμοποιούμε reliable stream connection. Οι send() και recv() μπορούν να χρησιμοποιηθούν με datagrams αν ο πελάτης έχει εκτελέσει κλήση connect() όπως αναφέραμε περιγράφωντας την κλήση.

Όλες οι παραπάνω κλήσεις επιστρέφουν αρνητικό ακέραιο σε περίπτωση λάθους ή τον αριθμό των χαρακτήρων που διαβάστηκαν σε περίπτωση επιτυχίας.

### Η κλήση

```
int close(int sockfd)
```

κλείνει το socket που είχε δημιουργηθεί με την αντίστοιχη κλήση. Αν το πρωτόκολλο με το οποίο είχε δημιουργηθεί το socket ήταν TCP τότε πριν κλείσει η σύνδεση το σύστημα προσπαθεί να μεταφέρει ότι υπάρχει σε αναμονή στις ουρές του πελάτη και του εξυπηρετητή.

Για περισσότερη και λεπτομερέστερη ενημέρωση ανατρέξτε στο [STEV90] της βιβλιογραφίας. Στο βιβλίο αυτό περιέχεται επίσης πληθώρα επεξηγημένων με λεπτομέρεια παραδειγμάτων. Για δύο καλά αλλά όχι τόσο εκτενώς επεξηγημένα παραδείγματα αναφερθείτε στο [TOMA91].

Ακολουθούν τέσσερις εφαρμογές προγραμματισμού με sockets στο Unix. Αυτές είναι οι εξής:

- α. receive      (connection oriented)
- β. send        (connection oriented)
- γ. services     (connectionless)
- δ. getserv     (connectionless)

Τρόπος χρήσης:

O 'receive' server επικοινωνεί με έναν ή περισσότερους 'send' clients

O 'services' server επικοινωνεί με έναν ή περισσότερους 'getservice' clients

### Παράδειγμα 1°

Κάντε login σε κάποιο Unix μηχάνημα και πληκτρολογήστε την εντολή

```
receive&
```

Κάντε login σε ένα άλλο ή το ίδιο μηχάνημα και πληκτρολογήστε την εντολή

```
send daidalos
```

Οι γραμμές που γράφετε γίνονται echo από την 'receive' εφαρμογή.

### Παράδειγμα 2°

Κάντε login σε κάποιο Unix μηχάνημα και πληκτρολογήστε την εντολή

services&

Κάντε login σε ένα άλλο ή το ίδιο μηχάνημα και πληκτρολογήστε την εντολή  
getservice phgasos

Δίνοντας το είδος του service του οποίου θέλετε να μάθετε το port number, και αν αυτό υπάρχει στο directory /etc/services θα σας επιστραφεί το port number και το πρωτόκολλο που χρησιμοποιείται. (π.χ. δίνοντας ftp θα σας επιστραφεί 21 για port number και tcp για το πρωτόκολλο).

```
/* receive.c Server Program */

#include    <stdio.h>
#include    <sys/types.h>
#include    <sys/socket.h>
#include    <netdb.h>
#include    <netinet/in.h>
#include    <signal.h>
#include    <setjmp.h>

#define      FAIL      1
#define      SUCC      0
#define      PORT_NUMBER 2227

#define      MAX_CONN 5

int      sock_descr;           /*listen socket descriptor */
int      new_sockds;          /*connected socket descriptor*/
struct   sockaddr_in client_addr; /*peer socket address*/

jmp_buf   env;
static int warning =0;

main()
{
    void      error(),server();
    int       timeout();
    struct   sockaddr_in my_sock_addr;
    int       length = sizeof(struct sockaddr);

    if((sock_descr = socket(AF_INET, SOCK_STREAM, 0)) == -1)
        error("error opening stream socket in server", "");

    my_sock_addr.sin_family      = AF_INET;
    my_sock_addr.sin_addr.s_addr = INADDR_ANY;
    my_sock_addr.sin_port        = PORT_NUMBER;

    if(bind(sock_descr, (struct sockaddr*)&my_sock_addr, length) != 0)
        error("error binding to stream socket", "");

    (void)signal(SIGCLD,SIG_IGN);
    (void)signal(SIGALRM, timeout);

    if(listen(sock_descr, MAX_CONN) != 0)
        error("error in listen() call", "");

    switch(setjmp(env))
    {
    case 0:
        printf("\nIf someone will send me something in one minute
              live for another minute\n\n");
        alarm(60);

        for(;;)
        {
I'll
```

```

        if((new_sockds = accept(sock_descr, (struct
                                         sockaddr*)&client_addr,&length))==-1)
                                         error("accept","");
        else
        {
            alarm(0);
            alarm(60);
            printf("\n\n'receive' will live for another
                   minute because somebody keeps it
                   bussy!\n\n");
        }
        switch(fork())
        {
        case -1:
            error("fork: can't create new
process", "");
        case 0: /*client*/
            server();
            exit(SUCC);
        default:/*father*/
            if(close(new_sockds)==-1)
                error("father,close
socket","");
            putchar('\n');
        }
    }
    break;
}
case 1:
    printf("\n\nServer IS DOWN \n\n");
    exit(SUCC);
}

void server()
{
    char          *inet_ntoa();
    char      client_name[50], *p=client_name;
    struct     hostent      *client_info, *gethostbyaddr();
    char      buffer[256];
    int nread;
    char      message[12];

    (void)strcpy(message,"-- ALL DONE");

    if(close(sock_descr) == -1)
        error("child: close listen socket","");
    if((client_info = gethostbyaddr((char*)&client_addr.sin_addr.s
n_addr,
                                         sizeof(struct n_addr),client_addr.sin_family))==0L)
    {
        (void)strcpy(client_name, "Unknown Host");
        (void)strcat(client_name,"");
    }
    (void)strcat(client_name,inet_ntoa(client_addr.sin_addr));
}
else
{
    p = client_info->h_name;
}
printf("'send' process on host '%s' connected to 'receive'
\n",p);

while((nread = read(new_sockds,buffer,sizeof buffer))>0)
    printf("server received: '%s'\n", buffer);
if(nread < 0)
    error("socket read error","");
if(write(new_sockds, message, sizeof message) <0)
    error("server write error","");
if(close(new_sockds) == -1)

```

```

        error("close socket","");
}

void error(s1,s2)
char    *s1,*s2;
{
    extern      int           errno, sys_nerr;
    extern      char          *sys_errlist[];

    fprintf(stderr, "%s %s",s1,s2);
    if(errno > 0 && errno < sys_nerr)
        fprintf(stderr, " (%s)",sys_errlist[errno]);
    fprintf(stderr, "\n");
    exit(FAIL);
}

int timeout(sig)
{
    longjmp(env,1);
}

/* send.c Client Program */

#include   <stdio.h>
#include   <sys/types.h>
#include   <sys/socket.h>
#include   <netdb.h>
#include   <netinet/in.h>

#define     FAIL      1
#define     SUCC      0
#define     PROMPT   printf("type string to be sent (ctrl-D to exit)
:")
#define     PORT_NUMBER 2227

main(ac,av)
int      ac;
char    **av;
{
    int                  sock;
    struct      sockaddr_in      sock_addr;
    char        line[256];
    char        message[12];
    struct      hostent *host_struct, *gethostbyname();
    char        *server_name=av[1];
    char        *progr_name=av[0];
    void       error();

    if(ac !=2)
    {
        fprintf(stderr,"error -- usage: %s
                            <server_name>\n",progr_name);
        exit(FAIL);
    }

    if((host_struct = gethostbyname(server_name))==0L)
    {
        fprintf(stderr,"%s: unknown server
                            %s\n",progr_name,server_name);
        exit(FAIL);
    }

    if((sock=socket(AF_INET,SOCK_STREAM,0))==-1)
        error(progr_name,:error opening socket");

    bcopy(host_struct->h_addr,(char *)&sock.sin_addr.s_addr,
          host_struct->h_length);
    sock_addr.sin_port = PORT_NUMBER;
}

```

```

        sock_addr.sin_family = AF_INET;

        if(connect(sock,(struct sockaddr *)&sock_addr, sizeof
                   sock_addr)==-1)
        {
            if(close(sock) == -1)
                error("error closing client socket","");
            error(progr_name,: socket connection error");
        }

        while(PROMPT,gets(line) != NULL)
            if(write (sock, line , sizeof line)<0)
                error(progr_name,: error writing to remote
host");

        if(shutdown(sock,1) == -1)
            error(progr_name,:shutdown");

        if(read(sock, message, sizeof message) != sizeof message)
            error(progr_name,: read");

        printf ("\n%s\n",message);

        if(close(sock)==-1)
            error("error closing client socket","");
        exit(SUCC);
    }

void error(s1,s2)
char      *s1,*s2;
{
    extern int      errno, sys_nerr;
    extern char     *sys_errlist[];

    fprintf(stderr, "%s %s",s1,s2);
    if(errno > 0 && errno < sys_nerr)
        fprintf(stderr, " (%s)",sys_errlist[errno]);
    fprintf(stderr, "\n");
    exit(FAIL);
}

/* services.c Server Program */

#include    <stdio.h>
#include    <sys/types.h>
#include    <sys/socket.h>
#include    <netinet/in.h>
#include    <netdb.h>

#define      FAIL          1
#define      SUCC          0
#define      PORT_NUMBER   2229

main(ac,av)
int      ac;
char    **av;
{
    int                  sock;
    struct    sockaddr_in sock_addr;
    struct    sockaddr_in peer_addr;
    char           *progr_name = av[0];
    char           service_name[30];
    char           responce[256];
    struct    servent *serv_info;
    struct    servent *getservbyname();
    void           error();
    int            addr_len = sizeof(struct
sockaddr_in);
}

```

```

if((sock = socket(AF_INET, SOCK_DGRAM, 0)) == -1)
    error(progr_name, ":error opening socket");

sock_addr.sin_port          = PORT_NUMBER;
sock_addr.sin_family         = AF_INET;
sock_addr.sin_addr.s_addr   = INADDR_ANY;

if(bind(sock,(struct sockaddr*)&sock_addr, sizeof
       sock_addr)!=0)
    error(progr_name,": error binding to datagram socket");

setservent(1); /* keeps the "/etc/services" file open */

for(;;)
{
    if(recvfrom(sock,service_name,sizeof service_name,
                0,&peer_addr,&addr_len) < 0)
        error(progr_name,: receive error");

    if((serv_info=getservbyname(service_name, (char
                           *)0))==NULL)
        strcpy(responce, "Service Not Found");
    else
        sprintf(responce,
                  "%s service found at port %d with protocol
%s",
                  serv_info->s_name,
                  serv_info->s_port,
                  serv_info->s_proto);

    if(sendto(sock, responce, sizeof responce, 0,
&peer_addr,
           addr_len)==-1)
        error(progr_name,: send to client failed");
}
}

void error(s1,s2)
char *s1,*s2;
{
    extern     int    errno, sys_nerr;
    extern     char   *sys_errlist[];

fprintf(stderr, "%s %s",s1,s2);
if(errno > 0 && errno < sys_nerr)
    fprintf(stderr, " (%s)",sys_errlist[errno]);
fprintf(stderr, "\n");
exit(FAIL);
}

/* getserv.c Client Program */

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <signal.h>
#include <sys/errno.h>

#define FAIL    1
#define SUCC   0

#define PORT_NUMBER 2229
#define MAXTRIES 3
#define PROMPT printf("Service Sought(ctrl-d to exit):")

main(ac,av)

```

```

int ac;
char *av[ ];
{
    int             sock;
    struct sockaddr_in sock_addr;
    char            *progr_name = av[ 0 ];
    char            *server_name = av[ 1 ];
    char            service_name[ 30 ];
    char            return_message[ 256 ];
    struct hostent *host_struct;
    struct hostent *gethostbyname( );
    void            error();
    int             num_tries;
    int             alarm_handler();
    extern int      errno;

    if(ac !=2)
    {
        fprintf(stderr, "error -- usage: %s
<server_name>\n",
                progr_name);
        exit(FAIL);
    }

    if((host_struct = gethostbyname(server_name)) == NULL)
    {
        fprintf(stderr, "%s: unknown server %s\n",progr_name,
                server_name);
        exit(FAIL);
    }

    if((sock = socket(AF_INET, SOCK_DGRAM, 0))==-1)
        error(progr_name, ": error opening socket");

    bcopy(host_struct->h_addr, (char
*)&sock.sin_addr.s_addr,
          host_struct->h_length);
    sock.sin_port           = PORT_NUMBER;
    sock.sin_family         = AF_INET;

    (void)signal(SIGALRM, alarm_handler);

    while(PROMPT, gets(service_name)!=NULL)
    {
        num_tries = 0;
        for(;;)
        {
            int nread;
            if(sendto(sock,service_name,sizeof
service_name,0,
                      (struct sockaddr
*)&sock_addr,
                      sizeof(sock_addr)==-1)
               error(progr_name,"error sending
datagram"));

            (void)alarm(5);
            if(   (nread = recv(sock,return_message,
                                sizeof return_message,0))
                <=0 )
            {
                if(nread < 0 && errno !=EINTR)
                    error(progr_name,": receive
failed");
                if(num_tries++<MAXTRIES)
                    continue;
                else
                {
                    fprintf(stderr,

```

```

                "timeout: no response
from %s\n",
                           server_name);
exit(FAIL);
}
else
break;
}
(void)alarm(0);
printf("%s\n",return_message);
}
putchar('\n');
if (close(sock)==-1)
error(progr_name, ": error closing socket");
exit(SUCC);
}

int alarm_handler()
{
    (void)signal(SIGALRM, alarm_handler);
}

void error(s1,s2)
char    *s1,*s2;
{
    extern int      errno, sys_nerr;
    extern char    *sys_errlist[];

fprintf(stderr, "%s %s",s1,s2);
if(errno > 0 && errno < sys_nerr)
    fprintf(stderr, " (%s)",sys_errlist[errno]);
fprintf(stderr, "\n");
exit(FAIL);
}

```

## 2.7. Ασκήσεις

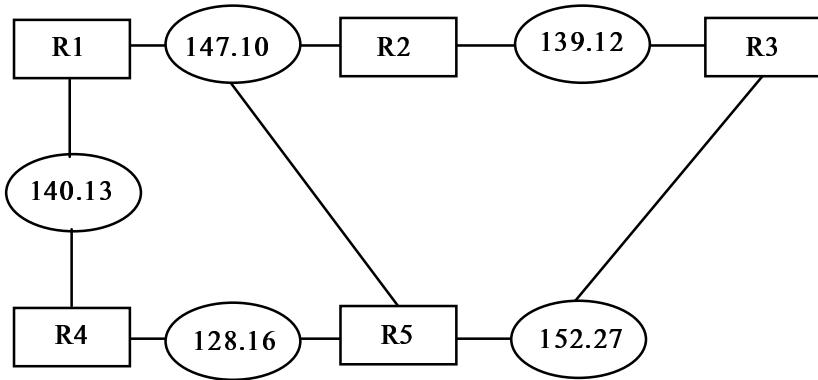
- [1]. Πόσα ακριβώς δίκτυα τάξης A, B, C μπορούν να υπάρξουν; Σε κάθε μία από τις τάξεις αυτές, πόσοι υπολογιστές με IP διευθύνσεις μπορούν να χρησιμοποιηθούν; Πώς μπορούμε γρήγορα να αναγνωρίσουμε αν μία IP διεύθυνση ανήκει σε δίκτυο τάξης A, B και C;
- [2]. Ποιο είναι το ελάχιστο MTU (Maximum Transfer Unit) που απαιτείται από ένα δίκτυο προκειμένου να μεταφερθεί ένα IP datagram με 1 byte πληροφορίας;
- [3]. Ποια είναι τα πλεονεκτήματα και τα μειονεκτήματα της επανασύνδεσης (μετά από κατακερματισμό) των κομματιών ενός IP datagram στον τελικό προορισμό του και όχι αφού διασχίσει κάποιο δίκτυο;
- [4]. Περιγράψτε κάποια εφαρμογή, όπου η στοίβα πρωτοκόλλων TCP/IP περιλαμβάνει και επίπεδο παρουσίασης.

- [5]. Ποια είναι τα πλεονεκτήματα και τα μειονεκτήματα της χρήσης προπροσδιορισμένων port-numbers στο UDP και το TCP πρωτόκολλο για κάποιο αριθμό υπηρεσιών;
- [6]. Υποθέστε ότι δεν γνωρίζετε την IP-διεύθυνση ενός υπολογιστή του τοπικού σας δικτύου, στον οποίο τρέχει ο UDP echo server απαντώντας στη πόρτα 7. Υπάρχει κάποια IP-διεύθυνση που μπορείτε να χρησιμοποιήσετε για έχετε πρόσβαση σ' αυτό;
- [7]. Περιγράψτε εφαρμογές του χρήστη που απαιτούν πρόσβαση στο IP πρωτόκολλο.
- [8]. Είναι πολύ δύσκολο για μια κεντρική διαχειριστική αρχή να μοιράζει τις IP διευθύνσεις με τέτοιο ρυθμό, ώστε να καλύπτει τις παγκόσμιες ανάγκες. Περιγράψτε κάποια μέθοδο διαμερισμού της εργασίας από την κεντρική αρχή σε περισσότερες ομάδες, χωρίς να διακινδυνεύεται η μοναδικότητα των IP διευθύνσεων.
- [9]. Αναμένεται τα τοπικά δίκτυα υψηλών ταχυτήτων (high-speed LANs) να έχουν μικρότερο ή μεγαλύτερο MTU (Maximum Transfer Unit) από τα αργά δίκτυα ευρείας περιοχής (long-haul networks);
- [10]. Είναι δυνατό να βρεθεί σ' ένα TCP/IP δίκτυο IP datagram με διεύθυνση προορισμού κάποιο δρομολογητή; Αν αυτό είναι δυνατό, ποια θα ήταν η λογική του;
- [11]. Δοκιμάστε να στείλετε σε ένα δρομολογητή μηνύματα με τέτοιο ρυθμό, ώστε να αναγκαστεί να στείλει ένα ICMP source quence μήνυμα.
- [12]. Θα πρέπει κάποιο ICMP μήνυμα να περιέχει και τη χρονική στιγμή που συνέβη το οποιοδήποτε πρόβλημα; Για ποιο λόγο;
- [13]. Μήπως θα έπρεπε η έννοια των πολλαπλών προορισμών σε ένα μηχάνημα που υλοποιείται στα πρωτόκολλα UDP, TCP με τους αριθμούς θυρών να είχε συμπεριληφθεί στο IP πρωτόκολλο; Για ποιο λόγο;
- [14]. Ποιο πλεονέκτημα έχει το γεγονός ότι η αναγνώριση διεργασιών σε ένα μηχάνημα που θέλουν να επικοινωνήσουν με διεργασίες σε άλλα μηχανήματα

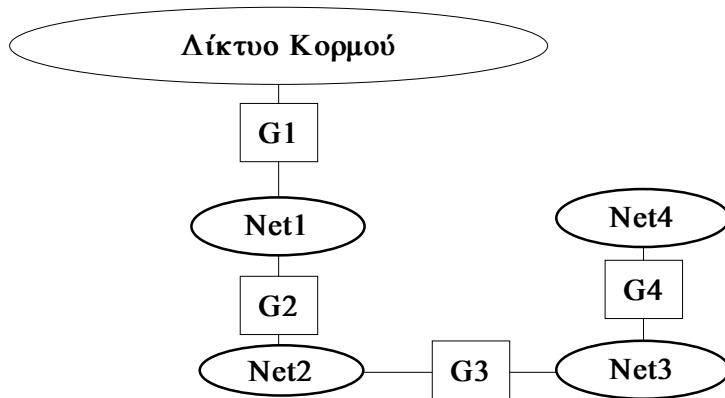
γίνεται μέσω αριθμών θυρών (port numbers) και όχι μέσω των αναγνωριστικών τους (process identifier, PID);

- [15]. Προσπαθήστε να τρέξετε τα παραδείγματα send-receive και services-getservice στο τοπικό υπολογιστικό σας σύστημα.
- [16]. Βρείτε και διαβάστε για το AT&T System V Transport Library Interface (TLI). Ποια είναι η κυριότερη διαφορά από το socket interface;
- [17]. Κάθε λειτουργικό σύστημα περιορίζει τον αριθμό των sockets που κάποιο συγκεκριμένο πρόγραμμα μπορεί να χρησιμοποιήσει. Βρείτε ποιος είναι ο αριθμός αυτός για το λειτουργικό σύστημα με το οποίο δουλεύετε.
- [18]. Υπάρχει κάποια φανερή διαφορά (στην πράξη) μεταξύ των εφαρμογών telnet και rlogin.
- [19]. Ποιο είναι το σημαντικότερο πλεονέκτημα της χρήσης δύο TCP συνδέσεων (control και data) από το FTP;
- [20]. Υλοποιήστε κάποια client διεργασία η οποία θα εξετάζει και θα τυπώνει όλες τις διαθέσιμες πληροφορίες για κάθε μηχάνημα στο δίκτυο με τη συνάρτηση gethostbyname.
- [21]. Με τη βοήθεια του προγράμματος ping βρείτε το μέγεθος του μεγαλύτερου datagram που το δίκτυο σας μπορεί να στείλει ή να δεχτεί. Τι σχέση έχει το μέγεθος αυτό με το MTU (Maximum Transfer Unit) του δικτύου σας; Τι άλλα συμπεράσματα μπορείτε να βγάλετε με τη χρήση του ping (π.χ. για την απόδοση);
- [22]. Εξηγήστε το σκοπό ύπαρξης του μοντέλου αναφοράς πρωτοκόλλων OSI, και πολύ σύντομα παρουσιάστε τις λειτουργίες κάθε επιπέδου.
- [23]. Πρωτοκόλλα όπως τα SNMP, TELNET, FTP, TFTP, SMTP, NFS (Network File System) χρησιμοποιούν υπηρεσίες επιπέδου μεταφοράς με σύνδεση ή χωρίς σύνδεση; Εξηγήστε τους πιθανούς λόγους για αυτή τη διαφοροποίηση.
- [24]. Εξετάστε το include file για τα sockets (συνήθως /usr/include/sys/sockets.h). Τι μορφές sockets επιτρέπονται; Ποιες μορφές χρησιμοποιούνται σε ένα TCP/IP περιβάλλον;

- [25]. Αν υποθέσουμε ότι τα διασυνδεδεμένα υπο-δίκτυα του σχήματος τρέχουν κάποιο distance vector αλγόριθμο (π.χ. RIP), όπου το μέτρος της απόστασης είναι hops, ποιος θα είναι ο πίνακας δρομολόγησης κάθε δρομολογητή στη μόνιμη κατάσταση.



- [26]. Υλοποιήσεις του πρωτοκόλλου EGP χρησιμοποιούν ένα μηχανισμό, ο οποίος αναγκάζει το πρωτόκολλο να καθυστερήσει την αποδοχή ενός acquisition request μηνύματος από κάποιο γείτονα για ένα καθορισμένο χρονικό διάστημα μετά τη λήψη ενός μηνύματος cease request από τον ίδιο γείτονα. διαβάστε προσεκτικά το σχετικό RFC για να βρείτε το λόγο.
- [27]. Για το δίκτυο του σχήματος ποιες μηχανές πρέπει να τρέχουν το EGP πρωτόκολλο;



- [28]. Σε ποιες περιπτώσεις πιστεύετε ότι το μέτρο δρομολόγησης πρέπει να είναι ο αριθμός των ενδιάμεσων κόμβων (hops) και σε ποιες περιπτώσεις η μέση καθυστέρηση (mean transit delay);

## 2.8. Βιβλιογραφία

- [BERT87] Bertsekas D., Gallager R., *Data Networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- [COME91] Comer E.D., *Internetworking with TCP/IP Volume I; Principles, Protocols, and Architectures*, 2nd Edition, Prentice Hall, N.J., 1991.
- [COME91] Comer E.D. and D.L.Stevens, *Internetworking with TCP/IP Volume II; Design, Implementation, and Internals*, Prentice Hall, N.J., 1991.
- [COME93] Comer E.D. and D.L.Stevens, *Internetworking with TCP/IP Volume III; Client-Server Programming and Applications*, Prentice Hall, N.J., 1993.
- [HALS92] Halsall F., *Data Communications, Computer Networks and Open Systems*, 3rd Edition, Addison-Wesley, 1992.
- [OPEN90] Rose M.T., *The Open Book, A Practical Perspective on OSI*, Prentice Hall, 1990.
- [ΣΤΑΣ89] Στασινόπουλος Γ., *Ψηφιακά Συστήματα Επικοινωνιών*, Ε.Μ.Πολυτεχνείο, Τμήμα Ηλεκτρολόγων, Αθήνα 1989.
- [ΠΟΜΠΙ90] Α.Σ.Πομπόρτσης, *Τοπικά Δίκτυα Υπολογιστών*, Θεσσαλονική 1990.
- [ROSE91] Marshall T. Rose, *The Simple Book: An Introduction to Management of TCP/IP - based Internets*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [ROSE92] Rose M. T., *The Little Black Book, Mail Bonding with OSI Directory Services*, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.
- [STEV90] W.R.Stevens, *Unix Network Programming*, Prentice Hall, 1990.
- [STEV92] W.R.Stevens, *Advanced Programming in the Unix Environment*, Addison-Wesley, 1992.
- [TANE91] Tanenbaum A.S., *Δίκτυα Υπολογιστών*, Δεύτερη Έκδοση, Prentice Hall, για την Ελληνική Εκδοση Παπασωτηρίου 1991.
- [TOMA91] M.Tomassini, Programming with sockets, *The C Users Journal*, September 1991, pp. 39-56.

## Κεφάλαιο 3

### 3. Το Μοντέλο Αναφοράς ISO/OSI

#### Περιεχόμενα του Κεφαλαίου 3

- 3.0. Εισαγωγή
- 3.1. Το μοντέλο αναφοράς πρωτοκόλλων OSI του ISO
- 3.2. Εισαγωγή στα επίπεδα OSI
- 3.3. Περιγραφή των επιπέδων
  - 3.3.1. Επίπεδο Εφαρμογής
  - 3.3.2. Επίπεδο Παρουσίασης
  - 3.3.3. Επίπεδο Συνόδου
  - 3.3.4. Επίπεδο Μεταφοράς
  - 3.3.5. Επίπεδο Δικτύου
  - 3.3.6. Επίπεδο Σύνδεσης Λεδομένων
  - 3.3.7. Φυσικό Επίπεδο
- 3.4. Διαφορές με τα πρωτόκολλα TCP/IP
- 3.5. Ειδικά στοιχεία υπηρεσίας στο επίπεδο εφαρμογής: FTAM, MHS
- 3.6. Περιβάλλον ανάπτυξης εφαρμογών ISO/OSI και TCP/IP, ISODE
- 3.7. Ασκήσεις
- 3.8. Βιβλιογραφία

#### 3.0. Εισαγωγή

Στο κεφάλαιο αυτό θα ασχοληθούμε με τα **πρωτόκολλα Διασύνδεσης Ανοικτών Συστημάτων Open Systems Interconnection - OSI** του **Διεθνούς Οργανισμού Τυποποίησης International Standards Organization - ISO**. Θα προσπαθήσουμε, όσο είναι αυτό δυνατό, να αποφύγουμε την επανάληψη θεμάτων, που με τον ένα ή τον άλλο τρόπο εξετάσαμε μαζί με τα πρωτόκολλα TCP/IP.

Το πρόβλημα της διασύνδεσης υπολογιστικών συστημάτων με την βοήθεια τοπικών δικτύων υπολογιστών (Local Area Networks - LANs) ή δικτύων ευρείας περιοχής (Wide Area Networks - WANs) είναι ιδιαίτερα πολύπλοκο και σίγουρα δεν μπορεί να αντιμετωπιστεί με ένα μοναδικό τρόπο. Μια τεχνική που επιλέχτηκε από πολλές εταιρείες ήταν η ανάπτυξη ιδιαίτερου υλικού/λογισμικού για κάθε περίπτωση. Το κόστος της λύσης αυτής και η αδυναμία, σε πολλές περιπτώσεις, επικοινωνίας μεταξύ ετερογενών υπολογιστικών συστημάτων διέγραψε την ανάγκη εισαγωγής προτύπων στο πρόβλημα της διασύνδεσης υπολογιστών.

Ταυτόχρονα με την ανάγκη ύπαρξης προτύπων, έγινε φανερή η ανάγκη χρησιμοποίησης της αρχιτεκτονικής της **διαστρωμάτωσης (layering)**. Το πρόβλημα της επικοινωνίας διαμέσου δικτύων είναι τόσο πολύπλοκο, ώστε να είναι δύσκολο να αντιμετωπιστεί σαν ένα ενιαίο θέμα. Η διαδικασία της διαστρωμάτωσης είναι μια βασική τεχνική στην λύση προβλημάτων επικοινωνίας διαμέσου δικτύων. Κατανέμει τη δυσκολία της μεταδοσης πληροφορίας σε περισσότερες από μία μονάδες υλικού/λογισμικού που ονομάζονται επίπεδα. Τα υψηλότερα επίπεδα μπορούν να χειρίζονται πιο πολύπλοκες μορφές πληροφορίας, ενώ τα χαμηλότερα απλούστερες. Κάθε επίπεδο καθώς ανεβαίνουμε προς τα πάνω, χρησιμοποιεί τις υπηρεσίες του αμέσως κατώτερου επιπέδου και προσφέρει υπηρεσίες στο αμέσως ανώτερο.

Με βάση τις δύο παραπάνω διαπιστώσεις, το 1977 ο Διεθνής Οργανισμός Τυποποίησης προχώρησε στη δημιουργία κατάλληλων προτύπων για την υλοποίηση δικτύων διασύνδεσης υπολογιστών. Το αποτέλεσμα της προσπάθειας αυτής ήταν η ανάπτυξη του Μοντέλου Αναφοράς Πρωτοκόλλων OSI, το οποίο είναι το πρότυπο το οποίο χρησιμοποιείται σήμερα ευρέως από την επιστημονική κοινότητα. Το μοντέλο OSI περιγράφει την ιδέα της διαστρωμάτωσης, της υπηρεσίας και του πρωτόκολλου, με την βοήθεια των οποίων επιδιώκεται η σύνδεση μέσω επικοινωνιακού δικτύου ετερογενών υπολογιστικών συστημάτων.

### 3.1. Το μοντέλο αναφοράς πρωτοκόλλων OSI του ISO

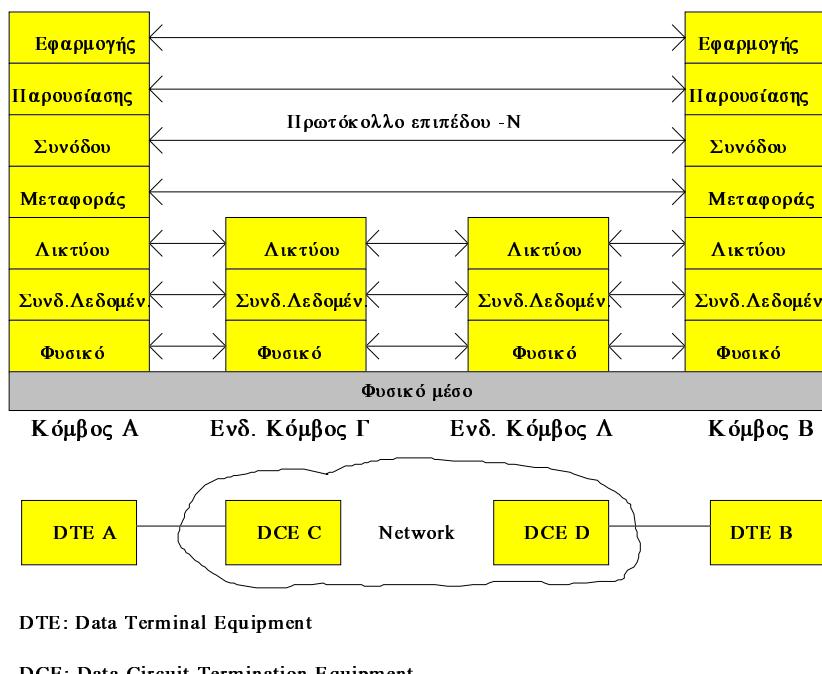
Ο σκοπός του Πρότυπου αυτού, είναι να προσφέρει μια κοινή βάση για το συντονισμό της ανάπτυξης προτύπων σχετικών με τη διασύνδεση συστημάτων. Ο όρος Διασύνδεση Ανοικτών Συστημάτων (OSI) προσδιορίζει πρότυπα που αφορούν την ανταλλαγή πληροφορίας μεταξύ συστημάτων. Τα συστήματα αυτά είναι "ανοικτά" το ένα στο άλλο, εξαιτίας της από κοινού εφαρμογής κατάλληλων προτύπων. Δηλ. το γεγονός ότι κάποιο σύστημα είναι ανοικτό δε συνεπάγεται κάποια συγκεκριμένη υλοποίηση του συστήματος, τεχνολογία ή μέσο διασύνδεσης, αλλά αναφέρεται στην από κοινού αναγνώριση και υποστήριξη κατάλληλων προτύπων.

Ένας άλλος στόχος του Πρότυπου OSI αυτού είναι να αναγνωρίσει περιοχές για την ανάπτυξη ή βελτίωση προτύπων, και να προσφέρει μια κοινή βάση για τη διατήρηση της ενότητας μεταξύ όλων των προτύπων αυτών. Το Πρότυπο αυτό δεν στοχεύει να λειτουργήσει σαν προδιαγραφή για υλοποιήσεις, είτε να αποτελέσει μια βάση για την εκτίμηση της συμμόρφωσης πραγματικών υλοποιήσεων. Ούτε στοχεύει στο να προσφέρει ένα ικανοποιητικό επίπεδο λεπτομέρειας ώστε να ορίσει με ακριβεία τις υπηρεσίες και τα πρωτόκολλα της αρχιτεκτονικής διασύνδεσης. Περισσότερο παρέχει ένα νοητό και λειτουργικό πλαίσιο, το οποίο επιτρέπει σε ομάδες ειδικών σε διεθνές επίπεδο να εργαστούν παραγωγικά και ανεξάρτητα για την ανάπτυξη προτύπων για κάθε επίπεδο του Μοντέλου Αναφοράς OSI.

Με βάση την αρχιτεκτονική της διαστρωμάτωσης, το πρότυπο OSI κατανέμει τη δυσκολία επικοινωνίας σε 7 (επτά) επίπεδα αρχίζοντας από το φυσικό επίπεδο, όπου έχουμε μεταφορά δυαδικών ψηφίων μεταξύ δύο υπολογιστών και φθάνοντας στο επίπεδο εφαρμογής, το οποίο είναι το τελευταίο σύνορο μεταξύ του ανοικτού συστήματος και της εφαρμογής του χρήστη. Τα 7 επίπεδα φαίνονται στο Σχήμα 3.1. Η αρχιτεκτονική της διαστρώματωσης μπορεί να επεκταθεί και μέσα στα επίπεδα με τη δημιουργία υποεπιπέδων (π.χ. το υποεπίπεδο ελέγχου προσπέλασης του μέσου (MAC) και το επίπεδο ελέγχου λογικών συνδέσεων στο επίπεδο σύνδεσης δεδομένων). Παρακάτω θα εξετάσουμε τα 7 επίπεδα του προτύπου OSI αναλυτικότερα, αλλά προς στιγμή ας δούμε εισαγωγικά μερικές ακόμα βασικές ιδέες του μοντέλου OSI.

Με βάση το μοντέλο OSI, η επικοινωνία μεταξύ δύο υπολογιστικών συστημάτων επιτυγχάνεται μέσω της επικοινωνίας μεταξύ ομοτίμων επιπέδων. Η επικοινωνία αυτή είναι πραγματική μόνο για το φυσικό επίπεδο, ενώ είναι νοητή για τα υπόλοιπα 6 υψηλότερα επίπεδα. Για να επιτευχθεί η νοητή αυτή επικοινωνία, και να επικοινωνήσουν δύο ομότιμα επίπεδα, κάθε ένα από αυτά χρησιμοποιεί κάθε φορά τις υπηρεσίες του αμέσως χαμηλότερου επιπέδου του. Η επικοινωνία μεταξύ ομοτίμων επιπέδων καθορίζεται πλήρως από το πρωτόκολλο του αντίστοιχου επιπέδου.

Η πληροφορία, κατά την επικοινωνία μεταξύ ομοτίμων επιπέδων, μεταφέρεται οριζόντια μέσα σε **μονάδες δεδομένων πρωτόκολλου (Protocol Data Units, PDUs)**. Η ονομασία και το περιεχόμενο κάθε PDU, καθώς και κάθε παραμέτρου στα διάφορα πεδία του καθορίζεται πλήρως. Επίσης καθορίζονται και οι διαδικασίες που πρέπει να εκτελεί κάθε επίπεδο για την αποστολή και κατά την λήψη ενός PDU. Αυτό σημαίνει ότι τόσο η σημασία (semantics) όσο και η σύνταξη (syntax) ενός PDU είναι καλά καθορισμένα. Οι μονάδες αυτές πληροφορίας, βέβαια, μεταφέρονται, στην πραγματικότητα, από PDUs του αμέσως χαμηλότερου επιπέδου προκειμένου τελικά να φθάσουν στο ομότιμο επίπεδο.



### Σχήμα 3.1 - Τα 7 επίπεδα του μοντέλου OSI

Κάθε επίπεδο OSI συναλλάσσεται μονάχα με τα γειτονικά του επίπεδα (το αμέσως ψηλότερο και το αμέσως χαμηλότερο), προκειμένου να επικοινωνήσει με το ομότιμό του επίπεδο. Δηλ. χρησιμοποιεί τις υπηρεσίες του αμέσως χαμηλότερου του επιπέδου προκειμένου να εκτελέσει τις λειτουργίες, που οφείλει, ενώ προσφέρει τις υπηρεσίες του στο αμέσως υψηλότερό του επίπεδο, προκειμένου και αυτό να εκτελέσει τις λειτουργίες του. Οι υπηρεσίες διατίθενται από κάθε επίπεδο στο αμέσως υψηλότερό του σε καλά καθορισμένα σημεία, τα **σημεία πρόσβασης υπηρεσίας (Service Access Points, SAPs)**. Για να είναι δυνατή η ταυτόχρονη επικοινωνία μεταξύ περισσοτέρων του ενός ζευγαριών διεργασιών, σε κάθε επίπεδο δημιουργούνται συνδέσεις, σε κάθε μία από τις οποίες αντιστοιχεί ξεχωριστό ζευγάρι από SAPs. Για την προσφορά και ζήτηση υπηρεσίων ορίζεται μονάχα το σημαντικό μέρος, ενώ το συντακτικό αφήνεται ελεύθερο. Με τον τρόπο αυτό επιτυγχάνεται η επικοινωνία μεταξύ ανοικτών συστημάτων (π.χ. συστημάτων με διαφορετικό λειτουργικό σύστημα).

Οι λειτουργίες κάθε επιπέδου υλοποιούνται από κατάλληλες οντότητες (entities). Μια οντότητα κάποιου επιπέδου υλοποιεί λειτουργίες του επιπέδου αυτού (π.χ. λογισμικό υλοποίησης του πρωτοκόλλου CMIS), καθώς και το πρωτόκολλο για την επικοινωνία με ομότιμες οντότητες σε άλλα συστήματα (π.χ. λογισμικό υλοποίησης του πρωτοκόλλου CMIP).

Μερικές βασικές λειτουργίες που μπορεί να συναντήσει κανείς στα επίπεδα του μοντέλου OSI είναι:

- a. **(Encapsulation).** Η τεχνική του encapsulation χρησιμοποιείται ευρέως στο μοντέλο OSI, όπως έγινε φανερό παραπάνω, από την στιγμή που τα δεδομένα που ανταλλάσσονται δύο μηχανήματα δεν είναι τίποτα άλλο από πακέτα το ένα μέσα στο άλλο. Με αρχή το επίπεδο εφαρμογής, κάθε PDU μαζί με την αντίστοιχή του επικεφαλίδα ενθηλακώνεται στο PDU του αμέσως χαμηλότερου επιπέδου.
- β. **Τμηματοποίηση (Segmentation).** Η τμηματοποίηση αναφέρεται στην λειτουργία κάποιου επιπέδου να τμηματοποίησει το PDU του υψηλότερου επιπέδου, προκειμένου να ικανοποιήσει κάποιες απαιτήσεις. Βέβαια, το ομότιμο επίπεδο στον προορισμό θα πρέπει να επανασυνδέσει το PDU προκειμένου να το παραδόσει στο υψηλότερο επίπεδο. Μια περίπτωση της λειτουργίας του segmentation συναντήσαμε στο πρωτόκολλο IP.
- γ. **Εγκατάσταση σύνδεσης (Connection Establishment).** Η ανταλλαγή δεδομένων μεταξύ δύο ομοτίμων οντοτήτων μπορεί να επιτευχθεί με δύο τρόπους. Ο πρώτος τρόπος απαιτεί μια εκ των προτέρων λογική σύνδεση η οποία δημιουργεί την έννοια των νοητών κυκλωμάτων. Ο δεύτερος τρόπος δεν απαιτεί κάποια δημιουργία κυκλώματος. Σημειώνουμε ότι, η ύπαρξη σύνδεσης διευκολύνει τον έλεγχο ροής.
- δ. **Έλεγχος ροής (Flow Control).** Ο έλεγχος ροής αναφέρεται στους μηχανισμούς περιορισμού της ροής δεδομένων από ένα πρωτόκολλο προς το ομότιμό του. Μια πιθανή υλοποίηση του μηχανισμού αυτού εξετάσαμε με το πρωτόκολλο TCP.
- ε. **Έλεγχος σφαλμάτων (Error Control).** Ο έλεγχος σφαλμάτων αναφέρεται στους μηχανισμούς ανίχνευσης και διόρθωσης λαθών που συμβαίνουν κατά την μετάδοση PDUs μεταξύ ομοτίμων πρωτοκόλλων.
- στ. **Πολυπλεξία (Multiplexing).** Ο όρος πολυπλεξία αναφέρεται στην πολύπλεξη πολλών συνδέσεων ενός επιπέδου σε μία σύνδεση του αμέσως χαμηλότερου επιπέδου, κυρίως για λόγους οικονομίας.

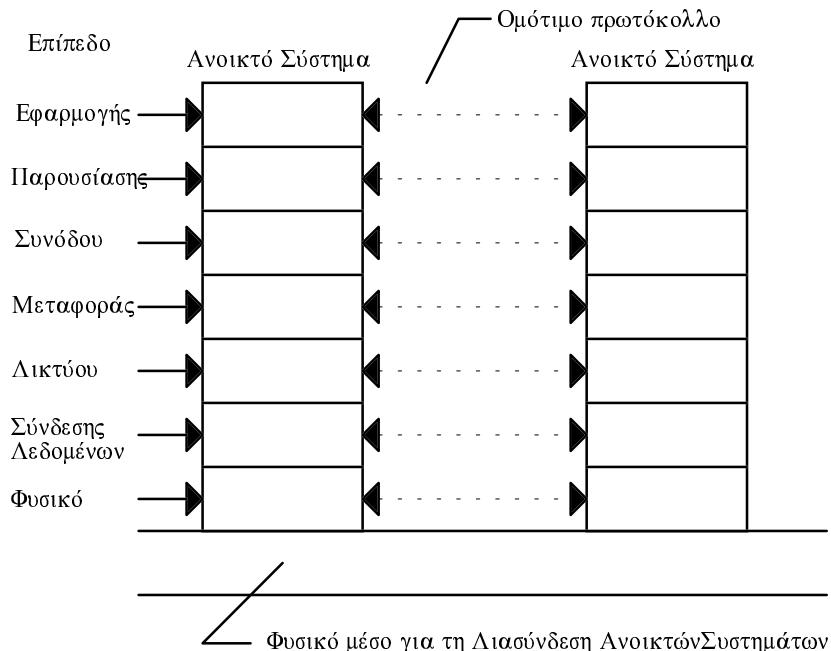
Στις επόμενες παραγράφους, θα εξετάσουμε τα πρότυπα ISO/OSI για τα 7 επίπεδα του μοντέλου.

### 3.2. Εισαγωγή στα επίπεδα OSI

Το Μοντέλο Αναφοράς αποτελείται από επτά επίπεδα:

- α) το επίπεδο Εφαρμογής (Application Layer - 7),
- β) το επίπεδο Παρουσίασης (Presentation Layer - 6),
- γ) το επίπεδο Συνόδου (Session Layer - 5),
- δ) το επίπεδο Μεταφοράς (Transport Layer - 4),
- ε) το επίπεδο Δικτύου (Network Layer - 3),
- στ) το επίπεδο Σύνδεσης Λεδομένων (Data Link Layer - 2), και
- ζ) το Φυσικό επίπεδο (Physical Layer - 1).

Τα επίπεδα αυτά παρουσιάζονται στο σχήμα 3.2. Το υψηλότερο από αυτά είναι το επίπεδο Εφαρμογής, το οποίο αποτελείται από οντότητες-εφαρμογής, που συνεργάζονται στο περιβάλλον OSI. Τα χαμηλότερα επίπεδα παρέχουν τις υπηρεσίες μέσα από τις οποίες οι οντότητες-εφαρμογής συνεργάζονται.

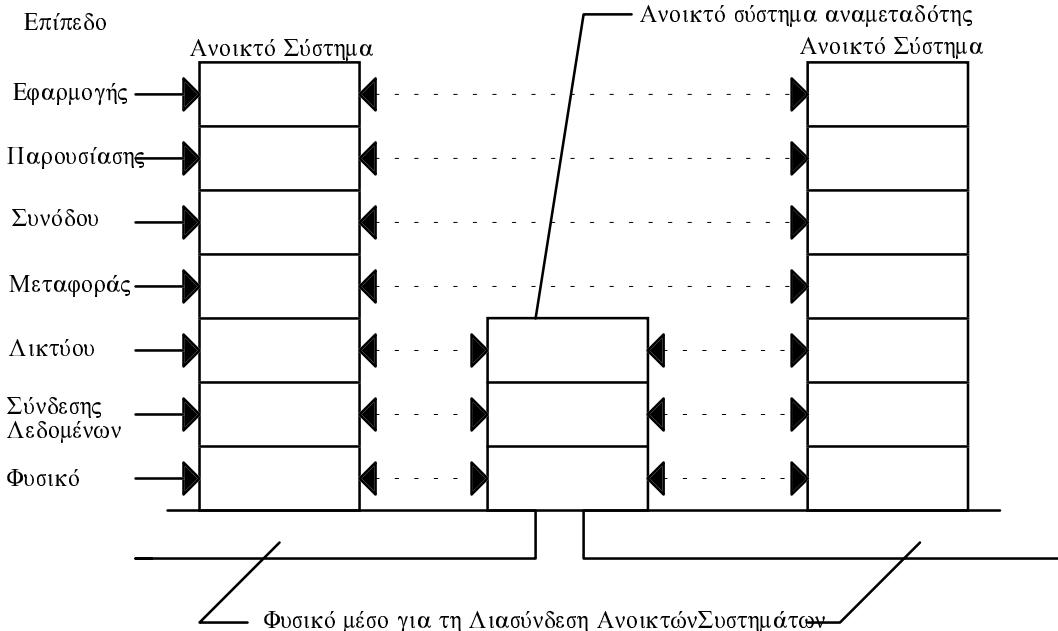


**Σχήμα 3.2 - Μοντέλο αναφοράς επτά επιπέδων και ομότιμα πρωτόκολλα**

Τα επίπεδα 1 έως 6, μαζί με το φυσικό μέσο για τη Διασύνδεση Ανοικτών Συστημάτων παρέχουν ένα σταδιακό εμπλουτισμό επικοινωνιακών υπηρεσιών. Τα σύνορα μεταξύ δύο επιπέδων αναγνωρίζουν ένα στάδιο στον εμπλουτισμό αυτό υπηρεσιών στο οποίο ορίζεται ένα πρότυπο για μια υπηρεσία OSI (π.χ. το πρότυπο Common Management Information Service - CMIS για το επίπεδο 7), ενώ η λειτουργία των επιπέδων καθορίζεται από πρότυπα πρωτοκόλλων OSI (π.χ. το πρότυπο Common Management Information Protocol - CMIP για το επίπεδο 7).

Τα ανοικτά συστήματα δεν αποτελούν πάντοτε την αρχική πηγή ή τον τελικό προορισμό δεδομένων. Όταν το φυσικό μέσο για τη Διασύνδεση Ανοικτών Συστημάτων δεν συνδέει

τα ανοικτά συστήματα άμεσα, τότε μερικά ανοικτά συστήματα λειτουργούν σαν ανοικτά συστήματα αναμεταδότες, αναμεταδίδοντας απλά δεδομένα σε άλλα ανοικτά συστήματα. Οι λειτουργίες και τα πρωτόκολλα τα οποία θα υποστηρίζουν την προώθηση δεδομένων παρέχονται σε χαμηλότερα επίπεδα. Αυτό φαίνεται στο σχήμα 3.3.



**Σχήμα 3.3 - Επικοινωνία που χρησιμοποιεί ανοικτό σύστημα αναμεταδότη**

### 3.3. Περιγραφή των επιπέδων

Οι περιγραφές που ακολουθούν από μόνες τους δεν παρέχουν έναν πλήρη ορισμό των υπηρεσιών και των πρωτοκόλλων για κάθε επίπεδο. Αυτά είναι θέματα ξεχωριστών προτύπων.

#### 3.3.1 Επίπεδο Εφαρμογής

Σαν το υψηλότερο επίπεδο στο Μοντέλο Αναφοράς για τη Διασύνδεση Ανοικτών Συστημάτων, το Επίπεδο Εφαρμογής παρέχει τα μέσα στις διαδικασίες εφαρμογής προκειμένου αυτές να έχουν πρόσβαση στο περιβάλλον OSI. Ο σκοπός του Επιπέδου Εφαρμογής είναι να εξυπηρετήσει τις διαδικασίες-εφαρμογής οι οποίες θέλουν να επικοινωνήσουν, προκειμένου να ανταλλάξουν χρήσιμη πληροφορία. Οι διαδικασίες-εφαρμογής ανταλλάσσουν πληροφορία με την βοήθεια οντότητων-εφαρμογής, πρωτοκόλλων-εφαρμογής, και υπηρεσιών-παρουσίασης.

Καθώς το Επίπεδο Εφαρμογής είναι το μοναδικό επίπεδο στο Μοντέλο Αναφοράς το οποίο άμεσα παρέχει υπηρεσίες στις διαδικασίες-εφαρμογής, παρέχει όλες τις υπηρεσίες OSI κατά τρόπο ώστε να είναι δυνατόν να χρησιμοποιηθούν άμεσα από τις διαδικασίες-εφαρμογής (χωρίς περαιτέρω εμπλουτισμό).

Μια οντότητα-εφαρμογής περιέχει ένα στοιχείο-χρήστη και ένα σύνολο από στοιχεία-υπηρεσίας-εφαρμογής (βλ. σχήμα 3.4). Το στοιχείο-χρήστη παριστάνει εκείνο το κομμάτι των διαδικασιών-εφαρμογής το οποίο χρησιμοποιεί τα απαραίτητα στοιχεία-υπηρεσίας-εφαρμογής προκειμένου να εκτελέσει τους επικοινωνιακούς στόχους της συγκεκριμένης διαδικασίας-εφαρμογής. Στοιχεία-υπηρεσίας-εφαρμογής είναι δυνατόν να χρησιμοποιούν το ένα το άλλο και/ή υπηρεσίες παρουσίασης προκειμένου να εκτελέσουν τη λειτουργία τους. Ο μοναδικός τρόπος προκειμένου να επικοινωνήσουν δύο στοιχεία-χρήστη είναι μέσω μονάδων-δεδομένων-πρωτοκόλλου-εφαρμογής. Αυτές οι μονάδες-δεδομένων-πρωτοκόλλου-εφαρμογής δημιουργούνται από στοιχεία-υπηρεσίας-εφαρμογής.

Πέρα από μεταφορά πληροφορίας, τέτοιες υπηρεσίες μπορεί να περιλαμβάνουν, αλλά δεν είναι περιορισμένες, στα παρακάτω:

- α) αναγνώριση πιθανών ζευγαριών επικοινωνίας (για παράδειγμα με το όνομα, τη διεύθυνση, ορισμένη περιγραφή, ή γενική περιγραφή),
- β) καθορισμός της τωρινής διαθεσιμότητας πιθανών ζευγαριών επικοινωνίας,
- γ) εγκατάσταση του ελέγχου δυνατότητας επικοινωνίας,
- δ) συμφωνία για τους μηχανισμούς ασφαλείας,
- ε) πιστοποίηση των πιθανών ζευγαριών επικοινωνίας,
- στ) επιλογή της μεθοδολογίας καθορισμού κόστους,
- ζ) καθορισμός της επάρκειας των στοιχείων,
- η) καθορισμός της αποδεκτής ποιότητας υπηρεσίας (για παράδειγμα χρόνος απόκρισης, ανεκτός ρυθμός λαθών, κόστος),
- θ) συγχρονισμός των συνεργαζόμενων εφαρμογών,
- ι) επιλογή της μεθόδου διαλόγου η οποία περιλαμβάνει και τις διαδικασίες αρχικοποίησης και κατάργησης,
- κ) συμφωνία για την ευθύνη επαναφοράς από λάθη,
- λ) συμφωνία στις διαδικασίες για τον έλεγχο της ακεραιότητας των δεδομένων, και
- μ) αναγνώριση των περιορισμών στο συντακτικό των δεδομένων (σύνολα χαρακτηρών, δομές δεδομένων).

Το Επίπεδο Εφαρμογής περιέχει όλες τις λειτουργίες εκείνες, οι οποίες αφορούν την επικοινωνία μεταξύ ανοικτών συστημάτων και δεν έχουν ήδη αντιμετωπιστεί σε χαμηλότερα επίπεδα. Αυτές περιλαμβάνουν τις λειτουργίες που εκτελούνται από τα προγράμματα, καθώς και τις λειτουργίες που εκτελούνται από τους χρήστες.

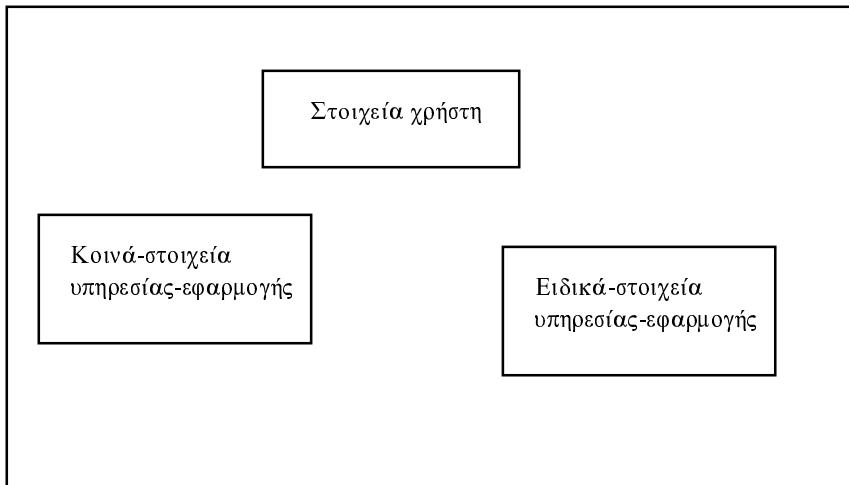
Όταν ένα στιγμιότυπο μιας διαδικασίας-εφαρμογής θελήσει να επικοινωνήσει με ένα άλλο σε διαφορετικό ανοικτό σύστημα, θα πρέπει να καλέσει ένα στιγμιότυπο μιας οντότητας-εφαρμογής στο Επίπεδο Εφαρμογής του δικού της ανοικτού συστήματος. Το στιγμιότυπο της οντότητας-εφαρμογής αυτής, αναλαμβάνει να εγκαταστήσει μια συσχέτιση με την κατάλληλη οντότητα-εφαρμογής στο ανοικτό σύστημα-προορισμού. Η διαδικασία αυτή συμβαίνει μέσα από διαδοχικές κλήσεις στιγμιότυπων σε χαμηλότερα

επίπεδα. Όταν η συσχέτιση μεταξύ δύο οντότητων-εφαρμογής έχει εγκαταστηθεί, οι διαδικασίες-εφαρμογής μπορούν να επικοινωνήσουν.

Μια οντότητα-εφαρμογής μπορεί να είναι δομημένη εσωτερικά σε ομάδες λειτουργιών. Η χρησιμοποίηση μιας ομάδας από λειτουργίες μπορεί να εξαρτάται από την χρησιμοποίηση κάποιων άλλων λειτουργιών, και οι λειτουργίες οι οποίες χρησιμοποιούνται κάθε φορά μπορεί να μεταβάλλονται κατά τη διάρκεια μιας σύνδεσης.

Η δόμηση των οντοτήτων-εφαρμογής σε στοιχεία-υπηρεσίας-εφαρμογής και στοιχεία-χρήστη παρέχει μια οργάνωση των λειτουργιών των οντοτήτων-εφαρμογής. Ακόμα περισσότερο, κάθε δεδομένο υποσύνολο στοιχείων-υπηρεσίας-εφαρμογής, μαζί με τα στοιχεία-χρήστη, αποτελούν ένα είδος οντότητας-εφαρμογής. Κάθε είδος οντότητας-εφαρμογής, και κάθε στιγμιότυπο μπορούν να αναγνωρισθούν με μοναδικό τρόπο. Μια διαδικασία-εφαρμογής μπορεί να καθορίζει την ομάδα λειτουργιών που θα αποτελέσουν την οντότητα-εφαρμογής.

Δύο κατηγορίες στοιχείων-υπηρεσίας-εφαρμογής αναγνωρίζονται: **στοιχεία-υπηρεσίας-κοινής-εφαρμογής** και **στοιχεία-υπηρεσίας-ειδικής-εφαρμογής**. Τα στοιχεία-υπηρεσίας-κοινής-εφαρμογής παρέχουν δυνατότητες που είναι γενικά χρήσιμες σε μια μεγάλη ποικιλία εφαρμογών. Τα στοιχεία-υπηρεσίας-ειδικής-εφαρμογής παρέχουν δυνατότητες που απαιτούνται για την ικανοποίηση συγκεκριμένων αναγκών ειδικών εφαρμογών (για παράδειγμα μεταφορά αρχείων, πρόσβαση βάσεων δεδομένων, τραπεζικές ανάγκες, εισαγωγή παραγγελιών). Οι οντότητες-εφαρμογής μπορεί να περιέχουν στοιχεία-υπηρεσίας-εφαρμογής και από τις δύο κατηγορίες, όπως φαίνεται στο σχήμα 3.4.



**Σχήμα 3.4 - Οντότητα-εφαρμογής**

Ο διαχωρισμός των στοιχείων-υπηρεσίας-εφαρμογής στις δύο κατηγορίες αυτές δεν συνεπάγεται την ύπαρξη δύο διαφορετικών πρωτοκόλλων.

### 3.3.2. Επίπεδο Παρουσίασης

Το Επίπεδο Παρουσίασης παρέχει τα μέσα για την αναπαράσταση της πληροφορίας την οποία ανταλλάσσουν οντότητες-εφαρμογής ή στην οποία αναφέρονται.

Το Επίπεδο Παρουσίασης καλύπτει δύο συμπληρωματικές πλευρές της αναπαράστασης της πληροφορίας:

- α) την αναπαράσταση των δεδομένων που θα μεταφερθούν μεταξύ των οντοτήτων-εφαρμογής, και
- β) την αναπαράσταση των δομών δεδομένων στις οποίες οι οντότητες-εφαρμογής αναφέρονται κατά την επικοινωνία τους, και την αναπαράσταση μιας σειράς από λειτουργίες οι οποίες είναι δυνατόν να εκτελεστούν στις δομές δεδομένων αυτές.

Οι συμπληρωματικές πλευρές στην αναπαράσταση της πληροφορίας που περιγράφηκαν παραπάνω αναφέρονται στην γενική έννοια του συντακτικού μεταφοράς (βλ. και Abstract Syntax Notation 1 - ASN.1 και Basic Encoding Rules - BER στο Κεφάλαιο 4).

Το Επίπεδο Παρουσίασης ασχολείται μονάχα με την σύνταξη, δηλ. την αναπαράσταση των δεδομένων και όχι με την σημασία τους, δηλ. την σημασία τους για το Επίπεδο Εφαρμογής, η οποία είναι γνωστή μονάχα στις οντότητες-εφαρμογής. Παρέχει τα μέσα για την χρήση μιας κοινής αναπαράστασης από τις οντότητες-εφαρμογής, απαλλάσσοντας τις οντότητες-εφαρμογής από την ανάγκη ενασχόλησης με το πρόβλημα της "κοινής" αναπαράστασης της πληροφορίας, δηλ. τους παρέχει μια ανεξαρτησία από τη σύνταξη. Αυτή η ανεξαρτησία από τη σύνταξη μπορεί να περιγραφεί με δύο τρόπους:

- α) το Επίπεδο Παρουσίασης παρέχει κοινά συντακτικά στοιχεία τα οποία χρησιμοποιούνται από οντότητες-εφαρμογής, και
- β) το Επίπεδο Παρουσίασης επιτρέπει στις οντότητες-εφαρμογής να χρησιμοποιήσουν οποιοδήποτε συντακτικό επιθυμούν, ενώ εκείνο παρέχει τα μέσα για την μετατροπή μεταξύ των συντακτικών αυτών και ενός κοινού συντακτικού που απαιτείται για την επικοινωνία οντοτήτων-εφαρμογής. Η μετατροπή αυτή πραγματοποιείται μέσα στα ανοικτά συστήματα, δεν είναι ορατή σε άλλα ανοικτά συστήματα και για τον λόγο αυτό δεν περιορίζει την προτυποποίηση των πρωτοκόλλων-παρουσίασης.

Το Επίπεδο Παρουσίασης παρέχει υπηρεσίες-συνόδου και τις ακόλουθες ευκολίες:

- α) μετατροπή συντακτικού, και
- β) επιλογή συντακτικού.

Η μετατροπή συντακτικού ασχολείται με αλλαγές κώδικα και συνόλου χαρακτήρων, και με την αποδοχή λειτουργιών πάνω στις δομές δεδομένων. Η επιλογή συντακτικού παρέχει τα μέσα για την αρχική επιλογή ενός συντακτικού και την μετέπειτα μεταβολή της επιλογής αυτής.

Το Επίπεδο Παρουσίασης εκτελεί τις παρακάτω λειτουργίες προκειμένου να προσφερθούν οι υπηρεσίες παρουσίασης:

- α) αίτηση εγκατάστασης συνόδου,
- β) μεταφορά δεδομένων,
- γ) διαπραγμάτευση και επαναδιαπραγμάτευση του συντακτικού,

δ) μετατροπή του συντακτικού η οποία περιλαμβάνει και μετατροπή των δεδομένων, δόμηση και μετατροπές ειδικού σκοπού (για παράδειγμα συμπίεση), και

ε) αίτηση τερματισμού συνόδου.

Το γεγονός ότι υπάρχει ή δεν υπάρχει πραγματική μετατροπή συντακτικού δεν έχει καμμιά επίδραση στο πρωτόκολλο-παρουσίασης.

Υπάρχουν τρεις συντακτικές εκδόσεις των δεδομένων: η σύνταξη που χρησιμοποιείται από την αρχική οντότητα-εφαρμογής, η σύνταξη που χρησιμοποιείται από την τελική οντότητα-εφαρμογής, και η σύνταξη που χρησιμοποιείται μεταξύ των οντοτήτων-παρουσίασης (το συντακτικό μεταφοράς). Είναι ολοφάνερα πιθανό ότι δύο ή και οι τρεις από τις παραπάνω συντάξεις μπορεί να είναι οι ίδιες. Το Επίπεδο Παρουσίασης περιέχει τις απαραίτητες υπηρεσίες για την μετατροπή μεταξύ του συντακτικού μεταφοράς και των άλλων δύο, όπως απαιτείται κάθε φορά.

Δεν υπάρχει ένα μοναδικό προκαθορισμένο συντακτικό μεταφοράς για όλες τις Διασύνδεσεις Ανοικτών Συστημάτων. Το συντακτικό μεταφοράς το οποίο θα χρησιμοποιηθεί σε μια σύνδεση-παρουσίασης είναι αντικείμενο διαπραγμάτευσης μεταξύ των οντοτήτων παρουσίασης που θα επικοινωνήσουν. Έτσι, μια οντότητα-παρουσίασης πρέπει να ξέρει το συντακτικό της οντότητας-εφαρμογής της και το συμφωνημένο συντακτικό μεταφοράς. Μόνο το συντακτικό μεταφοράς πρέπει να αναφέρεται στα πρωτόκολλα του Επιπέδου Παρουσίασης.

Προκειμένου να ικανοποιηθεί την απαιτούμενη υπηρεσία που καθορίζεται από τις οντότητες-εφαρμογής κατά την φάση αρχικοποίησης το Επίπεδο Παρουσίασης πρέπει να χρησιμοποιηθεί οποιοδήποτε διαθέσιμο συντακτικό μεταφοράς. Προκειμένου να ικανοποιηθούν άλλοι στόχοι (για παράδειγμα συμπίεσης του όγκου των δεδομένων, για μείωση του κόστους μεταφοράς των δεδομένων), μετατροπή του συντακτικού μπορεί να πραγματοποιηθεί είτε σαν μια υπηρεσία ταιριάσματος-συντάξεων που παρέχεται στις οντότητες-εφαρμογής, είτε σαν μια λειτουργία εσωτερική στο Επίπεδο Παρουσίασης.

Η διαπραγμάτευση του συντακτικού πραγματοποιείται με επικοινωνία μεταξύ των οντοτήτων-παρουσίασης για λογαριασμό των οντοτήτων-εφαρμογής προκειμένου να καθοριστεί η μορφή που τα δεδομένα θα έχουν στο περιβάλλον OSI. Οι διαπραγματεύσεις θα καθορίσουν τι μετατροπές χρειάζονται (αν χρειάζονται) και που θα εκτελεστούν. Οι διαπραγματεύσεις μπορεί να περιοριστούν κατά τη φάση αρχικοποίησης είτε μπορεί να λάβουν χώρα οποιαδήποτε στιγμή της συνόδου.

Στη Διασύνδεση Ανοικτών Συστημάτων, τα συντακτικά που χρησιμοποιούνται από τις οντότητες-εφαρμογής που επιθυμούν να επικοινωνήσουν μπορεί να είναι πολύ όμοια ή εντελώς ανόμοια. Όταν είναι όμοια, οι λειτουργίες μετατροπής μπορεί να μην χρειαστούν καθόλου. Εντούτοις, όταν είναι ανόμοια, οι υπηρεσίες του Επιπέδου Παρουσίασης παρέχουν τα μέσα για την μετατροπή και για την απόφαση που θα λάβουν χώρα οι μετατροπές.

### 3.3.3. Επίπεδο Συνόδου

Ο σκοπός του Επιπέδου Συνόδου είναι να προσφέρει τα απαραίτητα μέσα για την συνεργασία οντοτήτων-παρουσίασης, την οργάνωση και το συγχρονισμό του διαλόγου τους, και τη διαχείριση της ανταλλαγής δεδομένων μεταξύ αυτών. Προκειμένου να το επιτύχει, το Επίπεδο Συνόδου παρέχει υπηρεσίες για την εγκατάσταση συνδέσεων-συνόδου μεταξύ δύο οντοτήτων-παρουσίασης.

Για την υλοποίηση της μεταφοράς δεδομένων μεταξύ οντοτήτων-παρουσίασης, η σύνδεση-συνόδου αντιστοιχίζεται και χρησιμοποιεί μια σύνδεση-μεταφοράς. Μια σύνδεση-συνόδου δημιουργείται όταν αυτό ζητείται από μια οντότητα-παρουσίασης σε ένα σημείο-πρόσβασης-υπηρεσίας-συνόδου. Κατά τη διάρκεια της ζωής της σύνδεσης-συνόδου, υπηρεσίες συνόδου χρησιμοποιούνται από τις οντότητες-παρουσίασης προκειμένου να ρυθμίσουν το διάλογό τους, και να εξασφαλίσουν μια τακτική ανταλλαγή μηνυμάτων. Η σύνδεση-συνόδου υπάρχει μέχρι να καταργηθεί, είτε από τις οντότητες-παρουσίασης, είτε από τις οντότητες-συνόδου. Για όσο χρόνο η σύνδεση-συνόδου υφίσταται, οι υπηρεσίες συνόδου διατηρούν την κατάσταση του διαλόγου ακόμα και σε περιπτώσεις απώλειας δεδομένων στο Επίπεδο Μεταφοράς.

Μια οντότητα-παρουσίασης μπορεί να έχει πρόσβαση σε μια άλλη οντότητα-παρουσίασης μονάχα αρχικοποιώντας ή αποδεχόμενη μια σύνδεση-συνόδου. Μια οντότητα-παρουσίασης μπορεί να σχετίζεται με πολλαπλές συνδέσεις-συνόδου ταυτόχρονα. Μεταξύ δύο οντοτήτων-παρουσίασης είναι πιθανές τόσο παράλληλες, όσο και διαδοχικές συνδέσεις-συνόδου.

Η οντότητα-παρουσίασης που ξεκινά κάποια σύνδεση-συνόδου καθορίζει την οντότητα-παρουσίασης προορισμό με μία διεύθυνση-συνόδου. Σε πολλά συστήματα, μια διεύθυνση-μεταφοράς μπορεί να χρησιμοποιηθεί σαν διεύθυνση-συνόδου, δηλ. υπάρχει αντιστοιχία μία-προς-μία μεταξύ διευθύνσεων-συνόδου και διευθύνσεων-μεταφοράς. Στην γενική περίπτωση, εντούτοις, υπάρχει αντιστοιχία πολλές-προς-μία μεταξύ διευθύνσεων-συνόδου και διευθύνσεων-μεταφοράς.

Οι ακόλουθες υπηρεσίες που προσφέρονται από το Επίπεδο Συνόδου περιγραφονται παρακάτω:

- α) εγκατάσταση σύνδεσης-συνόδου,
- β) κατάργηση σύνδεσης-συνόδου,
- γ) ανταλλαγή κανονικών-δεδομένων,
- δ) υπηρεσία απομόνωσης,
- ε) ανταλλαγή επειγόντων-δεδομένων,
- στ) αλληλεπίδραση διαχείρισης,
- ζ) συγχρονισμός σύνδεσης-συνόδου, και
- η) αναφορά εξαιρέσεων

Η υπηρεσία εγκατάστασης σύνδεσης-συνόδου επιτρέπει σε δύο οντότητες-παρουσίασης να εγκαταστήσουν μια σύνδεση-συνόδου μεταξύ τους. Οι οντότητες-παρουσίασης αναγνωρίζονται από διευθύνσεις-συνόδου που χρησιμοποιούνται κατά την αίτηση εγκατάστασης σύνδεσης-συνόδου. Η υπηρεσία εγκατάστασης σύνδεσης-συνόδου επιτρέπει στις οντότητες-παρουσίασης να συνεργαστούν προκειμένου να καθορίσουν μοναδικές τιμές για τις παραμέτρους σύνδεσης-συνόδου την χρονική στιγμή εγκατάστασης της σύνδεσης-συνόδου. Ταυτόχρονες αιτήσεις εγκατάστασης σύνδεσης-συνόδου τυπικά καταλήγουν σε ένα αντίστοιχο αριθμό από συνδέσεις-συνόδου, αλλά μια οντότητα-συνόδου μπορεί πάντοτε να απορρίψει κάποια αίτηση σύνδεσης.

Η υπηρεσία εγκατάστασης σύνδεσης-συνόδου παρέχει στις οντότητες-παρουσίασης ένα αναγνωριστικό-σύνδεσης-υπηρεσίας-συνόδου το οποίο καθορίζει με μοναδικό τρόπο τη σύνδεση-συνόδου των οντοτήτων-παρουσίασης, με έναν χρόνο ζωής ο οποίος μπορεί να είναι μεγαλύτερος από εκείνον της σύνδεσης-συνόδου. Το αναγνωριστικό αυτό μπορεί

να χρησιμοποιηθεί από τις οντότητες-παρουσίασης προκειμένου να αναφερθούν σε μια σύνδεση-συνόδου κατά τη διάρκεια του χρόνου ζωής της σύνδεσης-συνόδου, και μπορεί επίσης να χρησιμοποιηθεί από οντότητες-διαχείρισης για διαχειριστικούς σκοπούς όπως λογιστική διαχείριση.

Η υπηρεσία κατάργησης σύνδεσης-συνόδου επιτρέπει στις οντότητες-παρουσίασης να λύσουν μια σύνδεση-συνόδου με ένα τακτικό τρόπο χωρίς απώλειες δεδομένων. Επίσης επιτρέπει σε οποιαδήποτε από τις οντότητες-παρουσίασης που επικοινωνούν να εγκαταλείψουν μια σύνδεση-συνόδου, στην περίπτωση αυτή όμως, δεδομένα μπορεί να χαθούν.

Η κατάργηση μιας σύνδεσης-συνόδου μπορεί επίσης να ξεκινήσει και από τις οντότητες-συνόδου που την υποστηρίζουν. Η υπηρεσία ανταλλαγής κανονικών δεδομένων επιτρέπει σε μια οντότητα-παρουσίασης που στέλνει να μεταφέρει μια μονάδα-δεδομένων-υπηρεσίας-συνόδου προς μια οντότητα-παρουσίασης που λαμβάνει. Η υπηρεσία αυτή επιτρέπει στην οντότητα-παρουσίασης που λαμβάνει να εξασφαλίσει ότι δεν θα υπερχειλίσει από δεδομένα.

Η υπηρεσία απομόνωσης επιτρέπει σε μια οντότητα-παρουσίασης που στέλνει να απαιτήσει ένας αριθμός από μονάδες-δεδομένων-υπηρεσίας-συνόδου να μην παραδοθούν στην οντότητα-παρακολούθησης που λαμβάνει μέχρις ότου αυτό επιτραπεί από την οντότητα-παρουσίασης που στέλνει. Η οντότητα-παρακολούθησης που στέλνει μπορεί να απαιτήσει όλα τα δεδομένα που είναι κάποια στιγμή σε απομόνωση να απορριφθούν. Η οντότητα-παρουσίασης που λαμβάνει δεν παίρνει κάποια ειδοποίηση για το ότι τα δεδομένα που λαμβάνει ήταν κάποια στιγμή σε απομόνωση, ή για το ότι κάποια δεδομένα απορρίφθηκαν.

Η υπηρεσία ανταλλαγής επειγόντων δεδομένων παρέχει επείγων χειρισμό κατά την μεταφορά επειγόντων μονάδων-δεδομένων-υπηρεσίας-συνόδου. Τίθεται περιορισμός μεγέθους στις επείγουσες μονάδες-δεδομένων-υπηρεσίας-συνόδου. Η υπηρεσία αυτή μπορεί να χρησιμοποιηθεί από μια οντότητα-παρουσίασης οποιαδήποτε στιγμή στον χρόνο ζωής μιας σύνδεσης συνόδου.

Η υπηρεσία αλληλεπίδρασης διαχείρισης επιτρέπει στις οντότητες-παρουσίασης να ελέγξουν τη σειρά με την οποία θα εκτελέσουν συγκεκριμένες λειτουργίες ελέγχου. Η υπηρεσία προσφέρει τη δυνατότητα εθελοντικής ανταλλαγής σειράς, όπου η οντότητα-παρουσίασης που έχει σειρά παραιτείται εθελοντικά. Η υπηρεσία επίσης προσφέρει τη δυνατότητα εσπευμένης αλλαγής σειράς, όπου μετά από αίτηση από οντότητα-παρουσίασης η οποία δεν έχει σειρά, η υπηρεσία-συνόδου μπορεί να αναγκάσει την οντότητα-παρουσίασης που έχει τη σειρά να την παρατήσει. Στην περίπτωση αυτή, δεδομένα μπορεί να χαθούν.

Ορίζονται οι ακόλουθοι τύποι μονάδων-δεδομένων-υπηρεσίας-συνόδου ανταλλαγής αλληλεπίδρασης:

- α) ταυτόχρονα-αμφίδρομη (TWS),
- β) εναλλακτικά-αμφίδρομη (TWA), και
- γ) μονόδρομη αλληλεπίδραση.

Η υπηρεσία συγχρονισμού σύνδεσης-συνόδου επιτρέπει στις οντότητες-παρουσίασης να:

- α) ορίζουν και να αναγνωρίζουν σημεία συγχρονισμού, και
- β) επαναφέρουν την σύνδεση-συνόδου σε μια ορισμένη κατάσταση και ένα συμφωνημένο σημείο συγχρονισμού.

Το Επίπεδο Συνόδου δεν είναι υπεύθυνο για κάθε σχετικό έλεγχο των σημείων αυτών και για κάθε σχετική με το συγχρονισμό δεσμευτική λειτουργία.

Η υπηρεσία αναφοράς εξαιρέσεων επιτρέπει στις οντότητες-παρουσίασης να ειδοποιηθούν για εξαιρετικές καταστάσεις που δεν καλύπτονται από άλλες υπηρεσίες, τέτοιες όπως μη επαναφερομένες δυσλειτουργίες συνόδου.

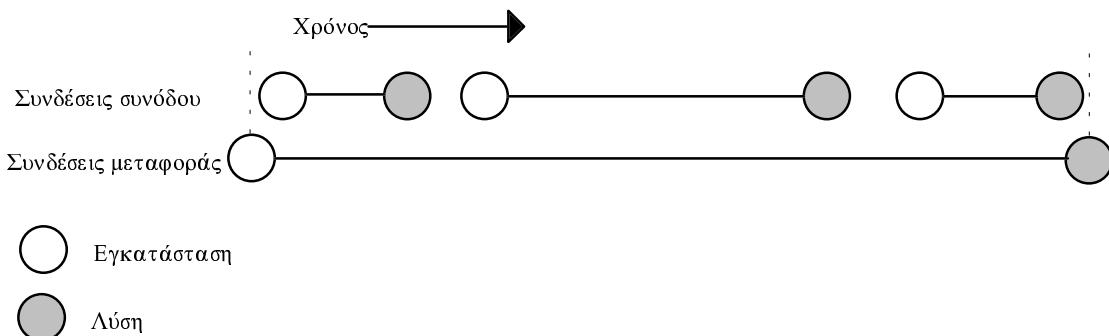
- α) αριθμηση διαδοχής μονάδων-δεδομένων-υπηρεσίας-συνόδου,
- β) παρενθέσεις,
- γ) σταμάτημα-συνέχεια, και
- δ) ασφάλεια.

Οι λειτουργίες στο Επίπεδο Συνόδου είναι εκείνες οι οποίες πρέπει να εκτελεστούν από οντότητες-συνόδου προκειμένου να προσφερθούν υπηρεσίες-συνόδου. Οι περισσότερες από τις απαιτούμενες λειτουργίες είναι άμεσα προφανείς από τις παρεχόμενες υπηρεσίες. Περισσότερες λεπτομέρειες δίνονται παρακάτω για τις εξής λειτουργίες:

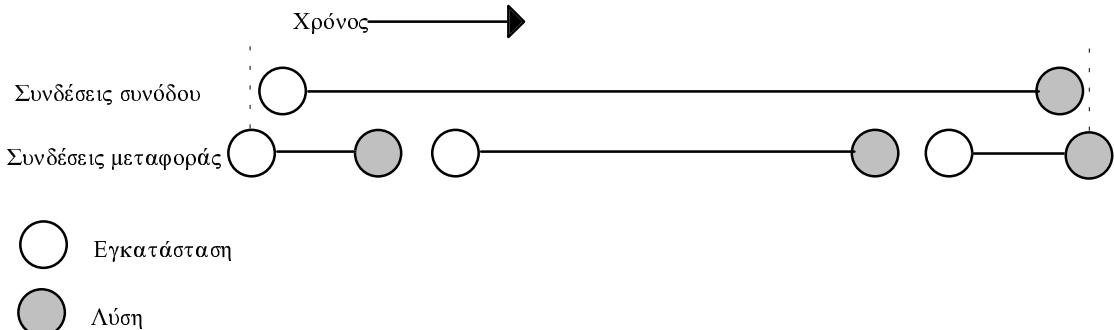
- α) αντιστοιχία σύνδεσης-συνόδου σε σύνδεση-μεταφοράς,
- β) έλεγχος ροής σύνδεσης-συνόδου,
- γ) μεταφορά επειγόντων δεδομένων,
- δ) επαναφορά σύνδεσης-συνόδου,
- ε) κατάργηση σύνδεσης-συνόδου, και
- στ) διαχείριση Επιπέδου Συνόδου.

Υπάρχει μία-προς-μία αντιστοιχία μεταξύ μιας σύνδεσης-συνόδου και μιας σύνδεσης-μεταφοράς σε μια δεδομένη χρονική στιγμή. Εντούτοις, ο χρόνος ζωής μιας σύνδεσης μεταφοράς και της αντίστοιχης σύνδεσης-συνόδου μπορεί να είναι διαφορετικοί οπότε οι παρακάτω περιπτώσεις ορίζονται:

- α) μια σύνδεση μεταφοράς υποστηρίζει πολλαπλές διαδοχικές συνδέσεις-συνόδου (βλ. σχήμα 3.5), και
- β) πολλαπλές διαδοχικές συνδέσεις-μεταφοράς μπορεί να υποστηρίζουν μια σύνδεση-συνόδου (βλ. σχήμα 3.6).



### Σχήμα 3.5 - Μια σειρά από διαδοχικές συνδέσεις-συνόδου



### Σχήμα 3.6 - Μια σειρά από διαδοχικές συνδέσεις-μεταφοράς

Δεν υπάρχει ομότιμος έλεγχος ροής στο Επίπεδο Συνόδου. Για να προστατευθεί η οντότητα-συνόδου που λαμβάνει από πιθανή υπερχείλιση με δεδομένα, αυτή χρησιμοποιεί τον έλεγχο ροής του επιπέδου-μεταφοράς προκειμένου να περιορίσει τη μεταφορά δεδομένων στη σύνδεση μεταφοράς.

Η μεταφορά επειγόντων δεδομένων γενικά πραγματοποιείται με τη χρήση υπηρεσιών επείγουσας μεταφοράς.

Για τη περίπτωση της αναφοράς καταστροφής υποκείμενης σύνδεσης-μεταφοράς, το Επίπεδο Συνόδου περιλαμβάνει απαραίτητες λειτουργίες ώστε να επαναεγκαταστήσει μια άλλη σύνδεση-μεταφοράς, προκειμένου να υποστηριχθεί η σύνδεση-συνόδου που εξακολουθεί να υφίσταται. Οι οντότητες-συνόδου που λαμβάνουν μέρος ειδοποιούν τις οντότητες-παρουσίασης με την υπηρεσία αναφοράς εξαιρέσεων ότι η υπηρεσία έχει διακοπεί και επαναφέρουν την υπηρεσία ανάλογα με την εντολή των οντοτήτων-παρουσίασης. Αυτό επιτρέπει στις οντότητες-παρουσίασης να επανασυγχρονιστούν και να συνεχίσουν από μια συμφωνημένη κατάσταση.

Το Επίπεδο Συνόδου περιέχει τις απαραίτητες λειτουργίες για την κατάργηση συνδέσεων-συνόδου με ένα τακτικό τρόπο, χωρίς απώλεια δεδομένων, μετά από αίτηση από οντότητες-παρουσίασης. Το Επίπεδο Συνόδου επίσης περιέχει τις απαραίτητες λειτουργίες για τη διακοπή μιας σύνδεσης-συνόδου με πιθανή απώλεια δεδομένων.

#### 3.3.4 Επίπεδο Μεταφοράς

Η υπηρεσία-μεταφοράς παρέχει διαφανή μεταφορά δεδομένων μεταξύ οντοτήτων-συνόδου και τις απαλλάσσει από κάθε απασχόληση με τον ακριβή τρόπο επίτευξης αξιόπιστης και οικονομικά συμφέρουσας μεταφοράς δεδομένων.

Το Επίπεδο Μεταφοράς βελτιστοποιεί τη χρήση της διαθέσιμης υπηρεσίας-δικτύου προκειμένου να προσφέρει την απαιτούμενη επίδοση σε κάθε οντότητα-συνόδου και με το ελάχιστο κόστος. Η βελτιστοποίηση αυτή επιτυγχάνεται λαμβάνοντας υπ' όψιν τους συνολικούς περιορισμούς που τίθονται από όλες τις οντότητες-συνόδου, καθώς και την

**συνολική ποιότητα και χωρητικότητα της υπηρεσίας-δικτύου που γίνεται διαθέσιμη στο Επίπεδο Μεταφοράς.**

Όλα τα πρωτόκολλα που ορίζονται για το επίπεδο μεταφοράς έχουν σημασία από άκρη-σε-άκρη, όπου οι άκρες ορίζονται σαν ανταποκρινόμενες οντότητες-μεταφοράς. **Για το λόγο αυτό το Επίπεδο Μεταφοράς είναι προσανατολισμένο προς τα τελικά ανοικτά συστήματα OSI, ενώ τα πρωτόκολλα μεταφοράς λειτουργούν πάντοτε μεταξύ τελικών ανοικτών συστημάτων OSI (βλ. Σχήμα 3.1 και 3.3).**

Το Επίπεδο Μεταφοράς έχει απαλλαχτεί από κάθε ευθύνη για τη δρολόγηση και την αναμετάδοση από την στιγμή που οι οντότητες-δικτύου παρέχουν συνδέσεις-δικτύου από κάθε οντότητα-μεταφοράς προς κάθε άλλη, συμπεριλαμβάνοντας και την περίπτωση συνεχόμενων υποδικτύων.

Οι λειτουργίες μεταφοράς που καλούνται στο Επίπεδο Μεταφοράς για να προσφέρουν μια απαιτούμενη ποιότητα υπηρεσίας εξαρτώνται άμεσα από τη ποιότητα της υπηρεσίας-δικτύου. Η ποιότητα της υπηρεσίας-δικτύου εξαρτάται από τον τρόπο με τον οποίο επιτυγχάνεται η υπηρεσία-δικτύου.

Το Επίπεδο Μεταφοράς αναγνωρίζει με μοναδικό τρόπο κάθε οντότητα-συνόδου από τη διεύθυνσή-μεταφοράς. Η υπηρεσία-μεταφοράς παρέχει τα μέσα για την εγκατάσταση, διατήρηση και κατάργηση συνδέσεων-μεταφοράς. Συνδέσεις-μεταφοράς παρέχουν τη δυνατότητα αμφίδρομης μετάξυ ενός ζευγαριού διευθύνσεων-μεταφοράς. Περισσότερες από μία συνδέσεις-μεταφοράς είναι δυνατόν να εγκατασταθούν μεταξύ του ίδιου ζευγαριού διευθύνσεων-μεταφοράς. Μια οντότητα-συνόδου χρησιμοποιεί αναγνωριστικά-τερματισμού-συνδέσεων-μεταφοράς που δίνονται από το Επίπεδο Μεταφοράς για το διαχωρισμό μεταξύ τερματισμών-συνδέσεων-μεταφοράς.

Η λειτουργία μιας σύνδεσης-μεταφοράς είναι ανεξάρτητη από τη λειτουργία κάθε άλλης εκτός των περιορισμών που εισάγονται από τα πεπερασμένα στοιχεία που είναι διαθέσιμα στο Επίπεδο Μεταφοράς.

Η ποιότητα της υπηρεσίας που παρέχεται σε μια σύνδεση-μεταφοράς εξαρτάται από την κλάση υπηρεσίας που ζητήθηκε από τις οντότητες-συνόδου κατά την εγκατάσταση σύνδεσης-μεταφοράς. Η επιλεγμένη ποιότητα υπηρεσίας διατηρείται καθόλη τη διάρκεια της σύνδεσης-μεταφοράς. Η οντότητα-συνόδου ειδοποιείται για οποιαδήποτε αποτυχία στην προσπάθεια διατήρησης της επιλεγμένης ποιότητας υπηρεσίας μιας δεδομένης σύνδεσης μεταφοράς.

Οι ακόλουθες υπηρεσίες παρέχονται από το Επίπεδο Μεταφοράς και περιγράφονται παρακάτω:

- α) εγκατάσταση σύνδεση-μεταφοράς,
- β) μεταφορά δεδομένων, και
- γ) κατάργηση σύνδεσης μεταφοράς.

Συνδέσεις-μεταφοράς πραγματοποιούνται μεταξύ οντοτήτων-συνόδου και αναγνωρίζονται με τη βοήθεια διευθύνσεων-μεταφοράς. Η ποιότητα-υπηρεσίας για την σύνδεση-μεταφοράς είναι αντικείμενο διαπραγμάτευσης μεταξύ των οντοτήτων-συνόδου και της υπηρεσίας-μεταφοράς. Τη χρονική στιγμή εγκατάστασης σύνδεσης-μεταφοράς η κλάση της υπηρεσίας-μεταφοράς που θα προσφερθεί μπορεί να επιλεχθεί μέσα από ένα ορισμένο σύνολο διαθέσιμων κλάσεων υπηρεσίας. Οι κλάσεις υπηρεσίας αυτές χαρακτηρίζονται από συνδυασμούς επιλεγμένων τιμών παραμέτρων όπως ρυθμός απόδοσης, καθυστέρηση μεταφοράς, και καθυστέρηση εγκατάστασης σύνδεσης, καθώς και από εγγυημένων τιμών παραμέτρων όπως ρυθμός υπολειπόμενων σφαλμάτων και

διαθεσιμότητα υπηρεσίας. Οι κλάσεις υπηρεσίας αυτές παριστάνουν καθολικά ορισμένους συνδυασμούς παραμέτρων που ελέγχουν την ποιότητα υπηρεσίας και στοχεύουν στην κάλυψη των απαιτήσεων για υπηρεσίες-μεταφοράς διαφόρων ειδών φορτίου που γεννιέται από οντότητες-συνόδου.

Η υπηρεσία μεταφοράς δεδομένων δίνει τη δυνατότητα για μεταφορά δεδομένων σύμφωνα με την επιλεγμένη ποιότητα υπηρεσίας. Όταν η δεδομένη ποιότητα υπηρεσίας δεν είναι δυνατόν να προσφερθεί και όλες οι πιθανές προσπάθειες επαναφοράς αποτύχουν, η σύνδεση-μεταφοράς τερματίζεται και οι οντότητες-συνόδου ειδοποιούνται.

- α) Η υπηρεσία μεταφοράς μονάδων-δεδομένων-υπηρεσίας-μεταφοράς παρέχει τα μέσα με τα οποία μονάδες-δεδομένων-υπηρεσίας-μεταφοράς τυχαίου μήκους μεταφέρονται διαφανώς με σειρά από ένα σημείο-πρόσβασης-υπηρεσίας-μεταφοράς που στέλνει δεδομένα προς ένα ένα σημείο-πρόσβασης-υπηρεσίας-μεταφοράς που δέχεται δεδομένα πάνω από μια σύνδεση μεταφοράς. Η υπηρεσία αυτή υπόκειται σε έλεγχο ροής.
- β) Η υπηρεσία μεταφοράς επειγόντων-μονάδων-δεδομένων-υπηρεσίας-μεταφοράς παρέχει ένα πρόσθετο τρόπο για την ανταλλαγή πληροφορίας πάνω από μια σύνδεση-μεταφοράς. Οι επείγουσες-μονάδες-δεδομένων-μεταφοράς υπόκεινται σε ένα δικό τους σύνολο χαρακτηριστικών υπηρεσίας-μεταφοράς και ελέγχου ροής. Το μέγιστο μέγεθος της επείγουσας-μονάδας-δεδομένων-υπηρεσίας-μεταφοράς είναι περιορισμένο.

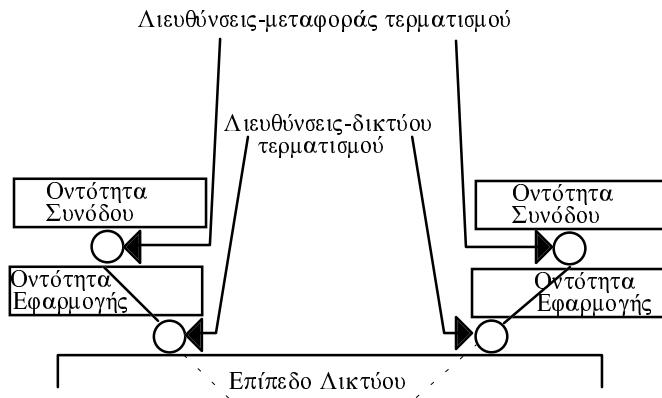
Η υπηρεσία κατάργησης σύνδεσης μεταφοράς παρέχει τα μέσα προκειμένου κάποια οντότητα-συνόδου να μπορεί να τερματίσει μια σύνδεση-μεταφοράς και να ειδοποιήσει την ανταποκρινόμενη οντότητα-συνόδου για την κατάργηση.

Οι λειτουργίες στο Επίπεδο Μεταφοράς περιλαμβάνουν:

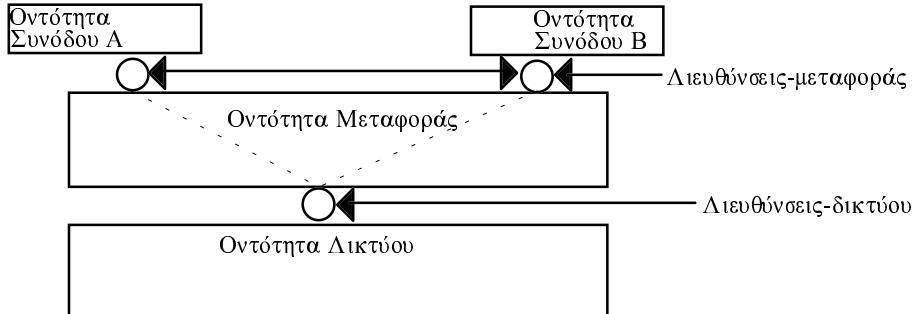
- α) αντιστοιχία των διευθύνσεων μεταφοράς σε διευθύνσεις-δικτύου,
- β) πολυπλεξία (από άκρη σε άκρη) συνδέσεων-μεταφοράς πάνω από συνδέσεις δικτύου,
- γ) εγκατάσταση και κατάργηση συνδέσεων μεταφοράς,
- δ) από άκρη σε άκρη έλεγχος διαδοχής για κάθε σύνδεση,
- ε) από άκρη σε άκρη αναγνώριση λαθών και κάθε αναγκαία παρακολούθηση της ποιότητας υπηρεσίας,
- σ) από άκρη σε άκρη επαναφορά μετά από λάθη,
- ζ) από άκρη σε άκρη κατάτμηση, ένωση και συνένωση,
- η) από άκρη σε άκρη έλεγχος ροής για κάθε σύνδεση,
- θ) λειτουργίες επίβλεψης, και
- ι) μεταφορά επειγόντων-μονάδων-δεδομένων-υπηρεσίας-μεταφοράς.

Επειδή οι οντότητες-μεταφοράς υποστηρίζουν υπηρεσίες σε μια βάση από άκρη σε άκρη, δεν απαιτείται κάποια οντότητα-μεταφοράς να χρησιμοποιηθεί σαν αναμεταδότης κάπου μεταξύ των τελικών οντοτήτων-μεταφοράς. Για το λόγο αυτό το Επίπεδο Μεταφοράς αντιστοιχεί διευθύνσεις-μεταφοράς σε διευθύνσεις-δικτύου οι οποίες χαρακτηρίζουν τις τελικές οντότητες-μεταφοράς (βλ. σχήμα 3.7).

Μια οντότητα-μεταφοράς μπορεί να εξυπηρετεί περισσότερες από μία οντότητες-συνόδου. Επιπλέον, πολλές διευθύνσεις-μεταφοράς μπορεί να σχετίζονται με μία διεύθυνση-δικτύου μέσα στα όρια της ίδιας οντότητας-μεταφοράς. Οι λειτουργίες αντιστοιχίας, που ακολουθούν, εκτελούνται μέσα στις οντότητες-μεταφοράς προκειμένου να προσφερθούν οι ευκολίες αυτές (βλ. σχήμα 3.8).



**Σχήμα 3.7 - Συσχέτιση διευθύνσεων-μεταφοράς και διευθύνσεων-δικτύου**



**Σχήμα 3.8 - Συσχέτιση μιας διεύθυνσης-δικτύου με πολλές διευθύνσεις-μεταφοράς**

Προκειμένου να βελτιστοποιηθεί η χρήση των συνδέσεων-δικτύου, η αντιστοιχία των συνδέσεων-μεταφοράς σε συνδέσεις-δικτύου δεν πρέπει να ακολουθεί μια βάση μία-προς-μία. Μπορεί να χρησιμοποιηθεί τόσο πολυπλεξία, όσο και διαχωρισμός προκειμένου να ελαχιστοποιηθεί το κόστος χρήσης της υπηρεσίας-δικτύου.

Οι φάσεις λειτουργίας στο Επίπεδο Μεταφοράς είναι:

- α) φάση εγκατάστασης,
- β) φάση μεταφοράς δεδομένων, και

γ) φάση κατάργησης.

Η μεταφορά από την μια φάση στην άλλη θα καθοριστεί με λεπτομέρεια στο πρωτόκολλο του Επιπέδου Μεταφοράς.

Κατά την φάση εγκατάστασης, το Επίπεδο Μεταφοράς εγκαθιστά μια σύνδεση-μεταφοράς μεταξύ δύο οντοτήτων συνόδου. Οι λειτουργίες του Επιπέδου Μεταφοράς κατά την φάση αυτή πρέπει να ταιριάζουν την επιλεγμένη κλάση υπηρεσίας με τις υπηρεσίες που παρέχονται από το Επίπεδο Δικτύου. Οι ακόλουθες λειτουργίες είναι δυνατόν να εκτελεστούν κατά την φάση αυτήν:

- α) λήψη μιας σύνδεσης-δικτύου η οποία ταιριάζει με τον καλύτερο τρόπο τις απαιτήσεις της οντότητας-συνόδου, λαμβάνοντας υπ' όψιν κόστος και ποιότητα υπηρεσίας,
- β) απόφαση για την ανάγκη χρήσης πολυπλεξίας ή διαχωρισμού για την βελτιστοποίηση της χρήσης των συνδέσεων-δικτύου,
- γ) καθορισμός του βέλτιστου μήκους για την μονάδα-δεδομένων-πρωτοκόλλου-μεταφοράς,
- δ) επιλογή των λειτουργιών που θα χρησιμοποιηθούν μετά το πέρασμα στη φάση μεταφοράς δεδομένων,
- ε) αντιστοιχία διευθύνσεων-μεταφοράς σε διευθύνσεις-δικτύου,
- στ) παροχή αναγνώρισης των διαφορετικών συνδέσεων μεταφοράς μεταξύ του ίδιου ζευγαριού σημείων-πρόσβασης-υπηρεσίας-μεταφοράς (λειτουργία αναγνώρισης συνδέσεων), και
- ζ) μεταφορά δεδομένων.

Ο σκοπός της φάσης μεταφοράς δεδομένων είναι η μεταφορά μονάδων-δεδομένων-υπηρεσίας-μεταφοράς μεταξύ δύο οντοτήτων-συνόδου συνδεδεμένων με μια σύνδεση-μεταφοράς. Αυτό επιτυγχάνεται με την μετάδοση μονάδων-δεδομένων-πρωτοκόλλου-μεταφοράς καθώς και την εκτέλεση των ακόλουθων λειτουργιών, κάθε μια από τις οποίες χρησιμοποιείται ή δεν χρησιμοποιείται ανάλογα με την επιλεγμένη κλάση υπηρεσίας στην φάση εγκατάστασης:

- α) διαδοχή,
- β) ένωση,
- γ) συνένωση,
- δ) κατάτμηση,
- ε) πολυπλεξία ή διαχωρισμός,
- στ) έλεγχος ροής,
- ζ) αναγνώριση λαθών,
- η) επαναφορά μετά από λάθη,
- θ) μεταφορά επειγόντων δεδομένων,

- ι) αναγνώριση ορίων μονάδων-δεδομένων-υπηρεσίας-μεταφοράς, και
- κ) αναγνώριση συνδέσεων-μεταφοράς.

Ο σκοπός της κατάργησης είναι ο τερματισμός μιας σύνδεσης μεταφοράς. Μπορεί να περιλαμβάνει τις ακόλουθες λειτουργίες:

- α) αναφορά του λόγου κατάργησης,
- β) αναγνώριση της σύνδεσης-μεταφοράς που τερματίζεται, και
- γ) μεταφορά δεδομένων.

### 3.3.5 Επίπεδο Λικτύου

Το Επίπεδο Λικτύου παρέχει τα μέσα για την εγκατάσταση, διατήρηση, και κατάργηση συνδέσεων-δικτύου μεταξύ ανοικτών συστημάτων που περιέχουν οντότητες-εφαρμογής οι οποίες επικοινωνούν, καθώς και τα λειτουργικά και διαδικαστικά μέσα για την ανταλλαγή μονάδων-δεδομένων-υπηρεσίας-δικτύου μεταξύ οντοτήτων-μεταφοράς πάνω από συνδέσεις-δικτύου.

Παρέχει στις οντότητες-μεταφοράς ανεξαρτησία από ευθύνες δρομολόγησης και αναμετάδοσης που σχετίζονται με την εγκατάσταση και την λειτουργία μιας σύνδεσης-δικτύου. Αυτό περιλαμβάνει την περίπτωση όπου διαφορετικά υποδίκτυα χρησιμοποιούνται στη σειρά ή παράλληλα. Κάνει διαφανές στις οντότητες-μεταφοράς τον τρόπο με τον οποίο υποκείμενα στοιχεία, όπως συνδέσεις-σύνδεσης-δεδομένων χρησιμοποιούνται προκειμένου να προσφέρουν συνδέσεις-δικτύου. Η βασική υπηρεσία του Επίπεδου Δικτύου είναι να παρέχει διαφανή μετάδοση δεδομένων μεταξύ οντοτήτων μεταφοράς. Η υπηρεσία αυτή επιτρέπει η δομή και το ακριβές περιεχόμενο των δεδομένων που μεταφέρονται από το Επίπεδο Λικτύου να καθορίζονται από υψηλότερα επίπεδα. Όλες οι υπηρεσίες παρέχονται στο Επίπεδο Μεταφοράς με ένα γνωστό κόστος.

Το Επίπεδο Δικτύου περιέχει τις απαραίτητες λειτουργίες προκειμένου να προσφέρει στο Επίπεδο Μεταφοράς ένα αυστηρό σύνορο Επίπεδου Δικτύου/Μεταφοράς, το οποίο είναι ανεξάρτητο των υποκείμενων επικοινωνιακών μέσων σε οτιδήποτε εκτός της ποιότητας υπηρεσίας. Δηλ. το Επίπεδο Δικτύου περιέχει λειτουργίες απαραίτητες προκειμένου να καλύψει τις διαφορές στα χαρακτηριστικά των διαφόρων τεχνολογιών υποδικτύων και μετάδοσης σε μια σταθερή υπηρεσία δικτύου. Η υπηρεσία η οποία παρέχεται στα δύο άκρα μιας σύνδεσης δικτύου είναι η ίδια ακόμα και αν μια σύνδεση-δικτύου καλύπτει πολλά υποδίκτυα, καθένα από τα οποία προσφέρει διαφορετικές υπηρεσίες.

Η ποιότητα υπηρεσίας είναι αντικείμενο διαπραγμάτευσης μεταξύ των οντοτήτων μεταφοράς και της υπηρεσίας-δικτύου την στιγμή της εγκατάστασης σύνδεσης-δικτύου. Ενώ αυτή η ποιότητα υπηρεσίας μπορεί να μεταβάλλεται από μια σύνδεση δικτύου σε μια άλλη, θα είναι σταθερή και στις δύο άκρες για μια συγκεκριμένη σύνδεση-δικτύου.

Οι ακόλουθες υπηρεσίες ή στοιχεία υπηρεσίας παρέχονται από το Επίπεδο Δικτύου και περιγράφονται παρακάτω:

- α) διευθύνσεις δικτύου,
- β) συνδέσεις δικτύου,

- γ) αναγνωριστικά-τερματισμών-συνδέσεων-δικτύου,
- δ) μεταφορά μονάδων-δεδομένων-υπηρεσίας-δικτύου,
- ε) παράμετροι ποιότητας υπηρεσίας,
- στ) ειδοποίηση για λάθη,
- ζ) διαδοχή,
- η) έλεγχος ροής,
- θ) μεταφορά επειγόντων μονάδων-δεδομένων-υπηρεσίας-δικτύου,
- ι) αρχικοποίηση, και
- κ) κατάργηση.

Μερικές από τις λειτουργίες που περιγράφονται παρακάτω είναι προαιρετικές. Αυτό σημαίνει ότι:

- 1) ο χρήστης μπορεί να ζητήσει την υπηρεσία, και
- 2) ο προμηθευτής των υπηρεσιών-δικτύου μπορεί να εκτελέσει την λειτουργία ή να ειδοποιήσει ότι η υπηρεσία δεν είναι διαθέσιμη.

Οι οντότητες-μεταφοράς είναι γνωστές στο Επίπεδο Δικτύου με τη βοήθεια των διευθύνσεων-δικτύου. Οι διευθύνσεις-δικτύου παρέχονται από το Επίπεδο Δικτύου και μπορούν να χρησιμοποιηθούν από οντότητες-μεταφοράς με μοναδικό τρόπο προκειμένου να αναγνωρίσουν άλλες οντότητες-μεταφοράς, π.χ. οι διευθύνσεις-δικτύου είναι απαραίτητες προκειμένου οντότητες-μεταφοράς να μπορούν να επικοινωνήσουν χρησιμοποιώντας υπηρεσίες δικτύου. Το Επίπεδο Δικτύου αναγνωρίζει με μοναδικό τρόπο κάθε ένα από τα τελικά ανοικτά συστήματα (που παριστάνονται από οντότητες μεταφοράς) με τη βοήθεια των διευθύνσεων-δικτύου. Το γεγονός αυτό μπορεί να είναι ανεξάρτητο του σχήματος διευθυνσιοδότησης που χρησιμοποιείται από τα υποκείμενα επίπεδα.

Μια σύνδεση-δικτύου παρέχει τα μέσα για την μεταφορά δεδομένων μεταξύ οντοτήτων-μεταφοράς καλά καθορισμένων από τις διευθύνσεις-δικτύου. Το Επίπεδο Δικτύου παρέχει τα μέσα για την εγκατάσταση, διατήρηση και κατάργηση συνδέσεων-δικτύου. Μια σύνδεση-δικτύου είναι σύνδεση σημείου-προς-σημείο. Περισσότερες από μία συνδέσεις-δικτύου μπορεί να υφίστανται μεταξύ του ίδιου ζευγαριού διευθύνσεων-δικτύου.

Το Επίπεδο Δικτύου προσφέρει στις οντότητες-μεταφοράς ένα αναγνωριστικό-τερματισμού-σύνδεσης-δικτύου το οποίο αναγνωρίζει με μοναδικό τρόπο τον τερματισμό-σύνδεσης-δικτύου σε μια διεύθυνση-δικτύου.

Σε μια σύνδεση-δικτύου, το Επίπεδο Δικτύου δίνει τη δυνατότητα μετάδοσης μονάδων-δεδομένων-υπηρεσίας-δικτύου. Οι μονάδες αυτές έχουν μια καλά καθορισμένη αρχή και τέλος και η ακεραιότητα του περιεχομένου της μονάδας διατηρείται από το Επίπεδο Δικτύου. Κανένα όριο δεν υπεισέρχεται στο μέγιστο μέγεθος μιας μονάδας-δεδομένων υπηρεσίας-δικτύου. Οι μονάδες-δεδομένων-υπηρεσίας-δικτύου μεταδίδονται με διαφανή τρόπο μεταξύ οντότητων-μεταφοράς.

Το Επίπεδο Δικτύου εγκαθιστά και διατηρεί μια συγκεκριμένη ποιότητα υπηρεσίας για όλη τη διάρκεια της σύνδεσης-δικτύου. Οι παράμετροι ποιότητας υπηρεσίας

περιλαμβάνουν ρυθμό υπολοιπόμενων λαθών, διαθεσιμότητα υπηρεσίας, αξιοπιστία, ρυθμό απόδοσης, καθυστέρηση μεταφοράς (συμπεριλαμβάνοντας διακυμάνσεις), και καθυστέρηση για την εγκατάσταση σύνδεσης δικτύου. Τα λάθη τα οποία δεν είναι δυνατόν να διορθωθούν, αναγνωρίστηκαν όμως από το Επίπεδο Δικτύου αναφέρονται στις οντότητες-μεταφοράς. Η ειδοποίηση για λάθη μπορεί να οδηγήσει ή όχι σε τερματισμό της σύνδεσης-δικτύου, σύμφωνα με τη προδιαγραφή της συγκεκριμένης υπηρεσίας-δικτύου.

Το Επίπεδο Δικτύου μπορεί να παρέχει παράδοση μονάδων-δεδομένων-υπηρεσίας-δικτύου με τη σωστή τους σειρά εφόσον αυτό ζητηθεί από τις οντότητες-μεταφοράς.

Μια οντότητα-μεταφοράς η οποία λαμβάνει στο ένα άκρο μιας σύνδεσης-δικτύου μπορεί να σταματήσει την υπηρεσία-δικτύου, ώστε να μην στέλνει μονάδες-δεδομένων-υπηρεσίας-δικτύου μέσα από κάποιο σημείο-πρόσβασης-υπηρεσίας. Η συνθήκη αυτή ελέγχου ροής μπορεί να διαδοθεί ή όχι μέσα από το δίκτυο στο άλλο άκρο της σύνδεσης δικτύου και να αντικατοπτριστεί στην οντότητα-μεταφοράς που μεταδίδει, ανάλογα με την προδιαγραφή της συγκεκριμένης υπηρεσίας-δικτύου.

Η μεταφορά επειγόντων μονάδων-δεδομένων-υπηρεσίας-δικτύου είναι προαιρετική και παρέχει ένα πρόσθετο τρόπο ανταλλαγής πληροφορίας μέσα από μια σύνδεση-δικτύου. Η μεταφορά επειγόντων μονάδων-δεδομένων-υπηρεσίας-δικτύου υπόκειται σε ένα διαφορετικό σύνολο από χαρακτηριστικά υπηρεσιών-δικτύου και σε ξεχωριστό έλεγχο ροής. Το μέγιστο μέγεθος των επειγόντων μονάδων-δεδομένων-υπηρεσίας-δικτύου είναι περιορισμένο.

Η υπηρεσία αρχικοποίησης είναι προαιρετική και όταν καλείται αναγκάζει το Επίπεδο Δικτύου να απορρίψει όλες τις μονάδες-δεδομένων-υπηρεσίας-δικτύου που βρίσκονται σε μεταφορά σε μια σύνδεση-δικτύου και να ειδοποιήσει την οντότητα-μεταφοράς στο άλλο άκρο της σύνδεσης-δικτύου ότι μια αρχικοποίηση έχει συμβεί.

Μια οντότητα-μεταφοράς μπορεί να ζητήσει κατάργηση της σύνδεσης-δικτύου. Η υπηρεσία-δικτύου δεν εγγυάται την παράδοση των δεδομένων που πριν την αίτηση κατάργησης βρίσκονταν σε μεταφορά. Η σύνδεση-δικτύου λύεται ανεξάρτητα της πράξης που θα πραγματοποιήσει η οντότητα-μεταφοράς που συμμετείχε στην επικοινωνία.

Μια οντότητα μεταφοράς μπορεί να επιβεβαιώσει την λήψη δεδομένων πάνω από σύνδεση-δικτύου. Η χρήση της υπηρεσίας επιβεβαίωσης παραλαβής πρέπει να συμφωνηθεί από τους δύο χρήστες της σύνδεσης δικτύου κατά την εγκατάσταση της σύνδεσης. Η υπηρεσία είναι προαιρετική και είναι πιθανών να μην είναι πάντοτε διαθέσιμη.

Οι λειτουργίες του Επιπέδου Δικτύου παρέχουν μια ευρεία ποικιλία από διαρθρώσεις που υποστηρίζουν συνδέσεις-δικτύου και οι οποίες μπορεί να μεταβάλλονται από συνδέσεις-δικτύου υποστηριζόμενες από διαρθρώσεις σημείου-προς-σημείο, έως συνδέσεις-δικτύου υποστηριζόμενες από πολύπλοκους συνδυασμούς υποδικτύων με διαφορετικά χαρακτηριστικά.

Οι λειτουργίες Επιπέδου Δικτύου που ακολουθούν περιγράφονται παρακάτω:

- α) δρομολόγηση και αναμετάδοση,
- β) συνδέσεις-δικτύου,
- γ) πολυπλεξία συνδέσεων-δικτύου,
- δ) κατάτμηση και ένωση,

- ε) αναγνώριση λαθών,
- στ) επαναφορά από λάθη,
- ζ) διαδοχή,
- η) έλεγχος ροής,
- θ) μεταφορά επειγόντων δεδομένων,
- ι) αρχικοποίηση,
- κ) επιλογή υπηρεσίας, και
- λ) διαχείριση επιπέδου δικτύου.

Οι συνδέσεις-δικτύου παρέχονται από οντότητες-δικτύου σε τελικά ανοικτά συστήματα αλλά μπορεί να συμπεριλάβουν ενδιάμεσα ανοικτά συστήματα τα οποία θα λειτουργήσουν σαν αναμεταδότες. Τα ενδιάμεσα ανοικτά συστήματα αυτά είναι δυνατό να διασυνδέσουν συνδέσεις-υποδικτύου, συνδέσεις-σύνδεσης-δεδομένων, και κυκλώματα δεδομένων. Οι λειτουργίες δρομολόγησης καθορίζουν ένα κατάλληλο δρόμο μεταξύ διευθύνσεων-δικτύου. Προκειμένου να καθοριστεί ο τελικός τρόπος επικοινωνίας, είναι δυνατό το Επίπεδο Δικτύου να χρησιμοποιήσει τις υπηρεσίες του Επιπέδου Σύνδεσης Δεδομένων προκειμένου να ελέγχει τη διασύνδεση κυκλωμάτων δεδομένων.

Ο έλεγχος της διασύνδεσης κυκλωμάτων δεδομένων (τα οποία βρίσκονται στο Φυσικό Επίπεδο) από το Επίπεδο Δικτύου απαιτεί αλληλεπίδραση μεταξύ μιας οντότητας-δικτύου και μιας φυσικής-οντότητας στο ίδιο ανοικτό σύστημα. Από την στιγμή που το Μοντέλο Αναφοράς επιτρέπει την άμεση αλληλεπίδραση μονάχα σε οντότητες που ανήκουν σε διαδοχικά επίπεδα, η οντότητα-δικτύου δεν μπορεί να αλληλεπιδράσει άμεσα με τη φυσική-οντότητα. Η αλληλεπίδραση αυτή λοιπόν περιγράφεται μέσω του Επιπέδου Σύνδεσης Δεδομένων το οποίο μεσολαβεί για να μεταφερθεί με διαφανή τρόπο η αλληλεπίδραση από το Επίπεδο Δικτύου στο Φυσικό Επίπεδο.

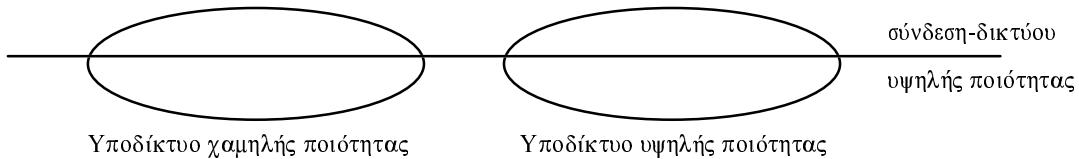
Εντούτοις, όταν τα υποδίκτυα χρησιμοποιούν πρωτόκολλα πρόσβασης που υποστηρίζουν όλη τη λειτουργικότητα της υπηρεσίας δικτύου OSI, δεν απαιτείται χωρισμός σε υποεπίπεδα στο Επίπεδο Δικτύου.

Μια σύνδεση-δικτύου μπορεί επίσης να προσφερθεί σαν μια σειρά από συνδέσεις-υποδικτύων, π.χ. χρησιμοποιώντας διάφορα ανεξάρτητα υποδίκτυα τα οποία έχουν τις ίδιες ή διαφορετικές δυνατότητες προσφοράς υπηρεσιών. Κάθε τέλος μιας σύνδεσης-υποδικτύου μπορεί να λειτουργεί με ένα διαφορετικό πρωτόκολλο υποδικτύου.

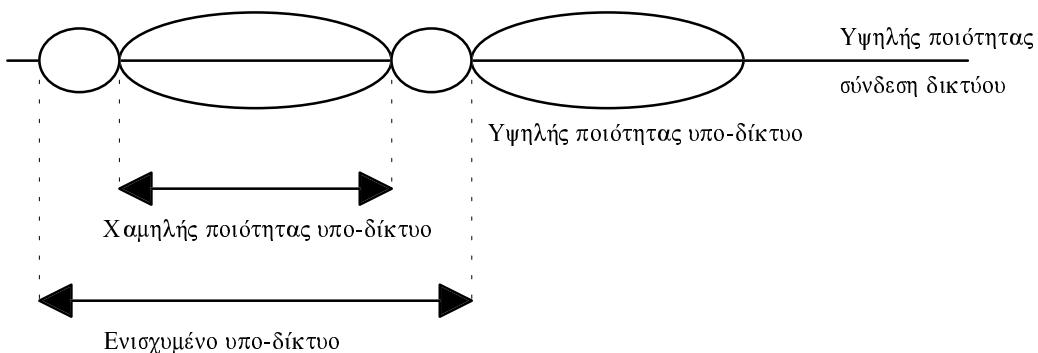
Η διασύνδεση ενός ζευγαριού από υποδίκτυα διαφορετικών ποιοτήτων μπορεί να πραγματοποιηθεί με δύο τρόπους. Για να τους δείξουμε αυτούς, θεωρείστε ένα ζευγάρι υποδίκτυα, ένα υψηλής ποιότητας και ένα χαμηλής ποιότητας:

- α) Τα δύο υποδίκτυα διασύνδεονται όπως είναι. Η ποιότητα των διασυνδεδεμένων δικτύων δεν ξεπερνά αυτή του υποδικτύου χαμηλής ποιότητας (βλ. Σχήμα 3.9).
- β) Το υποδίκτυο χαμηλής ποιότητας αναβαθμίζεται στην ποιότητα του άλλου υποδικτύου και τα υποδίκτυα διασυνδέονται στην συνέχεια. Η ποιότητα των διασυνδεδεμένων δικτύων είναι περίπου η ίδια με αυτή του υποδικτύου υψηλής ποιότητας.

Η τελική επιλογή μεταξύ των δύο αυτών εναλλακτικών λύσεων εξαρτάται από το βαθμό διαφοράς στην ποιότητα, το κόστος της αναβάθμισης, και άλλους οικονομικούς παράγοντες.



**Σχήμα 3.9 - Λιασύνδεση ενός υποδικτύου χαμηλής ποιότητας με υποδίκτυο υψηλής ποιότητας**



**Σχήμα 3.10 - Λιασύνδεση ενισχυμένου υποδικτύου χαμηλής ποιότητας και ενός υποδικτύου υψηλής ποιότητας**

Η λειτουργία αυτή μπορεί να χρησιμοποιηθεί για την πολύπλεξη συνδέσεων-δικτύου πάνω σε συνδέσεις-σύνδεσης-δεδομένων για την βελτιστοποίηση της χρήσης τους. Στην περίπτωση συνδέσεων-υποδικτύων στη σειρά, πολυπλεξία των ανεξάρτητων συνδέσεων-υποδικτύων μπορεί να πραγματοποιηθεί και πάλι για την βελτιστοποίηση της χρήσης τους.

Το Επίπεδο Δικτύου μπορεί να κόβει και/ή να ενώνει μονάδες-δεδομένων-υπηρεσίας-δικτύου προκειμένου να διευκολύνει τη μεταφορά τους. Εντούτοις, τα όρια μεταξύ των μονάδων-δεδομένων-υπηρεσίας-δικτύου διατηρούνται σε μία σύνδεση-δικτύου.

Η λειτουργία αναγνώρισης λαθών χρησιμοποιείται για να ελέγξει τη διατήρηση της ποιότητας υπηρεσίας που παρέχεται σε μια σύνδεση-δικτύου. Η αναγνώριση λαθών στο Επίπεδο Δικτύου χρησιμοποιεί την λειτουργία ειδοποίησης για λάθη του Επιπέδου Σύνδεσης Δεδομένων. Επιπρόσθετες δυνατότητες αναγνώρισης λαθών μπορεί να κριθούν απαραίτητες για την παροχή μιας δεδομένης ποιότητας υπηρεσίας.

### 3.3.6. Επίπεδο Σύνδεσης Λεδομένων

Το Επίπεδο Σύνδεσης Δεδομένων παρέχει λειτουργικά και διαδικαστικά μέσα για την εγκατάσταση, διατήρηση και κατάργηση συνδέσεων-σύνδεσης-δεδομένων μεταξύ οντοτήτων-δικτύου και τη μεταφορά μονάδων-δεδομένων-υπηρεσιών-σύνδεσης-δεδομένων. Μια σύνδεση-σύνδεσης-δεδομένων κτίζεται πάνω από μία ή περισσότερες φυσικές-συνδέσεις.

Το Επίπεδο Σύνδεσης Δεδομένων αναγνωρίζει και πιθανών διορθώνει λάθη τα οποία μπορεί να συμβούν στο Φυσικό Επίπεδο. Επιπλέον, δίνει τη δυνατότητα στο Επίπεδο Δικτύου να ελέγξει τη διασύνδεση κυκλωμάτων-δεδομένων στο Φυσικό Επίπεδο.

Οι ακόλουθες υπηρεσίες ή στοιχεία υπηρεσίας παρέχονται από το Επίπεδο Σύνδεσης Δεδομένων και περιγράφονται παρακάτω:

- α) Σύνδεση-σύνδεσης-δεδομένων,
- β) μονάδες-δεδομένων-υπηρεσίας-σύνδεσης-δεδομένων,
- γ) αναγνωριστικά-τερματισμού-σύνδεσης-σύνδεσης-δεδομένων,
- δ) διαδοχή,
- ε) ειδοποίηση για λάθη,
- στ) έλεγχος ροής, και
- ζ) παράμετροι ποιότητας υπηρεσίας.

Το Επίπεδο Σύνδεσης Δεδομένων παρέχει μία ή περισσότερες συνδέσεις-σύνδεσης-δεδομένων μεταξύ δύο οντοτήτων-δικτύου. Μια σύνδεση-σύνδεσης-δεδομένων πάντοτε δημιουργείται και καταστέφεται δυναμικά. Επιτρέπει την ανταλλαγή μονάδων-δεδομένων-υπηρεσίας-σύνδεσης-δεδομένων πάνω από μια σύνδεση-σύνδεσης-δεδομένων. Το μέγεθος των μονάδων-δεδομένων-σύνδεσης-σύνδεσης-δεδομένων μπορεί να περιοριστεί από τη σχέση μεταξύ του ρυθμού λαθών της φυσικής-σύνδεσης και της δυνατότητας αναγνώρισης λαθών του Επίπεδου Σύνδεσης Δεδομένων. Εάν είναι απαραίτητο, το Επίπεδο Σύνδεσης Δεδομένων παρέχει αναγνωριστικά-τερματισμού-σύνδεσης-σύνδεσης-δεδομένων που μπορεί να χρησιμοποιηθούν από μια οντότητα-δικτύου προκειμένου να αναγνωρίσει μια ανταποκρινόμενη οντότητα-δικτύου.

Όπου απαιτείται η ακεραιότητα της διαδοχής των μονάδων-δεδομένων-υπηρεσίας-σύνδεσης-δεδομένων αυτή διατηρείται. Ειδοποίηση παρέχεται στην οντότητα-δικτύου σε περίπτωση που αναγνωριστεί κάποιο λάθος που δεν μπορεί να διορθωθεί από το Επίπεδο Σύνδεσης Δεδομένων. Κάθε οντότητα-δικτύου μπορεί δυναμικά να ελέγξει (μέχρι το συμφωνημένο μέγιστο) το ρυθμό με τον οποίο λαμβάνει μονάδες-δεδομένων-υπηρεσίας-σύνδεσης-δεδομένων σε μια σύνδεση-σύνδεσης-δεδομένων. Ο έλεγχος αυτός μπορεί να αντικατοπτριστεί στο ρυθμό με τον οποίο το Επίπεδο Σύνδεσης Δεδομένων δέχεται μονάδες-δεδομένων-υπηρεσίας-σύνδεσης-δεδομένων σε έναν ανταποκρινόμενο τερματισμό-σύνδεσης-σύνδεσης-δεδομένων.

Παράμετροι ποιότητας υπηρεσίας μπορεί να επιλέγονται προαιρετικά. Το Επίπεδο Σύνδεσης Δεδομένων εγκαθιστά και διατηρεί μια επιλεγμένη ποιότητα υπηρεσίας για όλη τη διάρκεια της σύνδεσης-σύνδεσης-δεδομένων. Οι παράμετροι ποιότητας υπηρεσίας περιλαμβάνουν μέση τιμή του χρόνου μεταξύ δύο λαθών που αναγνωρίστηκαν αλλά που δεν ήταν δυνατό να διορθωθούν, υπολειπόμενο ρυθμό λαθών (όπου τα λάθη μπορεί να έχουν προκύψει από μεταβολές, απώλεια, διπλασιασμό, λάθος σειρά, λάθος παράδοση μονάδων-δεδομένων-υπηρεσίας-σύνδεσης-δεδομένων και άλλες αιτίες), διαθεσιμότητα υπηρεσίας, καθυστέρηση μεταφοράς, και ρυθμό απόδοσης.

Οι ακόλουθες λειτουργίες εκτελούνται από το Επίπεδο Σύνδεσης Δεδομένων:

- α) εγκατάσταση και κατάργηση σύνδεσης-σύνδεσης-δεδομένων,
- β) αντιστοιχία μονάδων-δεδομένων-υπηρεσίας-σύνδεσης-δεδομένων,
- γ) διαχωρισμός σύνδεσης-σύνδεσης-δεδομένων,
- δ) εύρεση ορίων και συγχρονισμός,
- ε) έλεγχος διαδοχής,
- στ) αναγνώριση λαθών,
- ζ) επαναφορά μετά από λάθος,
- η) έλεγχος ροής,
- θ) αναγνώριση και ανταλλαγή παραμέτρων,
- ι) έλεγχος της διασύνδεσης κυκλωμάτων-δεδομένων, και
- κ) διαχείριση Επιπέδου Σύνδεσης Δεδομένων

### 3.3.7. Επίπεδο Φυσικό

Το Φυσικό Επίπεδο παρέχει μηχανικά, ηλεκτρικά, λειτουργικά και διαδικαστικά μέσα για την ενεργοποίηση, διατήρηση και απενεργοποίηση φυσικών συνδέσεων για την μετάδοση ψηφίων μεταξύ οντοτήτων-σύνδεσης-δεδομένων. Μια φυσική-σύνδεση μπορεί να εμπειρίζει ενδιάμεσα ανοικτά συστήματα, κάθε ένα από τα οποία αναμεταδίδει τα ψηφία μέσα στο Φυσικό Επίπεδο. Οι οντότητες του Φυσικού Επιπέδου διασυνδέονται μέσω ενός φυσικού μέσου.

Οι ακόλουθες υπηρεσίες ή στοιχεία υπηρεσίας που παρέχονται από το Φυσικό Επίπεδο περιγράφονται παρακάτω:

- α) φυσικές συνδέσεις,
- β) μονάδες-δεδομένων-φυσικών-υπηρεσιών,
- γ) τερματισμοί-φυσικών-συνδέσεων,
- δ) αναγνώριση κυκλωμάτων δεδομένων,
- ε) διαδοχή,
- στ) ειδοποίηση λανθασμένης κατάστασης, και
- ζ) παράμετροι ποιότητας υπηρεσίας.

Το Φυσικό Επίπεδο παρέχει τη δυνατότητα διαφανής μετάδοσης συρμών ψηφίων μεταξύ οντοτήτων-σύνδεσης-δεδομένων μέσα από φυσικές-συνδέσεις. Ένα κύκλωμα δεδομένων είναι ένα επικοινωνιακό μονοπάτι στο φυσικό μέσο μεταξύ δύο φυσικών οντοτήτων, μαζί

με τις απαραίτητες ευκολίες στο Φυσικό Επίπεδο για την μετάδοση ψηφίων. Μια φυσική-σύνδεση μπορεί να παρέχεται μέσα από τη διασύνδεση κυκλωμάτων-δεδομένων με την χρήση λειτουργιών αναμετάδοσης στο Φυσικό Επίπεδο.

Ο έλεγχος της διασύνδεσης κυκλωμάτων-δεδομένων προσφέρεται σαν υπηρεσία σε οντότητες-σύνδεσης-δεδομένων. Μια μονάδα-δεδομένων-φυσικών-υπηρεσιών αποτελείται από ένα ψηφίο σε σειριακή μετάδοση και 'ν' ψηφία σε παράλληλη μετάδοση. Μια φυσική-σύνδεση μπορεί να επιτρέπει αμφίδρομη ή μονόδρομη μετάδοση συρμάτων ψηφίων.

Το Φυσικό Επίπεδο παρέχει αναγνωριστικά-τερματισμών-φυσικών-συνδέσεων τα οποία μπορούν να χρησιμοποιηθούν από μια οντότητα-σύνδεσης-δεδομένων για την αναγνώριση τερματισμών-φυσικών-συνδέσεων.

Μια φυσική-σύνδεση μπορεί να έχει δύο (σημείο-προς-σημείο) ή περισσότερους (πολλαπλών τερματικών σημείων) τερματισμούς-φυσικών-συνδέσεων.

Το Φυσικό Επίπεδο παρέχει αναγνωριστικά τα οποία καθορίζουν με μοναδικό τρόπο τα κυκλώματα δεδομένων μεταξύ διαδοχικών ανοικτών συστημάτων.

Το Φυσικό Επίπεδο παραδίδει τα ψηφία με την ίδια σειρά με αυτήν που του δόθηκαν.

Οντότητες-σύνδεσης-δεδομένων ειδοποιούνται για λανθασμένες καταστάσεις που ανιχνεύτηκαν στο Φυσικό Επίπεδο.

Η ποιότητα υπηρεσίας μιας φυσικής σύνδεσης παράγεται από τα κυκλώματα-δεδομένων που την αποτελούν. Η ποιότητα υπηρεσίας μπορεί να χαρακτηριστεί από:

- α) ρυθμούς λαθών, όπου τα λάθη μπορεί να προκύψουν από αλλαγές, χάσιμο, δημιουργία, ή άλλες αιτίες,
- β) διαθεσιμότητα υπηρεσίας,
- γ) ρυθμός μετάδοσης, και
- δ) καθυστέρηση μεταφοράς.

Οι παρακάτω λειτουργίες εκτελούνται στο Φυσικό Επίπεδο:

- α) ενεργοποίηση και απενεργοποίηση φυσικών-συνδέσεων,
- β) μετάδοση μονάδων-δεδομένων-φυσικών-υπηρεσιών, και
- γ) διαχείριση Φυσικού Επιπέδου.

Οι λειτουργίες αυτές παρέχουν τα μέσα για την ενεργοποίηση και την απενεργοποίηση φυσικών-συνδέσεων μεταξύ δύο οντοτήτων-σύνδεσης-δεδομένων μετά από απαίτηση του Επιπέδου Σύνδεσης Δεδομένων. Αυτές περιλαμβάνουν μια λειτουργία αναμετάδοσης η οποία παρέχει τα μέσα για τη διασύνδεση κυκλωμάτων-δεδομένων.

Η μετάδοση μονάδων-δεδομένων-φυσικών-υπηρεσιών (π.χ. ψηφίων) μπορεί να είναι σύγχρονη ή ασύγχρονη.

Τα μηχανικά, ηλεκτρομαγνητικά, και άλλα χαρακτηριστικά που εξαρτώνται από το μέσο, των συνδέσεων φυσικών μέσων ορίζονται στα σύνορα μεταξύ του Φυσικού Επιπέδου και του φυσικού μέσου. Τα χαρακτηριστικά αυτά καθορίζονται σε άλλα πρότυπα.

### 3.4. Διαφορές με τα πρωτόκολλα TCP/IP

Υπάρχουν σημαντικές διαφορές μεταξύ του TCP/IP σχήματος διαστρωμάτωσης και αυτού των πρωτοκόλλων ISO/OSI.

Μια πρώτη βασική διαφορά μεταξύ των TCP/IP και των ISO/OSI πρωτοκόλλων βρίσκεται στον τρόπο διευθέτησης του προβλήματος της αξιόπιστης μεταφοράς δεδομένων. Τα πρωτόκολλα X.25 και γενικότερα τα πρωτόκολλα ISO/OSI αναγνωρίζουν και διορθώνουν λάθη σ' όλα τα επίπεδα του OSI μοντέλου αναφοράς πρωτοκόλλων. Στο επίπεδο σύνδεσης δεδομένων, πολύπλοκα πρωτόκολλα εγγυούνται την ασφαλή μεταφορά δεδομένων μεταξύ του τερματικού σταθμού εργασίας (Data Terminal Equipment, DTE) και του διακόπτη μεταγωγής πακέτων (Data Circuit Termination Equipment, DTE) με τον οποίο αυτό συνδέεται. Ειδικά πεδία ελέγχου σφάλματος (Frame Checking Sequences, FCS) συνοδεύουν κάθε κομμάτι πληροφορίας και επιπλέον ο παραλήπτης επιβεβαιώνει κάθε κομμάτι πληροφορίας που παραλαμβάνει. Το πρωτόκολλο του επιπέδου σύνδεσης δεδομένων επιπρόσθετα με κατάλληλους μηχανισμούς λήξης χρονιστών και αλγορίθμους επαναμετάδοσης διασφαλίζει την μεταφορά δεδομένων ακόμα και σε περίπτωση σφάλματος των μηχανημάτων.

Πάνω από το δεύτερο επίπεδο, το επίπεδο δικτύου και το επίπεδο μεταφοράς με τη δική τους πολυπλοκότητα, και ειδικότερα με την από άκρη σε άκρη επιβεβαίωση του επιπέδου μεταφοράς προσθέτουν ακόμα μεγαλύτερη αξιοπιστία στην μεταφορά των δεδομένων.

Αντίθετα με το παραπάνω σχήμα, τα πρωτόκολλα TCP/IP παρουσιάζουν μια διαφορετική νοοτροπία, στηρίζοντας την αρχιτεκτονική τους στην ιδέα: η αξιοπιστία είναι ένα πρόβλημα που πρέπει να λύσουν οι δύο κόμβοι που επικοινωνούν (από άκρη σε άκρη).

Σύμφωνα με αυτή την ιδέα η αρχιτεκτονική των πρωτοκόλλων TCP/IP διαμορφώνεται ως εξής: το Internet προσφέρει υπηρεσίες μεταφοράς φορτίου, χωρίς όμως να ενδιαφέρεται για το αν οι ενδιάμεσοι κόμβοι χάνουν ή καταστρέφουν δεδομένα και χωρίς να προσπαθεί να επανορθώσει μετά από τέτοιους είδους προβλήματα. Στην πραγματικότητα τα επίπεδα δικτύου (Internet Protocol, IP) και γραμμής (Medium Access Control, MAC) προσφέρουν ελάχιστη ή και καθόλου αξιοπιστία στην μεταφορά δεδομένων. Βέβαια με δεδομένο τα παραπάνω το πρωτόκολλο μεταφοράς (Transport Control Protocol, TCP) πρέπει να έχει μεγάλες δυνατότητες ανίχνευσης σφαλμάτων, λάθους στη διαδοχή, διπλασιασμών κ.ά.

Το αποτέλεσμα και ταυτόχρονα το πλεονέχτημα αυτής της αρχιτεκτονικής είναι η ευκολότερη κατανόηση των πρωτοκόλλων TCP/IP και η καλύτερη υλοποίησή τους. Χάρη σ' αυτήν την ελευθερία οι ενδιάμεσοι δρομολογητές μπορούν να απορρίπτουν datagrams με λάθη, να απορρίπτουν datagrams που αδυνατούν να δρομολόγησουν, να απορρίπτουν datagrams όταν ο ρυθμός άφιξής τους υπερβαίνει τις δυνατότητες επεξεργασίας του δρομολογητή. Τέλος ένας δρομολογητής μπορεί να επιλέγει τα μονοπάτια με το λιγότερο φορτίο για τη δρομολόγηση, χωρίς να ειδοποιεί πηγή ή προορισμό (σύγκριση μεταφοράς δεδομένων με σύνδεση και μεταφοράς δεδομένων χωρίς σύνδεση).

Η ύπαρξη των μη αξιόπιστων αυτών ενδιάμεσων συνδέσμων συνεπάγεται το χάσιμο κάποιων datagrams. Η διάγνωση και η διόρθωση τέτοιων ανωμαλιών πραγματοποιείται από άκρη σε άκρη στους σταθμούς γέννησης και παραλαβής των

datagrams. Το λογισμικό υλοποίησης του επιπέδου μεταφοράς χρησιμοποιώντας χρονιστές, επιβεβαιώσεις, ελέγχους σφαλμάτων και άλλα πολύπλοκα πρωτόκολλα προσφέρει την απαραίτητη αξιοπιστία στην υπηρεσία μεταφοράς δεδομένων.

Ένα ακόμα υπέρ των πρωτόκολλων TCP/IP είναι η παρουσία του πρωτοκόλλου UDP, το οποίο επεκτείνει ακόμα ψηλότερα τις μη αξιόπιστες υπηρεσίες του πρωτοκόλλου IP για εφαρμογές, που πολύ απλά δεν απαιτούν αξιοπιστία.

Άλλη διαφορά μεταξύ των πρωτόκολλων TCP/IP και των πρωτοκόλλων ISO/OSI προκύπτει αν αναλογιστεί κανείς το που ασκείται ο έλεγχος του δικτύου. Έτσι στα πρωτόκολλα X.25 ο ιδιοκτήτης του δικτύου ελέγχει όλο το δίκτυο, την πρόσβαση σ' αυτό, το φορτίο που διέρχεται από αυτό, ενώ ταυτόχρονα κρατάει αρχεία για λογιστικούς λόγους. Είναι γνωστό ότι τα πρωτόκολλα X.25 είναι η συνηθισμένη λύση για την υλοποίηση δημοσίων δικτύων, όπου είναι αναγκαίος ένας τέτοιος έλεγχος. Αντίθετα στα δίκτυα TCP/IP οι τερματικοί σταθμοί (χρήστες) συμμετέχουν σε όλες τις διαδικασίες, όπως δρομολόγηση, χειρισμός μηνυμάτων ICMP, κ.ά.

Άλλη σημαντική διαφορά, που μπορούμε να παρατηρήσουμε, είναι ότι τα πρωτόκολλα εφαρμογής (ή οι διεργασίες στο επίπεδο εφαρμογής) στα πρωτόκολλα TCP/IP επικοινωνούν απευθείας με το επίπεδο μεταφοράς, όπου χρησιμοποιούν τις υπηρεσίες των πρωτοκόλλων TCP και UDP.

Αντίθετα στα πρωτόκολλα ISO/OSI οι διεργασίες που τρέχουν στο επίπεδο εφαρμογής χρησιμοποιούν τις υπηρεσίες του επιπέδου μεταφοράς, διαμέσου όμως των υπηρεσιών πολλών άλλων οντοτήτων στα επίπεδα εφαρμογής, παρουσίασης και συνόδου. Τα επίπεδα παρουσίασης και συνόδου θα παρουσιαστούν παρακάτω. Θα συμπληρώσουμε στο σημείο αυτό ότι, μια διεργασία στο επίπεδο εφαρμογής μπορεί να χρησιμοποιήσει και τις υπηρεσίες άλλων εφαρμογών του ίδιου επιπέδου. Τέτοιες εφαρμογές που προσφέρουν συχνά χρησιμοποιούμενες υπηρεσίες θα εξετάσουμε επίσης παρακάτω.

Από τα παραπάνω συμπεραίνουμε ότι, στα πρωτόκολλα TCP/IP η λειτουργικότητα των επιπέδων παρουσίασης, συνόδου και των συχνά χρησιμοποιούμενων πρωτόκολλων του επιπέδου εφαρμογής εμπεριέχεται κάθε φορά στην συγκεκριμένη εφαρμογή. Επίσης συμπεραίνουμε ότι, στα πρωτόκολλα ISO/OSI η στοίβα OSI έχει μια σειρά από πρωτόκολλα των οποίων η λειτουργία είναι προσανατολισμένη προς το επίπεδο εφαρμογής.

### **3.5. Ειδικά στοιχεία υπηρεσίας στο επίπεδο εφαρμογής: FTAM, MHS, το OSI Directory**

Πέρα από τα πρωτόκολλα υποστήρικτης εφαρμογών, που συναντήσαμε στην προηγούμενη παράγραφο (είτε του επιπέδου συνόδου, είτε του επιπέδου παρουσίασης, είτε του επιπέδου εφαρμογής), επιπλέον μία πλήρη σειρά από στοιχεία υπηρεσίας στο επίπεδο εφαρμογής (Application Service Elements, ASEs) έχουν οριστεί και προσφέρουν υπηρεσίες στις εφαρμογές του χρήστη. Τέτοια ASEs είναι τα εξής:

- **Vitual Terminal (VT)**, το οποίο προσφέρει ανάλογες υπηρεσίες με αυτές του πρωτοκόλλου TELNET των πρωτοκόλλων TCP/IP.
- **File Transfer Access and Management (FTAM)**, το οποίο προσφέρει ανάλογες υπηρεσίες με το πρωτόκολλο FTP.

- **Message Oriented Text Interchange Standard (MOTIS)**, το οποίο προσφέρει υπηρεσίες ανάλογες με του πρωτοκόλλου SMTP.
- **Common Management Information Protocol (CMIP)**, το οποίο προσφέρει ανάλογες υπηρεσίες με του πρωτοκόλλου SNMP.
- **Job Transfer and Manipulation (JTM)**, το οποίο προσφέρει δυνατότητες μεταφοράς εργασίας επεξεργασίας σε άλλη διεργασία και.
- **Manufacturing Messaging Service (MMS)**, το οποίο προσφέρει υπηρεσίες μεταφοράς μηνυμάτων μεταξύ του ελεγκτή μηχανήματος και των διαφόρων διεργασιών που ελέγχουν μηχανήματα (π.χ. robots) της γραμμής παραγωγής.

Παρακάτω θα εξετάσουμε κάποια από αυτά.

#### 1. Το πρωτόκολλο μεταφοράς αρχείων και διαχείρισης πρόσβασης (File Transfer and Access Management, FTAM)

Το FTAM παρέχει υπηρεσίες διαχείρισης αρχείων στις εφαρμογές OSI. Συγκεκριμένα, το στοιχείο υπηρεσίας αυτό, παρέχει πολύ περισσότερες υπηρεσίες από μία απλή μεταφορά αρχείων, για παράδειγμα το FTAM μπορεί να χρησιμοποιηθεί για μεταφορά αρχείων, για πρόσβαση αρχείων από σταθμούς εργασίας χωρίς δίσκο, για ειδικές εφαρμογές όπως εκτυπώσεις, και για προσβάσεις σε βάσεις δεδομένων. Η ανάγκη για κάποιο στοιχείο υπηρεσίας, όπως το FTAM, προέρχεται από την χρησιμοποίηση διαφορετικών τρόπων από τα λειτουργικά συστήματα για τον χειρισμό των αρχείων. Το FTAM δέχεται ένα νοητό μοντέλο αποθήκευσης αρχείων και σ' αυτό απεικονίζει κάθε άλλο ιδιαίτερο τρόπο, που μπορεί να συναντήσει σε κάποιο λειτουργικό σύστημα.

Το νοητό μοντέλο αποθήκευσης αρχείων, είναι μια οντότητα με την οποία κάποιος χρήστης μπορεί να επικοινωνεί (να έχει μια συσχέτιση). Για την ακρίβεια περισσότεροι από ένας χρήστες μπορούν κάθε χρονική στιγμή να έχουν μια συσχέτιση με μια οντότητα αποθήκευσης αρχείων. Μία τέτοια οντότητα μπορεί να αποτελείται από κάποιο αριθμό αρχείων, κάθε ένα από τα οποία έχει κάποια χαρακτηριστικά, όπως:

- Κάποιο όνομα, το οποίο το ξεχωρίζει από τα υπόλοιπα.
- Τις επιτρεπόμενες σ' αυτό λειτουργίες (read, insert, replace, κ.ά.).
- Έλεγχο πρόσβασης σ' αυτό (read-only, read-write, κ.ά.).
- Μέγεθος του αρχείου.
- Τρόπος παρουσιάσης των περιεχομένων.
- Αναγνωριστικό του δημιουργού του αρχείου.
- Ημέρα και ώρα δημιουργίας.
- Αναγνωριστικό του τελευταίου χρήστη του αρχείου.
- Ημέρα και ώρα της τελευταίας πρόσβασης.
- Είδος περιεχομένων.
- Κλειδί κωδικοποίησης.

Υπάρχουν τρεις κύριες δομές αρχείων που χρησιμοποιούνται: unstructured, flat και hierarchical.

Μετά από αυτή την πολλή σύντομη παρουσίαση του νοητού μοντέλου αποθήκευσης αρχείων, μπορούμε να προχωρήσουμε στην εξέταση των υπηρεσιών που προσφέρει το FTAM. Οι υπηρεσίες είναι οργανωμένες σε regimes κάθε ένα από τα οποία φωλιάζει μέσα σ' ένα άλλο. Δηλ. όσο οδηγούμαστε σε πιο μεγαλύτερες λεπτομερείες, εισαγόμαστε σε ένα νέο regime. Αυτά είναι: Application connection (association), File selection, File access και Data transfer. Κάθε ένα από αυτά παρέχει τις ανάλογες υπηρεσίες. Περισσότερες πληροφορίες υπάρχουν στα ανάλογα πρότυπα του ISO.

## 2. Το σύστημα χειρισμού μηνυμάτων (Message Handling System, MHS)

MOTIS, είναι η υπηρεσία ταχυδρομείου του ISO και προσφέρει ανάλογες υπηρεσίες με το πρωτόκολλο SMTP της κοινότητας TCP/IP. Στην ουσία, το MOTIS είναι περισσότερο ένα ολοκληρωμένο σύστημα μεταφοράς μηνυμάτων (ταχυδρομείου), παρά ένα απλό πρωτόκολλο. Είναι επίσης γνωστό σαν το ISO Message Handling System (MHS) και είναι βασισμένο στην υπηρεσία χειρισμού μηνυμάτων X.400, που είχε οριστεί από την CCITT.

Η σύσταση X.400 της CCITT ορίστηκε για να παρέχει μια διεθνή υπηρεσία χειρισμού μηνυμάτων, παρόμοια με αυτή που προσφέρεται σήμερα από τα ταχυδρομεία κάθε χώρας.

Η πρόσβαση του χρήστη στο MHS γίνεται με κάποιο τερματικό - συνήθως κάποιον προσωπικό υπολογιστή - που του δίνει τη δυνατότητα να δημιουργήσει κάποιο μήνυμα ή να αποθηκεύσει άλλα μηνύματα. Επίσης του δίνει τη δυνατότητα εξέτασης όλων των εισερχόμενων μηνυμάτων. Αυτή είναι μια από τις λειτουργίες του user agent (UA).

Για την τελική μεταφορά του μηνύματος, πολλές άλλες λειτουργίες χρησιμοποιούνται, όπως λειτουργίες παρουσίασης του περιεχομένου του μηνύματος, υπηρεσίες παράδοσης του μηνύματος, υπηρεσίες μετάδοσης του μηνύματος, αποθήκευσης, κ.ά.

## 3.6. Περιβάλλον ανάπτυξης εφαρμογών ISO/OSI και TCP/IP εφαρμογών, ISODE

Το ISO Development Environment είναι εργαλείο έρευνας πάνω στα υψηλά επίπεδα του μοντέλου αναφοράς πρωτοκόλλων OSI. Παρ' όλα αυτά, πολύ συχνά χρησιμοποιείται σαν πλατφόρμα ανάπτυξης υπηρεσιών OSI.

Στα δύο πρώτα μέρη παρουσιάζονται οι υπηρεσίες των υψηλότερων επιπέδων (από πάνω προς τα κάτω) καθώς και οι βάσεις δεδομένων, που χρησιμοποιούνται από τις υπηρεσίες αυτές. Στο τρίτο μέρος παρουσιάζονται ορισμένες εφαρμογές, που αναπτύχθηκαν με την βοήθεια των παραπάνω υπηρεσιών. Στο τέταρτο μέρος περιγράφει κάποιες δυνατότητες του ISODE για ανάπτυξη εφαρμογών περισσότερο με την βοήθεια μιας γλώσσας προγραμματισμού, παρά με κάποιο μοντέλο βασισμένο στο δίκτυο. Στο πέμπτο μέρος αναπτύσσεται πλήρως μια ανάπτυξη του OSI directory.

### 1. Application Services.

Στο μέρος αυτό περιγράφονται διάφορες δυνατότητες, οι οποίες είναι διαθέσιμες στις εφαρμογές. Για αυτές υπάρχουν τέσσερις βιβλιοθήκες:

`libacsap(3n)` - η οποία υλοποιεί το OSI association control service (ASC)

`librosap(3n)` - η οποία υλοποιεί διάφορους μηχανισμούς για το OSI remote operations service (ROS)

`librtsap(3n)` - η οποία υλοποιεί το OSI remote operations service (ROS)

`libsap(3)` - η οποία υλοποιεί το OSI abstract syntax και transfer mechanisms

### 2. Underlying Services.

Στο μέρος αυτό περιγράφονται οι υπηρεσίες, οι οποίες είναι διαθέσιμες στις εφαρμογές. Για αυτές υπάρχουν τέσσερις βιβλιοθήκες:

`libpsap2(3n)` - η οποία υλοποιεί τις υπηρεσίες του επιπέδου παρουσίασης

`libssap(3n)` - η οποία υλοποιεί τις υπηρεσίες του επιπέδου συνόδου

`libtsap(3n)` - η οποία υλοποιεί το σημείο πρόσβασης υπηρεσιών στο επίπεδο μεταφοράς

### 3. Applications.

Στο μέρος αυτό περιγράφονται εφαρμογές, οι οποίες αναπτύχθηκαν με την βοήθεια του ISODE. Τέτοιες εφαρμογές είναι:

- Μια υλοποίηση του ISO FTAM (File Transfer Access and Management), το οποίο τρέχει σε Berkeley ή AT&T UNIX. Η υλοποίηση περιλαμβάνει μόνο τέσσερες κύριες υπηρεσίες (μεταφορά αρχείων κειμένου, δυαδικών αρχείων, παρουσίαση λιστών από αρχεία και διαχείριση αρχείων), αλλά οι υπηρεσίες είναι ολοκληρωμένες.
- Μια υλοποίηση μιας διασύνδεσης μεταξύ FTAM και FTP, η οποία τρέχει σε Berkeley UNIX.
- Μια υλοποίηση του ISO VT (Virtual Terminal), η οποία τρέχει σε Berkeley UNIX.
- Μια υλοποίηση κάποιων "μικρών υπηρεσιών", που χρειάζονται για debugging και διασκέδαση.
- Μια υλοποίηση μιας υπηρεσίας τράπεζας εικόνων (image database service).

#### 4. The Applications Cookbook.

Στο μέρος αυτό περιγράφονται κάποιες έτοιμο interface για εφαρμογές, το οποίο χρησιμοποιεί remote operations. Αποτελείται από τρία προγράμματα και μια βιβλιοθήκη:

**rosy(1)** - το οποίο είναι ένα εργαλείο δημιουργίας προγραμμάτων, που δημιουργεί τον C κώδικα, ο οποίος χρειάζεται για τη δημιουργία κατανεμημένων εφαρμογών που χρησιμοποιούν RPC (Remote Procedures Calls)\*.

**posy(1)** - η οποία είναι μια γεννήτρια δομών για προδιαγραφές σε ASN.1

**pergy(1)** - η οποία είναι μια γεννήτρια, η οποία διαβάζει προδιαγραφές για μία εφαρμογή και παράγει ένα κομμάτι προγράμματος, το οποίο σχηματίζει ή αναγνωρίζει δομές (τα application PDUs) μέσω των οποίων επικοινωνείη εφαρμογή αυτή.

**pepsy(1)** - η οποία είναι μια νεώτερη γεννήτρια, η οποία θα αντικαταστήσει τις **rosy** και **posy** σε μια μελλοντική έκδοση του ISODE.

**librosy(3n)** - η οποία είναι μια βιβλιοθήκη για εφαρμογές, που χρησιμοποιούν αυτό το κατανεμημένο πλαίσιο αναφοράς.

#### 5. QUIPU.

Στο μέρος αυτός περιγράφονται διάφορα προγράμματα και μια βιβλιοθήκη για υλοποίηση του ISO directory. Αυτές είναι:

**quiρu(8c)** - ο οποίος είναι ένας directory system agent (DSA)

**dish(1c)** - η οποία είναι μια οικογένεια προγραμμάτων, τα οποία είναι ένα σύνολο από directory shell εντολές.

**libdsap(3n)** - η οποία είναι μια βιβλιοθήκη για εφαρμογές, που χρησιμοποιούν το directory.

Το ISODE περιέχει και μια υλοποίηση του πλαισίου διαχείρισης TCP/IP δικτύων. Το ISODE SNMP περιέχει μια πλήρη υλοποίηση ενός SNMP agent, δεν περιλαμβάνει όμως κάποια υλοποίηση ενός διαχειριστικού συστήματος. Περιέχει όμως κάποια εργαλεία για γρήγορη ανάπτυξη διαχειριστικών εφαρμογών. Για παράδειγμα, υπάρχει το πρόγραμμα snmpi, το οποίο υλοποιεί SNMP λειτουργίες (εκτός της trap).

```
% snmpi -a 192.33.4.21 -c secret
snmpi> get sysUpTime.0
sysUpTime.0=45366736 (5 days, 6 hours, 1 minutes, 7.36
seconds)
snmpi>
```

---

\* Οι προγραμματιστές συχνά αναφέρουν σε τέτοιες γεννήτριες σαν stub generators. Βλ. και D.Comer, D.Stevens, *Internetworking with TCP/IP, Volume III, Client - Server Programming and Applications*, Prentice Hall, 1993, σσ. 255-266, Chapter 21 Distributed Program Generation.

Σημειώνουμε ότι, το ISODE SNMP χρησιμοποιεί τον mosy compiler για την μετάφραση MIB ASN.1 δομών σε δομές σε C κώδικα. Για περισσότερες πληροφορίες μπορεί να δει κανείς τα [OPEN90], [ROSE91].

### 3.7. Ασκήσεις

- [1]. Εξηγήστε το στόχο του OSI μοντέλου αναφοράς πρωτοκόλλων, και περιγράψτε σύντομα τις λειτουργίες που εκτελεί κάθε επίπεδο.
- [2]. Αναφέρατε διαφορές μεταξύ των πρωτοκόλλων TCP/IP και μια πιθανής υλοποίησης του μοντέλου αναφοράς ISO/OSI.
- [3]. Εξετάστε προσεκτικά το μοντέλο διαστρωμάτωσης OSI. Πόσο καλά ένα τοπικό δίκτυο υπολογιστών, όπως το Ethernet ακολουθεί το μοντέλο αυτό;

### 3.8. Βιβλιογραφία

- [BERT87] Bertsekas D., Gallager R., *Data Networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- [HALS92] Halsall F., *Data Communications, Computer Networks and Open Systems*, 3nd Edition, Addison-Wesley, 1992.
- [OPEN90] Rose M.T., *The Open Book, A Practical Perspective on OSI*, Prentice Hall, 1990.
- [ΣΤΑΣ89] Στασινόπουλος Γ., *Ψηφιακά Συστήματα Επικοινωνιών*, Ε.Μ.Πολυτεχνείο, Τμήμα Ηλεκτρολόγων, Αθήνα 1989.
- [ΠΟΜΠΙ90] Α.Σ.Πομπόρτσης, *Τοπικά Δίκτυα Υπολογιστών*, Θεσσαλονική 1990.
- [ROSE92] Rose M. T., *The Little Black Book, Mail Bonding with OSI Directory Services*, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.
- [TANE91] Tanenbaum A.S., *Δίκτυα Υπολογιστών*, Δεύτερη Έκδοση, Prentice Hall, για την Ελληνική Εκδοση Παπασωτηρίου 1991.
- [STAL93] Stallings, W. SNMP, SMMPv2, & CMIP: The Practical Guide to Network Management Standards, Addison-Wesley Publishing Company, Incorporated, 1993

## Κεφάλαιο 4

### 4. Πρότυπα και Πρωτόκολλα Διαχείρισης Δικτύων TCP/IP

#### Περιεχόμενα του Κεφαλαίου 4

- 4.0. Εισαγωγή
- 4.1. Μοντέλο διαχείρισης δικτύων
  - 4.1.1. Διαχειριζόμενοι κόμβοι (agents)
  - 4.1.2. Σταθμοί διαχείρισης δικτύων (managers)
  - 4.1.3. Πρωτόκολλα διαχείρισης δικτύων
  - 4.1.4. Διαχείριση με πληρεξούσιους κόμβους (proxy agents)
- 4.2. Διαχείριση δικτύων TCP/IP
- 4.3. Το πρωτόκολλο SNMP
- 4.4. Λομή και αποθήκευση της διαχειριζόμενης πληροφορίας: SMI και MIB
- 4.5. Δυνατές λειτουργίες στη διαχειριζόμενη πληροφορία. Μορφή και σημασία των ανταλασσόμενων μηνυμάτων
- 4.6. Σύνταξη και κωδικοποίηση πληροφορίας (ASN.1 και BER)
- 4.7. Υλοποίηση διαχειριστικής εφαρμογής με βάση το SNMP
  - 4.7.1. Παρουσίαση της τοπολογίας του δικτύου
  - 4.7.2. Παρακολούθηση της απόδοσης του δικτύου
  - 4.7.3. Άλλες ενδείξεις
- 4.8. Πραγματοποιώντας διαχείριση επιδόσεων με το SNMP
  - 4.8.1. Υπολογισμός χρησιμοποίησης τμημάτων/συνδέσμων του δικτύου
  - 4.8.2. Διαχείριση καταστάσεων συμφόρησης
  - 4.8.3. Υπολογισμός ρυθμών και ποσοστών σφαλμάτων
  - 4.8.4. Αναγνώριση προτύπων για το φορτίο στο δίκτυο
- 4.9. Ασφάλεια: απειλές και μηχανισμοί
- 4.10. Remote Network Monitoring (RMON)
  - 4.10.1. Βασικές έννοιες
  - 4.10.2. RMON MIB
- 4.11. Το πρωτόκολλο SNMPv2 (SNMP version 2)
  - 4.11.1. Εισαγωγή
  - 4.11.2. Λομή της Πληροφορίας Διαχείρισης
  - 4.11.3. Λειτουργίες του Πρωτοκόλλου

- 4.11.4. SNMPv2 MIB
- 4.11.5. Manager-to-Manager MIB
- 4.11.6. Συνύπαρξη με το SNMP
- 4.11.7. SNMPv2: Ασφάλεια
- 4.12. Ασκήσεις
- 4.13. Βιβλιογραφία

## 4.0. Εισαγωγή

Η αρχιτεκτονική που προτείνεται και χρησιμοποιείται σήμερα για την διαχείριση τηλεπικοινωνιακών δικτύων και δικτύων υπολογιστών αποτελείται από το σύστημα διαχείρισης του δικτύου (Network Management System, NMS) ή το Σύστημα Λειτουργίας (Operation System, OS) και τα στοιχεία εκείνα των δικτύων (Network Elements, NE), τα οποία θέλουμε να διαχειριστούμε. Τέτοια NE's σε ένα δίκτυο είναι κυρίως μηχανήματα αποθήκευσης ή επεξεργασίας πληροφοριών, όπως hosts (workstations, terminal servers κ.ά.), καθώς και μηχανήματα διασύνδεσης δικτύων, όπως routers, bridges, repeaters κ.ά., στα οποία τρέχουν διαδικασίες διαχείρισης, που ονομάζονται αντιπρόσωποι διαχείρισης (agents), και είναι υπεύθυνες για την εκτέλεση των συναρτήσεων που καλούν τα συστήματα διαχείρισης. Για την μεταφορά της πληροφορίας μεταξύ των διαχειριστικών συστημάτων και των διαχειριζόμενων στοιχείων χρησιμοποιούνται κατάλληλα πρωτόκολλα μεταφοράς της πληροφορίας που αφορά τη διαχείριση. Τα πρωτόκολλα αυτά καθορίζουν με σαφήνεια τον τρόπο επικοινωνίας, τη μορφή και την σημασία των μηνυμάτων που θα ανταλλαχθούν, όπως επίσης και τον τρόπο ορισμού και περιγραφής των στοιχείων που θέλουμε να διαχειριστούμε.

Τα δύο γνωστότερα από τα πρωτόκολλα αυτά είναι το **SNMP (Simple Network Management Protocol)**, και το **CMIP (Common Management Information Protocol)**. Το πρώτο συμπληρώνεται με τις προδιαγραφές για τη δομή της πληροφορίας που αφορά την διαχείριση (*Structure of Management Information, SMI*), και τη βάση πληροφορίας διαχείρισης (*Management Information Base, MIB*) - προϊόντα της **Internet Architecture Board (IAB)**, της επιτροπής που εγκρίνει πρότυπα Request for Commens (RFCs) για την ομάδα πρωτόκολλων TCP/IP - ορίζει ένα απλό και λειτουργικό τρόπο διαχείρισης δικτύων TCP/IP. Το CMIP τυποποιήθηκε από τον Διεθνή Οργανισμό Τυποποίησης (**International Organization for Standardization, ISO**) και αποτελεί μαζί με τη γενικότερη προσέγγιση για την διαχείριση δικτύων OSI, μια μακροπρόθεσμη λύση για το πρόβλημα της διαχείρισης μεγάλων ετερογενών δικτύων (βλ. και επόμενο κεφάλαιο).

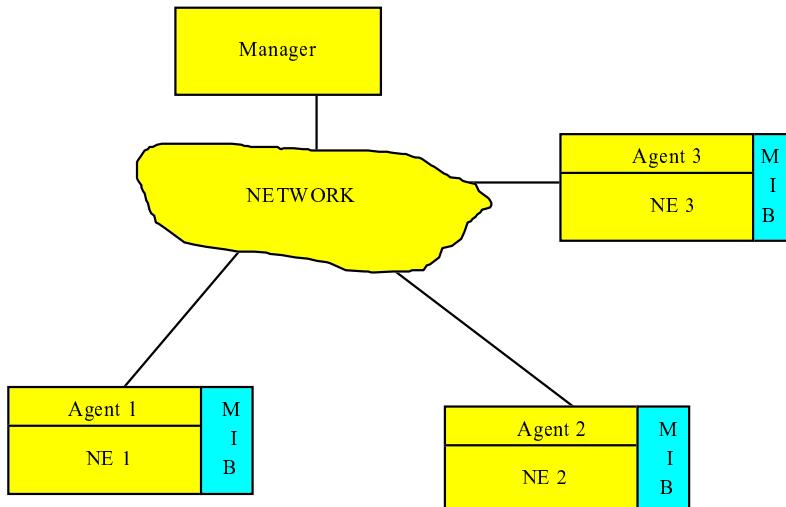
## 4.1. Μοντέλο διαχείρισης δικτύων

Το μοντέλο που χρησιμοποιείται σήμερα στην διαχείριση δικτύων, ακολουθεί την γνωστή αρχιτεκτονική του πελάτη - εξυπηρετητή (client - server) και ονομάζεται για την ειδική αυτή περίπτωση μοντέλο διαχειριστή - αντιπροσώπου (manager - agent model) (βλέπε και Σχήμα 4.1). Ο agent είναι κάποιο πρόγραμμα εξυπηρετητής (server software) που προσφέρει πληροφορία, σχετική πάντα με τη διαχείριση. Κάποιο διαχειριστικό σύστημα πρέπει να καλέσει ένα πρόγραμμα πελάτη (client software), καθορίζοντας τον εξυπηρετητή με τον οποίο θα συνδεθεί\*. Μετά τη σύνδεση έχει τη

---

\* Το πιο πρόγραμμα αποτελεί τον client και πιο τον server καθορίζεται από την πλευρά που αρχικοποιεί την επικοινωνία. Δηλαδή, το πρόγραμμα που ανοίγει κάποια συνομιλία είναι στις

δυνατότητα αποστολής αιτήσεων στον agent για την ανάκτηση πληροφορίας διαχείρισης. Πέρα από τη λειτουργία αυτή, που στην ουσία είναι μια λειτουργία παρακολούθησης (monitoring function), κάποιο NMS μπορεί επίσης να ρυθμίζει τον τρόπο λειτουργίας ενός κόμβου του δικτύου, στέλνοντας εντολές στον agent. Αυτό βέβαια σημαίνει ότι κάθε agent πέρα της πληροφορίας που μπορεί να προσφέρει, έχει την δυνατότητα επίσης να ρυθμίζει τον τρόπο λειτουργίας του κόμβου του δικτύου πάνω στον οποίο τρέχει. Σαν διαχειριζόμενα στοιχεία μπορούν να θεωρηθούν σταθμοί εργασίας (hosts), δρομολογητές (routers), γέφυρες (bridges), επαναλήπτες (repeaters), διαμορφωτές/αποδιαμορφωτές (modems) και άλλα μηχανήματα τα οποία μπορούμε να βρούμε σε ένα δίκτυο. Οι agents είναι υπεύθυνοι για την εκτέλεση των λειτουργιών διαχείρισης δικτύου, τις οποίες καλεί ο διαχειριστής. Η μεταφορά της πληροφορίας, που αφορά την διαχείριση, μεταξύ του διαχειριστή και των agents πραγματοποιείται με τη βοήθεια ειδικών πρωτοκόλλων επικοινωνίας, που ονομάζονται Πρωτόκολλα Διαχείρισης Δικτύων (Network Management Protocols, NMP's).



### Σχήμα 4.1 - Μοντέλο διαχειριστή - αντιπροσώπου

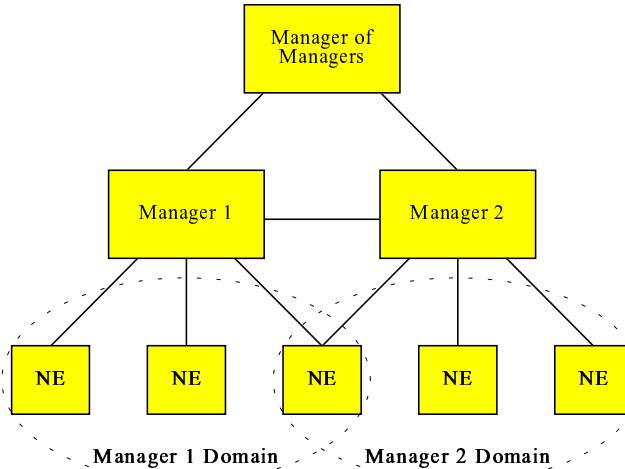
Γενικότερα ένα σύνθετο σύστημα διαχείρισης μπορεί να περιέχει πολλούς διαχειριστές και πολλούς agents. Ένας διαχειριστής μπορεί να αποτελεί διαδικασία διαχείρισης για τους δικούς του agents, ενώ να αποτελεί agent για κάποια άλλη διαδικασία διαχείρισης. Διαχειριστές και agents μ' αυτό τον τρόπο ορίζουν μια μορφή ιεραρχίας από χώρους διαχείρισης, κάθε ένας από τους οποίους αποτελεί το πεδίο δράσης κάθε NMS (βλέπε και Σχήμα 4.2). Οι χώροι αυτοί διαχείρισης ονομάζονται κατά την OSI ορολογία **management domains** δηλ. περιοχές αρμοδιότητας διαχείρισης. Πιθανές απαιτήσεις που μπορεί να οδηγήσουν στην υλοποίηση τέτοιων σύνθετων δομών διαχείρισης είναι οι παρακάτω [SMGO92]:

- Η ανάγκη διαίρεσης του διαχειριζόμενου περιβάλλοντος για την εκπλήρωση κάποιων λειτουργικών αναγκών, όπως ασφάλεια, λογιστικές ανάγκες, διαχείριση βλαβών και άλλες. Επίσης, η ανάγκη διαίρεσης του διαχειριζόμενου περιβάλλοντος για άλλους διαχειριστικούς σκοπούς, όπως σύμφωνα με γεωγραφικές, τεχνολογικές ή οργανωτικές δομές.

περισσότερες περιπτώσεις το client software. Οι τελικοί χρήστες συνήθως χρησιμοποιούν προγράμματα πελάτες προκειμένου να έχουν πρόσβαση σε κάποια υπηρεσία ενός δικτύου. Για περισσότερες λεπτομέρειες στην αρχιτεκτονική πελάτης-εξυπηρετητής, βλ. Comer E.D., Stevens L.D., "Internetworking with TCP/IP", Volume III, Client-Server Programming and Applications, Prentice Hall, Englewood Cliffs, New Jersey, 1993, και ειδικότερα Chapter 2, The Client Server Model and Software Design, pp. 9-19.

- Η ανάγκη προσωρινής ανάθεσης ή αλλαγής των ρόλων διαχειριστή - αντιπροσώπου για καθένα από τους παραπάνω λόγους και για μια συγκεκριμένη συλλογή διαχειριζόμενων αντικειμένων.
- Η ανάγκη εξάσκησης ελέγχου με ενιαίο τρόπο.

Παρακάτω θα εξετάσουμε τις δομές των διαχειριζόμενων κόμβων, του συστήματος διαχείρισης δικτύων και των πρωτοκόλλων διαχείρισης δικτύων με πιο πολλή λεπτομέρεια.



**Σχήμα 4.2 - Ιεραρχία διαχειριστικών συστημάτων**

#### 4.1.1. Διαχειριζόμενοι κόμβοι (agents)

Όλοι οι διαχειριζόμενοι κόμβοι ακολουθούν ένα κοινό μοντέλο [ROSE91]. Σύμφωνα με αυτό το μοντέλο κάθε διαχειριζόμενος κόμβος:

- Υλοποιεί μια στοίβα πρωτοκόλλων, για την παροχή επικοινωνιακών υπηρεσιών στους ανάλογους χρήστες (π.χ. TCP/IP, ISO/OSI, SNA, DECnet, κ.ά.)
- Υλοποιεί κάποιο πρωτόκολλο διαχείρισης δικτύων (π.χ. SNMP ή CMIP), το οποίο αποτελεί το κρίκο σύνδεσης του διαχειριστή με το κόμβο που θέλει να διαχειριστεί.
- Τέλος, υλοποιεί ένα σχήμα αλληλεπίδρασης μεταξύ των συγκεκριμένων διαχειριζόμενων αντικειμένων του κόμβου και του πρωτοκόλλου διαχείρισης.

Η υλοποίηση του σχήματος αλληλεπίδρασης μαζί με το συγκεκριμένο πρωτόκολλο διαχείρισης, αποτελεί ουσιαστικά τον **αντιπρόσωπο διαχείρισης (agent)**. Ο κάθε agent εκτελεί ουσιαστικά δύο λειτουργίες:

- Μέσω του σχήματος αλληλεπίδρασης, επιτυγχάνει την πρόσβαση διαφόρων δομών δεδομένων, οι οποίες ανήκουν στα διάφορα πρωτόκολλα που τρέχουν στο διαχειριζόμενο κόμβο. Με τον τρόπο αυτό μπορεί να διαβάσει τις τιμές των δομών αυτών ή ακόμα και να αλλάξει τις τιμές τους.

- Μέσω του πρωτοκόλλου διαχείρισης, επικοινωνεί με κάθε υπεύθυνο διαχειριστή, προκειμένου να ανταλλάξουν πληροφορίες σχετικά με τις δομές δεδομένων αυτές.

Ονομάζουμε **Βάση Πληροφορίας Διαχείρισης (Management Information Base, MIB)** το σύνολο των παραπάνω δομών δεδομένων και κάθε μία από αυτές **διαχειριζόμενο αντικείμενο (Managed Object)**. Ένα διαχειριζόμενο αντικείμενο αποτελεί κάποια αφαίρεση ενός πραγματικού στοιχείου του δικτύου, π.χ. μια δομή δεδομένων σε κάποιο επίπεδο (από τα επτά του μοντέλου αναφοράς πρωτοκόλλων OSI), μία σύνδεση, ή ένα λειτουργικό κομμάτι του δικτύου. Η αφαίρεση αυτή αναπαριστά τις ιδιότητες του στοιχείου του δικτύου, όπως θέλει να τις δει (και για τους σκοπούς της) διαχείρισης [SMGO92]. Τυποποιημένα διαχειριζόμενα αντικείμενα καθορίζονται από διεθνείς οργανισμούς τυποποίησης. Ωστόσο, προσφέρεται στο διαχειριστή η δυνατότητα, με τη βοήθεια κατάλληλων μηχανισμών, να ορίσει τα επιθυμητά γιαυτόν διαχειριζόμενα αντικείμενα.

#### 4.1.2. Σταθμοί διαχείρισης δικτύων

Ένα **Σύστημα Διαχείρισης Δικτύων (Network Management System - NMS)**, που αποτελεί το διαχειριστή του δικτύου, αποτελείται από τα παρακάτω επιμέρους στοιχεία:

- Πρωτόκολλα επικοινωνίας, κυρίως για την παροχή επικοινωνιακών υπηρεσιών.
- Το πρωτόκολλο διαχείρισης δικτύου, υλοποιημένο στο επίπεδο εφαρμογής.
- Τέλος, από κάποιες εφαρμογές διαχείρισης δικτύων, οι οποίες χρησιμοποιούν τις υπηρεσίες του πρωτοκόλλου διαχείρισης δικτύων, προκειμένου να επιτευχθούν διάφορες λειτουργίες διαχείρισης.

Το πιο σημαντικό εδώ, είναι η υλοποίηση των διαχειριστικών εφαρμογών καθώς και του σχήματος διασύνδεσης μεταξύ αυτών και των πρωτοκόλλων διαχείρισης. Αυτό το σχήμα διαχείρισης παρέχει στο χρήστη πολλές δυνατότητες [STAM92]:

- Τη δυνατότητα να καθορίζει τις δικές του διαχειριστικές εφαρμογές, πάνω από μια κοινή αρχιτεκτονική διαχείρισης δικτύων (διαχειριστική πλατφόρμα).
- Τη δυνατότητα ταυτόχρονης εκτέλεσης διαφόρων διαχειριστικών εφαρμογών.
- Την ευκολότερη ανάπτυξη και συντήρηση του λογισμικού του συστήματος.
- Τέλος, προσφέρεται στον χρήστη μια αρχιτεκτονική διαχείρισης, που μπορεί να επεκταθεί και να προσαρμοστεί στις δικές του ειδικές ανάγκες.

#### 4.1.3. Πρωτόκολλα διαχείρισης δικτύων

Όπως αναφέρθηκε παραπάνω, τα πρωτόκολλα διαχείρισης δικτύων χρησιμοποιούνται για την ανταλλαγή πληροφορίας που αφορά τη διαχείριση, μεταξύ του διαχειριστή του δικτύου και των διαχειριζόμενων στοιχείων του δικτύου. Τα πρωτόκολλα αυτά προσφέρουν πολλές λύσεις στο πρόβλημα της διαχείρισης δικτύων. Δύο από αυτές είναι:

- Ο καθορισμός του ακριβή τρόπου επικοινωνίας μεταξύ του διαχειριστή και του αντιπροσώπου (το πρωτόκολλο επικοινωνίας των δύο οντοτήτων).
- Ο ορισμός της μορφής και της σημασίας των ανταλλασσόμενων μηνυμάτων.

To Simple Network Management Protocol (SNMP) και το Common Management Information Protocol (CMIP) είναι τα δύο πρωτόκολλα διαχείρισης που θα εξετάσουμε. Το πρώτο μαζί με τον ορισμό της Δομής της Πληροφορίας Διαχείρισης (Structure of Management Information, SMI) και τον ορισμό της αποθήκης των διαχειριζομένων αντικειμένων, της MIB, αποτελούν ένα απλό αλλά λειτουργικό τρόπο διαχείρισης TCP/IP διαδικτύων. Το δεύτερο, μαζί με τον ορισμό των υπηρεσιών, των παρεχόμενων από το Common Management Information Service Element (CMISE) και τη γενικότερη αρχιτεκτονική δικτύου OSI, αποτελεί μια μακροπρόθεσμη λύση στην διαχείριση μεγάλων ετερογενών δικτύων.

#### 4.1.4. Διαχείριση με πληρεξούσιους αντιπροσώπους

Στο σημείο αυτό θα εξετάσουμε την μορφή ειδικότερων agents διαχείρισης, οι οποίοι ονομάζονται **πληρεξούσιοι αντιπρόσωποι (proxy agents)**. Τα πρωτόκολλα διαχείρισης, τα οποία εξετάσαμε μέχρι εδώ, αποτελούν μέρος του στρώματος εφαρμογής, της γνωστής στοίβας πρωτοκόλλων του OSI. Πολλές φορές όμως στο δίκτυο, κάποιοι κόμβοι δεν υλοποιούν μια πλήρη στοίβα πρωτοκόλλων από αυτές που αναφέραμε παραπάνω. Στην περίπτωση αυτή, δεν είναι δυνατό για το διαχειριστή του δικτύου να επικοινωνήσει με τους κόμβους αυτούς. Δύο λύσεις έχουν προταθεί [CASS89]:

- Η υλοποίηση μιας **λεπτής στοίβας πρωτοκόλλων (thin stack)**, η οποία θα προσφέρει τις απολύτως απαραίτητες υπηρεσίες για την επικοινωνία με τους κόμβους αυτούς.
- Η δεύτερη λύση προτείνει την χρησιμοποίηση ενός ενδιάμεσου κόμβου, ο οποίος ονομάζεται πληρεξούσιος κόμβος (proxy node). Ο κόμβος αυτός επαναθέτει τις διαχειριστικές εντολές μέσω ενός διαφορετικού πρωτοκόλλου στο διαχειριζόμενο κόμβο.

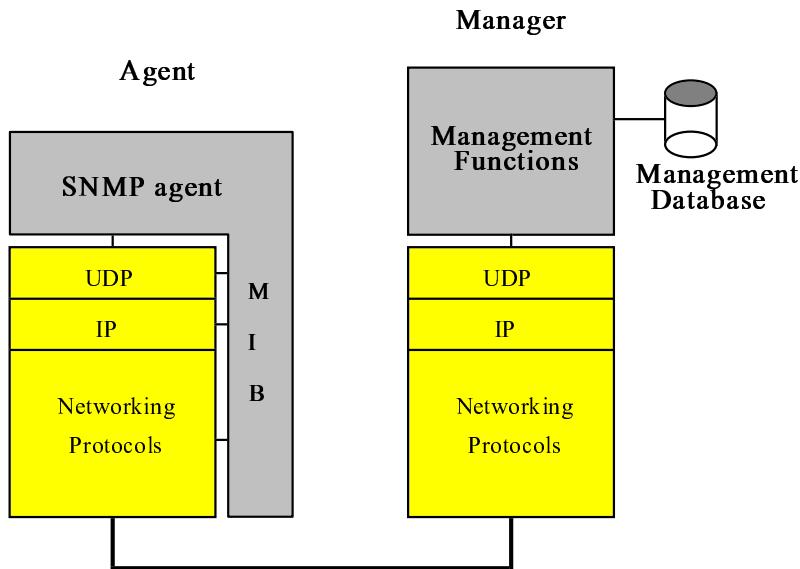
Κρίνοντας τις παραπάνω λύσεις μπορούμε να συμπεράνουμε τα παρακάτω. Η πρώτη λύση δεν θεωρείται κατάλληλη για στοιχεία δικτύων με περιορισμένες υπολογιστικές δυνατότητες, (π.χ. modems), από την στιγμή που η υλοποίηση της πρόσθετης στοίβας πρωτοκόλλων απαιτεί κάποια μνήμη. Η δεύτερη, είναι μια κατάλληλη λύση για τέτοιες περιπτώσεις.

### 4.3. Διαχείριση δικτύων TCP/IP

Η μεγάλη ανάπτυξη της τεχνολογίας των TCP/IP δικτύων οδήγησε στην ανάγκη για εργαλεία διαχείρισης των δικτύων αυτών. Οι δραστηριότητες των ερευνητικών ομάδων στην TCP/IP κοινότητα, όσο αναφορά την διαχείριση, εστίασαν την προσοχή τους στη δημιουργία προτύπων, το πιο σημαντικό από τα οποία είναι το SNMP πρωτόκολλο, η αρχιτεκτονική του οποίου φαίνεται στο Σχήμα 4.3. Παρακάτω θα εξετάσουμε το πλαίσιο διαχείρισης των TCP/IP δικτύων, καθώς και την αρχιτεκτονική που προτείνει το SNMP πρωτόκολλο.

#### 4.4. Το πρωτόκολλο SNMP

Η αρχιτεκτονική που προτείνει το SNMP πρωτόκολλο διαχείρισης ακολουθεί το μοντέλο που περιγράφαμε παραπάνω, με τους σταθμούς διαχείρισης και τα στοιχεία του δικτύου τα οποία θέλουμε να διαχειριστούμε. Κάθε agent που τρέχει έχει στην κατοχή του μια συλλογή από μεταβλητές (στιγμιότυπα αντικειμένων), όπως διευθύνσεις, τύπους interfaces, μετρητές κ.ά, των οποίων μεταβλητών οφείλει να γνωρίζει τις τιμές και να τις αποδίδει. Τα αντικείμενα αυτά είναι αφαιρέσις πραγματικών στοιχείων του δικτύου, από τα οποία άλλα έχουν ένα στιγμιότυπο και άλλα περισσότερα (όπως μια TCP σύνδεση) και οργανώνονται, σύμφωνα με το SNMP, σε ένα πίνακα.



**Σχήμα 4.3 - Μοντέλο διαχείρισης TCP/IP δικτύων**

Το σύνολο των μεταβλητών αυτών ονομάζεται MIB. Το πρωτόκολλο SNMP δίνει τη δυνατότητα σε έναν σταθμό διαχείρισης να ελέγχει ή να μεταβάλλει τις μεταβλητές της MIB ενός agent. Με τον τρόπο αυτό είναι δυνατό να παρακολουθηθεί η απόδοση και η κατάσταση ενός δικτύου, να ελεγχθούν παράμετροι που αφορούν την λειτουργία του, να αναφερθούν, αναλυθούν και να απομονωθούν σφάλματα.

Από τα παραπάνω βγάζουμε το συμπέρασμα ότι η στρατηγική διαχείρισης που υπονοείται στο SNMP, απαιτεί η παρακολούθηση της κατάστασης ενός δικτύου, για κάθε επίπεδο λεπτομέρειας, να πραγματοποιείται με αναζήτηση της κατάλληλης πληροφορίας. Ακολουθείται δηλαδή ένα **polling-based μοντέλο διαχείρισης**. Εντούτοις ένας περιορισμένος αριθμός αυτόκλητων μηνυμάτων (TRAPs), καθοδηγούν το χρονισμό και την προσοχή των ερωτήσεων. Ο αριθμός των μηνυμάτων αυτών είναι σχετικά μικρός, συμβάλοντας στην ευκολία υλοποίησης ενός agent, και μπορεί να επεκταθεί μονάχα αν αξιοποιηθεί η δυνατότητα που δίνεται στους κατασκευαστές να ορίσουν τα δικά τους TRAP μηνύματα. Είναι πιθανό ότι μια τέτοια αξιοποίηση, είναι πολύ σημαντική γιατί δίνει την δυνατότητα να παρακαμφθεί το παραπάνω polling-based μοντέλο, και να ελαττωθεί το φορτίο που “ρίχνει” η διαχείριση στο δίκτυο.

Το SNMP αποτελεί μια βελτίωση που έκανε η IAB στο πρωτόκολλο **Simple Gateway Monitoring Protocol (SGMP)**, το οποίο χρησιμοποιούταν παλαιότερα για τη παρακολούθηση IP routers. Το SNMP είναι παρόμοιο με το SGMP στην αρχιτεκτονική και στην φιλοσοφία σχεδιασμού, παρ' όλα αυτά η σύνταξή του είναι διαφορετική και έτσι τα δύο πρωτόκολλα είναι ασύμβατα. Το όλο επιχείρημα της δημιουργίας του

SNMP είχε σαν στόχο μια βραχυπρόθεσμη λύση για την διαχείριση TCP/IP δικτύων και πραγματικά πέτυχε τον στόχο του.

Για να πετύχει το στόχο αυτό, το SNMP ελαχιστοποιεί τον αριθμό και την πολυπλοκότητα των συναρτήσεων διαχείρισης που πρέπει να πραγματοποιήσει κάποιος agent, αφήνοντας βέβαια την πολλή επεξεργασία στους managers. Μ' αυτόν τον τρόπο:

- α) Το κόστος ανάπτυξης του λογισμικού ενός agent διαχείρισης καθώς και η πολυπλοκότητά του μειώνεται, όπως βέβαια και ο χρόνος υλοποίησής του. Αναφέρουμε ότι SNMP agents έχουν υλοποιηθεί σε λιγότερο από 10K bytes κώδικα.
- β) Παρ' όλα αυτά επειδή οι συναρτήσεις διαχείρισης που υποστηρίζονται έχουν σχετικά αυξημένη λειτουργικότητα, έχουμε καλύτερη χρησιμοποίηση του δικτύου.
- γ) Τα απλοποιημένα σύνολα των συναρτήσεων διαχείρισης γίνονται εύκολα αντιληπτά και έχουμε στην πράξη παραγωγή εργαλείων διαχείρισης δικτύων, τα οποία, όπως είναι φανερό, έχουν και μικρό κόστος.
- δ) Τέλος έχουμε ολοφάνερα την ελάχιστη πολυπλοκότητα και στα εργαλεία διαχείρισης.

Άλλοι στόχοι που είχαν τεθεί κατά την ανάπτυξη του SNMP ήταν η επεκτασιμότητα και η ανεξαρτησία από την αρχιτεκτονική των μηχανημάτων που θα διαχειριζόταν, στόχοι οι οποίοι επιτεύχθηκαν μέχρι κάποιο σημείο. Και αυτό γιατί η υλοποίηση του SNMP μπορεί να οδηγήσει σε προβλήματα όταν έρθει η στιγμή να διαχειριστεί μηχανήματα με διαφορετική λογική (όπως modems, multiplexers, T1 switches κ.α.), οπότε θα πρέπει να γραφτούν κατάλληλες προεκτάσεις και να ανοιχθούν μονοπάτια επικοινωνίας μεταξύ των μηχανημάτων αυτών και των agents.

Η τυποποίηση SMI (RFC 1155) επιτρέπει την έκδοση νεωτέρων ορισμών MIB με προεκτάσεις. Αποτέλεσμα των προεκτάσεων αποτέλεσε ο ορισμός **MIB-II** η οποία αποτελεί το σημερινό standard (RFC 1213, το οποίο δίνεται στο Παράρτημα B). Για παράδειγμα, στην MIB-II δίνεται η δυνατότητα σε ένα συγκεκριμένο προμηθευτή να πάρει ένα υποδένδρο από το object identifier tree (βλέπε και Σχήμα 4.4), το οποίο θα κρέμεται από τον κόμβο enterprises.private, και να ορίσει καινούργια αντικείμενα στην MIB, τα οποία να μπορούν να ανταποκρίνονται καλύτερα στα ιδιαίτερα προϊόντα του. Οι ορισμοί των MIB αυτών μπορούν μάλιστα να βρεθούν εύκολα μέσω του ηλεκτρονικού ταχυδρομείου (venera.isi.edu), πράγμα που φέρνει την διαχείριση ετερογενών δικτύων πιο κοντά στην πραγματικότητα. Πολλοί κατασκευαστές χρησιμοποιούν τη δυνατότητα αυτή προκειμένου να ορίσουν αντικείμενα που περιγράφουν το φυσικό επίπεδο (transmission group) όπως IEEE 802.3, IEEE 802.5, T1 κ.ά. τα οποία δεν υπάρχουν στην MIB-II.

Η MIB-II διαχειρίζεται συγκεκριμένους κόμβους (υπολογιστές, routers). Για την συλλογή συνολικών στοιχείων υποδικτύων (π.χ. στοιχεία φορτίου για τμήματα Ethernet) ορίστηκε η **RMON MIB** (Remote Monitoring MIB, RFC 1271). Οπως θα φανεί στη συνέχεια (Ενότητα 4.10) η RMON MIB παρακολουθεί απομακρυσμένα τμήματα του δικτύου για λογαριασμό του Συστήματος Διαχείρισης.

Κάθε agent κρατάει πληροφορίες μονάχα για ένα υποσύνολο αντικειμένων της MIB (MIB view), ανάλογα με τα πρωτόκολλα που είναι υλοποιημένα στο μηχάνημα που τρέχει ο agent (IP, TCP, UDP, EGP κ.ά), ενώ κάθε manager διαθέτει διαφορετικό τρόπο πρόσβασης για κάθε αντικείμενο της MIB (read-only, read-write) (SNMP access mode). Με τους μηχανισμούς αυτούς επιτρέπεται η υλοποίηση κάποιου σχήματος ασφαλείας. Αν το MIB view που κρατάει ο agent δεν αναφέρεται στο μηχάνημα στο οποίο τρέχει αλλά σε κάποιο άλλο, έχουμε να κάνουμε με κάποιο proxy agent. Ο μηχανισμός αυτός χρησιμοποιείται για τη διαχείριση μηχανημάτων που δεν

μπορούν να επικοινωνήσουν χρησιμοποιώντας το πρωτόκολλο SNMP, οπότε ο proxy agent θα πρέπει να μετατρέψει κατάλληλα τα πρωτόκολλα. Η ιδέα ενός proxy agent είναι μάλλον μια ανεπαρκής μέθοδος για την επιτυχία μιας ολοκληρωμένης λύσης. Μια άλλη άποψη είναι αυτή της πολύγλωσσης πλατφόρμας (βλέπε Κεφάλαιο 8). Ο σταθμός διαχείρισης θα πρέπει να μπορεί να επικοινωνεί με οποιαδήποτε μηχανήματα πάνω στο δίκτυο, χρησιμοποιώντας το πρωτόκολλο που καταλαβαίνει το συγκεκριμένο μηχάνημα. Με ένα συνδυασμό από εφαρμογές και δυναμικές σχεσιακές βάσεις δεδομένων, μπορεί ο σταθμός διαχείρισης να προσφέρει ολοκληρωμένη πληροφορία.

Το SNMP δίνει στις εφαρμογές διαχείρισης ένα πολύ μικρό σύνολο από στοιχεία υπηρεσίας (get-request, set-request, get-next-request, get-response, trap) για τον έλεγχο ή την αλλαγή των τιμών των αντικειμένων των MIBs στους διάφορους agents. Η επικοινωνία μεταξύ managers και agents επιτυγχάνεται με την ανταλλαγή μηνυμάτων κάθε ένα από τα οποία εξ' ολοκλήρου, και ανεξάρτητα από τα άλλα, κωδικοποιείται σύμφωνα με τους **Basic Encoding Rules (BER)** οι οποίοι είναι σχετικοί με το **Abstract Syntax Notation (ASN.1)** του ISO μέσα σε ένα μοναδικό UDP (βλέπε και σχήμα 4.3). Για να επιτευχθεί η παραπάνω απλότητα τέθηκαν όπως είναι λογικό κάποιοι περιορισμοί, τους οποίους θα εξετάσουμε παρακάτω πιο αναλυτικά.

#### 4.4. Λομή και αποθήκευση της διαχειριζόμενης πληροφορίας: SMI και MIB

Η πληροφορία που χρησιμοποιείται κατά τη λειτουργία του πρωτοκόλλου SNMP παριστάνεται σύμφωνα με ένα υποσύνολο του συντακτικού ASN.1 και κωδικοποιείται κατά την μεταφορά της, σύμφωνα με το αντίστοιχο υποσύνολο των BERs (βλέπε και παράγραφο 4.6). Αυτό σημαίνει ότι τόσο τα αντικείμενα που θα διαχειριστούμε, όσο και τα PDUs που θα μεταφέρουν τις τιμές των αντικειμένων αυτών είναι ορισμένα σύμφωνα με το συντακτικό ASN.1. Επιτρέπονται μονάχα οι στοιχειώδεις τύποι: **INTEGER**, **OCTET STRING**, **OBJECT IDENTIFIER**, **NULL** και οι σύνθετοι τύποι **SEQUENCE**, **SEQUENCE OF** του συντακτικού ASN.1. Με τον τρόπο αυτό οι ρουτίνες κωδικοποίησης /αποκωδικοποίησης απλοποιούνται, αφού ελέγχουν ένα μικρό αριθμό περιπτώσεων και έτσι έχουμε λιγότερο κώδικα και μικρότερους χρόνους επεξεργασίας. Όσο αφορά το υποσύνολο των BERs το οποίο χρησιμοποιείται στο πρωτόκολλο SNMP, αυτό περιλαμβάνει μονάχα κωδικοποιήσεις ορισμένου μήκους, δηλαδή κάθε ASN.1 τύπος που κωδικοποιείται πληροφορεί στην αρχή της κωδικοποίησής του για το μήκος της τιμής του. Επίσης σε όποιες περιπτώσεις είναι επιτρεπτό χρησιμοποιούνται κωδικοποιήσεις μη σύνθετων τύπων και όχι σύνθετων τύπων. Το μειονέκτημα σ' όλα αυτά είναι ότι δεν μας δίνεται η δυνατότητα να ορίσουμε ότι αντικείμενο θα θέλαμε. Στην πιο σύνθετη περίπτωση μπορούμε να ορίσουμε μια λίστα ή ένα πίνακα σαν μια σειρά από λίστες.

Το πρότυπο SMI αποτελεί το πλαίσιο αναφοράς για τον ορισμό μιας MIB μέσα στην οποία θα υπάρξουν οι ορισμοί όλων των αντικειμένων που θέλουμε να διαχειριστούμε. Στο πρότυπο αυτό ορίζονται οι τύποι των αντικειμένων που θα διαχειριστούμε μέσω της MIB, ο τρόπος που θα προσπελάσουμε τα αντικείμενα αυτά, ο διαχωρισμός τους σε ομάδες (groups), ο τρόπος ονομασίας τους και οτιδήποτε άλλο χρήσιμο. Η πληροφορία διαχείρισης που μεταφέρεται κατά την λειτουργία του πρωτοκόλλου SNMP περιορίζεται σε στιγμιότυπα μη σύνθετων τύπων ορισμένων είτε μέσα στην standard MIB, είτε αλλού σύμφωνα με τους περιορισμούς που τίθονται από το παραπάνω έγγραφο. Αυτό σημαίνει ότι το SNMP δεν βοηθά στην εξέταση μεγάλων ποσοτήτων πληροφορίας (σύνθετων τύπων), όπως για παράδειγμα πινάκων ή μιας ολόκληρης MIB. Το SNMP απαιτεί από τις εφαρμογές να ονομάζουν τα στιγμιότυπα των αντικειμένων που θέλουν να διαχειριστούν ακριβώς (με εξαίρεση την εντολή get-next-request που επιτρέπει την διαχείριση του επόμενου στην MIB αντικειμένου χωρίς τη

γνώση του ονόματός του, ξέροντας βέβαια το όνομα του προηγούμενου), και έτσι η σάρωση μιας νέας MIB θα πρέπει να γίνει ελέγχοντας ένα αντικείμενο κάθε φορά, με διαδοχικές εντολές get-next-request. Κάτι τέτοιο βέβαια δεν χρησιμοποιεί άριστα το δίκτυο, αφού τα πακέτα που στέλνονται είναι πολύ μικρά. Το πρόβλημα αυτό έχει λυθεί κατά κάποιο τρόπο στο SNMPv2 με τη λειτουργία get-bulk-request (βλ. παρακάτω).

Το δεύτερο πρότυπο είναι αυτό που περιγράφει τη Βάση Πληροφορίας Διαχείρισης - MIB. Το πρότυπο αυτό περιγράφει τα αντικείμενα για τα οποία μπορεί να κρατάει πληροφορίες ένας agent και να ζητάει πληροφορίες κάποιος διαχειριστής. Στην MIB-I (η οποία αποτελεί ιστορία) τα αντικείμενα αυτά ήταν οργανωμένα σε οκτώ groups, υποχρεώνοντας τους κατασκευαστές να υλοποιούν όλα τα αντικείμενα ενός group το οποίο τους ήταν χρήσιμο, αλλά όχι υποχρεωτικά όλα τα groups. Τα οκτώ αυτά groups ήταν τα εξής: **system** με πληροφορίες για τον συγκεκριμένο κόμβο, **interfaces** με πληροφορίες για τον τρόπο πρόσβασης του κόμβου στο δίκτυο, **at** με πίνακες για την μετάφραση IP διευθύνσεων σε διευθύνσεις του επιπέδου interface, **ip** με πληροφορίες σχετικά με την λειτουργία του πρωτοκόλλου IP που υλοποιεί ο κόμβος, **icmp** με πληροφορίες σχετικά με τα μηνύματα που ανταλλάσσει ο κόμβος με το πρωτόκολλο ICMP που υλοποιεί, **tcp** με πληροφορίες σχετικά με το πρωτόκολλο TCP που υλοποιεί ο κόμβος, **udp** με πληροφορίες σχετικά με το πρωτόκολλο UDP που υλοποιεί ο κόμβος, **egp** με πληροφορίες σχετικά με το πρωτόκολλο UDP που υλοποιεί ο κόμβος. Στην MIB-II έχουμε διάφορα νέα αντικείμενα, έναν επιπλέον πίνακα για την μετάφραση των διευθύνσεων του επιπέδου του interface σε διευθύνσεις IP, τη δημιουργία του **snmp** group για την διαχείριση και σχετικών με το SNMP αντικειμένων, καθώς και του **transmission** group το οποίο όμως συμπληρώνεται προς το παρόν από τους διάφορους κατασκευαστές.

#### 4.5. Λυνατές λειτουργίες στη διαχειριζόμενη πληροφορία. Μορφή και σημασία των ανταλασσόμενων μηνυμάτων.

Οι λειτουργίες που πρέπει να εκτελεί ένας agent για την ικανοποίηση των αιτήσεων του διαχειριστή είναι ο έλεγχος ή η αλλαγή μεταβλητών της MIB του. Έτσι τα μηνύματα που μπορεί να στείλει ένας manager σ' ένα agent είναι:

- α) το μήνυμα **get-request** με το οποίο ζητά την τιμή ενός συγκεκριμένου στιγμιότυπου ενός αντικειμένου.
- β) το μήνυμα **get-next-request** με το οποίο ζητά την τιμή του αμέσως επόμενου στιγμιότυπου μέσα στην ιεραρχική MIB.
- γ) το μήνυμα **set-request** με το οποίο ζητά να τεθεί ένα συγκεκριμένο στιγμιότυπο ενός αντικειμένου σε μια ορισμένη τιμή.

Τέλος υπάρχει και το κατάλληλο μήνυμα προκειμένου να μπορεί να απαντήσει ο agent σε αυτές τις ερωτήσεις:

- δ) το μήνυμα **get-response** με το οποίο επιστρέφεται κάποια τιμή ενός στιγμιότυπου αντικειμένου της MIB, μετά από κάποιο get-request ή get-next-request μήνυμα.

Η λογική λοιπόν του SNMP είναι ότι η παρακολούθηση της κατάστασης ενός δικτύου πραγματοποιείται με την εξεύρεση πληροφορίας από τους agents για λογαριασμό των managers. Υπάρχει βέβαια, όπως προηγουμένως είπαμε, και ένας περιορισμένος αριθμός μηνυμάτων TRAP που αυτόκλητα οι agents στέλνουν στους managers κατά κάποιο τρόπο για να οδηγήσουν την διαδικασία εύρεσης πληροφοριών, αναφέροντας

διάφορα σημαντικά συμβάντα. Τα TRAPs αυτά είναι τα εξής: *coldStart* και *warmStart* για τις αρχικοποιήσεις του agent, *linkDown* και *linkUp* για τις μεταβολές στα interfaces του agent, *authenticationFailure* για τις χωρίς δικαίωμα προσβάσεις στον agent, *egpNeighborLoss* για το κατέβασμα ενός gateway γείτονα EGP, και *enterpriseSpecific* για τον ορισμό TRAPs από τον κάθε κατασκευαστή. Είναι σαφής βέβαια η λιτότητα στον ορισμό των μηνυμάτων αυτών.

Ο μικρός αριθμός στοιχείων υπηρεσίας και ο μικρός αριθμός μηνυμάτων TRAPs δικαιολογείται βέβαια από την στρατηγική που στοχεύει στην απλότητα του SNMP. Παρατηρούμε επίσης στον ορισμό του πρωτοκόλλου την έλλειψη οποιονδήποτε προστακτικών εντολών διαχείρισης. Με τον τρόπο αυτό το SNMP αποφεύγει το γεγονός ότι ο αριθμός των εντολών αυτών μπορεί να γίνει απεριόριστα μεγάλος και η σημασία τους πολύ πολύπλοκη. Παρ' όλα αυτά όμως με τις εντολές που διαθέτει και κυρίως με τη set-request επιτρέπει την υλοποίηση οποιασδήποτε συνάρτησης διαχείρισης με την αλλαγή της τιμής μιας κατάλληλα ορισμένης μεταβλητής, αλλαγή η οποία θα έχει σαν αποτέλεσμα την εκτέλεση της εντολής.\* Οι περισσότερες set-request εντολές χρησιμοποιούνται πάντως για την αλλαγή των εγγραφών ενός routing table ή την αλλαγή της κατάστασης ενός interface.

Τα μηνύματα που ανταλλάσσονται κατά την λειτουργία του SNMP αποτελούνται από ένα αναγνωριστικό της version του πρωτοκόλλου, ένα όνομα κοινότητας SNMP, το οποίο καθορίζει κάποιον agent και τις εφαρμογές που μπορούν να επικοινωνήσουν μαζί του, και ένα PDU. Το PDU είναι μια από τις πέντε εντολές (get-request, get-next-request, get-response, set-request, TRAP). Μια υλοποίηση του πρωτοκόλλου περιμένει τα μηνύματα αυτά στη UDP port 161 του μηνύματος στον οποίο τρέχει, εκτός από τα μηνύματα TRAPs τα οποία περιμένει στην UDP port 162. Οι απαντήσεις στο διαχειριστή στέλνονται αφού πρώτα αντιστραφούν οι ports προορισμού προέλευσης. Δηλαδή μια get-request εντολή από την port 1000 θα προκαλέσει την αποστολή μιας get-response εντολής από την port 161 στην port 1000.

Γενικά το SNMP είναι ένα ασύγχρονο πρωτόκολλο ερώτησης/απόκρισης. Αυτό σημαίνει ότι μια εφαρμογή SNMP δεν χρειάζεται να περιμένει για την απάντηση, μετά την αποστολή ενός μηνύματος. Μπορεί να στείλει άλλα μηνύματα, ή να ασχοληθεί με άλλες υπηρεσίες. Παραπέρα από την στιγμή που μια ερώτηση ή μια απόκριση μπορεί να χαθεί λόγω της μη αξιόπιστης υπηρεσίας μεταφοράς, η διαχειριστική εφαρμογή θα πρέπει να φροντίζει για το επίπεδο αξιόπιστίας που επιθυμεί.

Η ανταλλάγη μηνυμάτων SNMP απαιτεί λοιπόν μια μη αξιόπιστη υπηρεσία μεταφοράς datagram (όπως αυτή που προσδιορίζει το πρωτόκολλο UDP). Παρ' όλα αυτά μπορεί να χρησιμοποιήσει οποιαδήποτε υπηρεσία μεταφοράς. Το SNMP είναι ανεξάρτητο των πρωτοκόλλων TCP/IP και μπορεί να υλοποιηθεί κατ' ευθείαν πάνω στο Ethernet, ή να χρησιμοποιήσει πρωτόκολλα OSI. Η υλοποίηση κατ' ευθείαν πάνω στο Ethernet μπορεί να χρησιμοποιηθεί, όπως είναι φανερό, μονάχα για την διαχείριση του ίδιου τοπικού δικτύου, και δεν συμφέρει, αφού με την μικρού κόστους υλοποίηση των πρωτοκόλλων UDP και IP η διαχειριστική πλατφόρμα μπορεί να δει ολόκληρο το Internet.

#### 4.6. Σύνταξη και κωδικοποίηση της πληροφορίας (ASN.1 και BER).

Θα ξεκινήσουμε την περιγραφή του συντακτικού ASN.1, ορίζοντας τι είναι αφηρημένο συντακτικό και σε τι είναι απαραίτητο. Στα κατώτερα στρώματα του μοντέλου OSI, όπως επίσης και στα στρώματα IP και TCP που διακρίνουμε σε ένα δίκτυο TCP/IP,

---

\* Αναφέρουμε εδώ το γνωστό παράδειγμα της υλοποίησης της "reboot command" με το μηδενισμό του στιγμιότυπου sysUpTime.0 που δίνει το χρόνο από το τελευταίο reboot.

κάθε PDU που ανταλλάσσεται μεταφέρει πληροφορία ορισμένης δομής. Διακρίνουμε διευθύνσεις, δεδομένα, CRCs, κ.ά. Από την στιγμή που η δομή της παραπάνω πληροφορίας είναι απλή, το μόνο που έχουμε να κάνουμε είναι να διατάξουμε την πληροφορία αυτή με κάποια γνωστή σειρά και να την μεταφέρουμε. Στην συνέχεια η δυαδική τιμή της σειράς από octets που αποτελούν κάποιο πεδίο καθορίζει το δεδομένο που μεταφέρεται.

Στο στρώμα εφαρμογής όμως οι δομές δεδομένων που ανταλλάσσονται είναι πολύ πιο σύνθετες. Οι τιμές που μεταφέρονται μπορεί να ανήκουν σε πολύ σύνθετους τύπους, όπως strings από χαρακτήρες, records, κ.ά. Για το λόγο αυτό χρειαζόμαστε έναν νέο τρόπο περιγραφής των δομών αυτών. Ο νέος τρόπος αυτός ονομάζεται **αφηρημένο συντακτικό (abstract syntax)** και χρησιμοποιείται για να ορίσει τύπους και τιμές αυτών των τύπων, χωρίς να περιορίζεται στις τιμές τύπων που προκύπτουν από κάποιο συγκεκριμένο μηχάνημα.

Από την στιγμή που για όλα τα μηχανήματα υπάρχει κάποιο κοινό σύνολο τύπων, θα πρέπει κατά την μεταφορά κάποιου δεδομένου να είναι δυνατό να γίνει κατανοητό, το συγκεκριμένο δεδομένο σε πιο τύπο ανήκει. Υπάρχει λοιπόν ανάγκη κωδικοποίησης των τιμών που προκύπτουν από το αφηρημένο συντακτικό για την μεταφορά τους. Οι λύσεις στα παραπάνω προβλήματα είναι το ASN.1 και οι **Basic Encoding Rules (BER)** αντίστοιχα.

Το ASN.1 προσφέρει ένα επίσημο συμβολισμό για τον ορισμό τύπων και για τον προσδιορισμό τιμών των τύπων αυτών. Ο τρόπος συμβολισμού αυτός έχει δύο πλεονεκτήματα: όταν χρησιμοποιείται στα χαρτιά είναι κατανοητός στους ανθρώπους, ενώ τιμές ορισμένες με τον συμβολισμό αυτό μπορούν εύκολα να κωδικοποιηθούν, προκειμένου να μεταφερθούν με σαφή και συμπαγή τρόπο, με την βοήθεια των BERs.

Πολύ σύντομα το ASN.1 ορίζει μια σειρά από απλούς τύπους και τους προσδιορίζει μια ετικέτα. Με την ετικέτα αυτή μπορούμε να αναφερόμαστε στους παραπάνω τύπους. Πέρα από τον ορισμό τύπων το ASN.1 ορίζει και τον συμβολισμό, για τον καθορισμό των τιμών των τύπων αυτών. Στην συνέχεια ορίζει μηχανισμούς για την κατασκευή συνθετότερων τύπων, και για τον προσδιορισμό ετικετών σ' αυτούς τους σύνθετους τύπους.

Στα παρακάτω πρέπει να έχουμε υπ' όψην μας ότι το ASN.1 χρησιμοποιείται και για τον ορισμό των δομών δεδομένων που ανταλλάσσονται με τη βοήθεια ενός πρωτοκόλλου (π.χ. SNMP), αλλά και για τον ορισμό της πληροφορίας που μεταφέρεται μέσα στις δομές αυτές (π.χ. αντικείμενα της MIB). Δηλαδή το μήνυμα GetResponse (ipRouteNextHop.9.1.2.3 = "99.0.0.3") που πρέπει να στείλει κάποιος agent με την βοήθεια του SNMP, έχει ορισθεί σαν κάποιος τύπος δεδομένων με την βοήθεια του ASN.1 και κατά την μεταφορά του κωδικοποιείται κατά τους BERs. Πέρα όμως του μηνύματος και το αντικείμενο της MIB ipRouteNextHop και το στιγμιότυπο του αντικειμένου ipRouteNextHop.9.1.2.3 και η τιμή του στιγμιότυπου 99.0.0.3 έχουν ορισθεί με την βοήθεια του ASN.1 και κωδικοποιούνται με την βοήθεια των BERs για την μεταφορά τους.

Βασικά στοιχεία του ASN.1 είναι όλοι οι αγγλικοί χαρακτήρες (case sensitive), οι αριθμητικοί χαρακτήρες και κάποιοι ειδικοί. Μία γραμμή - στην οποία περιγράφουμε τον τύπο μας - μπορεί να είναι όσο μακριά θέλουμε, υπάρχει τέλος δυνατότητα για σχόλια (τα οποία μπορεί να είναι σημαντικά για τον ορισμό ενός τύπου).

Ένα σύνολο από ορισμούς τύπων περιέχονται μέσα σε κάποιο module. Π.χ. οι ορισμοί των αντικειμένων μιας MIB ανήκουν συνήθως στο ίδιο module. Η γενική μορφή ενός module είναι:

## RFC1156-MIB DEFINITIONS ::= BEGIN

&lt;module body&gt;

END

Ένα module μπορεί να περιέχει προτάσεις IMPORTS και EXPORTS προκειμένου να εισάγει/εξάγει ορισμούς τύπων από/σε άλλα modules. Π.χ. από το module SMI προς τα modules MIB και SNMP. Για να είναι αυτό δυνατό, το όνομα ενός module θα πρέπει να είναι η τιμή ενός τύπου OBJECT IDENTIFIER προκειμένου να έχει ένα μοναδικό αναγνωριστικό. Γενικά έναν τύπο τον ορίζουμε με το ακόλουθο format

&lt;typerefERENCE&gt; ::= &lt;type&gt;

Π.χ. Counter ::= INTEGER

όπου ο τύπος INTEGER είτε έχει ορισθεί αλλού με τον ίδιο τρόπο, είτε είναι κάποιος build-in (όπως και είναι) τύπος του ASN.1.

Σε κάθε τύπο ορισμένο με την βοήθεια του ASN.1 προσδιορίζουμε μια ετικέτα, προκειμένου να μπορούμε να αναφερόμαστε σ' αυτόν. Κάθε ετικέτα αποτελείται από δύο μέρη:

α) την τάξη (class) του τύπου, με τέσσερεις εναλλαχτικές classes:

- i) ***Universal*** (class τύπων ορισμένων ως build-in τύπων της ASN.1).
- ii) ***Application*** (class τύπων ορισμένων σε ένα module).
- iii) ***Private*** (class τύπων ορισμένων μέσα σε κάποια ιδιωτική εταιρεία).
- iv) ***Context-specific*** (class τύπων ορισμένων μέσα σε κάποιο συνθετότερο τύπο της ASN.1)

β) κάποιο φυσικό αριθμό. Προκειμένου π.χ. να υπάρχουν πολλοί UNIVERSAL τύποι υπάρχει κάποιος φυσικός αριθμός που χαρακτηρίζει τον καθένα.

Ας τονίσουμε ότι η ετικέτα είναι ένα πολύ σημαντικό μέρος του ορισμού ενός τύπου από την στιγμή που οι κανόνες κωδικοποίησης BER χρησιμοποιούν τις ετικέτες αυτές προκειμένου να καθορίσουν ποιος είναι ο τύπος που μεταφέρεται.

Υπάρχουν δύο τρόποι προκειμένου να αλλάξεις την ετικέτα ενός τύπου και κατά συνέπεια να δημιουργήσεις ένα διαφορετικό τύπο:

- α) **EXPLICITly** όπου μια καινούργια ετικέτα προστίθεται, η παλιά όμως διατηρείται και κατά την κωδικοποίηση και μεταφορά μέσα από το δίκτυο της τιμής ενός τέτοιου τύπου μεταφέρονται και οι δύο ετικέττες.
- β) **IMPLICITly** όπου μια καινούργια ετικέτα αντικαθιστά την παλιά. Είναι ολοφάνερο ότι ο δεύτερος τρόπος είναι ο καλύτερος, αφού ρίχνει λιγότερη overhead πληροφορία στο δίκτυο.

Π.χ. Counter ::= [APPLICATION 1] IMPLICIT INTEGER

όπου ενώ ο τύπος INTEGER έχει ετικέτα UNIVERSAL 2 και μ' αυτήν αναγνωρίζεται, δημιουργούμε τον νέο τύπο Counter με ετικέτα APPLICATION 1, ο οποίος δεν παύει να είναι κάποιος integer, αλλά κατά την μεταφορά του μέσα από ένα δίκτυο αναγνωρίζεται μονάχα με την καινούργια ετικέτα του.

Πριν προχωρήσουμε στην αναφορά μερικών τύπων αναφέρουμε το γενικό format του προσδιορισμού μιας τιμής:

<valuereference> <type> ::= <value>

Π.χ. ifInOctets Counter ::= 102345

Στο πρότυπο SMI αναφέρεται ότι προκειμένου να διατηρηθεί κάποια απλότητα στα πρωτόκολλα που αφορούν την διαχείριση TCP/IP δικτύων δεν χρησιμοποιείται πλήρως το συντακτικό ASN.1, αλλά θέτονται περιορισμοί στην χρησιμοποίηση των τύπων του. Μ' αυτό το βλέμμα θα δούμε την ASN.1 σε αυτή την σύντομη αναφορά. Ετσι οι στοιχειώδεις τύποι της ASN.1 είναι οι εξής:

<b>INTEGER</b>	Συχνά κατά τον ορισμό ενός τέτοιου τύπου ακολουθεί μια λίστα από αναγνωριστικά. Δηλαδή:
	ifOperStatus ::= INTEGER {up(1), down(2), testing(3)}
	Η ετικέτα του τύπου αυτού είναι UNIVERSAL 2.
<b>OCTET STRING</b>	Μια λίστα από 0 ή περισσότερα octets. Η ετικέτα του τύπου αυτού είναι UNIVERSAL 4.
<b>NULL</b>	Ενας τύπος δεδομένων που λειτουργεί απλά σαν placeholder, με ετικέτα UNIVERSAL 5.

(Στην ASN.1 υπάρχουν και άλλοι στοιχειώδεις τύποι όπως BOOLEAN, BIT STRING, REAL κ.ά.)

Από τους παραπάνω τύπους μπορούμε να κατασκευάσουμε άλλους ως εξής:

- α) Με αλλάγη της ετικέτας όπως είδαμε παραπάνω.
- β) Με την βοήθεια των constructed τύπων του ASN.1 οι οποίοι είναι:

#### i) SEQUENCE, SEQUENCE OF

Με τον τύπο SEQUENCE μπορούμε να κατασκευάσουμε διατεταγμένες λίστες τιμών καθορισμένων τύπων. Οπως είναι φανερό ο τύπος αυτός έχει την δική του ετικέτα (UNIVERSAL 16), ενώ με αλλάγη ετικέτας μπορούμε να έχουμε τον δικό μας τύπο SEQUENCE.

```
Π.χ. atEntry ::= SEQUENCE {
    atIfIndex
        INTEGER
    atPhysAddress
        OCTET STRING
    atNetAddress
        NetworkAddress
}
```

Στα παραπάνω παραδείγματα οι τύποι SEQUENCE, INTEGER, OCTET STRING έχουν κρατήσει τις universal ετικέτες, ενώ ο τύπος NetworkAddress είναι ένας τύπος που θέλει παραπέρα ανάλυση.

Το παραπάνω θα μπορούσε να γραφεί σαν:

```
atEntry ::= SEQUENCE {INTEGER, OCTET STRING, OCTET STRING}
```

Με τον τύπο SEQUENCE OF μπορούμε να φτιάξουμε νέους τύπους οι οποίοι είναι arrays από τις παραπάνω λίστες. Δηλαδή:

```
atTable ::= SEQUENCE OF atEntry
```

Με το SEQUENCE φτιάχνουμε λίστες διαφορετικών τύπων, με το SEQUENCE OF φτιάχνουμε λίστες των ίδιων τύπων.

## ii) SET, SET OF

Χρησιμοποιούνται για την κατασκευή τύπων οι οποίοι είναι μη διατεταγμένες λίστες καθορισμένων τύπων.

## iii) CHOICE

Ένα τυχαίο υποσύνολο από διάφορους προκαθορισμένους τύπους.

```
II.χ. NetworkAddress ::= CHOICE {
    internet
        IpAddress
}
```

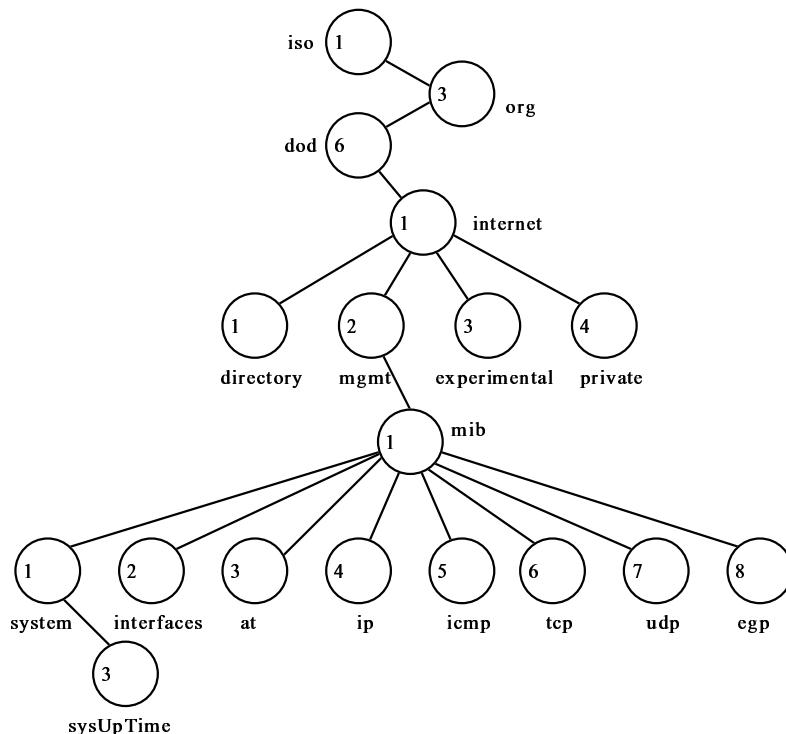
Με την βοήθεια του CHOICE μπορούμε να έχουμε μια διεύθυνση από πολλές πιθανές οικογένειες πρωτοκόλλων. Προς το παρόν όμως υπάρχει μονάχα η IP διεύθυνση.

Υπάρχουν στην ASN.1 και άλλοι constructed τύποι όπως πληθώρα τύπων Character Strings, ο τύπος EXTERNAL, ANY κ.ά.

Συνεχίζουμε με μια αναφορά στον τύπο OBJECT IDENTIFIER. Πολλά αντικείμενα (ένα document με standards, ένα module του ASN.1, ένα αντικείμενο ορισμένο σε μια MIB) στην κοινότητα OSI ή TCP/IP χρειάζονται μοναδικό όνομα, ένα μοναδικό αναγνωριστικό. Για να επιτρέψει η ASN.1 τον προσδιορισμό μοναδικών ονομάτων, δημιουργησε τον τύπο OBJECT IDENTIFIER. Οι πιθανές τιμές του τύπου αυτού, ορίζονται με αναφορά στο object identifier tree και είναι μια σειρά από φυσικούς αριθμούς κάθε ένας από τους οποίους είναι μια ετικέτα σε κάποιο κόμβο του δένδρου αυτού. Για να βρούμε το όνομα ενός αντικειμένου δεν έχουμε παρά να ξεκινήσουμε από την ρίζα του δένδρου, και να διαβάζουμε τις ετικέττες μέχρι το αντικείμενο.

Το όνομα του αντικειμένου sysUpTime της MIB το οποίο είναι κρεμασμένο από το κλαδί system 1 δίνεται λοιπόν σαν:

```
sysUpTime OBJECT IDENTIFIER ::= {1 3 6 1 2 1 1 3}
```



#### Σχήμα 4.4 - To object identifier tree

Θα μπορούσαμε να αναφερθούμε στο αντικείμενο αυτό με την ακολουθία 1.3.6.1.2.1.1.3

Παραπάνω παρατηρούμε ότι κάθε ετικέτα σ' ένα δένδρο περιέχει και ένα χαρακτηριστικό text. Μια σειρά από τέτοιου είδους texts είναι μια τιμή του τύπου ObjectDescriptor ο οποίος παρέχει μια κατανοητή στους ανθρώπους περιγραφή μιας τιμής του τύπου OBJECT IDENTIFIER.

Μετά το 1988 εισάχθηκε στο ASN.1 η έννοια του subtype. Ενας ASN.1 subtype αποτελεί μια τελειοποίηση κάποιου άλλου ASN.1 τύπου. Ο ορισμός του είναι ο ακόλουθος:

Subtype ::=  
Parenttype (--subtype's specifications--)

Π.χ. έχουμε:

Counter ::=  
[APPLICATION 1]  
IMPLICIT INTEGER (0..4294967295)

ο παραπάνω subtype μας καθορίζει και τα όρια που μπορούν να έχουν οι τιμές του τύπου.

ipAddress ::=  
[APPLICATION 0]  
IMPLICIT OCTET STRING (SIZE(4))

ο παραπάνω subtype μας καθορίζει ότι το μήκος του string θα είναι αυστηρά 4 octets.

Το ASN.1 ορίζει επίσης αυστηρά τον κατάλληλο συμβολισμό για τον ορισμό ενός macro, για ορισμένες εφαρμογές όπου είναι επιθυμητό να προστεθούν προεκτάσεις στο συντακτικό ASN.1 από τον χρήστη.

Ας σκεφτούμε το εξής: το αντικείμενο sysUpTime στην MIB είναι ο χρόνος από το τελευταίο reboot του μηχανήματος στο οποίο βρίσκεται ο agent ο οποίος κρατάει την MIB. Το αντικείμενο αυτό θα πρέπει να έχει κάποιο μοναδικό όνομα για να μπορούμε να το ξεχωρίζουμε με σιγουριά, άρα θα πρέπει να είναι η τιμή ενός τύπου OBJECT IDENTIFIER. Παρ' όλα αυτά σαν timer δεν παύει να είναι κάποιος αριθμός, τιμή δηλαδή κάποιου τύπου σαν τον INTEGER. Η ASN.1 όπως την έχουμε γνωρίσει ως τώρα δεν μας βοηθά στον ορισμό ενός τέτοιου αντικειμένου. Ο σκοπός λοιπόν του ορισμού ενός macro, είναι να δώσει κανείς μέσα από ένα τύπο περισσότερες σημασιολογικές πληροφορίες.

Ο ορισμός του είναι:

```
<name> MACRO ::=  
BEGIN  
    TYPE NOTATION ::= <new type notation>  
    VALUE NOTATION ::= <new value notation>  
END
```

Έχουμε λοιπόν νέο τρόπο συμβολισμού και του τύπου και των τιμών του.

Π.χ. OBJECT-TYPE MACRO ::=  
BEGIN

```
    TYPE NOTATION ::=  
        "SYNTAX" type(TYPE Objectsyntax)  
        "ACCESS" Access  
        "STATUS" Status  
    VALUE NOTATION ::=  
        value(VALUE Objectname)
```

```
        Access ::= "read-only"  
            | "read-write"  
            | "write-only"  
            | "not-accecible"
```

```
        Status ::= "mandatory"  
            | "obsolete"  
            | "optional"
```

END

Με το macro αυτό επιτρέπουμε οτιδήποτε χρήσιμο γύρω από ένα αντικείμενο προς διαχείριση να παρουσιαστεί με ένα επίσημο τρόπο. Αντί λοιπόν για το γνωστό format:

<valueresference> <type> ::= <value>

έχουμε:

```
atPhysAddress OBJECT-TYPE  
    SYNTAX OCTET STRING  
    ACCESS read-write  
    STATUS mandatory  
    ::= {atEntry 2}
```

Καλό είναι να προσέξει κανείς το νέο συμβολισμό για το <type>. Επίσης καλό είναι να προσέξει κανείς το καινούργιο value notation το οποίο θα καθορίσει τον τρόπο που θα κωδικοποιηθεί μια τιμή για την μεταφορά της.

Από την στιγμή που έχουμε ορίσει κάποιο abstract syntax, χρειαζόμαστε και κάποιο transfer syntax. Το συντακτικό αυτό θα καθορίσει κατά την μεταφορά μιας τιμής μέσα από ένα δίκτυο, σε ποιο τύπο ανήκει η τιμή αυτή. Οι Basic Encoding Rules παρέχουν ένα τέτοιο συμβολισμό. Το γενικό πλαίσιο κωδικοποίησης είναι:

Type - Length - Value

όπου:

**Type:** ο τύπος της μεταδιδόμενης τιμής.

**Length:** το μήκος σε octets του Value.

**Value:** η μεταδιδόμενη τιμή.

Προκειμένου να μπορούν να μεταφερθούν constructed τύποι, κάθε πεδίο Value μπορεί να αποτελείται από άλλες TLV κωδικοποίησεις.

T	L	T	L	V	T	L	V
---	---	---	---	---	---	---	---

--

V

Ο τρόπος κωδικοποίησης του τύπου χρησιμοποιεί την ετικέτα του τύπου όπως φαίνεται παρακάτω:

8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---

-----

class P / C

Με τα bits 7 και 8 κωδικοποιούμε τα τέσσερα είδη classes ως εξής:

Universal	00
Application	01
Context-spec.	10
Private	11

Το bit 6 δηλώνει αν είναι 0 ότι ο τύπος είναι στοιχειώδης (primitive), οπότε το Value που ακολουθεί την TLV κωδικοποίηση είναι η τιμή του. Αν το bit 6 είναι 1 ο τύπος είναι σύνθετος (constructed), και το Value που ακολουθεί την TLV κωδικοποίηση, αποτελείται από άλλες TLV κωδικοποίησεις.

Τα άλλα 5 bits χρησιμοποιούνται για την κωδικοποίηση του φυσικού αυτού αριθμού που αποτελεί το δεύτερο μέρος της ετικέτας ενός τύπου. Με τα 5 αυτά bits μπορούμε να έχουμε 31 διαφορετικούς τύπους σε μια class. Αν χρειαζόμαστε περισσότερους χρησιμοποιούμε και άλλα octets, τα οποία όλα εκτός του τελευταίου θα πρέπει να έχουν το MSB τους 1, ενώ τα 5 bits στο πρώτο octet βρίκονται υψηλά στο 1. Τα

υπόλοιπα 7 bits από κάθε octet χρησιμοποιούνται για την κωδικοποίηση του φυσικού αριθμού που χαρακτηρίζει τον τύπο.

Π.χ.

0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

ο τύπος αυτός έχει ετικέτα [APPLICATION 232] και είναι επίσης constructed (π.χ. SEQUENCE).

Υπάρχουν τρεις τρόποι για την κωδικοποίηση του πεδίου Length.

α) Μορφή short

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

(μήκος 5 octets)

με τα 7 τελευταία bits να μας δείχνουν το μήκος του Value σε octets, ενώ το πρώτο bit είναι 0.

β) Μορφή long

1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(μήκος 5 octets)

όπου με τα 7 τελευταία bits έχουμε τον αριθμό των octets (2 στο παράδειγμα) που θα περιέχουν το μήκος του Value σε octets.

γ) Μορφή αόριστου μήκους (η οποία δεν χρησιμοποιείται στο SNMP)

8	0	contents	0	0	0	0	0
---	---	----------	---	---	---	---	---

(σε hex)

Προηγείται το 10000000, ακολουθεί το Value, ενώ δύο octets ίσα με μηδέν δηλώνουν το τέλος του Value. Επειδή χρησιμοποιείται με constructed τύπους, δεν υπάρχει πρόβλημα σχετικά με το αν θα έχουμε δύο συνεχόμενα μηδενικά octets μέσα στο Value.

Ας δούμε τώρα τον τρόπο κωδικοποίησης των τιμών των τύπων:

Ένας INTEGER τύπος κωδικοποιείται σε ένα ή περισσότερα octets, πάντα σαν στοιχειώδης τύπος, και η τιμή του είναι γραμμένη σε συμπλήρωμα ως προς 2. Επίσης κωδικοποιείται με τον συντομότερο τρόπο, π.χ. ο ακέραιος 5 θα κωδικοποιηθεί σε ένα octet χωρίς να χρησιμοποιήσει περισσότερα μηδενικά octets.

Π.χ. ας θυμηθούμε το εξής παράδειγμα:

Counter ::= [APPLICATION 1] IMPLICIT INTEGER (0..4294967295)  
και

tcpAttemptFails Counter ::= 7

η κωδικοποίηση θα έχει ως εξής:

0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Ένας τύπος OCTET STRING κωδικοποιείται σε κανένα, ένα ή περισσότερα octets σαν (αν αποτελείται από μικρότερα octet strings) τύπος στοιχειώδης ή constructed.

Π.χ. ας θυμηθούμε το εξής παράδειγμα:

ipAddress ::= [APPLICATION 0] IMPLICIT OCTET STRING (SIZE(4))

και

ipAdEntAddr ipAddress ::= 192.33.4.20

η κωδικοποίηση θα έχει ως εξής:

4	0	0	4	C	0	2	1	0	4	1	4
---	---	---	---	---	---	---	---	---	---	---	---

(σε octets)

ας προσέξουμε στα παραπάνω ότι οι universal ετικέττες, τις οποίες εμείς IMPLICITLY αντικαταστήσαμε, καθώς και οι περιορισμοί που περιέχονται στους ορισμούς των subtypes δεν μεταφέρονται μέσα από το δίκτυο μετά την κωδικοποίηση.

Ένας OBJECT IDENTIFIER τύπος κωδικοποιείται σε ένα ή περισσότερα octets σαν στοιχειώδης τύπος. Ας προσέξουμε τα εξής:

- Οι δύο πρώτοι φυσικοί αριθμοί που βρίσκουμε καθώς διαβάζουμε το object identifier tree κωδικοποιούνται σαν ένας σύμφωνα με την σχέση  $40^*X+Y$  όπου X ο πρώτος και Y ο δεύτερος.
- Κάθε ένας φυσικός αριθμός κωδικοποιείται σε ένα ή περισσότερα octets. Χρησιμοποιούνται τα 7 τελευταία bits, ενώ το όγδοο που είναι το MSBit δείχνει όταν είναι 1, αν και το επόμενο octet χρησιμοποιείται για την περιγραφή του ίδιου αριθμού. Αν δηλαδή είναι 1 και το MSBit του επόμενου octet είναι 0, τότε ο φυσικός αριθμός περιγράφεται από τα υπόλοιπα 14 bits.

Π.χ. η τιμή 1.1.3.4.1200 κωδικοποιείται σαν:

0	6	0	5	2	9	0	3	0	4	0	4	B	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---

(σε hex)

(η ετικέτα που χρησιμοποιήθηκε ήταν η UNIVERSAL 6 και ο τύπος είναι στοιχειώδης).

Ένας SEQUENCE τύπος κωδικοποιείται σε κανένα, ένα ή περισσότερα octets πάντα σαν σύνθετος τύπος. Οι τύποι από τους οποίους αποτελείται κωδικοποιούνται (μέσα πια στον SEQUENCE τύπο) σύμφωνα με τα παραπάνω.

Π.χ. ο τύπος:

```
VarBind ::=  
SEQUENCE {  
    name  
        ObjectName
```

```

    value
ObjectSyntax
}

```

είναι ουσιαστικά το πεδίο εκείνο ενός PDU στο SNMP που μεταφέρει την τιμή ενός αντικειμένου που μας ενδιαφέρει. Το αντικείμενο κωδικοποιείται πάντα σαν τύπος OBJECT IDENTIFIER, ενώ η τιμή του κωδικοποιείται όπως ο τύπος ObjectSyntax. Ετσι αν είχαμε σε κάποια MIB ενός agent την μεταβλητή sysDescr ίση με "UNIX", θα βλέπαμε στο SNMP πακέτο την κωδικοποίηση:

3	0	1	0	0	6	0	8	2	B	0	6	0	1	0	2	0	1
0	1	0	1	0	0	0	4	0	4	7	5	6	E	6	9	7	8

(σε hex)

Ο τύπος SEQUENCE έχει ετικέτα UNIVERSAL 16 και είναι σύνθετος απ' όπου προκύπτει το 30. Οι τύποι OBJECT IDENTIFIER και OCTET STRING έχουν ετικέττες UNIVERSAL 6 και UNIVERSAL 4 αντίστοιχα και είναι στοιχειώδεις, απ' όπου προκύπτουν τα 06 και 04. Σαν τιμή του OBJECT IDENTIFIER κωδικοποιήσαμε την 1.3.6.1.2.1.1.1.0 όπου το suffix 0 έχει προστεθεί για να υποδεικνύει ότι η μεταβλητή sysDescr σεν αποτελεί στήλη κάποιου πίνακα, είναι δηλαδή φύλλο στο object identifier tree.

Το να γράψεις κώδικα που να κωδικοποιεί και να αποκωδικοποιεί ASN.1 ορισμένες - BER κωδικοποιημένες τιμές τύπων, δεν είναι κάτι εύκολο. Η χρήση ενός ASN.1 compiler απλοποιεί το πρόβλημα.

Ένας ASN.1 compiler διαβάζει τους ορισμούς των τύπων και παράγει κάποιο κώδικα (συνήθως σε C). Ο κώδικας αυτός περιέχει κάποιες δομές δεδομένων και κάποιες συναρτήσεις που ενεργούν σε αυτές τις δομές.

Στην συνέχεια όλοι οι ορισμοί σε ASN.1 και ο compiler δεν εμφανίζονται πουθενά, απλά οι ρουτίνες κωδικοποίησης / αποκωδικοποίησης θα έχουν σαν κάποιες από τις παραμέτρους τους τις δομές που έβγαλε ο compiler. Π.χ. ένας τύπος INTEGER θα αντιστοιχούσε σε μια δομή integer της C, ενώ ένας τύπος ipAddress θα αντιστοιχούσε σε μια δομή struct sockaddr\_in \* στην C, γνωστή από το κεφάλαιο programming with sockets.

Μπορεί κανείς να δει τις ρουτίνες snmp\_parse\_var\_op() και snmp\_build\_var\_op() στην περιγραφή του snmpd server του CMU-SNMP.

## 4.7. Υλοποίηση λειτουργιών διαχείρισης με βάση το SNMP

Παρακάτω θα παρουσιάσουμε μια γενική μεθοδολογία υλοποίησης λειτουργιών διαχείρισης βασισμένων στη φιλοσοφία του SNMP. Πολύ σύντομα θα αναπτύξουμε τον τρόπο με τον οποίο μπορούμε να χρησιμοποιήσουμε τις δυνατότητες που μας προσφέρει το πρωτόκολλο SNMP, λαμβάνοντας υπ' όψη τα αντικείμενα για τα οποία κάποιος agent μπορεί να μας δώσει πληροφορίες.

### 4.7.1. Παρουσίαση της τοπολογίας του δικτύου

Μια απλή υλοποίηση της παραπάνω λειτουργίας θα μας έδινε τους hosts και τους routers του δικτύου που μας ενδιαφέρει (με έμφαση στους δεύτερους), καθώς και τα interfaces που αυτοί βλέπουν.

Κάνοντας polling στους διάφορους agents και ελέγχοντας την μεταβλητή ifOperStatus ο manager διαπιστώνει ποια interfaces είναι "πάνω" και ποια είναι "κάτω". Απαραίτητη βέβαια είναι και η παρακολούθηση των linkUp και linkDown TRAPs που καθοδηγούν κατά κάποιο τρόπο το polling. Έχοντας τις πληροφορίες αυτές ο manager μπορεί :

- α) Να επαληθεύσει κάποιο γεγονός που μαθαίνει από κάποιο TRAP-PDU, κάνοντας get την τιμή της κατάλληλης μεταβλητής.
- β) Να παρουσιάσει τα interfaces που είναι σε κατάσταση "πάνω".
- γ) Να αφαιρέσει από την εικόνα του δικτύου τα interfaces που πέρασαν σε κατάσταση "κάτω".
- δ) Να σημειώσει κάπου το γεγονός για κάποια αναφορά.

Όσο αναφορά την παρουσίαση των μηχανημάτων (hosts, routers), απαραίτητη κρίνεται η προσθήκη δύο νέων μεταβλητών στην MIB-I με το όνομα και τη θέση των μηχανημάτων (sysName, sysLocation) που θα διευκόλυναν την αυτόματη δημιουργία χάρτη του δικτύου, και οι οποίες υπάρχουν στην MIB-II.

Για την αφαίρεση ενός μηχανήματος από την εικόνα του δικτύου, αυτό που θα πρέπει να ληφθεί υπ' όψην είναι η σιωπή του αντίστοιχου agent στις ερωτήσεις του manager.

#### 4.7.2. Παρακολούθηση της απόδοσης του δικτύου

Στο interface group της MIB-I υπάρχουν διάφοροι μετρητές που μας επιτρέπουν να παρακολουθήσουμε την απόδοση και να εντοπίσουμε πιθανά προβλήματα σε κάθε interface. Ο χρόνος που κάποιο interface βρίσκεται σε μια συγκεκριμένη κατάσταση (π.χ σε λειτουργία) δίνεται από την διαφορά sysUpTime - ifLastChange, αλλά επειδή ο χρόνος αυτός είναι πολύ μεγάλος, οι μετρητές θα πρέπει να εξετάζονται μέσα σε ένα συγκεκριμένο διάστημα δειγματοληψίας με μια συγκεκριμένη συχνότητα αν αυτό είναι δυνατό, προφανώς κρατώντας τις προηγούμενες τιμές των μετρητών που μας ενδιαφέρουν και του sysUpTime. Συμπληρώνουμε εδώ ότι η μεταβλητή sysUpTime μας δίνει τον χρόνο σε εκατοστά του δευτερολέπτου.

Στο επίπεδο του interface μπορούμε να μετρήσουμε τον ρυθμό εξυπηρέτησης (throughput) για κάθε σύνδεσμο. Οι μεταβλητές που μας δίνουν την πληροφορία αυτή σε κάθε κατεύθυνση είναι οι ifInOctets και ifOutOctets και περιέχουν και τους χαρακτήρες του framing (overhead). Ο ρυθμός εξυπηρέτησης μετριέται ξεχωριστά σε κάθε κατεύθυνση. Από την στιγμή που έχουμε τον πραγματικό ρυθμό εξυπηρέτησης μπορούμε να τον συγκρίνουμε με την χωρητικότητα του δικτύου που δίνεται από την μεταβλητή ifSpeed σε bits/sec, και πιθανά να δημιουργήσουμε κάποιο συναγερμό όταν ο πραγματικός ρυθμός εξυπηρέτησης περνάει κάποιο κατώφλι.

Μπορούμε ακόμα αν γνωρίζουμε τα interfaces που είναι συνδεδεμένα σε ένα κομμάτι δικτύου να αθροίσουμε όλα τα octets που αυτά δίνουν στο δίκτυο, ώστε να υπολογίσουμε το φορτίο στο κομμάτι αυτό του δικτύου. Το παραπάνω είναι μάλλον πολύπλοκο για ένα μεγάλο δίκτυο, οπότε και πρέπει να καταφύγουμε σε άλλες λύσεις πέρα από την MIB-II (βλ. παρακάτω RMON-MIB) προκειμένου να υπολογίσουμε το συνολικό φορτίο σε ένα τιμήμα δικτύου (π.χ. σε ένα Ethernet segment)

Ο ρυθμός εξυπηρέτησης σε κάθε interface θα μπορούσε να μετρηθεί επίσης σε πακέτα ανά μονάδα χρόνου. Οι μεταβλητές που μας δίνουν τα πακέτα που φθάνουν σε κάποιο interface είναι: ifInUcastPkts, ifInNUcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ενώ οι μεταβλητές που μας δίνουν τα πακέτα που ζεκινούν από κάποιο interface είναι: ifOutUcastPkts, ifOutNUcastPkts, ενώ τα πακέτα ifOutDiscards και ifOutErrors επιβαρύνουν με φορτίο το interface, αλλά όχι τελικά το δίκτυο.

Μπορούμε να υπολογίσουμε την πιθανότητα λανθασμένης μεταφοράς πακέτου προς κάποιο interface με το λόγο του ifInErrors προς τα συνολικά πακέτα που έφθασαν στο interface, όπως επίσης τις πιθανότητες απόρριψης πακέτων από το interface, παρ' ότι αυτά είναι σωστά (σε κάθε κατεύθυνση αν θέλουμε) με τους λόγους των ifInDiscards και ifOutDiscards προς τα συνολικά πακέτα. Η πρώτη πιθανότητα είναι παράμετρος ποιότητας της γραμμής επικοινωνίας, ενώ η δεύτερη είναι παράμετρος ποιότητας της κάρτας του interface. Υπάρχει και η πιθανότητα άφιξης πακέτου το οποίο αναφέρει μη γνωστό πρωτόκολλο.

Τέλος στο επίπεδο του interface ενδιαφέρον μέγεθος είναι το ifOutQLen που είναι το μήκος (σε πακέτα) της ουράς εξόδου.

Παρακολουθώντας το ρυθμό εξυπηρέτησης και τα λάθη σε κάθε σύνδεσμο, και πληροφορώντας τον χρήστη όταν αυτά ξεπεράσουν κάποιο κατώφλι, μπορούν να αναγνωριστούν δυνατά σημεία συμφόρησης και ίσως να βρεθεί κάποια λύση.

Στο IP group της MIB-I υπάρχουν επίσης διάφοροι μετρητές οι οποίοι μας επιτρέπουν να παρακολουθήσουμε την απόδοση, όσο αναφορά τη μεταφορά datagrams. Υπάρχει ακόμα και πίνακας δρομολόγησης ο οποίος βοηθά στην αναγνώριση πιθανών προβλημάτων στην δρομολογηση.

Κατά τη διαχείριση του επιπέδου IP θα πρέπει να δίνεται προσοχή από το διαχειριστή στο αν διαχειρίζεται host ή gateway μια και τα αντικείμενα στην MIB αποκτούν συχνά διαφορετική έννοια για κάθε από τις λειτουργίες αυτές. Η μεταβλητή ipForwarding μας καθορίζει το αν το μηχάνημα λειτουργεί σαν host ή σαν gateway.

Ο συνολικός αριθμός datagrams που έφθασαν στη συγκεκριμένη IP-διεύθυνση είναι ipInReceives, ενώ ο συνολικός αριθμός datagrams που δημιουργησε το συγκεκριμένο μηχάνημα είναι ipOutRequests (host). Υπάρχει η δυνατότητα για τον υπολογισμό διαφόρων πιθανοτήτων. Αναφέρουμε τις εξής: πιθανότητα λήψης datagram με κάποιο λάθος που ισούται με τον λόγο ipInHdrErrors προς το ipInReceives. Η πιθανότητα απόρριψης datagram παρ' ότι αυτό είναι σωστό με τους λόγους των ipInDiscards και ipOutDiscards προς ipInReceives και ipOutRequests αντίστοιχα.

Πριν περάσουμε στο θέμα της δρομολόγησης θα πρέπει να προσθέσουμε ότι στο επίπεδο IP πραγματοποιείται και η λειτουργία του fragmentation, όταν το μέγεθος κάποιου datagram ξεπερνά το data field του φυσικού πλαισίου το οποίο θα το μεταφέρει. Μπορούμε να παρακολουθήσουμε την απόδοση της λειτουργίας αυτής, μετρώντας τις πιθανότητες μη σωστής επανασύνδεσης datagram με τον λόγο ipReasmFails προς ipReasmReqds και απόρριψης datagram λόγω μη ύπαρξης δυνατότητας κατάτμησης με τον λόγο ipFragFails προς ipFragOKs + ipFragFails.

Φθάνοντας, τέλος, στο θέμα της δρομολόγησης, σε μια πρώτη ματιά υπάρχουν η πιθανότητα απόρριψης datagram από router λόγω μη εύρεσης δρόμου για την δρομολόγηση. Είναι ipOutNoRoutes προς IpForwDatagrams.

Γενικά πάντως ο διαχειριστής μπορεί περιοδικά να εξετάζει τα μέτρα που αξιολογούν τη δρομολόγηση και το πρωτεύων ipRouteMetric1, η σημασία του οποίου καθορίζεται

από το πρωτόκολλο δρομολόγησης. Η συχνή αλλάγη της τιμής του μέτρου αυτού είναι μια προειδοποίηση για προβλήματα στην δρομολόγηση. Με κάποιο αλγόριθμο θα μπορούσε ο διαχειριστής να ακολουθήσει βήμα-βήμα τη διαδρομή προσπαθώντας να βρει το σημείο του προβλήματος. Σε ένα τέτοιο αλγόριθμο θα βοηθούσε και ο τύπος του δρόμου που μας δίνεται από την MIB σαν ipRouteType (invalid, direct, remote, ...).

#### 4.7.3. Άλλες ενδείξεις

Ενδιαφέρον φαίνεται και το group ICMP της MIB. Σ' αυτό υπάρχουν μετρητές που καταγράφουν τον αριθμό των ICMP μηνυμάτων που έστειλε ή έλαβε το μηχάνημα. Επειδή τα μηνύματα αυτά μεταφέρουν αναφορές σχετικά με προβλήματα, η αύξηση σε κάποιο χρονικό διάστημα ενός συγκεκριμένου είδους μηνυμάτων σημαίνει πιθανά κάποιο ιδιαίτερο πρόβλημα. Έτσι αντικείμενα όπως icmpInTimeExcds, icmpInParmProbs, icmpInscrQuenches που μετρούν τον αριθμό των μηνυμάτων τα οποία πληροφορούν για timeouts και προβλήματα στα datagrams που έστειλε το μηχάνημα και για απαίτηση επιβράδυνσης του ρυθμού μετάδοσης του μηχανήματος, μπορούν να βοηθήσουν στον εντοπισμό κάποιου προβλήματος. Προφανώς υπάρχουν και τα αντίστοιχα μηνύματα που έστειλε το μηχάνημα.

Σαν παρένθεση μπορούμε να προσθέσουμε ότι μια σειρά από timestamp μηνύματα ICMP θα μας έδιναν τη δυνατότητα υπολογισμού της καθυστέρησης μετάδοσης (transit delay) μεταξύ δύο μηχανημάτων.

Προφανώς υπάρχει η δυνατότητα για τον υπολογισμό της πιθανότητας άφιξης λανθασμένου datagram ή αδυναμίας δημιουργίας datagram, τόσο στο επίπεδο ICMP όσο και στο επίπεδο UDP. Στο group UDP έχουμε επίσης τον αριθμό των datagrams για τα οποία δεν υπήρξε πόρτα προορισμού, μια παραπάνω λοιπόν πιθανότητα μη σωστής χρησιμοποίησης του δικτύου.

Τέλος στο επίπεδο TCP ο λόγος των συνδέσεων που είναι ενεργές tcpCurrEstab προς το μέγιστο αριθμό TCP συνδέσεων που επιτρέπονται ταυτόχρονα tcpMaxConn μας δίνει μια ένδειξη της χρησιμοποίησης των υπηρεσίων του δικτύου, ενώ ο λόγος των segments που επαναμεταδόθηκαν tcpRetransSegs προς τα segments που στάλθηκαν μέσα από τις εγκατεστημένες συνδέσεις tcpOutSegs είναι ένδειξη προβλημάτων στο δίκτυο.

Η υλοποίηση του πρωτοκόλλου SNMP καθώς και της διαχειριστικής εφαρμογής που θα το χρησιμοποιήσει, είναι αυτό που έχει σημασία μια και η συγκεκριμένη υλοποίηση θα αποδείξει την απόδοση, την αξιοπιστία και την επεκτασιμότητα του πρωτοκόλλου.

### 4.8. Πραγματοποιώντας διαχείριση επιδόσεων με το SNMP [ST1592]

**Διαχείριση επιδόσεων (Performance Management)** ονομάζουμε τη διαδικασία μέτρησης των επιδόσεων όλων των στοιχείων που αποτελούν ένα δίκτυο. Η διαδικασία αυτή περιλαμβάνει συναρτήσεις εύρεσης της χρησιμοποίησης όλων των συνδέσμων και των τμημάτων του δικτύου, αναγνώρισης περιοχών που τείνουν να συμφορηθούν, και τέλος την ανεύρεση προτύπων που πιθανά ακολουθεί το φορτίο στο δίκτυο. Κάθε μια από τις λειτουργίες αυτές μπορεί να βοηθήσει το διαχειριστή του δικτύου στο να καθορίσει και να διασφαλίσει, το κατά πόσον το δίκτυο αποδίδει κατά τις επιθυμίες του χρήστη. Η διαχείριση επιδόσεων επίσης μπορεί να βοηθήσει το διαχειριστή να λύσει τόσο βραχυπρόθεσμα προβλήματα, όπως αργούς χρόνους απόκρισης, όσο και να παρατηρήσει μακροπρόθεσμες τάσεις τους δικτύου.

Στην παράγραφο αυτή θα δούμε, πως πραγματοποιείται η διαχείριση επιδόσεων όταν ο διαχειριστής έχει στην διάθεσή του πλατφόρμα που μπορεί και στέλνει μηνύματα SNMP. Θα συμπληρώσουμε, ότι απαραίτητοι είναι και οι agents που τρέχουν στους διαφόρους κόμβους και διατηρούν βάσεις με πληροφορίες για την κατάσταση του δικτύου. Μάλιστα θα εξετάσουμε την διαχείριση επιδόσεων με βάση τις διαχειριστικές πληροφορίες που μπορεί να βρει κανείς στις MIB II (RFC1213) και RMON MIB (RFC1271). Η MIB II περιγράφει αντικείμενα, για τα οποία μπορεί να βρει κανείς πληροφορίες σε ένα μηχάνημα που χρησιμοποιεί πρωτόκολλα TCP/IP. Η RMON MIB περιγράφει τις πληροφορίες που μπορεί να βρει κανείς σε μηχανήματα που τρέχουν προγράμματα παρακολούθησης απομακρυσμένων δικτύων. Έτσι από τα αντικείμενα της RMON MIB μπορεί να πάρει κανείς πληροφορίες για τμήματα δικτύων, όπου τα μηχανήματα δεν "καταλαβαίνουν" το SNMP. Επίσης μπορεί να πάρει πληροφορίες για τη συνολική χρησιμοποίηση για ένα τμήμα τοπικού δικτύου.

Η διαχείριση επιδόσεων συχνά απαιτεί την εξέταση στατιστικών μεγεθών σε ένα συγκεκριμένο χρονικό διάστημα. Το χρονικό αυτό διάστημα μπορεί να μεταβάλλεται από λίγα δευτερόλεπτα, κάποιες ώρες ή και περισσότερο. Για παράδειγμα, κάποιος μπορεί να θέλει να εξετάσει το ρυθμό λαθών σε ένα σύνδεσμο κάθε λίγα δευτερόλεπτα. Ενώ, κάποιος άλλος μπορεί να θέλει να εξετάσει την χρησιμοποίηση ενός τμήματος ενός δικτύου, την τελευταία ημέρα. Και στις δύο περιπτώσεις είναι απαραίτητος ο υπολογισμός της μεταβολής ενός στατιστικού μεγέθους από την χρονική στιγμή  $t_0$  μέχρι την χρονική στιγμή  $t_1$ . Δηλαδή είναι απαραίτητος ο υπολογισμός μιας ποσότητας:

$$\times(t_1) - x(t_0) / t_1 - t_0$$

Παρακάτω θα συναντήσουμε πολλές φορές υπολογισμούς τέτοιων ποσοτήτων.

#### **4.8.1. Υπολογισμός χρησιμοποίησης τμημάτων/συνδέσμων του δικτύου.**

Μια πιθανή εφαρμογή διαχείρισης επιδόσεων είναι ο υπολογισμός της χρησιμοποίησης ενός συνδέσμου του δικτύου. Ο υπολογισμός αυτός μπορεί να βοηθήσει, είτε βραχυπρόθεσμα στην απομόνωση κάποιας προβληματικής περιοχής, είτε μακροπρόθεσμα στο σχεδιασμό του δικτύου με σωστό υπολογισμό των απαραίτητων χωρητικοτήτων. Η εξέταση των χρησιμοποιήσεων των συνδέσμων είναι ένα πρώτο βήμα προς την πραγματοποίηση της διαχείρισης επιδόσεων.

Η χρησιμοποίηση ενός συνδέσμου σε ένα δίκτυο μπορεί να εξαρτάται από πολλούς παράγοντες, όπως το πρωτόκολλο για το στρώμα σύνδεσης δεδομένων, τους αλγόριθμους επαναμετάδοσης που χρησιμοποιούνται από τους διάφορους κόμβους, τις εφαρμογές που τρέχουν στο δίκτυο κ.ά. Όταν η χρησιμοποίηση ενός συνδέσμου γίνεται υψηλή, το γεγονός αυτό γίνεται άμεσα αντιληπτό στους χρήστες με την μορφή αργών χρόνων απόκρισης. Θα πρέπει λοιπόν να αντιμετωπιστεί.

Χρησιμοποιώντας αντικείμενα της MIB II μπορεί κανείς να υπολογίσει το ποσοστό χρησιμοποίησης για ένα μοναδικό μηχάνημα σε ένα μέσο, όπως ένα τμήμα Ethernet. Χρησιμοποιώντας τα ίδια αντικείμενα σε ένα full-duplex point-to-point σύνδεσμο μπορεί να υπολογίσει την χρησιμοποίηση του συνδέσμου.

Τα αντικείμενα ifInOctets και ifOutOctets δίνουν το συνολικό αριθμό bytes που έλαβε και έστειλε μια φυσική σύνδεση. Εξετάζοντας την μεταβολή των ποσοτήτων αυτών στην μονάδα του χρόνου, και διαιρώντας με την ανάλογη χωρητικότητα μπορεί να υπολογίσει ποσοστά χρησιμοποίησης. Η χωρητικότητα σε kilobits per second (Kbps) μιας φυσικής σύνδεσης δίνεται από το αντικείμενο ifSpeed. Για παράδειγμα, η χρησιμοποίηση σε μια Ethernet φυσική σύνδεση μπορεί να βρεθεί από την σχέση:

```
utilization = 8 * ((ifInOctets(t1)-ifInOctets(t0)) +
(ifOutOctets(t1)-ifOutOctets(t0)))/((t1-t0)*ifSpeed)
```

Ας τονίσουμε για μια ακόμη φορά, ότι η σχέση αυτή θα υπολογίσει την χρησιμοποίηση στην κάρτα του κόμβου, και όχι στο μέσο μεταφοράς που συνδέει τους κόμβους.

Σε full-duplex point-to-point μέσα μεταφοράς, η παραπάνω σχέση πρέπει να τροποποιηθεί κατάλληλα, λαμβάνοντας ως' όψη μονάχα τον μεγαλύτερο από τους δύο ρυθμούς εισόδου και εξόδου. Εάν δε γίνει αυτό μπορεί να υπολογιστεί ποσοστό χρησιμοποίησης 200% σε περίπτωση πλήρης χρήσης της διαθέσιμης χωρητικότητας και προς τις δύο κατευθύνσεις. Η ακόλουθη σχέση λοιπόν είναι κατάλληλη:

```
utilization = 8 * max(ifInOctets(t1)-ifInOctets(t0),
ifOutOctets(t1)-ifOutOctets(t0))/((t1-t0)*ifSpeed)
```

Με εξέταση του αντικειμένου ifType μπορεί να καθοριστεί το ποια σχέση πρέπει να χρησιμοποιηθεί σε κάθε περίπτωση. Παραδείγματα δυνατών full-duplex φυσικών συνδέσεων είναι: 2 (regular1822), 3 (hdh1822), 4 (ddn-x25), 5 (rfc877-x25), 16 (lapb), 17 (sdlc), 18 (dsl), 19 (el), 20 (basicISDN), 21 (primaryISDN), 22 (proprietaryPointToPointSerial), 23 (ppp), 28 (slip), 30 (ds3), 31 (smds), και 32 (frame-relay).

Για συγκεκριμένους τύπους μέσων μεταφοράς, όπως το X.25 ίσως έχει νόημα να κοιτάξει κανείς τη χρησιμοποίηση και στις δύο άκρες του νοητού κυκλώματος όπου τα τερματικά ενώνονται στο δίκτυο κορμού. Μετά, τα δεδομένα αυτά μπορούν να χρησιμοποιηθούν για να καθοριστεί αν τα φορτία από όλα τα τερματικά θα συμφορήσουν το δίκτυο.

Τα αντικείμενα στην MIB II επιτρέπουν να υπολογίσεις την χρησιμοποίηση μιας φυσικής σύνδεσης σε ένα multicast μέσο μεταφοράς και όχι τη χρησιμοποίηση σε ένα ολόκληρο τμήμα του μέσου μεταφοράς. Ο υπολογισμός της συνολικής χρησιμοποίησης μπορεί να υπολογιστεί με χρήση του αντικειμένου etherStatsOctets που βρίσκεται στο statistics group της RMON MIB. Το αντικείμενο etherStatsOctets δίνει το συνολικό αριθμό από octets που μεταδώθηκαν σε ένα προσαρτημένο στον κόμβο τμήμα Ethernet. Από τον αριθμό αυτό των octets μπορεί να υπολογίσει κανείς εύκολα την χρησιμοποίηση.

Για μακροπρόθεσμη διαχείριση επιδόσεων ενός τμήματος Ethernet, το History group της RMON MIB περιέχει το αντικείμενο etherHistoryUtilization (Το History group περιέχει πληροφορία που συλλέκτηκε από το Statistics group στο παρελθόν, για μετέπειτα ανάλυση). Το etherHistoryUtilization δίνει την καλύτερη εκτίμηση της μέσης χρησιμοποίησης του φυσικού μέσου για το χρονικό διάστημα για το οποίο υπάρχει πληροφορία.

#### 4.8.2. Διαχείριση καταστάσεων συμφόρησης

Ονομάζουμε **συμφόρηση (Congestion)** το σημείο εκείνο στο οποίο η ρυθμαπόδοση (throughput) του δικτύου πέφτει στο μηδέν και αυτό συμβαίνει γιατί το μέσο μεταφοράς δεν έχει τη διαθέσιμη χωρητικότητα, ώστε να μεταδώσει πληροφορία χωρίς να οδηγηθεί σε λάθος ή επαναμετάδοση. Τότε, τα πρωτόκολλα στα υψηλότερα επίπεδα (π.χ. TCP) επαναμεταδίουν την πληροφορία για την οποία δεν παίρνουν επιβεβαίωση και η κατάσταση συνεχώς χειροτερεύει. Συμφόρηση, μπορούμε ακόμα να ονομάσουμε την κατάσταση εκείνη στην οποία ο χρόνος απόκρισης γίνεται άπειρος.

Ελέγχοντας κανείς την χρησιμοποίηση των συνδέσμων σε ένα δίκτυο μπορεί να αναγνωρίσει ενδείξεις για μια πιθανή συμφόρησή του, οπότε είναι δυνατή η αποφυγή με αύξηση των χωρητικοτήτων ή επαναδρομολόγηση του φορτίου. Πέρα όμως από την χρησιμοποίηση υπάρχουν και άλλα αντικείμενα στην MIB II, που μπορούν να βοηθήσουν στην διάγνωση μιας κατάστασης συμφόρησης.

Κατ' αρχήν τα αντικείμενα ifInDiscards και ifOutDiscards περιέχουν τον αριθμό των IP datagrams που το σύστημα απόρριψε παρ' ότι δεν περιείχαν λάθη. Ένας λόγος για μια τέτοια απόρριψη είναι η έλλειψη χώρου στους καταχωρητές, όπου τα IP datagrams διατηρούνται πριν την εκπομπή τους ή μετά την λήψη τους. Πολλοί λόγοι υπάρχουν για μια τέτοια έλλειψη χώρου (π.χ. υψηλή χρησιμοποίηση του συστήματος, έλλειψη χωρητικότητας στο μέσο μεταφοράς κ.ά.). Μεγάλες τιμές στα δύο αντικείμενα αυτά σημαίνει μεγάλο αριθμό επαναμεταδόσεων, και μεγάλος αριθμός επαναμεταδόσεων σημαίνει χαμηλό throughput και πιθανώς συμφόρηση.

Άλλο χρήσιμο αντικείμενο είναι το ifOutQLen το οποίο μας δίνει το μήκος της ουράς της φυσικής σύνδεσης με το δίκτυο σε πακέτα. Μεγάλα μήκη ουρών σημαίνουν μεγάλες καθυστερήσεις για τα πακέτα, μεγάλες καθυστερήσεις δημιουργούν επαναμεταδόσεις, μείωση του throughput και πιθανή συμφόρηση.

Τέλος, ακόμα ψηλότερα στο επίπεδο μεταφοράς υπάρχουν πληροφορίες για το TCP και τις επαναμεταδόσεις. Συγκεκριμένα το αντικείμενο tcpRtoAlgorithm μας δίνει το αλγόριθμο που ελέγχει τις επαναμεταδόσεις, ο οποίος ίσως να σημαίνει πολλά για κάποιους διαχειριστές. Ενώ το αντικείμενο tcpRetransSegs δίνει τον αριθμό των TCP segments που έχουν επαναμεταδοθεί. Φυσικά η αύξηση αυτού του μετρητή σημαίνει χαμηλό throughput για τις εφαρμογές που χρησιμοποιούν το πρωτόκολλο και πιθανή συμφόρηση.

#### 4.8.3. Υπολογισμός ρυθμών και ποσοστών σφαλμάτων

Τα λάθη στους συνδέσμους ενός δικτύου επίσης μπορεί να επηρεάσουν τις επιδόσεις του δικτύου, όπως προηγούμενα η υψηλή χρησιμοποίηση. Τα λάθη σε ένα σύνδεσμο μπορούν να προκαλέσουν συμφόρηση, χαμηλό throughput, και μη επιθυμητούς χρόνους απόκρισης. Ο έλεγχος των λαθών σε ένα σύνδεσμο του δικτύου είτε σε πραγματικό χρόνο, είτε με ανάλυση πληροφορίας που έχει συλλεχθεί σε μακρό χρονικό διάστημα μπορεί να βοηθήσει στην αποφυγή και στην διόρθωση προβληματικών κατάστασεων.

Τα αντικείμενα ifInErrors και ifOutErrors της MIB II δίνουν τον αριθμό των λαθών σε μια κάρτα κατά την είσοδο και έξοδο πλαισίων από/προς το δίκτυο. Υπολογίζοντας κανείς την μεταβολή αυτών στο χρόνο μπορεί να υπολογίσει το ρυθμό των λαθών από και προς το δίκτυο.

```
input_error_rate = ifInErrors(t1) - ifInErrors(t0) / (t1 - t0)
output_error_rate = ifOutErrors(t1) - ifOutErrors(t0) / (t1 - t0)
```

Αξίζει να τονιστεί, ότι αυτό που έχει σημασία είναι το ποσοστό των λαθών στο συνολικό αριθμό των πακέτων που περνούν από την κάρτα και όχι ο απόλυτος αριθμός των λαθών. Ο αριθμός αυτός, όπως είδαμε και σε προηγούμενη παράγραφο μπορεί να υπολογιστεί σε κάθε κατεύθυνση χρησιμοποιώντας διάφορα αντικείμενα ανάλογα με το τι θέλει να λάβει υπ' όψην του κανείς.

Εξετάζοντας τους ρυθμούς των λαθών ξεχωριστά στις δύο κατευθύνσεις συχνά είναι χρήσιμο στην αναγνώριση σφαλμάτων. Λάθη κατά την είσοδο μπορεί να σημαίνουν προβλήματα με τα δεδομένα που λαμβάνονται (π.χ. πολύ μεγάλα ή πολύ μικρά πλαίσια ή θέματα συγχρονισμού κατά την μετάδοση). Ενώ σφάλματα κατά την έξοδο

μπορεί να σημαίνουν προβλήματα στο φυσικό μέσο μεταφοράς, χάσιμο του συγχρονισμού σε σειριακή γραμμή κ.ά.

Η RMON MIB περιέχει αντικείμενα στο Host group, τα οποία είναι χρήσιμα για τον υπολογισμό του ρυθμού λαθών κατά την έξοδο πακέτων από ένα σταθμό εργασίας. Τα αντικείμενα αυτά είναι τα hostOutPkts και hostOutErrors και είναι ιδιαίτερα χρήσιμα για την διαχείριση επιδόσεων σε ένα τμήμα, όπου μηχανήματα δεν έχουν SNMP agents.

Σε ψηλότερα επίπεδα το SNMP δίνει την δυνατότητα υπολογισμού ρυθμών λαθών σε όλα τα TCP/IP πρωτόκολλα, όπως IP, ICMP, TCP, UDP, EGP, SNMP. Με κατάλληλες αποφάσεις είναι και πάλι δυνατή η αποφυγή και η διόρθωση προβληματικών και άλλων ανεπιθύμητων κατάστασεων.

Τέλος, σε περιπτώσεις που είναι χρήσιμη η εξέταση των ρυθμών σφαλμάτων σε ολόκληρο το τμήμα ενός δικτύου, η χρήση της RMON MIB είναι απαραίτητη. Τα αντικείμενα αυτής στο Statistics group:

- etherStatsCRCAlignErrors
- etherStatsUndersizePkts
- etherStatsOversizePkts
- etherStatsFragments
- etherStatsJabbers
- etherStatsCollision

μας δίνουν όλες τις απαραίτητες πληροφορίες για κάποιο Ethernet τμήμα. Η εφαρμογή των παραπάνω αφήνεται σαν άσκηση. Τέλος υπάρχει και το History group για μακροπρόθεσμο σχεδιασμό και διαχείριση επιδόσεων.

#### **4.8.4. Αναγνώριση προτύπων για το φορτίο στο δίκτυο**

Η μορφή που έχει το φορτίο σε ένα δίκτυο μπορεί επίσης να επηρεάσει τις επιδόσεις ενός δικτύου. Στην παράγραφο αυτή θα εξετάσουμε δύο ιδιαίτερους παράγοντες του τύπου του φορτίου και την ροή του φορτίου μεταξύ δύο σταθμών εργασίας.

Ο τύπος του φορτίου σε κάποιο τμήμα του δικτύου μπορεί να είναι ένας από τους εξής: non-broadcast και broadcast. Το non-broadcast φορτίο κατευθύνεται προς ένα σταθμό εργασίας του τμήματος του δικτύου, ενώ το broadcast φορτίο κατευθύνεται προς όλους τους κόμβους σε ένα τμήμα ενός δικτύου. Είναι λοιπόν φανερό ότι συχνές broadcast εκπομπές μπορούν να μειώσουν επικίνδυνα τις επιδόσεις ενός δικτύου, δεδομένου μάλιστα, ότι απαιτούν και επεξεργασία από όλους του κόμβους του δικτύου. Επίσης σε διαφανή γεφυρωμένα δίκτυα το broadcast φορτίο θα περάσει σε όλα τα διασυνδεδεμένα τμήματα (βλ. και Κεφάλαιο 5).

Η MIB II δίνει δυνατότητες μέτρησης των ρυθμών του broadcast και του non-broadcast φορτίου σε ένα δίκτυο. Βέβαια στην συνέχεια θα πρέπει να υπάρχουν κατάλληλοι συναγερμοί ειδοποίησης του διαχειριστή του δικτύου, όταν οι ρυθμοί αυτοί ξεπεράσουν κάποια κατώφλια.

Έχουμε λοιπόν για το ρυθμό του broadcast φορτίου σε μια φυσική σύνδεση, χρησιμοποιώντας τα αντικείμενα ifInNUcastPkts και ifOutNUcastPkts της MIB II:

$$\text{broadcast\_rate} = (\text{ifInNUcastPkts}(t_1) - \text{ifInNUcastPkts}(t_0)) + \\ (\text{ifOutNUcastPkts}(t_1) - \text{ifOutNUcastPkts}(t_0)) / (t_1 - t_0)$$

Παρόμοια, έχουμε για το ρυθμό του non-broadcast φορτίου σε μια φυσική σύνδεση, αυτή την φορά, χρησιμοποιώντας τα αντικείμενα ifInUcastPkts και ifOutUcastPkts της MIB II:

$$\text{non\_broadcast\_rate} = (\text{ifInUcastPkts}(t_1) - \text{ifInUcastPkts}(t_0)) + \\ (\text{ifOutUcastPkts}(t_1) - \text{ifOutUcastPkts}(t_0)) / t_1 - t_0$$

To Host group στην RMON MIB επίσης δίνει πληροφορίες για τον τύπο του φορτίου που προσφέρει κάθε κόμβος σε ένα τμήμα του δικτύου. Με την χρήση των αντικειμένων hostOutBroadcastPkts και hostOutMulticastPkts μπορεί κανείς εύκολα να προσδιόρισει τον τύπο του φορτίου που στέλνει κάθε κόμβος.

Για να εξεταστεί η ροή του φορτίου σε ένα τμήμα του δικτύου, πρέπει να χρησιμοποιηθεί ο πίνακας matrix που βρίσκεται στο Matrix group της RMON MIB. Ο πίνακας αυτός δίνει για κάθε συνδυασμό διευθύνσεων πηγής και προορισμού αριθμών πακέτων (matrixSDPkts), αριθμό bytes (matrixSDOtets) και λαθών (matrixSDErrors) που ανταλάσσονται. Αυτοί οι μετρήτες είναι χρήσιμοι για την αναγνώριση κόμβων του δικτύου, που ελέγχουν την ροή του φορτίου (π.χ. δρομολογητές) και κόμβων του δικτύου που δημιουργούν το μεγαλύτερο μέρος του φορτίου (π.χ. file servers).

Η εξέταση του ρυθμού του φορτίου μεταξύ ζευγαριών από κόμβους μπορεί να βοηθήσει τον διαχειριστή του δικτύου σε μια μακροπρόθεσμη καλύτερη σχεδίαση του δικτύου, απομονώντας κάποιες περιοχές υψηλής χρησιμοποίησης από άλλες με την βοήθεια γεφυρών ή δρομολογητών. Για παράδειγμα, αν ένα ζεύγαρι κόμβων που επικοινωνεί πολύ συχνά χωρίζεται από κάποιο δρομολογητή, θα ήταν σωστό, εάν ήταν και δυνατόν, οι δύο κόμβοι να βρεθούν στο ίδιο υποδίκτυο, ώστε το μεταξύ τους φορτίο να μην χρειάζεται να φορτώνει κάποιον δρομολογητή.

#### 4.9. Ασφάλεια: απειλές και μηχανισμοί

Προκειμένου να υλοποιηθούν χρήσιμες και αποτελεσματικές συναρτήσεις διαχείρισης βλαβών ή διάρθρωσης θα πρέπει να υπάρχει κατάλληλος μηχανισμός για την πιστοποίηση της αυθεντικότητας των μηνυμάτων SNMP. Πολλοί vendors έχουν αποφασίσει να μην υλοποιούν την εντολή set-request με την οποία κάποιος μπορεί μεταξύ των άλλων να κάνει reboot ή να κατεβάσει κάποιο κόμβο. Και αυτό γιατί μέχρι τώρα χρησιμοποιείται ένα τετριμένο σχήμα για να βρεθεί αν κάποιο SNMP μήνυμα είναι αυθεντικό. Στο σχήμα αυτό το πεδίο δεδομένων στο μήνυμα SNMP είναι ακριβώς το PDU, αν και θα μπορούσε να υλοποιηθεί κάποιο service το οποίο χρησιμοποιώντας την τεχνική της κρυπτογράφησης (encryption), ή κάποια άλλη να μπορεί να προσφέρει μεγαλύτερη ασφάλεια.

Το μόνο που κάνει η συνάρτηση πιστοποίησης είναι να ελέγχει αν το όνομα της κοινότητας (community name) που περιέχει το μήνυμα SNMP είναι γνωστό ή όχι στον agent. Σε καλύτερες υλοποιήσεις μπορεί σχετικά με μια κοινότητα να υπάρχει μια λίστα από εξουσιοδοτημένες διευθύνσεις (IP διεύθυνση - UDP port) που μπορούν να στείλουν γνήσια SNMP μηνύματα.

Μερικοί κατασκευαστές έχουν υλοποιήσει δικά τους σχήματα ασφάλειας και πιστοποίησης του γνησίου των μηνυμάτων, προσπαθώντας να ξεπεράσουν το όλο πρόβλημα. Πάντως ειδική ομάδα της IAB αναπτύσσει ένα standard κρυπτογράφησης των δεδομένων και ένα σχήμα για τη διανομή των απαραίτητων κλειδιών, έτσι ένα RFC αναμένεται πολύ σύντομα να κυκλοφορήσει. Προσπάθεια που σίγουρα θα δώσει μεγαλύτερες δυνατότητες στο SNMP.

Πιθανές απειλές στο σχήμα ασφαλείας που υποστηρίζει το SNMP είναι:

- Η δημιουργία ενός SNMP μηνύματος από μη υπεύθυνο διαχειριστή
- Η μεταβολή ενός SNMP μηνύματος κατά τη μεταφορά του.
- Μια πιθανή επαναμετάδοση παλαιότερου νόμιμου SNMP μηνύματος.
- Μια χωρίς δικαιώμα αποκάλυψη του περιεχομένου ενός SNMP μηνύματος.

Οι παραπάνω απειλές μας οδηγούν στις προδιαγραφές απαιτήσεων από ένα νέο σχήμα ασφάλειας.

- Πιστοποίηση πηγής μηνύματος.
- Ακεραιότητα μηνύματος κατά τη μεταφορά του.
- Εξασφάλιση μη επαναμετάδοσης μηνύματος.
- Μυστικότητα σχετικά με το περιεχόμενο ενός μηνύματος.

Οι μηχανισμοί που προτείνονται από επιτροπές (υπεύθυνες για τη βελτίωση του SNMP) είναι ο αλγόριθμος MD5 digest, το πρότυπο απόκρυψης δεδομένων (Data Encryption Standard, DES) και η χρήση περίπου συγχρονισμένων ρολογιών. Οι MD5 και DES χρησιμοποιούνται σαν συμμετρικοί κρυπτογραφικοί αλγόριθμοι (δηλαδή χρησιμοποιούν κάποια κρυφή τιμή που πρέπει να γνωρίζει και η πηγή του μηνύματος SNMP και ο προορισμός). Τέλος η χρήση περίπου συγχρονισμένων ρολογιών επιτρέπει στη πηγή να βάλει κάποιο timestamp στο μήνυμα SNMP ώστε ο προορισμός να διαπιστώσει ότι το μήνυμα έφτασε σε κάποιο προκαθορισμένο χρονικό διάστημα.

## 4.12. Ασκήσεις

- [1]. Σχεδιάστε κάποιο πρόγραμμα το οποίο με τη βοήθεια του SNMP πρωτοκόλλου θα εξετάζει τον πίνακα δρομολόγησης η δρομολογητών, προκειμένου να βρει πιθανά routing loops.
- [2]. Εξετάστε τη MIB II στο παράρτημα. Τι στοιχεία καθορίζει ότι πρέπει να διατηρούνται για κάθε διασύνδεση με το δίκτυο (network interface);
- [3]. Υποθέστε ότι τα αντικείμενα της MIB I ήταν αριθμημένα από 1 έως 89 αντί να έχουν κάποια τιμή του object identifier τύπου του ASN.1 σαν αναγνωριστικό τους. Ποια θα ήταν τα πλεονεκτήματα και ποια τα μειονεκτήματα;
- [4]. Τι αναπαριστά το πεδίο community-name σε ένα μήνυμα SNMP (v1); Ποια άλλα πεδία του SNMP-pdu γνωρίζετε;
- [5]. Πόσο χρόνο πρέπει κάποιος SNMP-client να περιμένει για μία απάντηση από κάποιο agent; Πόσες φορές πρέπει να επαναμεταδίδει κάποιο μήνυμα για το οποίο δεν έλαβε απάντηση;
- [6]. Εξετάστε τη MIB II στο παράρτημα. Καθορίζονται σ' αυτήν μεταβλητές οι οποίες μπορούν να πάρουν τιμές από ένα συγκεκριμένο σύνολο (π.χ. ipForwarding); Εάν ναι, δώστε παραδείγματα τέτοιων μεταβλητών και τις τιμές που αυτές μπορούν να πάρουν. Τι θα πρέπει κάποιος agent να ελέγχει όταν δέχεται κάποιο SNMP μήνυμα set-request για μια τέτοια μεταβλητή;
- [7]. Πρέπει κάποιος διαχειριστής να μπορεί να θέσει (set) τη φυσική διεύθυνση κάποιου μηχανήματος σε μια επιθυμητή τιμή; Γιατί ναι ή γιατί όχι;
- [8]. Ποιος είναι ο ακριβής object identifier τύπος για κάποια εγγραφή στο πεδίο ipRouteNextHop στο ipRoutingTable για κάποια IP διεύθυνση 147.102.1.1;
- [9]. Χρησιμοποιώντας κάποια εφαρμογή SNMP (π.χ. snmpget του CMU-SNMP) να υπολογιστεί η χρησιμοποίηση (% utilization) στο επίπεδο interface προς και από τον host dolly.ntua.gr. Προτείνατε διάφορες τεχνικές για τη μέτρηση αυτή, ώστε το τελικό αποτέλεσμα να παρουσιάζει κάποιο νόημα για ένα διαχειριστικό

σύστημα. Ενδιαφέρον παρουσιάζουν θέματα όπως ο αριθμός των δειγμάτων, η συχνότητα της δειγματοληψίας, κ.ά.

- [10]. Χρησιμοποιώντας κάποια εφαρμογή SNMP (π.χ. snmpget του CMU-SNMP) να υπολογιστεί ο ρυθμός απόδοσης (throughput σε bytes/sec) στο επίπεδο interface προς και από τον host dolly.ntua.gr. Προτείνατε διάφορες τεχνικές για τη μέτρηση αυτή, ώστε το τελικό αποτέλεσμα να παρουσιάζει κάποιο νόημα για ένα διαχειριστικό σύστημα. Ενδιαφέρον παρουσιάζουν θέματα όπως ο αριθμός των δειγμάτων, η συχνότητα της δειγματοληψίας, κ.ά.
  
- [11]. Υπάρχει τρόπος να υπολογίσετε (έστω και προσεγγιστικά) το λόγο του φορτίου που δημιουργεί το SNMP πρωτόκολλο στο δίκτυο προς το συνολικό φορτίο του δικτύου.
  - α) Για τον υπολογισμό χρησιμοποιείστε τα στοιχεία της MIB II.
  - β) Θεωρήστε δεδομένο ότι η διαχειριστική εφαρμογή μετρά το ρυθμό απόδοσης (throughput σε bytes/sec) στο επίπεδο interface προς και από κάποιο host κάθε 5 sec.

Κατά τη γνώμη σας πόσο θα έπρεπε να ήταν το ποσοστό αυτό;
  
- [12]. Θεωρήστε το παρακάτω σενάριο: σε ένα τοπικό δίκτυο (LAN) κάθε διαχειριζόμενο αντικείμενο δέχεται ερώτηση από το διαχειριστικό σύστημα κάθε 15 min. Αν υποθέσουμε ότι ο χρόνος επεξεργασίας του μηνύματος, πριν την αποστολή από τον manager, κατά την αποδοχή στον agent, κατά την αποστολή της απάντησης από τον agent και κατά την αποδοχή της απάντησης από τον manager είναι ο ίδιος και ίσος με 50 ms, ενώ ο χρόνος μεταφοράς του μηνύματος μέσα από το δίκτυο είναι 1 ms. Ποιος είναι ο μέγιστος αριθμός αντικειμένων που μπορεί να διαχειριστεί το σύστημα διαχείρισης; Επαναλάβετε το σενάριο σε ένα δίκτυο ευρείας περιοχής (WAN), όπου οι παραπάνω χρόνοι είναι αντίστοιχα 15 min, 50 ms, 500 ms. Ποια είναι τα συμπεράσματά σας;
  
- [13]. Εξηγήστε και συγκρίνατε τις παρακάτω τεχνικές διαχείρισης δικτύων υπολογιστών:
  - α) Polling-based management, και
  - β) event-based management.

Γνωρίζετε πρωτόκολλα που να υποστηρίζουν τη μία ή την άλλη τεχνική;
  
- [14]. Χρησιμοποιώντας κάποια εφαρμογή SNMP (π.χ. snmptrap & snmpget του CMU-SNMP) προσπαθήστε να στείλετε κάποιο trap-pdu στον κόμβο

dolly.ntua.gr και στην συνέχεια εξετάστε αν ο κόμβος το έλαβε. Εξηγήστε το αποτέλεσμα του πειράματος.

- [15]. Προτείνετε ιδέες (π.χ. αλγορίθμους) για την αναβάθμιση του πλαισίου ασφάλειας που υποστηρίζει το SNMP. Σας υπενθυμίζουμε ότι τα SNMP μηνύματα καθορίζονται και κωδικοποιούνται σύμφωνα με τα πρότυπα ASN.1 - BER.
- [16]. Μέσα στις προδιαγραφές της MIB II μήπως περιέχεται κάποια τεχνική περιορισμού του φορτίου που δημιουργείται από τα SNMP-trap-pdus.
  - α) Ποιο είναι το προαπαιτούμενο για να μπορεί να εφαρμοστεί η παραπάνω τεχνική;
  - β) Για ποιο λόγο επιτρέπεται ο περιορισμός μονάχα του συγκεκριμένου trap;
- [17]. Σε MIB διαφόρων κόμβων ενός διασυνδεδεμένου δικτύου βρέθηκαν τα παρακάτω στοιχεία (με Name δίνεται το όνομα του στιγμιότυπου στην MIB και με ipAddress η τιμή του):
  - Στον κόμβο 147.102.12.3,  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3  
ipAddress: 147.102.12.1
  - Στον κόμβο 147.102.12.1,  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3  
ipAddress: 147.102.1.1
  - Στον κόμβο 147.102.1.1,  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3  
ipAddress: 139.42.1.1
  - Στον κόμβο 139.42.1.1,  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3  
ipAddress: 139.42.10.1
  - Στον κόμβο 139.42.10.1,  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3  
ipAddress: 139.42.1.1

Στον κόμβο 147.102.12.3,

```
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3
ipAddress: 147.102.12.1
```

Στον κόμβο 147.102.12.1,

```
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3
ipAddress: 147.102.1.1
```

Στον κόμβο 147.102.1.1,

```
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3
ipAddress: 139.42.1.1
```

Στον κόμβο 139.42.1.1,

```
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3
ipAddress: 139.42.10.1
```

Στον κόμβο 139.42.10.1,

```
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.139.42.10.3
ipAddress: 139.42.1.1
```

- α) Παρατηρείται κάποιο πρόβλημα;
- β) Με ποιο τρόπο μπορεί να λυθεί το συγκεκριμένο πρόβλημα;
- γ) Περιγράψτε μηχανισμούς με τους οποίους θα ήταν δυνατή η έγκαιρη διάγνωση τέτοιων σφαλμάτων.
- [18]. Θεωρείται ότι μια εφαρμογή διαχείρισης TCP/IP δικτύων βασισμένη στο SNMP ακολουθεί την αρχιτεκτονική client-server; Περιγράψτε τις αντιστοιχίες που παρουσιάζονται.
- [19]. Χρησιμοποιώντας κάποια εφαρμογή SNMP (π.χ. snmpget του CMU-SNMP) να υπολογιστεί ο ρυθμός απόδοσης (throughput σε pkts/sec, πακέτα το δευτερόλεπτο) στο επίπεδο interface προς και από τον host dolly.ntua.gr. Προτείνατε διάφορες τεχνικές για τη μέτρηση αυτή, ώστε το τελικό αποτέλεσμα να παρουσιάζει κάποιο νόημα για ένα διαχειριστικό σύστημα. Ενδιαφέρον παρουσιάζουν θέματα όπως ο αριθμός των δειγμάτων, η συχνότητα της δειγματοληψίας, κ.ά.
- [20]. Χρησιμοποιώντας κάποια εφαρμογή SNMP (π.χ. snmpget του CMU-SNMP) να υπολογιστεί η πιθανότητα άφιξης λανθασμένου πακέτου και εύρεσης λανθασμένου πακέτου κατά τη μετάδοση στο επίπεδο interface του host dolly.ntua.gr. Να υπολογιστεί επίσης ο ρυθμός των παραπάνω λαθών (σε λανθασμένα πακέτα το δευτερόλεπτο). Συγκρίνατε τα δύο μεγέθη (δηλ. πιθανότητα και ρυθμό) και αναφέρατε που θα μπορούσε να χρησιμοποιηθεί καλύτερα το καθένα.
- [21]. Χρησιμοποιώντας κάποια εφαρμογή SNMP (π.χ. snmpget του CMU-SNMP) να υπολογιστεί η πιθανότητα απόρριψης πακέτου στο επίπεδο interface προς και από τον host dolly.ntua.gr. Να υπολογιστεί επίσης ο ρυθμός των παραπάνω απορρίψεων (σε πακέτα που απορρίπτονται το δευτερόλεπτο). Συγκρίνατε τα δύο μεγέθη (δηλ. πιθανότητα και ρυθμό) και αναφέρατε που θα μπορούσε να χρησιμοποιηθεί καλύτερα το καθένα.
- [22]. Στείλτε ένα SNMP πακέτο (π.χ. get-request pdu) σε διάφορους κόμβους χρησιμοποιώντας community name διαφορετικό από public. Τι περιμένετε να παρατηρήσετε και τι παρατηρείται; Ποια εξήγηση δίνεται;

- [23]. Δώστε παραδείγματα παραμέτρων πρωτοκόλλων οι οποίες θα μπορούσαν να διαπραγματευθούν όταν εγκαθίσταται μια σύνδεση και οι οποίες θα ήταν ενδιαφέρουσες για τη διαχείριση.
- [24]. Προτείνετε τεχνικές διαχείρισης κάποιας γέφυρας ή κάποιου δρομολογητή χωρίς/με IP διεύθυνση.
- [25]. Σχεδιάστε ένα MIB (Management Information Base) (απλοποιημένο) για το δίκτυο του Πολυτεχνείου όπου ένας αριθμός από τοπικά δίκτυα διασυνδέονται μεταξύ τους με δρομολογητές, γέφυρες, κ.ά.

#### 4.13. Βιβλιογραφία

- [ALAR92] Information processing systems - Open System Interconnection - *Systems Management: Alarm reporting function* - International Organization for Standardization - International Standard 10164-4 - December 1992.
- [ARTZ90] Amatzia Ben-Artzi, Asheem Chandna, Unni Warrier, "Network Management of TCP/IP Networks: Present and Future", IEEE Network Magazine, July 1990.
- [ASN187] Information processing systems - Open System Interconnection - *Specification of Abstract Syntax Notation One (ASN.1)* - International Organization for Standardization - International Standard 8824 - December 1987.
- [BER\_87] Information processing systems - Open System Interconnection - *Specification of Basic Encoding Rules for Abstract Notation One (ASN.1)* - International Organization for Standardization - International Standard 8825 - December 1987.
- [CASE90] J. D. Case, M. S. Fedor, M. L. Schoffstall, and J. R. Davin, "A Simple Network Management Protocol (SNMP)", RFC 1157, SNMP Research, PSI and MIT Laboratory for Computer Science, May 1990.
- [CASS89] Lillian N. Cassel, Graig Partridge, and Jil Westcott, "Network Management Architectures and Protocols: Problems and Approaches", IEEE Journal On Selected Areas In Communicationns, Vol. 7, No. 7, September 1989.
- [CMIP91] Information processing systems - Open System Interconnection - *Common Management Information Protocol (CMIP)* - International Organization for Standardization - International Standard 9596 - June 1991.
- [CMIS91] Information technology - Open System Interconnection - *Common Management Information Service (CMIS)* - International Organization for Standardization - International Standard 9595 - April 1991.
- [DEMI92] Information processing systems - Open System Interconnection - *Structure of management information: Definition of management information* - International Organization for Standardization - International Standard 10165-2 - October 1992.
- [EMBR90] Jock Embry, Peter Manson, Dave Milham, "An Open Network Management Architecture: OSI/NM Forum Architecture and Concepts", IEEE Network Magazine, July 1990.
- [FRAID91] Fraidoon Mazda, "Convergence or Collision: SNMP and CMIP (Part 1)", Data Communications, September 1991.
- [HERM90] James Herman, "Enterprise Management Vendors Shoot It Out", Data Communications International, November 1990.

- [KRAL90] Gary Krall, "SNMP Opens New Lines of Sight", Data Communications, March 21, 1990.
- [LOAD92] Information processing systems - Open System Interconnection - *Systems Management: Workload monitoring function* - International Organization for Standardization - Draft International Standard 10164-11 - November 1992.
- [MCCL90] McCloghrie K., and M. Rose, "*Structure and Identification of Management Information for TCP/IP - based Internets*", RFC 1155, Performance Systems International and Hughes LAN Systems, May 1990.
- [MCCL91] McCloghrie K., and M. Rose, "*Management Information Base for Network Management of TCP/IP - based Internets, MIB II*", RFC 1213, Performance Systems International and Hughes LAN Systems, March 1991.
- [MODI91] N. Modiri, "An Implementation of the Common Network Management Information Service Element Interfaces", IEEE Communications Magazine, July 1991.
- [OSIM89] Information processing systems - Open System Interconnection - *OSI Management Framework* - International Organization for Standardization - International Standard 7498/4 - April 1989.
- [PRES90] Randy Presuhn, "Considering CMIP", Data Communications, March 21, 1990.
- [ROSE91] Marshall T. Rose, *The Simple Book: An Introduction to Management of TCP/IP - based Internets*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [SCOK90] Karyl Scott, "Taking Care of Business with SNMP", Data Communications, March 21, 1990.
- [SCOT90] Karyl Scott, "Order to Chaos", Data Communications, March 21, 1990.
- [SECU92] Information processing systems - Open System Interconnection - *Systems Management: Security alarm reporting function* - International Organization for Standardization - International Standard 10164-7 - May 1992.
- [SMGO92] Information processing systems - Open System Interconnection - *Systems Management overview* - International Organization for Standardization - International Standard 10040 - November 1992.
- [ST1192] "The Simple Times", The Bi-Monthly Newsletter of SNMP Technology, Comment, and Events, Volume 1, Number 1, March/April 1992.
- [ST1292] "The Simple Times", The Bi-Monthly Newsletter of SNMP Technology, Comment, and Events, Volume 1, Number 2, May/June 1992.

- [ST1392] "The Simple Times", The Bi-Monthly Newsletter of SNMP Technology, Comment, and Events, Volume 1, Number 3, July/August 1992.
- [ST1492] "The Simple Times", The Bi-Monthly Newsletter of SNMP Technology, Comment, and Events, Volume 1, Number 4, September/October 1992.
- [ST1592] "The Simple Times", The Bi-Monthly Newsletter of SNMP Technology, Comment, and Events, Volume 1, Number 5, November/December 1992.
- [ST2193] "The Simple Times", The Bi-Monthly Newsletter of SNMP Technology, Comment, and Events, Volume 2, Number 1, January/February 1993.
- [ST2293] "The Simple Times", The Bi-Monthly Newsletter of SNMP Technology, Comment, and Events, Volume 2, Number 2, March/April 1993.
- [STAM92] Stamatelopoulos F., Stathatos K., Karounos T. & Maglaris B., "Cerberus Network Management System", ERSIM International Workshop, Crete, Greece, October 1992.
- [STAL93] Stallings, W. SNMP, SMMPv2, & CMIP: The Practical Guide to Network Management Standards, Addison-Wesley Publishing Company, Incorporated, 1993

## Κεφάλαιο 5

### 5. Διαχείριση OSI

#### Περιεχόμενα του Κεφαλαίου 5

- 5.0. Εισαγωγή στη Διαχείριση OSI
- 5.1. Διαχείριση δικτύων βασισμένη στο πρότυπο OSI
  - 5.1.1. Πρότυπα Διαχείρισης OSI
    - 5.1.1.1. Αρχιτεκτονική και Δομή
    - 5.1.1.2. Μεταφορά της Πληροφορίας Διαχείρισης
    - 5.1.1.3. Δομή της Πληροφορίας Διαχείρισης
    - 5.1.1.4. Λειτουργίες Διαχείρισης Συστημάτων
  - 5.1.2. Περιοχές Λειτουργιών Διαχείρισης OSI
    - 5.1.2.1. Διαχείριση Σφαλμάτων (Fault Management)
    - 5.1.2.2. Λογιστική Διαχείριση (Accounting Management)
    - 5.1.2.3. Διαχείριση Διάρθρωσης (Configuration Management)
    - 5.1.2.4. Διαχείριση Επιδόσεων (Performance Management)
    - 5.1.2.5. Διαχείριση Ασφάλειας (Security Management)
- 5.2. Διαχείριση Συστημάτων
  - 5.2.1. Μοντέλο Διαχείρισης Συστημάτων
    - 5.2.1.1. Θέματα Πληροφορίας
    - 5.2.1.2. Λειτουργικές Πλευρές
    - 5.2.1.3. Θέματα OSI Επικοινωνιών
    - 5.2.1.4. Οργανωτικές Πλευρές
- 5.3. Τα πρωτόκολλα CMIP/CMIS
  - 5.3.1. Common Management Information Service
  - 5.3.2. Common Management Information Protocol
- 5.4. Βάση Πληροφορίας Διαχείρισης OSI (OSI-MIB)
  - 5.4.1. Μοντέλο Πληροφορίας Διαχείρισης
  - 5.4.2. Ορισμός της Πληροφορίας Διαχείρισης
  - 5.4.3. GDMO (Guidelines for the Definition of Managed Objects)
  - 5.4.4. Πρακτικά ζητήματα

- 5.5. Περιγραφή των λειτουργιών διαχείρισης συστημάτων
- 5.6. Το όφελος από μια τυποποιημένη στοίβα πρωτοκόλλων.
- 5.7. Σύγκριση μεταξύ των SNMP και CMIP
- 5.8. Το πρωτόκολλο CMOT
- 5.9. Ασκήσεις
- 5.10. Βιβλιογραφία

## 5.0. Εισαγωγή στη Διαχείριση OSI

Η διαχείριση OSI αποτελείται από λειτουργίες ελέγχου, συντονισμού και παρακολούθησης των στοιχείων ενός δικτύου, τα οποία προσφέρουν τη δυνατότητα επικοινωνίας σε ένα ανοικτό περιβάλλον. Υπεύθυνοι για τη διαχείριση του περιβάλλοντος OSI είναι, κατά κύριο λόγο, οι χειριστές, αλλά συχνά αυτοματοποιημένες διαδικασίες μπορούν να αναλάβουν κάποιες ευθύνες. Η διαχείριση OSI επιτυγχάνεται μεταξύ ανοικτών συστημάτων μέσω της συνεργασίας δύο ή περισσοτέρων οντοτήτων από τις οποίες άλλες αναλαμβάνουν διαχειριστικό και άλλες διαχειριζόμενο ρόλο.

Πριν προχωρήσουμε στην μελέτη του σχεδιασμού της διαχείρισης OSI ας δούμε τις απαιτήσεις του χρήστη από αυτήν. Τέτοιες απαιτήσεις είναι [OSIM89]:

- Λειτουργίες που επιτρέπουν στο διαχειριστή το σχεδιασμό, την οργάνωση, την επίβλεψη, τους λογιστικούς υπολογισμούς και τον έλεγχο υπηρεσιών διασύνδεσης.
- Δυνατότητα απόκρισης σε πιθανές αλλαγές των αναγκών.
- Ευκολίες διασφάλισης μιας προβλεπόμενης συμπεριφοράς.
- Ευκολίες προστασίας της πληροφορίας και αναγνώρισης της αυθεντικότητας πηγών και προορισμών μεταδιδόμενων δεδομένων.

## 5.1. Διαχείριση δικτύων βασισμένη στο πρότυπο OSI

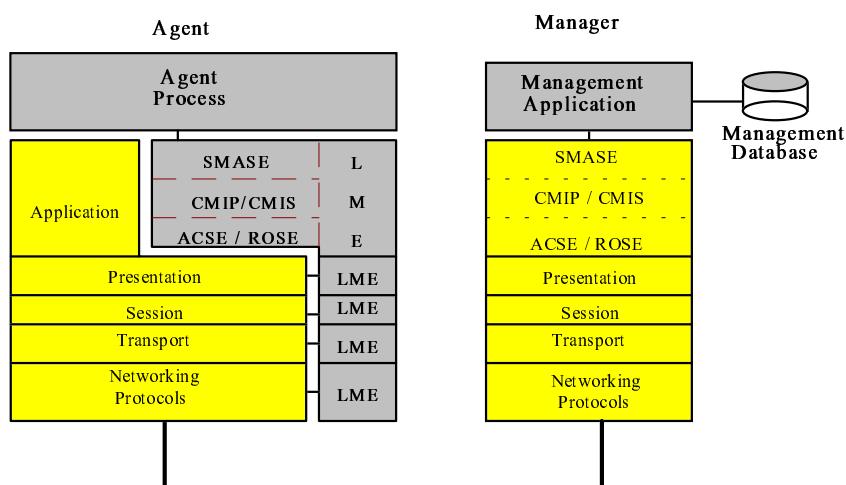
Το γενικό πλαίσιο διαχείρισης OSI ορίσθηκε για να παρέχει τα μέσα για την παρακολούθηση και τον έλεγχο των τηλεπικοινωνιακών υποσυστημάτων σε όλους τους κόμβους τους συνδεδεμένους σε ένα δίκτυο. Το γενικό πλαίσιο αυτό ορίζει ένα σύνολο από ομάδες λειτουργιών διαχείρισης, συχνά επικαλυπτόμενες, οι οποίες αποτελούν τις απαραίτητες δραστηριότητες για μια ουσιαστική διαχείριση, όπως επίσης και ένα μοντέλο που περιγράφει την συνολική δομή που πρέπει να έχει ένα σύστημα διαχείρισης. Για την καλύτερη λειτουργία της όλης λογικής, το πλαίσιο διαχείρισης χωρίζεται σε υποπεριοχές αρμοδιότητας (domains), σε κάθε μια από τις οποίες υπάρχει κάποιος υπεύθυνος διαχειριστής. Η γενική δομή του προτεινόμενου πλαισίου για τη διαχείριση δικτύων OSI έχει ως εξής:

Σε κάθε domain υπάρχουν υλοποιημένες οντότητες που στο εξής θα τις ονομάζουμε **System Management Application Entities (SMAEs)** και οι οποίες αποτελούν ένα ανοικτό interface για κάθε διαχειριστικό σύστημα. Με το interface αυτό κάθε διαχειριστικό σύστημα μπορεί να προσεγγίσει τα στοιχεία που θέλει να διαχειριστεί.

Προκειμένου να επιτευχθεί αυτό, η οντότητα αυτή επικοινωνεί με ένα σύνολο από άλλες SMAE οντότητες, κάτι αντίστοιχο με τους agents στο SNMP, οι οποίες αποτελούν το αντίστοιχο interface αντίστοιχα για τα διαχειριζόμενα σύστηματα. Η δομή μιας ολοκληρωμένης οντότητας SMAE φαίνεται στο σχήμα 5.1. Αποτελείται από τα εξής επιμέρους στοιχεία. Το **System Management Application Service Element (SMASE)**, το **Common Management Information Service Element (CMISE)**, και τα **Association Control Service Element (ACSE)** και **Remote Operation Service Element (ROSE)**. Όλα τα πρωτόκολλα αυτά, θα τα εξετάσουμε αναλυτικότερα σε επόμενες παραγράφους.

Από αυτά, το πρωτόκολλο CMISE είναι το ισοδύναμο με το SNMP στη διαχείριση TCP/IP δικτύων, και παρέχει μόνο τις απαραίτητες υπηρεσίες για την ανταλλαγή μηνυμάτων σχετικά με τη διαχείριση. Μια οντότητα CMISE αποτελεί μια υλοποίηση των προτύπων του ISO CMIS και CMIP, το πρώτο από τα οποία καθορίζει τις υπηρεσίες που μπορεί να προσφέρει κάποιο CMISE στις διαχειριστικές εφαρμογές που θα το χρησιμοποιήσουν, ενώ το δεύτερο καθορίζει τον τρόπο μεταφοράς της πληροφορίας που αφορά τις παραπάνω υπηρεσίες (τα PDUs).

Κάθε SMAE μπορεί να αλληλεπιδρά με τα διάφορα στρώματα (layers) πρωτοκόλλων του τηλεπικοινωνιακού υποσυστήματος στον κόμβο στον οποίο βρίσκεται, και να διεκπεραιώνει με τον τρόπο αυτό λειτουργίες που απαιτεί το υπεύθυνο διαχειριστικό σύστημα. Προκειμένου να επιτευχθεί αυτό, σε κάθε στρώμα έχουμε κάποιες προεκτάσεις τις οποίες μπορούμε να ονομάσουμε LME (Layer Management Entity). Αυτές είναι προεκτάσεις στις υλοποιήσεις των μηχανών καταστάσεων των πρωτοκόλλων κάθε στρώματος, με λίγα λόγια δηλαδή κάποια παραπάνω στοιχεία υπηρεσίας για κάθε στρώμα. Κάθε LME από αυτά "ξέρει" ή μπορεί να μάθει πληροφορίες" σχετικά με το πρωτόκολλο που λειτουργεί στο στρώμα το οποίο ελέγχει.



**Σχήμα 5.1 - OSI διαχείριση**

Το μεγαλύτερο μέρος της ευφυίας σχετικά με τη διαχείριση μπορεί να συγκεντρωθεί στις συναρτήσεις διαχείρισης που χρησιμοποιεί ο διαχειριστής, έτσι ώστε οι υλοποιήσεις του πρωτοκόλλου CMISE να έχουν μικρή πολυπλοκότητα και μονάχα ένας μικρός αριθμός από καινούργια στοιχεία υπηρεσίας να χρειάζεται να προστεθούν στα πρωτόκολλα κάθε στρώματος. Παρ' όλα αυτά όμως η απλότητα των CMIS/CMIP δεν πλησιάζει την απλότητα των SNMP agents.

Ένα σημαντικό πρόβλημα του παραπάνω μοντέλου διαχείρισης είναι ότι δεν αντιμετωπίζει πλήρως το θέμα κόμβων που δεν υλοποιούν και τα επτά στρώματα του OSI. Και εδώ όπως και στο SNMP δίνεται η δυνατότητα για proxy agents. Πιο συγκεκριμένα το μοντέλο OSI προτείνει δύο λύσεις για το συγκεκριμένο πρόβλημα.

- a) Την υλοποίηση των ελάχιστων εκείνων μηχανισμών που χρειάζονται για την λειτουργία και των επτά στρωμάτων σε ένα thin stack. Η λύση αυτή δουλεύει καλά σε μηχανήματα που δεν έχουν πρόβλημα περιορισμένης μνήμης όπως routers, αντίθετα όμως παρουσιάζονται προβλήματα σε μηχανήματα όπως για παράδειγμα ένα modem, όπου η μνήμη δεν είναι αρκετή για τον επιπρόσθετο κώδικα.
- β) Η δεύτερη προσέγγιση χρησιμοποιεί ειδικούς κόμβους τους οποίους ονομάζει proxy και translator. Σύμφωνα με την προσέγγιση αυτή μηχανήματα που δεν υλοποιούν και τα επτά στρώματα του OSI μπορούν να διαχειριστούν με την βοήθεια άλλων μηχανημάτων με την χρησιμοποίηση του κατάλληλου (N)-στρώματος πρωτοκόλλου. Οι proxy κόμβοι αναλαμβάνουν τον αποκλειστικό έλεγχο κάποιου μηχανήματος, ενώ translators μπορούμε να ονομάσουμε μια ομάδα κόμβων από τους οποίους μπορεί να ζητηθεί κάθε στιγμή η άσκηση ελέγχου.

Από όλα τα παραπάνω γίνεται φανερό ότι: όσο αναφορά την διάρθρωση το γενικό πλαίσιο αναφοράς OSI για την διαχείριση δικτύων προσφέρει ένα κατανοητό πλαίσιο για μια κατανεμημένη διαχείριση δικτύων και εφαρμογών. Η κατανεμημένη διαχείριση αποτελείται από μια ιεραρχία διαχειριστικών συστημάτων, όπου κάποια κατώτερα στάδια ενεργούν για λογαριασμό κάποιων υψηλοτέρων. Επίσης άλλο ένα χαρακτηριστικό της κατανεμημένης διαχείρισης είναι η δυνατότητα να υποστηρίζει ένα πλήθος από διαχειριστικές εφαρμογές.

Στην συνέχεια θα παρουσιάσουμε τα πρότυπα αυτά.

### 5.1.1. Πρότυπα Διαχείρισης OSI

Η παράγραφος αυτή περιγράφει τα διάφορα πρότυπα για την διαχείριση OSI, καθώς και τις σχέσεις που υφίστανται μεταξύ τους. Το Σχήμα 5.2 διευκρινίζει τις σχέσεις αυτές. Επίσης δείχνει και άλλα πρότυπα τα οποία παρέχουν πληροφορίες για τη διαχείριση, καθώς και τον τρόπο που αυτά συσχετίζονται με τα πρότυπα διαχείρισης συστημάτων.

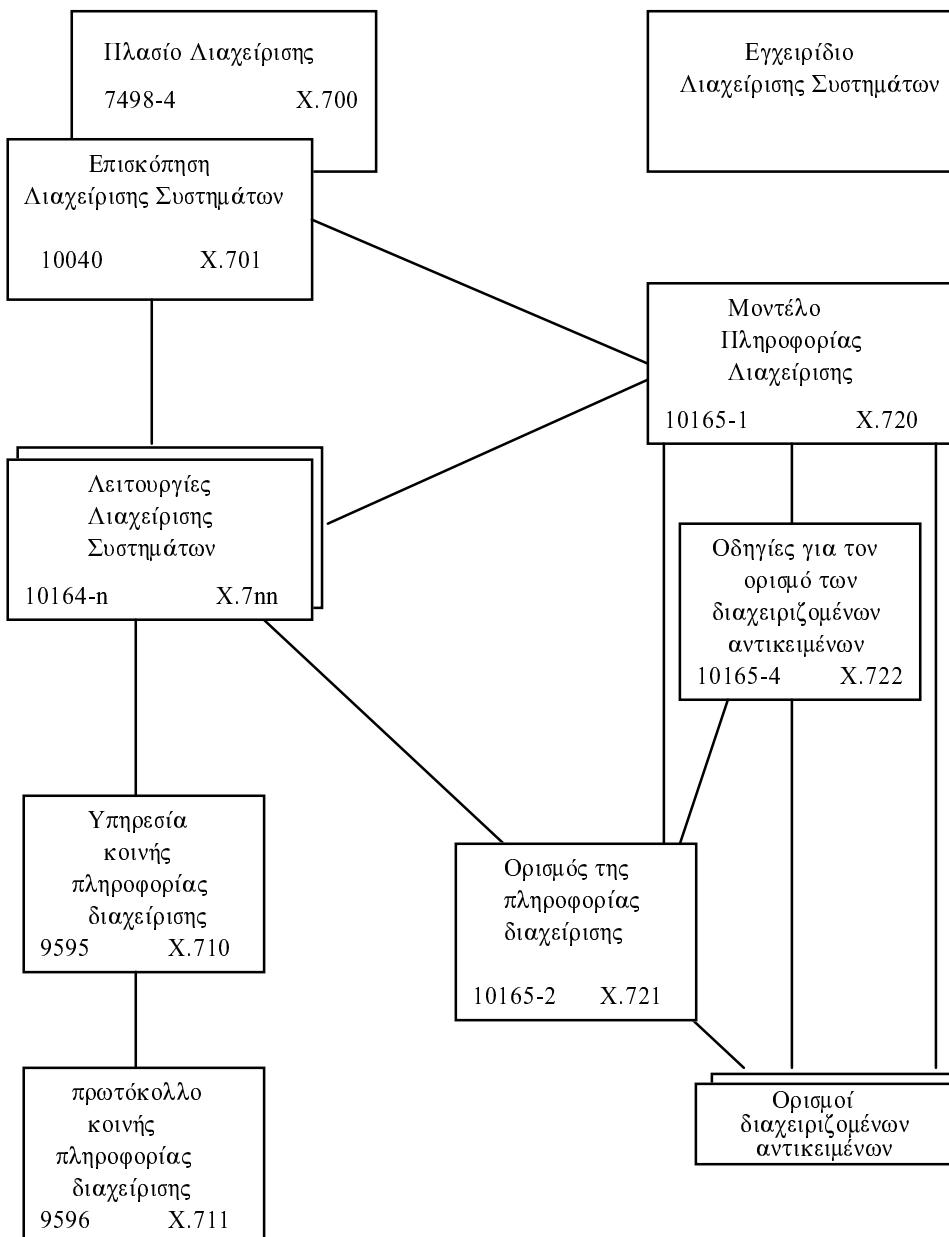
Τα πρότυπα που είναι σχετικά με τη διαχείριση συστημάτων μπορούν να ταξινομηθούν στις παρακάτω κατηγορίες:

- πρότυπα που καθορίζουν τη δομή της πληροφορίας διαχείρισης,
- πρότυπα για τη μεταφορά της πληροφορίας διαχείρισης,
- πρότυπα που είναι σχετικά με τον ορισμό της πληροφορίας διαχείρισης, και
- πρότυπα για λειτουργίες διαχείρισης συστημάτων.

#### 5.1.1.1. Αρχιτεκτονική και Λομή

Στην πρώτη κατηγορία προτύπων έχουμε πρότυπα που καθορίζουν τη δομή της πληροφορίας διαχείρισης. Πιο συγκεκριμένα στην Σύσταση X.700 | ISO 7498-4, η CCITT παρέχει ένα πλαίσιο για το συντονισμό της ανάπτυξης προτύπων για την

διαχείριση OSI, ορίζοντας ορολογία, προδιαγράφοντας κάποια δομή και περιγράφοντας δραστηριότητες της διαχείρισης OSI.



**Σχήμα 5.2 - Σχέσεις μεταξύ προτύπων**

Η Σύσταση | Διεθνές Πρότυπο X.701 | ISO 10040 παρέχει μια επισκόπηση της Διαχείρισης Συστημάτων OSI. Αυτή η προπαρασκευή στη διαχείριση συστημάτων παρέχει μια πληροφοριακή εισαγωγή και περιλαμβάνει όλη τη λογική και τις απαίτησεις που υπεισέρχονται στην ανάπτυξη ενός συνόλου από πρότυπα διαχείρισης συστημάτων.

### 5.1.1.2. Μεταφορά της Πληροφορίας Διαχείρισης

Στην δεύτερη κατηγορία προτύπων έχουμε πρότυπα για τη μεταφορά της πληροφορίας διαχείρισης. Πιο συγκεκριμένα, η CCITT στην Σύσταση X.710 | ISO/IEC 9595 ορίζει

ένα στοιχείο υπηρεσίας εφαρμογής (το Στοιχείο Υπηρεσίας Κοινής Πληροφορίας Διαχείρισης, **Common Management Information Service - CMIS**), το οποίο μπορεί να χρησιμοποιηθεί από μία διαδικασία εφαρμογής (σε ένα κατανευμημένο ή όχι περιβάλλον διαχείρισης) για την ανταλλαγή πληροφορίας με λειτουργίες και μηνύματα διαχείρισης και με σκοπό τη διαχείριση συστημάτων. Επίσης, η CCITT στην Σύσταση X.710 | ISO/IEC 9595 ορίζει ένα σύνολο από στοιχειώδεις υπηρεσίες (που όλες μαζί αποτελούν το στοιχείο υπηρεσίας εφαρμογής) τις σχετικές με αυτές παραμέτρους και κάθε άλλη απαραίτητη πληροφορία για τη περιγραφή της σημασίας κάθε στοιχειώδους υπηρεσίας. Οι στοιχειώδεις υπηρεσίες CMIS μεταφέρουν αιτήσεις για λειτουργίες διαχείρισης, αποτελέσματα λειτουργιών διαχείρισης και αναφορές γεγονότων, που αντιστοιχούν στις λειτουργίες και στα μηνύματα που ορίστηκαν στο Μοντέλο Πληροφορίας Διαχείρισης μεταξύ ανοικτών συστημάτων.

Η CCITT στην Σύσταση X.711 | ISO/IEC 9596-1 καθορίζει το πρωτόκολλο που παρέχει Υπηρεσίες Κοινής Πληροφορίας Διαχείρισης (**Common Management Information Protocol - CMIP**). Αυτό χρησιμοποιείται από οντότητες επιπέδου εφαρμογής για την ανταλλαγή πληροφορίας διαχείρισης. Επίσης, στην Σύσταση X.711 | ISO/IEC 9596-1 καθορίζονται διαδικασίες για τη μετάδοση πληροφορίας διαχείρισης μεταξύ οντοτήτων εφαρμογής, το αφηρημένο συντακτικό του CMIP, διαδικασίες για τη σωστή ερμηνεία της πληροφορίας ελέγχου του πρωτοκόλλου, και τις απαιτήσεις συμμόρφωσης για τις υλοποιήσεις.

Στην περίπτωση ειδικών αναγκών, άλλα Στοιχεία Υπηρεσίας Εφαρμογής (Application Service Elements - ASEs), τέτοια όπως τα TP και FTAM, μπορούν να χρησιμοποιηθούν για τη μεταφορά πληροφορίας διαχείρισης.

### 5.1.1.3. Δομή της Πληροφορίας Διαχείρισης

Τα πρότυπα που είναι σχετικά με τη πληροφορία διαχείρισης μπορούν να χωριστούν σε δύο κατηγορίες: πρότυπα ορισμού των κλάσεων των διαχειριζόμενων αντικειμένων και πρότυπα υποστήριξης του ορισμού των κλάσεων των διαχειριζόμενων αντικειμένων. Οι περισσότεροι ορισμοί των κλάσεων των διαχειριζόμενων αντικειμένων θα δοθούν από ομάδες εργασίας για τα επίπεδα OSI και συνεργαζόμενους οργανισμούς. Από την άλλη πλευρά υπάρχει ανάγκη για κάποια διαχειριζόμενα αντικείμενα προκειμένου αυτά να υποστηρίζουν λειτουργίες της διαχείρισης OSI, όπως π.χ. τα διαχειριζόμενα αντικείμενα που παριστάνουν φίλτρα προώθησης γεγονότων και ημερολόγια διαχείρισης. Πρότυπα για αυτά τα διαχειριζόμενα αντικείμενα αποτελούν μέρος του συνόλου των προτύπων διαχείρισης συστημάτων.

Τα πρότυπα που παρέχουν οδηγίες για τον ορισμό των κλάσεων των διαχειριζόμενων αντικειμένων περιλαμβάνουν τα εξής:

- CCITT Rec. X.720 | ISO/IEC 10165-1, το οποίο ορίζει το μοντέλο για τα διαχειριζόμενα αντικείμενα, καλύπτοντας τα κατηγορήματά τους, τις λειτουργίες διαχείρισης εκείνες που μπορούν να εκτελεστούν σε αυτά, τα μηνύματα που μπορούν αυτά να εκδώσουν και κατάλληλα σχήματα ονομασίας έτσι ώστε τα διαχειριζόμενα αντικείμενα και τα κατηγορήματά τους να μπορούν να αναγνωριστούν με μοναδικό τρόπο από πρωτόκολλα διαχείρισης,
- CCITT Rec. X.721 | ISO/IEC 10165-2, το οποίο ορίζει αντικείμενα διαχείρισης συστημάτων, και περιγράμματα που μπορούν να εισαχθούν στον ορισμό κλάσεων διαχειριζόμενων αντικειμένων, για να υποστηρίζουν το συνεπή ορισμό των κατηγορημάτων τους, των μηνυμάτων, και των διαχειριστικών λειτουργιών συμπεριλαμβανόμενου και των παραμέτρων τους,

- CCITT Rec. X.722 | ISO/IEC 10165-4, η οποία παρέχει καθοδήγηση, μεθόδους και τεχνικές σημειογραφίας για τις κλάσεις των διαχειριζόμενων αντικειμένων και άλλες πληροφορίες διαχείρισης.

#### 5.1.1.4. Λειτουργίες Διαχείρισης Συστημάτων

Τα πρότυπα που αναφέρονται σε λειτουργίες διαχείρισης συστημάτων μπορεί να περιέχουν ένα ή περισσότερα από τα παρακάτω στοιχεία:

- α) Ορισμό ενός συνόλου από υπηρεσίες διαχείρισης συστημάτων που ικανοποιούν συγκεκριμένες απαιτήσεις. Σε πρότυπα που περιλαμβάνουν έναν τέτοιο ορισμό, λειτουργικότητα που παριστάνει προστιθέμενη αξία πέρα από αυτή που είναι διαθέσιμη από το CMISE (ή από άλλα ASEs που χρησιμοποιούνται για να υποστηρίζουν δραστηριότητες διαχείρισης) τεκμηριώνεται με την μορφή υπηρεσιών. Υπηρεσίες προστιθέμενης αξίας ορίζονται σε περιπτώσεις που υπάρχουν περιορισμοί για το περιεχόμενο της πληροφορίας ενός υποστηριζόμενου στοιχείου υπηρεσίας κάποιου ASE (π.χ. περιορίζοντας τους τύπους παραμέτρων που μπορούν να υπάρξουν στο στοιχείο υπηρεσίας, ή περιορίζοντας το στοιχείο υπηρεσίας ώστε να μπορεί να λειτουργήσει σε μια συγκεκριμένη υποστηριζόμενη κλάση αντικειμένων). Υπηρεσίες προστιθέμενης αξίας επίσης ορίζονται σε περιπτώσεις όπου απαιτείται μια συγκεκριμένη χρήση υποστηριζόμενων υπηρεσιών.

Το στοιχείο αυτό αποτελείται από:

- 1) απαιτήσεις του χρήστη,
- 2) ένα μοντέλο που συσχετίζει υπηρεσίες διαχείρισης συστημάτων με απαιτήσεις του χρήστη,
- 3) έναν ορισμό υπηρεσίας που καταγράφει υπηρεσίες διαχείρισης συστημάτων που απαιτούνται και τη σημασία αυτών,
- 4) μια προδιαγραφή πρωτοκόλλου που καθορίζει την απεικόνιση υπηρεσιών διαχείρισης συστημάτων και των παραμέτρων τους σε υποκείμενες υπηρεσίες,
- 5) ορισμούς των σχέσεων μεταξύ υπηρεσιών διαχείρισης συστημάτων και λειτουργιών και μηνυμάτων διαχείρισης SMI,
- 6) σχέσεις με άλλες λειτουργίες διαχείρισης συστημάτων,
- 7) απαιτήσεις συμμόρφωσης.

Πρότυπα τα οποία περιλαμβάνουν αυτό το στοιχείο μπορεί να περιέχουν ή να απαιτούν την χρήση συγκεκριμένων γενικών ορισμών, και μπορεί ακόμα να ορίζουν ομάδες λειτουργιών διαχείρισης συστημάτων.

- β) Απαιτήσεις και μοντέλα για γενικούς ορισμούς. Τέτοια στοιχεία προτύπων λειτουργιών διαχείρισης συστημάτων ασχολούνται μονάχα με την παροχή γενικών ορισμών διαχειριζόμενων αντικειμένων, κατηγορημάτων, λειτουργιών διαχείρισης και μηνυμάτων που ικανοποιούν συγκεκριμένες λειτουργικές ανάγκες.

Τα διαχειριζόμενα αντικείμενα, κατηγορήματα, διαχειριστικές λειτουργίες και μηνύματα που απαιτούνται από πρότυπα που περιλαμβάνουν το στοιχείο αυτό είναι διαθέσιμα για χρησιμοποίηση στην υπηρεσία Pass-through που ορίζεται στις CCITT Rec. X.730 | ISO/IEC 10164-1.

Το στοιχείο αυτό αποτελείται από:

- 1) απαιτήσεις του χρήστη,
- 2) μοντέλα που συσχετίζουν γενικούς ορισμούς με απαιτήσεις του χρήστη,
- 3) δηλώσεις των απαιτήσεων συμμόρφωσης που τοποθετούνται σε άλλα πρότυπα που χρησιμοποιούν τους γενικούς ορισμούς.

γ) Ορισμό των δυνατών ομάδων από λειτουργιές διαχείρισης συστημάτων. Πρότυπα που περιλαμβάνουν το στοιχείο αυτό αναγνωρίζουν συγκεκριμένα σύνολα από υπηρεσίες διαχείρισης συστημάτων όπου υπάρχει κάποια απαίτηση για αποδοχή γνώσης της χρήσης τέτοιας λειτουργικότητας σε μια συσχέτιση σαν μέρος της αποδοχής της γνώσης για τη διαχείριση. Μια ομάδα από λειτουργίες μπορεί να περιλαμβάνει υπηρεσίες ορισμένες σε περισσότερα από ένα πρότυπα, και μπορεί να ορίζει τη χρήση των υπηρεσιών σε συνάρτηση με κλάσεις διαχειριζόμενων αντικειμένων.

Το στοιχείο αυτό αποτελείται από:

- 1) απαιτήσεις του χρήστη,
- 2) μοντέλα που συσχετίζουν ομάδες λειτουργιών διαχείρισης συστημάτων με απαιτήσεις του χρήστη,
- 3) λίστες υπηρεσιών διαχείρισης συστημάτων που απαιτούνται από μία ομάδα λειτουργιών, μαζί με οποιουσδήποτε περιορισμούς μιας κλάσης διαχειριζόμενων αντικειμένων που σχετίζονται με τις υπηρεσίες αυτές, όπως και αυτές σχετίζονται με τις ομάδες λειτουργιών,
- 4) ορισμούς ομάδων λειτουργιών,
- 5) το απαραίτητο αφηρημένο συντακτικό για την αναγνώριση κάποιας ομάδας από λειτουργίες από το πρωτόκολλο,
- 6) περιγραφές οποιονδήποτε σχέσεων μεταξύ ομάδων λειτουργιών,
- 7) περιγραφές οποιονδήποτε σχέσεων μεταξύ ομάδων λειτουργιών και λειτουργιών διαχείρισης συστημάτων,
- 8) απαιτήσεις συμμόρφωσης.

Κάθε ένα από τα στοιχεία αυτά μπορεί να υπάρξει αυτόνομα σε ένα πρότυπο περιγραφής λειτουργιών διαχείρισης συστημάτων. Μπορεί ακόμα να συνδυαστούν με οποιονδήποτε τρόπο εκτός της περίπτωσης που ένα στοιχείο με γενικούς ορισμούς συνδυάζεται με ένα στοιχείο ομάδας λειτουργιών, οπότε απαιτείται είτε αναφορά, είτε περίληψη ενός στοιχείου ορισμού υπηρεσιών.

### 5.1.2. Περιοχές Λειτουργιών Διαχείρισης OSI

Η διαχείριση OSI είναι απαραίτητη προκειμένου να ικανοποιήσει έναν αριθμό από απαιτήσεις. Οι απαιτήσεις αυτές τοποθετούνται σε πέντε περιοχές λειτουργιών:

- α) Διαχείριση Διάρθρωσης (*Configuration Management*),
- β) Διαχείριση Σφαλμάτων (*Fault Management*),
- γ) Λογιστική Διαχείριση (*Accounting Management*),
- δ) Διαχείριση Επιδόσεων (*Performance Management*), και
- ε) Διαχείριση Ασφάλειας (*Security Management*).

Τα αρχικά των ανωτέρω πέντε περιοχών **CFAPS** αποτελούν μνημονικό κανόνα. Ιδιαίτερες διαχειριστικές λειτουργίες μέσα σ' αυτές τις περιοχές λειτουργιών παρέχονται από μηχανισμούς της διαχείρισης OSI. Πολλοί από τους μηχανισμούς αυτούς είναι γενικοί με την έννοια ότι μπορούν να χρησιμοποιηθούν για την ικανοποίηση απαιτήσεων σε περισσότερες από μία περιοχές λειτουργιών. Παρόμοια, τα διαχειριζόμενα αντικείμενα είναι γενικά με την έννοια ότι μπορεί να είναι κοινά για περισσότερες από μία περιοχές λειτουργιών.

Κάθε μία από αυτές τις περιοχές λειτουργιών περιγράφονται σύντομα παρακάτω. Οι λίστες των δυνατών λειτουργιών που αναφέρονται δεν είναι κατ' ανάγκη πλήρεις.

#### 5.1.2.1. Διαχείριση Σφαλμάτων (*Fault Management*)

Η διαχείριση σφαλμάτων επιτρέπει την ανακάλυψη σφαλμάτων, την απομόνωση και τη διόρθωση των αντικανονικών λειτουργιών του περιβάλλοντος OSI. Τα σφάλματα είναι αιτίες εξαιτίας των οποίων τα ανοικτά συστήματα δεν επιτυγχάνουν τους λειτουργικούς στόχους τους και μπορεί να είναι συνεχιζόμενα ή μεταβατικά. Τα σφάλματα εμφανίζονται σαν ιδιαίτερα γεγονότα (π.χ. λάθη) κατά τη λειτουργία του ανοικτού συστήματος. Η ανακάλυψη λαθών παρέχει τη δυνατότητα αναγνώρισης σφαλμάτων. Η διαχείριση σφαλμάτων περιλαμβάνει λειτουργίες προκειμένου:

- α) να διατηρεί και να εξετάζει ημερολόγια σφαλμάτων,
- β) να αποδέχεται μηνύματα εύρεσης σφαλμάτων και να δρα ανάλογα,
- γ) να βρίσκει και να αναγνωρίζει σφάλματα,
- δ) να διεξάγει σειρές από διαγνωστικές δοκιμές, και
- ε) να διορθώνει σφάλματα.

#### 5.1.2.2. Λογιστική διαχείριση (*Accounting Management*)

Η λογιστική διαχείριση επιτρέπει την επιβολή επιβάρυνσης για τη χρησιμοποίηση των πόρων στο OSIE και την απόδειξη των χρεώσεων για τη χρησιμοποίηση των πόρων αυτών. Η λογιστική διαχείριση περιλαμβάνει λειτουργίες προκειμένου:

- α) να πληροφορεί τους χρήστες για τις χρεώσεις που εκδόθηκαν ή για τους πόρους που χρησιμοποιήθηκαν,
- β) να επιτρέπει να τεθούν διάφορα λογιστικά όρια, και να συσχετιστούν σχέδια τιμολόγησης με τη χρήση πόρων, και
- γ) να επιτρέπει το συνδυασμό διαφόρων τιμολογίων, σε περιπτώσεις που πολλά στοιχεία χρησιμοποιούνται για την επίτευξη στόχων μεταφοράς της πληροφορίας.

### **5.1.2.3. Διαχείριση Διάρθρωσης (Configuration Management)**

Η διαχείριση διάρθρωσης αναγνωρίζει, εξασκεί έλεγχο, συλλέγει δεδομένα και παρέχει δεδομένα στα ανοικτά συστήματα για τους σκοπούς της αρχικοποίησης, εκκίνησης, συντήρησης της συνεχής λειτουργίας και τερματισμού υπηρεσιών διασύνδεσης. Η διαχείριση διάρθρωσης περιλαμβάνει λειτουργίες προκειμένου:

- α) να θέτει τις παραμέτρους που ελέγχουν την κανονική λειτουργία του ανοικτού συστήματος,
- β) να συσχετίζει ονόματα με διαχειριζόμενα αντικείμενα και σύνολα διαχειριζόμενων αντικειμένων,
- γ) να αρχικοποιεί και να "κατεβάζει" διαχειριζόμενα αντικείμενα,
- δ) να συλλέγει πληροφορία για την παρούσα κατάσταση του ανοικτού συστήματος, την στιγμή που αυτή είναι απαραίτητη,
- ε) να δέχεται μηνύματα σημαντικών αλλαγών στην κατάσταση των ανοικτών συστημάτων, και
- στ) να αλλάζει τη διάρθρωση του ανοικτού συστήματος.

### **5.1.2.4. Διαχείριση Επιδόσεων (Performance Management)**

Η διαχείριση επιδόσεων επιτρέπει την αξιολόγηση της συμπεριφοράς των πόρων μέσα στο OSIE και της αποτελεσματικότητας των δραστηριοτήτων μεταφοράς της πληροφορίας. Η διαχείριση επιδόσεων περιλαμβάνει λειτουργίες προκειμένου:

- α) να συλλέγει στατιστικά στοιχεία,
- β) να διατηρεί και να εξετάζει ημερολόγια της κατάστασης του συστήματος στο παρελθόν,
- γ) να καθορίζει την απόδοση του συστήματος κάτω από κανονικές, είτε τεχνητές συνθήκες, και
- δ) να αλλάζει τους τρόπους λειτουργίας του συστήματος με σκοπό τη διεξαγωγή διαχειριστικών δραστηριοτήτων.

### **5.1.2.5. Διαχείριση Ασφάλειας (Security Management)**

Ο σκοπός της διαχείρισης ασφάλειας είναι η υποστήριξη εφαρμογών ασφάλειας μέσω λειτουργιών οι οποίες περιλαμβάνουν:

- α) τη δημιουργία, τη διαγραφή και τον έλεγχο υπηρεσιών και μηχανισμών ασφάλειας,
- β) τη διανομή πληροφοριών σχετιζόμενων με την ασφάλεια, και
- γ) την αναφορά γεγονότων σχετικών με την ασφάλεια.

## 5.2. Διαχείριση Συστημάτων

Η διαχείριση συστημάτων προσφέρει μηχανισμούς για τη παρακολούθηση, τον έλεγχο και τον συντονισμό των στοιχείων ενός περιβάλλοντος OSI, καθώς και πρότυπα OSI περιγραφής πρωτοκόλλων για τη μεταφορά της πληροφορίας που είναι σχετική με τα στοιχεία αυτά. Προκειμένου να περιγραφούν οι λειτουργίες που είναι δυνατόν να εφαρμοστούν στα στοιχεία του περιβάλλοντος OSI, τα στοιχεία αυτά αντιμετωπίζονται σαν διαχειριζόμενα αντικείμενα με ορισμένες ιδιότητες. Η πληροφορία που απαιτείται για τους σκοπούς της διαχείρισης συστημάτων σε ένα ανοικτό σύστημα μπορεί να προκύψει είτε μέσω κάποιας τοπικής εισόδου, είτε μέσω μεταφοράς από άλλο ανοικτό σύστημα με επικοινωνία διαχείρισης συστημάτων (στο επίπεδο εφαρμογής), ή τέλος μπορεί να προκύψει μετά από ανταλλαγές με τη βοήθεια κάποιου πρωτοκόλλου σε χαμηλότερο επίπεδο.

Η διαχείριση συστημάτων είναι εφαρμόσιμη σε μια ευρεία περιοχή από περιβάλλοντα κατανεμημένης επεξεργασίας και επικοινωνίας. Τα περιβάλλοντα αυτά περιλαμβάνουν από τοπικά δίκτυα που διασυνδέουν μικρά συστήματα, μέχρι διασυνδεδεμένα δίκτυα εταιριών και δημόσια δίκτυα σε παγκόσμια κλίμακα. Περιβάλλοντα μικρής κλίμακας μπορούν να ικανοποιήσουν τις διαχειριστικές τους ανάγκες με μικρής κλίμακας διαχειριστικά συστήματα, αποτελούμενα από ένα διαχειριστή ικανό να ελέγχει και να συντονίζει το ανοικτό επικοινωνιακό περιβάλλον μέσω ενός αριθμού αντιπροσώπων. Τα πρότυπα και οι ιδέες μπορούν επίσης να εφαρμοστούν σε μεγάλης κλίμακας περιβάλλοντα που υποστηρίζουν πολλούς διαχειριστές.

Μπορούμε να ξεχωρίσουμε τρεις κύριες ομάδες στο σύνολο των προτύπων διαχείρισης συστημάτων. Αυτές είναι:

- α) ένα σύνολο από πρότυπα που καθορίζουν λειτουργίες διαχείρισης συστημάτων,
- β) ένα σύνολο από πρότυπα σχετικά με το καθορισμό των διαχειριζόμενων αντικειμένων,
- γ) ένα σύνολο από πρότυπα υπηρεσιών και πρωτοκόλλων επιπέδου εφαρμογής για την μεταφορά της πληροφορίας που είναι σχετική με τις λειτουργίες διαχείρισης.

Οι απαιτήσεις που πρέπει να ικανοποιηθούν από τις δραστηριότητες διαχείρισης συστημάτων μπορούν να χωριστούν στις πέντε περιοχές που αναφέραμε. Κάθε μία από αυτές δίνει ένα ή περισσότερα πρότυπα, τα οποία καλύπτουν μία ή περισσότερες λειτουργίες.

Εντούτοις, πολλά κομμάτια της πληροφορίας, οι σχετικές με αυτή την πληροφορία λειτουργίες διαχείρισης και τα πρωτόκολλα μεταφοράς της πληροφορίας μπορεί να είναι κοινά για περισσότερες από μία περιοχές. Επίσης, κατά την εκτέλεση

δραστηριοτήτων διαχείρισης, είναι δυνατόν να συνδυαστούν σύνολα από λειτουργίες διαχείρισης για την επίτευξη μιας συγκεκριμένης πολιτικής διαχείρισης.

Για τους λόγους αυτούς, τα πρότυπα διαχείρισης συστημάτων σχηματίζουν ένα σύνολο στενά αλληλοεξαρτημένων προτύπων.

### 5.2.1. Μοντέλο Διαχείρισης Συστημάτων

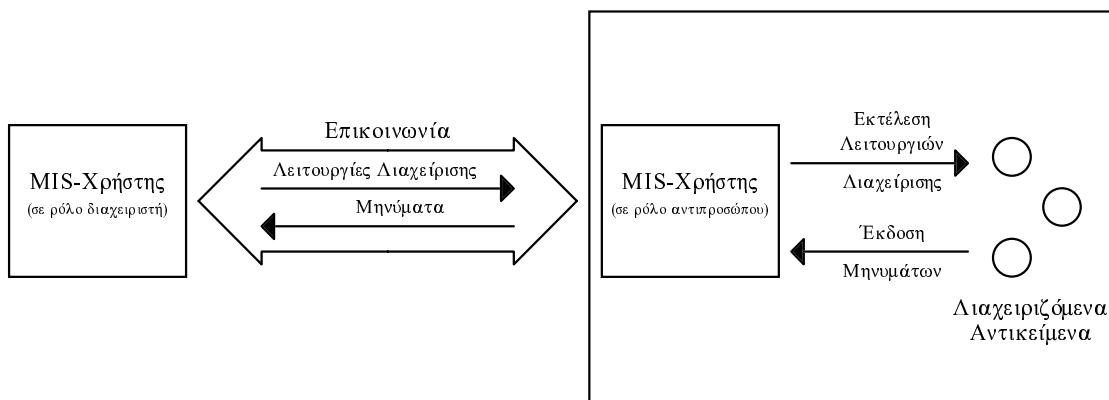
Η παράγραφος αυτή αναγνωρίζει ένα αριθμό από ιδέες σχετικές με τη διαχείριση συστημάτων και παρέχει ένα μοντέλο για τη διασαφήνιση των ιδεών αυτών και των σχέσεων μεταξύ τους.

Οι παράγραφοι που ακολουθούν περιγράφουν διάφορες πλευρές του μοντέλου διαχείρισης συστημάτων, όπως για παράδειγμα:

- πλευρές που αφορούν την πληροφορία,
- πλευρές που αφορούν τις λειτουργίες,
- πλευρές επικοινωνιών OSI,
- πλευρές οργάνωσης.

Η διαχείριση ενός περιβάλλοντος επικοινωνιών είναι μία εφαρμογή επεξεργασίας πληροφορίας. Επειδή το διαχειριζόμενο περιβάλλον είναι κατανευμένο, τα ανεξάρτητα μέλη των διαχειριστικών δραστηριοτήτων είναι τα ίδια κατανευμένα. Οι διαχειριστικές εφαρμογές εκτελούν τις διαχειριστικές δραστηριότητες με ένα κατανευμένο τρόπο, μετά από εγκατάσταση συσχετίσεων μεταξύ οντοτήτων εφαρμογής διαχείρισης συστημάτων.

Όπως φαίνεται στο παρακάτω σχήμα οι αλληλεπιδράσεις που λαμβάνουν χώρα μεταξύ οντοτήτων εφαρμογής διαχείρισης συστημάτων μπορούν να εκφραστούν αφηρημένα μέσα από λειτουργίες και μηνύματα διαχείρισης, τα οποία μεταδίδονται από τη μία οντότητα στην άλλη. Αυτά μεταφέρονται από υπηρεσίες και πρωτόκολλα διαχείρισης συστημάτων.



**Σχήμα 5.3 - Αλληλεπιδράσεις Διαχείρισης Συστημάτων**

Οι δραστηριότητες διαχείρισης επιτυγχάνονται μέσω χειρισμού των διαχειριζόμενων αντικειμένων. Για τους σκοπούς της διαχείρισης συστημάτων, οι εφαρμογές διαχείρισης χωρίζονται σε κατηγορίες σαν MIS-Xρήστες (Management Information Service -

Χρήστες). Κάθε αλληλεπίδραση λαμβάνει χώρα μεταξύ δύο MIS-Χρηστών, ενός που έχει το ρόλο του διαχειριστή και ενός που έχει το ρόλο του αντιπροσώπου.

Ένας MIS-Χρήστης που παίρνει το ρόλο του αντιπροσώπου αποτελεί εκείνο το μέρος μιας κατανεμημένης εφαρμογής που διαχειρίζεται τα διαχειριζόμενα αντικείμενα στο τοπικό περιβάλλον του συστήματος. Ένας αντιπρόσωπος εκτελεί λειτουργίες διαχείρισης στα διαχειριζόμενα αντικείμενα σαν μια συνέπεια λειτουργιών διαχείρισης που ζητήθηκαν από ένα διαχειριστή. Ένας αντιπρόσωπος μπορεί ακόμα να προωθήσει μηνύματα που εκδόθηκαν από διαχειριζόμενα αντικείμενα προς ένα διαχειριστή.

Ένας MIS-Χρήστης που παίρνει το ρόλο του διαχειριστή είναι εκείνο το μέρος μιας κατανεμημένης εφαρμογής το οποίο είναι υπεύθυνο για μία ή περισσότερες δραστηριότητες διαχείρισης, εκδίδοντας λειτουργίες διαχείρισης και λαμβάνοντας μηνύματα.

Η έννοια του διαχειριστή δεν περιορίζεται σε εφαρμογές που λαμβάνουν μέρος μονάχα στη διαχείριση συστημάτων. Και άλλες εφαρμογές που απαιτούν πρόσβαση σε πληροφορία διαχείρισης μπορούν να χρησιμοποιήσουν σχετικές υπηρεσίες.

Οι ρόλοι δε δίνονται μόνιμα στους MIS-Χρήστες. Μερικοί MIS-Χρήστες μπορεί να περιορίζονται στο ρόλο του αντιπροσώπου, άλλοι στο ρόλο του διαχειριστή, ενώ άλλοι μπορεί να είναι δυνατόν να λάβουν το ρόλο του αντιπροσώπου για κάποια αλληλεπίδραση και το ρόλο του διαχειριστή για κάποια ξεχωριστή αλληλεπίδραση.

Είναι σημαντικό να αναγνωρίσουμε ότι εδώ καθορίζουμε ένα νοητό μοντέλο το οποίο περιγράφει τη δομή και το περιεχόμενο της πληροφορίας που στη πράξη μεταφέρεται με τη βοήθεια υπηρεσιών πληροφορίας διαχείρισης, που καθορίζονται από πρότυπα. Οποτεδήποτε γίνεται μεταφορά πληροφορίας διαχείρισης, η μεταφορά ακολουθεί το παραπάνω μοντέλο.

Το εάν, πού και πώς τα συστήματα παριστάνουν και αποθηκεύουν τα πραγματικά δεδομένα από τα οποία προκύπτει η πληροφορία διαχείρισης είναι ένα τοπικό ζήτημα, το οποίο δεν υπόκειται σε προτυποποίηση.

### 5.2.1.1. Θέματα πληροφορίας

Η παράγραφος αυτή εισάγει τα θέματα πληροφορίας του μοντέλου διαχείρισης συστημάτων. Η οριστική περιγραφή του μοντέλου πληροφορίας δίνεται στα CCITT Rec. X.720 | ISO/IEC 10165-1. Εκεί, βελτιώνεται η έννοια των Διαχειριζόμενων Αντικειμένων, όπως αυτή ορίστηκε στα CCITT Rec. X.700 | ISO/IEC 7498-4. Η παράγραφος αυτή ασχολείται με τα κατηγορήματά τους, τις διαχειριστικές λειτουργίες που είναι δυνατό να εκτελεστούν σ' αυτά, και τα μηνύματα που μπορούν αυτά να εκδώσουν. Το σύνολο των διαχειριζόμενων αντικειμένων σε ένα σύστημα, μαζί με τα κατηγορήματά τους αποτελούν τη **Βάση Πληροφορίας Διαχείρισης (Management Information Base - MIB)** του συστήματος αυτού.

Διαχειριζόμενα αντικείμενα που θα καθορίζονται από πρότυπα αναμένεται να προδιαγραφούν από τους οργανισμούς που είναι υπεύθυνοι για την προτυποποίηση των στοιχείων, τα οποία τα διαχειριζόμενα αντικείμενα αυτά αναπαριστάνουν (π.χ. η ομάδα που είναι υπεύθυνη για την προτυποποίηση μιας οντότητας πρωτοκόλλου στο (N)-επίπεδο, είναι υπεύθυνη και για την προτυποποίηση του διαχειριζόμενου αντικειμένου που αποτελεί τη διαχειριστική πλευρά της οντότητας αυτής). Οδηγίες και εργαλεία για την υποστήριξη του ορισμού των διαχειριζόμενων αντικειμένων

παρέχονται, όπως για παράδειγμα μία συλλογή από ορισμούς πληροφορίας διαχείρισης που υποστηρίζει τους ορισμούς των διαχειριζόμενων αντικειμένων και τον ορισμό των λειτουργιών διαχείρισης συστημάτων.

Ένα **διαχειριζόμενο αντικείμενο (Managed Object)** είναι η διαχειριστική πλευρά ενός στοιχείου αντικείμενο διαχείρισης, τέτοιο όπως μια οντότητα σε κάποιο επίπεδο, μια σύνδεση ή ένα μέρος μιας συσκευής επικοινωνίας. Δηλ., ένα διαχειριζόμενο αντικείμενο αποτελεί μια αφαίρεση ενός πραγματικού στοιχείου και παριστάνει τις ιδιότητές του όπως αντιμετωπίζονται από την πλευρά (και για τους σκοπούς της) διαχείρισης. Ένα ουσιαστικό μέρος του ορισμού ενός διαχειριζόμενου αντικειμένου είναι η σχέση μεταξύ των ιδιοτήτων αυτών και της λειτουργικής συμπεριφοράς του στοιχείου. Η σχέση αυτή δεν ακολουθεί κάποιο γενικό μοντέλο.

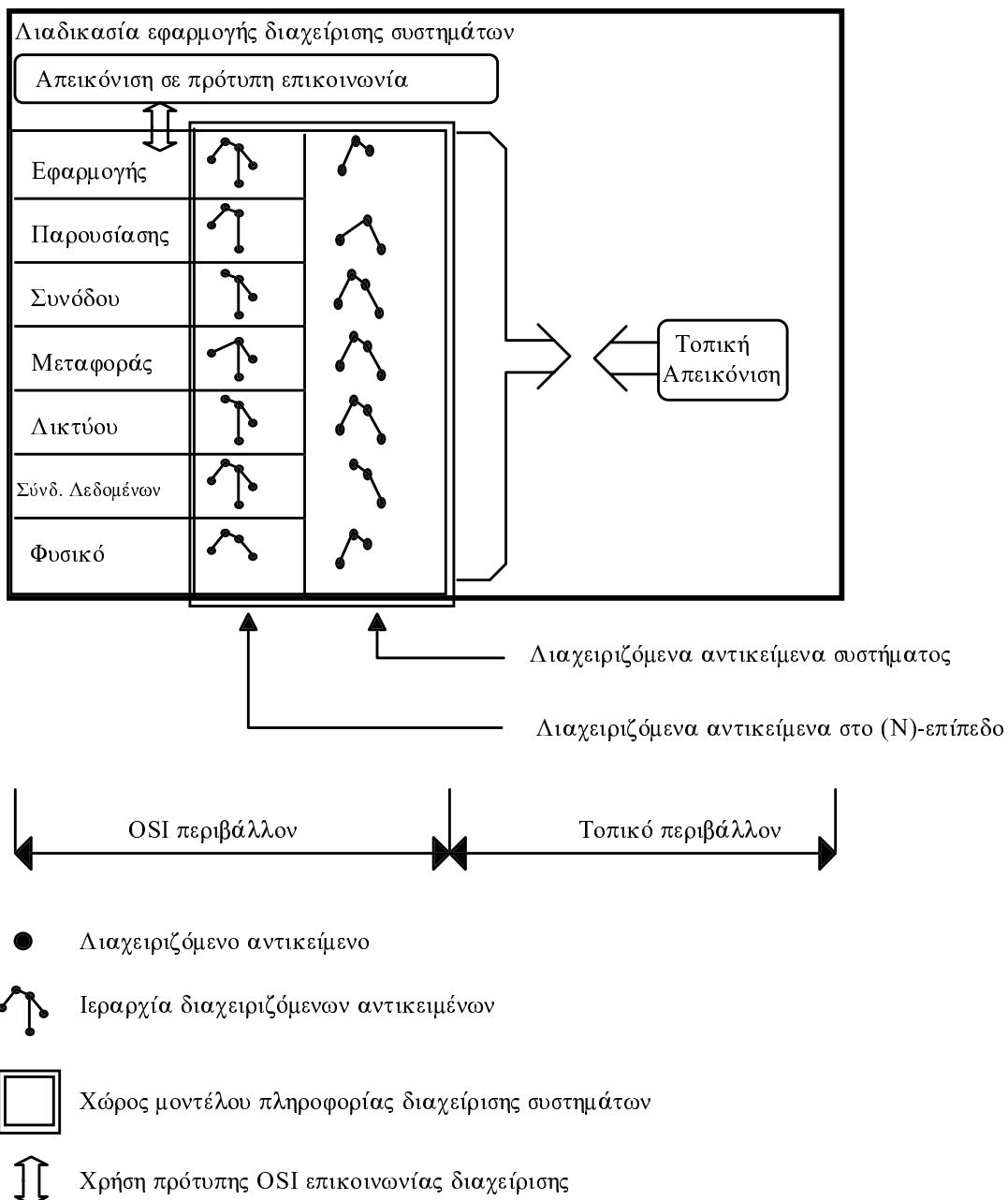
Τα διαχειριζόμενα αντικείμενα μπορεί να είναι σχετικά με ένα συγκεκριμένο επίπεδο, οπότε και είναι γνωστά σαν διαχειριζόμενα αντικείμενα στο (N)-επίπεδο. Εκείνα τα διαχειριζόμενα αντικείμενα τα οποία είναι σχετικά με περισσότερα του ενός επίπεδα, για μια συγκεκριμένη λειτουργία διαχείρισης συστημάτων (αντικείμενο υποστήριξης διαχείρισης) ή για το συνολικό σύστημα είναι γνωστά σαν διαχειριζόμενα αντικείμενα συστήματος.

Τα κατηγορήματα παριστάνουν ιδιότητες των διαχειριζόμενων αντικειμένων. Ένα κατηγόρημα έχει μία σχετική τιμή, η οποία μπορεί να έχει μια απλή ή μία σύνθετη δομή.

Μέρος του ορισμού ενός διαχειριζόμενου αντικειμένου είναι η προδιαγραφή της ομάδας των διαχειριστικών λειτουργιών που μπορούν να εκτελεστούν σ' αυτό και των συνεπειών που αυτές οι διαχειριστικές λειτουργίες θα έχουν στο διαχειριζόμενο αντικείμενο και στα κατηγορήματα αυτού. Ο ορισμός μπορεί ακόμα να καθορίζει τις συνέπειες, αν υπάρχουν, σε σχετικά διαχειριζόμενα αντικείμενα. Η εκτέλεση μιας διαχειριστικής λειτουργίας μπορεί επίσης να πραγματοποιείται υπό όρους και να εξαρτηθεί από την κατάσταση του διαχειριζόμενου αντικειμένου και των κατηγορημάτων αυτού. Ένα σημαντικό μέρος του ορισμού των διαχειριστικών λειτουργιών είναι επίσης το σύνολο των σφαλμάτων που είναι δυνατό να συμβούν.

Τα διαχειριζόμενα αντικείμενα μπορεί ακόμα να εκδίδουν μηνύματα, τα οποία περιέχουν πληροφορία που αφορά την εμφάνιση ενός σχετικού με αυτά γεγονότος.

Οι μηχανισμοί για τη μεταφορά διαχειριστικών λειτουργιών και μηνυμάτων υπόκεινται σε προτυποποίηση σύμφωνα με τη διαχείριση OSI, ενώ αντίθετα οι μηχανισμοί εκτέλεσης των διαχειριστικών λειτουργιών και μηνυμάτων δεν υπόκεινται. Κανένα αντίστοιχο εσωτερικό σύστημα διασύνδεσης δεν είναι αντικείμενο προτυποποίησης. Η σχέση μεταξύ διαχειριστικών λειτουργιών στα σύνορα του διαχειριζόμενου αντικειμένου και του τι θα μεταφέρθει με το πρωτόκολλο μεταξύ των ανοικτών συστημάτων περιγράφεται παρακάτω.



### 5.2.1.2. Λειτουργικές πλευρές

Η παράγραφος αυτή περιγράφει λειτουργικές πλευρές του μοντέλου διαχείρισης συστημάτων.

Μια λειτουργία διαχείρισης συστημάτων μπορεί να ικανοποιεί περισσότερες από μία απαιτήσεις και για να ικανοποιηθούν κάποιες απαιτήσεις μπορεί να χρειαστούν περισσότερες από μία λειτουργίας. Έτσι, μια σχέση πολλές-προς-πολλές ισχύει μεταξύ λειτουργιών και απαιτήσεων.

Η προδιαγραφή μιας λειτουργίας διαχείρισης συστημάτων ορίζει τις διαχειριστικές δραστηριότητες και την πληροφορία που είναι απαραίτητη προκειμένου να ικανοποιηθούν οι απαιτήσεις.

Διαχειριστικές λειτουργίες μπορούν να συνδυαστούν προκειμένου να πραγματοποιηθεί μια συγκεκριμένη διαχειριστική δραστηριότητα.

Από τη στιγμή που δεν είναι όλες οι υπηρεσίες απαραίτητες σε μια δεδομένη συσχέτιση, οι υπηρεσίες των λειτουργιών διαχείρισης συστημάτων μπορούν να χωριστούν σε περισσότερες από μία ομάδες λειτουργιών, οι οποίες θα αποτελέσουν τις βασικές μονάδες διαπραγμάτευσης μεταξύ MIS-Χρηστών. Επιπλέον, ομάδες λειτουργιών οι οποίες επικαλύπτουν υπηρεσίες περισσότερων της μίας λειτουργιών μπορεί να οριστούν.

Ομάδες λειτουργιών οι οποίες περιέχουν υπηρεσίες περισσότερων της μίας λειτουργιών παρέχονται για να υποστηρίζουν το παρακάτω σύνολο δυνατοτήτων:

- α) μόνο αποστολής μηνυμάτων,
- β) μόνο εκτέλεσης λειτουργιών διαχείρισης,
- γ) αποστολή μηνυμάτων και εκτέλεσης λειτουργιών διαχείρισης.

Ο αντιπρόσωπος γενικά δεν μπορεί να καθορίσει το σκοπό των διαχειριστικών λειτουργιών που λαμβάνει ή των μηνυμάτων που εκδίδει. Για παράδειγμα, ένα ανοικτό σύστημα δεν μπορεί γενικά να καθορίσει αν οι αποκρίσεις του σε αιτήσεις ανάγνωσης μετρητών λαθών θα χρησιμοποιηθούν για διαχείριση σφαλμάτων ή για διαχείριση επιδόσεων. Ο αντιπρόσωπος ανταποκρίνεται στις ερωτήσεις του διαχειριστή, χωρίς να χρειάζεται κάποιο ευρύτερο πλαίσιο για να διεκπεραιώσει την απαίτηση αυτή.

### 5.2.1.3. Θέματα επικοινωνιών OSI

Οι αλληλεπιδράσεις μεταξύ MIS-Χρηστών που ενεργούν σαν διαχειριστές ή σαν αντιπρόσωποι αντίστοιχα πραγματοποιούνται μέσω ανταλλαγής πληροφορίας διαχείρισης. Αυτή η επικοινωνία επιτυγχάνεται με τη χρήση πρωτοκόλλων OSI.

Η γενική υπηρεσία επικοινωνίας OSI για τη διαχείριση συστημάτων είναι το CMIS (Common Management Information Service). Παρακάτω περιγράφουμα πως το CMIS μπορεί να χρησιμοποιηθεί για να υποστηρίζει επικοινωνίες που αφορούν λειτουργίες και μηνύματα διαχείρισης που εφαρμόζονται σε διαχειριζόμενα αντικείμενα ενός διαχειριζόμενου συστήματος. Επίσης εξηγούμε πως η υποστήριξη της μεταφοράς πληροφορίας διαχείρισης ενσωματώνεται στη δομή του επιπέδου εφαρμογής.

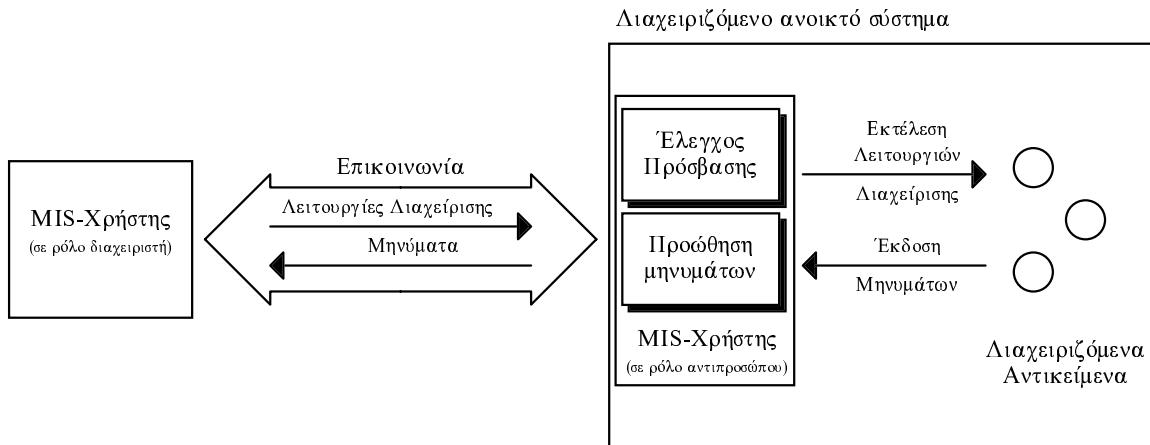
Οι MIS-Χρήστες μπορεί να χρησιμοποιούν και άλλες υπηρεσίες OSI (τέτοιες όπως οι TP και FTAM) και οι οποίες μπορεί να υποστηρίζουν ή να μην υποστηρίζουν τη διάκριση μεταξύ των ρόλων διαχειριστή/αντιπροσώπου. Εντούτοις, οι MIS-Χρήστες πρέπει να υποστηρίζουν τη διάκριση μεταξύ ρόλων αυτών.

Υπάρχουν δύο πλευρές στην επικοινωνιακή υποστήριξη των λειτουργιών και των μηνυμάτων διαχείρισης:

- α) υποστήριξη της μεταφοράς των αιτήσεων για λειτουργίες διαχείρισης και μηνυμάτων μεταξύ MIS-Χρηστών,

- β) υποστήριξη του ελέγχου της πρόσβασης σε διαχειριζόμενα αντικείμενα και την εξωτερική διάδοση της πληροφορίας των μηνυμάτων.

Τα κύρια κομμάτια φαίνονται στο παρακάτω σχήμα.



**Σχήμα 5.5 - Επικοινωνιακή υποστήριξη για μηνύματα και λειτουργίες διαχείρισης**

Οι υπηρεσίες διαχείρισης συστημάτων περιέχουν στοιχεία για τη μεταφορά των αιτήσεων για τις διάφορες λειτουργίες διαχείρισης που ορίζονται στη CCITT Rec. X.720 | ISO/IEC 10165-1, και στοιχεία για τη μεταφορά της πληροφορίας των μηνυμάτων. Κατ' αυτό τον τρόπο, οι υπηρεσίες διαχείρισης συστημάτων αντικατοπτρίζουν τις ανταλλαγές που ορίζονται στο σύνορο των διαχειριζόμενων αντικειμένων. Οι υπηρεσίες διαχείρισης συστημάτων προσφέρουν επιπρόσθετη υποστήριξη για την επιλογή των επιθυμητών διαχειριζόμενων αντικειμένων μέσω **οριοθέτησης (scoping)** και **φίλτρων (filtering)**.

Η CCITT Rec. X.730 | ISO/IEC 10164-1 ορίζει τον τρόπο με τον οποίο οι υπηρεσίες διαχείρισης συστημάτων απεικονίζονται σε CMIS υπηρεσίες. Υπάρχει μια αυστηρή αντιστοιχία μεταξύ των μορφών ανταλλαγής που ορίζονται (στο μοντέλο πληροφορίας) στο σύνορο των διαχειριζόμενων αντικειμένων και της υποστηριζόμενης επικοινωνίας από τις υπηρεσίες διαχείρισης συστημάτων. Εντούτοις, σε ανεξάρτητες ανταλλαγές (ή δυνατές ανταλλαγές) πληροφορίας, οι μηχανισμοί αυτοί μπορεί να μεσολαβούν για να ελέγξουν τη ροή της πληροφορίας.

Μηχανισμοί ελέγχου της πρόσβασης μπορούν να αρνηθούν αιτήσεις για εκτέλεση λειτουργιών διαχείρισης από συγκεκριμένους διαχειριστές σε επιλεγμένα διαχειριζόμενα αντικείμενα.

Για την εξωτερική μεταφορά μηνυμάτων διαχείρισης που εκδίδονται από κάποιο διαχειριζόμενο αντικείμενο, ένας μηχανισμός ορίζεται για την αναγνώριση προορισμών καθώς και κριτηρίων που η πληροφορία των μηνυμάτων πρέπει να ικανοποιεί. Ανεξάρτητα από αυτά, ορίζεται άλλος ένας μηχανισμός, ο οποίος μπορεί να οδηγήσει σε καταχώρηση της πληροφορίας για μετέπειτα ανάκτηση.

Η οντότητα εφαρμογής διαχείρισης συστημάτων (SMAE - Systems Management Application Entity) αποτελείται από το στοιχείο υπηρεσίας εφαρμογής διαχείρισης συστημάτων (SMASE - System Management Application Service Element) και το στοιχείο υπηρεσίας ελέγχου συσχετίσεων (ACSE - Association Control Service Element, CCITT Rec.X.217 | ISO 8649). Άλλα στοιχεία υπηρεσίας εφαρμογής που απαιτούνται από το SMAE περιγράφονται παρακάτω.

Το παρακάτω σχήμα δείχνει πως ακριβώς τα διάφορα στοιχεία διαχείρισης συστημάτων ενσωματώνονται στην δομή του επιπέδου εφαρμογής.

Το SMASE ορίζει τη σημασία και το αφηρημένο συντακτικό της πληροφορίας της σχετικής με τη διαχείριση OSI που μεταφέρεται με Μονάδες Δεδομένων Πρωτοκόλλου Εφαρμογής Διαχείρισης (MAPDUs - Management Application Protocol Data Unit). Το MAPDU είναι η πραγματοποίηση με πρωτόκολλο OSI της αφηρημένης έννοιας των λειτουργιών και των μηνυμάτων διαχείρισης που ανταλλάσσονται μεταξύ οντοτήτων εφαρμογής διαχείρισης συστημάτων (βλ. παραπάνω). Για κάθε ορισμένο MAPDU, καθορίζεται επίσης η απεικόνιση σε υποστηριζόμενες υπηρεσίες.

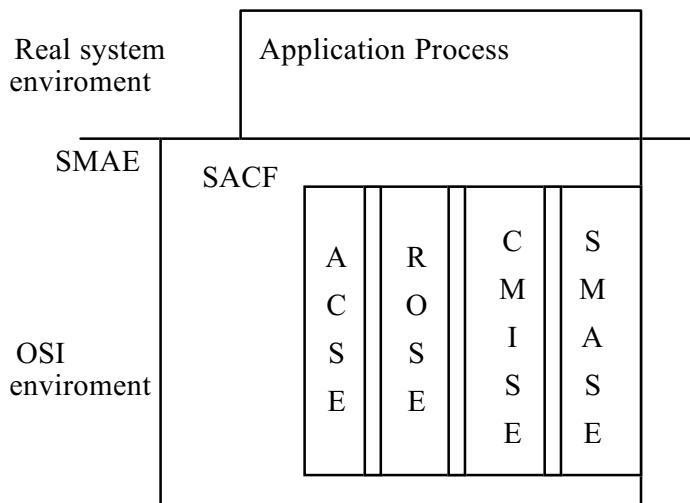
Οι υπηρεσίες που προσφέρονται από το SMASE μπορεί να χωριστούν σε ομάδες για σκοπούς διαπραγμάτευσης. Το SMASE καθορίζει ότι πληροφορία διαχείρισης πρέπει να ανταλλάσσεται μεταξύ οντοτήτων εφαρμογής διαχείρισης συστημάτων. Οι υπηρεσίες επικοινωνίας που χρησιμοποιούνται από το SMASE μπορεί να προσφερθούν από το Στοιχείο Υπηρεσίας Κοινής Πληροφορίας Διαχείρισης (CMISE - Common Management Information Service Element) ASE (Application Service Element) ή άλλα ASEs τέτοια όπως Μεταφορά, Προσπέλαση και Διαχείριση Αρχείων (FTAM, ISO 8571) ή Επεξεργασία Συναλλαγών (TP, ISO/IEC 10026). Η χρήση του CMISE επίσης συνεπάγεται την παρουσία του Στοιχείου Υπηρεσίας Απομακρυσμένων Λειτουργιών (ROSE, CCITT, Rec. X.219, ISO/IEC 9072). Το CMISE καθορίζει τις υπηρεσίες και τις διαδικασίες για τη μεταφορά Μονάδων Δεδομένων Πρωτοκόλλου Κοινής Πληροφορίας Διαχείρισης (CMIPDUs). Το CMISE παρέχει τα μέσα για την ανταλλαγή πληροφορίας σε λειτουργίες και μηνύματα για σκοπούς διαχείρισης με ένα κοινό τρόπο.

Δύο οντότητες εφαρμογής διαχείρισης συστημάτων εγκαθιστούν μία συσχέτιση μετά από συμφωνία στο πλαίσιο αναφοράς της εφαρμογής. Το πλαίσιο αυτό καθορίζει την αρχικά κοινή διαχειριστική γνώση για τη συσχέτιση αυτή, η οποία περιλαμβάνει τα διάφορα στοιχεία υπηρεσίας εφαρμογής που θα χρησιμοποιηθούν.

Προκειμένου να έχουμε στην πράξη διαχείριση συστημάτων, πρέπει να υπάρχει για το διαχειριστή και τον αντιπρόσωπο κοινή γνώση για τη διαχείριση.

Η διαχειριστική γνώση για επικοινωνίες διαχείρισης συστημάτων περιλαμβάνει (αλλά δεν περιορίζεται σε):

- γνώση των πρωτοκόλλων (π.χ. του πλαισίου αναφοράς εφαρμογής)
- γνώση των λειτουργιών (π.χ. λειτουργιών και ομάδων λειτουργιών)
- γνώση των διαχειριζόμενων αντικειμένων (π.χ. κλάσεις, στιγμιότυπα και αναγνώριση των διαχειριζόμενων αντικείμενων, και των κατηγορημάτων αυτών)
- περιορισμούς στις υποστηριζόμενες λειτουργίες και σχέσεις μεταξύ των λειτουργιών αυτών και των διαχειριζόμενων αντικειμένων. Πιο συγκεκριμένα, σε ένα ανοικτό σύστημα πρέπει να υπάρχουν κάποια ορισμένα διαχειριζόμενα αντικείμενα προκειμένου να υποστηρίζονται συγκεκριμένες λειτουργίες διαχείρισης.



### Σχήμα 5.6 - Λομή του επιπέδου εφαρμογής σε ότι αφορά τη διαχείριση

Η έννοια της κοινής γνώσης για τη διαχείριση γίνεται φανερή στην περίπτωση κατανεμημένων εφαρμογών διαχείρισης, όπου η σχετική γνώση κάθε τελικού συστήματος μπορεί να είναι διαφορετική εάν τα διαχειριζόμενα αντικείμενα που περιέχονται σ' αυτό είναι διαφορετικά (βλ. παρακάτω σχήμα). Η κοινή γνώση για τη διαχείριση αναφέρεται στην κοινή γνώση μεταξύ των δύο συστημάτων, π.χ. στο κοινό σχήμα διαχείρισης.

Υπάρχει ανάγκη δυνατότητας αποδοχής και μεταβολής της κοινής γνώσης για τη διαχείριση που υφίσταται μεταξύ δύο συστημάτων που λαμβάνουν μέρος σε μια ανταλλαγή πληροφορίας διαχείρισης.

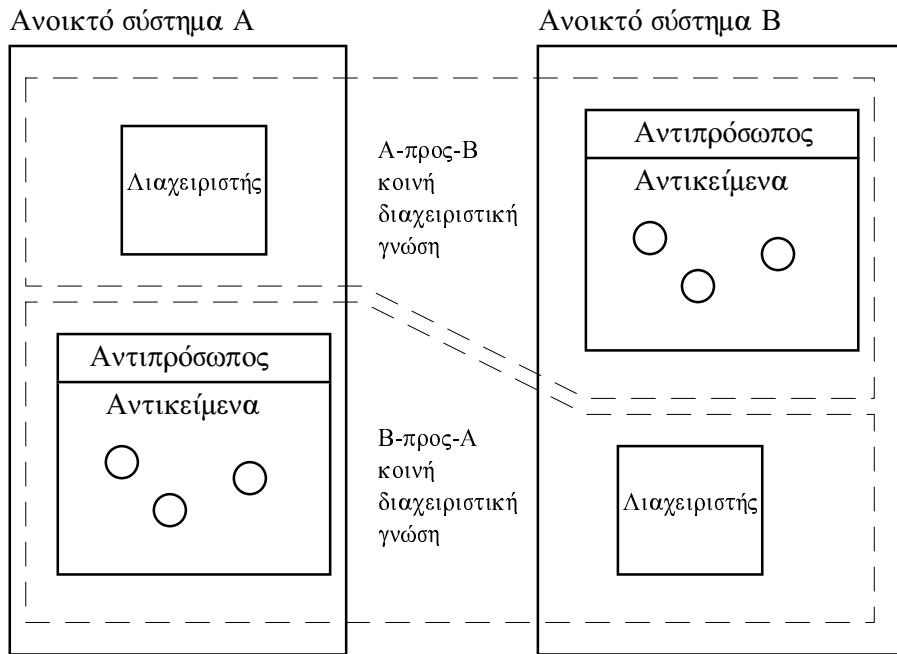
Η κοινή γνώση για τη διαχείριση μπορεί να επιβεβαιωθεί οποιαδήποτε στιγμή, συγκεκριμένα:

- πριν οποιαδήποτε επικοινωνία λάβει χώρα (π.χ. κατά τη σχεδίαση του συστήματος ή την υλοποίηση αυτού, ή να υπάρχει από προηγούμενη συσχέτιση),
- κατά τη φάση εγκατάστασης μιας συσχέτισης,
- στην συνέχεια, κατά τη διάρκεια μιας συσχέτισης.

Η a priori γνώση που απαιτείται για να γίνει δυνατή η μεταφορά πληροφορίας διαχείρισης, είναι ένα παράδειγμα αποδοχής γνώσης για τη διαχείριση.

Κατά τον χρόνο εγκατάστασης μιας συσχέτισης πρέπει να είναι δυνατή είτε η αποδοχή, είτε η μεταβολή της γνώσης για τη διαχείριση.

Μετά την εγκατάσταση μιας συσχέτισης με σκοπό τη διαχείριση συστημάτων, ένας μηχανισμός μπορεί να χρησιμοποιηθεί για την μεταβολή της γνώσης αυτής. Για παράδειγμα ένας μηχανισμός ανακάλυψης της γνώσης μπορεί να υποστηρίζεται από συστήματα που μπορούν να λάβουν τον ρόλο του αντιπροσώπου, προκειμένου να δωθεί η δυνατότητα εξέτασης ενός συστήματος. (Η χρήση ενός τέτοιου μηχανισμού από διαχειριστές πρέπει να αφεθεί σαν προαιρετική).



**Σχήμα 5.7 - Όψεις της κοινής γνώσης διαχείρισης**

Οποιεδήποτε αλλαγές στη κοινή διαχειριστική γνώση μετά το χρόνο συσχέτισης μπορεί να γίνουν με ένα μηχανισμό ανανέωσης της γνώσης.

Κάθε λειτουργία απαιτεί διαφορετικές υπηρεσίες επικοινωνίας. Για παράδειγμα, κάποιες λειτουργίες μπορεί να απαιτήσουν υπηρεσίες προσανατολισμένες προς τη διαχείριση αρχείων, ενώ άλλες μπορεί να απαιτούν ένα απλό πρωτόκολλο αίτησης/απόκρισης.

#### 5.2.1.4. Οργανωτικές πλευρές

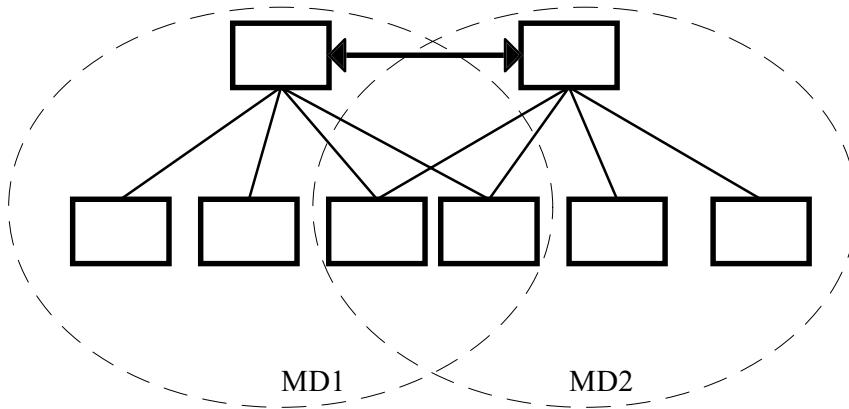
Οι οργανωτικές πλευρές του μοντέλου διαχείρισης συστημάτων περιγράφουν τη κατανεμημένη φύση της διαχείρισης OSI. Πολλές από τις έννοιες που είναι σχετικές με τις οργανωτικές πλευρές της διαχείρισης συστημάτων (π.χ. διαχειριστής, αντιπρόσωπος) έχουν αναφερθεί παραπάνω. Η παράγραφος αυτή αναγνωρίζει και άλλες οργανωτικές πλευρές.

Οι οργανωτικές απαιτήσεις για τη διαχείριση ενός συνόλου από διαχειριζόμενα αντικείμενα περιλαμβάνουν τα ακόλουθα:

- τη διαίρεση του διαχειριζόμενου περιβάλλοντος OSI για ένα σύνολο από λειτουργικούς στόχους (ή "πολιτικές"), τέτοιους όπως ασφάλεια, λογιστικά, διαχείριση σφαλμάτων κ.ά., ή τη διαίρεση του διαχειριζόμενου περιβάλλοντος OSI για άλλους διαχειριστικούς στόχους, για παράδειγμα ανάλογα με γεωγραφικές, τεχνολογικές ή οργανωτικές δομές,
- τη προσωρινή ανάθεση ή πιθανώς μεταβολή των ρόλων διαχειριστή και αντιπροσώπου για κάθε έναν από τους στόχους αυτούς και για κάθε σύνολο από διαχειριζόμενα αντικείμενα, και

- την εξάσκηση μορφών ελέγχου (π.χ. κάποιας πολιτικής ασφάλειας) με ένα σταθερό τρόπο.

Όταν τα διαχειριζόμενα αντικείμενα οργανώνονται σε σύνολα για να ικανοποιήσουν τις παραπάνω απαιτήσεις, τα σύνολα αυτά ονομάζονται **διαχειριστικές περιοχές αρμοδιότητας (Management Domains)**. Οι έννοιες αυτές επεξηγούνται στο παρακάτω σχήμα.



- |  |  |
|--|--|
| <span style="border: 1px solid black; padding: 2px;"> </span>  | Πραγματικό Ανοικτό Σύστημα   |
| MDv  | Διαχειριστική Περιοχή Αρμοδιότητας v   |
| —  | Αλληλεπιδράσεις μεταξύ διαχειριζόμενων συστημάτων σε μία διαχειριστική περιοχή αρμοδιότητας                        |
| <span style="border: 1px solid black; padding: 2px;">←→</span> | Αλληλεπιδράσεις μεταξύ διαχειριζόμενων συστημάτων που ανήκουν σε διαφορετικές διαχειριστικές περιοχές αρμοδιότητας |

### Σχήμα 5.8 - Η έννοια της διαχειριστικής περιοχής αρμοδιότητας

Οι διαχειριστικές περιοχές αρμοδιότητας όταν χρησιμοποιούνται για σκοπούς διαχείρισης συστημάτων, μπορούν να παριστάνονται με την μορφή διαχειριζόμενων αντικειμένων. Μία διαχειριστική περιοχή αρμοδιότητας κατέχει τουλάχιστον τα παρακάτω χαρακτηριστικά:

- μια ονομασία που την αναγνωρίζει με μοναδικό τρόπο,
- ένα αναγνωριστικό του συνόλου των διαχειριζόμενων αντικειμένων που είναι μέλη της διαχειριστικής περιοχής αρμοδιότητας αυτής, και
- ένα αναγνωριστικό των σχέσεων μεταξύ διαφορετικών διαχειριστικών περιοχών αρμοδιότητας.

Οι διαχειριστικές περιοχές αρμοδιότητας μπορεί ή δεν μπορεί να επικαλύπτονται. Όταν διαχειριστικές περιοχές αρμοδιότητας δημιουργημένες για τον ίδιο σκοπό επικαλύπτονται, τότε κάποιες ειδικές συνθήκες πρέπει να εφαρμόζονται.

Πέρα από τις οργανωτικές απαιτήσεις, υπάρχουν επίσης και διοικητικές απαιτήσεις. Αυτές περιλαμβάνουν τα παρακάτω:

- την εγκατάσταση και τη διατήρηση των αντίστοιχων αρχών κάθε διαχειριστικής περιοχής αρμοδιότητας, την εκτέλεση αλλαγών στα σύνορα αυτών, και τη οργάνωση του τρόπου με τον οποίο μερικές από αυτές μπορεί να επικαλύπτονται,
- τη διαχείριση της μεταφοράς του ελέγχου από μία διαχειριστική περιοχή αρμοδιότητας σε μία άλλη.

Για να ικανοποιηθούν οι παραπάνω απαιτήσεις, μια ειδική διαχειριστική περιοχή αρμοδιότητας που ονομάζεται διοικητική διαχειριστική περιοχή αρμοδιότητας ορίζεται. Μια διοικητική διαχειριστική περιοχή αρμοδιότητας είναι μια διαχειριστική περιοχή αρμοδιότητας στην οποία όλα τα διαχειριζόμενα αντικείμενα είναι υπό την ευθύνη μίας και μόνο μίας διοικητικής αρχής.

Διοικητική αρχή για μία διοικητική διαχειριστική περιοχή αρμοδιότητας μπορεί να είναι κάποια διοίκηση (μια διοίκηση δημόσιου τηλεπικοινωνιακού οργανισμού ή κάποιος άλλος οργανισμός που προσφέρει υπηρεσίες επικοινωνίας) ή ένας ιδιωτικός οργανισμός. Ο υπεύθυνος οργανισμός μπορεί να επιλέξει τη χρησιμοποίηση ή όχι διαχειριστης συστημάτων για τον έλεγχο των διαχειριζόμενων αντικειμένων και των διαχειριστικών περιοχών αρμοδιότητας που ανήκουν εξ ολοκλήρου στην διοικητική διαχειριστική περιοχή αρμοδιότητας.

Για το λόγο αυτό η έννοια των διαχειριστικών περιοχών αρμοδιότητας δεν υπόκειται σε προτυποποίηση. Εντούτοις οι αλληλεπιδράσεις που σχετίζονται με την έννοια των διαχειριστικών περιοχών αρμοδιότητας υπόκεινται σε προτυποποίηση.

Ένα παράδειγμα μιας διαχειριστικής περιοχής αρμοδιότητας είναι μια περιοχή διαχειριστης ασφάλειας που περιγράφει τα όρια μιας συγκεκριμένης πολιτικής ασφάλειας σε μια διοικητική διαχειριστική περιοχή αρμοδιότητας. Δύο διαφορετικές διοικητικές αρχές μπορεί να υποστηρίζουν την ίδια πολιτική ασφάλειας μέσα στην ίδια ή και σε διαφορετικές διοικητικές διαχειριστικές περιοχές αρμοδιότητας.

### 5.3. Τα πρωτόκολλα CMIP/CMIS

Στην συνέχεια θα εξετάσουμε αναλυτικότερα την υπηρεσία μεταφοράς της πληροφορίας διαχείρισης. Αναφέραμε και παραπάνω ότι το σύνολο των σχετικών λειτουργιών περιγράφεται από την οντότητα **Common Management Information Service Element (CMISE)**. Η οντότητα CMISE καθορίζεται σε δύο πρότυπα:

- Το πρώτο καθορίζει τις υπηρεσίες που προσφέρονται στο χρήστη του CMISE, και είναι το **Common Management Information Service (CMIS)**.
- Το δεύτερο πρότυπο προδιαγράφει το πρωτόκολλο μεταφοράς της πληροφορίας διαχείρισης, δηλ. τη μορφή των μονάδων δεδομένων πρωτόκολλου (Protocol Data Units - PDUs), και τις σχετικές λειτουργίες. Αυτό είναι το **Common Management Information Protocol (CMIP)**

Το CMIS προσφέρει επτά (7) υπηρεσίες για την εκτέλεση λειτουργιών διαχείρισης, με τη μορφή στοιχειώδων λειτουργιών. Υπηρεσίες προσφέρονται επίσης και για την εγκατάσταση συσχετίσεων σε επίπεδο εφαρμογής. Οι υπηρεσίες αυτές, οι οποίες προσφέρονται από το **Association Control Service Element (ACSE)**, προσφέρονται στο χρήστη της οντότητας CMISE χωρίς τη παρεμβολή του CMIP (αυτός καλεί άμεσα

στοιχειώδεις λειτουργίες που προσφέρει η οντότητα ACSE). Για την εκτέλεση λειτουργιών διαχείρισης η οντότητα CMISE χρησιμοποιεί το CMIP για την ανταλλαγή κατάλληλων PDUs. Αυτό με τη σειρά χρησιμοποιεί τις υπηρεσίες της οντότητας **Remote Operations Service Element (ROSE)**.

### 5.3.1. Common Management Information Service

Το πρότυπο που ορίζει το CMIS περιγράφει τις υπηρεσίες που προσφέρονται προκειμένου να επιτευχθεί η διαχείριση συστημάτων OSI. Οι υπηρεσίες του CMIS καθορίζονται με τη μορφή στοιχειώδων λειτουργιών και μπορούν να θεωρηθούν σαν κλήσεις διαδικασιών με παραμέτρους. Οι υπηρεσίες αυτές είναι οι παρακάτω:

<b>(α) Υπηρεσίες αποστολής μηνυμάτων</b>		
<b>Υπηρεσία</b>	<b>Τύπος</b>	<b>Ορισμός</b>
M-EVENT-REPORT	Confirmed/nonconfirmed	Αναφέρει ένα γεγονός σχετικό με ένα διαχειριζόμενο αντικείμενο σε ένα ομότιμο χρήστη CMISE-υπηρεσιών
<b>(β) Υπηρεσίες λειτουργιών διαχείρισης</b>		
<b>Υπηρεσία</b>	<b>Τύπος</b>	<b>Ορισμός</b>
M-GET	Confirmed	Απαιτεί την ανάκτηση πληροφορίας διαχείρισης από ένα ομότιμο χρήστη CMISE-υπηρεσιών
M-SET	Confirmed/nonconfirmed	Απαιτεί τη μεταβολή πληροφορίας διαχείρισης από ένα ομότιμο χρήστη CMISE-υπηρεσιών
M-ACTION	Confirmed/nonconfirmed	Απαιτεί ο ομότιμος χρήστη CMISE-υπηρεσιών να εκτελέσει μια ενέργεια
M-CREATE	Confirmed	Απαιτεί ο ομότιμος χρήστη CMISE-υπηρεσιών να δημιουργήσει το στιγμιότυπο ενός αντικειμένου
M-DELETE	Confirmed	Απαιτεί ο ομότιμος χρήστη CMISE-υπηρεσιών να καταργήσει το στιγμιότυπο ενός αντικειμένου
M-CANCEL--GET	Confirmed	Απαιτεί ο ομότιμος χρήστη CMISE-υπηρεσιών να ακυρώσει μια προηγούμενη αίτηση M-GET υπηρεσίας

Όπως παρατηρούμε από τον παραπάνω πίνακα υπάρχουν δύο κατηγορίες υπηρεσιών διαχείρισης. Σε αυτές μπορούμε να προσθέσουμε και μια τρίτη, αυτή των υπηρεσιών του ACSE για την εγκατάσταση και την κατάργηση συσχετίσεων.

Το CMIS προσφέρει επιπρόσθετα ευκολίες για πιο αποδοτική χρησιμοποίηση των παραπάνω υπηρεσιών. Σύμφωνα με αυτές:

- Πολλαπλές αποκρίσεις σε μια λειτουργία με επιβεβαίωση μπορούν να συνδεθούν με τη λειτουργία αυτή με τη χρησιμοποίηση της παραμέτρου linked-identification.
- Λειτουργίες μπορούν να εκτελεστούν ταυτόχρονα σε περισσότερα από ένα διαχειριζόμενα αντικείμενα, τα οποία ικανοποιούν κάποια κριτήρια και υπόκεινται σε μια συνθήκη συγχρονισμού.

Πιο συγκεκριμένα, όσο αναφορά την πρώτη ευκολία θα προσθέσουμε τα εξής. Οι στοιχειώδεις υπηρεσίες M-GET, M-SET, M-ACTION, M-DELETE μπορούν να εφαρμοστούν σε περισσότερα από ένα διαχειριζόμενα αντικείμενα. Παρ' ότι η κλήση της υπηρεσίας είναι μία, θα έχουμε περισσότερες επιβεβαιώσεις. Χρειάζεται λοιπόν κάποιος τρόπος σύνδεσης των επιβεβαιώσεων αυτών με την αρχική λειτουργία, προκειμένου να γνωρίζουμε κάθε στιγμή ποια από τις υπηρεσίες που έχουν κληθεί έχει επιβεβαιωθεί. Για να επιτευχθεί αυτό αρκεί η παράμετρος linked-identifier που εμφανίζεται σε κάθε μία από τις στοιχειώδεις υπηρεσίες απόκρισης και επιβεβαιώσης να έχει την τιμή της παραμέτρου invoke-identifier που εμφανίζεται στις στοιχειώδεις υπηρεσίες αίτησης και ένδειξης.

Σε ότι αφορά τη δεύτερη ευκολία, το CMIS προσφέρει ένα σύνολο δυνατοτήτων προκειμένου ο χρήστης των υπηρεσιών του να μπορεί να επιλέξει ένα αντικείμενο ή περισσότερα σαν το στόχο των λειτουργιών διαχείρισης που επιθυμεί. Οι δυνατότητες αυτές προσφέρονται σαν παράμετροι στις στοιχειώδεις λειτουργίες M-GET, M-SET, M-ACTION και M-DELETE και οι εξής: **scoping (οριοθέτησης), filtering (φίλτραρισμάτος) και synchronization (συγχρονισμού)**.

- Η δυνατότητα scoping αναφέρεται στην αναγνώριση ενός ή περισσοτέρων αντικειμένων προκειμένου να εφαρμόσουμε σ' αυτά κάποιο φίλτρο. Η δυνατότητα αυτή αναφέρεται στα διαχειριζόμενα αντικείμενα χρησιμοποιώντας σαν σημείο αναφοράς ένα **base managed object**. Σημειώνουμε ότι τα στιγμιότυπα των διαχειριζόμενων αντικειμένων σχηματίζουν κάποιο δένδρο ιεραρχίας. Χρησιμοποιώντας λοιπόν κάθε φορά το base managed object σαν ρίζα, έχουμε ένα υπο-δένδρο, μέσα στο συνολικό δένδρο ιεραρχίας. Με βάση αυτά έχουμε στη διάθεσή μας τέσσερις τρόπους καθορισμού διαχειριζόμενων αντικειμένων.
  1. Μονάχα το based managed object
  2. Όλα τα αντικείμενα που βρίσκονται στο n-υποεπίπεδο, θεωρώντας το based managed object σαν 0-επίπεδο
  3. Το based managed object και τα υποκείμενά του μέχρι και το n-υποεπίπεδο.
  4. Το based managed object και όλα τα υποκείμενά του, δηλ. ολόκληρο το υποδένδρο.
- Το φίλτρο το οποίο χρησιμοποιείται στη συνέχεια προκειμένου να προσδιορίσουμε με μεγαλύτερη ακρίβεια τα διαχειριζόμενα αντικείμενα που μας ενδιαφέρουν είναι στην ουσία μια Boolean έκφραση, η οποία αποτελείται από μία ή περισσότερες υποθέσεις σχετικά με τις τιμές κατηγορημάτων των διαχειριζόμενων αντικειμένων που βρίσκονται μέσα στο scope. Στις υποθέσεις μπορούν να χρησιμοποιηθούν οι παρακάτω κανόνες.
  1. **Equality:** Η τιμή του κατηγορήματος πρέπει να είναι ίδια με αυτή της υπόθεσης
  2. **Greater or equal:** Η τιμή του κατηγορήματος πρέπει να είναι μεγαλύτερη ή ίση από αυτή της υπόθεσης
  3. **Less or equal:** Η τιμή του κατηγορήματος πρέπει να είναι μικρότερη ή ίση από αυτή της υπόθεσης
  4. **Present:** Το κατηγόρημα πρέπει να υφίσταται
  5. **Substrings:** Η τιμή του κατηγορήματος πρέπει να περιλαμβάνει τα καθορισμένα substrings με τη δεδομένη σειρά
  6. **Subset of:** Όλα τα μέλη της υπόθεσης πρέπει να είναι παρών στο κατηγόρημα

7. ***Superset of***: Όλα τα μέλη του κατηγορήματος πρέπει να είναι παρών στην υπόθεση
8. ***Non-null-set-intersection***: Τουλάχιστον ένα από τα μέλη της υπόθεσης πρέπει να είναι παρών στο κατηγόρημα

Μετά την επιτυχή ολοκλήρωση του scoping και του filtering μπορεί να έχουμε οδηγηθεί σε περισσότερα από ένα διαχειριζόμενα αντικείμενα στα οποία θα πρέπει να εφαρμοστεί η λειτουργία διαχείρισης. Τίθεται τώρα θέμα της σειράς με την οποία θα εφαρμοστεί η λειτουργία διαχείρισης στα επιλεγμένα αντικείμενα. Αν και το πρόβλημα αυτό λύνεται κατά την υλοποίηση των πρωτοκόλλων και όχι με πρότυπο τρόπο, το CMIS με κατάλληλες παραμέτρους στις κλήσεις των στοιχειωδών λειτουργιών δίνει δύο επιπλέον δυνατότητες εφαρμογής λειτουργιών διαχείρισης σε επιλεγμένα αντικείμενα.

- ***Atomic***: Στην περίπτωση αυτή γίνεται έλεγχος κατά πόσον τα διαχειριζόμενα αντικείμενα είναι όλα σε θέση να εκτελέσουν τη λειτουργία διαχείρισης. Αν ένα ή περισσότερα δεν μπορεί να εκτελέσει τη λειτουργία αυτή, τότε δεν εκτελείται από κανένα.
- ***Best effort***: Στην δεύτερη περίπτωση εκτελούν τη λειτουργία διαχείρισης όσα διαχειριζόμενα αντικείμενα έχουν τη δυνατότητα, ανεξάρτητα με το αν κάποια από αυτά δεν θα την εκτελέσουν.

### 5.3.2. Common Management Information Protocol

Το CMIP ορίζει τις διαδικασίες για τη μετάδοση της πληροφορίας διαχείρισης και το συντακτικό ορισμό των αναλόγων υπηρεσιών. Το πρωτόκολλο αυτό ορίζεται με βάση εννέα (9) μονάδες δεδομένων πρωτοκόλλου (PDUs) οι οποίες ανταλλάσσονται μεταξύ ομοτίμων CMISEs προκειμένου να υλοποιήσουν CMIS υπηρεσίες.

Τα PDUs τα οποία ορίζει το CMIP φαίνονται στο παρακάτω πίνακα, μαζί με τα ορίσματα και τα αποτελέσματα:

Σαν παράδειγμα της χρήσης του CMIP μπορούμε να εξετάσουμε την υπηρεσία M-GET. Η ακόλουθη αλληλουχία γεγονότων συμβαίνει:

1. Μια στοιχειώδης λειτουργία M-GET.request λαμβάνεται από ένα χρήστη του στοιχείου CMISE. Οι παράμετροι στην κλήση αυτή ξεχωρίζουν τη συγκεκριμένη λειτουργία από τις υπόλοιπες που εκρεμούν.
2. Η μηχανή υλοποίησης του πρωτοκόλλου κατασκευάζει μια m-Get μονάδα-δεδομένων-πρωτοκόλλου-εφαρμογής (Application PDU - APDU) η οποία βέβαια θα περιλαμβάνει τις παραμέτρους της στοιχειώδους λειτουργίας M-GET.request.
3. Η μηχανή υλοποίησης του πρωτοκόλλου χρησιμοποιεί μια στοιχειώδης λειτουργία RO-INVOKE.request του ROSE προκειμένου να στείλει την APDU στον προορισμό της.
4. Το ROSE παραδίδει την APDU στη μηχανή υλοποίησης του πρωτοκόλλου CMISE της μηχανής προορισμού με μια στοιχειώδης λειτουργία RO-INVOKE.indication.
5. Εάν η μονάδα δεδομένων είναι αποδεκτή η μηχανή υλοποίησης του πρωτοκόλλου εκδίδει μια στοιχειώδης λειτουργία M-GET.indication στο χρήστη

του στοιχείου CMISE, παρέχοντας τις απαραίτητες πληροφορίες. Αυτό κατευθύνεται στο ομότιμο χρήστη της υπηρεσίας προκειμένου αυτός να εκτελέσει την αιτούμενη GET λειτουργία και να αναφέρει τα αποτελέσματα.

CMIP PDUs	Ορίσματα	Αποτελέσματα
m-EventReport	managedObjectClass, managedObjectInstance, eventTime, eventType, eventInfo	-
m-EventReport-Confirmed	managedObjectClass, managedObjectInstance, eventTime, eventType, eventInfo	managedObjectClass, managedObjectInstance, currentTime, eventReply
m-Get	baseManagedObjectClass, baseManagedObjectInstance, accessControl, synchronization, scope, filter, attributeIdList	managedObjectClass, managedObjectInstance, currentTime, attributeList
m-Linked-Reply	getResult, getListError, setResult, setListError, actionResult, processingFailure, deleteResult, actionError, deleteError	-
m-Set	baseManagedObjectClass, baseManagedObjectInstance, accessControl, synchronization, scope, filter, modificationList	-
m-Set-Confirmed	baseManagedObjectClass, baseManagedObjectInstance, accessControl, synchronization, scope, filter, modificationList	managedObjectClass, managedObjectInstance, currentTime, attributeList
m-Action	baseManagedObjectClass, baseManagedObjectInstance, accessControl, synchronization, scope, filter, actionInfo	-
m-Action-Confirmed	baseManagedObjectClass, baseManagedObjectInstance, accessControl, synchronization, scope, filter, actionInfo	managedObjectClass, managedObjectInstance, currentTime, actionReply
m>Create	managedObjectClass, objectInstance, accessControl, referenceObjectInstance, attributeList	managedObjectClass, managedObjectInstance, currentTime, attributeList
m-Delete	baseManagedObjectClass, baseManagedObjectInstance, accessControl, synchronization, scope, filter	managedObjectClass, managedObjectInstance, currentTime
m-Cancel-Get-Confirmed	getInvokeId	-

6. Ο αποκρινόμενος χρήστης του CMISE εκδίδει μια στοιχειώδη λειτουργία M-GET.response. Οι παράμετροι στην απόκριση αυτή ξεχωρίζουν τη συγκεκριμένη λειτουργία από τις υπόλοιπες που εκρεμούν, και παρέχουν την πληροφορία που ζητήθηκε για το διαχειριζόμενο αντικείμενο και άλλη σχετική πληροφορία. Στην περίπτωση που η λειτουργία αυτή αποτύχει η στοιχειώδης λειτουργία περιλαμβάνει και μια παράμετρο λάθους προκειμένου να περιγράψει το είδος του λάθους.
7. Η αποκρινόμενη μηχανή υλοποίησης του πρωτοκόλλου CMISE κατασκευάζει ένα m-Get APDU το οποίο περιλαμβάνει τις παραμέτρους της στοιχειώδους λειτουργίας M-GET.response.
8. Εάν η λειτουργία είναι επιτυχής η μηχανή υλοποίησης του πρωτοκόλλου CMISE χρησιμοποιεί μια υπηρεσία του ROSE RO-RESULT.request για να στείλει την APDU πίσω στο αρχικό σύστημα.

9. Το ROSE παραδίδει την APDU στο αρχικό σύστημα με ένα RO-RESULT.indication.
10. Η μηχανή υλοποίησης του πρωτοκόλλου CMISE εκδίδει μια στοιχειώδη λειτουργία M-GET.confirmation στον αρχικό CMISE-χρήστη.

## 5.4. Βάση Πληροφορίας Διαχείρισης OSI (OSI-MIB)

Βασικό στοιχείο κάθε Συστήματος Διαχείρισης Δικτύων είναι **η Βάση Πληροφορίας Διαχείρισης (Management Information Base - MIB)**. Αυτή είναι μια νοητή αποθήκη πληροφορίας διαχείρισης σε ένα ανοικτό σύστημα. Η πληροφορία αυτή είναι δυνατό να μεταβληθεί ή να μεταφερθεί με τη βοήθεια πρωτοκόλλων διαχείρισης. Το γενικό πλαίσιο σύμφωνα με το οποίο κάποια MIB μπορεί να οριστεί ονομάζεται **Λογική Της Πληροφορίας Διαχείρισης (Structure of Management Information - SMI)**.

Η Λογική της Πληροφορίας Διαχείρισης OSI χρησιμοποιεί το συντακτικό ASN.1 και έννοιες από τον αντικειμενοστραφή σχεδιασμό, προκειμένου να ορίσει τη πληροφορία διαχείρισης. Κάθε στοιχείο του δικτύου που παρακολουθείται και ελέγχεται, παριστάνεται με ένα διαχειριζόμενο αντικείμενο, το οποίο είναι η βασική μονάδα πληροφορίας διαχείρισης. Το αντικείμενο αυτό μπορεί να περιλαμβάνει:

- **Κατηγορίατα (attributes):** δηλ. μεταβλητές που παριστάνουν χαρακτηριστικά των στοιχείων του δικτύου που θέλουμε να διαχειριστούμε.
- **Συμπεριφορά (behaviour):** δηλ. λειτουργίες που προκαλούνται από κάποιο διαχειριστή.
- **Μηνύματα (notifications):** δηλ. αναφορές γεγονότων που προκαλούνται από ορισμένα γεγονότα.

Η MIB είναι μια δομημένη συλλογή τέτοιων αντικειμένων. Διαχειριζόμενο αντικείμενο μπορεί να οριστεί για κάθε στοιχείο του δικτύου, το οποίο θέλουμε να διαχειριστούμε. Σαν παραδείγματα μπορούμε να αναφέρουμε: switches, workstations, PBX's (private branch exchanges), LANs (Local Area Networks), boards, transport protocols, routing algorithms, κ.ά. Σημειώνουμε τις εξής χρήσιμες πληροφορίες:

- Ένα διαχειριζόμενο αντικείμενο είναι μια αφαίρεση του πραγματικού στοιχείου του δικτύου, η οποία είναι διαθέσιμη για τους σκοπούς της διαχείρισης. Κάποιος μηχανισμός, ο οποίος δεν ενδιαφέρει τα πρότυπα διαχείρισης OSI, ελέγχει τη σχέση μεταξύ του διαχειριζόμενου αντικειμένου και του πραγματικού στοιχείου του δικτύου.
- Μερικά διαχειριζόμενα αντικείμενα ορίζονται απλά για να υποστηρίζουν τις λειτουργίες διαχείρισης και δεν παριστάνουν κάποιο πραγματικό στοιχείο του δικτύου. Παραδείγματα αποτελούν τα φίλτρα (filters) και τα ημερολόγια γεγονότων (event logs).
- Μεταξύ διαχειριζόμενων αντικειμένων και πραγματικών στοιχείων του δικτύου είναι δυνατό να υφίστανται σχέσεις όπως ένα-προς-ένα, ένα-προς-πολλά, ή πολλά-προς-ένα.

Από τα παραπάνω γίνεται φανερό ότι τα αντικείμενα της OSI-MIB είναι πολύ κοντά στα αντικείμενα που προκύπτουν από τον αντικειμενοστραφή σχεδιασμό. Εύκολα καταλαβαίνουμε ότι χρησιμοποιώντας ιδιότητες όπως η κληρονομικότητα (inheritance)

μπορούμε να κατασκευάσουμε μια ιεραρχία από κλάσεις και υπο-κλάσεις, η οποία θα παριστάνει τις σχέσεις μεταξύ διαφόρων τύπων αντικειμένων. Η ιδιότητα της κληρονομικότητας μας επιτρέπει να ορίζουμε έναν μεγάλο αριθμό αντικειμένων, γράφοντας πολύ λιγότερες γραμμές κώδικα. Επίσης είναι ένα χρήσιμο εργαλείο προκειμένου να σχεδιάζουμε αντικείμενα για MIBs. Παρ' όλα αυτά **η ιεραρχία κληρονομικότητας (inheritance tree)** δεν είναι αυτή που θα καθορίσει τη δομή μιας πραγματικής MIB.

Προκειμένου να δομήσουμε μια OSI-MIB χρησιμοποιούμε την έννοια **containment** η οποία επιτρέπει σε ένα αντικείμενο να "περιέχει" ένα άλλο. Αυτό επιτυγχάνεται περιλαμβάνοντας ένα δείκτη στο ανώτερο αντικείμενο που να δείχνει στο αντικείμενο που "περιέχει". Ο δείκτης αυτός αποτελεί την τιμή κάποιου κατηγορήματος του ανώτερου αντικειμένου. Ένα αντικείμενο επιτρέπεται να "περιέχεται" σε ένα μόνο ανώτερο αντικείμενο, αναγκάζοντας τη δομή της OSI-MIB να πάρει μορφή δένδρου. Προφανώς το ανώτερο αντικείμενο μπορεί να "περιέχεται" σε κάποιο άλλο, οπότε το βάθος του δένδρου είναι απεριόριστο. Η δομή αυτή μπορεί να υλοποιήσει εύκολα πραγματικές ιεραρχικές δομές, όπως στοιχεία δικτύου, καταλόγους, αρχεία, πεδία, κ.ά.

Ένα τρίτο δένδρο είναι το γνωστό μας **registration tree** το οποίο μας επιτρέπει να δίνουμε μοναδικά ονόματα σε αντικείμενα (π.χ. κλάσεις διαχειριζόμενων αντικειμένων). Το πρόβλημα στη διαχείριση OSI είναι ότι έχουμε διαφοροποίηση μεταξύ των κλάσεων των διαχειριζόμενων αντικειμένων και των στιγμιοτύπων των κλάσεων αυτών (στην περίπτωση του SNMP αυτά ταυτίζονται).

Η ονομασία των στιγμιοτύπων διαχειριζόμενων αντικειμένων δεν ακολουθεί το registration tree αλλά τις containment σχέσεις, που εξετάσαμε παραπάνω. Κάθε κλάση διαχειριζόμενων αντικειμένων περιλαμβάνει ένα κατηγόρημα, το οποίο χρησιμοποιείται για τις ανάγκες της ονομασίας. Το **relative distinguished name** ενός στιγμιότυπου αντικειμένου αντιστοιχεί σε μια συγκεκριμένη τιμή του παραπάνω κατηγορήματος. Η τιμή αυτή πρέπει να είναι μοναδική μεταξύ αντικειμένων που υπόκεινται στο ίδιο ανώτερο αντικείμενο. Η πραγματική μορφή ενός relative distinguished name είναι μια υπόθεση ότι το κατηγόρημα έχει μια συγκεκριμένη τιμή - για παράδειγμα VirtCircId = "12", όπου VirtCircId είναι το όνομα του κατηγορήματος και 12 είναι η επιθυμητή τιμή.

Το **distinguished name** ενός στιγμιοτύπου αντικειμένου σχηματίζεται σαν μια σειρά από relative distinguished name από την ρίζα του containment tree μέχρι το συγκεκριμένο στιγμιότυπο.

#### 5.4.1. Μοντέλο Πληροφορίας Διαχείρισης

Το πρότυπο ISO 10165-1 (X.720) περιγράφει ένα γενικό μοντέλο της πληροφορίας διαχείρισης για τη διαχείριση συστημάτων OSI. Πιο συγκεκριμένα το πρότυπο αυτό:

- ορίζει το μοντέλο πληροφορίας των διαχειριζόμενων αντικειμένων και τα κατηγορήματά τους,
- ορίζει τις αρχές της ονομασίας των διαχειριζόμενων αντικειμένων και των κατηγορημάτων τους, έτσι ώστε να είναι δυνατή η αναγνώρισή τους από πρωτόκολλα διαχείρισης,
- ορίζει τη λογική δομή της πληροφορίας διαχείρισης, δηλ το SMI,

- περιγράφει τις έννοιες των κλάσεων διαχειριζόμενων αντικειμένων και των σχέσεων που αυτές μπορεί να έχουν, όπως κληρονομικότητα (inheritance), specialization, allomorphism, και containment.

Πιο συγκεκριμένα, ένα διαχειριζόμενο αντικείμενο ορίζεται με βάση τα **κατηγορήματα (attributes)**, τα οποία κατέχει, τις **λειτουργίες (operations)**, που είναι δυνατό να εκτελεστούν σ' αυτό, τα **μηνύματα (notifications)** που μπορεί αυτό να εκδόσει, και τέλος τις **σχέσεις (relationships)** του με άλλα διαχειριζόμενα αντικείμενα. Προκειμένου να δώσουμε μια δομή στην MIB, θεωρούμε κάθε διαχειριζόμενο αντικείμενο σαν στιγμιότυπο μιας **κλάσης διαχειριζόμενων αντικειμένων (managed-object class)**. Μια κλάση διαχειριζόμενων αντικειμένων αποτελεί ένα μοντέλο ή ένα template για στιγμιότυπα διαχειριζόμενων αντικειμένων, τα οποία μοιράζονται τα ίδια κατηγορήματα, δέχονται τις ίδιες λειτουργίες και εκδίδουν τα ίδια μηνύματα.

Οι προδιαγραφές της OSI-SMI και MIB στηρίζονται σημαντικά στις ιδέες του αντικειμενοστραφή σχεδιασμού. Αυτό βέβαια δεν σημαίνει ότι οι MIBs πρέπει να υλοποιούνται με αντικειμενοστραφή συστήματα διαχείρισης βάσεων δεδομένων, ή αντικειμενοστραφή τεχνολογία. Το μόνο ζητούμενο είναι η πληροφορία που μεταφέρεται μεταξύ ανοικτών συστημάτων με πρωτόκολλα διαχείρισης συστημάτων (π.χ. CMIP - Common Management Information Protocol), πρέπει να έχει προσδιοριστεί σύμφωνα με τις αρχές της αντικειμενοστραφούς σχεδίασης.

#### 5.4.2. Ορισμός της Πληροφορίας Διαχείρισης

Το πρότυπο ISO/IEC 10165-2 | X. 721 "Δομή της πληροφορίας διαχείρισης: Ορισμός της πληροφορίας διαχείρισης":

- ορίζει βασικές κλάσεις διαχειριζόμενων αντικειμένων, τύπους κατηγορημάτων, ειδικά κατηγορήματα, τύπους μηνυμάτων σύμφωνα με τη σύσταση ISO/IEC 10165-4 | CCITT Rec. X.722. Οι κλάσεις αυτές μπορούν να χρησιμοποιηθούν σαν υπερ-κλάσεις με κύριο στόχο την κληρονομικότητα στον ορισμό νέων κλάσεων Διαχειριζόμενων αντικειμένων σε άλλα πρότυπα. Τις κλάσεις αυτές χρησιμοποιούν επίσης και οι λειτουργίες διαχείρισης συστημάτων (βλ. και παρακάτω).
- καθορίζει απαιτήσεις υπακοής για άλλες συστάσεις που θα χρησιμοποιήσουν τους ορισμούς αυτούς.

Κλάσεις διαχειριζόμενων αντικειμένων, οι οποίες ορίζονται, είναι οι εξής<sup>\*</sup>:

##### 1. Alarm record

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει καταγραμμένη πληροφορία η οποία έχει προκύψει από μηνύματα συναγερμών ή αναφορές γεγονότων.

##### 2. Attribute value change record

---

\* Στην συνέχεια δίνουμε μια σύντομη περίληψη μέρους των ορισμών που μπορεί να συναντήσει κανείς στο αντίστοιχο πρότυπο. Για τον πλήρη ορισμό της σύστασης μπορεί να δει κανείς: ISO/IEC 10165-4, Information technology - Open Systems Interconnection - Structure of Management Information: Definition of management information, International Standard, October 1992 ή άλλη νεώτερη έκδοση αυτού.

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει καταγραμμένη πληροφορία η οποία έχει προκύψει από μηνύματα σχετικά με την αλλαγή τιμών κατηγορημάτων ή σχετικών αναφορών γεγονότων.

### 3. Discriminator

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει τα κριτήρια ελέγχου υπηρεσιών διαχείρισης.

### 4. Event forwarding discriminator

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει τα κριτήρια τα οποία πρέπει να ικανοποιούν δυνατές αναφορές γεγονότων πριν το γεγονός προωθηθεί σε ένα συγκεκριμένο προορισμό.

### 5. Event log record

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει πληροφορία αποθηκευμένη σε ένα ημερολόγιο σαν το αποτέλεσμα λήψης μηνυμάτων ή αναφορών γεγονότων.

### 6. Log

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου για την αποθήκευση αναφορών γεγονότων και τοπικών μηνυμάτων που έστειλε το σύστημα.

### 7. Log record

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει πληροφορία αποθηκευμένη σε ένα ημερολόγια.

### 8. Object creation record

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει καταγραμμένη πληροφορία η οποία έχει προκύψει από μηνύματα σχετικά με τη δημιουργία αντικειμένων ή σχετικών αναφορών γεγονότων.

### 9. Object deletion record

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει καταγραμμένη πληροφορία η οποία έχει προκύψει από μηνύματα σχετικά με την κατάργηση αντικειμένων ή σχετικών αναφορών γεγονότων.

### 10. Relationship change record

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει καταγραμμένη πληροφορία η οποία έχει προκύψει από μηνύματα σχετικά με την αλλαγή σχέσεων ή σχετικών αναφορών γεγονότων.

## **11. Security alarm report record**

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει καταγραμμένη πληροφορία η οποία έχει προκύψει από μηνύματα που αφορούσαν συναγερμούς ασφάλειας ή σχετικών αναφορών γεγονότων.

## **12. State change record**

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει καταγραμμένη πληροφορία η οποία έχει προκύψει από μηνύματα σχετικά με την αλλαγή καταστάσεων ή σχετικών αναφορών γεγονότων.

## **13. System**

Το αντικείμενο αυτό χρησιμοποιείται προκειμένου να αναπαραστήσει ένα σύνολο από hardware και software το οποίο είναι αυτόνομο και έχει τη δυνατότητα επεξεργασίας της πληροφορίας και/ή μεταφοράς αυτής.

## **14. Top**

Αυτή είναι η υψηλότερη δυνατή κλάση της οποίας όλες οι υπόλοιπες κλάσεις είναι υποκλάσεις.

Προκειμένου να γίνουν φανερά και όσα αναφέραμε στην 5.4.1. παραθέτουμε τον πλήρη ορισμό μιας από τις παραπάνω κλάσεις.

```

securityAlarmReportRecord   MANAGED OBJECT CLASS
DERIVED FROM   eventLogRecord;
CHARACTERIZED BY
-- The appropriate object identifier values for the eventType attribute, inherited from
-- eventLogRecord managed object class, are integrityViolation, physicalViolation,
-- securityServiceOrMechanismViolation and timeDomainViolation --
securityAlarmRecordPackage      PACKAGE
                                BEHAVIOUR
                                securityAlarmReportRecordBehaviour          BEHAVIOUR
                                DEFINED AS "This managed object class is used to represent logged
                                information that resulted from security alarm notifications or event reports";;
ATTRIBUTES
                                securityAlarmCause        GET,
                                securityAlarmSeverity     GET,
                                securityAlarmDetector    GET,
                                serviceUser              GET,
                                serviceProvider           GET;;;
```

REGISTERED AS {smi2MObjectClass 11};

Τύποι μηνυμάτων οι οποίοι ορίζονται στο πρότυπο αυτό είναι οι παρακάτω:

### **1. Attribute value change**

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει αλλαγές σε ένα κατηγόρημα όπως την αλλαγή τιμών ενός ή περισσοτέρων κατηγορημάτων ή την αλλαγή των τιμών ενός κατηγορήματος στις default τιμές.

## 2. Communications alarm

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει ότι το αντικείμενο αναγνώρισε ένα επικοινωνιακό σφάλμα.

## 3. Environmental alarm

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει ένα σφάλμα στο περιβάλλον.

## 4. Equipment alarm

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει μια βλάβη στα μηχανήματα.

## 5. Integrity violation

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει ότι πιθανή διακοπή στη ροή της πληροφορίας έχει συμβεί τέτοια ώστε η πληροφορία να έχει πιθανώς μεταβληθεί χωρίς ανάλογο δικαίωμα.

## 6. Object creation

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει τη δημιουργία ενός αντικειμένου σε ένα ανοικτό σύστημα.

## 7. Object deletion

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει την κατάργηση ενός αντικειμένου σε ένα ανοικτό σύστημα.

## 8. Operational violation

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει ότι η παροχή της συγκεκριμένης υπηρεσίας δεν ήταν δυνατή λόγω μη διαθεσιμότητας, κακής λειτουργίας, ή λανθασμένης κλήσης της υπηρεσίας.

## 9. Physical violation

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει ότι κάποιος φυσικός πόρος έχει μεταβληθεί με τέτοιο τρόπο, ώστε να είναι πιθανή επίθεση στην ασφάλεια.

## 10. Processing error alarm

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει λανθασμένη επεξεργασία σε κάποιο αντικείμενο.

## 11. Quality of service alarm

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει μια αποτυχία στην ποιότητα υπηρεσίας του διαχειριζόμενου αντικειμένου.

## 12. Relationship change

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει μια αλλαγή στην τιμή ενός ή περισσοτέρων κατηγορημάτων σχέσεων ενός διαχειριζόμενου αντικειμένου, σαν αποτέλεσμα είτε εσωτερικής ενέργειας του διαχειριζόμενου αντικειμένου, είτε λειτουργίας διαχείρισης.

## 13. Security service or mechanism violation

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει μια επίθεση στην ασφάλεια που αναγνωρίστηκε από μια υπηρεσία ασφάλειας ή από ένα μηχανισμό ασφάλειας.

## 14. State change

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει μια αλλαγή στην τιμή ενός ή περισσοτέρων κατηγορημάτων κατάστασης ενός διαχειριζόμενου αντικειμένου, σαν αποτέλεσμα είτε εσωτερικής ενέργειας του διαχειριζόμενου αντικειμένου, είτε λειτουργίας διαχείρισης.

## 15. Time domain violation

Το μήνυμα αυτό χρησιμοποιείται προκειμένου να αναφέρει ότι ένα γεγονός έχει λάβει χώρα σε μη αναμενόμενο ή απαγορευμένο χρόνο.

Παραθέτουμε παρακάτω τον πλήρη ορισμό ενός τύπου μηνύματος για καλύτερη κατανόηση:

```
objectCreation NOTIFICATION
    BEHAVIOUR objectCreationBehaviour;
    WITH INFORMATION SYNTAX Notification-ASN1Module.ObjectInfo
        AND ATTRIBUTE IDS
            sourceIndicator          sourceIndicator,
            attributeList           attributeList,
            notificationIdentifier notificationIdentifier,
            correlatedNotifications correlatedNotifications,
            additionalText           additionalText,
            additionalInformation    additionalInformation;
```

REGISTERED AS {smi2Notification 6};

```
objectCreationBehaviour
BEHAVIOUR
    DEFINED AS "This notification type is used to report the creation of a
managed
    object to another open system.;"
```

### 5.4.3. GDMO (Guidelines for the Definition of Managed Objects)

Το 4ο μέρος της σειράς προτύπων για τον ορισμό της Δομής της Πληροφορίας Διαχείρισης OSI είναι το πρότυπο “*Guidelines for the definition of managed objects*”. Η προδιαγραφή αυτή παρέχει στους συγγραφείς MIBs οδηγίες:

- για τη διατήρηση της συμβατότητας μεταξύ ορισμών διαχειριζόμενων αντικειμένων,
- για την εξασφάλιση της ανάπτυξης ορισμών διαχειριζόμενων αντικειμένων συμβατών με τους υπόλοιπους ορισμούς της διαχείρισης OSI,
- για τον περιορισμό της επανάληψης εργασίας (με την παροχή κοινά χρησιμοποιούμενων μορφών εγγράφων, διαδικασιών και ορισμών).

Πιο συγκεκριμένα το πρότυπο αυτό καθορίζει:

- τις σχέσεις μεταξύ των σχετικών Συστάσεων/Προτύπων διαχείρισης OSI και των ορισμών των κλάσεων των διαχειριζόμενων αντικειμένων,
- κατάλληλες μεθόδους για τον ορισμό των κλάσεων διαχειριζόμενων αντικειμένων, των κατηγορημάτων τους, μηνυμάτων, κ.ά. Οι μέθοδοι αυτοί περιλαμβάνουν:
  - ◆ περιληψη των θεμάτων που πρέπει να αντιμετωπίζονται στους ορισμούς των κλάσεων διαχειριζόμενων αντικειμένων,
  - ◆ εργαλεία για το συμβολισμό που μπορούν να χρησιμοποιηθούν στους ορισμούς, και
  - ◆ οδηγίες για τη συνέχεια μεταξύ διαφορετικών ορισμών αντικειμένων.
- τις σχέσεις των διαχειριζόμενων αντικειμένων με το πρωτόκολλο διαχείρισης OSI, και
- την απαιτούμενη δομή των εγγράφων για τους ορισμούς.

### 5.4.4. Πρακτικά ζητήματα [STAL93]

Για ένα Σύστημα Διαχείρισης Δικτύων είναι δυνατό να καθοριστούν, όσο αναφορά τις επιδόσεις οι παρακάτω τρεις απαιτήσεις:

- (1) Το Σύστημα αυτό δεν θα πρέπει να υποβαθμίζει τη λειτουργία του δικτύου, την οποία έχει στόχο να υποστηρίξει,
- (2) Οι αποφάσεις του πρέπει να παίρνονται με μεγάλη ταχύτητα και εξίσου γρήγορα να εκτελούνται, πριν η κατάσταση του δικτύου μεταβληθεί σημαντικά

(στην τελευταία περίπτωση, ο βρόγχος μέτρηση/ενέργεια-ελέγχου μπορεί να μεταβεί σε κατάσταση αστάθειας), και τέλος

- (3) Πρέπει να παρέχει ένα πλούσιο σύνολο από υπηρεσίες προκειμένου να είναι δυνατή η πραγματοποίηση μιας μεγάλης ποικιλίας από λειτουργίες διαχείρισης δικτύου, παρέχοντας λεπτομερή παρακολούθηση και έλεγχο του δικτύου.

Όσο αναφορά την πρώτη απαίτηση ένας χρήσιμος κανόνας που έχει προταθεί στη βιβλιογραφία αναφέρει ότι: **η μέγιστη επιτρεπτή κατανάλωση χωρητικότητας από λειτουργίες διαχείρισης πρέπει να αποτελεί το 5% της συνολικής διαθέσιμης χωρητικότητας\***. Από την άλλη πλευρά, η απαίτηση της άμεσης αντίδρασης σημαίνει ότι σημαντικές ποσότητες πληροφορίας διαχείρισης πρέπει να διακινούνται μέσα από το δίκτυο. Επιπλέον και ένας μεγάλος αριθμός υπηρεσιών διαχείρισης υπονοεί ότι μεγάλες ποσότητες πληροφορίας διαχείρισης πρέπει να μεταφέρονται στο δίκτυο. Βλέπουμε δηλ. ότι οι παραπάνω απαίτησεις συγκρούονται κατά κάποιο τρόπο μεταξύ τους. Μας ενδιαφέρουν τα παραπάνω σε ότι αφορά την OSI διαχείριση.

Πιο συγκεκριμένα:

- Η διαχείριση συστημάτων OSI προσφέρει μια μεγάλη ποικιλία από λειτουργίες και υπηρεσίες. Εάν ο χρήστης εκμεταλευθεί τις υπηρεσίες αυτές, τότε οδηγούμαστε σε υλοποιήσεις μεγάλου όγκου και σημαντικό φορτίο στο δίκτυο.
- Η OSI MIB είναι μεγάλη και σύνθετη. Ένα διαχειριζόμενο αντικείμενο περιλαμβάνει στον ορισμό του κατηγορίατα, μηνύματα και ενέργειες. Επιπλέον είναι δυνατό να υπάρχουν πολλά τέτοια αντικείμενα σε μια σύνθετη δόμη η οποία περιγράφεται, άλλα σχετικά με τους πόρους και άλλα σχετικά με λειτουργίες διαχείρισης, όπως ημερολόγια, φίλτρα, κ.ά.
- Τα CMIP PDUs είναι σημαντικά μεγάλα λόγω της BER κωδικόποιησης και λόγω της χρησιμοποίησης της πλήρης στοίβας πρωτοκόλλων OSI (και των 7 επιπέδων).

Τα παραπάνω υποδεικνύουν ότι η διαχείριση OSI δεν ικανοποιεί τις παραπάνω απαίτησεις. Παρ' όλα αυτά υπάρχουν τρόποι προκειμένου να βελτιωθεί η κατάσταση αυτή.

- Μια τεχνική για τη μείωση του φορτίου που εισάγει το CMIP στο δίκτυο είναι η χρησιμοποίηση μιας ιεραρχίας από διαχειριστές. Τοπικοί διαχειριστές είναι δυνατό να χρησιμοποιούνται για τη διαχείριση τοπικών δικτύων υπολογιστών, μεταφέροντας μονάχα σημαντικές πληροφορίες ή περίληψη της πληροφορίας διαχείρισης. Δηλ. με την αναφορά σε ένα αντικείμενο ανακτούνται πολλά άλλα αντικείμενα. Στην αντίθετη περίπτωση η ανάκτηση ενός 32-bit μετρητή μπορεί να φορτώσει σημαντικά περισσότερο το δίκτυο.
- Ένας άλλος τρόπος για να ελαχιστοποιηθεί το φορτίο από το CMIP στο δίκτυο είναι ο καλός χειρισμός των υπηρεσιών που προσφέρει το πρωτόκολλο (π.χ. scoping και filtering). Με τον τρόπο αυτό, και εφόσον βέβαια η MIB είναι σωστά οργανωμένη δίνεται η δυνατότητα "ικανής" ανάκτησης της πληροφορίας διαχείρισης. Δηλ. με την αναφορά σε ένα αντικείμενο ανακτούνται πολλά άλλα αντικείμενα. Στην αντίθετη περίπτωση η ανάκτηση ενός 32-bit μετρητή μπορεί να φορτώσει σημαντικά περισσότερο το δίκτυο.

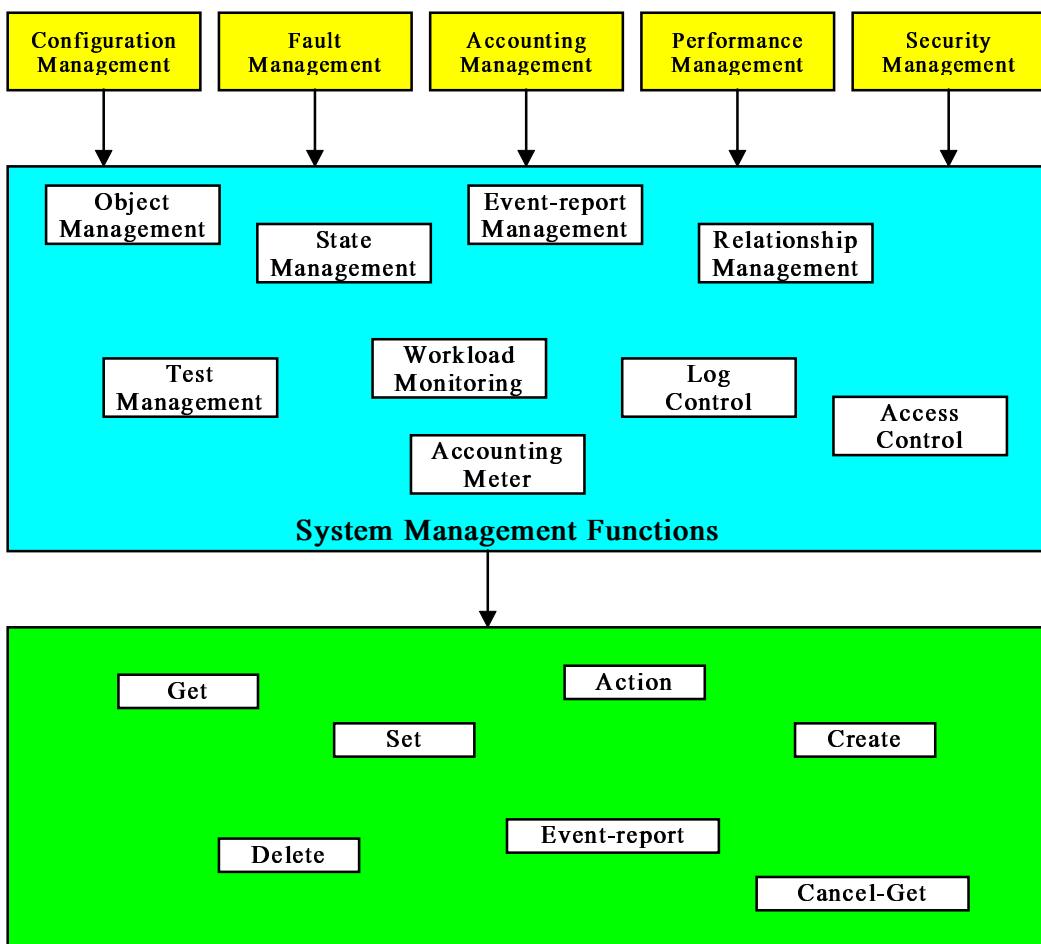
---

\* Bλ. Aronoff R., et al. "Management of Networks Based on Open Systems Interconnection (OSI) Standards: Functional Requirements and Analysis. Gaithersburg, Md.: National Institute of Standards and Technology, Special Publication 500-175, Nov. 1992.

- Τέλος η χρησιμοποίηση "thin" στοιβών πρωτοκόλλων για την υποστήριξη του πρωτοκόλλου CMIP, παρέχοντας την αυστηρά απαραίτητη λειτουργικότητα, ενώ δεν περιορίζουν τις λειτουργίες του πρωτοκόλλου περιορίζουν σημαντικά τις ανάγκες για επεξεργασία και επικοινωνία.

## 5.5. Λειτουργίες διαχείρισης συστημάτων

Οι υπηρεσίες που προσφέρει το Common Management Information Service Element είναι γενικής μορφής αφίνοντας στις εφαρμογές διαχείρισης πολλές υπευθυνότητες. Η ανάπτυξη πλήρων εφαρμογών διαχείρισης είναι δυνατή μονάχα στη περίπτωση μικρών τοπικών δικτύων με ανάλογες διαχειριστικές ανάγκες. Στην περίπτωση μεγάλων ετερογενών τηλεπικοινωνιακών δικτύων ευρείας περιοχής η πλήρης ικανοποίηση των απαιτήσεων από τη διαχείριση γίνεται προβληματική. Για το λόγο αυτό ένα σύνολο από πρότυπα έχουν εκδοθεί κάτω από τη γενική κατηγορία **λειτουργίες διαχείρισης συστημάτων (Systems Management Functions - SMFs)**. Οι λειτουργίες αυτές ορίστηκαν προκειμένου να παρέχουν την απαραίτητη λειτουργικότητα που καθορίζεται από τις πέντε περιοχές διαχείρισης (CFAPS), αποφεύγοντας την επανάληψη ήδη υλοποιημένων λειτουργιών (λόγω των γνωστών επικαλύψεων) (βλ. και παρακάτω σχήμα [STAL93]).



**Σχήμα 5.8 - Λειτουργίες διαχείρισης συστημάτων**

## 1. Λειτουργία Object Management

Το πρότυπο ISO/IEC 10164-1 | X.730 ορίζει τη λειτουργία **Object Management**. Η λειτουργία αυτή καθορίζει τον τρόπο με τον οποίο μπορεί κανείς να δημιουργεί, να καταργεί, να εξετάζει και να αλλάζει τις τιμές κατηγορημάτων διαχειριζόμενων αντικειμένων. Επίσης καθορίζει τα μηνύματα, τα οποία πρέπει να στέλνονται κατά την αλλαγή τιμών σε κατηγορήματα αντικειμένων.

Η λειτουργία διαχείρισης συστημάτων αυτή είναι θεμελιώδης και χρησιμοποιεί όλες τις υπηρεσίες του CMIS προκειμένου να προσφέρει τις δικές της υπηρεσίες.

## 2. Λειτουργία State Management

Το πρότυπο ISO/IEC 10164-2 | X.731 ορίζει τη λειτουργία διαχείρισης συστημάτων **State Management**. Η λειτουργία αυτή καθορίζει ένα μοντέλο για την αναπαράσταση της κατάστασης ενός διαχειριζόμενου αντικειμένου. Το μοντέλο αυτό επιτρέπει στο χρήστη της διαχείρισης OSI να παρακολουθεί τη κατάσταση των διαχειριζόμενων αντικειμένων στο παρελθόν και να δέχεται μηνύματα σχετικά με την αλλαγή κατάστασης διαχειριζόμενων αντικειμένων.

Η κατάσταση ενός διαχειριζόμενου αντικειμένου, σύμφωνα με το μοντέλο αυτό αναπαριστά τις στιγμιαίες τιμές της **διαθεσιμότητας (availability)**, της **κατάστασης λειτουργίας (operability)** και της **επιθυμίας του διαχειριστή (administration)** σε ότι αφορά τη κατάσταση του αντικειμένου. Στο πρότυπο ορίζονται τρία διαγράμματα καταστάσεων, τα οποία αντιστοιχούν στους τρεις κύριους παράγοντες που επηρεάζουν τη κατάσταση ενός διαχειριζόμενου αντικειμένου (βλ. και παρακάτω σχήμα).

Οι ανάλογες υπηρεσίες που προσφέρει η λειτουργία αυτή είναι:

- Η αναφορά αλλαγών στα κατηγορήματα που περιγράφουν τις καταστάσεις.
- Η ανάγνωση των κατηγορημάτων που περιγράφουν τις καταστάσεις.

## 3. Λειτουργία Relationship Management

Το πρότυπο ISO/IEC 10164-3 | X.732 παρέχει μοντέλα και αναγνωρίζει τύπους σχέσεων που είναι δυνατό να υφίστανται μεταξύ διαχειριζόμενων αντικειμένων που παριστάνουν διαφορετικά μέρη του ίδιου συστήματος. Το πρότυπο αυτό ορίζει υπηρεσίες για την εγκατάσταση, αναγνώριση, και παρακολούθηση των σχέσεων μεταξύ διαχειριζόμενων αντικειμένων. Οι υπηρεσίες αυτές μας δίνουν τη δυνατότητα να ανακαλύψουμε τον τρόπο με τον οποίο η λειτουργία ενός μέρους του συστήματος επηρεάζει ένα άλλο μέρος του συστήματος.

Γενικά μια σχέση είναι ένα σύνολο από κανόνες που περιγραφούν τον τρόπο με τον οποίο η λειτουργία ενός διαχειριζόμενου αντικειμένου επηρεάζει τη λειτουργία ενός άλλου διαχειριζόμενου αντικειμένου. Για παράδειγμα, θα μπορούσε η αποτυχία κάποιου αντικειμένου να ενεργοποιεί ένα δεύτερο αντικείμενο.

## 4. Λειτουργία Alarm Reporting

Το πρότυπο ISO/IEC 10164-4 | X.733 καθορίζει γενικά μηνύματα συναγερμών (γεγονότα), μαζί με τις παραμέτρους και τη σημασία τους. Τα μηνύματα αυτά είναι σχετικά κατά κύριο λόγο με την περιοχή διαχείρισης σφαλμάτων (Fault Management). Η πληροφορία που παρέχεται περιλαμβάνει τύπους λαθών, πιθανές αιτίες, και επίπεδα της σημαντικότητας του συναγερμού. Η λειτουργικότητα αυτή είναι απαραίτητη σε ένα περιβάλλον πολλών δικτύων με πολλαπλά ανοικτά συστήματα σε καθένα από αυτά όπου αναγκαία η εντόπιση της πηγής του σφάλματος.

Στο πρότυπο αυτό ορίζονται πέντε βασικές κατηγορίες συναγερμών:

- **Επικοινωνίας:** ο οποίος χρησιμοποιείται για την αναφορά λαθών στην επικοινωνία
- **Ποιότητας Υπηρεσίας:** ο οποίος χρησιμοποιείται για την αναφορά αποτυχίας ή πτώσης στην προσφερόμενη ποιότητα υπηρεσίας
- **Επεξεργασίας:** ο οποίος χρησιμοποιείται για την αναφορά λαθών κατά την επεξεργασία
- **Υλικού:** ο οποίος χρησιμοποιείται για την αναφορά σφαλμάτων στο υλικό
- **Περιβάλλοντος:** ο οποίος χρησιμοποιείται για την αναφορά προβλημάτων στο χώρο όπου βρίσκεται αποθηκευμένο το υλικό

Μερικά από τα κατηγορήματα για τους παραπάνω συναγερμούς είναι τα εξής: **probableCause** (loss of signal, framing error, local transmission error, call-establishment error, degraded signal, κ.ά.), **specificProblems**, **perceivedSeverity** (critical, major, minor, warning, κ.ά.), **backedUpStatus**, **backUpObject**, **trendIndication** (more severe, no change, less severe), **thresholdInfo**, κ.ά.

## 5. Λειτουργία Event Report Management

Το πρότυπο ISO/IEC 10164-5 | X.734 παρέχει ένα μοντέλο για τον έλεγχο της αναφοράς γεγονότων. Η λειτουργία **Event-Report-Management** επιτρέπει σε ένα διαχειριστή να ελέγχει τη μετάδοση αναφορών γεγονότων από τα διαχειριζόμενα αντικείμενα ανεξάρτητα του τι συμβαίνει στα αντικείμενα αυτά. Ο έλεγχος αυτός επιτυγχάνεται μέσω του ορισμού επιπρόσθετων αντικειμένων, τα οποία ονομάζονται **event-forwarding-discriminators**. Στην ουσία ορίζεται ένα φίλτρο, το οποίο καθορίζει ποια γεγονότα περνούν και ποια όχι. Το πότε θα είναι ενεργό το φίλτρο αυτό, όπως επίσης το για ποιους προορισμούς θα είναι ενεργό το φίλτρο αυτό μπορεί επίσης να καθοριστεί από τη λειτουργία αναφοράς γεγονότων.

Η λειτουργία αναφοράς γεγονότων παρέχει τις παρακάτω υπηρεσίες:

- Δημιουργία discriminator
- Κατάργηση discriminator
- Μεταβολή των κατηγορημάτων ενός discriminator
- Παύση της δραστηριότητας ενός discriminator

- Επανεργοποίηση της δραστηριότητας ενός discriminator

Ολοκληρώνουμε λέγοντας ότι η λειτουργία αυτή είναι ιδιαίτερα σημαντική σε δίκτυα ευρείας περιοχής (WANs), είτε σε άλλα περιβάλλοντα όπου υπάρχει περιορισμένη διαθέσιμη χωρητικότητα.

## 6. Λειτουργία Log Control

Το πρότυπο ISO/IEC 10164-6 | X.735 καθορίζει ένα μοντέλο για τον έλεγχο ημερολογίων γεγονότων. Ένα φίλτρο μπορεί να οριστεί το οποίο θα καθορίζει τα γεγονότα που έχουν θέση σε ένα ημερολόγιο. Όπως και η προηγούμενη λειτουργία διαχείρισης συστημάτων, η λειτουργία **log-control** επιτρέπει στο διαχειριστή να καθορίζει σχέδια αποθήκευσης γεγονότων καθώς και σχετικά κριτήρια.

Υπηρεσίες τις οποίες προσφέρει η λειτουργία αυτή είναι:

- Δημιουργία ημερολογίου
- Κατάργηση ημερολογίου
- Μεταβολή κατηγορημάτων ημερολογίου
- Παύση της δραστηριότητας ημερολογίου
- Διαγραφή εγγραφών ημερολογίου
- Ανάκτηση εγγραφών ημερολογίου
- Επανεργοποίηση ημερολογίου

## 7. Λειτουργία Security Alarm Reporting

Το πρότυπο ISO/IEC 10164-7 | X.736 ορίζει ένα μοντέλο για την αναφορά γεγονότων σχετικών με την ασφάλεια, και τις δυσλειτουργίες των υπηρεσιών και των μηχανισμών ασφάλειας. Η βασική απαίτηση από τη λειτουργία αυτή είναι να ικανοποιήσει την ανάγκη ειδοποίησης των σχετικών διαχειριστικών εφαρμογών κατά την αναγνώριση ενός γεγονότος που δείχνει επίθεση ή πιθανή επίθεση στο σύστημα ασφάλειας.

## 8. Λειτουργία Security Audit Trail

Το πρότυπο ISO/IEC 10164-8 | X.740 καθορίζει τα είδη των αναφορών γεγονότων που πρέπει να περιέχονται σε ένα ημερολόγιο το οποίο χρησιμοποιείται για την αξιολόγηση της ασφάλειας ενός ανοικτού συστήματος και των επιδόσεων των χρησιμοποιούμενων μηχανισμών ασφάλειας. **Security-audit-trails** μπορούν να χρησιμοποιηθούν για την εύρεση επιθέσεων κατά της ασφάλειας, οι οποίες δεν αναγνωρίστηκαν σε πραγματικό χρόνο.

Τύποι γεγονότων, οι οποίοι μπορεί να συμπεριληφθούν σε ένα τέτοιο ημερολόγιο είναι:

- Συνδέσεις

- Αποσυνδέσεις
- Χρησιμοποίηση μηχανισμού ασφάλειας
- Λειτουργίες διαχείρισης
- Μέτρησεις χρήσης

## 9. Λειτουργία Access Control Management

Το πρότυπο ISO/IEC 10164-9 | X.741 καθορίζει ένα μοντέλο για τον έλεγχο της πρόσβασης στην πληροφορία και τις λειτουργίες που αφορούν τη διαχείριση. Καθορίζει διαχειριζόμενα αντικείμενα τα οποία έχουν τη δυνατότητα απόδοσης ή απόρριψης της πρόσβασης σύμφωνα με κάποια πολιτική ελέγχου πρόσβασης.

## 10. Λειτουργία Accounting - Meter

Το πρότυπο ISO/IEC 10164-10 | X.742 καθορίζει ένα μοντέλο για το λογαριασμό της χρήσης των πόρων του συστήματος και ένα μηχανισμό για την επιβολή ορίων στη χρήση αυτή. Το πρότυπο ορίζει μετρητές της χρήσης, ημερολόγια, και καθορίζει υπηρεσίες για την ανάκτηση, αναφορά και εγγραφή δεδομένων που αφορούν χρησιμοποίηση πόρων.

## 11. Λειτουργία Workload Monitoring

Το πρότυπο ISO/IEC 10164-11 | X.739 καθορίζει ένα μοντέλο για την παρακολούθηση των κατηγορημάτων διαχειριζόμενων αντικειμένων. Ορίζει διαχειριζόμενα αντικείμενα τα οποία μπορούν να αναφέρουν γεγονότα βασισμένα σε τιμές counters και gauges και που αντανακλούν τις επιδόσεις του συστήματος.

## 12. Λειτουργία Test Management

Το πρότυπο ISO/IEC 10164-12 καθορίζει ένα μοντέλο για τη διαχείριση διαδικασιών confidence και diagnostic-tests.

## 13. Λειτουργία Summarization

Το πρότυπο ISO/IEC 10164-13 ορίζει ένα μοντέλο και ανάλογες κλάσεις διαχειριζόμενων αντικειμένων που χρησιμοποιούνται για την εφαρμογή περιήληψης και στατιστικής ανάλυσης της πληροφορίας διαχείρισης. Η περιήληψη των τιμών κατηγορημάτων περιλαμβάνει συγκεκριμένα στιγμιότυπα του αντικειμένου στη διάρκεια του χρόνου (time averages) και ένα σύνολο από στιγμιότυπα αντικειμένων σε συγκεκριμένους χρόνους (ensemble averages). Οι υπηρεσίες της λειτουργίας διαχείρισης συστημάτων αυτής περιλαμβάνουν το καθορισμό των διαχειριζόμενων αντικειμένων που θα συμπεριληφθούν στις αναφορές, το σχεδιασμό του χρόνου των παρατηρήσεων, και το σχεδιασμό του χρόνου των σχετικών αναφορών.

Τα περισσότερα από τα αντικείμενα που ορίζονται στη λειτουργία αυτή είναι παιδιά του αντικειμένου scanner.

```

scanner      MANAGED OBJECT CLASS
DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2: top;"*
CHARACTERIZED BY scannerPackage PACKAGE
BEHAVIOUR
scannerBehaviour BEHAVIOUR
    DEFINED AS "see 8.1.1.3";;
ATTRIBUTES
scaneerId GET,
"Rec. X.721 | ISO/IEC 10164-2: administrativeState GET_REPLACE,"
granularityPeriod GET_REPLACE
"Rec. X.721 | ISO/IEC 10164-2: operationalState GET;;"*
CONDITIONAL PACKAGES
.
.
.
REGISTERED AS {summarizationMObjectClass ???};
```

## 5.6. Το όφελος από μια τυποποιημένη στοίβα πρωτοκόλλων

Η χρήση προτύπων για την επίτευξη μιας ολοκληρωμένης διαχείρισης δικτύων προσφέρει πολλά πλεονεκτήματα, ιδιαίτερα όταν πρόκειται για αναγνωρισμένα διεθνή πρότυπα. Μερικά από αυτά τα πλεονεκτήματα είναι τα παρακάτω:

- Η χρήση κοινών επικοινωνιακών μέσων μεταξύ των εφαρμογών διαχείρισης δικτύων και των διαφόρων άλλων OSI υπηρεσιών. Σύμφωνα με τα διεθνή πρότυπα, τα μηνύματα που αφορούν τη διαχείριση μπορούν να χρησιμοποιήσουν τις υπηρεσίες δικτύωσης, που προσφέρονται από γνωστά πρωτόκολλα, όπως αυτά των 802.X τοπικών δικτύων, του X.25 και του ISDN. Με τον τρόπο αυτό επιτυγχάνεται πέρα από την ευκολία, εξοικονόμηση χρόνου και χρήματος μια και η υλοποίηση των πρωτοκόλλων αυτών έχει ήδη επιτευχθεί, όπως επίσης και αυξημένη αξιοπιστία μια και τα συγκεκριμένα πρωτόκολλα έχουν ήδη χρησιμοποιηθεί σε μεγάλη κλίμακα.
- Η χρήση συμβάσεων στην ονομασία των διαχειριζόμενων αντικειμένων, καθώς και η χρησιμοποίηση τυποποιημένων διαχειριστικών μηνυμάτων επιτρέπουν την επίτευξη της ολοκληρωμένης διαχείρισης σε ένα ετερογενές περιβάλλον. Για παράδειγμα, η τυποποίηση των τιμών που δείχνουν το επίπεδο σημαντικότητας κάποιου συναγερμού, επιτρέπει σε ένα σύστημα διαχείρισης να ελέγχει τους συναγερμούς που στέλνει κάθε διαχειριζόμενο αντικείμενο.
- Οι διάφορες υπηρεσίες που προσφέρονται στα διάφορα στρώματα του OSI μοντέλου αναφοράς πρωτοκόλλων, διαρκώς επεκτείνονται και βελτιώνονται στις επιτροπές διεθνούς τυποποίησης από ειδικούς. Οι αναβαθμίσεις αυτές θα χρησιμοποιούνται από εφαρμογές διαχείρισης δικτύων, όσο αυτές στηρίζονται σε διεθνή πρότυπα.
- Επίσης, θα είναι δυνατή εξοικονόμηση χρόνου και χρήματος και στην ανάπτυξη των διαχειριστικών εφαρμογών, από την στιγμή που θα χρησιμοποιούν μια κοινή διαχειριστική πλατφόρμα. Για παράδειγμα οι parsers των μηνυμάτων θα μπορούν να χρησιμοποιηθούν σε πολλαπλές εφαρμογές. Επίσης πιθανές προεκτάσεις στις εφαρμογές αυτές θα πραγματοποιούνται με την προσθήκη πεδίων στα μηνύματα που αφορούν τη διαχείριση, και προσθήκη τιμών στις μεταβλητές των πεδίων αυτών.

- Τέλος, η χρησιμοποίηση τυποποιημένη λειτουργικότητα στα διαχειριζόμενα αντικείμενα θα δώσει ώθηση και στην χρήση της τεχνολογίας των ευφυών συστημάτων, τα οποία θα έχουν δυνατότητες αυτόματης ανάλυσης λαθών, αρχικοποίησης δοκιμών και επεξεργασίας συναγερμών.

## 5.7. Σύγκριση μεταξύ των SNMP και CMIP

Σε ένα πραγματικά ολοκληρωμένο περιβάλλον δικτύων, υπάρχει χώρος τόσο για το SNMP και για το CMIP πρωτόκολλο, όπως βέβαια και για άλλα ανάλογα πρωτόκολλα διαχείρισης. Το CMIP μπορεί να ελέγχει τα packet switches και τα LAN/WAN gateways του δικτύου, το SNMP μπορεί να ελέγχει όλα τα τοπικά δίκτυα, το NETVIEW το SNA κομμάτι του δικτύου κ.ο.κ. Βέβαια, ακόμα καλύτερη λύση είναι όλα τα παραπάνω να έχουν ενοποιηθεί σε ένα μοναδικό διαχειριστικό σύστημα.

Όση κι αν είναι σήμερα η έκταση της χρησιμότητάς του, δεν πρέπει να ξεχνάμε ότι το SNMP ουσιαστικά σχεδιάστηκε για την διαχείριση μεγάλων δικτύων. Αυτό σημαίνει ότι αρχικά τουλάχιστον δεν προοριζόταν για την διαχείριση υπολογιστικών συστημάτων, ή για υψηλού επιπέδου διαχείριση διάρθρωσης, και οι δυνατότητες του τελικά περιορίζονται στην παρακολούθηση και απομόνωση λαθών. Το χαμηλό επίπεδο ασφάλειας που προσφέρει είναι σίγουρα ένα από τα μεγαλύτερα μειονεχτήματά του, και η σύντομη λύση αυτού του προβλήματος, σίγουρα θα δώσει αρκετά χρόνια ζωής στο SNMP, αφού από ένα απλό πρωτόκολλο παρακολούθησης, θα το αναβαθμίσει σε ένα πρωτόκολλο ουσιαστικής διαχείρισης.

Πάντως από κάθε πλευρά, το κλειδί της επιτυχίας του SNMP είναι η ευκολία της υλοποίησής του, από την στιγμή που ο ορισμός του είναι τόσο απλός. Ευκολία που είναι ιδιαίτερα φανερή στις πολλές υπάρχουσες υλοποίησεις agents, μια και έχει μικρές απαιτήσεις τόσο σε μνήμη, όσο και σε χρόνους επεξεργασίας. Η επιτυχία επίσης των TCP/IP πρωτοκόλλων και η χρησιμοποίησή τους σε κάθε είδους δικτύου αποτελεί άλλο ένα πλεονέκτημα υπέρ του SNMP.

Η σχεδίαση του SNMP είχε σαν πρώτιστο στόχο την ελαχιστοποίηση της πολυπλοκότητας της MIB, του πρωτοκόλλου ανάκτησης της πληροφορίας διαχείρισης, και της συνολικής υλοποίησης ενός SNMP agent. Η απλότητα αυτή βοήθησε στη δημιουργία ανοικτών υλοποιήσεων (η ερμηνεία και η συμμόρφωση με τα πρότυπα ήταν πολύ εύκολη), και στην αποδοχή του SNMP από την αγορά. Από την άλλη πλευρά η απλότητα αυτή οδήγησε σε διάφορους περιορισμούς. Περιορισμοί στην περίπτωση του SNMP είναι οι παρακάτω [YEMI94]:

**To μοντέλο της πληροφορίας διαχείρισης.** Τα διαχειριζόμενα αντικείμενα σε μια SNMP MIB ορίζονται κατά τη χρονική στιγμή σχεδιασμού της MIB. Αυτό σημαίνει ότι κάποια εφαρμογή διαχείρισης μπορεί σε κάποια περίπτωση να μην έχει πρόσβαση σε κάποιο αντικείμενο, το οποίο την ενδιαφέρει. Μοναδική δυνατότητα αποτελεί, για μια εφαρμογή διαχείρισης, η επεξεργασία των υπαρχόντων δεδομένων, προκειμένου να βγουν πρόσθετα συμπεράσματα για τη κατάσταση του δικτύου. Αυτό όμως έχει το μειονέκτημα της μεταφοράς μέσα από το δίκτυο μεγάλων ποσοτήτων ακατέργαστων δεδομένων (τα οποία συχνά δεν έχουν μεγάλη σημασία).

**Η υλοποίηση διαχειριστικών ενεργειών.** Αυτή πρέπει να επιτυγχάνεται σαν αποτέλεσμα ενός set\_request μηνύματος, όπως είπαμε και παραπάνω. Η τεχνική αυτή δουλεύει καλά για απλές εντολές (π.χ. reboot system). Στη περίπτωση συνθετότερων διαδικασιών, οι οποίες έχουν κάποια ορίσματα και πρέπει να επιστρέψουν κάποια τιμή, ένα set\_request δεν είναι αρκετό. Μια τέτοια διαδικασία μπορεί να υλοποιηθεί με δύο διαδοχικά set\_requests (πέρασμα παραμέτρων και κλήση της διαδικασίας) και με ένα

`get_request` (επιστροφή τιμής). Όσο και αν φαίνεται απλό σε ένα σύνθετο περιβάλλον με πολλούς διαχειριστές, όπου πολλά προβλήματα μπορεί να δημιουργηθούν, υπάρχει η ανάγκη συχρονισμού των παραπάνω μηνύματων.

**Μη υποστήριξη μηχανισμών μαζικής ανάκτησης δεδομένων.** Η ανάκτηση ενός πίνακα με το πρωτόκολλο SNMP πρέπει να γίνει γραμμή-γραμμή με διαδοχικά `get_next_requests`. Το SNMPv2 λύνει κάπως το πρόβλημα με το `get_bulk_request`.

**Μη υποστήριξη μηχανισμών φιλτραρίσματος των δεδομένων που ανακτούνται.** Το μοντέλο πρωτοκόλλου του SNMP αναγκάζει το διαχειριστή να ανακτήσει μεγάλες ποσότητες δεδομένων (συχνά όχι χρήσιμες) και να τις επεξεργαστεί τοπικά, προκειμένου να ανακαλύψει αν κάτι πάει στραβά στο δίκτυο.

Τελειώνοντας, επαναλαμβάνουμε ότι οι παραπάνω περιορισμοί είναι άμεσα αποτελέσματα της απλότητας του SNMP, η οποία είναι και το κυριότερο πλεονέκτημά του.

Συγκρίνοντας τα δύο πρωτόκολλα διαπιστώνουμε ότι τα πρότυπα που τα περιγράφουν δεν αναδεικνύουν ουσιαστικά πλεονεχτήματα ή μειονεχτήματα για κάποιο από τα δύο. Είναι οι υλοποιήσεις τους, εκείνες που θα αναδείξουν το καλύτερο για κάθε περίπτωση καθώς και οι λειτουργίες που θα απαιτήσει κάθε χρήστης.

Πραγματικά, όσο αναφορά την απόδοση (performance) υπάρχουν υλοποιήσεις του CMIP (αλλά και του SNMP) που χρησιμοποιούν πολλή μνήμη και απαιτούν μεγάλους χρόνους επεξεργασίας. Αν και το CMIP φαίνεται να έχει κάποιο προβάδισμα σε κατανάλωση μνήμης, λόγω των πολλών πρωτοκόλλων πάνω από τα οποία υλοποιείται (ACSE, ROSE, Presentation).

Όσο αναφορά την αξιοπιστία (reliability) και αυτή εξαρτάται αυστηρά από την συγκεκριμένη υλοποίηση του κάθε πρωτοκόλλου. Η ουσιαστική διαφορά μεταξύ των δύο βρίσκεται σίγουρα στο μοντέλο που χρησιμοποιούν για να παραστήσουν την πληροφορία. Εδώ το object-oriented μοντέλο του CMIP έχει από κάθε άποψη το προβάδισμα, έχοντας υπέρ του όλα τα πλεονεχτήματα του object-orientation. Η αρχιτεκτονική του CMIP είναι πιο ανοικτή σε επεκτάσεις, από την άποψη ότι βλέπει πιο αφηρημένα τα αντικείμενα που διαχειρίζεται, και έτσι μπορεί πιο εύκολα να διαχειριστεί νέα αντικείμενα, χωρίς να είναι απαραίτητη η αλλαγή κάποιου κώδικα. Το CMIP λοιπόν έχει ένα ένα προβάδισμα στην επεκτασιμότητα (extensibility).

Γενικότερα, μπορεί να πει κανείς, ότι το OSI μοντέλο διαχείρισης συστημάτων προσπαθεί να δώσει ένα πλήρες πλαίσιο διαχείρισης οσοδήποτε πολύπλοκων συστημάτων. Πιο συγκεκριμένα, αξίζει να σημειωθούν τα παρακάτω [YEMI94]:

**(α) Το μοντέλο για τη πληροφορία διαχείρισης.** Η OSI διαχείριση συστημάτων προσφέρει ένα εκτεταμένο πλαίσιο μοντελοποίησης της πληροφορίας διαχείρισης, το οποίο στηρίζεται κατά κύριο λόγο σε μια αντικειμενοστραφή προσέγγιση. Πλεονεκτήματα της προσέγγισης αυτής, που ταυτόχρονα αποτελούν και πλεονεκτήματα του OSI μοντέλου της πληροφορίας διαχείρισης είναι τα παρακάτω:

(α1) *Αφαίρεση και κληρονομικότητα.*

(α2) *Μη στατική OSI MIB.*

(α3) *Αναπαράσταση σχέσεων μεταξύ διαχειριζόμενων αντικειμένων.*

**(β) Το μοντέλο για τη πρόσβαση στη πληροφορία διαχείρισης.** Εδώ το OSI μοντέλο εισάγει δύο ουσιαστικές λειτουργίες που δεν προσφέρει το SNMP. Αυτές είναι: (α) η μαζική ανάκτηση πληροφορίας, και (β) η επιλεκτική ανάκτηση πληροφορίας. Στο

SNMPv2, όπως είπαμε και παραπάνω δίνεται η δυνατότητα ανάκτησης πληροφορίας, τουλάχιστον όστις επιτρέπει το πεδίο δεδομένων ενός UDP datagram. Από την άλλη πλευρά ούτε το SNMP, ούτε το SNMPv2 προσφέρουν κάποια δυνατότητα για φιλτράρισμα των δεδομένων που θα ανακτηθούν. Αντίθετα το CMIS προσφέρει ένα ισχυρό σύνολο εργαλείων (scoping, filtering, synchronization) προκειμένου ο διαχειριστής να μπορεί να επιλέξει το αντικείμενο ή τα αντικείμενα στα οποία θα εφαρμοστεί η διαχειριστική λειτουργία.

**(γ) Το μοντέλο για τη μεταφορά της πληροφορίας διαχείρισης.** Το χαρακτηριστικό του μοντέλου αυτού είναι η χρήση πρωτοκόλλων επικοινωνίας με σύνδεση για την OSI διαχείριση, και χωρίς σύνδεση για την TCP/IP διαχείριση. Η OSI διαχείριση στηρίζεται επιπλέον και σε πρωτόκολλα του επιπέδου εφαρμογής (ACSE, κ.ά.), ακόμα και στην περίπτωση που χρησιμοποιεί κάποιο αξιόπιστο πρωτόκολλο μεταφοράς με σύνδεση (π.χ. TCP). Τα μειονεκτήματα της χρησιμοποίησης επικοινωνίας με σύνδεση δείχνουν να είναι περισσότερα από τα αντίστοιχα πλεονεκτήματα. Οι λειτουργίες διαχείρισης έχουν νόημα κυρίως όταν το δίκτυο βρίσκεται σε κάποια άσχημη κατάσταση (υπερφορτισμένο), και στη περίπτωση αυτή οι διαχειριστικές οντότητες θα πρέπει ξοδεύουν χρόνο και πόρους στην αρχικοποίηση και επαναεγκατάσταση συνδέσεων διαχείρισης. Αντίθετα ένα μοντέλο επικοινωνίας που στηρίζεται σε πρωτόκολλα χωρίς σύνδεση, έχει μεγαλύτερες πιθανότητες να περάσει μέσα από το δίκτυο σε μια ιδιαίτερα άσχημη κατάσταση.

Τελειώνοντας θα πρέπει να προσθέσουμε ότι τα παραπάνω αποτελούν έναν πρόλογο σε μια σύγκριση των δύο πρωτοκόλλων. Θα πρέπει να υπάρξει κάποια συνέχεια, όπου θα δίνεται απάντηση σε ερωτήματα όπως τα παρακάτω:

- α) Τι περιέχουν στα MIBs τους SNMP-based προιόντα και τι CMIP-based προιόντα;
- β) Πόσο γρήγορα μπορεί να γίνει η πρόσβαση σ' αυτές τις μεταβλητές;
- γ) Πόσο γρήγορα μπορούν να βγουν κάποια στατιστικά στοιχεία;
- δ) Τέλος, μπορεί το SNMP με τα λιγότερα στοιχεία υπηρεσίας να προσφέρει ότι και το CMIP; Γίνονται αυτές οι επεξεργασίες στον ίδιο χρόνο και με το ίδιο φορτίο;

Ένα σοβαρό πλεονέχτημα υπέρ του SNMP είναι η ύπαρξη πολλών έτοιμων προιόντων βασισμένων σ' αυτό. Υπάρχουν μάλιστα υλοποιήσεις τις οποίες μπορεί να προμηθευτεί κανείς χωρίς κανένα κόστος (public domain implementations). Οι υλοποιήσεις αυτές είναι:

- α) το SNMP του Carnegie-Mellon University (CMU).
- β) το SNMP development kit του Massachusetts Institute of Technology (MIT).
- γ) το πακέτο 4BSD/ISODE SNMP.

Οι παραπάνω υλοποιήσεις αναφέρονται σε agents του SNMP που μπορούν να λειτουργήσουν σε ρόλο είτε εξυπηρετητή, είτε πελάτη και να χρησιμοποιηθούν λοιπόν σαν βάση για την ανάπτυξη διαχειριστικών εφαρμογών.

Για το πακέτο 4BSD/ISODE μάλιστα υπάρχουν και public domain υλοποιήσεις των CMIP/CMOT πρωτοκόλλων, για τα οποία βέβαια υπάρχουν και άλλες υπάρχουσες υλοποιήσεις.

Μια λίστα με όλα τα προιόντα, που είναι βασισμένα στο SNMP μπορεί να βρεθεί στο RFC 1147, Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices.

Όσο αναφορά τώρα τα ολοκληρωμένα συστήματα διαχείρισης δικτύων, τα πρώτα προιόντα παρουσιάστηκαν στο τέλος του 1988 από σχετικά μικρές εταιρίες στα TCP/IP προιόντα όπως: το NetCentral της Cisco Systems Inc. (Menlo Park, Calif.), το ACS 4800 NMS της Advanced Computer Communications (Santa Barbara, Calif.), το Overview NMS της Proteon Inc. (Westborough, Mass.), το SNMP Network Management Software της Wellfleet Communications Inc. (Bedford, Mass.). Παράλληλα όλες οι μεγάλες εταιρίες, όπως η IBM (Armonk, N.Y.), η Digital Equipment Corp. (Maynard, Mass.), η Hewlett-Packard Co. (Palo Alto, Calif.), όπως και η Sun Microsystems Inc. (Mountain View, Calif.) ανακοίνωσαν υποστήριξη του SNMP στα προιόντα τους, και μερικές από αυτές και του CMIP.

## 5.8. Το πρωτόκολλο CMOT

Στην παράγραφο αυτή θα προσθέσουμε κάποια στοιχεία όσο αναφορά την προσπάθεια που έκανε η IAB σχετικά με το ISO-CMIP. Όπως είπαμε και παραπάνω, στόχος της IAB ήταν η διαχείριση των δικτύων που βασίζονταν σε TCP/IP πρωτόκολλα. Πέρα λοιπόν από την εκκίνηση που δόθηκε για την υλοποίηση μιας βραχυπρόθεσμης λύσης (SNMP), μια δεύτερη προσπάθεια ξεκίνησε για την υλοποίηση μιας μακροπρόθεσμης λύσης για την διαχείριση TCP/IP ετερογενών δικτύων που να στηρίζεται στην OSI προσέγγιση "OSI Internet Management".

Προιόν αυτής της προσπάθειας ήταν το **CMOT (Common Management Information Services and Protocol over TCP/IP)**, μια υλοποίηση του CMIP πάνω από υπάρχοντα TCP/IP περιβάλλοντα, αντί για την κλασσική OSI TP4 προσέγγιση.

Η αρχιτεκτονική του CMOT έχει ως εξής: Οι υπηρεσίες που προσφέρονται στις εφαρμογές ορίζονται από το CMIS τον ίδιο ορισμό υπηρεσιών μ' αυτό του OSI CMIP πρωτοκόλλου. Το επίπεδο εφαρμογής παραμένει το ίδιο και περιέχει τα CMISE, ACSE, ROSE πρωτόκολλα, καθώς και τις αντίστοιχες υπηρεσίες που αυτά προσφέρουν. Τα στρώματα μεταφοράς και δικτύου είναι τα TCP/UDP (αντίθετα με το SNMP που χρησιμοποιεί μόνο το UDP) και IP αντίστοιχα, ενώ στο επίπεδο παρουσίασης υπάρχει το LPP (Lightweight Presentation Protocol), όπως ορίζεται στο RFC1085, και το οποίο προσφέρει κάποιους μηχανισμούς για να υποστηριχθούν υπηρεσίες εφαρμογών OSI πάνω από TCP/IP περιβάλλοντα. Το ουσιαστικό πάντως είναι ότι τα CMIP και CMOT είναι τα ίδια πρωτόκολλα.

Σημειώνουμε ότι η προσπάθεια αυτή δεν είχε κάποιο ιδιαίτερα θετικό αποτέλεσμα. Πολλοί κατασκευαστές προϊόντων διαχείρισης δικτύων δεν θέλησαν να ξοδέψουν χρόνο και πόρους υλοποιώντας μια interim λύση. Η βραχυπρόθεσμη λύση του SNMP αποδείχθηκε πολύ πιο ιδανική.

## 5.9. Ασκήσεις

- [1]. Σε ποια από τις πέντε (5) κατηγορίες λειτουργιών διαχείρισης ISO/OSI υπάγονται τα ακόλουθα:
  - α) Έκδοση δελτίου βλάβης,

- β) παρακολούθηση δικτύου με γραφικά,  
γ) ενημέρωση αποθήκης για ανταλλακτικά στοιχείων δικτύου,  
δ) μέτρηση ρυθμαπόδοσης (throughput) γραμμών, και  
ε) εντοπισμός βλάβης και ON-LINE αντιμετώπισή της (π.χ. εναλλακτική δρομολόγηση).
- [2]. Θεωρήστε το παρακάτω σενάριο: ένας CMIP manager δέχεται διάφορα events-notifications με ένα ρυθμό Λ και χρειάζεται για να τα επεξεργαστεί χρόνο 50 ms. Αν τα διαχειριζόμενα αντικείμενα δημιουργούν ένα event-notification κάθε 15 min, ποιος είναι ο μεγαλύτερος αριθμός αντικειμένων που μπορεί να διαχειριστεί ο CMIP-based manager; Με ποιο τρόπο μπορεί το παραπάνω διαχειριστικό σύστημα να μειώσει το φορτίο στο δίκτυο; Ποια είναι τα συμπεράσματά σας;
- [3]. Ποια είναι η απόψη σας για τις πέντε λειτουργικές περιοχές στις οποίες χωρίζεται η OSI διαχείριση; Είναι οι καταλληλότερες; Είναι πλήρεις; Μήπως υπάρχουν επικαλύψεις μεταξύ τους;
- [4]. Προτείνετε τεχνικές διαχείρισης κάποιου κόμβου ο οποίος δεν υλοποιεί και τα επτά (7) επίπεδα του OSI μοντέλου αναφοράς πρωτοκόλλων (π.χ. κάποιος ενδιάμεσος κόμβος που υλοποιεί τα τρία πρώτα επίπεδα).
- [5]. Καταγράψτε ανάγκες του χρήστη, οι οποίες οφείλουν να ικανοποιηθούν μέσα από τη διαχείριση ενός δικτύου.
- [6]. Πιστεύετε ότι μια διαχειριστική εφαρμογή χρειάζεται μια αξιόπιστη υπηρεσία μεταφοράς από άκρη σε άκρη ή όχι; Ποια είναι τα επιχειρήματά σας;
- [7]. Αναφέρετε ιδιαίτερες λειτουργίες τις οποίες θα μπορούσαμε να κατατάξουμε στις ευρύτερες λειτουργικές περιοχές της OSI διαχείρισης:  
α) διαχείριση διάρθρωσης,  
β) διαχείριση σφαλμάτων,  
γ) λογιστική διαχείριση,  
δ) διαχείριση επιδόσεων,  
ε) διαχείριση ασφάλειας.
- [8]. Ποιοι είναι πιθανοί λόγοι για το χωρισμό ενός δικτύου σε ιδιαίτερες διαχειριστικές περιοχές αρμοδιότητας (management domains); Ποιες άλλες

τεχνικές θα επέκτειναν τη λειτουργικότητα ενός τέτοιου δικτύου; Διευκρινίστε με ένα σχήμα τις παραπάνω έννοιες.

- [9]. Στην πράξη πώς θα μπορούσαν να παρασταθούν σε κάποιο διαχειριστικό σύστημα οι διαφορετικές διαχειριστικές περιοχές αρμοδιότητας (management domains); Τι χαρακτηριστικά πρέπει να περιέχει η παράσταση αυτή;
- [10]. Πώς καταλαβαίνετε τον όρο διαχείριση συστημάτων (systems management), τον οποίο εισάγει η OSI διαχείριση; Ποιες πιστεύετε είναι οι ελάχιστες απαιτήσεις ώστε να είναι δυνατή η διαχείριση συστημάτων;

## 5.10. Βιβλιογραφία

- [ALAR92] Information processing systems - Open System Interconnection - *Systems Management: Alarm reporting function* - International Organization for Standardization - International Standard 10164-4 - December 1992.
- [ASN187] Information processing systems - Open System Interconnection - *Specification of Abstract Syntax Notation One (ASN.1)* - International Organization for Standardization - International Standard 8824 - December 1987.
- [BER\_87] Information processing systems - Open System Interconnection - *Specification of Basic Encoding Rules for Abstract Notation One (ASN.1)* - International Organization for Standardization - International Standard 8825 - December 1987.
- [CASS89] Lillian N. Cassel, Graig Partridge, and Jil Westcott, "Network Management Architectures and Protocols: Problems and Approaches", IEEE Journal On Selected Areas In Communicationns, Vol. 7, No. 7, September 1989.
- [CMIP91] Information processing systems - Open System Interconnection - *Common Management Information Protocol (CMIP)* - International Organization for Standardization - International Standard 9596 - June 1991.
- [CMIS91] Information technology - Open System Interconnection - *Common Management Information Service (CMIS)* - International Organization for Standardization - International Standard 9595 - April 1991.
- [DEMI92] Information processing systems - Open System Interconnection - *Structure of management information: Definition of management information* - International Organization for Standardization - International Standard 10165-2 - October 1992.
- [EMBR90] Jock Embry, Peter Manson, Dave Milham, "An Open Network Management Architecture: OSI/NM Forum Architecture and Concepts", IEEE Network Magazine, July 1990.

- [FRAID91] Fraidoon Mazda, "Convergence or Collision: SNMP and CMIP (Part 1)", Data Communications, September 1991.
- [HERM90] James Herman, "Enterprise Management Vendors Shoot It Out", Data Communications International, November 1990.
- [LOAD92] Information processing systems - Open System Interconnection - *Systems Management: Workload monitoring function* - International Organization for Standardization - Draft International Standard 10164-11 - November 1992.
- [MODI91] N. Modiri, "An Implementation of the Common Network Management Information Service Element Interfaces", IEEE Communications Magazine, July 1991.
- [OSIM89] Information processing systems - Open System Interconnection - *OSI Management Framework* - International Organization for Standardization - International Standard 7498/4 - April 1989.
- [PRES90] Randy Presuhn, "Considering CMIP", Data Communications, March 21, 1990.
- [SECU92] Information processing systems - Open System Interconnection - *Systems Management: Security alarm reporting function* - International Organization for Standardization - International Standard 10164-7 - May 1992.
- [SMGO92] Information processing systems - Open System Interconnection - *Systems Management overview* - International Organization for Standardization - International Standard 10040 - November 1992.
- [STAL93] Stallings, W. SNMP, SMMPv2, & CMIP: The Practical Guide to Network Management Standards, Addison-Wesley Publishing Company, Incorporated, 1993
- [YEMI94] Yechiam Yemini, "A Critical Survey of Network Management Protocol Standards," Telecommunications Network Management into the 21st Century: Techniques, Standards, Technologies and Applications, ed. S. Aidarous and T. Plevyak, IEEE Press, 1994.

## Κεφάλαιο 6

### 6. Διαχείριση Χαμηλών Επιπέδων

#### Περιεχόμενα του Κεφαλαίου 6

- 6.0. Εισαγωγή
- 6.1. Διαχείριση φυσικού επιπέδου και επιπέδου MAC
  - 6.1.1. Συστήματα διαχείρισης γραμμών (Cable Management Systems)
  - 6.1.2. Μηχανές διάγνωσης προβλημάτων (Diagnostic Devices)
  - 6.1.3. Εφαρμογές των παραπάνω συστημάτων
  - 6.1.4. Δυνατότητες της MIB II για διαχείριση χαμηλών επιπέδων
- 6.2. Διαχείριση διαμορφωτών (τέστ βρόγχου)
- 6.3. Θέματα διαχείρισης δημόσιων και ιδιωτικών δικτύων X.25
- 6.4. HELASPAC
- 6.5. Ευφυείς πολυπλέκτες. Ιδιωτικά λογικά δίκτυα: HELASCOM
- 6.6. ISDN - Διαχείριση του ISDN
- 6.7. Ασκήσεις
- 6.7. Βιβλιογραφία

#### 6.0. Εισαγωγή

Στο κεφάλαιο αυτό θα ασχοληθούμε με την διαχείριση των χαμηλότερων επιπέδων της στοίβας αναφοράς πρωτοκόλλων του OSI, δηλαδή κυρίως με το φυσικό επίπεδο, το επίπεδο σύνδεσης δεδομένων, αλλά και το επίπεδο δικτύου. Επειδή η διαχείριση των επιπέδων αυτών συνήθως δεν ακολουθεί κάποιο πρότυπο, αλλά στηρίζεται κατά κύριο λόγο σε εξειδικευμένα προϊόντα, αξίζει να την μελετήσουμε σε ένα ξεχωριστό κεφάλαιο.

#### 6.1. Διαχείριση φυσικού επιπέδου και επιπέδου MAC

Μπορούμε να παρατηρήσουμε στο Κεφάλαιο 4, ότι η διαχείριση του φυσικού επιπέδου, όπως επίσης και του υποεπιπέδου ελέγχου προσπέλασης του μέσου (Medium Access Control, MAC) δεν αντιμετωπίζεται ικανοποιητικά στα υπάρχοντα πρότυπα. Προσπάθειες γίνονται τόσο στην κοινότητα των προτύπων TCP/IP, όσο και στην κοινότητα των προτύπων OSI για διεύρυνση του φάσματος εφαρμογής των τωρινών προτύπων και επίτευξη ολοκληρωμένης πολυεπίπεδης διαχείρισης.

Στο κεφάλαιο αυτό στοχεύουμε να εξετάσουμε τον τρόπο με τον οποίο επιτυγχάνεται η διαχείριση των χαμηλών επιπέδων σήμερα και ο οποίος κατά κύριο λόγο στηρίζεται σε εξειδικευμένα συστήματα. Επίσης θα εξετάσουμε τις προσπάθειες τις οποίες γίνονται για την επέκταση των προτύπων διαχείρισης δικτύων με σκοπό την επικοινωνία του υπεύθυνου διαχειριστικού συστήματος (Network Management System, NMS) με τα χαμηλά επίπεδα.

### 6.1.1. Συστήματα διαχείρισης γραμμών (Cable Management Systems)

Αρχικά θα δούμε μια σημαντική κατηγορία συστημάτων διαχείρισης (κυρίως) του φυσικού επιπέδου, τα οποία ονομάζονται συστήματα διαχείρισης γραμμών (Cable Management Systems). Τα συστήματα αυτά αποτελούν ένα ουσιαστικό μέρος της τοπολογικής διαχείρισης (με την έννοια του configuration management) σε ιδιωτικά τοπικά δίκτυα αλλά και σε ευρύτερα τηλεπικοινωνιακά δημόσια και ιδιωτικά δίκτυα.

Τα συστήματα διαχείρισης γραμμών είναι συνήθως εφαρμογές που τρέχουν σε προσωπικούς υπολογιστές και οι οποίες διατηρούν βάσεις δεδομένων, που επιτρέπουν στον χρήστη να διαχειριστεί πληροφορίες σχετικές με τις διάφορες γραμμές που υπάρχουν και χρησιμοποιούνται σε κάποιο περιβάλλον εργασίας. Οι εφαρμογές αυτές συχνά προσφέρουν γραφικές CAD (Computer Aided Design) δυνατότητες, για την καλύτερη γραφική απεικόνιση των καλωδιώσεων σε κάποιο όροφο. Τα πρώτα συστήματα διαχείρισης γραμμών αναπτύχθηκαν κυρίως για την διαχείριση τηλεπικοινωνιακών καλωδίων που σχετίζονταν με κάποιο PBX (Private Branch Exchange). Πιο σύγχρονες εκδόσεις μπορούν επιπλέον να παρακολουθήσουν ένα μεγαλύτερο αλλά πάντα περιορισμένο αριθμό από μέσα μεταφοράς, όπως κάποιο τοπικό δίκτυο (π.χ. IEEE 802.3 "Ethernet").

Ένα σύστημα διαχείρισης γραμμών μπορεί να λειτουργήσει πάνω από πολλών ειδών πλατφόρμες. Στην συνηθισμένη περίπτωση, η πλατφόρμα είναι κάποιος προσωπικός υπολογιστής, ή σε καλύτερες καταστάσεις κάποιος προσωπικός υπολογιστής συνδεδεμένος σε δίκτυο, ώστε και άλλοι υπολογιστές (όπως για παράδειγμα κάποιο NMS) να μπορούν να έχουν πρόσβαση στις βάσεις δεδομένων του.

Όσον αφορά το λογικό που τρέχει σε ένα σύστημα διαχείρισης γραμμών αυτό συνήθως έχει την δυνατότητα εκτέλεσης των παρακάτω λειτουργιών :

- Αναγνώριση γραμμών και μονοπατιών.
- Παρουσίαση της διαδρομής των γραμμών.
- Προσδιορισμό των δυνατοτήτων και χαρακτηριστικών γραμμών και διαδρομών.
- Παρακολούθηση της χρησιμοποίησης των γραμμών.
- Διατήρηση αρχείου των μηχανημάτων των συνδεδεμένων σε κάθε γραμμή.
- Διατήρηση αρχείου βλαβών και συναγερμών.
- Μηχανισμό διαχείρισης και αναφοράς λειτουργιών.

Στην περίπτωση που το σύστημα διαχείρισης γραμμών είναι κάποιος εξυπηρετητής (server) σε κάποιο περιβάλλον εργασίας αποτελούμενο και από διάφορους πελάτες (clients), θα πρέπει να ληφθεί υπ' όψη η δυνατότητα για RPCs (Remote Procedure Calls), μεταξύ server και clients, ώστε να ελαχιστοποιείται το μέγεθος της μεταφερόμενης πληροφορίας και η χρησιμοποίηση της υπολογιστικής δύναμης του πελάτη σταθμού εργασίας.

Τα συστήματα διαχείρισης γραμμών χρησιμοποιούν πολλά διαφορετικά είδη αρχιτεκτονικών για την αποθήκευση των διαθέσιμων στοιχείων. Τα σχεσιακά συστήματα προσφέρουν μάλλον την υψηλότερη ευελιξία, μια και προσφέρουν στον χρήστη μέσω των

δομημένων ερωτήσεων πολλές όψεις της πληροφορίας. Πολλές εταιρείες αναπτύσσουν ιδιαίτερους μηχανισμούς αποθήκευσης της πληροφορίας σε συστήματα διαχείρισης γραμμών, είναι όμως πολλές φορές καλύτερη η χρησιμοποίηση έτοιμων πακέτων (όπως dBASE, R:base, κ.ά.) ή προτύπων, προκειμένου να είναι δυνατή η επικοινωνία μεταξύ διαφορετικών συστημάτων.

Άλλα επιθυμητά χαρακτηριστικά σε ένα σύστημα διαχείρισης γραμμών είναι ένας σχεδιαστής/γεννήτρια αναφορών που επιτρέπει σε κάθε χρήστη να καθορίσει την προσωπική του επιλογή, όσο αναφορά την μορφή των αναφορών του συστήματος. Επίσης η δημιουργία ετικετών είναι ένα ακόμα σημαντικό χαρακτηριστικό, μια και η αναγνώριση των διαφορετικών γραμμών είναι ένας σημαντικός παράγοντας για την σωστή αρχικοποίηση του επικοινωνιακού συστήματος.

• Circuit ID
• Circuit Path
• Number of Cross-Connects in Circuit Path
• Cross-Connect Location
• Predecessor Circuit ID (for bus LANs)
• Successor Circuit ID (for bus LANs)
• Circuit Length
• Wire Type
• Characteristic Impedance
• Wire Source
• Date Installed
• Installer
• Date Deactivated
• Deactivation Reason
• Change Description/Date (Text)
• Number of Changes (Change Descriptions above)
• Network Type
• User Name
• User Location
• User Telephone
• Security Classification
• Devices Terminated (Manufacturer, Model, Serial Number)
• Number of Devices Terminated
• Trouble Report Date Opened
• Trouble Report Time Opened
• Trouble Report Date Closed
• Trouble Report Time Closed
• Trouble Report Priority
• Trouble Report Elapsed Time
• Trouble Report Problem Reporter
• Trouble Report Name, Location, Telephone
• Trouble Report Problem Description (Reporter perception)
• Trouble Report Problem Resolution (Technician perception)
• Trouble Report Technician ID
• Trouble Report Time Expended
• Trouble Report Keywords (Problem Resolution for search functions)
• Service Order Subscriber Name, Address, Telephone Number
• Service Order Equipment Information
• Service Order Pending
• Service Order Completed

**Πίνακας 6.1 - Εγγραφές σε βάση πληροφοριών συστήματος διαχείρισης γραμμών-MIB**

Θα ολοκληρώσουμε την παράγραφο αυτή κάνοντας ορισμένες παρατηρήσεις σχετικές με το μέλλον των συστημάτων διαχείρισης γραμμών. Έχουμε λοιπόν να παρατηρήσουμε τα εξής:

- Οι νέες εξελίξεις στα τοπικά δίκτυα υψηλών ταχυτήτων και στα ψηφιακά δίκτυα ολοκληρωμένων υπηρεσιών θα καθορίσουν τις νέες απαιτήσεις από τα συστήματα διαχείρισης γραμμών.
- Οι εξελίξεις στη δημιουργία προτύπων θα επηρεάσει θετικά την βιομηχανία των συστημάτων αυτών, προσφέροντας δυνατότητες επικοινωνίας μεταξύ ετερογενών συστημάτων.
- Επίσης οι εξελίξεις στις σχεσιακές αρχιτεκτονικές βάσεων δεδομένων και στο μοντέλο πελάτη-εξυπηρετητή θα βοηθήσουν στην καλύτερη χρησιμοποίηση της πληροφορίας που συλλέγει κάποιο σύστημα διαχείρισης γραμμών.
- Ιδιαίτερα, η επικοινωνία των προγραμμάτων διαχείρισης γραμμών μέσω του πρωτοκόλλου SNMP (ή και άλλου πρωτοκόλλου διαχείρισης) θα οδηγήσει σε μια ενοποιημένη και πολυεπίπεδη διαχείριση, όπου ο χρήστης από μια μοναδική κονσόλα θα μπορεί να διαχειριστεί κάθε μέρος του δικτύου που τον ενδιαφέρει.
- Τέλος η χρησιμοποίηση έμπειρων συστημάτων (expert systems) είναι πολύ πιθανή σε μελλοντικά συστήματα διαχείρισης γραμμών.

Στον Πίνακα 6.1 βλέπουμε τις ελάχιστες εγγραφές για μία βάση δεδομένων ενός συστήματος διαχείρισης γραμμών.

### 6.1.2. Εξοπλισμός διάγνωσης προβλημάτων (Diagnostic Devices)

Στην συνέχεια θα εξετάσουμε διάφορες συσκευές, οι οποίες προσφέρουν δυνατότητες διάγνωσης προβλημάτων. Οι συσκευές αυτές συχνά συνεργάζονται με τα συστήματα διαχείρισης γραμμών προκειμένου να επιτευχθούν καλύτερα αποτελέσματα.

Οι συσκευές διάγνωσης προβλημάτων αποτελούν κάποιο συνδυασμό υλικού (hardware) / λογικού (software) ή είναι μόνο υλικό και χρησιμοποιείται για παρακολούθηση ή και έλεγχο κάποιου τοπικού συνήθως δικτύου.

Οι πιο συνηθισμένες μηχανές διάγνωσης προβλημάτων είναι οι παρακάτω:

- **Protocol Analyzers:** Είναι τα πιο πολύπλευρα μηχανήματα διαχείρισης (όχι μόνο του φυσικού επιπέδου). Είναι βέβαια ακριβότερα από άλλες συσκευές διαχείρισης, αλλά μπορούν να αναλύσουν σε βάθος προβλήματα που έχουν κάποιο λογικό αίτιο. Ένας protocol analyzer συλλέγει πλαίσια και εξετάζει το περιεχόμενό τους, έχοντας πρόσβαση σε ολόκληρη την πληροφορία που περιέχει το πλαίσιο και η οποία αφορά το φυσικό έως και το επιπέδο εφαρμογής.
- **Time Domain Reflectometers (TDRs):** Η συσκευή αυτή λειτουργεί σύμφωνα με τις αρχές της θεωρίας μετάδοσης σημάτων σε γραμμές μεταφοράς. Αν η γραμμή μεταφοράς δεν είναι τερματισμένη στην χαρακτηριστική της αντίσταση (όπως για παράδειγμα σε κάποιο σπάσιμο της γραμμής), τότε έχουμε ανάκλαση κυμάτων. Το TDR δημιουργεί κύματα στην γραμμή και ταυτόχρονα ακούει για

τυχών ανακλάσεις. Στην συνέχεια μέσω απλών υπολογισμών δίνει στον χρήστη παραμέτρους σχετικές με την μετάδοση του σήματος και την απόσταση της τυχούσας βλάβης από το σημείο μέτρησης. Η ακρίβεια προοσδιορίσμου της θέσης της βλάβης (από μερικά πόδια μέχρι μερικές ίντσες) εξαρτάται από την τιμή της συσκευής. Το TDR μπορεί να συνοδεύει κάποιο σύστημα διαχείρισης γραμμών από αυτά που αναφέραμε στην προηγούμενη παράγραφο, αλλά μπορεί να είναι και αυτόνομο. Επίσης τις μετρήσεις ενός TDR μπορεί να επεξεργαστεί και κάποιος protocol analyzer.

- **Optical Reflectometers:** Ισχύουν τα ίδια που αναφέραμε και παραπάνω για τα TDRs, με την διαφορά ότι αναφερόμαστε σε οπτικές ίνες και ότι η τιμή ανεβαίνει σημαντικά. Επίσης στην περίπτωση των οπτικών ινών μια τέτοια συσκευή είναι αναγκαία από την άποψη ότι η οπτική ίνα είναι μέσο μεταφοράς πολύ υψηλής αξιοπιστίας και δεν επιτρέπεται να παρουσιάζει την οποιαδήποτε ανωμαλία.
- **Volt-Ohmmeters:** Τα βολτόμετρα είναι σίγουρα το πιο φθηνό όργανο μετρήσεων. Μπορεί να χρησιμοποιηθεί για την μέτρηση της αγωγιμότητας του χαλκού, όταν ψάχνει κανείς για διακοπές στην συνέχεια των γραμμών. Μπορεί προσεγγιστικά να χρησιμοποιηθεί για την εκτίμηση του επιπέδου των σημάτων και των απωλειών. Είναι φανερό, ότι δεν μπορούν να εντοπίσουν, όπως κάποιο TDR το σημείο της ασυνέχειας της γραμμής, ούτε να προσφέρουν γραφικές παραστάσεις. Σε γενικές γραμμές ένα βολτόμετρο μπορεί να δώσει μια εκτίμηση του επιπέδου των σημάτων και των γειώσεων.
- **Cable Tracers:** Είναι ζευγάρι συσκευών, που χρησιμοποιούνται για την ανίχνευση της διαδρομής μιας γραμμής. Η μία συσκευή μεταδίδει σήμα στην γραμμή και η άλλη ανιχνεύει το σήμα αυτό μέσα από τοίχους, ταβάνια, σωλήνες δείχνοντας έτσι τη διαδρομή της γραμμής. Περιορισμός είναι, ότι η δεύτερη συσκευή πρέπει να βρίσκεται σε μια απόσταση μικρότερη από ένα πόδι για να μπορεί να ανιχνεύει το σήμα.
- **Oscilloscopes:** Κάποιος παλμογράφος μπορεί επίσης να χρησιμοποιηθεί για την εκτίμηση της ποιότητας του μεταδιδόμενου σήματος. Μας βοηθάει στην διάγνωση προβλημάτων του υλικού στην μετάδοση των σημάτων.

Στην επόμενη παράγραφο θα προσπαθήσουμε να συνδέσουμε όλες τις συσκευές που αναφέραμε, με ότι ονομάζουμε διαχείριση χαμηλών επιπέδων.

### 6.1.3. Εφαρμογές των παραπάνω συστημάτων

Ας εξετάσουμε, όμως τώρα τις περιπτώσεις στις οποίες μπορεί να χρησιμοποιηθεί μια μηχανή διάγνωσης προβλημάτων. Είναι φανερό, ότι κύριος στόχος της είναι η έγκαιρη διάγνωση προβλημάτων. Έχουμε λοιπόν τις παρακάτω περιπτώσεις:

- **Ορισμός της κατάστασης λειτουργίας του δικτύου:** Όταν κάποιο δίκτυο εγκαθίσταται κάπου και η λειτουργία του συντονίστει, θα πρέπει οι παράμετροι λειτουργίας του να μετρηθούν και να αποθηκευτούν σαν ορισμός της ομαλής λειτουργίας του δικτύου. Στην συνέχεια οι παράμετροι αυτοί θα πρέπει να ελέγχονται περιοδικά ώστε να ανιχνεύονται οποιεσδήποτε τάσεις απόκλισης από τις τιμές αυτές. Από τις αποκλίσεις αυτές, βέβαια άλλες θα είναι δικαιολογημένες και το δίκτυο θα πρέπει να προσαρμοστεί σ' αυτές και άλλες θα είναι ενδείξεις σφαλμάτων και θα πρέπει να διορθωθούν. Μερικές βασικές τέτοιες μετρήσεις είναι:

- το μέσο φορτίο του δικτύου
- το ανώτατο φορτίο του δικτύου
- την συχνότητα λαθών και επαναμεταδόσεων
- την συχνότητα αρχικοποιήσεων στοιχείων του δικτύου

Κάθε είδος δικτύου - δακτύλιος με κουπόνι, αρτηρία με κουπόνι κ.ά. - έχει τις δικές του ιδιαίτερες παραμέτρους και χαρακτηριστικά. Οι περισσότερες μάλιστα από τις παραμέτρους αυτές δίνονται από τα αντίστοιχα πρότυπα (π.χ. 802.x), οπότε το έργο της διαχείρισης διευκολύνεται σημαντικά.

- **Προληπτικές διαγνώσεις:** Προληπτικοί έλεγχοι σ' ένα δίκτυο μπορούν να αποκαλύψουν:
  - προβλήματα που έρχονται και φεύγουν
  - προβλήματα στις γραμμές μεταφοράς
  - αύξηση του φορτίου από αλλαγές στο λειτουργικό σύστημα ή στις εφαρμογές που τρέχουν
  - αύξηση του φορτίου από αφίξεις από εξωτερικά δίκτυα ή άλλα υποδίκτυα
- **Επίλυση επιφανειακών προβλημάτων μέσω μη σχολαστικής παρακολούθησης του δικτύου:** Η μη σχολαστική παρακολούθηση δεν προσθέτει καθόλου φορτίο στο δίκτυο. Η πληροφορία αναλύεται σε πραγματικό χρόνο και οποιεσδήποτε ανώμαλες καταστάσεις καταγράφονται για παραπέρα επεξεργασία. Μια τέτοια παρακολούθηση επιτυγχάνεται, για παράδειγμα, με κάποιο protocol analyzer. Σημειώνουμε εδώ ότι μια γέφυρα ή ένας δρομολογητής περιορίζει την πληροφορία, που μπορεί να συλλέξει κάποιος protocol analyzer, αφού πλαίσια από άλλα τμήματα δεν φθάνουν στο τμήμα που ελέγχεται.
- **Επίλυση βαθύτερων προβλημάτων μέσω σχολαστικού ελέγχου του δικτύου:** Σχολαστικός έλεγχος σημαίνει την αποστολή ειδικής πληροφορίας μέσα από το δίκτυο, ώστε να ελεγχθεί αν θα φθάσει στον ακριβή προορισμό, να ελεχθεί ο αριθμός των λαθών, ο ρυθμός των λαθών, οι μορφές τους κ.ά. Ανάλογα με την περίπτωση κάποιος σχολαστικός έλεγχος είναι δυνατόν να διακόψει την λειτουργία του δικτύου ή απλά να αυξήσει το φορτίο του δικτύου.
- **Παρακολούθηση του δικτύου μετά την επίλυση του οποιοδήποτε προβλήματος:** Μετά από την επίλυση κάποιου προβλήματος το δίκτυο πρέπει να παρακολουθείται τόσο σχολαστικά, όσο και μη σχολαστικά, ώστε να επιβεβαιωθεί ότι το πρόβλημα έχει όντως λυθεί, και προκειμένου να μην εμφανιστεί κάποιο πρόβλημα που κρυβόταν από το προηγούμενο.

#### **6.1.4. Δυνατότητες της MIB II για διαχείριση χαμηλών επιπέδων**

Μέχρι τώρα είδαμε εξειδικευμένες εφαρμογές και προϊόντα που χρησιμοποιούνται για την διαχείριση των χαμηλών επιπέδων. Θα ρίξουμε τώρα μια σύντομη ματιά στις προσπάθειες τις οποίες γίνονται για την επέκταση των προτύπων διαχείρισης δικτύων με κύριο στόχο την επικοινωνία του διαχειριστικού συστήματος με τα χαμηλά επίπεδα.

Όπως μπορεί να παρατηρήσει κανείς, το transmission group στην MIB II {mib-2 10} (το οποίο σημειώνουμε ότι δεν υπήρχε στην MIB I) δεν έχει προς το παρόν ορισθεί, δηλ. είναι άδειο. Βρίσκεται στο σημείο αυτό της MIB II σαν place-holder για MIBs που

εξαρτώνται από το μέσο μεταφοράς. Προσπάθειες γίνονται στην TCP/IP κοινότητα για τον ορισμό βάσεων πληροφορίας διαχείρισης (Management Information Base, MIB) για κάθε δυνατό μέσο μεταφοράς. Όλα αυτά τα MIBs θα βρίσκονται ιεραρχικά κάτω από το transmission group και θα προσφέρουν τις απαραίτητες δυνατότητες για πολυεπίπεδη διαχείριση. Αυτή την στιγμή, οι ορισμοί βρίσκονται σε πειραματικό ακόμα στάδιο. Κάθε ομάδα εργασίας είναι πλήρως αφιερωμένη στην ανάπτυξη ενός MIB για ένα συγκεκριμένο μέσο μεταφοράς (X.25, Point-to-Point protocol, FDDI, Ethernet-like, κ.α.). Στον Πίνακα 6.2 βλέπουμε μια σειρά από RFCs με media-like MIBs. Από αυτά μόνο το RFC-1398 για Ether-like MIBs είναι Draft Standard, ενώ τα υπόλοιπα είναι πειραματικά πρότυπα.

1230 - IEEE 802.4 Token Bus Interface Type MIB
1231 - IEEE 802.5 Token Ring Interface Type MIB
1232 - DS1 Interface Type MIB
1233 - DS2 Interface Type MIB
1243 - AppleTalk MIB
1253 - OSPF version 2 MIB
1269 - BGP version 3 MIB
1271 - Remote LAN Monitoring MIB
1285 - FDDI Interface Type MIB
1286 - Bridge MIB
1289 - DECnet phase IV MIB
1304 - SMDS Interface Protocol (SIP) Interface Type MIB
1315 - Frame Relay DTE Interface Type MIB
1316 - Character Stream Device MIB
1317 - RS-232 Interface Type MIB
1318 - Parallel Printer Interface Type MIB
1368 - IEEE 802.3 Repeater MIB
1381 - X.25 LAPB MIB
1382 - X.25 PLP MIB
1389 - RIPv2 MIB
1398 - Ether-Like Interface Type MIB
1406 - DS1/E1 Interface Type MIB
1407 - DS3/E3 Interface Type MIB

**Πίνακας 6.2 - Προτεινόμενα για πρότυπα RFCs σχετικά με media like MIBs**

## 6.2. Διαχείριση διαμορφωτών (τέστ βρόγχου)

Η διαχείριση **διαμορφωτών/αποδιαμορφωτών (modems)** επιτυγχάνεται με τέστ βρόγχων. Οι βρόγχοι (loop) ελέγχου είναι ένα από τα πιο χρήσιμα διαγνωστικά εργαλεία που επιτρέπουν στον χειριστή του δικτύου δεδομένων να εντοπίσει το σημείο βλάβης σε μια επικοινωνιακή σύνδεση. Ο διαγνωστικός βρόγχος προκαλεί αναδρομολόγηση της πληροφορίας προς τα πίσω, δίνοντας τη δυνατότητα σύγκρισης με τη μεταδιδόμενη πληροφορία. Σε περίπτωση διαφορών τότε υπάρχει κάποιο πρόβλημα.

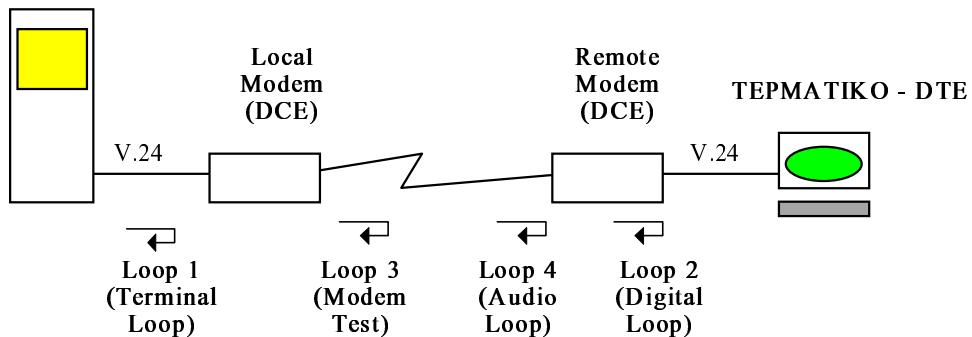
Οι κυριότεροι βρόγχοι ελέγχου που έχουν τυποποιηθεί από την CCITT στην σύσταση V.54 είναι οι εξής (βλ. και Σχήμα 6.1) [ΑΛΕΞ92]:

- **Βρόχος 1 (terminal loop).** Είναι σχεδιασμένο ώστε να ελέγχει τη τερματική συσκευή (DTE).
- **Βρόχος 3 (modem test).** Φροντίζει για τον έλεγχο του τοπικού modem.

- **Βρόχος 2 (digital test).** Φροντίζει για τον έλεγχο όλης της σύνδεσης, περιλαμβανομένων του τοπικού modem, της γραμμής και του απομακρυσμένου modem.
- **Βρόχος 4 (audio loop).** Ελέγχει την αξιοπιστία της αναλογικής γραμμής μεταξύ των δύο modems.

Επιπλέον, όταν στις γραμμές μετάδοσης παρατηρηθούν υψηλοί ρυθμοί λαθών ή υπερβολικός θόρυβος τότε τα modems αυτόματα αναδιπλώνονται (fall-back) σε χαμηλότερες ταχύτητες μετάδοσης (π.χ. από 9,6 Kbits/sec σε 4,8 Kbits/sec).

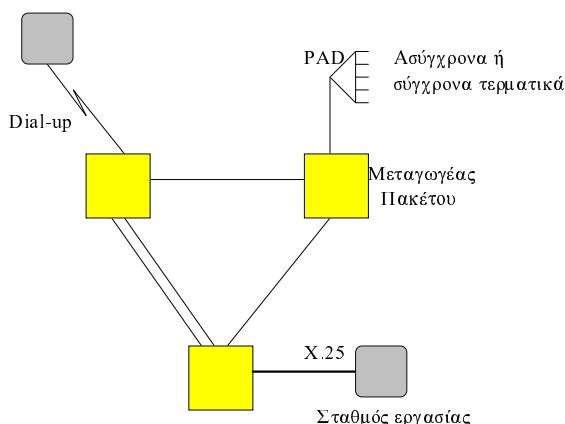
#### ΥΠΟΛΟΓΙΣΤΗΣ - DTE



Σχήμα 6.1 - Λιαγνωστικοί βρόχοι

### 6.3. Θέματα διαχείρισης δημόσιων και ιδιωτικών X.25 δικτύων

Η χρησιμοποίηση δικτύων μεταγωγής πακέτου για εφαρμογές μετάδοσης δεδομένων έχει εξαπλωθεί πολύ στις μέρες μας. Τα δίκτυα αυτά είναι δημοφιλή γιατί επιτρέπουν με ένα τρόπο χαμηλού κόστους την πρόσβαση σε μεγάλα υπολογιστικά συστήματα. Ο πιο συνηθισμένος τρόπος πρόσβασης σε ένα δίκτυο μεταγωγής πακέτου είναι οι X.25 γραμμές. Το X.25 είναι πρωτόκολλο της CCITT που περιγράφει τον τρόπο πρόσβασης των δικτύων μεταγωγής πακέτου και πιο συγκεκριμένα τον τρόπο επικοινωνίας ενός τερματικού (Data Termination Equipment, DTE) με τον πρώτο κόμβο του δικτύου που είναι συνδεδεμένο (Data Connection Equipment, DCE). Παρακάτω θα εξετάσουμε θέματα που αφορούν την αρχιτεκτονική των δικτύων αυτών, πριν προχωρήσουμε σε θέματα που αφορούν την διαχείρισή τους.



## Σχήμα 6.2 - Αρχιτεκτονική X.25 δικτύου

Η σύσταση X.25 της CCITT περιλαμβάνει πρωτόκολλα για τα τρία χαμηλότερα επίπεδα της στοιβας αναφοράς πρωτοκόλλων OSI.

- **Φυσικό:** Στο φυσικό επίπεδο έχουμε τις συστάσεις CCITT X.21 και X.21 bis οι οποίες καθορίζουν τα ηλεκτρικά και λειτουργικά χαρακτηριστικά των γραμμών.
- **Σύνδεσης δεδομένων:** Στο επίπεδο σύνδεσης δεδομένων προτείνεται η διαδικασία Higher Data Link Control Link Access Protocol Balanced (HDLC LAPB).
- **Δικτύου:** Τέλος, στο επίπεδο δικτύου έχουμε την σύσταση X.25 που καθορίζει τον τρόπο μεταφοράς δεδομένων μεταξύ των DTEs των χρηστών και του δικτύου μεταγωγής πακέτου.

Τα X.25 δίκτυα πακέτου συνήθως ακολουθούν την δομή του Σχήματος 6.2. Το δίκτυο χρησιμοποιεί κόμβους μεταγωγής πακέτου (Packet Switching Nodes), τους οποίους διασυνδέει με κατάλληλες γραμμές μεταφοράς. Το Σχήμα 6.2 επίσης δείχνει μερικούς από τους τρόπους πρόσβασης σε δίκτυο X.25. Ο σταθμός εργασίας, για παράδειγμα είναι συνδεδεμένος με κάποιο μεταγωγέα πακέτου με X.25 σύνδεσμο μέσα από τον οποίο (μία φυσική σύνδεση) μπορούν να περάσουν δεδομένα πολλών νοητών κυκλωμάτων. Αντίθετα κάποιο τερματικό συνήθως δεν μπορεί να συνδεθεί σ' ένα μεταγωγέα, οπότε συνδέεται σε μια ειδική συσκευή, που ονομάζεται Packet Assembler/Disassembler (PAD). Το PAD βέβαια είναι στην συνέχεια, το ίδιο συνδεδεμένο με κάποιο μεταγωγέα πακέτων. Άλλος τρόπος σε ένα δίκτυο X.25 είναι η χρήση διαμορφωτών / αποδιαμορφωτών (Modulators/Demodulators, MODEMs).

Άλλο σημείο, το οποίο παρατηρούμε, είναι η πιθανότητα χρησιμοποίησης περισσοτέρων γραμμών για την σύνδεση δύο μεταγωγέων, προφανώς για λόγους αξιοπιστίας, αυξημένης χωρητικότητας κ.ά. Φυσικά διπλές γραμμές θα μπορούσαν να βρεθούν και στην DTE-DCE σύνδεση για τους ίδιους λόγους. Η δόμηση των μεταγωγών είναι τέτοια, ώστε να υπάρχουν εναλλακτικοί δρόμοι για το φορτίο. Πραγματικά οι μεταγωγείς πακέτου μπορούν δυναμικά να προσπεράσουν ένα προβληματικό σύνδεσμο ή μια συμφορημένη διαδρομή. Αυτές οι αποφάσεις δρομολόγησης είναι διαφανείς στους χρήστες του X.25 δικτύου.

Η διαχείριση των X.25 δικτύων μεταγωγής πακέτου ακολουθεί τους γενικούς κανόνες διαχείρισης οποιουδήποτε δικτύου. Ειδικότερα όμως, οι παράμετροι του X.25 πρωτοκόλλου μπορούν να εισάγουν κάποια μεγαλύτερη πολυπλοκότητα στο όλο πρόβλημα. Γενικά τα συστήματα διαχείρισης X.25 δικτύων είναι βασισμένα σε προσωπικούς υπολογιστές ή και μεγαλύτερες υπολογιστικές μηχανές και συνήθως είναι κατασκευασμένα από την ίδια εταιρεία που αναπτύσσει και τους μεταγωγείς. Βέβαια η εξέλιξη των πρωτοκόλλων και στα χαμηλότερα επίπεδα θα επιτρέψει κάποια στιγμή την πολυεπίπεδη διαχείριση ετερογενών δικτύων από μια μοναδική κονσόλα.

Η διαχείριση βλαβών και σφαλμάτων στα X.25 δίκτυα απαιτεί παρόμοιες συναρτήσεις μ' αυές που χρησιμοποιούνται και σε οποιοδήποτε άλλο σύγχρονο δίκτυο. Η κύρια διαφορά είναι ότι τα X.25 πρωτόκολλα εγγυούνται την ασφαλή μεταφορά των πακέτων. Ετσι ακόμα και σε περίπτωση βλάβης σε σύνδεσμο, υλικό ή λογισμικό του δικτύου θα πρέπει τα πακέτα να επαναδρομολογούνται δυναμικά. Εξετάζοντας λοιπόν την διαχείριση βλαβών και σφαλμάτων, όσο αναφορά το υλικό, η αξιοπιστία που χρειάζεται ένα X.25 δίκτυο μπορεί να εξασφαλιστεί με διπλά κυκλώματα, διπλούς συνδέσμους, διπλές προσβάσεις στο δίκτυο, διπλές παροχές ισχύος κ.ά. Ετσι, για παράδειγμα σε περίπτωση βλάβης κάποιας γραμμής, το λογισμικό υλοποίησης του X.25 πρωτοκόλλου στον σταθμό εργασίας και οι πίνακες δρομολόγησης στον μεταγωγέα

αναγνωρίζουν την βλάβη και επαναδρομολογούν δυναμικά όλο το φορτίο στον εφεδρικό σύνδεσμο.

Τα X.25 δίκτυα απαιτούν καθορισμό διαφόρων λειτουργικών παραμέτρων. Τέτοιες παράμετροι είναι οι διευθύνσεις, ο μέγιστος αριθμός των ταυτόχρονα ανοικτών νοητών κυκλωμάτων, εναλλακτικές διαδρομές σε περίπτωση βλαβών. Μεγάλης σημασίας είναι ο αριθμός των ταυτόχρονα ανοικτών νοητών κυκλωμάτων. Ο αριθμός αυτός πρέπει να είναι αρκετά μεγάλος, ώστε να αποφεύγονται η απόρριψη κλήσεων τις ώρες υψηλής κυκλοφορίας. Μια X.25 γραμμή στα 64Kbps μπορεί να χειρίστει 20 με 30 νοητές κλήσεις στα 1200, 2400 και 9600 baud. Μια αύξηση βέβαια της χρησιμοποίησης πάνω από το 75% θα πρέπει να οδήγησει τον διαχειριστή σε εκτίμηση της πιθανότητας εγκατάστασης μιας δεύτερης γραμμής.

Ο διαχειριστής του δικτύου οφείλει να συλλέγει στοιχεία για τις επιδόσεις του δικτύου, όχι μόνο για την διάγνωση καταστάσεων συμφόρησης, αλλά και για την διασφάλιση της ικανοποιητικής λειτουργίας όλων των κόμβων.

Η λογιστική διαχείριση περιλαμβάνει την συγκέντρωση λειτουργικών στατιστικών στοιχείων για το δίκτυο. Σε αυτά τα στοιχεία περιλαμβάνονται ο αριθμός των χρηστών, το μέρος προέλευσής τους, ο χρόνος παραμονής τους στο σύστημα, το φορτίο που εισήγαγαν στο σύστημα. Η χρέωση θα βασιστεί σ' αυτά τα στοιχεία. Τα στοιχεία, βέβαια αυτά, βοηθούν και στην διαχείριση επιδόσεων, αφού φανερώνουν σημεία στα οποία το δίκτυο θα μπορούσε να βελτιωθεί.

## 6.4. HELASPAC

Το δημόσιο δίκτυο μεταγωγής πακέτου HELASPAC βρίσκεται σε λειτουργία από το 1989. Υποστηρίζει τις συστάσεις της CCITT για μεταφορά δεδομένων μέσα από δίκτυα μεταγωγής πακέτου X.25. Η πρόσβαση στο δίκτυο HELASPAC μπορεί να γίνει είτε μέσω μόνιμης σύνδεσης, είτε μέσω του επιλεγόμενου τηλεφωνικού δικτύου ενώ κατάλληλα πρωτόκολλα (π.χ. X.75) συνδέουν το δίκτυο HELASPAC με ανάλογα δημόσια δίκτυα μεταγωγής πακέτου στο εξωτερικό.

Το δίκτυο HELASPAC χρησιμοποιεί κόμβους μεταγωγής πακέτου (Packet Switching Nodes), τους οποίους συνδέει με κατάλληλες γραμμές μεταφοράς. Πάνω στους κόμβους αυτούς συνδέονται οι χρήστες του δικτύου. Οι συνδέσεις διακρίνονται σε δύο κατηγορίες: μόνιμες συνδέσεις και συνδέσεις μέσω του επιλεγόμενου τηλεφωνικού δικτύου. Ανάλογα με τον τύπο του τερματικού του χρήστη διακρίνουμε ακόμα τις συνδέσεις σε σύγχρονες όπου χρησιμοποιούνται σύγχρονα τερματικά πακέτου και το πρωτόκολλο X.25 και ασύγχρονες, όπου χρησιμοποιούνται ασύγχρονα τερματικά και το πρωτόκολλο X.28. Για να συνδεθεί οποιοδήποτε τερματικό στο HELASPAC απαιτείται η χρησιμοποίηση ενός ζευγαριού από modems. Τα modems μετατρέπουν την πολλαπλή έξοδο του τερματικού κατάλληλα ώστε να περάσει από τις δισύρματες ή τετρασύρματες γραμμές. Τα modem επίσης καθορίζουν και τη ταχύτητα μεταβιβασης δεδομένων στην γραμμή (bps). Οι ταχύτητες πρόσβασης κυμαίνονται από 300-2400 bps για ασύγχρονη πρόσβαση και από 2400-9600 bps για σύγχρονη επικοινωνία. Με σύνδεση βασικής ζώνης (baseband modems) είναι δυνατή και η επίτευξη ταχυτήτων πάνω από 19200 bps έως και 64 Kbps.

Η εμπορική χρήση του HELASPAC έχει ουσιαστικά ξεκινήσει από το 1990. Η αρχική του τοπολογία περιλαμβάνει 8 κόμβους (Αθήνα, Πειραιά, Θεσσαλονίκη, Καβάλα, Λάρισα, Ηράκλειο, Πάτρα, Τρίπολη) και ένα Κεντρικό Διαχειριστικό Σύστημα εγκατεστημένο στην Αθήνα, το οποίο χρεώνει την κίνηση του φορτίου, συλλέγει στατιστικά στοιχεία και ελέγχει τις μονάδες του δικτύου. Κατά την περίοδο '90-'93

προστέθηκαν άλλοι 8 κόμβοι, ενώ για την περίοδο '93-'94 σχεδιάζεται η επέκταση σε 35 κόμβους με 4 πρόσθετα διαχειριστικά συστήματα.

Το τιμολόγιο του HELASPAC είναι ανεξάρτητο της απόστασης μεταξύ ανταποκρινόμενων χρηστών και ανεξάρτητο της απόστασης μεταξύ του χρήστη και του σημείου πρόσβασή του στο δίκτυο. Το τιμολόγιο περιλαμβάνει το τέλος σύνδεσης το οποίο καταβάλλεται εφ' άπαξ, το πάγιο μηνιαίο τέλος το οποίο εξαρτάται από τις υπηρεσίες που επιλέγει ο χρήστης και το τέλος επικοινωνίας το οποίο περιλαμβάνει το τέλος αποκατάστασης επικοινωνίας, το τέλος διάρκειας της επικοινωνίας, και το τέλος όγκου κίνησης. Σε γενικές γραμμές το τιμολόγιο του HELASPAC είναι υψηλό, ειδικά αν συγκριθεί με ανάλογα δίκτυα στο εξωτερικό. Πάντως, το HELASPAC προσφέρει πολλές δυνατότητες για τη δημιουργία δικτύων σε τράπεζες, επιχειρήσεις, αεροπορικές εταιρείες, βιομηχανίες και άλλους που μέχρι τώρα στηρίζονταν στις μισθωμένες γραμμές.

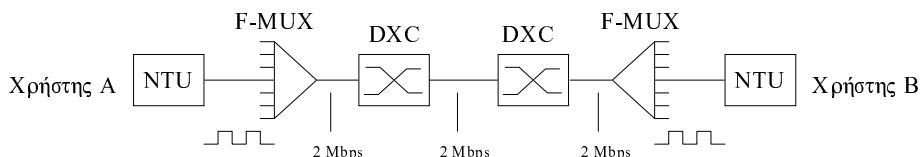
## 6.5. Ευφυείς πολυπλέκτες. Ιδιωτικά λογικά δίκτυα: HELASCOM

Για τον εκσυγχρονισμό της υπηρεσίας παροχής μισθωμένων γραμμών ο ΟΤΕ προχωρεί στη δημιουργία ενός ειδικού δικτύου ψηφιακών γραμμών το οποίο ονομάζεται HELASCOM.

Το δίκτυο HELASCOM σκοπό έχει να δημιουργήσει την υποδομή για την εκμίσθωση από τους πελάτες του ΟΤΕ ψηφιακών κυκλωμάτων, χαμηλών (2400 έως 19200 bps), και υψηλών ταχυτήτων (64 Kbps ή Nx64 Kbps, N=1,2,...,31 μέχρι 2,048 Mbps). Υστερα από προηγούμενη συμφωνία με το διαχειριστικό σύστημα του δικτύου παρέχεται η δυνατότητα χρήσης των γραμμών για μερικό χρόνο.

Το δίκτυο HELASCOM αποτελείται από:

- Διαχειριζόμενοι Ψηφιακοί Μικτονομητές (DXC, Digital Cross Connect Systems)
- Διατάξεις πολυπλεξίας, δηλ. πολυπλέκτες που οδηγούν μια σειρά από 64 Kbps γραμμές σε μια E1 γραμμή των 2,048 Mbps.
- Ένα κεντρικό σύστημα διαχείρισης του δικτύου (NMS, Network Management System)



DXC: Διατάξεις ψηφιακής διασύνδεσης

NTU: Διατάξεις τερματισμού δικτύου DATA

F-MUX: Ευέλικτοι πολυπλέκτες

### Σχήμα 6.3 - Πρόσβαση στο HELASCOM

Ένα σχηματικό δίαγραμμα λειτουργίας του δικτύου HELASCOM μεταξύ δύο χρηστών Α και Β φαίνεται στο Σχήμα 6.3. Τα αναφερόμενα στοιχεία του δικτύου έχουν τις εξής λειτουργίες:

- Οι διατάξεις τερματισμού (NTU) κωδικοποιούν τα ψηφία (data) σε μορφή κατάλληλη για ψηφιακή μετάδοση.
- Οι πολυπλέκτες σκοπό έχουν να συνθέσουν με χρονική διαδοχή τις ομάδες ψηφίων (ψηφιακή πρόσβαση στο δίκτυο μέχρι το χρήστη) που εισέρχονται στην κάθε είσοδο τους προκειμένου να εισαχθούν σε μια E1 γραμμή των 2,048 Mbps.
- Οι ψηφιακοί μικτονομητές (DXC) προγραμματίζονται από το Σύστημα Διαχείρισης (NMS) και συνδέουν τις εισόδους μεταξύ τους ή με άλλον ψηφιακό μικτονομητή. Με τον όρο σύνδεση εννοούμε χρονική αντιστοιχία χρονοσχισμών σε δίκτυα πολυπλεξίας χρόνου (TDM).

Το Σύστημα Διαχείρισης του Δικτύου (NMS) είναι ένα σημαντικό στοιχείο του HELLASCOM. Στον χειριστή της κονσόλας του διαχειριστικού συστήματος φθάνουν οι απαιτήσεις των συνδρομητών του δικτύου και από εκεί καθορίζεται η ώρα της κάθε σύνδεσης, η χρονική της διάρκεια, τα χαρακτηριστικά της (ταχύτητα μετάδοσης δεδομένων) στις διάφορες χρονικές περιόδους, κ.ά., ενώ ταυτόχρονα κρατούνται στατιστικά στοιχεία, στοιχεία χρέωσης, στοιχεία βλαβών και άλλα.

Το HELLASCOM με την πρώτη του κιόλας εγκατάσταση θα καλύψει σχεδόν ολόκληρη τη χώρα. Πιο συγκεκριμένα θα τοποθετηθούν:

- Οκτώ (8) ψηφιακοί μικτονομητές (Cross Connect Systems) στις πόλεις Αθήνα, Πειραιά, Θεσσαλονίκη, Πάτρα και Ηράκλειο, που αποτελούν τον κορμό του δικτύου, συνόλου 480 κυκλωμάτων των 2,048 Mbps.
- Εβδομήντα ένα (71) πολυπλέκτες που θα τοποθετηθούν σε ισάριθμα τοπικά τηλεφωνικά κέντρα, ώστε να καλυφθούν όλες οι μεγάλες πόλεις της χώρας.
- Χίλιες διακόσιες (1200) διατάξεις απόληξης δικτύου (DCE) και τριακόσια (300) modems προκειμένου να χρησιμοποιηθούν για πρόσβαση μέσω του τηλεφωνικού δικτύου, σε περιπτώσεις που υπάρχει έλλειψη συνδρομητικών γραμμών.

Οφέλη των χρηστών από το HELLASCOM μπορούν να θεωρηθούν τα παρακάτω:

- Η δυνατότητα μίσθωσης ψηφιακών κυκλωμάτων σε ολόκληρη τη χώρα, άσχετα με την ολοκλήρωση της ψηφιοποίησης του τηλεφωνικού δικτύου.
- Η δυνατότητα μίσθωσης κυκλωμάτων μονάχα κατά τις χρονικές περιόδους και τις ώρες της ημέρας που αυτά είναι απαραίτητα.
- Η δυνατότητα δημιουργίας κυκλωμάτων μεταξύ περισσοτέρων από δύο χρήστες.
- Η δυνατότητα αλλάγης της ταχύτητας μετάδοσης δεδομένων ανάλογα με τις ανάγκες και τις απαιτήσεις των χρηστών.

Τον Απρίλιο του 1993 θα ξεκινούσε η πιλοτική λειτουργία του HELLASCOM. Το ΕΜΠ χρησιμοποίησε δοκιμαστικά γραμμές του HELLASCOM για τη σύνδεσή του με το FORTHnet και για τη σύνδεση της Πολυτεχνειούπολης Ζωγράφου με τα κτίρια που βρίσκονται στην Πατησίων σε ταχύτητες 64 Kbps.

Εκτός των παραπάνω η ύπαρξη του HELLASCOM θα δώσει την ευκαιρία μίσθωσης κυκλωμάτων των 2,048 Mbps, επιτρέποντας την ανάπτυξη της υπηρεσίας της Τηλεδιάσκεψης (Videoconference). Διεθνείς συνδέσεις παρέχονται μέσω δορυφορικών συνδέσεων (EUTELSAT). Τέλος, ο ΟΤΕ θα προσφέρει και διαχειριστικές δυνατότητες στους τελικούς χρήστες επιτρέποντας τη δημιουργία νοητών ιδιωτικών δικτύων.

Ένα σημαντικό μέρος ενός ψηφιακού δικτύου, όπως του HELASCOM είναι το σύστημα διαχείρισης του δικτύου (Network Management System, NMS). Στην παράγραφο αυτή θα παρακολουθήσουμε τις κυριότερες απαιτήσεις από ένα τέτοιο σύστημα διαχείρισης. Οι λειτουργίες, λοιπόν, τις οποίες οφείλει να εκτελεί ένα σύστημα διαχείρισης δικτύου είναι οι παρακάτω (σαν Δίκτυο θα εννοούμε ιδιωτικό λογικό δίκτυο, όπως το HELASCOM):

**α) Διαχείριση του σχηματισμού του Δικτύου.**

Το NMS θα πρέπει να έχει τις ακόλουθες δυνατότητες για σχηματισμό του Δικτύου:

- Κράτηση εύρους διεύλεσης κατ' απαίτηση ή βασισμένη σε ημερομηνία ή ώρα της ημέρας.
- Αυτόματη διασύνδεση κυκλωμάτων από άκρο σε άκρο.
- Αυτόματη αναδρομολόγηση των κυκλωμάτων που διακόπτονται.
- Γραφική και υπό μορφή πινάκων απεικόνιση της πληροφορίας της σχετικής με το σχηματισμό του Δικτύου.
- Σχεδίαση και αποθήκευση χαρτών ανωμαλιών/ανάκτησης για φόρτωση στο Δίκτυο σε περίπτωση διακοπής.

**β) Ανασχηματισμός Δικτύου ελεγχόμενος από το συνδρομητή.**

Το NMS θα πρέπει να έχει τη δυνατότητα του επιμερισμού του Δικτύου σε ένα αριθμό υποδικτύων (ο αριθμός αυτός θα πρέπει να δηλωθεί), που θα βρίσκονται κάτω από τον έλεγχο των συνδρομητών. Ιδιαίτερα το NMS θα πρέπει να έχει τα παρακάτω χαρακτηριστικά:

- Επιμερισμό του εύρους διεύλεσης του Δικτύου σε όρια των 64Kbps.
- Γραφική απεικόνιση του σχηματισμού του Δικτύου, καθώς και πληροφοριών για την κατάσταση των υποδικτύων των συνδρομητών.
- Δυνατότητα σχηματισμού και διασύνδεσης κυκλωμάτων 64Kbps και Nx64Kbps, μεταξύ δύο ή περισσοτέρων σημείων του υποδικτύου.

**γ) Διαχείριση τράπεζας δεδομένων.**

Το NMS θα πρέπει να έχει τις ακόλουθες δυνατότητες διαχείρισης τράπεζας δεδομένων:

- Διατήρηση δεδομένων για συνδρομητές, κόμβους, ζεύξεις, συνδέσεις σε αναμονή, σηματοδοσίες ανωμαλιών και επιδόσεων, καθώς και διατήρηση των ειδικών βάσεων δεδομένων του NMS.
- Αποθήκευση των βάσεων δεδομένων των κόμβων σε διατάξεις μαζικής αποθήκευσης για επαναφόρτωσή τους.
- Επαναφόρτωση των βάσεων δεδομένων σε κάθε κόμβο.

- Απεικόνιση του εξοπλισμού και του σχηματισμού του συστήματος συμπεριλαμβανομένων και των ανταλλακτικών.
- Αυτόματη ή χειροκίνητη επισκόπηση (auditing) των βάσεων δεδομένων των κόμβων και ανάλυση τυχών διαφορών.

**δ) Συντήρηση και επίβλεψη.**

Το NMS θα πρέπει να έχει τις ακόλουθες δυνατότητες συντήρησης και επίβλεψης:

- Επεξεργασία των ενδείξεων συναγερμού για τον εντοπισμό ανωμαλιών μέχρι στάθμης μονάδας.
- Λειτουργίες αυτοδιάγνωσης.
- Εκτύπωση αναφορών σχετικών με σηματοδοσίες βλαβών, διαγνωστικών μηχανημάτων και στατιστικής.
- Έκδοση αναφορών, το είδος και η φόρμα των οποίων να καθορίζεται από τον χρήστη.
- Γραφική απεικόνιση όλων των επιπέδων του Δικτύου (μέχρι μονάδος) και των καταστάσεων συναγερμού.
- Επέκταση μέχρι το NMS των ευκολιών για δοκιμές που έχει κάθε κόμβος και δυνατότητα να ενεργοποιήσει βρόχους, σε κάθε σημείο του κυκλώματος.
- Παραχώρηση της δυνατότητας συντήρησης του συστήματος σε εξωτερικές συσκευές.

**ε) Παρακολούθηση των επιδόσεων.**

Το NMS θα πρέπει να έχει τις ακόλουθες δυνατότητες παρακολούθησης των επιδόσεων του συστήματος:

- Απεικόνιση των μετρήσεων επιδόσεων.
- Απεικόνιση της ποιότητας του σήματος.
- Δημιουργία στατιστικών για βλάβες του Δικτύου.

**ζ) Διαχείριση ασφάλειας του Δικτύου.**

Το NMS θα πρέπει να έχει τις ακόλουθες δυνατότητες για ασφάλεια του Δικτύου:

- Έλεγχος από το NMS της πρόσβασης του χρήστη.
- Το NMS θα πρέπει να υποστηρίζει διάφορες λειτουργίες σχετικά με τον έλεγχο του Δικτύου, δυνατότητες εκχώρησης μέρους της διαχείρισης σε συνδρομητές, συντήρηση κτλ.

Ειδικότερα:

- (1) Ο υπεύθυνος ελέγχου του Δικτύου θα πρέπει να έχει πρόσβαση σε όλες τις δυνατότητες του Δικτύου για να διαμορφώνει, παρακολουθεί και συντηρεί το όλο Δίκτυο και τα υποδίκτυα.
- (2) Ο χειριστής του υποδικτύου θα πρέπει να έχει πρόσβαση σε ένα σύνολο δυνατοτήτων του NMS μέσα στα όρια 64Kbps του υποδικτύου, όπως καθορίζονται από τον υπεύθυνο ελέγχου του Δικτύου.
- (3) Ο χειριστής του NMS θα πρέπει να είναι υπεύθυνος για τη συντήρηση του Hardware και του λειτουργικού συστήματος του NMS, εξαιρουμένων της δυνατότητας αναδρομολόγησης και λήψεως διαγνωστικών για το Δίκτυο.
- (4) Ο τοπικός χειριστής κάθε κόμβου θα είναι υπεύθυνος για την συντήρηση και τον έλεγχο του Hardware του κόμβου και των κυκλωμάτων της περιοχής του.

Ολοι οι παραπάνω χρήστες θα αναγνωρίζονται κατά τη χρονική στιγμή της εισόδου τους στο σύστημα (login time) με ειδικές ταυτότητες (IDs) και λέξεις προσπέλασης (passwords) που τους εκχωρούνται από τον υπεύθυνο ελέγχου του Δικτύου.

Τρόποι πρόσβασης στο σύστημα:

Το NMS θα πρέπει να υποστηρίζει τους ακόλουθους τρόπους πρόσβασης στο σύστημα:

- (1) Με κλήση: Μέσω του επιλεγόμενου τηλεφωνικού δικτύου.
- (2) Με οπισθόδοτη κλήση: Οπως και η προηγούμενη αλλά και με επιπλέον κλήση από το NMS προς το τερματικό του χρήστη.
- (3) Με αποκλειστική γραμμή.
- (4) Εναλλακτικά μέσω του Δημοσίου Δικτύου Μεταγωγής Πακέτου (HELLASPAC).

Εξασφάλιση της ακεραιότητας των βάσεων δεδομένων.

Τέλος, το NMS θα πρέπει να καταγραφεί και να διατηρεί για ένα μήνα τουλάχιστον πληροφορίες για τη διαθεσιμότητα των γραμμών και της χρησιμοποίησής τους από τους συνδρομητές, για σκοπούς χρέωσης.

## 6.6. ISDN - Διαχείριση του ISDN

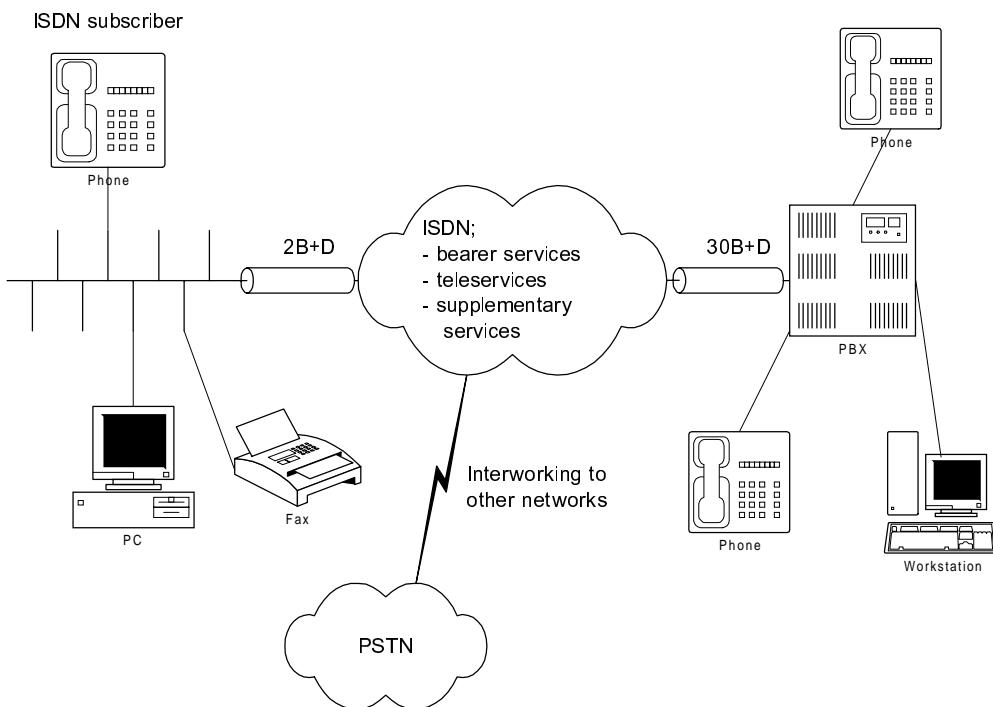
### A. Εισαγωγή στο ISDN

Μετά το 1994 θα μπορέσουν να προσφερθούν και στην Ελλάδα υπηρεσίες ψηφιακής επικοινωνίας (μετάδοση φωνής, δεδομένων, εικόνας, κ.ά.) μεταξύ συνδρομητών με τη βοήθεια δημοσίου ψηφιακού δικτύου ενοποιημένων υπηρεσιών ISDN (Integrated Services Digital Network). Το ISDN θα υλοποιηθεί με εγκατάσταση κατάλληλου υλικού και κυρίως λογισμικού σε ψηφιακά τηλεφωνικά κέντρα του ΟΤΕ.

Το Ψηφιακό Δίκτυο Ενοποιημένων Υπηρεσιών (Integrated Services Digital Network - ISDN) αποτελεί μία εξέλιξη των ψηφιακών τηλεφωνικών δικτύων, όπου η ψηφιακή πρόσβαση επεκτείνεται μέχρι τον τελικό χρήστη. Για την επίτευξη του παραπάνω στόχου, το ISDN απαιτεί την ύπαρξη ψηφιακής μεταγωγής και τη βαθμιαία ψηφιοποίησή του τηλεφωνικού δικτύου μέχρι το συνδρομητικό βρόχο.

Η βασική πρόσβαση στο ISDN (Basic Rate Access - BRA) περιλαμβάνει δύο κανάλια στα 64 Kbps (B-Channels) και ένα στα 16 Kbps (D-Channel) το οποίο χρησιμοποιείται κατά κύριο λόγο για σηματοδοσία. Έτσι, ο συνολικός ρυθμός που είναι διαθέσιμος στον χρήστη μπορεί να φθάσει τα 144 Kbps (2B+D). Για μεγάλους χρήστες προβλέπεται το Primary Rate Access (PRA) ISDN στα 2 Mbps με 30B+D κανάλια, όλα στα 64 Kbps. Το PRA-ISDN έχει εφαρμογή για σύνδεση μεγάλων συνδρομητικών κέντρων ISDN/PABX. Σε πρώτη φάση οι Τηλεπικοινωνιακοί Φορείς της Ευρώπης προσφέρουν κυρίως δημόσιο δίκτυο BRA-ISDN.

Το ISDN δίνει την ευκαιρία να βελτιωθεί η ποιότητα των υπαρχουσών τηλεφωνικών υπηρεσιών προσφέροντας ταυτόχρονα μετάδοση δεδομένων και άλλες νέες υπηρεσίες μέσω ενός ενιαίου δικτύου, βασισμένου σε σύγχρονους ψηφιακούς διαύλους των 64 Kbps.



#### Σχήμα 6.4 - Βασικές έννοιες στο ISDN

Συγκεντρωτικά, τα πλεονεκτήματα που προσφέρει το ISDN είναι τα εξής:

- Επιλεγόμενο ψηφιακό δίκτυο υψηλής ποιότητας ενοποιημένων υπηρεσιών (μεταφορά φωνής, εικόνων και δεδομένων).
- Μοναδικό σημείο πρόσβασης στο δίκτυο για όλες τις τερματικές συσκευές ενός χρήστη.
- Ρυθμούς μεταφοράς δεδομένων 64 Kbps σε επιλεγόμενο δημόσιο δίκτυο.
- Υπηρεσίες ευφυούς δικτύου από άκρο σε άκρο (σηματοδοσία κοινού διαύλου μέχρι τον ακραίο χρήστη σε συνεργασία με χρήση βάσεων δεδομένων)

Η Ευρωπαϊκή εμπειρία έχει δείξει πως το δημόσιο BRA-ISDN έχει δύο βασικές χρήσεις:

- Επιλεγόμενο δίκτυο μεταφοράς δεδομένων στα 64 Kbps (αντί modem).
- Εσωτερική μεταγωγή σε μικρούς χρήστες μέχρι 8 συσκευών (τηλεφωνικών, δεδομένων, FAX GIII/GIV, videotext) με ευθύνη του δημόσιου φορέα (OTE).

Στον Ελληνικό χώρο έχουν ήδη συντελεστεί δοκιμές χρήσης BRA-ISDN για μετάδοση δεδομένων στα 64-128 Kbps [ΧΙΩΤ93]. Επίσης δοκιμές πραγματοποιήθηκαν στα πλαίσια του "EURIE '93", 14-16 Δεκεμβρίου 1993, με επίδειξη διεθνών συνδέσεων ISDN σε κτίριο του OTE (Σταδίου 15) για τα Πανευρωπαϊκά Εγκαίνια του EURO-ISDN.

## B. Θέματα Διαχείρισης

Προμηθευτές προϊόντων ISDN έχουν ανακοινώσει υποστήριξη του πρωτοκόλλου SNMP για τη διαχείριση των προϊόντων τους και ορισμό κατάλληλης Βάσης Πληροφορίας Διαχείρισης για ISDN (**ISDN-MIB**). Τα διαχειριζόμενα αντικείμενα τα οποία περιγράφουν στους ορισμούς των MIBs προσανατολίζονται προς τη διαχείριση δικτύων δεδομένων, τα οποία χρησιμοποιούν το ISDN σε φυσικό επίπεδο για τη διασύνδεση τοπικών δικτύων υπολογιστών - LANs (Novell, TCP/IP, AppleTalk, κλπ.).

Οι ορισμοί περιορίζονται σε πληροφορίες για:

1. Κάρτες PC διασύνδεσης (**ISDN-Controllers**) στο δημόσιο ή ιδιωτικό δίκτυο ISDN (π.χ. BoardName, IOAddress, NumberOfPorts, SerialNumber, DChannelProtocol, IncomingConnectionsDenied, κλπ.),
2. Διαχείριση των καναλιών σύγχρονης επικοινωνίας B-Channels, που χρησιμοποιούνται στις συνδέσεις (π.χ. PortNumber, IncomingConnections, OutgoingConnections, TotalCharging, ConnectionTime, κλπ.),
3. Συνδέσεις (π.χ. DestinationAddress, BChannelsUsed, LogicalUpTime, InactivityTimer, IPX Packets Sent, IPPacketsSent, IPX PacketsReceived, IPPacketsReceived).

Η ISDN-MIB δεν είναι επίσημα νιοθετημένη σαν RFC και διερευνόνται διάφορες κατευθύνσεις τόσο από τη βιομηχανία όσο και από την ακαδημαϊκή ερευνητική κοινότητα (π.χ. RACE).

## 6.7. Ασκήσεις

- [1]. Αναφέρατε διαφορές μεταξύ ενός ευφυούς πολυπλέκτη (T1) και ενός μεταγωγέα κυκλώματος πολύπλεξης χρόνου (π.χ. ψηφιακού συνδρομητικού τηλεφωνικού κέντρου - PBX).
- [2]. Αναφέρατε και συγκρίνατε τρόπους μετάδοσης πληροφοριών διαχείρισης στοιχείων δικτύου. Εξηγήστε τους όρους τεστ βρόγχου (loop-back) και fall-back με παραδείγματα από διαχείριση (στο φυσικό επίπεδο) διαμορφωτών (modems).

- [3]. Να περιγράψετε σειρά από δοκιμές σε επικοινωνιακές συνδέσεις που χρησιμοποιούν modems για τον εντοπισμό προβλημάτων. Τι άλλα εργαλεία θα βοηθούσαν τον τεχνικό στον εντοπισμό των βλαβών;

## 6.7. Βιβλιογραφία

- [ΑΛΕΞ92] Α.Αλεξόπουλος, Γ.Λαγογιάννης, Τηλεπικοινωνίες και Δίκτυα Υπολογιστών, Δεύτερη Έκδοση, Αθήνα 1992.
- [CAMA91] Cable Management Systems: Technology Overview, DATAPRO, Datapro International Network Management, Network Management Tools, IJ06-998-201, September 1991.
- [HELD92] Held G., *Network Management, Techniques, Tools and Systems*, John Wiley, 1992.
- [HELL90] Τεχνικές Προδιαγραφές για το Ψηφιακό Δίκτυο Μετάδοσης Δεδομένων "HELLASCOM", HCM/31000/A'/1990, Αθήνα, Μάιος 1990.
- [LAN\_91] The LAN Diagnostic Process, DATAPRO, Datapro International Network Management, Network Management Tools, IJ06-002-101, September 1991.
- [MAGL94] B. Maglaris, T. Karounos, T. Chiotis, "Plans for ISDN infrastructure in Greece," Applications of Advanced Communications in Shipping, Seminar, Evgenidion Foundation, Athens, 7 June 1994.
- [RERF91] Performance Modelling, DATAPRO, Datapro International Network Management, WAN Management, IJ03-003-101, September 1991.
- [ROSE91] Rose T.M., The Simple Book: An Introduction to Management of TCP/IP-based Internets, Prentice Hall International Inc., Englewood Cliffs, New Jersey, 1991.
- [SITI92] The Simple Times, The Bi-Monthly Newsletter of SNMP Technology, Comment and Events, Vol.1, No.1-5, March-December 1992.
- [SITI93] The Simple Times, The Bi-Monthly Newsletter of SNMP Technology, Comment and Events, Vol.2, No.1, January-February 1993.
- [WAN\_91] WAN Management Systems: Technology Overview, DATAPRO, Datapro International Network Management, WAN Management, IJ03-010-071, September 1991.
- [X.2591] Managing X.25 Packet Switched Networks, DATAPRO, Datapro International Network Management, WAN Management, IJ03-002-101, September 1991.
- [ΧΙΩΤ93] Τρύφωνας Χιώτης, Θεόδωρος Καρούνος, "Διασύνδεση δικτύων H/Y μέσω ISDN στην Ελλάδα," Ενημερωτικό Δελτίο της ΕΠΥ, τ. 50, Σεπτέμβριος - Οκτώβριος '93.

- [ΧΙΩΤ94] Τ. Χιώτης, Θ. Καρούνος, Β. Μάγκλαρης, "Δίκτυα: Εξελίξεις και τρέχουσα πραγματικότητα στην Ελλάδα," ΤΕΕ-ΕΛΟΤ/ΤΕ22, Ημερίδα: OSI και εφαρμογή στις ελληνικές βιβλιοθήκες και Υπηρεσίες Τεκμηρίωσης και Πληροφόρησης, Αθήνα, 17 Ιανουαρίου 1994.

# Κεφάλαιο 7

## 7. Γέφυρες και Δρομολογητές

### Περιεχόμενα του Κεφαλαίου 7

- 7.0. Εισαγωγή
- 7.1. Τοπικά δίκτυα
- 7.2. Τα πρότυπα 802.X της IEEE
  - 7.2.1. Το επίπεδο ελέγχου προσπέλασης του μέσου (MAC)
  - 7.2.2. Το επίπεδο ελέγχου λογικών συνδέσεων (LLC)
- 7.3. Η ανάγκη για διασυνδεδεμένα τοπικά δίκτυα
- 7.4. Γέφυρες. Δρομολόγηση στο επίπεδο MAC
- 7.5. Διαφανείς γέφυρες
  - 7.5.1. Η λειτουργία ενημέρωσης της γέφυρας
  - 7.5.2. Ο αλγόριθμος επικαλύπτοντος δένδρου (Spanning Tree Algorithm)
  - 7.5.3. Συντονισμός της τοπολογίας
  - 7.5.4. Παράδειγμα
  - 7.5.5. Απομακρυσμένες γέφυρες (Remote Bridges)
- 7.6. Γέφυρες δρομολόγησης πηγής (Source Routing Bridges)
- 7.7. Σύγκριση μεταξύ διαφανών γεφυρών και γεφυρών δρομολόγησης πηγής
- 7.8. Η γεφύρωση διαφορετικών δικτύων
- 7.9. Θέματα διαχείρισης γεφυρών
- 7.10. Δρομολογητές (Routers), σύγκριση με τις γέφυρες
- 7.11. Ασκήσεις
- 7.12. Βιβλιογραφία

### 7.0. Εισαγωγή

Η χρησιμοποίηση **γεφυρών και δρομολογητών** (*bridges - routers*) για τη διασύνδεση τοπικών δικτύων υπολογιστών (Local Area Networks, LANs) είναι μια κοινή πρακτική σήμερα, που προσφέρει αυξημένες δυνατότητες για υπολογιστική επεξεργασία. Τέτοιου είδους μηχανήματα μπορούν να αυξήσουν τις υπολογιστικές επιδόσεις, με τη διαίρεση ενός μεγάλου τοπικού δικτύου σε πολλά μικρότερα και συνεπώς ευκολότερα διαχειριζόμενα τοπικά δίκτυα. Επίσης, χρησιμοποιώντας κατάλληλα γέφυρες και δρομολογητές, είναι δυνατή η διασύνδεση τοπικών δικτύων σε διεθνές επίπεδο, οπότε το όφελος είναι ακόμα μεγαλύτερο.

Στις παραπάνω τοπολογίες το φορτίο που δημιουργείται σε κάποιο κόμβο μπορεί να μεταφερθεί σε οποιοδήποτε άλλο σημείο του διασυνδεδεμένου δικτύου. Το γεγονός αυτό δεν θα πρέπει να μειώνει την απόδοση του γεφυρωμένου δικτύου. Αυτό είναι

υπευθυνότητα της διαχείρισης των διασυνδεδεμένων τοπικών δικτύων, η οποία θα πρέπει να περιορίζει το φορτίο που θα περνάει από δίκτυο σε δίκτυο, στο απόλυτα απαραίτητο. Με την διαχείριση των γεφυρωμένων δικτύων θα ασχοληθούμε σ' αυτό το κεφάλαιο.

## 7.1. Τοπικά δίκτυα

Η ανάπτυξη και διάδοση τοπικών δικτύων υπολογιστών προήλθε από την ανάγκη αλληλοσύνδεσης μεγάλου αριθμού υπολογιστών συγκεντρωμένων σε μια μικρή γεωγραφική περιοχή. Τα τοπικά δίκτυα διακρίνονται συνήθως από ιδιότητες, όπως οι παρακάτω:

- Την ύπαρξη πολλών συστημάτων συνδεδεμένων σε ένα κοινό μέσο μεταφοράς.
- Την υψηλή χωρητικότητα του μέσου μεταφοράς (η οποία μπορεί να χρησιμοποιηθεί από όλους τους σταθμούς).
- Τις "χαμηλές" καθυστερήσεις.
- Τους "μικρούς" ρυθμούς λαθών κατά την μετάδοση της πληροφορίας.
- Την δυνατότητα broadcast μετάδοσης της πληροφορίας (από έναν σε πολλούς σταθμούς).
- Τον περιορισμένο γεωγραφικό χώρο τοποθέτησης του δικτύου (μερικά χιλιόμετρα).
- Τον περιορισμένο αριθμό σταθμών (μερικές εκατοντάδες).
- Την ομότιμη σχέση μεταξύ γειτονικών σταθμών (αντίθετα με άλλα συστήματα αφέντη / σκλάβου - master / slave).
- Τέλος τα τοπικά δίκτυα υπολογιστών υπακούουν σε νόμους σχετικούς με την ιδιωτική περιουσία και είναι ανεξάρτητα των PTT (Post, Telegraph, Telephone) της κάθε χώρας (π.χ. O.T.E.).

Η χρησιμοποίηση ενός κοινού μέσου μεταφοράς από όλους τους σταθμούς εργασίας πραγματοποιείται για λόγους οικονομίας μια και η διασύνδεση κάθε σταθμού από N με όλους τους υπόλοιπους θα απαιτούσε  $N(N-1)/2$  συνδέσμους. Η χρησιμοποίηση ενός κοινού μέσου μεταφοράς από πολλούς τερματικούς σταθμούς εισάγει την έννοια του ανταγωνισμού. Έτσι έχουμε τις απαιτήσεις που πρέπει να ικανοποιούν οι μηχανισμοί διευθέτησης της χρήσης του μέσου και οι οποίοι είναι οι παρακάτω:

- Κάθε σταθμός μπορεί να χρησιμοποιεί ένα ίσο μέρος της χωρητικότητας (εφόσον όλοι οι σταθμοί είναι ισότιμοι).
- Κάθε σταθμός πρέπει να έχει τη δυνατότητα πρόσβασης στο μέσο μεταφοράς μέσα σε ένα σχετικά μικρό χρονικό διάστημα, από την στιγμή που θα την ζητήσει.
- Η σπατάλη χωρητικότητας εξαιτίας των παραπάνω δύο μηχανισμών θα πρέπει να είναι μηδαμινή.

Τα δύο πιο δημοφιλή σχήματα ανταγωνισμού για την κατάληψη του μέσου μεταφοράς είναι αυτό με τις συγκρούσεις και αυτό με το κουπόνι. Στο πρώτο σχήμα κάθε σταθμός

μεταδίδει όποτε το θελήσει. Αν δύο σταθμοί μεταδώσουν όμως την ίδια χρονική στιγμή έχουμε σύγκρουση, η μετάδοσή τους διακόπτεται και οι σταθμοί επαναμεταδίδουν αργότερα σε τυχαίες χρονικές στιγμές. Με κατάλληλα πιθανοθεωρητικά σχήματα μπορεί να μειωθεί σημαντικά η πιθανότητα συγκρούσεων.

Στα σχήματα με κουπόνι (για παράδειγμα σε δακτύλιο με κουπόνι) μια προκαθορισμένη σειρά από bits μεταφέρεται από σταθμό σε σταθμό. Ένας σταθμός εφόσον θέλει να μεταδώσει απούρει το κουπόνι από το δίκτυο και μεταδίδει τα δεδομένα του για κάποιο επιτρεπτό χρονικό διάστημα.

Μια πολύ σημαντική κατηγορία τοπικών δικτύων είναι τα πρωτόκολλα 802.X της IEEE και τα οποία θα εξετάσουμε παρακάτω.

## 7.2. Τα πρότυπα 802.X της IEEE

Το Φεβρουάριο του 1980 ιδρύθηκε η επιτροπή IEEE 802 από την IEEE Computer Society για τη δημιουργία προτύπων για τοπικά και μητροπολιτικά δίκτυα (Local and Metropolitan Area Networks, LANs). Η οικογένεια προτύπων IEEE 802 ασχολείται κυρίως με το φυσικό επίπεδο και το επίπεδο σύνδεσης δεδομένων του μοντέλου αναφοράς ISO/OSI. Ένας από τους βασικούς άξονες εργασίας της επιτροπής ήταν η ανάπτυξη των προτύπων των τοπικών δικτύων σύμφωνα με το μοντέλο αναφοράς πρωτοκόλλων OSI. Βέβαια δεν ήταν δυνατό να αναπτυχθεί μια ενιαία τεχνολογία που να καλύπτει όλες τις απαιτήσεις στον χώρο των τοπικών δικτύων και έτσι η όλη εργασία μοιράστηκε σε διάφορες υποεπιτροπές. Τα αντικείμενα της κάθε υποεπιτροπής είναι τα παρακάτω:

- **802.1 - Overview, Systems Management and Internetworking.** Η υποεπιτροπή αυτή δεν παρήγαγε πρότυπα, αλλά έδωσε μια περιληψη της συνολικής εργασίας της επιτροπής 802 και καθόρισε το μοντέλο αναφοράς των τοπικών δικτύων υπολογιστών. Επίσης, ασχολήθηκε με θέματα σχετικά με τη μορφή των διευθύνσεων, τη διαχείριση των δικτύων, και τη διασύνδεσή τους.
- **802.2 - Logical Link Control (LLC).** Έργο της υποεπιτροπής αυτής ήταν η περιγραφή του υποεπιτέδου ελέγχου λογικής γραμμής (Logical Link Control, LLC), στο επίπεδο σύνδεσης δεδομένων, που χρησιμοποιείται σε όλα τα τοπικά δίκτυα που καθορίζονται από την IEEE. Έγινε περιγραφή όλων των υπηρεσιών και λειτουργιών, που απαιτούνται για την παροχή αξιόπιστης επικοινωνιακής γραμμής μεταξύ δύο κόμβων.
- **802.3 - CSMA/CD Medium Access Method and Physical Layers.** Περιγράφηκε το φυσικό επίπεδο και ο έλεγχος πρόσβασης στο μέσο (MAC) για ένα δίκτυο που χρησιμοποιεί μετάδοση βασικής ζώνης, τοπολογίας αρτηρίας και τρόπου προσπέλασης του μέσου Carrier Sense Multiple Access with Collision Detection (CSMA/CD).
- **802.4 - Token-Passing Bus Medium Access Method and Physical Layers.** Περιγράφηκε το φυσικό επίπεδο και ο έλεγχος πρόσβασης στο μέσο (MAC) για ένα δίκτυο που χρησιμοποιεί μετάδοση βασικής ζώνης, τοπολογίας αρτηρίας και τρόπου προσπέλασης του μέσου με κουπόνι.
- **802.5 - Token-Passing Ring Medium Access Method and Physical Layers.** Περιγράφηκε το φυσικό επίπεδο και ο έλεγχος πρόσβασης στο μέσο (MAC) για ένα δίκτυο που χρησιμοποιεί μετάδοση βασικής ζώνης, τοπολογίας δακτυλίου και τρόπου προσπέλασης του μέσου με κουπόνι.

- **802.6 - Metropolitan Area Network Access Method and Physical Layers.** Καθόρισε τις προδιαγραφές για τα Μητροπολιτικά Δίκτυα Υπολογιστών (MAN).
- **802.7 - Broadband Technical Advisory and Physical Layer Issues.** Δημιουργήθηκε για να παρέχει τεχνικές συμβουλές στις υπόλοιπες υποεπιτροπές, σε θέματα σχετικά με την χρήση εκπομπής ευρείας ζώνης.
- **802.8 - Fibre Optic Technical Advisory and Physical Layer Issues.** Δημιουργήθηκε για να μελετήσει τρόπους με τους οποίους η τεχνολογία των οπτικών ινών θα μπορούσε να υποστηρίξει την εργασία των άλλων υποεπιτροπών.
- **802.9 - Integrated Voice and Data Medium Access Control and Physical Layers**
- **802.10 - Security and Privacy Access Method and Physical Layer Specifications**

### 7.2.1. Το επίπεδο ελέγχου προσπέλασης του μέσου (MAC)

Το **πρωτόκολλο ελέγχου προσπέλασης του μέσου (Media Access Control, MAC)** ρυθμίζει τον τρόπο διευθέτησης του μέσου μεταφοράς μεταξύ των διαφόρων σταθμών που χρησιμοποιούν το τοπικό δίκτυο.

Οι πιο συνηθισμένες μέθοδοι διευθέτησης του μέσου που χρησιμοποιούνται, είναι οι παρακάτω:

#### 1. CSMA/CD

Η μέθοδος αυτή προσπέλασης του μέσου χρησιμοποιείται κυρίως με δίκτυα τοπολογίας αρτηρίας (bus networks). Σύμφωνα με την τοπολογία αυτή όλοι οι τερματικοί σταθμοί είναι απευθείας συνδεδεμένοι στο ίδιο καλώδιο. Όλα τα δεδομένα, που μεταδίδονται από κάποιο σταθμό εργασίας ενθλακώνονται σε ένα πλαίσιο, στο οποίο τοποθετείται και η φυσική διεύθυνση του σταθμού προορισμού, και το πλαίσιο στην συνέχεια μεταδίδεται broadcast στο κοινό μέσο μεταφοράς. Όλοι οι άλλοι σταθμοί μπορούν να αντιληφθούν ότι κάποιος σταθμός μεταδίδει. Αν επιπλέον αναγγινώσουν στο πλαίσιο τη διεύθυνσή τους, τότε συνεχίζουν να διαβάζουν και την υπόλοιπη πληροφορία που υπάρχει στο πλαίσιο. Στο πλαίσιο περιλαμβάνεται και η διεύθυνση του κόμβου πηγή, ώστε ο προορισμός να μπορεί να απαντήσει στο μήνυμα. Με τον τρόπο αυτό διευθέτησης του μέσου είναι δυνατόν δύο σταθμοί να προσπαθήσουν περίπου την ίδια στιγμή να μεταδόσουν στο δίκτυο, οπότε η πληροφορία και των δύο θα καταστραφεί.

Για να ελαττωθεί η πιθανότητα αυτή το CSMA/CD καθορίζει ότι πριν κάποιος σταθμός επιχειρήσει να μεταδόσει στο μέσο, ελέγχει αν κάποιος άλλος σταθμός μεταδίδει (if a carrier signal is sensed). Στην περίπτωση αυτή αναβάλλει την μετάδοσή του, μέχρι την ολοκλήρωση της ήδη εξελισσόμενης μετάδοσης. Βέβαια, και πάλι υπάρχει η πιθανότητα, δύο σταθμοί ταυτόχρονα να διαπιστώσουν την αδράνεια του μέσου μεταφοράς και να επιχειρήσουν να μεταδόσουν. Στην περίπτωση αυτή η σύγκρουση είναι αναπόφευκτη, και το ουσιαστικό είναι να γίνει αντιληπτή όσο το δυνατόν γρηγορότερα. Αυτό επιτυγχάνεται με έλεγχο από κάθε σταθμό του σήματος στο καλώδιο και σύγκριση αυτού με το σήμα που μεταδίδει. Αν τα σήματα αυτά είναι διαφορετικά τότε η σύγκρουση έχει ανιχνευθεί (collision detected). Στην περίπτωση σύγκρουσης η μετάδοση

επαναλαμβάνεται μετά από ένα κατάλληλα επιλεγμένο τυχαίο χρονικό διάστημα, ώστε να μην ξανασυμβεί σύγκρουση μεταξύ των δύο σταθμών.

## 2. Token

Μια δεύτερη μέθοδος ελέγχου της πρόσβασης σε ένα μέσο μεταφοράς είναι η χρησιμοποίηση ενός κουπονιού ελέγχου της πρόσβασης (control permission token). Το κουπόνι αυτό περνάει από σταθμό σε σταθμό σύμφωνα με κάποιο σύνολο από κανόνες, που αποδέχονται όλοι οι σταθμοί. Κάποιος σταθμός μπορεί να μεταδώσει ένα πλαισιο, μόνο στην περίπτωση που έχει στην κατοχή του το κουπόνι, και μόλις ολοκληρώσει την μετάδοση του πλαισίου ελευθερώνει και πάλι το κουπόνι, περνώντας το στον επόμενο σταθμό. Στην τεχνική αυτή εισάγεται η έννοια μιας λογικής σειράς των σταθμών, ώστε να γνωρίζει ο καθένας τον επόμενό του. Κατά την αρχικοποίηση των συστημάτων εγκαθίσταται μια λογική συνδεσμολογία, και δημιουργείται ένα κουπόνι. Συναρτήσεις παρακολούθησης στους σταθμούς, που είναι συνδεδεμένοι στο κοινό μέσο παρέχουν υπηρεσίες αρχικοποίησης και επαναφοράς σε περίπτωση σφάλματος της λογικής συνδεσμολογίας. Αν και οι συναρτήσεις επαναλαμβάνονται από όλους τους σταθμούς, κάθε χρονική στιγμή ένας σταθμός είναι υπεύθυνος για το δίκτυο. Το φυσικό μέσο δεν είναι απαραίτητο να έχει την τοπολογία δακτυλίου. Η τεχνική ελέγχου προσπέλασης του μέσου με κουπόνι χρησιμοποιείται και σε δίκτυα τοπολογίας αρτηρίας.

## 3. Slotted Ring

Η τελευταία μέθοδος ελέγχου της πρόσβασης, που θα εξετάσουμε, χρησιμοποιείται μόνο σε δίκτυα τοπολογίας δακτυλίου. Ο δακτύλιος αρχικοποιείται έτσι ώστε να περιέχει ένα σταθερό μήκος από bits, τα οποία εισάγονται από κάποιο ειδικό κόμβο (monitor). Ο συρμός αυτός από bits κυκλοφορεί συνέχεια μέσα στο δίκτυο από σταθμό σε σταθμό, ενώ ο κόμβος monitor ελέγχει, τον αριθμό των bits, ο οποίος πρέπει να παραμένει σταθερός. Ο συρμός των bits χωρίζεται σε ένα σταθερό αριθμό από slots, τα οποία έχουν τη δυνατότητα μεταφοράς ενός πλαισίου. Στην αρχή κάθε slot υπάρχει κάποιο bit, το οποίο δείχνει αν το slot μεταφέρει κάποιο πλαισιο ή όχι. Ένας σταθμός που θέλει να μεταδώσει, βρίσκει ένα ελεύθερο slot στο οποίο εισάγει το πλαισιο. Στην συνέχεια κάθε σταθμός ελέγχει το γεμάτο πλαισιο προκειμένου να αναγνωρίσει την διεύθυνση προορισμού του πλαισίου. Στην περίπτωση, που είναι η δική του φυσική διεύθυνση, διαβάζει το πλαισιο. Μόλις ο σταθμός που μετάδοσε αρχικά το πλαισιο ξαναδεί το πλαισιο του στο δίκτυο, το αποσύρει, ελευθερώνοντας το slot.

### 7.2.2. Το επίπεδο ελέγχου λογικών συνδέσεων (LLC)

Το πρότυπο IEEE 802.2 περιγράφει το πρωτόκολλο και τις λειτουργίες του **Logical Link Control (LLC)** υποεπίπεδου των IEEE προτύπων για τοπικά δίκτυα υπολογιστών. Το LLC αποτελεί το υψηλότερο υποεπίπεδο στο επιπέδο σύνδεσης δεδομένων του OSI μοντέλου αναφοράς πρωτοκόλλων, και είναι το ίδιο για όλα τα υποεπίπεδα ελέγχου προσπέλασης του μέσου, που ορίζονται από τα IEEE 802 πρότυπα (802.3-802.6). Παρέχει ένα καλά καθορισμένο τρόπο μεταφοράς της πληροφορίας μεταξύ του οποιοδήποτε επιπέδου δικτύου και του οποιοδήποτε υποεπίπεδου ελέγχου προσπέλασης του μέσου, όπως επίσης ένα τρόπο διαχωρισμού συνδέσεων με σύνδεση, από συνδέσεις χωρίς σύνδεση.

Πιο συγκεκριμένα το υποεπίπεδο LLC εκτελεί τις παρακάτω λειτουργίες:

- Ανταλλαγή πληροφοριών ελέγχου.

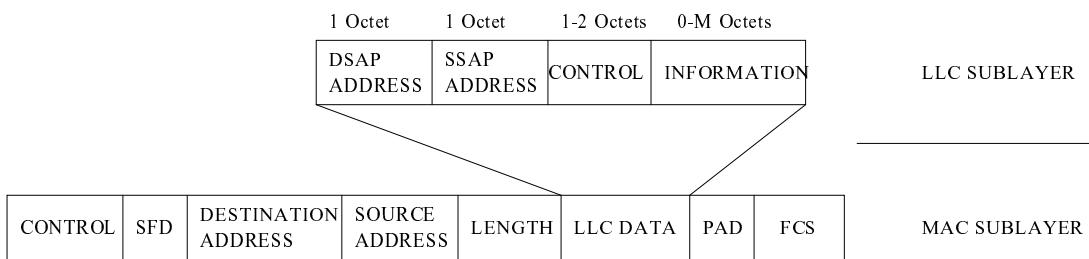
- Οργάνωση της ροής των δεδομένων.
- Ερμηνεία των λαμβανόμενων εντολών και δημιουργία και αποστολή της κατάλληλης απόκρισης.
- Ανίχνευση λαθών και ανάκτηση από συνθήκες λάθους.

Είναι διαθέσιμες τρεις μέθοδοι για την ανταλλαγή πληροφοριών μεταξύ ομότιμων LLC οντοτήτων:

- (1) **Υπηρεσία χωρίς επιβεβαίωση και σύνδεση.** Η υπηρεσία αυτή μας θυμίζει το IP πρωτόκολλο. Δεν δημιουργείται σύνδεση και το η LLC οντότητα μπορεί να στέλνει PDUs, χωρίς να περιμένει κάποια επιβεβαίωση.
- (2) **Υπηρεσία με σύνδεση.** Η υπηρεσία αυτή παρέχει μια μορφή νοητού κυκλώματος μεταξύ δύο SAPs. Με αυτή την υπηρεσία είναι δυνατός ο έλεγχος για λάθη, ο έλεγχος ροής και η άφιξη της πληροφορίας στον προορισμό με τη σειρά που μεταδόθηκε.
- (3) **Υπηρεσία με επιβεβαίωση και χωρίς σύνδεση.** μοιάζει με την πρώτη υπηρεσία, με τη διαφορά ότι δεν δημιουργείται κάποιο νοητό κύκλωμα.

Η σχέση μεταξύ των LLC και MAC υποεπιπέδων και του επιπέδου σύνδεσης δεδομένων φαίνεται στο Σχήμα 7.1. Το LLC χρησιμοποιεί τις υπηρεσίες που παρέχονται από κάποιο MAC υποεπίπεδο και παρέχει υπηρεσίες σύνδεσης δεδομένων στο επίπεδο δικτύου.

Η δομή ενός LLC PDU φαίνεται επίσης στο Σχήμα 7.1. Σε κάθε SDU που παραλαμβάνει από το επίπεδο δικτύου, το LLC προσθέτει τις διευθύνσεις πρόσβασης υπηρεσιών στην πηγή και στον προορισμό (Service Access Points, SAPs) και μία ή δύο οκτάδες με πληροφορία ελέγχου και το παραδίδει στο MAC υποεπίπεδο.



**Σχήμα 7.1 - Η μορφή του LLC PDU**

### 7.3. Η ανάγκη για διασυνδεδεμένα τοπικά δίκτυα

Κάθε μια από τις τεχνολογίες, που αναφέρθηκαν παραπάνω (IEEE 802.X) έχει συγκεκριμένους περιορισμούς, όπως:

- **Περιορισμένο αριθμό σταθμών εργασίας:** Για παράδειγμα στα δίκτυα με κουπόνι, κάθε σταθμός που είναι συνδεδεμένος στο δίκτυο προκαλεί αυξανόμενη καθυστέρηση, ακόμα και αν δεν μεταδίδει.

- Περιορισμένο μέγεθος:** Στο 802.3 τοπικό δίκτυο, το καλώδιο πρέπει να έχει τόσο μήκος, ώστε η πιθανότητα συγκρούσεων μεταξύ απομακρυσμένων σταθμών, λόγω καθυστέρησης διάδοσης του σήματος να είναι μικρή.
- Περιορισμένο μέγεθος φορτίου:** Σε όλα τα παραπάνω LANs, η διαθέσιμη χωρητικότητα πρέπει να μοιράζεται από όλους τους σταθμούς εργασίας. Ετσι όσοι περισσότεροι οι σταθμοί, τόσο μικρότερη είναι η διαθέσιμη χωρητικότητα για τον καθένα.

Για τους τρεις παραπάνω λόγους και άλλους, οι υπηρεσίες ενός μοναδικού LAN δεν είναι πάντα ικανοποιητικές για την εξυπηρέτηση των αναγκών κάποιου οργανισμού. Υπάρχει ανάγκη για διασύνδεση περισσότερων του ενός LANs. Θα εξετάσουμε στην συνέχεια τους μηχανισμούς διασύνδεσης τοπικών δίκτυων ρίχνοντας το βάρος κυρίως σε "έξυπνα" μηχανήματα που λειτουργούν πάνω από το φυσικό επίπεδο.

#### 7.4. Γέφυρες. Λρομολόγηση στο επίπεδο MAC

Πριν περιγράψουμε τη λειτουργία της γέφυρας, είναι ίσως χρήσιμο να δούμε τον τρόπο λειτουργίας ενός **επαναλήπτη (repeater)**. Οι επαναλήπτες χρησιμοποιούνται για να εξασφαλίσουν ότι τα ηλεκτρικά σήματα μιας κάρτας ενός σταθμού εργασίας μεταδίδονται σε όλο το διασυνδεδεμένο δίκτυο. Σύμφωνα με το μοντέλο αναφοράς πρωτοκόλλων ISO/OSI, οι επαναλήπτες λειτουργούν στο φυσικό επίπεδο. Όπως αναφέραμε και παραπάνω για κάθε τοπικό δίκτυο, υπάρχει ένα ανώτατο όριο για το μήκος του και για τον αριθμό των σταθμών εργασίας που αυτό φιλοξενεί. Χρησιμοποιώντας λοιπόν κάποιον επαναλήπτη μπορούμε να αυξήσουμε τα παραπάνω όρια, συνδέοντας πολλά τμήματα (πολλά καλώδια) μαζί και μειώνοντας ταυτόχρονα τις απαιτήσεις για ισχύ σήματος από τις κάρτες των σταθμών εργασίας στα επίπεδα ισχύος σήματος που θα χρειάζονταν μονάχα για ένα τμήμα. Η ύπαρξη πολλών τμήματων και η λειτουργία αναγέννησης που εκτελεί ο επαναλήπτης είναι διαφανής (transparent) σε όλα τα επίπεδα που είναι πάνω από το φυσικό.

Για την παραπάνω εργασία είναι φανερό ότι δεν απαιτείται κάποια νοημοσύνη (δηλαδή η ύπαρξη ενός μικροεπεξεργαστή), όσο αναφορά τον επαναλήπτη. Από την άλλη πλευρά, αυτό σημαίνει ότι, εάν διάφορα τοπικά δίκτυα συνδέονται μεταξύ τους με επαναλήπτες, τα πλαίσια που κυκλοφορούν σε κάποιο επιμέρους τμήμα, θα κυκλοφορούν και σε όλα τα υπόλοιπα τμήματα, δηλαδή το διασυνδεδεμένο δίκτυο θα συμπεριφέρεται σαν ένα τμήμα.

Πέρα από τη διασύνδεση τοπικών δίκτυων υπολογιστών, μια **γέφυρα (bridge)** αποθηκεύει τα πλαίσια που πρέπει να προωθήσει και τα ελέγχει για πιθανά λάθη. Στην περίπτωση που το πλαίσιο δεν κατευθύνεται σε κάποιο διαφορετικό τμήμα, η γέφυρα δεν το προωθεί. Κατά τον τρόπο αυτό, όλες οι μεταδόσεις μεταξύ σταθμών που ανήκουν στο ίδιο τμήμα διασυνδεδεμένων τοπικών δίκτυων δεν προωθούνται (συνήθως) έξω από το τμήμα αυτό, και έτσι δεν φορτώνουν άσκοπα τα υπόλοιπα τμήματα.

Είναι φανερό ότι μια γέφυρα λειτουργεί σε υψηλότερο επίπεδο από το φυσικό, αφού διαβάζει διευθύνσεις σταθμών και συγκεκριμένα στο υποεπίπεδο ελέγχου προσπέλασης του μέσου (Media Access Control, MAC) σύμφωνα πάντα με το μοντέλο OSI. Το LAN που προκύπτει ονομάζεται γεφυρωμένο LAN.

Η αποθήκευση των πλαισίων και ο έλεγχός τους παρουσιάζει σαν λειτουργία πλεονεκτήματα και μειονεκτήματα σε σχέση με την απλή λειτουργία αναγέννησης των σημάτων που εκτελεί ο επαναλήπτης. Τα πλεονεκτήματα μιας γέφυρας είναι:

- Η βελτίωση των περιορισμών σε ότι αφορά τον αριθμό των σταθμών εργασίας και τον αριθμό των επιμέρους τοπικών δικτύων, που θα συγκροτήσουν το διασυνδεδεμένο δίκτυο, το οποίο είναι ένα σημαντικό πλεονέκτημα, όταν στήνεις τοπικά δίκτυα κατανεμημένα σε μεγάλες γεωγραφικές εκτάσεις.
- Μια γέφυρα μπορεί να λειτουργήσει και μεταξύ τοπικών δικτύων με διαφορετικό πρωτόκολλο MAC, το οποίο είναι ένα σημαντικό πλεονέκτημα στο στήσιμο τοπικών δικτύων, δεδομένου μάλιστα του αριθμού των διαφορετικών MAC πρωτοκόλλων και των ανάλογων τεχνολογιών που υπάρχουν στο εμπόριο.
- Οι γέφυρες εκτελούν την λειτουργία μετάδοσης βασιζόμενες αποκλειστικά στις διευθύνσεις που βρίσκονται στο MAC υποεπίπεδο. Με αυτό τον τρόπο είναι διαφανείς, όσο αναφορά τα υψηλότερα επίπεδα και μπορούν έτσι να λειτουργήσουν και μεταξύ σταθμών εργασίας που υποστηρίζουν διαφορετικές στοίβες πρωτοκόλλων.
- Επιτρέπουν την ευκολότερη και πιο αποτελεσματική διαχείριση ενός μεγάλου διασυνδεδεμένου δικτύου. Για παράδειγμα, υπάρχει η δυνατότητα συγχώνευσης λογισμικού διαχείρισης (agent) μαζί με το λογισμικό της γέφυρας, οπότε όλα τα δεδομένα που αφορούν τις επιδόσεις ενός τμήματος διασυνδεδεμένου δικτύου μπορούν διαρκώς να καταγράφονται και να ελέγχονται, με τη βοήθεια του δαίμονα που τρέχει στην γέφυρα. Επίσης αφού χρησιμοποιηθούν μηχανισμοί ελέγχου προσπέλασης η τοπολογία του δικτύου μπορεί να αλλάζει δυναμικά με έλεγχο των πληροφοριών που αφορούν τις θύρες της κάθε γέφυρας.
- Τέλος, διαχωρίζοντας κάποιο τοπικό δίκτυο σε μικρότερα επιμέρους τμήματα, βελτιώνεται η ολική αξιοπιστία, η διαθεσιμότητα και η ευκολία συντήρησης του ολικού δικτύου.

Από την άλλη πλευρά πιθανά μειονεκτήματα είναι τα παρακάτω:

- Από την στιγμή που μια γέφυρα καταχωρεί και επεξεργάζεται όλα τα πλαίσια που πρέπει να προωθήσει, παρουσιάζει μια επιπρόσθετη καθυστέρηση στην μετάδοση πλαισίων μέσα από ένα δίκτυο, συγκρινόμενη με τον επαναλήπτη.
- Από την στιγμή που δεν υπάρχει έλεγχος ροής στο υποεπίπεδο MAC και ο χώρος αποθήκευσης πλαισίων σε μια γέφυρα είναι πεπερασμένος, υπάρχει πιθανότητα υπερχείλισης μιας γέφυρας σε καταστάσεις υψηλής φόρτωσης του δικτύου. Αυτή είναι μια ακόμη αιτία επιπρόσθετης καθυστέρησης, από την στιγμή που τα πλαίσια αυτά θα πρέπει να επαναμεταδώθονται μεταξύ των τελικών σταθμών.
- Τέλος η γεφύρωση τοπικών δικτύων με διαφορετικό πρωτόκολλο MAC σημαίνει επίσης προετοιμασία από την γέφυρα του περιεχομένου του πλαισίου που θα μεταδώθει στο επόμενο τμήμα σύμφωνα με το διαφορετικό πρωτόκολλο. Αυτό σημαίνει μια νέα καθυστέρηση, όπως επίσης μια επιπλέον πιθανότητα λαθών.

Παρ' όλα τα παραπάνω μειονεκτήματα οι γέφυρες χρησιμοποιούνται σε ευρεία κλίμακα, μια και τα πλεονεκτήματα που παρουσιάζουν είναι σπουδαιότερα. Τα δύο πιο πλατιά αποδεκτά μοντέλα γέφυρας είναι η **διαφανής γέφυρα (transparent bridge)** και η **γέφυρα δρομολόγησης πηγής (source routing bridge)**, μεταξύ των οποίων η σημαντικότερη διαφορά είναι ο αλγόριθμος δρομολόγησης που εφαρμόζουν.

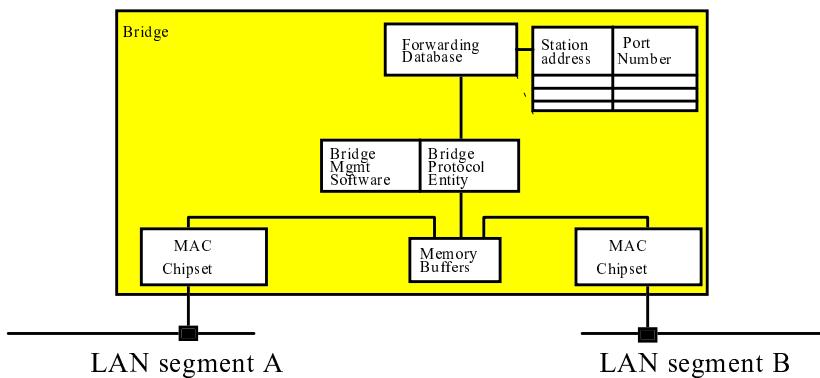
Στην περίπτωση των διαφανών γεφυρών, οι ίδιες οι γέφυρες αποφασίζουν τον τρόπο δρομολόγησης των πλαισίων, ενώ στην περίπτωση των γεφυρών δρομολόγησης πηγής, οι τερματικοί σταθμοί καθορίζουν το δρόμο των πλαισίων. Υπάρχει ένα διεθνές πρότυπο, το οποίο περιγράφει τις διαφανείς γέφυρες (IEEE 802.1 (D)), ενώ οι γέφυρες

δρομολόγησης πηγής περιγράφονται σαν μέρος του προτύπου IEEE 802.5 για τα τοπικά δίκτυα δακτυλίου με κουπόνι (Token Rings).

## 7.5. Διαφανείς γέφυρες

Στην περίπτωση των διαφανών γεφυρών, όπως και στην περίπτωση του επαναλήπτη, η παρουσία μιας (ή περισσότερων γεφυρών) σε μια διαδρομή ανάμεσα σε δύο σταθμούς εργασίας είναι διαφανής στους σταθμούς αυτούς, μια και όλες οι αποφάσεις δρομολόγησης παίρνονται αποκλειστικά από τις γέφυρες. Πολύ περισσότερο μια διαφανής γέφυρα αρχίζει αυτόματα τη λειτουργία της και αρχικοποιεί μόνη τον εαυτό της (όσο αναφορά τις πληροφορίες δρομολόγησης) με ένα δυναμικό τρόπο μόλις μπει σε λειτουργία, πράγμα που κάνει την λειτουργία της ακόμα πιο διαφανή.

Ονομάζουμε **θύρα της γέφυρας (bridge port)** τη φυσική σύνδεση της γέφυρας με το μέσο μεταφοράς. Μια βασική γέφυρα έχει δύο θύρες, ενώ υπάρχουν **πολύθυρες γέφυρες (multiport bridge)** με περισσότερες των δύο θύρες (οι οποίες συνδέουν μεγαλύτερο αριθμό τμημάτων τοπικών δικτύων). Στην πράξη, κάθε θύρα γέφυρας αποτελείται από την κάρτα (ολοκληρωμένο κύκλωμα) υλοποίησης του φυσικού επιπέδου και του MAC πρωτοκόλλου - CSMA/CD, token ring, token bus - καθώς και το ανάλογο λογισμικό διαχείρισης της θύρας. Το τελευταίο είναι υπεύθυνο για την αρχικοποίηση του ολοκληρωμένου κυκλώματος (κατά το ανέβασμα της γέφυρας) και για τη διαχείριση των καταχωρητών της γέφυρας. Συνήθως, η διαθέσιμη μνήμη είναι λογικά χωρισμένη σε ένα αριθμό από καταχωρητές καθορισμένου μήκους. Η διαχείριση των καταχωρητών αυτών, έχει το νόημα του περάσματος δεικτών στο ολοκληρωμένο κύκλωμα, οι οποίοι δείκτες δείχνουν σε ελεύθερους καταχωρητές για λήψη και αποστολή πλαισίων. Η όλη αρχιτεκτονική φαίνεται στο Σχήμα 7.2.



**Σχήμα 7.2 - Αρχιτεκτονική Γέφυρας**

Όλες οι γέφυρες λειτουργούν με αδιάκριτο τρόπο (promiscuous mode), το οποίο σημαίνει ότι λαμβάνουν και καταχωρούν όλα τα πλαίσια που λαμβάνονται από κάθε θύρα τους. Μόλις ένα πλαίσιο φθάσει σε κάποια από τις θύρες και τοποθετηθεί στον επιλεγμένο καταχωρητή από το ολοκληρωμένο κύκλωμα, το λογισμικό διαχείρισης της μνήμης προετοιμάζει το ολοκληρωμένο κύκλωμα για ένα νέο πλαίσιο, ενώ περνά τον δείκτη προς το πλαίσιο που μόλις έλαβε η γέφυρα στην οντότητα υλοποίησης του πρωτοκόλλου της γέφυρας (bridge protocol entity) για παραπέρα επεξεργασία. Για το πέρασμα αυτών δεικτών είναι προφανής η ανάγκη κάποιας ουράς αναμονής, με δεδομένο ότι υπάρχει πιθανότητα ταυτόχρονης άφιξης δύο πλαισίων, ή απαίτησης ταυτόχρονης εξόδου δύο πλαισίων από την ίδια θύρα.

Μια διαφανής γέφυρα διατηρεί μια βάση δεδομένων με πληροφορίες για την προώθηση πλαισίων (forwarding database), ή διαφορετικά ένα κατάλογο, που αφορά τη δρομολόγηση (routing directory). Ο κατάλογος αυτός υποδεικνύει τη θύρα εξόδου που πρέπει να χρησιμοποιηθεί για την προώθηση ενός πλαισίου. Στην περίπτωση που το πλαίσιο κατευθύνεται σε σταθμό του τμήματος από το οποίο έφθασε απορρίπτεται, διαφορετικά οδηγείται στην θύρα εξόδου, που υποδεικνύει ο πίνακας δρομολόγησης. Η απόφαση δρομολόγησης επιτυγχάνεται με ένα απλό ψάζιμο του πίνακα δρομολόγησης: Κατ' αρχήν διαβάζεται η MAC διεύθυνση προορισμού του πλαισίου και στην συνέχεια αναζητείται η αντίστοιχη θύρα. Αν τύχει να είναι η ίδια με τη θύρα εισόδου το πλαίσιο απορρίπτεται ή διαφορετικά μπαίνει στην ανάλογη ουρά. Η λειτουργία αυτή ονομάζεται **frame filtering**.

### 7.5.1. Η λειτουργία ενημέρωσης της γέφυρας

Ένα βασικό πρόβλημα με τις διαφανείς γέφυρες είναι η δημιουργία του πίνακα δρομολόγησης, που αναφέραμε παραπάνω. Μια προσέγγιση είναι η δημιουργία των περιεχομένων του πίνακα εκ των προτέρων και η κράτηση αυτού σε σταθερή μνήμη (Programmable Read-Only Memory, PROM). Το μειονέκτημα στην προσέγγιση αυτή είναι η ανάγκη για πλήρη αλλαγή και επαναποθήκευση του πίνακα δρομολόγησης σε κάθε απλή αλλαγή της τοπολογίας - πρόσθεση κάποιου νέου τμήματος ή αλλαγή θέσης κάποιου μηχανήματος - του γεφυρωμένου δικτύου.

Στην πράξη ποτέ δεν ακολουθείται η παραπάνω προσέγγιση, αλλά τα περιεχόμενα του πίνακα δρομολόγησης της διαφανής γέφυρας δημιουργούνται δυναμικά με την άνοδο της γέφυρας και διατηρούνται διαρκώς ενήμερα κατά τη διάρκεια της λειτουργίας της γέφυρας για τις οποιεσδήποτε αλλαγές στην τοπολογία των διασυνδεδεμένων δικτύων. Η λειτουργία αυτή επιτυγχάνεται με ένα συνδυασμό από υπολειτουργίες εκμάθυνσης της τοπολογίας και διαλόγου με τις υπόλοιπες γέφυρες.

Όταν μια γέφυρα μπαίνει σε λειτουργία οι πίνακες δρομολόγησης αρχικοποιούνται (αδειάζουν). Στα πρώτα πλαισία που θα φθάσουν, η γέφυρα θα διαβάσει τις MAC διεύθυνσεις αποστολής και θα τις εισάγει στον πίνακα μαζί με τον αριθμό θύρας από την οποία έφθασαν. Σε περίπτωση που για κάποιο από τα πλαισία αυτά δεν υπάρχει ήδη εισαγωγή στον πίνακα δρομολόγησης, το πλαίσιο αντιγράφεται σε όλες τις θύρες εξόδου της γέφυρας. Καθώς τα πλαισία διαδίδονται μέσα στο δίκτυο, η λειτουργία αυτή επαναλαμβάνεται από όλες τις γέφυρες του διασυνδεδεμένου δικτύου. Ο αριθμός της πόρτας άφιξης του πλαισίου μαζί με την διεύθυνση πηγής του πλαισίου τοποθετούνται στον πίνακα δρομολόγησης κάθε γέφυρας και στην συνέχεια το πλαίσιο προωθείται προς όλες τις άλλες δυνατές θύρες της γέφυρας. Η διαδικασία αυτή είναι γνωστή σαν **πλημμύρισμα (flooding)** μια και εξασφαλίζει ότι ένα αντίγραφο ενός πλαισίου θα βρεθεί σε όλα τα επιμέρους τμήματα του διασυνδεδεμένου LAN. Κατά τη φάση ενημέρωσης, η διαδικασία αυτή θα επαναληφθεί για όλα τα πλαισία που θα φθάσουν σε κάποια γέφυρα, και έτσι σταδιακά οι πίνακες δρομολόγησης όλων των γεφυρών θα συμπληρώσουν τα περιεχόμενά τους.

Η διαδικασία ενημέρωσης των πινάκων δρομολόγησης, όπως την περιγράψαμε παραπάνω εξαλουθεί να έχει μη δυναμικό χαρακτήρα, αφού οποιαδήποτε αλλαγή της τοπολογίας - πρόσθεση κάποιου νέου τμήματος ή αλλαγή θέσης κάποιου μηχανήματος - του γεφυρωμένου δικτύου, μετά την φάση ενημέρωσης, θα οδηγούσε σε διπλές εγγραφές στους πίνακες δρομολόγησης και διάφορα άλλα σφάλματα. Επίσης η διαδικασία αυτή έχει σωστό αποτέλεσμα αν η τοπολογία του γεφυρωμένου δικτύου έχει δομή απλού δένδρου χωρίς πολλαπλά μονοπάτια μεταξύ δύο σταθμών. Μια τέτοια δομή ονομάζεται δομή **επικαλύπτοντος δένδρου (Spanning Tree)**. Σε ένα μεγάλο γεφυρωμένο δίκτυο είναι πολύ πιθανό να δημιουργούνται βρόγχοι μεταξύ δύο τμήματων τοπικών δικτύων, τουλάχιστον για λόγους ασφάλειας.

Θα προσπαθήσουμε παρακάτω να αντιμετωπίσουμε και τα προβλήματα αυτά. Ας θυμίσουμε ότι η διεύθυνση του πρωτοκόλλου MAC για κάποιο μηχάνημα είναι σταθερή και καθορίζεται από τον κατασκευαστή της κάρτας διασύνδεσης του μηχανήματος με το δίκτυο. Έτσι η μετακίνηση ενός μηχανήματος σε μια διαφορετική θέση θα δημιουργήσει προβλήματα στους πίνακες δρομολόγησης. Υπάρχει λοιπόν ανάγκη για επανενημέρωση των πινάκων δρομολόγησης. Αυτό που γίνεται είναι το εξής: με κάθε εγγραφή μιας MAC διεύθυνσης πηγής και ενός αριθμού θύρας άφιξης πλαισίου στους πίνακες δρομολόγησης αρχικοποιείται ένας μετρητής. Ο μετρητής επαν-αρχικοποιείται και με την άφιξη νεώτερων πλαισίων με τα ίδια στοιχεία. Μόλις ο μετρητής αυτός ξεπεράσει κάποιο προκαθορισμένο κατώφλι, χωρίς να υπάρχουν καινούργια πλαισία με τα ίδια στοιχεία, η εγγραφή αφαιρείται από τον πίνακα δρομολόγησης. Οποιαδήποτε πλαισία φθάσουν μετά την αφαίρεση της εγγραφής, θα ενεργοποιήσουν από την αρχή τη διαδικασία ενημέρωσης, όπως την πρώτη φορά, και πιθανώς με διαφορετική αντιστοίχιση μεταξύ MAC διευθύνσεων και θυρών. Με τον τρόπο αυτόν οι εγγραφές των πινάκων δρομολόγησης στις γέφυρες των διασυνδεδεμένων δικτύων είναι διαρκώς ενημερωμένες για την τρέχουσα τοπολογία του δικτύου. Επίσης περιορίζουμε το μέγεθος των πινάκων δρομολόγησης σε ένα λογικό νούμερο, μια και τα περιεχόμενα περιορίζονται σε στοιχεία για τους ενεργούς μονάχα σταθμούς. Αυτό είναι σημαντικό, μια και το μέγεθος κάποιας βάσης δεδομένων επηρεάζει άμεσα την ταχύτητα πρόσβασης σε κάποιο στοιχείο.

Ο αλγόριθμος στον οποίο καταλήξαμε δεν μας λύνει όλα τα προβλήματα, που θέσαμε. Επιλύσαμε τα προβλήματα της αλλαγής της τοπολογίας, της αλλαγής της θέσης κάποιου μηχανήματος, ή της διακοπής της λειτουργίας κάποιου μηχανήματος. Δεν λύσαμε όμως το πρόβλημα, που δύο σταθμοί εργασίας συνδέονται μεταξύ τους μέσω πολλαπλών μονοπατιών. Κάτι τέτοιο μπορεί να συμβεί από λάθος κατά την συντήρηση ή την αναβάθμιση κάποιου δικτύου, ή συχνότερα μπορεί να συμβεί για λόγους αξιοπιστίας. Στην περίπτωση αυτή ο αλγόριθμος που αναπτύξαμε παραπάνω όχι μόνο δεν λειτουργεί, αλλά μπορεί να δημιουργήσει και μια καταιγίδα από πλαισία με πιθανό αποτέλεσμα την πτώση του δικτύου. Έτσι, για συνθετότερες τοπολογίες θα πρέπει να βελτιώσουμε τον βασικό αλγόριθμο που αναπτύξαμε παραπάνω.

Ο νέος αλγόριθμος θα πρέπει να επιλέγει ένα μονάχα μονοπάτι για κάθε πλαισίο, ώστε να ξεπεραστεί το σχετικό προβλήματα. Με τον τρόπο αυτόν θα υπάρχει κάποια λογική τοπολογία (γνωστή σαν **ενεργή τοπολογία, active topology**), και η οποία θα έχει την μορφή απλού επικαλύπτοντος δένδρου. Ο αλγόριθμος αυτός ονομάζεται **αλγόριθμος επικαλύπτοντος δένδρου (Spanning Tree Algorithm)**. Θα πρέπει εκ των προτέρων να τονιστεί, ότι αν και ο αλγόριθμος αυτός επιλέγει μονάχα μία γέφυρα για να διασυνδέει δύο τοπικά δίκτυα - κάνοντας περιττές όποιες άλλες γέφυρες έχουν χρησιμοποιηθεί για την διασύνδεση των δύο δικτύων, π.χ. για λόγους αύξησης της αξιοπιστίας - η επιλογή αυτή διαρκεί για κάποιο συγκεκριμένο χρονικό διάστημα, δηλαδή αλλάζει με δυναμικό τρόπο.

### 7.5.2. Ο αλγόριθμος επικαλύπτοντος δένδρου (Spanning Tree Algorithm)

Ο αλγόριθμος επικαλύπτοντος δένδρου θεωρεί ότι όλες οι γέφυρες τακτικά ανταλλάσσουν ειδικά πλαισία (μηνύματα) - γνωστά ως **μονάδες δεδομένων πρωτοκόλλου γέφυρας (Bridge Protocol Data Units - BPDUs)**. Επίσης θεωρεί ότι, κάθε γέφυρα έχει μία τιμή προτεραιότητας και ένα μοναδικό αναγνωριστικό. Με βάση τα στοιχεία αυτά, και σε τακτικά χρονικά διαστήματα, από το σύνολο των γεφυρών του γεφυρωμένου LAN, μία γέφυρα επιλέγεται δυναμικά να είναι η **γέφυρα ρίζα (root bridge)**. Αυτή είναι κάθε φορά η γέφυρα με την υψηλότερη προτεραιότητα και το μικρότερο αναγνωριστικό.

Μόλις η γέφυρα ρίζα έχει καθοριστεί, κάθε άλλη γέφυρα του διασυνδεδεμένου δικτύου καθορίζει το ποια από τις θύρες της σχηματίζει το **ελάχιστο κόστος διαδρομής (minimum path cost)** προς την γέφυρα ρίζα. Η θύρα αυτή ονομάζεται **θύρα ρίζα (root port)** για την συγκεκριμένη γέφυρα, και από την στιγμή που θα καθοριστεί είναι αυτή που θα λαμβάνει όλα τα BPDUs που θα στέλνονται από την γέφυρα ρίζα.

Το κόστος διαδρομής μεταξύ μιας θύρας κάποιας γέφυρας και της γέφυρας ρίζας μπορεί να καθοριστεί από τη χωρητικότητα (bit rate) του τμήματος στο οποίο είναι συνδεδεμένη η συγκεκριμένη θύρα. Όσο υψηλότερος είναι ο ρυθμός αυτός, τόσο μικρότερο είναι το κόστος. Αν υπάρχουν, για παράδειγμα, δύο εναλλακτικά μονοπάτια προς τη γέφυρα ρίζα από τα οποία το ένα περιλαμβάνει δύο 10Mbps CSMA/CD τμήματα και το άλλο δύο 2Mbps CSMA/CD τμήματα, τότε το μονοπάτι με τα δύο τμήματα, που έχουν τον υψηλότερο ρυθμό μετάδοσης θα έχει το χαμηλότερο κόστος διαδρομής. Στην περίπτωση όμως, που δύο θύρες, οι οποίες ανήκουν στην ίδια γέφυρα, είναι ισότιμες ως προς το παραπάνω κριτήριο, τότε χρησιμοποιούνται τα αναγνωριστικά των θυρών αυτών, σαν εναλλακτικά κριτήρια, για την εύρεση του ελάχιστου κόστους διαδρομής.

Μόλις καθοριστεί το κόστος διαδρομών, σε κάθε τμήμα τοπικού δικτύου μία μόνο γέφυρα επιλέγεται, προκειμένου να προωθεί τα πλαίσια του συγκεκριμένου τμήματος. Αυτή είναι γνωστή ως **υπεύθυνη γέφυρα (designated bridge)** για το τμήμα αυτό. Η εκλογή της βασίζεται στο ελάχιστο κόστος διαδρομής προς τη γέφυρα ρίζα. Αν δύο θύρες γέφυρων που είναι συνδεδεμένες στο ίδιο τμήμα τοπικού δικτύου, έχουν το ίδιο κόστος διαδρομής προς τη γέφυρα ρίζας, η γέφυρα με το μικρότερο αναγνωριστικό επιλέγεται. Η θύρα της υπεύθυνης γέφυρας που συνδέεται με το μέσο μεταφοράς (για το οποίο η γέφυρα είναι υπεύθυνη) είναι γνωστή ως **υπεύθυνη θύρα (designated port)**. Η γέφυρας ρίζας είναι πάντα υπεύθυνη γέφυρα για όλα τα τμήματα στα οποία είναι συνδεδεμένη, και έτσι όλες οι θύρες της είναι υπεύθυνες θύρες.

Κατά την διαδικασία εκλογής των υπευθύνων γεφυρών και φυσικά των αντιστοίχων θυρών, όσο αναφορά ένα τμήμα, θα πρέπει να σημειωθεί ότι, δεν συμμετέχουν στην διαδικασία τα root ports. Αυτό ισχύει, γιατί μια θύρα ρίζα λαμβάνει μονάχα, χωρίς να μεταδίδει ποτέ BPDUs. Η εκλογή έτσι της υπεύθυνης θύρας, γίνεται ανάμεσα στις θύρες που δεν είναι θύρες-ρίζες, και οι οποίες συνδέονται στο υπό εξέταση τμήμα. Η ανταλλαγή των καταλλήλων BPDUs ανάμεσα στις δύο (ή περισσότερες) γέφυρες που συμμετέχουν (στην διαδικασία της απόφασης) θα τους επιτρέψει να πάρουν μια συλλογική απόφαση για την επιλογή της υπεύθυνης θύρας.

Αφού καθοριστούν σε ένα γεφυρωμένο δίκτυο, η γέφυρα ρίζα, τα root ports και τα designated ports στις υπόλοιπες γέφυρες, η κατάσταση των θυρών κάθε γέφυρας μπορεί να τεθεί σε κατάσταση **προώθησης (forwarding)** ή σε κατάσταση **απομόνωσης (blocking)**. Αρχικά, αφού όλες οι θύρες της γέφυρας ρίζας είναι υπεύθυνες θύρες, βρίσκονται σε κατάσταση προώθησης. Για όλες τις άλλες γέφυρες, μόνο η θύρα ρίζα και οι υπεύθυνες θύρες τίθονται στην κατάσταση προώθησης ενώ όλες οι υπόλοιπες στην κατάσταση απομόνωσης. Η διαδικασία αυτή καθορίζει μια ενεργή τοπολογία, η οποία είναι ισοδύναμη με ένα επικαλύπτων δένδρο.

## 1. Αρχικοποίηση της τοπολογίας

Όλες οι γέφυρες σ' ένα LAN έχουν μια μοναδική ομαδική διεύθυνση MAC, την οποία και χρησιμοποιούν για την ανταλλαγή των BPDUs, που αναφέραμε. Τα BDU που λαμβάνονται από μία γέφυρα δεν προωθούνται άμεσα. Η πληροφορία που περιέχουν μπορεί να χρησιμοποιηθεί από την οντότητα υλοποίησης του πρωτοκόλλου γέφυρας (bridge protocol entity) για να δημιουργήσει το(a) BDU(s), το οποίο μεταγενέστερα προωθεί σε άλλη(ες) θύρα(ες) του.

Όταν μια γέφυρα μπαίνει για πρώτη φορά σε λειτουργία, υποθέτει ότι είναι η γέφυρα ρίζα. Μια γέφυρα που πιστεύει ότι είναι η ρίζα, ξεκινά να μεταδίδει BPDUs, με σκοπό να διαμορφώσει μ' αυτά την τοπολογία (configuration BPDUs) του δικτύου\*, σε όλες τις θύρες της (και έτοι στα τμήματα που συνδέονται σε αυτές) σε τακτικά χρονικά διαστήματα, γνωστά ως hello messages.

Κάθε hello message περιέχει έναν αριθμό πεδίων τα οποία περιλαμβάνονται:

- Το αναγνωριστικό της γέφυρας, για την οποία η γέφυρα που μεταδίδει το hello message πιστεύει ότι είναι η ρίζα.
- Το κόστος διαδρομής προς τη γέφυρα ρίζα από τη θύρα της γέφυρας στην οποία το BPDU λήφθηκε (μηδέν αρχικά).
- Το αναγνωριστικό της γέφυρας που μεταδίδει το BPDU.
- Το αναγνωριστικό της θύρας γέφυρας από την οποία το BPDU μεταδόθηκε.

Στη λήψη ενός hello message, κάθε γέφυρα, που συνδέεται στο τμήμα στο οποίο το BPDU είχε μεταδοθεί, μπορεί να καθορίσει, συγκρίνοντας το αναγνωριστικό ρίζας που περιέχεται μέσα σ' αυτό μαζί με το δικό της αναγνωριστικό, αν έχει υψηλότερη προτεραιότητα, ή σε περίπτωση ισοτιμίας, εάν το δικό της αναγνωριστικό είναι μικρότερο του αναγνωριστικού στο λαμβανόμενο πλαίσιο. Εάν συμβαίνει ένα από τα δύο, η γέφυρα αυτή θα συνεχίζει να θεωρεί τον εαυτό της γέφυρα ρίζα και απλά θα απορρίψει το πλαίσιο που έλαβε.

Εναλλακτικά αν οι συγκρίσεις με το αναγνωριστικό της ρίζας που υποδεικνύει το λαμβανόμενο BPDU αποδείξουν ότι δεν είναι η ίδια η γέφυρα ρίζα, τότε η γέφυρα προσθέτει το κόστος διαδρομής που συνδέεται με τη θύρα στην οποία το BPDU λήφθηκε, σ' αυτό που ήδη καθορίζεται από το ανάλογο πεδίο του πλαισίου, και προωθεί το πλαίσιο. Μια γέφυρα έχει γνώση του κόστους των τμημάτων με τα οποία συνδέονται οι θύρες της, ως αποτέλεσμα προηγούμενων μπνυμάτων από την διαχείριση του δικτύου που έχουν σταλεί σ' αυτήν. Στη συνέχεια δημιουργεί ένα νέο hello message που περιέχει αυτή την πληροφορία, μαζί με τα δικά της αναγνωριστικά (της γέφυρα και της θύρας) και προωθεί ένα αντίγραφο αυτού του message σε όλες τις άλλες θύρες της. Αυτή η διαδικασία επαναλαμβάνεται από όλες τις γέφυρες μέσα στο LAN. Με αυτό τον τρόπο hello messages πλημμυρίζουν όλο το δίκτυο από άκρη σε άκρη.

Καθώς τα hello messages επικαλύπτουν, ξεκινώντας από τη ρίζα, όλο το δίκτυο, το κόστος διαδρομής που αφορά κάθε θύρα σε όλες τις γέφυρες, τελικά θα υπολογιστεί. Έτσι, πέρα από τον καθορισμό μίας γέφυρας σαν ρίζα του δένδρου που τελικά θα επιτευχθεί, όλες οι άλλες γέφυρες ξέροντας το κόστος διαδρομής για κάθε μία από τις θύρες τους, θα μπορέσουν να επιλέξουν τα root ports τους. Επιπλέον με την ανταλλαγή των κατάλληλων BPDU(s), θα καθοριστεί και η υπεύθυνη γέφυρα για κάθε τμήμα. Συμπληρώνουμε ακόμα, ότι το αναγνωριστικό κάθε γέφυρας είναι αυτό που χρησιμοποιείται στην περίπτωση ισότιμου κόστους διαδρομών για την επιλογή της υπεύθυνης γέφυρας για το LAN. Μερικές βασικές παρατηρήσεις μπορούν να γίνουν πάνω στα προηγούμενα, ώστε να ολοκληρωθεί η εικόνα:

- Μια γέφυρα λαμβάνει BPDUs στη θύρα ρίζα της και τα μεταδίδει στην (ή στις) θύρες της, που είναι υπεύθυνες για κάποιο τμήμα.

---

\* Θα χρησιμοποιήσουμε εναλλακτικά και τον αγγλικό όρο hello message για την απόδοση του όρου configuration BPDU για διευκόλυνσή μας.

- Όλες οι θύρες ρίζες, καθώς και οι υπεύθυνες για κάποιο τμήμα θύρες είναι στο στάδιο προώθησης.
- Μια γέφυρα που έχει μια θύρα ρίζα συνδεδεμένη σ' ένα τμήμα δεν μπορεί να είναι η υπεύθυνη γέφυρα για το τμήμα αυτό.
- Μπορεί να υπάρξει μόνο μία υπεύθυνη θύρα σε κάθε τμήμα τοπικού δικτύου.

## 2. Αλλαγές στην τοπολογία

Όπως αναφέρθηκε προηγούμενα, συχνά εισάγονται επιπλέον γέφυρες σ' ένα γεφυρωμένο LAN για να βελτιώσουν την ολική αξιοπιστία του. Έτσι αφού η προηγούμενη διαδικασία θα θέσει μερικές (ή όλες) από τις θύρες που σχετίζονται με τέτοιες γέφυρες σε κατάσταση απομόνωσης (blocking), είναι απαραίτητο μια διαδικασία να είναι συγχωνευμένη μέσα στον αλγόριθμο για να επιτρέψει στην κατάσταση των γεφυρών και των θύρων τους να αλλάζουν δυναμικά σε περίπτωση σφάλματος κάποιας γέφυρας ή κάποιας θύρας. Αυτό είναι γνωστό ως διαδικασία αλλαγής τοπολογίας (topology change procedure).

Μόλις μια γέφυρα ρίζα και η συσχετισμένη ενεργή τοπολογία έχουν καλά καθοριστεί, μονάχα η γέφυρα ρίζα θα μεταδίδει BPDUs διαμόρφωσης της τοπολογίας. Αυτά εκπέμπονται σε τακτικά διαστήματα σε κάθε μία από τις θύρες της κάθε φορά που ο χρονιστής λήξης (hello timer), λήγει. Τέτοια BPDUs θα διαδοθούν σ' ολόκληρο το δίκτυο. Συνεπώς, καθώς η κάθε γέφυρα ενημερώνει, την πληροφορία που περιέχεται μέσα στα BPDU, η κατάσταση της κάθε γέφυρας και οι συσχετιζόμενες θύρες της, θα επιβεβαιώνονται σε τακτικά διαστήματα.

Για να μπορούν άλλες γέφυρες να ανιχνεύουν πότε συμβαίνει κάποιο σφάλμα, ένας χρονιστής ηλικίας του μηνύματος (message age timer) φυλάσσεται από κάθε γέφυρα για όλες τις θύρες της. Όσο δεν παρουσιάζονται σφάλματα, ο χρονιστής αυτός αρχικοποιείται κάθε φορά που ένα hello message λαμβάνεται. Αν όμως, μια υπεύθυνη γέφυρα ή μια ενεργή θύρα γέφυρας παρουσιάσει κάποια βλάβη, τα BPDUs θα σταματήσουν να προωθούνται διαμέσου αυτής της γέφυρας ή θύρας. Το αποτέλεσμα θα είναι ο χρονιστής ηλικίας μηνύματος, να λήξει στις γέφυρες οι οποίες ακολουθούν στο επικαλύπτων δένδρο, την γέφυρα ή την θύρα, που παρουσίασε την βλάβη.

Η λήξη ενός χρονιστή ηλικίας μηνύματος που σχετίζεται με μια θύρα, προκαλεί την οντότητα πρωτοκόλλου γέφυρας, να επικαλεστεί μια διαδικασία καθορισμού της υπεύθυνης θύρας. Ολοφάνερα, αυτή θα κληθεί από όλες τις επηρεασμένες γέφυρες. Αφού πραγματοποιηθεί αυτό, μία ή περισσότερες νέες υπεύθυνες γέφυρες και/ή θύρες θα έχουν εγκατασταθεί. Εάν ήταν η εν χρήσει γέφυρα ρίζα αυτή που παρουσίασε την βλάβη, τότε μία νέα γέφυρα ρίζα θα επιλεγεί.

Επιπρόσθετα, οποτεδήποτε η κατάσταση μίας θύρας αλλάζει από το στάδιο απομόνωσης, στο στάδιο προώθησης ένα BPDU ειδοποίησης αλλαγής της τοπολογίας εκπέμπεται από την θύρα υψηλότερου βαθμού προς την κατεύθυνση της γέφυρας ρίζας. Όλες οι υπεύθυνες γέφυρες ανάμεσα σ' αυτή και στην γέφυρα ρίζα, σημειώνουν την αλλαγή, και την αναμεταδίδουν προς τη γέφυρα ρίζα μέσω των root port τους. Μ' αυτό τον τρόπο, σε όλες τις γέφυρες που επηρεάζονται από την αποτυχία, γνωστοποιείται η αλλαγή της τοπολογίας. Για να εξασφαλιστεί ότι τέτοια BPDU, φτάνουν στη ρίζα, με αξιοπιστία, ένα BPDU επιβεβαίωσης και ένας χρονιστής χρησιμοποιούνται στην διαδικασία αποστολής.

Ολοφάνερα, αφού η τοπολογία έχει αλλάξει οι τερματικοί σταθμοί που συνδέονται σε κάθε τμήμα LAN, μπορούν να προσεγγιστούν από μία διαφορετική θύρα από αυτή που εκείνη τη στιγμή είναι στη βάση δεδομένων προώθησης της κάθε γέφυρας. Έτσι αφού ο

χρονιστής χρησιμοποιήθηκε από μία γέφυρα, για να διακόψει εγγραφές στη βάση δεδομένων της (αυτό σημαίνει, ότι οι εγγραφές σχετίζονται με τους τερματικούς σταθμούς οι οποίοι δεν έχουν μεταδόσει κανένα πλαίσιο από τη στιγμή που ο χρονιστής άρχισε), είναι σχετικά διαρκής, τα επόμενα hello messages που μεταδίδονται από τη ρίζα, μετά τη λήψη ενός BPDU ειδοποίησης αλλαγής της τοπολογίας, έχουν ένα πεδίο μέσα τους για να ειδοποιούν τις γέφυρες να ελαττώνουν αυτό το χρόνο. Μ' αυτό τον τρόπο, οι εγγραφές που υπάρχουν σε κάθε βάση δεδομένων μιας γέφυρας, θα έρθουν πιο κοντά στην χρονική στιγμή διαγραφής τους και στην συνέχεια νέες εγγραφές θα τις αντικαταστήσουν, κάθε φορά που κάποιος σταθμός στέλνει ένα πλαίσιο.

### 3. Κατάσταση των θυρών

Για να εξασφαλιστεί ότι δεν δημιουργούνται βρόγχοι, κατά την διάρκεια της περιόδου, όπου η ενεργή τοπολογία ιδρύεται, μία θύρα γέφυρας δεν επιτρέπεται να πάει κατευθείαν από την κατάσταση απομόνωσης στην κατάσταση προώθησης. Αντί γι' αυτό δύο ενδιάμεσες καταστάσεις ορίζονται, γνωστές ως κατάσταση ακρόασης (listening state) και ως κατάσταση ενημέρωσης (learning state). Μία πέμπτη κατάσταση, η μη ενεργή κατάσταση (disabled state), ορίζεται επίσης για να επιτρέψει στον διαχειριστή του δικτύου, διαμέσου ειδικών διαχειριστικών BPDUs, τα οποία στέλνονται διαμέσω του δικτύου, να απομονώσουν μόνιμα συγκεκριμένες θύρες γέφυρας.

Όσο χρονικό διάστημα κάποια γέφυρα βρίσκεται σε μία από τις πέντε καταστάσεις, κάποια από τα BPDUs επιτρέπεται ή δεν επιτρέπεται να προωθηθούν απ' αυτήν. Τα πλαίσια που προωθούνται σε κάθε κατάσταση έχουν ως εξής:

- Όταν μια θύρα είναι στην μη ενεργή κατάσταση, μόνο ειδικά BPDUs διαχείρισης λαμβάνονται και προωθούνται.
- Όταν μια θύρα βρίσκεται στην κατάσταση απομόνωσης, μόνο hello messages και ειδικά διαχειριστικά BPDUs λαμβάνονται και προωθούνται.
- Όταν μια θύρα βρίσκεται σε κατάσταση ακρόασης, όλα τα BPDUs μπορούν να ληφθούν και να προωθηθούν.
- Όταν μια θύρα είναι στην κατάσταση ενημέρωσης, όλα τα BPDUs λαμβάνονται και προωθούνται.
- Όταν μια θύρα είναι στην κατάσταση προώθησης, όλα τα BPDUs λαμβάνονται και προωθούνται. Επίσης όλα τα πλαίσια πληροφορίας του δικτύου λαμβάνονται, επεξεργάζονται και προωθούνται.

### 3. Μεταβάσεις μεταξύ των κατάστασεων

Οι μεταβάσεις μιας θύρας μεταξύ των παραπάνω καταστάσεων, κατευθύνονται από το λογισμικό της γέφυρας και πραγματοποιούνται ως αποτέλεσμα είτε κάποιων BPDUs που έχουν ληφθεί και σχετίζονται με μία θύρα της, είτε κάποιου χρονιστή που έληξε.

Κανονικά, όλες οι θύρες μιας γέφυρας τίθονται στην μη ενεργή κατάσταση, όταν η γέφυρα μπαίνει για πρώτη φορά σε λειτουργία. Η μετάβαση στην κατάσταση απομόνωσης συμβαίνει ως αποτέλεσμα ενός συγκεκριμένου διαχειριστικού BPDU που στέλνεται από τον διαχειριστή του δικτύου. Παρόμοια ο διαχειριστής του δικτύου

μπορεί να κάνει μη ενεργή μία συγκεκριμένη θύρα γέφυρας οποιαδήποτε στιγμή, στέλνοντας ένα κατάλληλο BPDU μέσω του δικτύου.

Μόλις μία γέφυρα λάβει εντολή αρχικοποίησης από τον διαχειριστή του δικτύου, θα θέσει όλες τις θύρες της στην κατάσταση απομόνωσης και θα αρχίσει να εκπέμπει hello messages. Όπως αναφέρθηκε προηγούμενα, η γέφυρα μετά συμμετέχει στην διαδικασία αρχικοποίησης της τοπολογίας (topology initialization procedure). Κατά την διάρκεια της διαδικασίας θα αρχίσει να θέτει τις θύρες της ως ρίζες ή υπεύθυνες θύρες. Από την προηγούμενη συζήτηση, μπορεί να εξαχθεί, ότι αυτή η κατάσταση των θυρών είναι αρχική, και είναι πολύ πιθανό να αλλάξει κατά την διάρκεια της διαδικασίας. Για παράδειγμα, οι γέφυρες μακριά από την τελική γέφυρα ρίζα, μπορούν να αρχίσουν να υποθέτουν ότι κάποιες από τις θύρες τους είναι υπεύθυνες ή θύρες ρίζα ως αποτέλεσμα τοπικών ανταλλαγών BPDU. Άλλα, καθώς τα BPDU συνεχίζουν να ρέουν, οι καταστάσεις τους μπορούν ν' αλλάξουν. Έτσι, αντί η ρίζα και οι υπεύθυνες θύρες να μεταφέρονται κατευθείαν από την κατάσταση απομόνωσης στην κατάσταση προώθησης οδηγούνται διαμέσου των μεσαίων καταστάσεων ακρόασης και ενημέρωσης, για μία σταθερά χρονική περίοδο.

Έτσι, όταν μία γέφυρα καθορίσει ότι μία από τις θύρες της είναι μία ρίζα ή μία υπεύθυνη θύρα, μεταφέρεται από την κατάσταση απομόνωσης στην κατάσταση ακρόασης, και ένας χρονιστής προώθησης (forwarding timer) ξεκινά. Εάν μία θύρα είναι ακόμα ρίζα ή υπεύθυνη θύρα όταν αυτός λήξει, θα μεταφερθεί στην κατάσταση ενημέρωσης.

Ο χρονιστής προώθησης μετά ξεκινά από την αρχή και η ίδια διαδικασία επαναλαμβάνεται όταν λήξει. Όμως αυτή τη φορά οι θύρες, που είναι ακόμα ρίζες ή υπεύθυνες θύρες τώρα τοποθετούνται στην κατάσταση προώθησης. Αν μία θύρα πάψει να είναι ρίζα ή υπεύθυνη θύρα κατά τη διάρκεια αυτής της περιόδου, θα επιστραφεί κατευθείαν στην κατάσταση απομόνωσης.

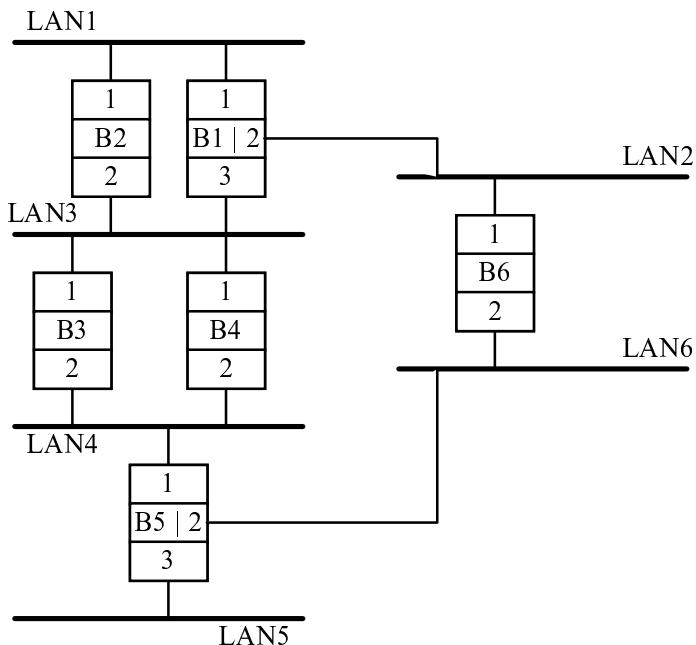
### 7.5.3. Συντονισμός της τοπολογίας

Εύκολα μπορεί να αποδειχθεί ότι, αν όλες οι γέφυρες στο LAN έχουν την ίδια προτεραιότητα, δεν είναι πολύ πιθανό ότι ο αλγόριθμος επικαλύπτοντος δένδρου θα παράγει την καλύτερη δυναμική τοπολογία, όσον αφορά τη χρήση του διαθέσιμου εύρους ζώνης. Αυτό μπορεί να αποτελέσει μειονεκτικό παράγοντα σε μεγάλα δίκτυα, εφόσον σ' αυτά είναι σημαντικό να μεγιστοποιηθεί η χρήση κάθε τμήματος τοπικού δικτύου υψηλού ρυθμού μετάδοσης.

Ο συνυπολογισμός του πεδίου προτεραιότητας σε κάθε αναγνωριστικό γέφυρας βοηθάει στο να επιτευχθεί ο σκοπός αυτός. Αν και το αναγνωριστικό μιας γέφυρας είναι μοναδικό την στιγμή της κατασκευής, το πεδίο προτεραιότητας μπορεί να τεθεί δυναμικά από το διαχειριστή του δικτύου μέσω του δικτύου στην επιθυμητή τιμή. Οι γέφυρες που μπορούν να ανταποκριθούν σε εντολές διαχείρισης του δικτύου είναι γνωστές ως **διαχειριζόμενες γέφυρες (managed bridges)**. Θέτοντας επιλεκτικά την προτεραιότητα των γεφυρών αυτών, ο διαχειριστής του δικτύου μπορεί με τον τρόπο αυτό να βελτιστοποιήσει ή να συντονίσει την απόδοση του ολικού δικτύου.

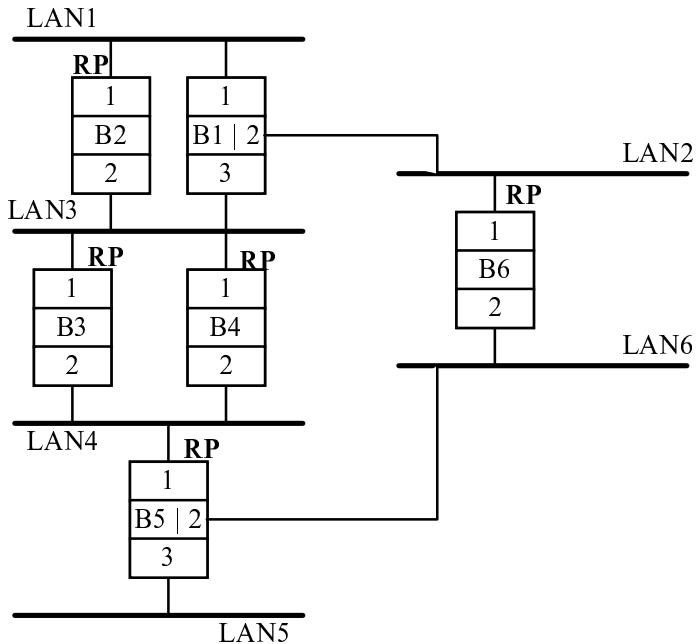
Πολύ συχνά, κάποιο πρότυπο της φυσικής τοπολογίας του γεφυρωμένου δικτύου, σχεδιάζεται και χρησιμοποιείται για την μελέτη της απόδοσης του δικτύου αυτού, με δεδομένες διάφορες τοπολογίες και κατανομές του φορτίου στις τοπολογίες αυτές. Με τον τρόπο αυτό είναι δυνατή η αναγνώριση ενδεχομένων περιοχών συμφόρησης του πραγματικού δικτύου. Στην συνέχεια προσεκτικός καθορισμός της προτεραιότητας των γεφυρών του δικτύου δίνει τη δυνατότητα βελτίωσης της απόδοσης της τοπολογίας.

### 7.5.4. Παράδειγμα

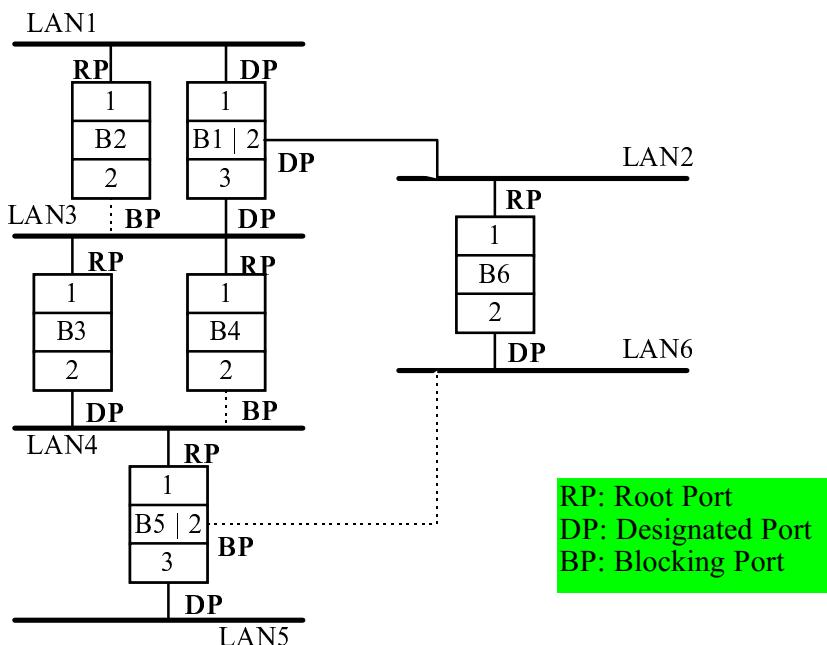


Θεωρείστε το γεφυρωμένο LAN του σχήματος. Ο σειριακός αριθμός κάθε γέφυρας φαίνεται στο σχήμα. Θεωρούμε ότι υπάρχουν κοινές προτεραιότητες για όλες τις γέφυρες και όλα τα LAN του δικτύου. Μετά τη πρώτη ανταλλαγή hello messages η γέφυρα B1 θα είναι η γέφυρα ρίζα, αφού έχει το μικρότερο αναγνωριστικό. Η ανταλλαγή των hello messages θα συνεχιστεί, οπότε και θα πραγματοποιηθεί ο υπολογισμός του κόστους του μονοπατιού από κάθε γέφυρα προς τη ρίζα. Οι πόρτες των γεφυρών με το μικρότερο κόστος θα επιλεχθούν σαν πόρτες ρίζα (root ports). Σε περίπτωση ίδιου κόστους (π.χ. B2) θα επιλεχθεί η πόρτα με το μικρότερο αναγνωριστικό.

Στο παρακάτω σχήμα βλέπουμε τις πόρτες που έχουν επιλεχθεί σαν root ports.



Τέλος, επιλέγονται οι υπεύθυνες πόρτες για κάθε LAN και οι πόρτες που θα περάσουν σε κατάσταση blocking.



Το δένδρο που προέκυψε είναι το επικαλύπτων δένδρο. Αυτό επιβεβαιώνεται περιοδικά με τα hello messages. Σε περίπτωση κάποιας αλλαγής (π.χ. μη λειτουργίας της γέφυρας B6) το δένδρο επαναπροσδιορίζεται.

### 7.5.5. Απομακρυσμένες γέφυρες (Remote Bridges)

Πολλές μεγάλες εταιρίες έχουν εγκαταστάσεις (και κατά συνέπεια τοπικά δίκτυα με υπολογιστές) κατανεμημένες σε όλη την έκταση μιας χώρας ή ακόμα και σε διαφορετικές χώρες. Πέρα λοιπόν από την ικανότητα να ανταλλάσσουν πληροφορίες ανάμεσα στους σταθμούς που συνδέονται στο ίδιο LAN σε κάποια εγκατάσταση, πολλές μεγάλες εταιρίες απαιτούν τη δυνατότητα να ανταλλάσσουν πληροφορίες μεταξύ σταθμών που συνδέονται σε LANs, που βρίσκονται σε απομακρυσμένα μέρη. Αυτό απαιτεί ένα μέσο διασυνδέσης των απομακρυσμένων LANs αυτών.

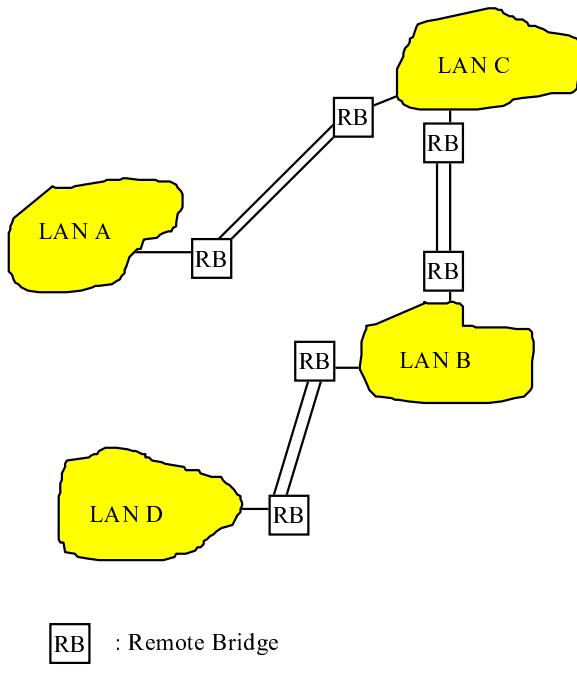
Πολλές εναλλακτικές λύσεις, είναι πιθανές. Μία λύση είναι να χρησιμοποιηθεί ένα δημόσιο ή ιδιωτικό δίκτυο μεταγωγής πακέτων (packet-switching network) για τις λειτουργίες επικοινωνίας inter-LAN. Τυπικά ένας περιορισμένος αριθμός μόνιμων νοητών κυκλωμάτων μπορεί να χρησιμοποιηθεί ή εναλλακτικά, διακοπτόμενες συνδέσεις που εγκαθίστανται δυναμικά. Ωστόσο και οι δύο αυτές λύσεις απαιτούν σ' όλο το δίκτυο διευθύνσεις επιπέδου δικτύου, για τη λειτουργία δρομολόγησης, κάνοντας απαραίτητη την χρησιμοποιήση ενός δρομολογητή, για τη σύνδεση του κάθε LAN στο δίκτυο μεταγωγής πακέτων.

Μια εναλλακτική και απλούστερη λύση είναι να διασυνδεθούν τα LANs μέσω αφερωμένων γραμμών αποκλειστικής ιδιωτικής χρήσεως (dedicated leased private lines), οι οποίες συνδέονται πάνω σε γέφυρες των απομακρυσμένων δικτύων. Αν και αυτή η προσέγγιση θυσιάζει μερικά από τα προτερήματα που αποκτώνται χρησιμοποιώντας δρομολογητές (routers), θα παρέχει πολλές φορές υπηρεσίες αναμετάδοσης γρηγορότερα.

Οι γραμμές αποκλειστικής χρήσης νοικιάζονται από μία δημόσια ή ιδιωτική τηλεφωνική εταιρία για τη δημιουργία ενός δικτύου άμεσων συνδέσεων (point to point) μεταξύ του κατανεμημένου συνόλου των LANs. Μ' αυτόν τον τρόπο δεν απαιτείται καμία δρομολόγηση στο δίκτυο ευρείας περιοχής που προκύπτει. Οι διευθύνσεις MAC 48-bit στην επικεφαλίδα του κάθε πλαισίου μπορούν να χρησιμοποιηθούν αντί γι' αυτή. Οι διευθύνσεις MAC που χρησιμοποιούνται στα LANs είναι μοναδικές στο WAN δίκτυο. Έτσι είναι δυνατό να χρησιμοποιηθούν οι διευθύνσεις MAC και γέφυρες για την πραγματοποίηση της λειτουργίας της δρομολόγησης.

Τέτοιες γέφυρες είναι συνδεδεμένες με την μία θύρα απ' ευθείας στις εγκαταστάσεις του υποδικτύου, και στις γραμμές αποκλειστικής χρήσης με την άλλη τους θύρα. Μία όμοια γέφυρα χρησιμοποιείται στο άλλο άκρο της γραμμής αποκλειστικής χρήσης. Για να διακρίνονται οι γέφυρες αυτές από τις γέφυρες που χρησιμοποιούνται για την διασύνδεση τμημάτων LAN, χρησιμοποιείται η ονομασία απομακρυσμένες γέφυρες (remote bridges).

Μια απομακρυσμένη γέφυρα πραγματοποιεί τις ίδιες συναρτήσεις με οποιαδήποτε άλλη γέφυρα, απλά χρειάζεται μια ειδική πρόσβαση για την σύνδεσή της με τη γραμμή αποκλειστικής χρήσης. Τυπικές μορφές τέτοιων προσβάσεων είναι το πρωτόκολλο X.21 για 64Kbps και το G.703 line coding για 2Mbps. Υπάρχουν επίσης προσβάσεις (κάρτες) και για X.25 δίκτυο, όπως 64Kbps ISDN δίκτυο στενής ζώνης. Ένα παράδειγμα δικτύου βασισμένο σε απομακρυσμένες γέφυρες φαίνεται στο Σχήμα 7.3.



**[RB]** : Remote Bridge

**=====** : Leased Line

### Σχήμα 7.3 - Λιασυνδεδεμένα LANs με Remote Bridges

Πολλές μεγάλες εταιρίες χρησιμοποιούν γραμμές αποκλειστικής χρήσης με αυτό τον τρόπο, για να διασυνδέσουν τις (ιδιωτικές) τηλεφωνικές ανταλλαγές σε κάθε εγκατάσταση, ως εκ τούτου δημιουργούν ένα ιδιωτικό τηλεφωνικό δίκτυο μεγάλης εμβέλειας. Οι γραμμές αποκλειστικής χρήσης που χρησιμοποιούνται για την επικοινωνία δεδομένων είναι κανονικά ενοποιημένες μαζί μ' αυτές που χρησιμοποιούνται για την τηλεφωνία. Ο ρυθμός των γραμμών αποκλειστικής χρήσης κυμαίνεται από πολλαπλάσια των 56Kbps (64Kbps στην Ευρώπη) μέχρι και πολλαπλάσια των 1.54Mbps (2.048Mbps στην Ευρώπη). Οι πιθανές μακρινές αποστάσεις που συνδέονται από τέτοιες γραμμές μας προδιαθέτουν για την καθυστέρηση διάδοσης των γραμμών αυτών, η οποία πρέπει να ληφθεί υπόψη όταν καθορίζεται η καθυστέρηση μετάδοσης.

Γενικά η αξιοπιστία μιας γραμμής αποκλειστικής χρήσης είναι σημαντικά μικρότερη από την αξιοπιστία ενός τμήματος LAN, έτσι είναι αναγκαίο να υπάρχουν πολλαπλά μονοπάτια (γραμμές) που να την διαφυλάσσουν σε περίπτωση αποτυχιών. Αν και αρχικά είναι δυνατόν να τεθεί σε εφαρμογή ο αλγόριθμος του δένδρου συνδέσεων σε απομακρυσμένες γέφυρες (έτσι εκτείνεται η κάλυψη του αλγόριθμου του επικαλύπτοτος δένδρου κατά μήκος όλου του δικτύου), στην πράξη αυτό δεν γίνεται πάντα.

Όπως είδαμε, με τον αλγόριθμο του επικαλύπτοντος δένδρου μερικές από τις θύρες που σχετίζονται με τις υπεύθυνες γέφυρες (οι οποίες παρουσιάζονται για να επαυξήσουν την αξιοπιστία για παράδειγμα) τοποθετούνται σε κατάσταση απομόνωσης εξασφαλίζοντας μία ενεργή τοπολογία δομής επικαλύπτοντος δένδρου. Όταν έχει κανείς γραμμές αποκλειστικής χρήσης (leased lines), αυτό σημαίνει ότι μία διαθέσιμη γραμμή μπορεί να μην χρησιμοποιηθεί, αφού η θύρα γέφυρας με την οποία είναι συνδεδεμένη είναι σε κατάσταση απομόνωσης. Αντίθετα, όμως με τα μέσα μετάδοσης που χρησιμοποιούνται στα LAN, οι γραμμές αποκλειστικής χρήσης είναι ακριβές, έτσι είναι σημαντικό να αυξηθεί η χρήση τους.

Σε πολλές περιπτώσεις οι γραμμές αποκλειστικής χρήσης αποτελούν μέρος ενός πολύ μεγαλύτερου δικτύου δεδομένων και φωνής. Κανονικά, τέτοια δίκτυα έχουν ολοκληρωμένες λύσεις για το πρόβλημα της διαχείρισής τους, των οποίων μία από τις

κύριες εργασίες είναι ο επανεύρεση του απαραίτητου εύρους ζώνης σε περίπτωση αποτυχίας μιας γραμμής αποκλειστικής χρήσης.

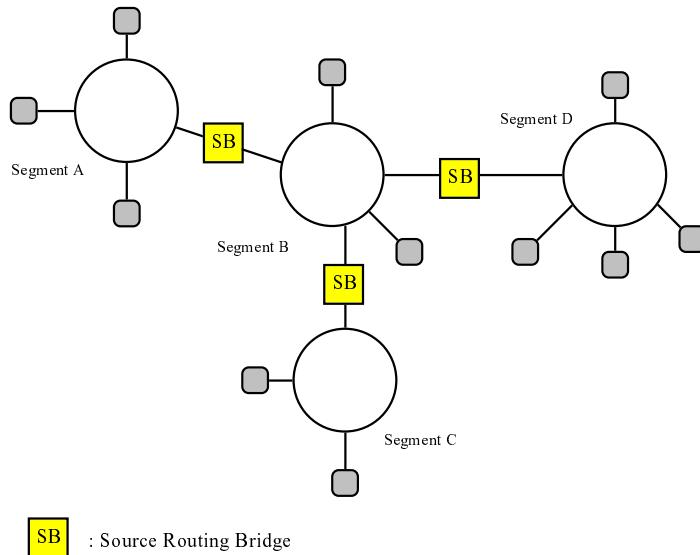
Μία κοινή λύση για τον διαχειριστή του δικτύου, είναι να ορίσει δυναμικά μια εναλλακτική γραμμή για χρήση, σε περίπτωση αποτυχίας. Οι απομονωμένες γέφυρες δεν έχουν σχέση με τον αλγόριθμο του επικαλύπτοντος δένδρου, αλλά απλά εκτελούν τις βασικές λειτουργίες ενημέρωσης και προώθησης. Παρόλο που αυτό θα οδηγήσει σε μια μικρή υποβίβαση των επιδόσεων κατά τη διάρκεια της περιόδου επανασχηματισμού, συχνά οδηγεί σε μια πιο επαρκή χρήση της διαθέσιμης χωρητικότητας μετάδοσης.

## 7.6. Γέφυρες δρομολόγησης πηγής (Source Routing Bridges)

Αν και οι γέφυρες δρομολόγησης πηγής μπορούν να χρησιμοποιηθούν με κάθε τύπο τμήματος LAN, είναι συνηθισμένο να χρησιμοποιούνται για την διασύνδεση τμημάτων LAN δακτύλιου με κουπόνι. Ένα τυπικό δίκτυο βασισμένο στις γέφυρες δρομολόγησης πηγής δείχνεται στο Σχήμα 7.4.

Η μεγαλύτερη διαφορά μεταξύ ενός LAN βασισμένου σε γέφυρες δρομολόγησης πηγής και σ' ένα άλλο βασισμένο σε γέφυρες με δρομολόγηση επικαλύπτοντος δένδρου, είναι ότι με το τελευταίο οι γέφυρες συλλογικά εκτελούν τη λειτουργία δρομολόγησης μ' έναν τρόπο που είναι διαφανής στους τερματικούς σταθμούς. Αντίθετα, στην περίπτωση της δρομολόγησης πηγής, είναι οι τερματικοί σταθμοί εκείνοι που εκτελούν τη λειτουργία της δρομολόγησης. Στην περίπτωση της δρομολόγησης πηγής, ένας σταθμός εξακριβώνει τη διαδρομή που θα ακολουθηθεί από κάποιο πλαίσιο, για κάθε προορισμό, πριν το πλαίσιο αυτό μεταδοθεί. Αυτή η πληροφορία μπαίνει στην επικεφαλίδα του πλαισίου και χρησιμοποιείται από κάθε γέφυρα για να καθορίσει εαν το λαμβανόμενο πλαίσιο θα πρέπει να προωθηθεί σ' ένα άλλο τμήμα ή όχι. Η πληροφορία δρομολόγησης περιλαμβάνει μια ακολουθία από ζευγάρια αναγνωριστικών τμημάτων τοπικών δικτύων και γεφυρών.

Λαμβάνοντας κάθε πλαίσιο, η γέφυρα χρειάζεται να ψάξει μονάχα το πεδίο δρομολόγησης στην επικεφαλίδα του πλαισίου για το δικό της αναγνωριστικό. Εάν αυτό υπάρχει μαζί με το αναγνωριστικό ενός τμήματος συνδεδεμένου με μία από τις θύρες της γέφυρας, τότε η γέφυρα προωθεί το πλαίσιο στο καθορισμένο τμήμα LAN, διαφορετικά δεν προωθεί το πλαίσιο. Και στις δύο περιπτώσεις το πλαίσιο επαναλαμβάνεται στο δακτύλιο από τη γέφυρα, και επιπλέον στην περίπτωση που η γέφυρα έχει προωθήσει το πλαίσιο τα bits αναγνωρισμένη διεύθυνση (A) και αντιγραμμένο πλαίσιο (C) στο πεδίο κατάστασης πλαισίου (Frame Status, FS) (βλ. και Σχήμα - 7.6) στην ουρά του πλαισίου θέτονται για να δείξουν στο σταθμό πηγή (γέφυρα) ότι το πλαίσιο έχει ληφθεί (προωθηθεί) από τον σταθμό προορισμού (γέφυρα).



**Σχήμα 7.4 - Γεφυρωμένο LAN δρομολόγησης πηγής**

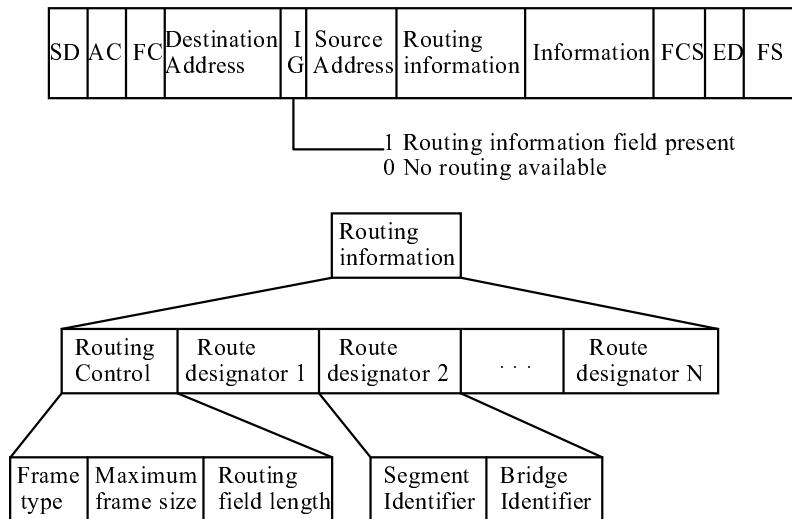
### 1. Αλγόριθμος δρομολόγησης (Routing Algorithm)

Το πεδίο πληροφορίας δρομολόγησης που περιέχεται μέσα σε κάθε πλαίσιο, βρίσκεται αμέσως μετά το πεδίο διεύθυνσης πηγής στην επικεφαλίδα του πλαισίου (IEEE 802.5).

Από τη στιγμή που ένα πεδίο πληροφορίας δρομολόγησης δεν απαιτείται πάντα, για παράδειγμα, εάν οι σταθμοί πηγής και προορισμού είναι στο ίδιο τοπικό δίκτυο, το πρώτο bit από την διεύθυνση πηγής - την ξεχωριστή/ομαδική (I/G) διεύθυνση του bit - χρησιμοποιείται για να δείξει πότε υπάρχει πληροφορία δρομολόγησης στο πλαίσιο (λογικό 1) και πότε όχι (λογικό 0). Αυτό μπορεί να γίνει από τη στιγμή που η διεύθυνση πηγής σ' ένα πλαίσιο πρέπει πάντα να είναι μια ξεχωριστή διεύθυνση, έτσι το I/G bit δεν χρειάζεται γι' αυτό το σκοπό.

Εάν υπάρχει πληροφορία δρομολόγησης, το πεδίο πληροφορίας δρομολόγησης αποτελείται από ένα πεδίο ελέγχου δρομολόγησης και από ένα ή περισσότερα πεδία προσδιορισμού της διαδρομής. Το ίδιο το πεδίο ελέγχου δρομολόγησης περιλαμβάνει τρία υποπεδία: τον τύπο του πλαισίου, το μέγιστο μέγεθος πλαισίου και το μήκος του πεδίου προσδιορισμού της διαδρομής. Όπως θα εξηγηθεί παρακάτω, επιπρόσθετα με τα κανονικά πλαίσια, δύο άλλοι τύποι πλαισίων σχετίζονται με τον αλγόριθμο δρομολόγησης. Έτσι το πεδίο τύπου πλαισίου (frame type), καθορίζει τον τύπο του πλαισίου μεταξύ των τριών δυνατών πλαισίων.

Οι γέφυρες δρομολόγησης πηγής μπορούν να χρησιμοποιηθούν για την διασύνδεση διαφορετικών τύπων τμημάτων LAN, επιπρόσθετα με τα τοπικά δίκτυα, τοπολογίας δακτυλίου, με κουπόνι. Συνεπώς, αφού μπορεί να υπάρχει ένα διαφορετικό μέγιστο μέγεθος πλαισίου με κάθε τύπο τοπικού δικτύου, το πεδίο μέγιστο μέγεθος πλαισίου χρησιμοποιείται για να καθορίσει το μεγαλύτερο μέγεθος πλαισίου, που μπορεί να χρησιμοποιηθεί όταν μεταδίδεται ένα πλαίσιο ανάμεσα σε οποιουσδήποτε δύο σταθμούς συνδεδεμένους με το LAN.



### Σχήμα 7.5 - Μορφή πλαισίου δικτύου δακτυλίου με κουπόνι

Για να επιτευχθεί μια ουσιαστική χρήση του πεδίου αυτού, πριν από την μετάδοση ενός πλαισίου εύρεσης διαδρομής, ένας σταθμός τοποθετεί το μέγιστο μέγεθος πλαισίου που μπορεί να χρησιμοποιηθεί στο ολικό LAN στο πεδίο αυτό. Πριν μια γέφυρα προωθήσει το πλαίσιο πάνω σ' ένα τμήμα, συγκρίνει αυτό το πεδίο με το (γνωστό) μέγιστο μέγεθος πλαισίου του τμήματος, στο οποίο θα προωθήσει το πλαίσιο. Εάν το τελευταίο είναι μικρότερο, ελαττώνει το μέγεθος, που βρίσκεται στο πεδίο αυτό στην χαμηλότερη τιμή. Μ' αυτόν τον τρόπο ο σταθμός πηγή όταν θα λάβει το πλαίσιο απάντηση θα βρει μέσα το μέγιστο μέγεθος πλαισίου για την συγκεκριμένη διαδρομή, και κατά συνέπεια μπορεί να χρησιμοποιήσει την πληροφορία αυτή, όταν προετοιμάζει πλαίσια για μετάδοση σ' αυτόν τον προορισμό.

Τέλος, αφού ο αριθμός των τμημάτων και των γεφυρών, από τα οποία θα περάσει ένα πλαίσιο, όταν πηγαίνει από την πηγή στον προορισμό, μπορεί να ποικίλει, το μήκος πεδίου δρομολόγησης δείχνει τον αριθμό των ζευγαριών αναγνωριστικών (τμημάτων - γεφυρών), που παρουσιάζονται στο υπόλοιπο πεδίο της πληροφορίας δρομολόγησης.

Οι δύο πρόσθετοι τύποι πλαισίων που σχετίζονται με τον αλγόριθμο εύρεσης διαδρομής είναι ένα πλαίσιο broadcast εκπομπής, το οποίο ακολουθεί μία μόνο διαδρομή και ένα πλαίσιο broadcast εκπομπής, το οποίο ακολουθεί όλες τις δυνατές διαδρομές. Για να βρει μια διαδρομή, ένας σταθμός πρώτα δημιουργεί και μεταδίδει ένα πλαίσιο broadcast εκπομπής μονής διαδρομής με ένα άδειο πεδίο δρομολόγησης και με το πεδίο μεγίστου μεγέθους πλαισίου να έχει την μεγαλύτερη δυνατή τιμή για το ολικό LAN. Όπως και στην περίπτωση των γεφυρών δρομολόγησης επικαλύπτοντος δένδρου, οι γέφυρες δρομολόγησης πηγής λειτουργούν με αδιάκριτο τρόπο και έτσι θα λάβουν και θα καταχωρήσουν όλα τα πλαίσια σε κάθε θύρα τους. Στην λήψη ενός πλαισίου broadcast εκπομπής μονής διαδρομής, η γέφυρα απλά εκπέμπει ένα αντίγραφο του πλαισίου σε κάθε τμήμα που συνδέεται με τις θύρες της. Αφού αυτή η διαδικασία επαναλαμβάνεται από κάθε γέφυρα μέσα στο LAN, ένα αντίγραφο, του πλαισίου θα διαδοθεί σε ολόκληρο το LAN και έτσι θα ληφθεί από το καθορισμένο σταθμό προορισμού ανεξάρτητα από το τμήμα στο οποίο είναι προσαρτημένος.

Όπως έχει δειχθεί στις προηγούμενες ενότητες, όμως, εάν υπάρχουν περιττές γέφυρες (και έτσι βρόχοι) στην τοπολογία του LAN, πολλαπλά αντίγραφα του πλαισίου θα μεταδοθούν γύρω από το LAN. Για να αποφύγουμε αυτό, πριν σταλούν οποιαδήποτε πλαίσια εύρεσης διαδρομής, οι ουρές γέφυρας διαμορφώνονται για να δώσουν μία ενεργή τοπολογία δένδρου συνδέσεων. Επιφανειακά, αυτό μπορεί να εμφανίζεται να

είναι η ίδια διαδικασία που χρησιμοποιείται στις διαφανής γέφυρες. Με τις γέφυρες δρομολόγησης πηγής όμως, η ενεργή τοπολογία δένδρου συνδέσεων που προκύπτει χρησιμοποιείται μόνο για την δρομολόγηση των αρχικών πλαισίων εκπομπής μονής διαδρομής. Αυτό εξασφαλίζει ότι μόνο ένα αντίγραφο του πλαισίου μεταδίδεται διαμέσου του δικτύου. Δεν χρησιμοποιείται για την δρομολόγηση ούτε κανονικών πλαισίων πληροφορίας, ούτε για πλαίσια broadcast εκπομπής όλων των διαδρομών.

Στη λήψη ενός πλαισίου broadcast εκπομπής μονής διαδρομής, ο σταθμός προορισμού επιστρέφει ένα broadcast πλαίσιο εκπομπής όλων των δυνατών διαδρομών στον σταθμό πηγής. Αντίθετα από την broadcast εκπομπή μονής διαδρομής όμως, αυτό το πλαίσιο δεν είναι εξαναγκασμένο να ακολουθήσει την ενεργή τοπολογία του επικαλύπτοντος δένδρου σε κάθε ενδιάμεση γέφυρα. Αντί γι' αυτό στη λήψη τέτοιων πλαισίων, η γέφυρα απλά προσθέτει ένα νέο πεδίο προσδιορισμού διαδρομής (συμπεριλαμβάνοντας το αναγνωριστικό τμήματος, πάνω στο οποίο το πλαίσιο είχε ληφθεί και το δικό της αναγνωριστικό), αιξάνει το μήκος του πεδίου δρομολόγησης και μετά εκπέμπει ένα αντίγραφο του πλαισίου σε μία από τις άλλες θύρες της.

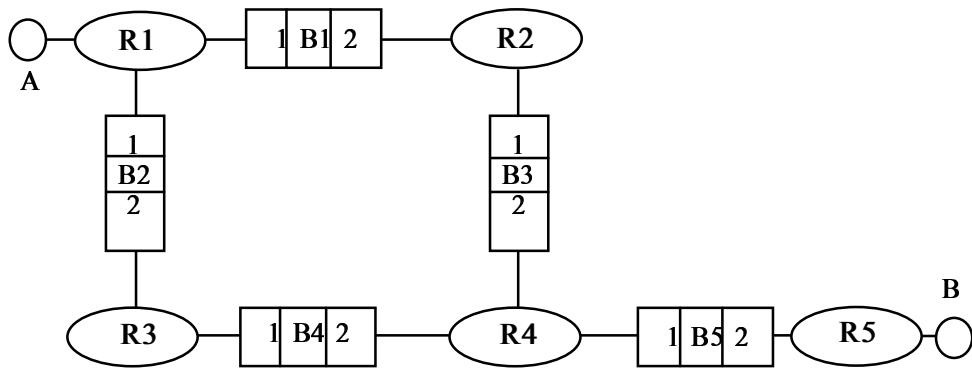
Μ' αυτόν τον τρόπο, ένα ή περισσότερα αντίγραφα του πλαισίου θα ληφθούν από τον σταθμό πηγής, μέσω όλων των πιθανών διαδρομών μεταξύ των δύο σταθμών. Εξετάζοντας τα πεδία ελέγχου δρομολόγησης του κάθε πλαισίου, ο σταθμός πηγή μπορεί να επιλέξει για να χρησιμοποιήσει τον καλύτερο δρόμο, προκειμένου να μεταδώσει ένα πλαίσιο σ' αυτόν τον προορισμό. Αυτό μετά εισάγεται μέσα στον πίνακα δρομολόγησης και μεταγενέστερα χρησιμοποιείται όταν οποιαδήποτε πλαίσια μεταδίδονται προς αυτόν τον σταθμό.

Αφού το πλαίσιο broadcast εκπομπής όλων των διαδρομών δεν εξαναγκάζεται να ακολουθήσει την ενεργή τοπολογία του επικαλύπτοντος δένδρου, στη λήψη τέτοιων πλαισίων, επιπρόσθετα βήματα πρέπει να γίνουν από κάθε γέφυρα για να εξασφαλίσει ότι κανένα πλαίσιο δεν θα κυκλοφορεί απλά μέσα σε βρόγχους του δικτύου. Πριν μεταδοθεί ένα αντίγραφο του πλαισίου broadcast εκπομπής όλων των διαδρομών σ' ένα τμήμα, η κάθε γέφυρα πρώτα ελέγχει την ήδη υπαρκτή πληροφορία δρομολόγησης στο πλαίσιο, για να καθορίσει αν την έχει επαναεπισκεφτεί το συγκεκριμένο πλαίσιο. Εάν συμβαίνει αυτό, ένα αντίγραφο του πλαισίου είναι ήδη κατά μήκος της διαδρομής προς τον σταθμό πηγής, έτσι το αντίγραφο αυτό του πλαισίου δεν προωθείται.

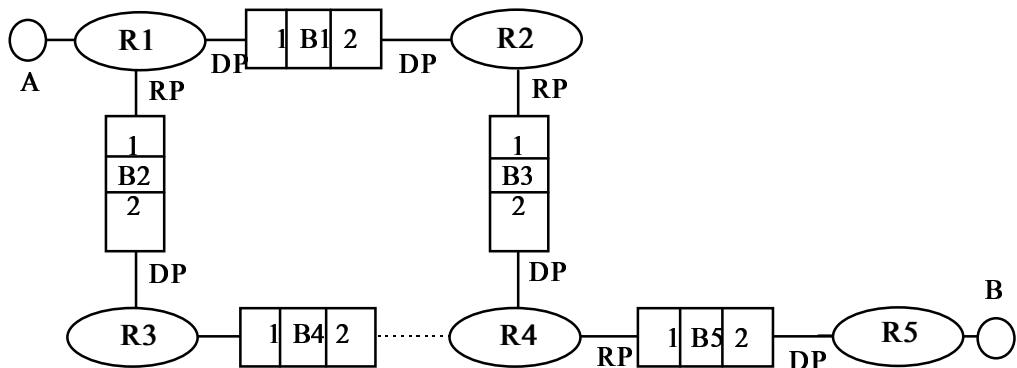
Θα έπρεπε να τονιστεί ότι δεν είναι απαραίτητο να εκτελέστει η λειτουργία εύρεσης διαδρομής για κάθε πλαίσιο που μεταδίδεται. Από την στιγμή που μία διαδρομή προς έναν προορισμό έχει ήδη καθοριστεί και εισαχθεί μέσα στον πίνακα δρομολόγησης ενός σταθμού, αυτή θα χρησιμοποιηθεί για την μετάδοση όλων των μεταγενέστερων πλαισίων σ' αυτόν τον προορισμό. Επιπλέον, αφού οι περισσότεροι σταθμοί μεταδίδουν την πλειοψηφία των πλαισίων τους, σ'έναν περιορισμένο αριθμό προορισμών, ο αριθμός των πλαισίων εύρεσης διαδρομής είναι σχετικά μικρός συγκρινόμενος με τα πλαίσια πληροφορίας για μετρίου μεγέθους LANs.

### 7.6.1. Παράδειγμα

Θεωρείστε ότι το γεφυρωμένο LAN του παρακάτω σχήματος θα λειτουργήσει χρησιμοποιώντας το πρωτόκολλο δρομολόγησης πηγής.



Εφαρμόζοντας το spanning tree algorithm θα προκύψει το επικαλύπτων δένδρο για το γεφυρωμένο LAN του σχήματος, το οποίο θα είναι σύμφωνα με όσα είπαμε παραπάνω:



Προκειμένου ο σταθμός A να βρει το σταθμό B, στέλνει ένα single-root broadcast frame το οποίο θα ακολουθήσει την εξής διαδρομή:

R1→B1→R2→B3→R4→B5→R5  
 ↳ B2→R3

Λαμβάνοντας το πλαίσιο αυτό ο σταθμός B θα στείλει ένα all-routes broadcast frame το οποίο θα ακολουθήσει τις εξής διαδρομές:

R5→B5→R4→B3→R2→B1→R1→B2→R3  
 ↳ B4→R3→B2→R1→B1→R2

Όποιο από τα δύο πλαίσια φθάσει πρώτο στο σταθμό A, αυτό είναι που θα καθορίσει τη διαδρομή που θα ακολουθηθεί στη συνέχεια.

## 7.7. Σύγκριση μεταξύ διαφανών γεφυρών και γεφυρών δρομολόγησης πηγής

Κάθε ένα από τα δύο σχέδια γεφύρωσης - αυτό της δρομολόγησης πηγής και αυτό της διαφάνειας - έχει πλεονεκτήματα και μειονεκτήματα σε σχέση με το άλλο. Στις επόμενες παραγράφους γίνεται σύγκριση των δύο σχεδίων, βασισμένη σε διάφορα κριτήρια.

### 1. Η φιλοσοφία της δρομολόγησης

Σ' ένα LAN, βασισμένο στις διαφανείς γέφυρες, η δρομολόγηση ενός πλαισίου είναι διαφανής στους τερματικούς σταθμούς. Οι τελευταίοι απλά προσθέτουν την διεύθυνση MAC του σταθμού προορισμού στην επικεφαλίδα του κάθε πλαισίου και οι γέφυρες έπειτα συλλογικά δρομολογούν τα πλαίσια διαμέσου του δικτύου, στους προκαθορισμένους προορισμούς τους. Έτσι λοιπόν, είναι οι γέφυρες που συνεργάζονται για να εκτελέσουν την λειτουργία εύρεσης της διαδρομής.

Με ένα LAN βασισμένο σε γέφυρες δρομολόγησης πηγής, η διαδρομή που θα ακολουθηθεί από ένα πλαίσιο εισάγεται στην επικεφαλίδα του πλαισίου από τον σταθμό πηγής πριν τη μετάδοση. Έτσι η διαδρομή που θα ακολουθηθεί από ένα πλαίσιο καθορίζεται από τους τερματικούς σταθμούς και όχι από τις γέφυρες. Η γέφυρα δρομολόγησης πηγής απλά προσθεί κάθε πλαίσιο από το ένα τμήμα στο άλλο, με την βοήθεια της πληροφορίας δρομολόγησης από την επικεφαλίδα του πλαισίου.

### 2. Η ποιότητα των διαδρομών

Στην περίπτωση των διαφανών γεφυρών, ο αλγόριθμος του επικαλύπτοντος δένδρου εξασφαλίζει ότι δεν υπάρχουν βρόγχοι στην ενεργή τοπολογία και το δένδρο που προκύπτει από τον αλγόριθμο αυτό χρησιμοποιείται σαν βάση για την δρομολόγηση όλων των πλαισίων μέσα στο δίκτυο. Αν και ο προσδιορισμός των προτεραιοτήτων κάθε γέφυρας και των υπεύθυνων γεφυρών για κάθε τμήμα τοπικού δικτύου βοηθούν, δεν είναι πιθανό ότι η τοπολογία του επικαλύπτοντος δένδρου που προκύπτει θα λαμβάνει υπ' όψη όλες τις βέλτιστες διαδρομές ανάμεσα σε κόμβους του διασυνδεδεμένου δικτύου. Σαφώς, όπως έχει περιγραφεί, ο αλγόριθμος του επικαλύπτοντος δένδρου αναπόφευκτα θα μπλοκάρει μερικά τμήματα τα οποία θα μπορούσαν να χρησιμοποιηθούν σε μια ιδανική περίπτωση κατά την δρομολόγηση πλαισίων μεταξύ δύο σταθμών.

Με τις γέφυρες δρομολόγησης πηγής, τα broadcast πλαίσια που αρχικά πλημμυρίζουν το δίκτυο θα αναγνωρίσουν, όλες τις πιθανές διαδρομές ανάμεσα στην κόμβο πηγή και στον κόμβο προορισμού. Έτσι η πηγή μπορεί να διαλέξει την βέλτιστη διαδρομή που θα πρέπει να ακολουθηθεί για κάθε προορισμό. Θα πρέπει να τονίσουμε, ότι αυτό είναι αληθές υπό τον όρο ότι η εκλογή δρομολόγησης είναι βασισμένη όχι μόνο στον αριθμό των τμημάτων (και έτσι των γεφυρών) σε μια διαδρομή, αλλά επίσης στο κόστος (bit rate) του κάθε τμήματος. Επιπλέον, σε διασυνδεδεμένα δίκτυα με εύκολα μεταβαλλόμενες κατανομές φορτίου, αυτό μπορεί ακόμα, να μη δώσει την βέλτιστη διαδρομή, μια και μια διαδρομή βέλτιστη κάποια χρονική στιγμή, μπορεί να μην είναι βέλτιστη μια άλλη χρονική στιγμή. Μία εναλλακτική λύση είναι απλά να επιλέξει τη διαδρομή από το broadcast πλαίσιο που λαμβάνεται πρώτο, επειδή αυτό είναι πιθανό να έχει υποστεί την μικρότερη καθυστέρηση.

### 3. Σωστή χρήση του διαθέσιμου εύρους ζώνης

Για να εξασφαλιστεί η ενεργή τοπολογία του επικαλύπτοντος δένδρου, οι διαφανείς γέφυρες θα απομονώσουν κάποιες από τις θύρες τους. Έτσι τα προσαρτόμενα σε αυτές τις θύρες τμήματα δεν θα χρησιμοποιηθούν για την προώθηση πλαισίων από αυτή τη γέφυρα και κατά συνέπεια το συνολικό διαθέσιμο εύρος ζώνης, που παρέχεται απ' όλα τα τμήματα στο γεφυρωμένο LAN δεν θα χρησιμοποιείται διαρκώς.

Με τις γέφυρες δρομολόγησης πηγής, όλα τα διαθέσιμα τμήματα χρησιμοποιούνται κατά την διάρκεια της διαδικασίας εύρεσης διαδρομής (route-finding process), έτσι, θεωρητικά, το ολικό διαθέσιμο εύρος ζώνης θα χρησιμοποιηθεί. Ξανά, όμως, αυτό θα συμβεί, εάν οι διαδρομές που επιλέγονται από τους σταθμούς πηγής χρησιμοποιούν μια

στρατηγική που εξασφαλίζει μία ομαλή (even) φόρτωση όλων των τμημάτων του δικτύου. Στην πράξη είναι μάλλον απίθανο ότι θα συμβεί κάτι τέτοιο.

#### 4. Overhead επεξεργασία κατά την επιλογή διαδρομής

Γενικά μπορούμε να πούμε ότι, με μια διαφανή γέφυρα, η επεξεργασία του κάθε λαμβανόμενου πλαισίου θα διαρκεί περισσότερο από ότι με μια γέφυρα δρομολόγησης πηγής. Αυτό συμβαίνει γιατί η διαφανής γέφυρα πρέπει να διατηρήσει μια εγγραφή στη βάση δεδομένων της για κάθε σταθμό μέσα στο δίκτυο, ο οποίος είναι ενεργός, και ο οποίος σταθμός χρησιμοποιεί διαδρομές που περνούν μέσα από αυτή τη γέφυρα. Έτσι σε μεγάλα LAN, η βάση δεδομένων προώθησης μπορεί να έχει πολλές εγγραφές, με αποτέλεσμα ο χρόνος που χρειάζεται για την επεξεργασία κάθε πλαισίου (αυτό σημαίνει να καθορίσεις τη θύρα γέφυρας που σχετίζεται με τον σταθμό προορισμού ενός λαμβανόμενου πλαισίου) μπορεί να είναι ιδιαίτερα σημαντικός.

Για να ελαχιστοποιήσουμε τον αριθμό εισόδων μέσα στη βάση δεδομένων, ένας χρονιστής αδράνειας (inactivity timer) χρησιμοποιείται για κάθε εγγραφή. Εάν δεν φθάσει κάποιο νέο πλαισίο από έναν σταθμό μέχρι τη λήξη του χρονιστή, τότε η εγγραφή θεωρείται ότι δεν ισχύει και αφαιρείται. Ολοφάνερα, εάν κάποια είσοδος αφαιρεθεί, στην συνέχεια όταν ένας σταθμός πηγή (source station) στείλει ένα νέο πλαίσιο, το πλαίσιο αυτό θα πρέπει να μεταδοθεί σε όλες τις θύρες εξόδου της γέφυρας (broadcast) οι οποίες είναι σε κατάσταση προώθησης. Για να ελαχιστοποιήσουμε τον αριθμό τέτοιων broadcast μεταδόσεων ο αδρανής χρόνος είναι προσεκτικά επιλεγμένος ώστε να έχει λογική διάρκεια με την προϋπόθεση ότι η βάση δεδομένων προώθησης πρέπει να περιέχει έναν λογικό αριθμό εισόδων σε κάθε σημείο στο χρόνο.

Αντίθετα, σε μια γέφυρα δρομολόγησης πηγής, η επιπλέον επεξεργασία για κάθε πλαίσιο που απαίτεται από τη λειτουργία δρομολόγησης είναι μικρή από την στιγμή που η μόνη απαίτηση είναι το ψάξιμο του πεδίου δρομολόγησης στην επικεφαλίδα του κάθε πλαισίου.

Η επιπλέον επεξεργασία, δεν είναι ιδιαίτερα σημαντική σε τμήματα τοπικών δικτύων με χαμηλό ρυθμό μετάδοσης (για παράδειγμα 16Mbps). Όμως σε τοπικά δίκτυα με μεγαλύτερο ρυθμό μετάδοσης, όπως το FDDI, το οποίο λειτουργεί σε ρυθμούς μεγαλύτερους από τα 100Mbps, γίνεται απαραίτητο να ελαχιστοποιήθει η επιπλέον επεξεργασία σε κάθε πλαίσιο, εάν η χρήση του διαθέσιμου εύρους ζώνης μετάδοσης πρέπει να μεγιστοποιηθεί.

#### 4. Ικανότητα εύρεσης διαδρομής

Με τις διαφανείς γέφυρες, από τη στιγμή που μια ενεργή τοπολογία επικαλύπτοντος δένδρου έχει καθοριστεί, η γέφυρα θα αρχίσει να γεμίζει πολύ γρήγορα τη βάση δεδομένων της με τις κατάλληλες εισόδους, καθώς οι σταθμοί αρχίζουν να μεταδίδουν πλαίσια. Οι εγγραφές μέσα στη βάση δεδομένων πρέπει να είναι έχουν μείνει αρκετό καιρό στην θλεση τους για να εξασφαλίσουν ότι αναφέρονται σε σταθμούς, οι οποίοι τακτικά μεταδίδουν πλαίσια και έχουν μια διαδρομή μέσα από τη συγκεκριμένη γέφυρα. Το αποτέλεσμα είναι ότι ένα αντίγραφο από οποιαδήποτε πλαίσιο που φτάνει από έναν σταθμό για τον οποίο δεν υπάρχει πια ενεργή είσοδος στη βάση πληροφοριών της γέφυρας πρέπει να μεταδοθεί μέσω της γέφυρας σε κάθε άλλη θύρα. Εξαιτίας της ενεργής τοπολογίας του επικαλύπτοντος δένδρου, ο αριθμός πλαισίων που προκύπτει θα περιορισθεί σε ένα πλαίσιο για κάθε κλάδο του δένδρου που οδηγείται από αυτό το κόμβο (γέφυρας).

Σε αντίθεση, στις γέφυρες δρομολόγησης πηγής, αν και τα αρχικά πλαίσια μονής διαδρομής είναι εξαναγκασμένα να ακολουθήσουν ένα δένδρο συνδέσεων, τα επόμενα broadcast πλαίσια όλων των δυνατών διαδρομών που προέρχονται από τον προορισμό, δεν είναι. Αυτό σημαίνει ότι μπορεί να υπάρξει σημαντική επιπλέον επεξεργασία (από την άποψη, χρήσης του εύρους ζώνης μετάδοσης και της επεξεργασίας στις γέφυρες) που σχετίζονται με αυτά τα πλαίσια, ιδιαίτερα σε μεγάλα διασυνδεδεμένα δίκτυα αποτελούμενα από πολλά επιμέρους τοπικά δίκτυα και πολλές γέφυρες.

## 5. Θέματα αξιοπιστίας

Με ένα διαφανές γεφυρωμένο LAN, οι γέφυρες περιοδικά ελέγχουν για τυχόν σφάλματα σε γέφυρες ή συνδέσεις. Αυτό ουσιαστικά πραγματοποιείται από τον κόμβο ρίζα του δένδρου, ο οποίος μεταδίδει κάποια BPDUs σε τακτά χρονικά διαστήματα (καθορισμένα από ένα χρονιστή hello). Έαν κάποιο σφάλμα συμβεί, η τοπολογία αλλάζει, και ο αλγόριθμος του επικαλύπτοντος δένδρου, θα καθορίσει μια νέα ενεργή τοπολογία που θα χρησιμοποιηθεί από τις γέφυρες που έχουν μείνει.

Με τις γέφυρες πηγαίας δρομολόγησης, οι σταθμοί πηγής πρέπει να ερευνούν για τυχόν σφάλματα που συμβαίνουν στο δίκτυο, από την στιγμή που αυτοί κρατάνε την πληροφορία δρομολόγησης. Ολοφάνερα, όταν παρουσιαστεί ένα σφάλμα στο δίκτυο, η πληροφορία δρομολόγησης που κρατιέται από κάθε σταθμό πηγή μπορεί να είναι λανθασμένη με αποτέλεσμα να μεταδοθούν πλαίσια στα οποία εισάγεται λάθος πληροφορία για την δρομολόγησή τους. Αν τα παραπάνω οδηγήσουν σε απώλεια ενός πλαισίου, θα οδηγηθούμε σε λήξη κάποιου χρονιστή επιβεβαίωσης, συνήθως στο επίπεδο μεταφοράς, και τελικά θα έχουμε μετάδοση δεύτερου αντίγραφου του πλαισίου που είχε μεταδοθεί από τον σταθμό πηγής. Υποθέτοντας ότι το σφάλμα δεν διορθώθηκε ενδιάμεσα, τα πλαίσια αυτά από την αναμετάδοση θα φορτώσουν ανώφελα το δίκτυο. Έτσι είναι αναγκαίο για τους σταθμούς πηγής να ενεργούν σωστά (να ενημερώνουν τους πίνακες δρομολόγησής τους, ώστε να αντανακλούν την ανικανότητα ενός σταθμού ή κάποιας διαδρομής), όσο το δυνατόν πιο γρήγορα μετά την παρουσία ενός σφάλματος στο δίκτυο.

Μία προσέγγιση είναι, κάθε σταθμός πηγή να έχει ένα χρονιστή - πανομοιότυπο με τον χρονιστή αδράνειας που χρησιμοποιείται από τις γέφυρες - και ο οποίος να είναι υπεύθυνος για μία είσοδο του πίνακα δρομολόγησης. Τότε, εάν κάποιος χρονιστής για μία εγγραφή λήξει, αυτό θα σημαίνει ότι κανένα πλαίσιο δεν στάλθηκε στον συγκεκριμένο σταθμό προορισμού κατά την διάρκεια της περιόδου αυτής, οπότε μια νέα διαδρομή θα πρέπει να καθοριστεί πριν σταλθεί οποιοδήποτε νεώτερο πλαίσιο σε αυτόν τον σταθμό. Ολοφάνερα, όσο μικρότερος είναι αυτός ο χρόνος, τόσο λιγότερο πιθανόν είναι ότι πλαίσια με λανθασμένες πληροφορίες δρομολόγησης θα παραχθούν. Έτσι ένας χρόνος παρόμοιος μ' αυτόν που χρησιμοποιείται στις γέφυρες πρέπει να χρησιμοποιηθεί. Επειδή η λειτουργία αυτή θα πρέπει να πραγματοποιείται από κάθε σταθμό και όχι μονάχα από τις γέφυρες, το αποτέλεσμα, όσο αναφορά το επιπλέον φορτίο στο δίκτυο, θα είναι σημαντικά καλύτερο.

Μια εναλλακτική προσέγγιση είναι η χρήση της διαδικασίας αλλαγής τοπολογίας του αλγορίθμου του επικαλύπτοντος δένδρου. Από τη στιγμή που ο αλγόριθμος αυτός χρειάζεται για την δρομολόγηση πλαισίων broadcast εκπομπής μονής διαδρομής, θα μπορούσε επίσης να ελέγχει και την σχετική διαδικασία αλλαγής της τοπολογίας. Ας θυμηθούμε ότι σύμφωνα με την διαδικασία αυτή, ο κόμβος ρίζα μεταδίδει ένα topology change notification BPDU, κάθε φορά που μια αλλαγή στη τοπολογία προκαλείται από μία αποτυχία συνδέσμου ή γέφυρας. Αυτό χρησιμοποιείται από κάθε γέφυρα για να σβήσει ανάλογες εισόδους της από τη βάση δεδομένων θυρών προώθησης, έτσι ώστε αυτές να ενημερωθούν για να εκφράζουν (reflect) τη νέα τοπολογία. Επιπλέον κατά τη λήψη ενός τέτοιου BPDU, μια γέφυρα μπορεί να στείλει ένα αντίστοιχο πλαίσιο σε καθέναν από τους σταθμούς πάνω στα τμήματα τους, πληροφορώντας αυτά ότι παρουσιάστηκε

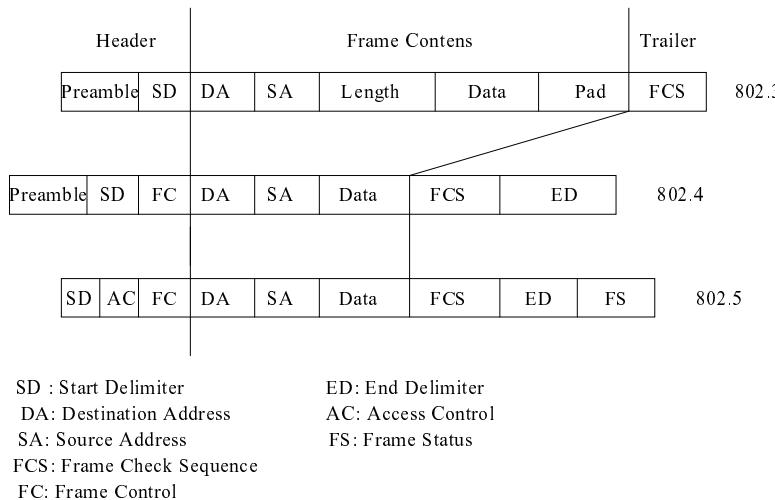
μια αλλαγή τοπολογίας. Αυτό στην συνέχεια θα οδηγήσει όλους τους σταθμούς να σβήσουν τις ανάλογες εισόδους τους στους πίνακες δρομολόγησης και με αυτό το τρόπο να μειώσουν σημαντικά το επιπλέον φορτίο στο δίκτυο.

## 7.8. Η γεφύρωση διαφορετικών δικτύων

Όπως αναφέρθηκε παραπάνω, εξαιτίας του τρόπου αποθήκευσης και προώθησης των πλαισίων, οι γέφυρες μπορούν να χρησιμοποιηθούν για να διασυνδέσουν ακόμα και τμήματα LAN που λειτουργούν με διαφορετικό τρόπο ελέγχου προσπέλασης του μέσου (MAC). Πρακτικά για πολλούς λόγους αυτό δεν είναι τόσο ξεκάθαρο όσο μπορεί να νομίζει κάποιος. Δύο βασικά προβλήματα είναι, η μορφή του πλαισίου, η οποίο μπορεί να είναι διαφορετική για κάθε διαφορετικό MAC, καθώς και η χωρητικότητα του δικτύου. Θα εξετάσουμε παρακάτω τέτοιου είδους προβλήματα.

### 1. Η μορφή του πλαισίου

Εξαιτίας του διαφορετικού τρόπου μετάδοσης, που χρησιμοποιείται στους τρεις βασικούς τύπους LAN - broadcast εκπομπή με τα 802.3 και 802.4 και από σημείο σε σημείο εκπομπή με το 802.5 - και των διαφορετικών τους μεθόδων ελέγχου προσπέλασης του μέσου - CSMA/CD για το 802.3 και κουπόνι για τα 802.4 και 802.5 - τα παραπάνω έχουν διαφορετικές μορφές πλαισίων όπως άλλωστε φαίνεται και στο Σχήμα 7.6. Για παράδειγμα, η χρήση broadcast εκπομπής από τα 802.3 και 802.4 σημαίνει ότι θα πρέπει να χρησιμοποιηθεί ένα προοίμιο (preamble) στο ξεκίνημα του κάθε πλαισίου, προκειμένου να επιτραπεί στο σταθμό λήψης να αποκτήσει συγχρονισμό ρολογιού (clock bit synchronization) πριν να επεξεργαστεί το περιεχόμενο του πλαισίου που έχει ληφθεί. Αυτό δεν είναι απαραίτητο για δίκτυο τοπολογίας δακτυλίου με κουπόνι, αφού τα τοπικά ρολόγια σε όλους τους σταθμούς μένουν συγχρονισμένα από τη συνεχή κυκλοφορούμενη ροή των bits.



### Σχήμα 7.6 - Μορφές πλαισίων 802.X τοπικών δικτύων

Παρόμοια, η χρήση ενός κουπονιού για τον έλεγχο προσπέλασης του μέσου σημαίνει ότι τα 802.4 και 802.5 έχουν ένα πεδίο ελέγχου στο πλαίσιο, που προηγείται από τα πεδία διευθύνσεων πηγής και προορισμού, αλλά και ένα τελικό (end delimiter) διαχωριστικό μετά από το FCS. Αντίθετα, ένα LAN 802.3, δεν χρησιμοποιεί αυτά τα πεδία. Χρησιμοποιεί, όμως, ένα πεδίο καθορισμού του μήκους και πιθανόν μερικά πρόσθετα

ακολουθούμενα bits (padding) για την διαμόρφωση ενός ελάχιστου μήκους στα μικρά πλαίσια.

Για να μπερδέψουμε τα πράγματα ακόμα περισσότερο, ένα δίκτυο δακτυλίου με κουπόνι χρησιμοποιεί επίσης ένα πρόσθετο πεδίο ελέγχου προσπέλασης στο ξεκίνημα του πλαισίου για να διευθύνει τα χαρακτηριστικά προτεραιότητας και κράτησης (reservation).

Συμπερασματικά, η παρουσία αυτών των πεδίων σημαίνει ότι, όταν ένα πλαίσιο περνάει από έναν τύπο τμήματος LAN σε ένα άλλο, θα πρέπει να ανασχηματιστεί πριν προωθηθεί στο διαφορετικού τύπου LAN. Κατ' αρχήν, αυτό δεν αποτελεί ένα αληθινό πρόβλημα αφού τα περισσότερα πεδία είτε προσθέτονται αυτόματα (είτε διαγράφονται) από τα ολοκληρωμένα κυκλώματα υλοποίησης του MAC υποεπιπέδου, πριν τα ακριβή περιεχόμενα πλαισίου μεταδοθούν (και αποθηκευτούν). Όμως, αυτό δεν ισχύει για τα πεδία μήκους και τα padding bits που χρησιμοποιούνται στα LANs 802.3, αφού αυτά θεωρούνται ως μέρος του περιεχομένου του πλαισίου από τα ολοκληρωμένα κυκλώματα. Αυτό σημαίνει ότι όσον αφορά τα 802.3 LANs, το ολικό περιεχόμενο του πλαισίου θα πρέπει να ανασχηματιστεί από το λογισμικό της γέφυρας που βρίσκεται πριν το πλαίσιο προωθηθεί.

Η ανάγκη να ανασχηματιστεί ένα πλαίσιο αυξάνει την επιπλέον επεξεργασία (overhead), και κατ' αυτό τον τρόπο την καθυστέρηση μέσα στη γέφυρα. Επιπρόσθετα, και πιο σημαντικά, αυτό σημαίνει ότι ένα νέο πεδίο FCS θα πρέπει να χρησιμοποιηθεί όταν το πλαίσιο προωθείται. Αυτό μπορεί να γίνει εύκολα, από τη στιγμή που είναι υπολογισμένο και προστιθέμενο από το chipset MAC. Όμως, μια δυναμική πηγή λαθών των γεφυρωμένων LAN δημιουργείται από πρόσθετα λανθασμένα bit που παρουσιάζονται μέσα στα πλαίσια, ενώ παράλληλα αποθηκεύονται και επανέρχονται από την μνήμη μέσα σε κάθε γέφυρα. Ξεκάθαρα, τέτοια λάθη θα περάσουν απαρατήρητα από τα νέα FCS.

Μια συνηθισμένη λύση σ' αυτό το πρόβλημα, όταν όλα τα τμήματα του LAN είναι του ίδιου τύπου, είναι να χρησιμοποιηθεί το ίδιο πεδίο FCS από την πηγή μέχρι τον προορισμό. Αυτό δεν είναι δυνατό, εάν το πλαίσιο πρέπει να ανασχηματιστεί από μια γέφυρα και σ' αυτή την περίπτωση κάθε λάθος bit που παρουσιάζεται (κατά την διάρκεια της επεξεργασίας και αποθήκευσης) θα περάσει απαρατήρητο. Αυτό (το λάθος) θα μεταφερθεί στην συνέχεια ως υπολειμματικό λάθος (residual). Για να ελαχιστοποιήσουμε αυτή την πιθανότητα συχνά χρησιμοποιείται στις γέφυρες μνήμη διόρθωσης λάθων (error correction memory).

## 2. Διαφορά στις τιμές των ρυθμών μετάδοσης

Ένα δεύτερο πρόβλημα στην γεφύρωση διαφορετικής τεχνολογίας τοπικών δικτύων με γέφυρες είναι ο ρυθμός μετάδοσης κάθε διαφορετικής τεχνολογίας. Μια σειρά διαφορετικών ρυθμών μετάδοσης μπορούν να χρησιμοποιηθούν από τα LAN. Η σειρά αυτή περιλαμβάνει:

- 802.3 - 1, 2, 10Mbps
- 802.4 - 1, 5, 10Mbps
- 802.5 - 1, 4, 16Mbps

Κανένα πρόβλημα δεν δημιουργείται, αν λαμβάνονται πλαίσια σ' ένα αργό τμήμα τοπικού δικτύου και πρόκειται να προωθηθούν σ' ένα γρηγορότερο. Εάν συμβαίνει όμως το αντίθετο και πολύ περισσότερο, εάν το LAN προορισμού είναι υπερφορτωμένο, μπορεί να προκύψει κάποιο πρόβλημα ως αποτέλεσμα των πλαισίων που συσσωρεύονται

στην θύρα εξόδου της γέφυρας, που μας οδηγεί στο πιο αργό LAN. Αυτό βέβαια θα ισχύει ακόμα κι αν τα δύο LAN είναι του ίδιου τύπου, αλλά διαφορετικής χωρητικότητας. Για παράδειγμα, εάν από δύο τμήματα LAN τοπολογίας αρτηρίας με κουπόνι, τα οποία είναι γεφυρωμένα, το ένα λειτουργεί στα 10Mbps και το άλλο στο 1Mbps, κατά την διάρκεια περιόδων μεγάλης κίνησης, θα παρουσιάζονται συγκεντρώσεις πλαισίων στην γέφυρα που τα συνδέει. Από τη στιγμή που το σύνολο της μνήμης που είναι διαθέσιμο περιορίζεται, η γέφυρα θα αρχίσει να απορρίπτει πλαίσια γιατί ο χώρος αποθήκευσης θα είναι πια ανεπαρκής. Πρακτικά, πρώτες οι οντότητες πρωτοκόλλου μεταφοράς στους σταθμούς πηγής, που επηρεάζονται, θα αναγνωρίσουν την απώλεια των πλαισίων και θα αρχίσουν την επαναμετάδοση άλλου αντίγραφου των πλαισίων αυτών. Η μεγάλη όμως παύση που σχετίζεται μ'αυτή την ενέργεια, σημαίνει ότι η καθυστέρηση μεταφοράς των πλαισίων θα αυξηθεί σημαντικά και επιπλέον δεν υπάρχει εγγύηση ότι τα νέα αντίγραφα δεν θα δοκιμάσουν την ίδια μοίρα, πράγμα που θα μας οδηγήσει σε ανεπιθύμητες καταστάσεις συμφόρησης.

## 7.9. Θέματα διαχείρισης γεφυρών

Αναφέραμε παραπάνω, ότι οι γέφυρες μεταξύ των άλλων επιτρέπουν την ευκολότερη και πιο αποτελεσματική διαχείριση ενός μεγάλου διασυνδεδεμένου δικτύου. Αυτό επιτυγχάνεται με την συγχώνευση λογισμικού διαχείρισης μαζί με το λογισμικό της γέφυρας (ύπαρξη agent διαχείρισης που τρέχει σαν δαιμόνας στην γέφυρα), οπότε όλα τα δεδομένα που αφορούν τις επιδόσεις ενός τμήματος διασυνδεδεμένου δικτύου μπορούν εύκολα να καταγράφονται και να ελέγχονται, όποτε είναι ανάγκη, με τη βοήθεια του δαιμονιά αυτού.

Επίσης, εφόσον έχουν χρησιμοποιηθεί μηχανισμοί ελέγχου πρόσβασης στην γέφυρα, για λόγους ασφάλειας του δικτύου, η τοπολογία του δικτύου μπορεί να αλλάξει δυναμικά με έλεγχο των πληροφοριών που αφορούν τις θύρες της κάθε γέφυρας από τον διαχειριστή του συστήματος.

Για μια ακόμη φορά θα δούμε το πλαίσιο αναφοράς της ISO διαχείρισης να εφαρμόζεται σε ένα συγκεκριμένο διαχειριστικό θέμα, αυτό της διαχείρισης γεφυρών διασύνδεσης τοπικών δικτύων. Οι συναρτήσεις τις οποίες υποστηρίζει η διαχείριση γεφυρών (Bridge Management) ελέγχουν και παρακολουθούν τις λειτουργίες προώθησης και φίλτραρισμάτος πλαισίων που παρέχουν οι γέφυρες (Frame Relaying and Frame Filtering). Συγκεκριμένα, όσο αφορά την διαχείριση διάρθρωσης (Configuration Management) έχουμε τα εξής:

- Δυνατότητα αναγνώρισης όλων των γεφυρών, που σχηματίζουν το γεφυρωμένο LAN, καθώς και των σχετικών τους θέσεων.
- Δυνατότητα αρχικοποίησης συγκεκριμένων γεφυρών.
- Δυνατότητα ελέγχου της προτεραιότητας, με βάση την οποία η γέφυρα μεταδίδει πλαίσια.
- Δυνατότητα επιβολής συγκεκριμένης μορφής επικαλύπτοντος δένδρου.
- Δυνατότητα ελέγχου της μετάδοσης πλαισίων με συγκεκριμένες ομαδικές MAC διευθύνσεις σε συγκεκριμένα μέρη του γεφυρωμένου LAN.

Οσο αφορά την διαχείριση σφαλμάτων (Fault Management) έχουμε τις εξής δυνατότητες:

- Δυνατότητες πρόληψης, αναγνώρισης, διάγνωσης και διόρθωσης λαθών.

- Δυνατότητες αναγνώρισης και διόρθωσης δυσλειτουργιών των γεφυρών.
- Δυνατότητες καταγραφής και αναφοράς λαθών.

Σχετικά με την διαχείριση επιδόσεων (Performance Management) υπάρχουν οι εξής δυνατότητες:

- Δυνατότητα αξιολόγησης της συμπεριφοράς και της αποτελεσματικότητας των γεφυρών.
- Δυνατότητες συλλογής στατιστικών δεδομένων για ανάλυση και εύρεση παραμέτρων σχετικών με τις επιδόσεις και το φορτίο.

Η διαχείριση γεφυρών δεν προσφέρει δυνατότητες κατάλληλες για λογιστική διαχείριση (Accounting Management). Ούτε το επίπεδο στο οποίο δουλεύουν οι γέφυρες, ούτε τα μηνύματα τα οποία ανταλλάσσουν επιτρέπουν την ανταλλαγή μηνυμάτων που να αφορούν πληροφορίες σχετικές με τη λογιστική διαχείριση.

Η διαχείριση γεφυρών δεν προσφέρει, κατάλληλες δυνατότητες για τη διαχείριση ασφάλειας του δικτύου (Security Management).

## 7.10. Δρομολογητές (Routers), σύγκριση με τις γέφυρες

Ένας δρομολογητής λειτουργεί στο επίπεδο δικτύου - το 3ο επίπεδο του μοντέλου αναφοράς πρωτοκόλλων OSI. Ο δρομολογητής διαφέρει από μια γέφυρα στο ότι η γέφυρα εκτελεί την δρομολόγηση με διαφανή τρόπο, ενώ αντίθετα ο δρομολογητής έχει δική του διεύθυνση επιπέδου δικτύου και σ' αυτόν απευθύνονται (χρησιμοποιώντας την διεύθυνση αυτή) οι σταθμοί που χρειάζονται υπηρεσίες δρομολόγησης. Με τον τρόπο αυτό, οι σταθμοί του τοπικού δικτύου μπορούν να ξεχωρίσουν κάποια επικοινωνία με σταθμό του ίδιου τοπικού δικτύου από επικοινωνία με σταθμό διαφορετικού δικτύου.

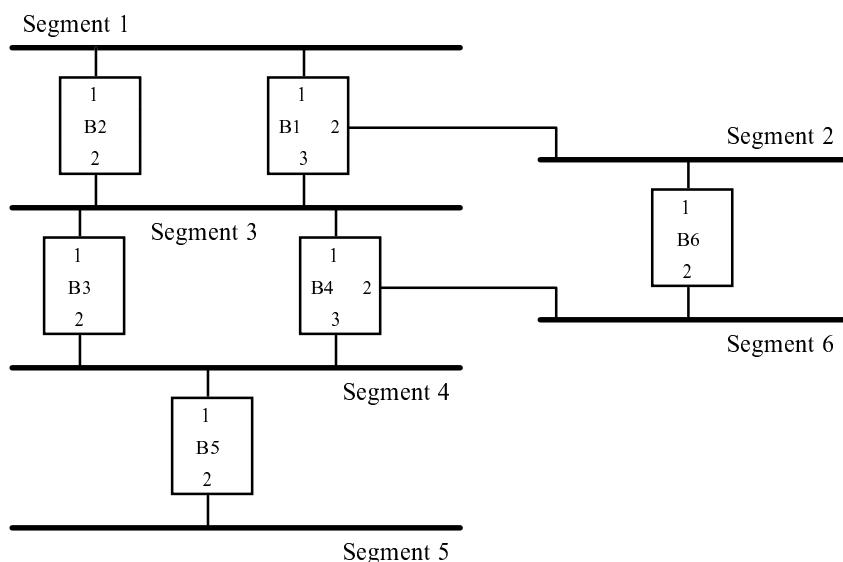
Το γεγονός ότι οι δρομολογητές λειτουργούν στο επίπεδο δικτύου τους υποχρεώνει να χρησιμοποιούν συγκεκριμένο πρωτόκολλο, σε αντίθεση με τις γέφυρες. Βέβαια, ενώ υπάρχουν δρομολογητές που συνδέουν δίκτυα της ίδιας οικογένειας πρωτοκόλλων, υπάρχουν και άλλοι που υποστηρίζουν περισσότερα πρωτόκολλα (multiprotocol router) όπως DECnet, XNS, TCP/IP και OSI.

Γενικά ο δρομολογητής προσφέρει ένα υψηλότερο επίπεδο αξιοπιστίας στις υπηρεσίες διασύνδεσης και δρομολόγησης από αυτό που προσφέρει μια γέφυρα. Κάποιος δρομολογητής μπορεί να επιλέξει ένα μονοπάτι μεταξύ πολλών προκειμένου να προωθήσει κάποιο πακέτο και να βασιστεί πάνω σε πολλές παραμέτρους, όπως καθυστέρηση μεταφοράς, επίπεδο συμφόρησης σε άλλους δρομολογητές, αριθμός ενδιάμεσων συνδέσμων κ.ά., ανάλογα βέβαια με τους αλγορίθμους δρομολόγησης, που χρησιμοποιεί. Επίσης, μπορεί να μοιράσει το φορτίο μεταξύ όλων των δυνατών μονοπατιών που συνδέουν δύο δίκτυα.

Κάποιος δρομολογητής μπορεί να επικοινωνήσει με άλλους κόμβους μέσω καταλλήλων μηνυμάτων ελέγχου, περιορίζοντας, με τον τρόπο αυτό, τον ρυθμό τους παραγωγής πακέτων (π.χ. μηνύματα ICMP στα TCP/IP πρωτόκολλα), δηλαδή μπορεί να ασκήσει κάποια μορφή ελέγχου ροής.

## 7.11. Ασκήσεις

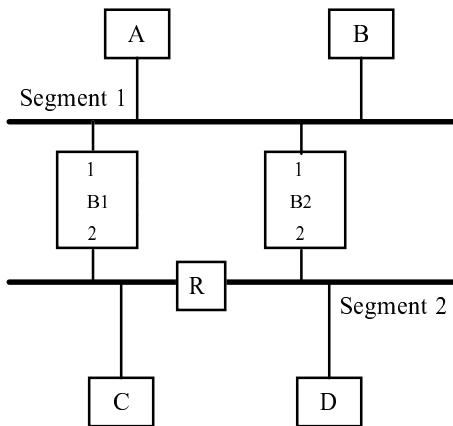
- [1]. Ποια είναι τα πλεονεκτήματα και τα μειονεκτήματα των γεφυρών σε σύγκριση με τους επαναμεταδότες;
- [2]. Συγκρίνατε τη λειτουργικότητα δίθυρων και πολύθυρων γεφυρών. Μπορεί κάθε διάρθρωση δίκτυου υλοποιημένη με πολύθυρες γέφυρες να υλοποιηθεί και με δίθυρες γέφυρες; Υποθέστε ότι ν τοπικά δίκτυα έχουν διασυνδεθεί με μία ν-θυρη γέφυρα. Σε μια τέτοια περίπτωση κάθε πακέτο μπορεί να φθάσει από το ένα δίκτυο στο άλλο σε ένα βήμα. Είναι κάτι τέτοιο δυνατό αν είχαν χρησιμοποιηθεί δίθυρες γέφυρες για τη διασύνδεση των ν τοπικών δικτύων;
- [3]. Στο δίκτυο του παρακάτω σχήματος να εφαρμοστεί ο αλγόριθμος του επικαλύπτοντος δέντρου. Το μοναδικό αναγνωριστικό κάθε γέφυρας εμφανίζεται στο σχήμα της μαζί με τους αριθμούς των θυρών που συνδέουν τη γέφυρα σε κάθε τοπικό δίκτυο.
- α) Να εξετάσετε την περίπτωση βλάβης της γέφυρας B1.
- β) Τι θα συμβεί αν με παρέμβαση του διαχειριστικού συστήματος η γέφυρα B2 αποκτήσει τη μεγαλύτερη προτεραιότητα;



- [4]. Συγκρίνατε τους αλγορίθμους δρομολόγησης σε γεφυρωμένα τοπικά δίκτυα: δρομολόγησης πηγής και επικαλύπτοντος δέντρου. Ενδιαφέρον παρουσιάζουν

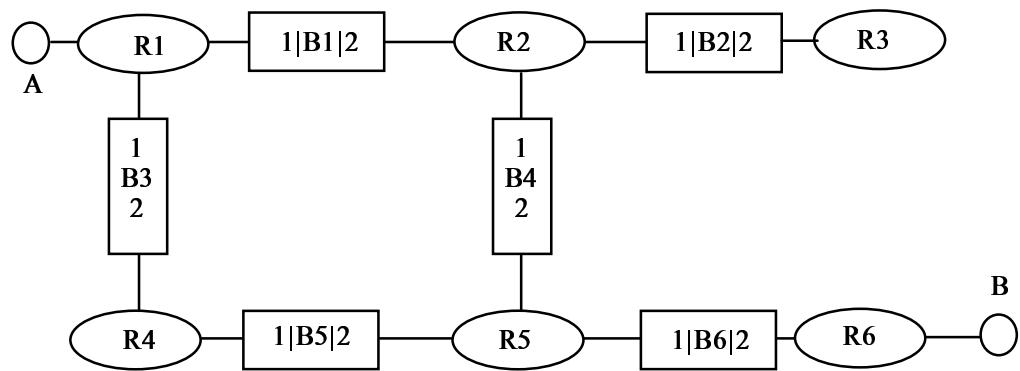
θέματα όπως η αξιοπιστία, η χρησιμοποίηση του διαθέσιμου εύρους ζώνης, η φιλοσοφία, η ικανότητα επιλογής διαδρομής, κ.ά.

- [5]. Δώστε ένα παράδειγμα δημιουργίας καταιγίδων (storm) με βλάβη και επανόρθωση του αναμεταδότη (repeater) R, εξηγήσατε πως αυτές επηρεάζουν τη λειτουργία του δικτύου και πως σταματούν;



- [6]. Να αναφέρετε και να αναλύσετε προβλήματα που προκύπτουν κατά τη διασύνδεση τοπικών δικτύων υπολογιστών που υποστηρίζουν διαφορετικά πρωτόκολλα στο επίπεδο ελέγχου προσπέλασης του μέσου (MAC).
- [7]. Να συγκρίνετε διεξοδικά τις γέφυρες και τους δρομολογητές σαν λύσεις για τη διασύνδεση τοπικών δικτύων υπολογιστών. Ποια είναι τα πλεονεκτήματα και ποια τα μειονεκτήματα κάθε τεχνικής;
- [8]. Θεωρείστε τη λειτουργία δύο γεφυρών οι οποίες συνδέουν τοπικά δίκτυα υπολογιστών (LANs), που χρησιμοποιούν ίδια πρωτόκολλα. Η πρώτη γέφυρα οφείλει να προωθήσει 1000 πλαίσια των 500 χαρακτήρων το δευτερόλεπτο, ενώ η δεύτερη 500 πλαίσια των 1000 χαρακτήρων το δευτερόλεπτο. Για ποια από τις δύο, ο διαχειριστής πρέπει να μεριμνήσει για ένα γρήγορο μηχάνημα; Εξηγήστε.
- [9]. Για τα διασυνδεδεμένα τοπικά δίκτυα τοπολογίας δακτυλίου του σχήματος, υποθέστε ότι ο σταθμός A θέλει να επικοινωνήσει με τον σταθμό B.
- (1) Βρείτε το επικαλύπτων δένδρο.

(2) Χρησιμοποιείστε τον αλγόριθμο δρομολόγησης πηγής για να βρείτε τον δρόμο που θα ακολουθήσουν τα πλαίσια.



## 7.12. Βιβλιογραφία

- [COME91] Comer D.E., *Internetworking with TCP/IP Volume I; Principles, Protocols, and Architecture*, Second Edition, Prentice Hall, N.J., 1991.
- [HALS92] Halsall F., Data Communications, Computer Networks, and Open Systems, Third Edition, Addison Wesley, 1992.
- [ΠΟΜΠΙ90] Πομπόρτσης Α., Τοπικά Δίκτυα Υπολογιστών, Θεσσαλονική 1990.
- [RACE91] Report on Routers and Bridges, 91/BT/ESP/DS/A/001/b1, RACE-Project 1091, Version 01, 28 May 1991.
- [RADI92] Perlman R., *Interconnections: Bridges and Routers*, Addison-Wesley Professional Computing Series, 1992.

## Κεφάλαιο 8

### 8. Αρχιτεκτονικές και Ανάπτυξη Συστήματων Διαχείρισης

#### Περιεχόμενα του Κεφαλαίου 8

- 8.0. Εισαγωγή
- 8.1. Γενικά χαρακτηριστικά ενός ΣΔΔ
- 8.2. Αρχιτεκτονικές ΣΔΔ
  - 8.2.1. Κεντροποιημένο ΣΔΔ
  - 8.2.2. Κατανεμημένο ΣΔΔ
  - 8.2.3. Ιεραρχικό ΣΔΔ
  - 8.2.4. Δικτυωμένο ΣΔΔ
- 8.3. Συμμόρφωση με τα πρότυπα - Απαιτήσεις ενός ΣΔΔ
- 8.4. Αντικειμενοστραφής φιλοσοφία και διαχείριση δικτύων
  - 8.4.1. Τι είναι η αντικειμενοστραφής φιλοσοφία
  - 8.4.2. Αντικείμενα και MIB
    - 8.4.2.1. SNMP MIB
    - 8.4.2.2. CMIP MIB
  - 8.4.3. Εφαρμογή σε ΣΔΔ
  - 8.4.4. Πρότυπα - Τυποποιήσεις
- 8.5. Ανάπτυξη του λογισμικού ενός ΣΔΔ
- 8.6. Έμπειρα συστημάτων
  - 8.6.1. Έμπειρα συστήματα και διαχείριση δικτύων
  - 8.6.2. Τι κάνει ένα έμπειρο σύστημα
  - 8.6.3. Κατηγορίες έμπειρων συστημάτων
  - 8.6.4. Στοιχεία που αποτελούν το έμπειρο σύστημα
  - 8.6.5. Υλοποίηση εμπείρων συστημάτων
- 8.7. Ασκήσεις
- 8.8. Βιβλιογραφία

#### 8.0. Εισαγωγή

Σύμφωνα με το μοντέλο διαχείρισης που έχει καθιερωθεί, ένα σύστημα διαχείρισης δικτύων (ΣΔΔ) αποτελείται από τρία λειτουργικά μέρη [ROSE91] :

- (α) τους διαχειριζόμενους κόμβους (managed nodes) του δικτύου και τους agents διαχείρισης που τρέχουν σε αυτούς,
- (β) ένα ή περισσότερους σταθμούς διαχείρισης (network management station - manager),
- (γ) ένα πρωτόκολλο διαχείρισης, μέσω του οποίου ο κάθε σταθμός διαχείρισης παρακολουθεί, ελέγχει και αντλεί πληροφορίες από το δίκτυο.

Λεπτομέριες για τα παραπάνω, καθώς και παρουσίαση και σύγκριση δύο πρωτοκόλλων διαχείρισης (CMIP/CMOT - SNMP), δίνονται στο κεφάλαιο 3. Στο κεφάλαιο αυτό περιγράφονται οι λειτουργικές απαιτήσεις και οι περιορισμοί ενός συστήματος διαχείρισης δικτύων και παρουσιάζονται αρχιτεκτονικές και τεχνικές σχεδίασης ενός τέτοιου συστήματος.

## 8.1. Γενικά χαρακτηριστικά ενός ΣΛΔ

Λαμβάνοντας υπόψη τις λειτουργίες που ένα διαχειριστικό σύστημα απαιτείται να υποστηρίζει, τα παρακάτω γενικά χαρακτηριστικά - περιορισμοί ενισχύουν την λειτουργικότητα του συστήματος [LEIN91] :

- Το σύστημα πρέπει να παρέχει ένα γραφικό σύστημα παρουσίασης της τοπολογίας του δικτύου. Είναι προτιμότερο η παρουσίαση να γίνεται με ιεραρχικό τρόπο και να υπάρχουν λογικές συνδέσεις μεταξύ των διαφορετικών επιπέδων της ιεραρχίας. Για παράδειγμα, σε ένα επίπεδο παρουσιάζονται μόνο τα LANs και οι συνδέσεις μεταξύ τους, ενώ σε κατώτερο επίπεδο παρουσιάζονται τα τμήματα (segments) του κάθε LAN, στο επόμενο επίπεδο οι κόμβοι των segments κ.ο.κ. Πρέπει ακόμη το σύστημα να είναι σε θέση να αναγνωρίζει τις συνδέσεις μεταξύ των επιπέδων και το πως αυτές συσχετίζονται με την απόδοση και την λειτουργία ολόκληρου του δικτύου. Η ενοποιημένη εικόνα του διαχειριζόμενου δικτύου διατηρείται από το σύστημα, ενώ ο χρήστης μπορεί να επικεντρώνει την προσοχή του σε ορισμένα επίπεδα της ιεραρχίας. Είναι λειτουργικό, τέλος, να υπάρχει ομογενής αντιμετώπιση των στοιχείων του δικτύου σε επίπεδο διαπροσωπείας χρήστη, έστω και αν εσωτερικά υπάρχει ετερογένεια. Για παράδειγμα, σταθμοί εργασίας που διαχειρίζονται με διαφορετικά πρωτόκολλα πρέπει να παρουσιάζονται με τον ίδιο τρόπο στον χρήστη, και οι μέθοδοι άντλησης πληροφοριών για αυτούς να είναι όσο το δυνατόν παρόμοιοι. Οι ανομοιογένειες πρέπει να κρύβονται από τον χρήστη, εκτός βέβαια αν ζητηθούν ή αποτελούν αιτία προβλημάτων.
- Το σύστημα πρέπει να είναι ικανό να συλλέγει όλες τις πληροφορίες από τους διαχειριζόμενους κόμβους, με όσο είναι δυνατόν μεγαλύτερη διαφάνεια και ιδανικά μέσω ενός μόνο πρωτοκόλλου διαχείρισης. Βέβαια, σε ετερογενή περιβάλλοντα το σύστημα πρέπει να είναι σε θέση να χρησιμοποιεί διαφορετικά πρωτόκολλα διαχείρισης και/ή proxy agents.
- Η επεκτασιμότητα (expandability) και η δυνατότητα προσαρμογής σε διαφορετικές ανάγκες διαχείρισης (customization) είναι δύο ακόμη σημαντικά χαρακτηριστικά - απαιτήσεις. Δεν υπάρχει σύστημα που να καλύπτει τις ανάγκες διαχείρισης κάθε δυνατού δικτύου. Έτσι το σύστημα πρέπει να επιτρέπει την εύκολη προσθήκη νέων δυνατοτήτων και εργαλείων διαχείρισης ανάλογα με τις απαιτήσεις της κάθε εφαρμογής. Χαρακτηριστικό παράδειγμα τέτοιου συστήματος είναι το Cerberus Network Management System [STAM92] που παρουσιάζεται στο κεφάλαιο 11.

- Μια ακόμη βασική λειτουργία ενός συστήματος διαχείρισης είναι η δυνατότητα ανίχνευσης και αναφοράς λαθών και προβλημάτων στο δίκτυο. Καθώς το διαχειριζόμενο δίκτυο επεκτείνεται, μια τέτοια υπηρεσία γίνεται όλο και περισσότερο πολύτιμο. Έστω και αν η διαχείριση λαθών δεν υποστηρίζεται, η ανίχνευση και η ειδοποίηση είναι απαραίτητα χαρακτηριστικά ενός διαχειριστικού συστήματος.
- Το σύστημα πρέπει να παρέχει ένα αποδοτικό τρόπο φύλαξης του όγκου πληροφοριών που χρειάζεται για την διαχείριση, ιδιαίτερα όταν τα διαχειριζόμενα δίκτυα είναι μεγάλα. Συχνά ένα σύστημα διαχείρισης βάσης δεδομένων (DBMS) είναι απαραίτητο καθώς εφαρμογές που configuration και accounting management είναι αδύνατον να λειτουργήσουν αποδοτικά χωρίς αυτό. Συνήθως χρησιμοποιείται το σχεσιακό μοντέλο (relational data model), ενώ γίνονται προσπάθειες να σχεδιαστούν και να υλοποιηθούν αντικειμενοστραφείς βάσεις δεδομένων (Object Oriented DBMS - OODBMS) ειδικά για χρήση σε συστήματα διαχείρισης δικτύων. Ανάλογα με την αρχιτεκτονική του ΣΔΔ (βλέπε παράγραφο 8.2.) και τις απαιτήσεις απόδοσης μπορεί να χρησιμοποιηθεί κεντροποιημένη (centralized) ή κατανεμημένη (distributed) βάση δεδομένων.

Έχουν διατυπωθεί και άλλοι περιορισμοί που αφορούν την αλληλεπίδραση διαχειριζόμενου δικτύου και διαχειριστικού συστήματος :

- Διατυπώνουν μια μινιμαλιστική φιλοσοφία στις επιδράσεις του ΣΔΔ στο διαχειριζόμενο δίκτυο που συνοψίζεται στο εξής : "Το αποτέλεσμα της εγκατάστασης ενός ΣΔΔ σε ένα δίκτυο πρέπει να είναι το ελάχιστο δυνατό, αντανακλώντας τον ελάχιστο κοινό παρανομαστή" [ROSE91]. Η ανάγκη ελάχιστης επιροής στους διαχειριζόμενους κόμβους ενισχύεται από τις μεγάλες διαφορές μεταξύ των κόμβων [CASE89]. Η διαδικασία άντλησης πληροφοριών και παρακολούθησης των κόμβων δεν πρέπει να προκαλεί σημαντικές καθυστερήσεις στην λειτουργία των κόμβων, καθώς κατι τέτοιο οξύνει τις διαφορές απόδοσης. Τέλος, το φορτίο που εισάγει στο δίκτυο η λειτουργία του ΣΔΔ πρέπει να είναι όσο το δυνατόν μικρότερο, αλλιώς το κέρδος της δυνατότητας διαχείρισης, αντισταθμίζεται από την παρενέργεια της πεσμένης απόδοσης και των προβλημάτων που μπορεί να προκαλέσει η συμφόρηση του δίκτυου.
- Μια άλλη απαίτηση είναι η βιωσιμότητα του διαχειριστικού συστήματος σε κρίσιμες καταστάσεις. Όταν το διαχειριζόμενο δίκτυο "πέφτει" και γενικά σε καταστάσεις σημαντικών προβλημάτων και λαθών, το ΣΔΔ πρέπει να παραμείνει σε λειτουργία (σε όποιο βαθμό είναι αυτό δυνατό) [ROSE91]. Όσο περισσότερο ανεκτικό στα λάθη του διαχειριζόμενου δικτύου είναι το ΣΔΔ, τόσο καλύτερα εκπληρώνει τον ρόλο του σε περιπτώσεις προβλημάτων.

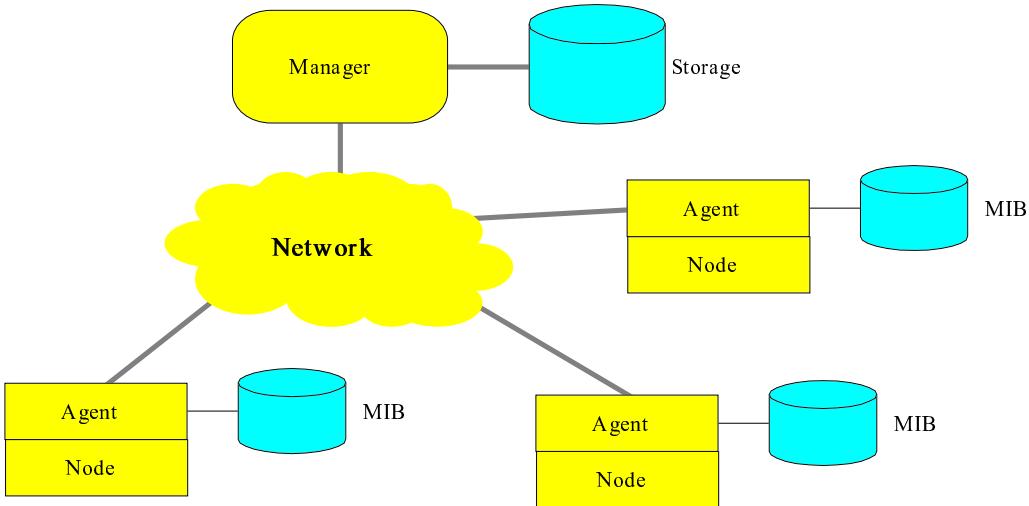
## 8.2. Αρχιτεκτονικές ΣΔΔ

Τρεις αρχιτεκτονικές προς το παρόν υπερισχύουν [LEIN93] [HERM90] : (α) κεντροποιημένο ΣΔΔ, (β) κατανεμημένο ΣΔΔ και (γ) ιεραρχικό ΣΔΔ. Μια παραλλαγή τους που συνδιάζει τα δύο τελευταία είναι το (δ) δικτυωμένο ΣΔΔ. Οι διαφορές αυτών των αρχιτεκτονικών αναφέρονται κυρίως στον αριθμό διαχειριστών και στον βαθμό επικοινωνίας - ανεξαρτησίας τους. Κάθε μια προσφέρει κάποια πλεονεκτήματα και μειονεκτήματα εναντί των άλλων. Η επιλογή εξαρτάται από της απαιτήσεις διαχείρισης και τον χαρακτήρα του δίκτυου που απαιτείται η διαχείριση.

Γενική περιγραφή των τεσσάρων αρχιτεκτονικών, των πλεονεκτημάτων και μειονεκτημάτων τους δίνεται παρακάτω.

### 8.2.1. Κεντροποιημένο ΣΔΔ [ROSE91] [CASS89]

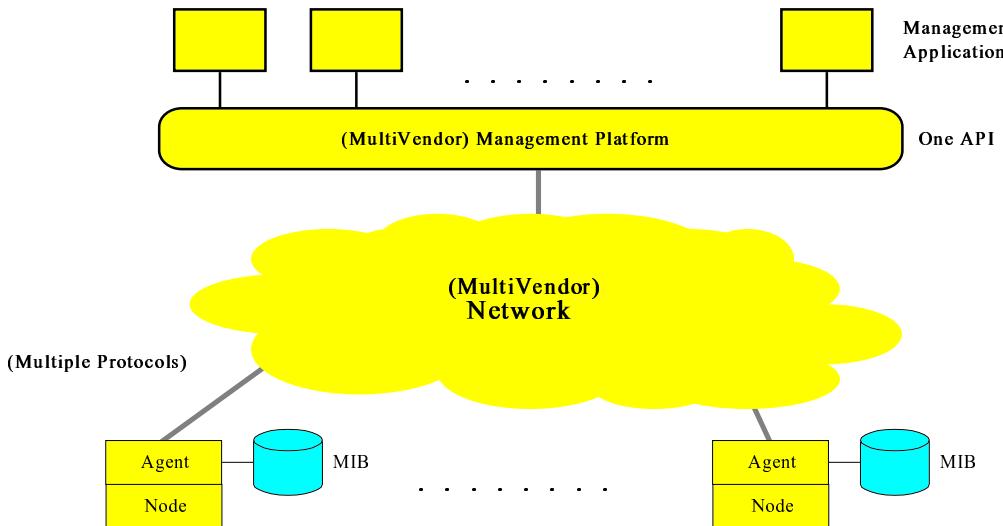
Η αρχιτεκτονική αυτή παρουσιάζεται στο σχήμα 8.1 και ακολουθεί το γνωστό μοντέλο διαχειριστή - αντιπροσώπου (manager - agent) που περιγράφεται στο κεφάλαιο 3. Είναι η απλούστερη και πιό κλασική αρχιτεκτονική ΣΔΔ και παρουσιάζει όλα τα πλεονεκτήματα και μειονεκτήματα ενός κεντροποιημένου συστήματος.



**Σχήμα 8.1 - Κεντροποιημένο ΣΔΔ**

Το ΣΔΔ περιέχει μόνο ένα κεντρικό διαχειριστή, ο οποίος αναλαμβάνει την επικοινωνία με όλους τους διαχειριζόμενους κόμβους (μέσω των agents και του πρωτοκόλλου διαχείρισης), διαχειρίζεται την αποθήκευση των πληροφοριών του ΣΔΔ και παρέχει μια ενοποιημένη εικόνα του διαχειριζόμενου δικτύου στον χειριστή του ΣΔΔ μέσω κατάλληλης διαπροσωπείας χρήστη (user interface). Η αποθήκευση των δεδομένων διαχείρισης (RDBMS, OODBMS ή άλλη απλούστερη μέθοδος) μπορεί να είναι κεντροποιημένη ή κατανεμημένη, αλλά ο έλεγχος είναι καθαρά κεντροποιημένος όπως και όλη η φιλοσοφία αυτής της αρχιτεκτονικής.

Μια παραλλαγή αυτής της αρχιτεκτονικής, η "μέθοδος της πλατφόρμας διαχείρισης" (The platform approach) [HERM90] [STAM92], φαίνεται στο σχήμα 8.2. Εδώ ο κεντρικός διαχειριστής χωρίζεται σε δύο μέρη : τις διαχειριστικές εφαρμογές και την κεντροποιημένη πλατφόρμα διαχείρισης. Τα δύο αυτά τμήματα βρίσκονται σε σχέση πελάτη - εξυπηρετητή, ένα μοντέλο που κερδίζει όλο και περισσότερο έδαφος τα τελευταία χρόνια [GUTT93]. Η πλατφόρμα αναλαμβάνει την επικοινωνία με το διαχειριζόμενο δίκτυο, την παρακολούθηση και συλλογή πληροφοριών και παρέχει υπηρεσίες διαχείρισης στις εφαρμογές - πελάτες. Η επικοινωνία πλατφόρμας - εφαρμογών/πελατών γίνεται μέσω ενός κοινού για όλες τις εφαρμογές API (Application Programming Interface) που προσφέρει η πλατφόρμα. Αντιθέτα η επικοινωνία μεταξύ πλατφόρμας - agents μπορεί να γίνεται με πολλά διαφορετικά πρωτόκολλα όταν το δίκτυο είναι ετερογενές [HERM90].



**Σχήμα 8.2 - Μέθοδος της πλατφόρμας διαχείρισης**

Η αρχιτεκτονική της πλατφόρμας διαχείρισης προσφέρει ενοποιημένη διαχείριση σε ετερογενή ή ομοιογενή περιβάλλοντα, απαιτώντας μικρό αριθμό διαχειριστικών συστημάτων σε σχέση με την αρχιτεκτονική του Ιεραρχικού ΣΔΔ (βλέπε παρακάτω), αυξάνοντας, ίσως, την πολυπλοκότητα στην υλοποίηση. Διευκολύνει, επίσης, την ανάπτυξη ολοκληρωμένων (integrated) εφαρμογών που "βλέπουν" ετερογενή τμήματα του δικτύου, χωρίς να αντιμετωπίζουν από την αρχή το πρόβλημα της ετερογένειας, καθώς η πλατφόρμα κρύβει την λεπτομέρια από την εφαρμογή-πελάτη και προσφέρει τυποποιημένο και ομοιογενές interface. Η ενοποιημένη αναπαράσταση και η ομοιογενής αντιμετώπιση των στοιχείων του δικτύου επιτρέπει την δημιουργία φιλικότερων και λειτουργικότερων τρόπων παρουσίασης του δικτύου στον χρήστη (αντικειμενοσταφής φιλισοφία στη διαπροσωπεία χρήστη, βλέπε παράγραφο 8.4).

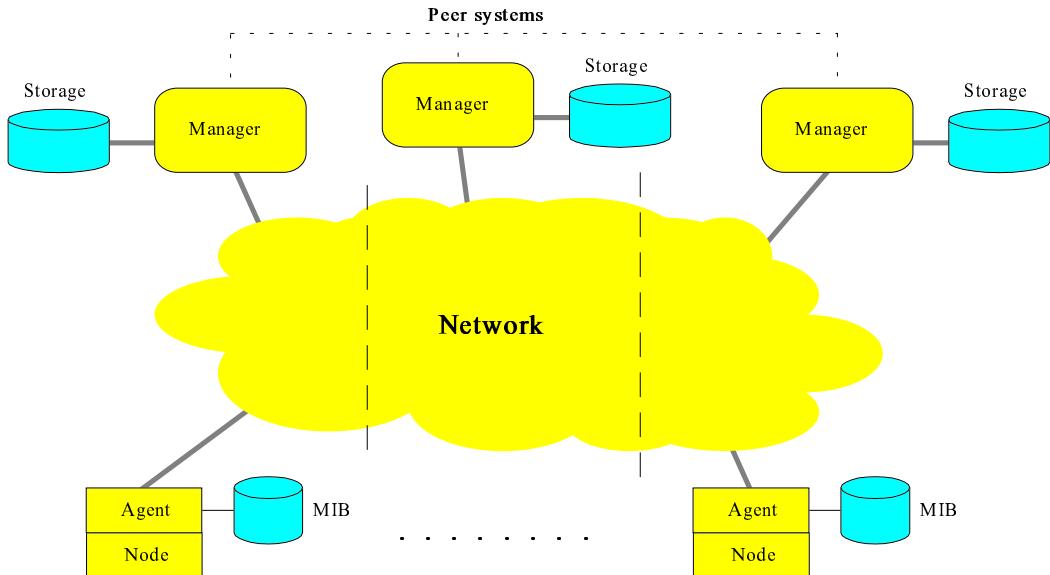
Τέλος, η αρχιτεκτονική αυτή δίνει την δυνατότητα ευκολότερης συντήρησης του λογισμικού του ΣΔΔ. Βελτιώσεις και αλλαγές στην πλατφόρμα αυτόματα επιδρούν σε όλες τις εφαρμογές-πελάτες που έχουν γραφτεί, χωρίς να χρειάζεται συντήρηση του λογισμικού τους. Για παράδειγμα, αν στο υπό διαχείριση δύκτυο προστεθούν νέοι κόμβοι που απαιτούν διαφορετικά πρωτόκολλα επικοινωνίας, η παρουσιάζουν κάποια άλλη μορφή ετερογένειας, για να προστεθεί δυνατότητα διαχείρισή τους στο ΣΔΔ, αρκεί να επεκταθεί η πλατφόρμα-εξυπηρετητής και όχι οι εφαρμογές πελάτες. Έτσι, είναι καλή τακτική στην σχεδίαση μιας τέτοιας αρχιτεκτονικής, η πλατφόρμα να περιέχει κυρίως τα τμήματα λογισμικού που αφορούν τις λεπτομέριες επικοινωνίας, της χρήσης διαφορετικών πρωτοκόλλων και drivers, της ενοποιημένης αναπαράστασης των στοιχείων του δικτύου και γενικά όλα όσα κοινά σε όλες τις εφαρμογές και μόνο αυτά. Κάτι τέτοιο οδηγεί σε μικρό κώδικα για την πλατφόρμα που συνεπάγεται ευκολότερη συντήρηση.

Κατασκευαστές ΣΔΔ που έχουν παρουσιάσει προϊόντα που ακολουθούν αυτή την αρχιτεκτονική είναι οι : HP, DEC, Sun, 3Com, IBM.

### 8.2.2. Κατανεμημένο ΣΔΔ [SMGO92] [LEIN93]

Η αρχιτεκτονική αυτή παρουσιάζεται στο σχήμα 8.3. Ο έλεγχος κατανέμεται σε ομότιμους διαχειριστές (peer managers) που διαχειρίζονται τμήματα του δικτύου που πιθανόν και να επικαλύπτονται (έννοια του manager domain - βλέπε κεφάλαιο 4). Η

αποθήκευση των πληροφοριών διαχείρισης μπορεί να γίνεται σε διαφορετικές βάσεις δεδομένων, ή σε μια κεντρική βάση δεδομένων (κατανεμημένη ή κετροποιημένη). Η φιλοσοφία, όμως, είναι κατανεμημένη και κανένας διαχειριστής δεν έχει πλήρη εικόνα του δικτύου (αν δεν επικοινωνήσει με κάποιο ομότιμο σύστημα).



### Σχήμα 8.3 - Κατανεμημένο ΣΛΔ

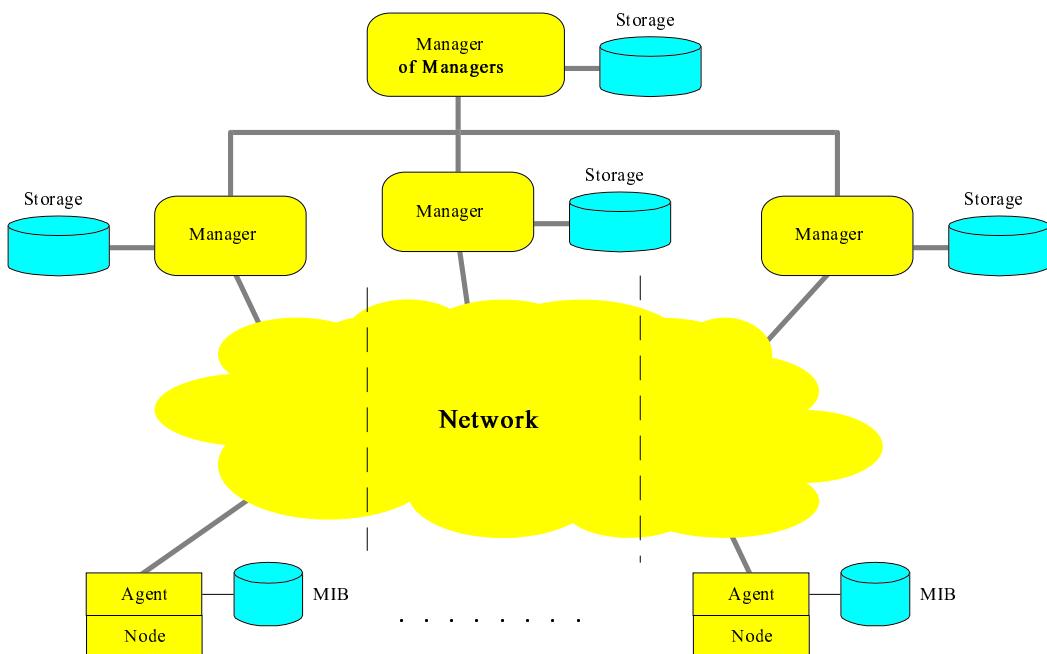
Η διαχείριση κατανέμεται και αντιμετωπίζεται τοπικά από διαχειριστές με λιγότερες απαιτήσεις σε hardware και υπολογιστική δύναμη, σε σχέση με τον κετροποιημένο διαχειριστή. Επιπλέον, ο κάθε τοπικός διαχειριστής απαλαγμένος από το βάρος της παρακολούθησης ολόκληρου του δικτύου, μπορεί να χειρίστει με μεγαλύτερη λεπτομέρια το μικρότερο domain του. Όταν κάποιες πληροφορίες χρειαστούν για περιοχές του δικτύου εκτός αυτού του domain, ο διαχειριστής τις ζητάει από τον αντίστοιχο ομότιμο του. Η επικοινωνία αυτή μπορεί να γίνεται μέσω του διαχειριζόμενου δικτύου, ή όταν υπάρχουν υψηλές απαιτήσεις αξιοπιστίας, μέσω ενός ανεξάρτητου δικτύου διαχείρισης. Ακόμη, μπορεί να γίνεται μέσω μηνυμάτων (request/response) ή με το να μοιράζονται όλοι οι διαχειριστές ένα κοινό τρόπο αποθήκευσης πληροφορίων (βάση δεδομένων ή άλλο τρόπο). Στην περίπτωση μιας κοινής βάσης δεδομένων, οι απαιτούμενες πληροφορίες μπορούν να ληφθούν στέλνοντας τα ανάλογα queries στο σύστημα διαχείρισης της ΒΔ (DBMS).

Μια τέτοια αρχιτεκτονική μπορεί να εφαρμοστεί σε αρκετά μεγάλα δίκτυα με σχετικά ανεξάρτητα αυτοδιαχειριζόμενα μέρη. Για παράδειγμα, σε ένα μεγάλο δίκτυο που απλώνεται γεωγραφικά, ένας διαχειριστής αναλαμβάνει το τμήμα που βρίσκεται στην Ευρώπη, άλλος το τμήμα της Αμερικής κ.ο.κ.

Τέλος, κάποιες τυποποιήσεις πάνω στην επικοινωνία αρχίζουν να εμφανίζονται. Το RFC1451 (Manager to Manager MIB) περιγράφει την δομή μιας MIB όπου αποθηκέυονται πληροφορίες σχετικές με την επικοινωνία δυο διαχειριστών. Η ύπαρξη προτύπων θα διευκολύνει την ανάπτυξη τέτοιων αρχιτεκτονικών, ιδιαίτερα σε ετερογενή περιβάλλοντα διαχείρισης.

### 8.2.3. Ιεραρχικό ΣΔΔ [HERM90] [RABI92] [LEIN93]

Η αρχιτεκτονική αυτή παρουσιάζεται στο σχήμα 8.4. Όπως και στην αρχιτεκτονική κατανεμημένου ΣΔΔ, ομότιμοι διαχειριστές αναλαμβάνουν κάποια manager domains και με την σειρά τους διαχειρίζονται από ένα διαχειριστή που βρίσκεται σε υψηλότερο επίπεδο ιεραρχίας (**Manager of Managers - MOM**). Ο MOM παίζει το ρόλο του κεντρικού συστήματος ανάλογο με αυτό του μοναδικού manager στην αρχιτεκτονική του κεντροποιημένου ΣΔΔ, και συγκεντρώνει μόνο τις σημαντικές πληροφορίες, αφήνοντας τις λεπτομέριες στους managers που διαχειρίζεται. Οι managers του χαμηλότερου επιπέδου, μπορούν να έχουν κονσόλα και άνθρωπο - χειριστή, ή μπορούν να παρακολουθούνται αυτόμata, κατευθείαν από την κονσόλα του MOM. Όπως και στην προηγούμενη αρχιτεκτονική, η επικοινωνία MOM και managers μπορεί να γίνεται μέσω του διαχειριζόμενου δικτύου ή μέσω ειδικού, ανεξάρτητου δικτύου διαχείρισης, ανάλογα με τις απαιτήσεις αξιοπιστίας.



**Σχήμα 8.4 - Ιεραρχικό ΣΔΔ**

Η αρχιτεκτονική αυτή μέσα από την αφαίρεση (abstraction) και την παράλληλη λειτουργία των managers, προσφέρει καλύτερο έλεγχο και απόδοση στην παρακαλούθηση του δικτύου. Επιπλέον, σε ένα ετερογενές δίκτυο (multivendor), κρύβει από το επίπεδο του MOM τα διαφορετικά πρωτόκολλα / διαχειριστές (proprietary protocols / managers) που βρίσκονται χαμηλότερα στην ιεραρχία, παρέχοντας έτσι ένα ολοκληρωμένο (integrated) διαχειριστικό περιβάλλον, ενοποιημένη αναπαράσταση του δικτύου και των τμημάτων του και ένα κοινό user interface. Το κέρδος είναι μειωμένο κόστος σε hardware και λιγότερες απαιτήσεις χώρου, ευκολότερη εκπαίδευση των χειριστών του ΣΔΔ και ευκολότερη και λιγότερο δαπανηρή ανάπτυξη και συντήρηση των εφαρμογών διαχείρησης (επίπεδο MOM). Ακόμη, σε ένα ετερογενές δίκτυο, η αρχιτεκτονική προσφέρει ευκολότερη ανάπτυξη ευφυών εφαρμογών που χρησιμοποιούν στοιχεία από πολλά επίπεδα και ετερογενή τμήματα του δικτύου [ERIC89]. Για παράδειγμα, μια ενοποιημένη εφαρμογή fault ή performance analysis, μέσω των διαχειριστικών πληροφοριών που παρέχει ο MOM, έχει στην διάθεσή της δεδομένα από τα συστήματα διαχείρισης των πολυπλεκτών, των modems, των γεφυρών κτλ,

πληρέστερη εικόνα του δικτύου και μεγαλύτερη ακρίβεια στα εξαγώμενα συμπεράσματα.

Αξίζει να σημειωθεί ότι υπάρχουν επίσημες τυποποιήσεις και πρότυπα που συσχετίζονται ή ακολουθούν αυτή την αρχιτεκτονική. Σχεδιάζεται και τυποποιείται τύπος MIB ειδικός για επικοινωνία μεταξύ διαχειριστών (manager to manager MIB -RFC1451-proposed standard). Η MIB αυτή γενικεύει κάποια αντικείμενα της RMON MIB (Remote LAN Monitoring MIB). Η RMON MIB (RFC1271 - βλέπε και κεφάλαιο 3) ορίζει μια δομή αποθήκευσης πληροφοριών που συγκεντρώνονται από αντιπροσώπους ανιχνευτές (probe-agents) που βρίσκονται σε κάθε τμήμα ενός LAN. Οι αντιπρόσωποι αυτοί ανιχνεύουν κάθε πακέτο και παράγουν μια περίληψη με πληροφορίες για το είδος και τις ιδιότητες των πακέτων και γεγονότα (events) όταν εντοπιστούν ιδιόμορφες καταστάσεις, συγκρούσεις κλπ. Μπορούν ακόμη να παρακολουθούν πακέτα σύμφωνα με συγκεκριμένα κριτήρια που ορίζονται από τον κεντρικό διαχειριστή. Έτσι, κάθε στιγμή ένας διαχειριστής μπορεί να αντλήσει πληροφορίες και λεπτομερής αναλύσεις από τους αντιπροσώπους αυτούς μέσω του SNMP πρωτοκόλλου διαχείρισης.

Τέλος, η αρχιτεκτονική αυτή βρίσκει εφαρμογές και σε δίκτυα όπου υπάρχει ανάγκη διαίρεσης του διαχειριζόμενου περιβάλλοντος, καθώς, σε αντίθεση με την αρχιτεκτονική του κατανεμημένου ΣΔΔ (παράγραφος 8.2.2), οι ομότιμοι (peer) διαχειριστές που διαχειρίζονται από το MOM, δεν επικοινωνούν μεταξύ τους. Τέτοια απομόνωση μπορεί να επιβάλεται για λόγους ασφαλείας, λογιστικών αναγκών, απαίτηση τοπικής διαχείρισης λαθών, ή τέλος, για λόγους συμμόρφωσης με δεδομένες τεχνολογικές, οργανωτικές ή γεωγραφικές δομές.

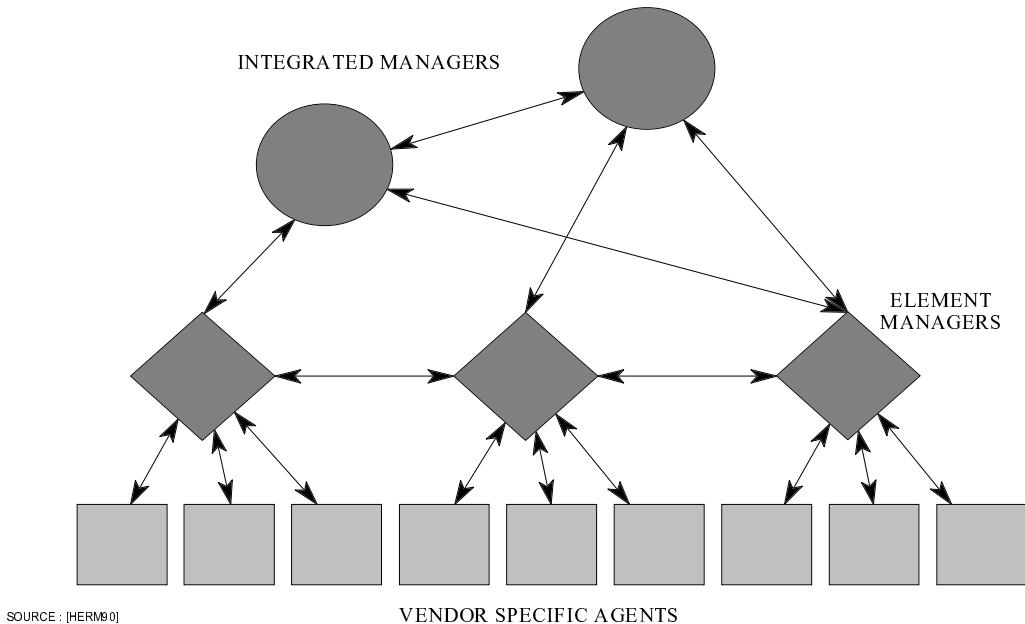
#### 8.2.4. Δικτυωμένο ΣΔΔ [HERM90]

Ένα παράδειγμα δικτυωμένου ΣΔΔ παρουσιάζεται στο σχήμα 8.5. Η αρχιτεκτονική αυτή συνδιάζει στοιχεία από τις αρχιτεκτονικές του κατανεμημένου και ιεραρχικού ΣΔΔ. Εδώ έχουμε περισσότερους από ένα MOM (Integrated Managers), ο καθένας από τους οποίους διαχειρίζεται μια ομάδα managers, κάθε ένας από τους οποίους διαχειρίζεται με την σειρά του μια ομάδα κόμβων (έννοια manager domain). Η αρχιτεκτονική αυτή ενθαρρύνεται από το γεγονός ότι όλο και περισσότερα διαχειριστικά συστήματα αναπτύσσουν τυποποιημένα interface, διευκολύνοντας έτσι την επικοινωνία με άλλα συστήματα. Το OSI Network Management Forum υποστηρίζει την προσπάθεια να επιτευχθεί ένα ευέλικτο και ισχυρό σύστημα που να ακολουθεί αυτή την αρχιτεκτονική.

Η αρχιτεκτονική του δικτυωμένου ΣΔΔ συνδυάζει τα πλεονεκτήματα των κατανεμημένων και ιεραρχικών συστημάτων που παρουσιάστηκαν παραπάνω. Το βασικότερο μειονέκτημα είναι ότι αυξάνει τον αριθμό των διαχειριστικών συστημάτων που χρησιμοποιεί, γεγονός που αυξάνει το κόστος. Βέβαια, ο ιεραρχικός χαρακτήρας αυτής της αρχιτεκτονικής μπορεί να οδηγήσει στην μείωση του αριθμού των χειριστών στο κέντρο διαχείρισης, αλλά δεν μειώνει τον αριθμό των διαχειριστικών συστημάτων που πρέπει να αγοραστούν και να συντηρηθούν. Όπως προαναφέρθηκε, μια τεχνική που οδηγεί σε μείωση του αριθμού των συστημάτων, είναι η μέθοδος της πλατφόρμας διαχείρισης (βλέπε παράγραφο 8.2(a) και σχήμα 8.2).

Τέλος, αξίζει να σημειωθεί ότι οι τρεις προηγούμενες αρχιτεκτονικές προσφέρουν λύσεις στα προβλήματα της ολοκληρωμένης διαχείρισης σε ετερογενή δίκτυα (βλέπε κεφάλαιο 8). Ειδικότερα η ιεραρχική και κατανεμημένη (peer-to-peer) αντιμετώπιση έχουν προταθεί για χρήση στην διαχείριση multi domain δικτύων [CHAO90]. Τέτοια δίκτυα είναι σαφώς χωρισμένα σε ξεχωριστά τμήματα (domains) που διαχειρίζονται από τοπικούς διαχειριστές (εξαρτώμενοι από τους κατασκευαστες των τμημάτων). Τα προβλήματα που αντιμετωπίζονται είναι η έλειψη ολοκληρωμένης πληροφορίας και η περιορισμένη δυνατότητα ελέγχου των τοπικών διαχειριστών. Έχουν προταθεί

αλγόριθμοι που αντιμετωπίζουν αυτά τα προβλήματα [DIMI89] και εφαρμόζονται στην κατανεμημένη αρχιτεκτονική, όπου ο κάθε διαχειριστής "βλέπει" τα άλλα τμήματα ως μοναδικούς κόμβους μέσω των τοπικών διαχειριστών.



**Σχήμα 8.5 - Δικτυωμένο ΣΔΔ**

### 8.3. Συμμόρφωση με τα πρότυπα - Απαιτήσεις ενός ΣΔΔ

Ο Διεθνής Οργανισμός Τυποποίησης (ISO) έχει καθορίσει πρότυπα για τις λειτουργικές περιοχές διαχείρισης δικτύων (Network Management Functional Areas) [OSIM89]. Αυτές περιγράφονται από ένα λειτουργικό μοντέλο γνωστό ως CFAPS (Configuration, Fault, Accounting, Performance, Security management) [LEIN93] [SMFA]. Οι πέντε αυτές περιοχές περιγράφονται στην παράγραφο 3.2.1 και αποτελούν ένα γενικό πλάνο-οδηγό για την σχεδίαση και ανάπτυξη ενός ΣΔΔ. Έχει όμως παρατηρηθεί ότι εμφανίζεται σημαντική επικάλυψη στις πέντε παραπάνω περιοχές [NMIS91]. Έτσι, ορίστηκαν νέα πρότυπα για τις λειτουργίες διαχείρισης συστημάτων και γίνεται προσπάθεια να τυποποιηθούν. Προϊόν αυτών των προσπαθειών είναι το system management application service element (SMASE [HALS92]) που αναφέρεται στην παράγραφο 3.2.4. Από τις λειτουργίες που προσφέρει και αρχικά αναφέρθηκαν στα draft international standards DIS 10164-1 έως 15 του ISO (καθώς και ένα πλήθος working drafts : WD 10164-tmf, WD 10164-smf κτλ)\*, μερικές μόνο έχουν τυποποιηθεί και αναφέρονται σε international standards [ALRM92] [OBJM91] [STAT91] [EVNT91] [LOGC91] [SECU92] [SCAD91].

\* Η διαδικασία έκδοσης προτύπων (ISO/IEC, CCITT) έχει ως εξής : αρχικά δημιουργείται ένα WD (working drafts) που αποτελεί μια πολύ γενική - υπό συζήτηση - μορφή του προτύπου. Από αυτό προκύπτει το CD (committee draft) που βρίσκεται σε ένα ανώτερο επίπεδο τυποποίησης, και αυτό με την σειρά μετατρέπεται σε DIS (draft international standard) που έχει κάποιο χαρακτήρα τυποποίησης, αλλά περισσότερο την μορφή πρότασης για τυποποίηση. Τέλος, προκύπτει το IS (international standard) που είναι το τελικό προϊόν αυτής της διαδικασίας.

Ακολουθεί μια επιλογή από τις λειτουργίες που αναφέρονται σε αυτά τα πρότυπα που ονομάζονται System Management Functions (SMF) :

- **Alarm reporting function** (λειτουργία αναφοράς συναγερμών) : Η λειτουργία αυτή επιτρέπει τον ορισμό, ενεργοποίηση και απενεργοποίηση συναγερμών για συγκεκριμένα διαχειριζόμενα στοιχεία του δικτύου. Το αντίστοιχο πρότυπο (SMF) ορίζει τις δομές δεδομένων που χρησιμοποιούνται (alarm record [ALRM92], κλπ), την σύνταξη και την σημασιολογία των συναγερμών.
- **Event Management reporting function** (λειτουργία διαχείρισης αναφοράς γεγονότων) : Η λειτουργία αυτή αναλαμβάνει τον έλεγχο, την συλογή, αναφορά και το φιλτράρισμα των γεγονότων (events) του δικτύου. Τα γεγονότα αυτά μπορεί να είναι αλλαγές στην τοπολογία, προβλήματα σε κάποιες συνδέσεις, SNMP (predefined) traps, events ορισμένα από τον χρήστη, κτλ. Με αυτή την λειτουργία μπορεί να οριστεί ένα φίλτρο που να ελέγχει ποια γεγονότα θα αναφέρονται και ποια όχι ή και να οριστούν νέες μορφές γεγονότων. Το αντίστοιχο πρότυπο ορίζει αυτούς τους μηχανισμούς και την μορφή και τον τρόπο αναφοράς.
- **Log Control function** (λειτουργία ελέγχου ημερολογίου) : Παρέχει ένα μηχανισμό διατήρησης και ελέγχου των ημερολογίων (logs) του διαχειριστικού συστήματος. Σε αυτά καταγράφονται αλλαγές στην τοπολογία, βλάβες και λάθη, γεγονότα (events), συναγερμοί, συναγερμοί ασφαλείας, κτλ. Το αντίστοιχο πρότυπο, ορίζει και περιγράφει τον μηχανισμό ελέγχου των ημερολογίων και τον τρόπο επιλογής των περιεχομένων των αναφορών που προκύπτουν από αυτά.
- **Security Alarm reporting function** (λειτουργία αναφοράς συναγερμών ασφαλείας) : Προσφέρει ένα μηχανισμό αναφοράς παραβάσεων ασφαλείας υπό την μορφή συναγερμών. Συναγερμοί μπορούν να οριστούν, να ενεργοποιηθούν, να απενεργοποιηθούν και να φιλτραριστούν. Το αντίστοιχο πρότυπο ορίζει την σύνταξη και την σημασιολογία αυτών των συναγερμών.
- **Security Audit Trail function** (λειτουργία παρακολούθησης ασφαλείας) : Προσφέρει ένα μηχανισμό παρακολούθησης παρακολούθησης της εισόδου των χρηστών στα διάφορα access points του δικτύου και καταγράφει την χρήση των devices και πόρων του δικτύου σε περιοδική βάση. Το αντίστοιχο πρότυπο περιγράφει αυτόν τον μηχανισμό.
- **Accounting Metering function** (λειτουργία μέτρησης λογιστικών μεγεθών) : Η λειτουργία αυτή παρακολουθεί και υπολογίζει στοιχεία χρησιμοποίησης των πόρων του δικτύου από κάθε χρήστη ή ομάδα χρηστών για λογιστικούς σκοπούς και καλύτερης κατανομής των πόρων. Το αντίστοιχο πρότυπο ορίζει τον μηχανισμό καταγραφής και αναφοράς τέτοιων πληροφοριών.
- **Workload Monitoring function** (λειτουργία παρακολούθησης φορτίου δικτύου) : Παρέχει μηχανισμούς παρακολούθησης της χρήσης των πόρων του δικτύου με σκοπό την αξιολόγηση των επιδόσεων. Το αντίστοιχο πρότυπο ορίζει τον μηχανισμό παρακολούθησης, ορισμού μεγεθών και αναφοράς μεγεθών απόδοσης και φορτίου.
- **Measurement Summary function** (λειτουργία περίληψης μετρήσεων) : Παρέχει δυνατότητα δημιουργίας στατιστικών περιλήψεων για τα χαρακτηριστικά λειτουργίας του δικτύου.
- **Object Management function** (λειτουργία διαχείρισης αντικειμένων) : Παρέχει μηχανισμούς δημιουργίας και καταστροφής αντικειμένων (αναπαραστάσεων στοιχείων του δικτύου) και μηχανισμούς χειρισμού των των ιδιοτήτων που συσχετίζονται με αυτά. Σχετική με αυτή την λειτουργία είναι η έννοια του Object Manager [GUTT93], ο οποίος ειναι υπεύθυνος για τον χειρισμό των αντικειμένων

του διαχειριζόμενου δικτύου. Αποτελείται από μια συλλογή υπηρεσιών που επιτρέπει τον ορισμό, καταστροφή και λειτουργία των αντικειμένων αντικειμένων. Ο Object Manager μπορεί να είναι μια βάση δεδομένων (αντικειμενοστραφής ή όχι) μαζί με κάποιο interface που να παρέχει στον διαχειριστή της λειτουργίες που προσφέρει η Object Management function.

- **State Management function** (λειτουργία διαχείρισης καταστάσεων) : Ορίζει τις πιθανές καταστάσεις του κάθε στοιχείου του δικτύου και μηχανισμούς παρακολούθησης και αλλαγής των καταστάσεων αυτών. Για παράδειγμα, ένα link μπορεί να βρίσκεται σε μια από τις ακόλουθες καταστάσεις : σε λειτουργία, εκτός λειτουργίας, προβληματική λειτουργία ή κάποιες άλλες σχετικές με τον τύπο του link. Μέσω state management function πρέπει να μπορεί να αναγνωριστεί η κατάσταση του link αυτού ανά πάσα στιγμή και να αλλαχθεί κατά βούληση.
- **Relationship Management function** (λειτουργία διαχείρισης σχέσεων - συσχετίσεων) : Παρέχει μηχανισμούς για την εγκαθίδρυση και διατήρηση συσχετίσεων μεταξύ διαχειριζόμενων στοιχείων - αντικειμένων του δικτύου.

Αυτές είναι οι βασικότερες λειτουργίες οι περισσότερες από τις οποίες είναι ήδη International Standards. Αξίζει να σημειωθεί ότι τα standards αυτά είναι κυρίως γενικές περιγραφές απαιτήσεων : περιγράφουν το "τι" και όχι το "πως". Το "πως" αφήνεται στον σχεδιαστή του ΣΔΔ, ο οποίος πρέπει να προσαρμόσει τα πρότυπα στις ανάγκες του συγκεκριμένου συστήματος.

Ακολουθεί μια αντιστοίχιση των λειτουργιών που περιγράφηκαν παραπάνω (System Management Functions) με το μοντέλο CFAPS. Είναι φανερό ότι κάποιες λειτουργίες υπάγονται σε παραπάνω από μια περιοχές διαχείρισης του CFAPS. Το γεγονός αυτό δείχνει την επικαλύψη λειτουργιών που παρουσιάζει αυτό το μοντέλο και δικαιολογεί την δημιουργία των SMF.

**1. Configuration Management (διαχείριση διάρθρωσης)** : Η διάρθρωση (configuration) των κόμβων και ιδιαίτερα κάποιων συγκεκριμένων στοιχείων (όπως γεφυρών, δρομολογητών κ.α.) ελέγχει την συμπεριφορά και την απόδοση του δικτύου. Η διαχειριστική περιοχή του configuration management είναι υπεύθυνη για την παρακολούθηση και την οργάνωση της διάρθρωσης ολόκληρου του δικτύου με έμφαση σε αυτούς του κρίσιμους κόμβους. Η αντίστοιχη λειτουργία πρέπει να είναι σε θέση να συλλέγει πληροφορίες από το δίκτυο για την διάρθρωσή του, να χρησιμοποιεί τα δεδομένα αυτά για να επεμβαίνει και να αλλάζει όταν η απόδοση του δικτύου το απαιτεί, να αποθηκεύει τις πληροφορίες αυτές (στην λεγόμενη Configuration Base - CB), να διατηρεί ενημερωμένο κατάλογο των πόρων του δικτύου (inventory) και ημερολόγιο αλλαγών (configuration log) και να παράγει αναφορές όταν γίνονται αλλαγές στην διάρθρωση ή όταν αυτό ζητηθεί από τον χρήστη. Βάση των παραπάνω χαρακτηριστικών λειτουργιών, οι SMF που υπάγονται στην περιοχή αυτή είναι : event reporting function (αναφορά αλλαγών στην διάρθρωση), log control function (ημερολόγιο αλλαγών διάρθρωσης), object management function (κατάλογος πόρων και CB) και state management function (παρακολούθηση καταστάσεων των στοιχείων).

**2. Fault Management (διαχείριση σφαλμάτων)** : Είναι η διαδικασία εντοπισμού προβλημάτων και λαθών στο δίκτυο. Αποτελείται από τέσσερα μέρη : ανακάλυψη του προβλήματος, εντοπισμός της περιοχής που εμφανίζεται το πρόβλημα, διάγνωση της αιτίας, αναφορά και προσπάθεια διόρθωσης. Στην περιοχή αυτή μπορούν να συμπεριληφθούν δυνατότητες πρόβλεψης προβλημάτων, πριν αυτά προκύψουν. Οι SMF που αντιστοιχούν είναι : alarm reporting function (εντοπισμός ανόμαλων καταστάσεων και προβλημάτων), event reporting function (εντοπισμός λαθών / διάγνωση από αλλαγές στην τοπολογία-διάρθρωση), log control function

(διατήρηση ημερολογίων λαθών), workload monitoring function (παρακολούθηση ρυθμών λαθών για εντοπισμό / προβλεψη / διαγνώση) και state management function (εντοπισμός λαθών / διάγνωση από αλλαγή κατάστασης). Επιπλέον, αν υπάρχει δυνατότητα πρόβλεψης έχουμε και τις παρακάτω SMF : object management function και measurement summary function. Αυτές οι δυο λειτουργίες είναι ίσως χρήσιμες και στην διαδικασία διάγνωσης.

- 3. Accounting Management (λογιστική διαχείριση)** : Αφορά παρακολούθηση των πόρων του δικτύου από τους χρήστες, ώστε να ελέγχεται αν παρέχεται στους χρήστες η ποσότητα των πόρων που χρειάζονται. Στην περιοχή αυτή περιλαμβάνεται και η διαδικασία αφαίρεσης ή προσφορά άδειας πρόσβασης σε πόρους σε συγκεκριμένους χρήστες. Άλλες υπηρεσίες που μπορεί να προσφέρει αυτή η περιοχή είναι η τήρηση κάποιων λογιστικών ορίων και η πληροφόρηση των χρηστών για το κόστος των υπηρεσιών που χρησιμοποιούν. Σύμφωνα με αυτά, οι SMF που υπάγονται σε αυτή την περιοχή είναι η account metering function και ίσως οι measurement summary και object management function, αν στοιχεία απόδοσης χρησιμοποιούνται για να τεθούν όρια και οι πληροφορίες χρησιμοποιήσης των πόρων αποθηκεύονται.
- 4. Performance Management (διαχείριση απόδοσης)** : Αναλαμβάνει την μέτρηση και παρακαλούθηση της απόδοσης του δικτύου συνολικά, αλλά με μονωμένων κόμβων, λογισμικού ή μέσου (δίσκου κάποιου file server κλπ). Η απόδοση μπορεί να μετριέται με συνολική ρυθμαπόδοση (overall throughput), χρησιμοποίηση % (utilization), ρυθμούς λαθών (εδώ υπάρχει μια επικάλυψη με το fault management), ή χρόνο απόκρισης και καθυστέρηση μεταξύ κάποιων σημείων του δικτύου. Τα αποτελέσματα μπορούν να χρησιμοποιηθούν για να εξασφαλιστεί ότι το δίκτυο θα παραμείνει σε καλή λειτουργία και απόδοση, σε ομοιόμορφη χρησιμοποίηση και χωρίς συμφόρηση. Για να επιτευχθούν όλα αυτά συλέγονται δεδομένα χρησιμοποίησης και ρυθμαπόδοσης από τους κόμβους και τα links τους, αναλύονται τα δεδομένα με σκοπό τον εντωπισμό περιοχών συνωστισμού στο δίκτυο, ορισμός συναγερμών με κατώφλια χρησιμοποίησης ή ρυθμαπόδοσης (δηλαδή δημιουργήται κάποια αναφορά σε περίπτωση που η χρησιμοποίηση ή η ρυθμαπόδοση σε κάποιο κόμβο ξεπεράσει κάποιο κατώφλι) και τέλος, μπορεί να χρησιμοποιηθεί προσομείωση για να ελεγχθούν σενάρια βελτίωσης της απόδοσης και χρησιμοποίησης του δικτύου. Σύμφωνα με αυτές τις απαιτήσεις, οι SMF που υπάγονται σε αυτή την περιοχή είναι : alarm reporting function (για τον οριμό κατωφλιών και συναγερμών), workload monitoring function (παρακολούθηση μεγεθών απόδοσης, φορτίου και λαθών) και measurement management function (εξαγωγή συγκεντρωτικών στατιστικών συμπερασμάτων).
- 5. Security Management (διαχείριση ασφαλείας)** : Είναι η διαδικασία ελέγχου της πρόσβασης της πληροφορίας στο δίκτυο. Αυτό μπορεί να επιτευχθεί περιορίζοντας την πρόσβαση των χρηστών σε ευαίσθητους κόμβους ή συσκευές του δικτύου, ειδοποιώντας καποιο υπεύθυνο σε περίπτωση παράνομης πρόσβασης ή άλλης παραβίασης και διατηρώντας ημερολόγιο παραβιάσεων. Η διαδικασία αποτελείται από τέσσερα μέρη : εντοπισμός της ευαίσθητης πληροφορίας, εντοπισμός των σημείων πρόσβασης σε αυτή, ασφάλιση των σημείων αυτών και διατήρηση της ασφάλειας. Σύμφωνα με αυτές τις απαιτήσεις οι SMF που αντιστοιχούν στην περιοχή αυτή είναι : log control function (ημερολόγια παραβιάσεων), security alarm reporting και security audit trail function.

## 8.4. Αντικειμενοστραφής φιλοσοφία και διαχείριση δικτύων [ANDE91] [MULL93]

Η αντικειμενοστραφής φιλοσοφία (object orientation) σηματοδοτεί μια νέα μεθοδολογία στην κατασκευή και την χρήση των δικτύων, εφαρμογών, βάσεων δεδομένων και λειτουργικών συστημάτων. Η χρήση αντικειμένων και η οργάνωση γύρω από τα δεδομένα που θα επεξεργαστούν και όχι γύρω από τις ενέργειες που πρέπει να παρθούν, δεν εφαρμόζεται μόνο στον προγραμματισμό. Η αντικειμενοστραφής τεχνολογία έχει βρεί χρήση και στα τηλεποινωνιακά δίκτυα για χρόνια και με μεγάλη επιτυχία. Επίσης, είναι βασικό χαρακτηριστικό του πρωτοκόλλων διαχείρισης SNMP και CMIP. Πρότυπα και τυποποιήσεις από ομιλίες/οργανισμούς όπως το OSF και OMG θα παίζουν βασικό ρόλο στην εμπορική επιτυχία ή αποτυχία της αντικειμενοστραφούς τεχνολογίας και της εφαρμογής της στα δίκτυα επικοινωνιών και στην διαχείρισή τους.

### 8.4.1. Νέες έννοιες και ορολογία [MCGR90] [MULL93]

Η αντικειμενοστραφής φιλοσοφία και ο αντικειμενοστραφής προγραμματισμός από την οποία πηγάζει, έχουν εισάγει νέες έννοιες και ορολογία. Ακολουθεί μια περιγραφή κάποιων βασικών νέων όρων που μεταφέρουν τις περισσότερες από τις ιδέες που κρύβονται σε αυτή την τεχνολογία. Οι έννοιες αυτές έχουν εφαρμοστεί για χρόνια με διάφορους τρόπους στις γραφικές διαπροσωπείες χρήστη (graphical user interface - GUI), στην ανάπτυξη εφαρμογών, ακόμη και στα δίκτυα και την διαχείρισή τους.

- **Αντικείμενα (objects)** : Είναι η βασική δομική μονάδα ενός συστήματος ή προγράμματος. Κάθε αντικείμενο έχει μια μοναδική ταυτότητα και συγκεκριμένες ιδιότητες (δεδομένα και/ή ενέργειες που μπορεί να εκτελέσει) και μπορεί να επικοινωνεί με άλλα αντικείμενα μέσω του μηχανισμού των μηνυμάτων (messaging).
- **Κατηγορήματα (attributes ή data members)** : τα κατηγορήματα ορίζουν τις "στατικές" ιδιότητες του αντικειμένου. Όλα τα δεδομένα που συσχετίζονται με ένα αντικείμενο αντιστοιχούν σε κατάλληλο κατηγόρημα.
- **Υπηρεσίες (services ή methods)** : οι υπηρεσίες ορίζουν τις "δυναμικές" ιδιότητες του αντικειμένου. Δηλώνουν τον τρόπο με τον οποίο το αντικείμενο ενεργεί και αντιδρά στα μηνύματα από άλλα αντικείμενα.
- **Μηνύματα (messages)** : Η επικοινωνία μεταξύ των αντικειμένων γίνεται στην αφηρημένη μορφή μέσω μηνυμάτων. Στον αντικειμενοστραφή προγραμματισμό ως μήνυμα συνήθως, ορίζεται η απλή κλήση κάποιας υπηρεσίας ενός αντικειμένου. Σε άλλες εφαρμογές της αντικειμενοστραφούς φιλοσοφίας, τα μηνύματα αυτά είναι λιγότερο αφηρημένα καθώς μπορεί να είναι πακέτα που ανταλλάσσονται μέσω δικτύου ή μέσω κάποιου άλλου μηχανισμού.
- **Κλάση (class)** : Αποτελεί τον ορισμό ενός αντικειμένου, δηλαδή ορίζει τις ιδιότητες και την συμπεριφορά του αντικειμένου και τον τρόπο πρόσβασης στις υπηρεσίες του. Η κλάση - ορισμός είναι διαφορετική οντότητα από το αντικείμενο και αποτελεί ένα πλαίσιο παραγωγής αντικειμένων μιας συγκεκριμένης μορφής. Κάθε αντικείμενο που δημιουργείται από μια κλάση ονομάζεται **στιγμιότυπο (instance)** αυτής της κλάσης. Τα στιγμιότυπα μιας κλάσης έχουν την ίδια μορφή, πιθανόν διαφορετικές τιμές στα κατηγορήματά τους, αλλά αποτελούν διαφορετικές οντότητες με μοναδική ταυτότητα το καθένα, ακόμη κι αν είναι ισομορφικά.
- **Κληρονομικότητα (inheritance)** : Νέες κλάσεις μπορούν να οριστούν από κάποιες ήδη ορισμένες κλάσεις. Η **παραγώμενη (derived)** κλάση κληρονομεί όλες τις

ιδιότητες της κλάσης από την οποία προέκυψε, αλλά μπορεί και να προσθέσει στον ορισμό νέα στοιχεία που συμπληρώνουν ή ακόμη και αναιρούν τις κληρονομημένες ιδιότητες. Ακόμη, είναι δυνατόν μια κλάση να προκύψει από περισσότερες της μιας κλάσεις και να κληρονομήσει ιδιότητες από όλες αυτές. Αυτή η διαδικασία ονομάζεται **πολλαπλή κληρονομικότητα** (**multiple inheritance**). Έτσι δημιουργείται μια ιεραρχία κληρονομικότητας με την μορφή γράφου, όπου οι κόμβοι συνδέονται με την σχέση υπερκλάσης - υποκλάσης (superclass - subclass) και ειδίκευσης - γενίκευσης. Έτσι μπορεί κανείς να ορίσει γενικές κλάσεις και ειδικεύοντας να δημιουργήσει κλάσεις για μια ειδικευμένη εφαρμογή, κάτι που προσφέρει μεγαλύτερη ευελιξία στον ορισμό κλάσεων και στον χειρισμό πολύπλοκων προβλημάτων.

- **Βιβλιοθήκη κλάσεων (class library)** : Μια συλλογή επαναχρησιμοποιήσημων μονάδων λογισμικού (κλάσεων και αντικειμένων) ονομάζεται βιβλιοθήκη κλάσεων. Τέτοιες βιβλιοθήκες διευκολύνουν την ανάπτυξη αντικειμενοστραφών εφαρμογών και μειώνουν τον απαιτούμενο χρόνο υλοποίησης μέσω της επαναχρησιμοποίησης. Υπάρχουν διαφόρων ειδών βιβλιοθήκες κλάσεων, ανάλογα με το επίπεδο ειδίκευσης. Για παράδειγμα υπάρχουν θεμελιακές βιβλιοθήκες που περιέχουν βασικές κλάσεις που υλοποιούν κοινές δομές ή συμπεριφορά και η εφαρμογή τους είναι πολύ γενική και μπορούν να χρησιμοποιηθούν σε όλα σχεδόν τα συστήματα. Οι βιβλιοθήκες - πλαίσια εφαρμογών (application frameworks), από την άλλη, είναι συσχετισμένες με συγκεκριμένες εφαρμογές και προσφέρουν ειδικευμένες κλάσεις.
- **Ενθυλάκωση (encapsulation)** : Αποτελεί μια επέκταση του παραδοσιακού δομημένου προγραμματισμού. Η κλασική αντιμετώπιση είναι ο χωρισμός δεδομένων και διαδικασιών. Με την ενθυλάκωση, τα δεδομένα (κατηγορήματα) συνδιάζονται με ένα σύνολο διαδικασιών (υπηρεσίες) και χρησιμοποιούνται για τον χειρισμό του αντικειμένου. Μόνο διαδικασίες που ορίζονται από τις υπηρεσίες του αντικειμένου, μπορούν να εκτελεστούν "πάνω" στο αντικείμενο.
- **Αφαίρεση (abstraction)** : Η ίδια η δομή του αντικειμένου και οι μηχανισμοί των μηνυμάτων, της ιεραρχίας και της ενθυλάκωσης κρύβουν τις εσωτερικές λεπτομέριες και προσφέρουν αφαίρεση σε πολλά επίπεδα.
- **Πολυμορφισμός (polymorphism)** : Είναι η δυνατότητα όμοιας συμπεριφοράς ή εφαρμογής του ίδιου τελεστή, σε διαφορετικά είδη αντικειμένων. Η ιδιότητα αυτή απλοποιεί την ανάπτυξη συστημάτων, καθώς προσφέρει ενοποιημένη αντιμετώπιση διαφορετικών αντικειμένων.
- **Δυναμική δέσμευση (dynamic / late binding)** : Είναι η διαδικασία κατά την οποία η κλήση μιας συνάρτησης αντιστοιχίζεται σε μια διεύθυνση σε χρόνο εκτέλεσης και όχι κατά την μεταγλώτηση. Η ιδιότητα αυτή συσχετίζεται άμεσα με την προηγούμενη.
- **Επεκτασιμότητα του κώδικα (code extensibility)** : Είναι ένα βασικό πλεονέκτημα της αντικειμενοστραφούς φιλοσοφίας, που επιτρέπει την επαναχρησιμοποίηση έτοιμων αντικειμένων και κλάσεων χωρίς να χρειάζεται πρόσβαση στην υλοποίηση. Η δυνατότητα επαναχρησιμοποίησης έτοιμων μονάδων διευκολύνει και επιταχύνει την ανάπτυξη, επέκταση και συντήρηση ενός συστήματος.

#### 8.4.2. Αντικείμενα και MIB

Όπως έχει αναφερθεί και σε προηγούμενα κεφάλαια η MIB είναι μια μονάδα αποθήκευσης πληροφοριών κατάλληλων για την διαχείριση των διαφόρων αντικειμένων - στοιχείων του δικτύου. Και τα δύο πρωτόκολλα διαχείρισης (SNMP, CMIP)

εγκαθιστούν agents (ή proxy agents) που διατηρούν και ενημερώνουν MIBs σε κάθε διαχειρίζομενο κόμβο. Οι δομές τους και ο τρόπος άντλησης πληροφοριών από την MIB παρουσιάζουν κάποιες διαφορές και ομοιότητες, αλλά κοινό χαρακτηριστικό είναι η αντικειμενοστραφής φιλοσοφία με την οποία αντιμετωπίζονται τα στοιχεία του δικτύου. Κάθε στοιχείο (hardware, software, ή μια αφηρημένη έννοια όπως μια σύνδεση ή ένα εικονικό κύκλωμα - virtual circuit) αντιμετωπίζεται σαν αντικείμενο με κάποια κατηγορήματα σχετικά με τον χαρακτήρα του. Οι περιγραφές των αντικειμένων φυλάσσονται στην MIB και ιεραρχίες μπορούν να οριστούν. Έτσι, κάθε MIB είναι μια συλλογή από αντικείμενα που συσχετίζονται με τον κόμβο στον οποίο βρίσκεται ο agent που την ενημερώνει. Στις επόμενες δύο παραγράφους παρουσιάζεται η αντικειμενοστραφής φιλοσοφία παρουσιάζει η MIB του SNMP και του CMIP.

#### 8.4.2.1. SNMP MIB (RFC 1155, 1156, 1212, 1213)

Το SNMP (RFC 1157) είναι πρωτόκολλο αίτησης/απάντησης (request/response) που υποστηρίζει τρεις μεθόδους άντλησης πληροφοριών. Η αίτηση get επιτρέπει την άντληση τιμών συγκεκριμένων αντικειμένων από την MIB, η αίτηση get-next προσφέρει μια μεθόδο πλοϊγήσης μέσα στην δομή της MIB, επιτρέποντας την ανάκτηση του επόμενου αντικειμένου. Τέλος, ο μηχανισμός ειδοποίησης των traps ειδοποιεί τον διαχειριστή σε περίπτωση συγκεκριμένων γεγονότων, μέσω ασύγχρονων μηνυμάτων.

Η πρώτη τυποποιημένη MIB του πρωτοκόλλου SNMP (MIB I - RFC 1156) περιέχει 110 αντικείμενα, η πλούσιότερη δεύτερη τυποποίηση (MIB II - RFC 1213) υποστηρίζει 165 αντικείμενα, ενώ πολλοί κατασκευαστές κατασκευάζουν επεκτάσεις με παραπάνω των 200 αντικείμενα. Τέλος, μια άλλη τυποποιημένη MIB είναι η RMON MIB που αναφέρθηκε στην παράγραφο 8.2.3, η οποία προς το παρόν υποστηρίζει αντικείμενα που αφορούν στοιχεία μέσου για ethernet δικτυα. Επιπλέον των ήδη τυποποιημένων MIB, νέα πρότυπα δημιουργούνται όπως MIB ειδικά για διαχείριση γεφυρών (μέχρι τώρα στις γέφυρες - routers τοποθετείται υποσύνολο της MIB II), για διαχείριση των modems, για διαχείριση των συνδέσεων σε δίκτυα ATM, για διαχείριση δικτύων FDDI έως και MIB για εκτυπωτή που είναι συνδεδεμένος στο δίκτυο [ST2393]. Έτσι, εκτός από την αντικειμενοστραφή φιλοσοφία στην αντιμετώπιση των πληροφοριών που φυλάσσονται σε μια MIB, αρχίζει να παρουσιάζεται και ανάλογη φιλοσοφία και στην διαχείριση στοιχείων του δικτύου σε μεγαλύτερη κλίμακα. Μια διαχειριστική πλατφόρμα μπορεί να βλέπει μια γέφυρα, ένα εκτυπωτή, μια σύνδεση σε ένα ATM δίκτυο, ή ένα τμήμα (segment) ενός δικτύου με ένα ενοποιημένο τρόπο : ως αντικείμενα αποτελούμενα από πιο ειδικά αντικείμενα (ή ίσως κατηγορήματα) - τα αντικείμενα που φυλάσσονται στην MIB. Κάτι τέτοιο προσφέρει καλύτερη δυνατότητα αναπαράστασης και αποθήκευσης των αφηρημένων αντικειμένων του δικτύου στο διαχειριστικό σύστημα.

Κλείνοντας, μπορούμε να παρατηρήσουμε τα εξής : κάθε MIB αποτελεί μια αναπαράσταση του διαχειριζόμενου στοιχείου στο οποίο βρίσκεται και μπορεί να θεωρηθεί ως αντικείμενο ή μια συλλογή από αντικείμενα (τα περιεχόμενα της MIB). Η λειτουργικότητα (functionality) των αντικειμένων, και στις δύο περιπτώσεις, είναι έμεση καθώς αποτελείται από παρενέργειες των SNMP εντολών SET. Αυτή η αντικειμενοστραφής φιλοσοφία βρίσκεται στο επίπεδο του διαχειριστικού συστήματος και όχι στο επίπεδο της διαχειριστικού πρωτοκόλλου. Άλλωστε, το SNMP (SIMPLE Network Management Protocol) είναι ένα απλό, μινιμαλιστικό πρωτόκολλο που αφήνει την πολυπλοκότητα στον διαχειριστή. Έτσι η αντικειμενοστραφής φιλοσοφία βρίσκει εφαρμογή περισσότερο στην αναπαράσταση των στοιχείων της MIB και του δικτύου στον διαχειριστή και στον τρόπο με τον οποίο αντιμετωπίζονται από αυτόν.

#### 8.4.2.2. CMIP MIB [SMIB92]

Η MIB του πρωτοκόλλου CMIP είναι ανάλογη σε γενικές γραμμές με αυτή του SNMP και προσφέρει ανάλογα αντικείμενα. Παρουσιάζει, όμως, πιο "καθαρό" αντικειμενοστραφή χαρακτήρα. Τα αντικείμενα οργανώνονται σε ιεραρχία ανάλογη με αυτή των κλάσεων και διαχειρίζονται από το CMIP με περισσότερο αντικειμενοστραφή χαρακτήρα από ότι στην περίπτωση του SNMP. Υπάρχουν και εδώ εντολές που διαβάζουν ή γράφουν τα στοιχεία που φυλάσσονται στην MIB και ασύγχρονες ειδοποιήσεις (events). Η λειτουργικότητα όμως των αντικειμένων (αντικείμενα με την ίδια έννοια που αναφέρθηκαν στην προηγούμενη παράγραφο) είναι περισσότερο άμεση, καθώς το CMIP υποστηρίζει και εντολές-ενέργειες (action commands).

Ένα ακόμη στοιχείο της CMIP MIB που προσθέτει στον αντικειμενοστραφή χαρακτήρα της, είναι κάποια ικανότητα ορισμού νέων αντικειμένων (στοιχείων) ή σβήσιμο ήδη ορισμένων αντικειμένων μέσα στην MIB. Για παράδειγμα μπορούν να οριστούν νέοι τύποι log record για συγκεκριμένα γεγονότα (events), που μπορούν να παραχθούν (και να κληρονομήσουν στοιχεία) από αντίστοιχο αντικείμενο που ήδη υπάρχει. Ένα άλλο παράδειγμα είναι ο ορισμός νέων συναγερμών. Οι συναγερμοί είναι αντικείμενα με κατηγορήματα ανάλογο με τον τύπο τους. Απλά αντικείμενα και κατηγορήματα είναι ήδη ορισμένα στην MIB και μπορούν να χρησιμοποιηθούν για να οριστούν νέοι τύποι.

#### 8.4.3. Εφαρμογή σε ΣΔΔ

Η αντικειμενοστραφής φιλοσοφία μπορεί να εφαρμοστεί στην αναπαράσταση των στοιχείων του δικτύου σε εσωτερικές δομές, αλλά και στην παρουσίασή τους στην διαπροσωπεία χρήστη και τον τρόπο με τον οποίο ο χρήστης μπορεί να τα χειρίστει.

Όπως προαναφέρθηκε, στο επίπεδο του διαχειριστή, κλάσεις διαχειριζόμενων στοιχείων του δικτύου μπορούν να οριστούν, οργανωμένα σε ιεραρχίες. Τα κατηγορήματά τους μπορούν να περιλαμβάνουν στοιχεία που βρίσκονται στην MIB, αλλά και πρόσθετα που έχουν σχέση με άλλες δομές που χρησιμοποιεί ο διαχειριστής. Επιπλέον, υπηρεσίες μπορούν να οριστούν που να συσχετίζονται με τις ενέργειες που μπορούν να επιτευχθούν μέσω της MIB και άλλες που έχουν σχέση με την διαπροσωπεία χρήστη και εσωτερικές λειτουργίες του διαχειριστή. Για παράδειγμα, μια κλάση που παριστάνει την σύνδεση μεταξύ δρομολογητή και τμήματος του LAN, μπορεί να περιλαμβάνει ως κατηγορήματα τα στοιχεία που φυλάσσονται στην MIB για ένα interface αλλά και πρόσθετα στοιχεία όπως ποιό τμήμα του δικτύου συνδέει με ποιό δρομολογητή, ποιό εικονίδειο θα χρησιμοποιείται για την απεικόνιση της σύνδεσης κτλ. Ακόμη, οι υπηρεσίες μια τέτοιας κλάσης θα περιέχουν την δυνατότητα αλλαγής κατάστασης της σύνδεσης (UP/DOWN), που αντιστοιχεί στην αντίστοιχη λειτουργία που προσφέρει η MIB, αλλά και υπηρεσίες που ορίζουν την συμπεριφορά της απεικόνισης της σύνδεσης στην οθόνη, κτλ.

Η αντικειμενοστραφής φιλοσοφία μπορεί να εφαρμοστεί στην διαπροσωπεία χρήστη και να απλουστευση την παρουσίαση των διαχειριζόμενων στοιχείων και την χρήση του ΣΔΔ. Μέσω της ιδότητας του πολυμορφισμού, ορίζονταις κατάλληλες υπηρεσίες, όλα τα διαχειριζόμενα στοιχεία (ουσιαστικά οι αναπαραστάσεις τους στον διαχειριστή) μπορούν να αντιμετωπιστούν με ενοποιημένο και φιλικό τρόπο από τον χρήστη. Για παράδειγμα, σε ένα γραφικό περιβάλλον παρουσίασης του δικτύου, όλα τα στοιχεία μπορούν να παρουσιάζονται με εικονίδια (icons). "Πιάνοντας" με το ποντίκι ένα τέτοιο εικονίδιο και ρίχνοντας το πάνω στο εικονίδιο ενός εκτυπωτή, πληροφορίες ή και στοιχεία απόδοσης του στοιχείου (ανάλογα με το είδος του στοιχείου) μπορούν να εκτυπώνονται στον ανάλογο εκτυπωτή. Επιπλέον, μια τέτοια αντιμετώπιση κάνει περισσότερο φιλικό και πιο εύκολο στην εκμάθηση της κονσόλας ενός ΣΔΔ.

Η ενοποιημένη αναπαράσταση όλων των στοιχείων του δικτύου με αντικείμενα εισαγάγει την έννοια του **κεντρικού διαχειριστή αντικειμένων - ΚΛΑ (global**

**object manager).** Ο ΚΔΑ είναι υπέυθυνος για την ενοποιημένη διαχείριση και αποθήκευση όλων των διαχειριζόμενων αντικειμένων του δικτύου. Μπορεί να είναι το front-end ενός συστήματος διαχείρισης ΒΔ (αντικειμενοστραφούς ΒΔ ή όχι), ή μια συλλογή αντικειμενοστραφών υπηρεσιών (μορφή πλατφόρμας) που επιτρέπουν την ύπαρξη και διαχείριση αντικειμένων σε ένα αυθαίρετα ετερογενές περιβάλλον.

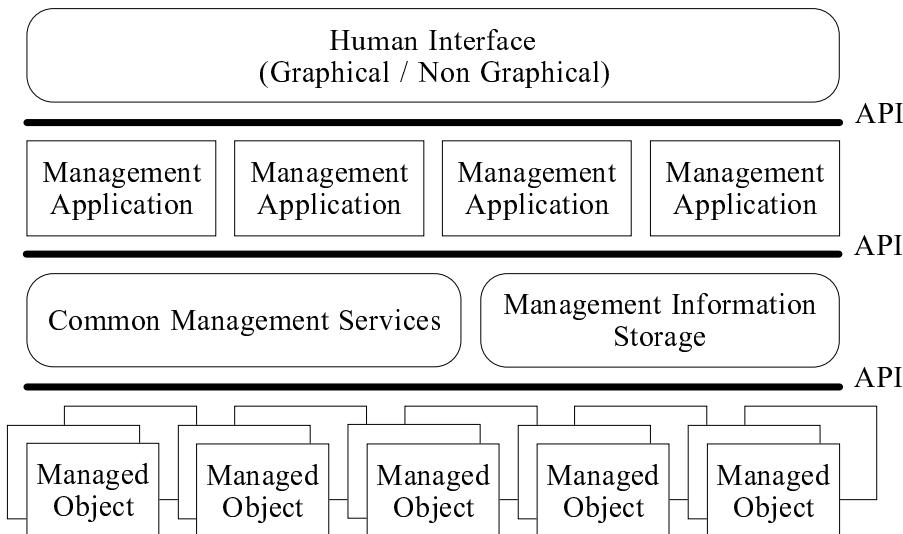
Τέλος, η αντικειμενοστραφής φιλοσοφία μπορεί να χρησιμοποιηθεί στην ανάπτυξη ενός ΣΔΔ (βλέπε παράγραφο 8.5). Οι αντικειμενοστραφείς τεχνικές ανάλυσης, σχεδίασης και προγραμματισμού προσφέρουν μείωση στον χρόνο ανάπτυξης, απλοποιούν την διαδικασία συντήρησης και επέκτασης και αυξάνουν την αξιοπιστία. Τα αντικείμενα προσφέρουν μεγαλύτερη λειτουργικότητα, συνδέοντας τις παραδοσιακά ξεχωριστές περιοχές κώδικα και δεδομένων. Λεπτομέριες για την εφαρμογή αντικειμενοστραφών τεχνικών στην ανάπτυξη ενός ΣΔΔ δίνονται στην ενότητα 8.5.

#### 8.4.4. Πρότυπα - Τυποποιήσεις

Η ύπαρξη προτύπων και τυποποιήσεων θα προσφέρει ένα κοινό τρόπο αναπαράστασης και ελέγχου αντικειμένων σε ένα ΣΔΔ. Το Object Management Group (OMG) είναι μια βιομηχανική ομάδα που έχει συσταθεί από περίπου 200 κατασκευαστές, που στόχο έχει την ανάπτυξη τέτοιων προτύπων. Προωθεί ένα περιβάλλον λογισμικού που ονομάζει Object Management Architecture (OMA) που συνδιάζει τις αρχές της αντικειμενοστραφής φιλοσοφίας και της κατανεμημένης επεξεργασίας. Το OMA χωρίζεται σε τέσσερις λειτουργικές μονάδες που ορίζουν τα interfaces και τις υπηρεσίες που επιτρέπουν ένα αντικείμενο να μεταδίδει και να δέχεται μηνύματα από άλλα αντικείμενα σε ένα ετερογενές δίκτυο. Παρόλο που το OMA είναι ακόμη υπό ανάπτυξη το OMG δημοσίευσε στο τέλος του 1991 τις προδιαγραφές της μονάδας Object Request Broker (ORB). Η ORB καθορίζει τον μηχανισμό επικοινωνίας με τον οποίο μηνύματα ανταλλάσσονται μεταξύ των αντικειμένων. Η OMG θεωρεί ότι η μονάδα αυτή αποτέλει ένα βασικό στοιχείο πάνω στο οποίο μπορεί να χτιστούν αντικειμενοστραφή συστήματα.

Υπάρχουν όμως και άλλα πρότυπα και τυποποιήσεις έξω από την εμβέλεια του OMA, που πρέπει να οριστούν πριν ένα ολοκληρωμένο προτυπο οριστεί που να επιτρέπει την ανάπτυξη ενός συστήματος διαχείρισης αντικειμένων. Για παράδειγμα, τυποποιήση χρειάζεται σε θέματα διαχείρισης αντικειμένων σε βάσεις δεδομένων. Ο οργανισμός τυποποίησης ANSI και το SQLAccess Group δουλεύουν πάνω στην ανάπτυξη ενός τέτοιου προτύπου.

Τέλος, το Open Software Foundation (OSF), ένας αφιλοκερδής, χρηματοδοτούμενος από την βιομηχανεία οργανισμός, επιχειρεί να αναπτύξει ενός τέτοιου πρότυπου συστήματος. Αποτέλεσμα είναι το σύστημα Distributed Management Environment (DME) του OSF που είναι μια αρχιτεκτονική προορισμένη για διαχείριση στοιχείων δικτύου ή υπηρεσιών σύμφωνα με την φιλοσοφία του CMIP και του SNMP. Το DME είναι η προσπάθεια του OSF να αναπτύξει ένα ΣΔΔ ικανό να διαχειριστεί ένα ευρύ φάσμα τύπων δικτύων : απλά και μικρά δίκτυα μέχρι μεγάλα, κατανεμημένα και ετερογενή δίκτυα. Ένα περίγραμμα της αρχιτεκτονικής του DME παρουσιάζεται στο σχήμα 8.6. Όπως φαίνεται και στο σχήμα, επιτρέπει την ανάπτυξη κατανεμημένων διαχειριστικών εφαρμογών πάνω από μια ενοποιημένη πλατφόρμα διαχείρισης. Ακόμη, το DME αντιμετωπίζει τα δύσκολα προβλήματα που παρουσιάζονται στην διαχείριση ετερογενών δικτύων παρέχοντας ένα σύνολο application programming interfaces (APIs) που επιτρέπουν την επικοινωνία διαφορετικών μονάδων λογισμικού από ετερογενή συστήματα.



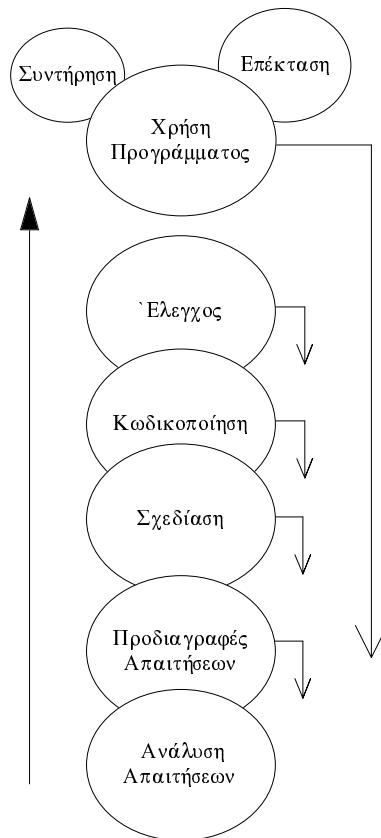
### Σχήμα 8.6 - OSF DME

## 8.5. Ανάπτυξη του λογισμικού ενός ΣΔΔ

Η σχεδίαση και ανάπτυξη ενός ΣΔΔ που να περιλαμβάνει όλες τις λειτουργίες που χρειάζονται για να επιτευχθεί πλήρης διαχείριση είναι μια εξαιρετικά δύσκολη εργασία. Πρέπει να κατανοηθούν οι απαιτήσεις ενός τέτοιου συστήματος, να υπάρξει συμμόρφωση με τα πρότυπα και τους γενικούς περιορισμούς που αναφέρθηκαν παραπάνω και τέλος να σχεδιαστεί η αρχιτεκτονική του ΣΔΔ και μια σειρά από εργαλεία που πραγματοποιούν τις λειτουργίες διαχείρισης. Στην παράγραφο αυτή δίνεται μια περιληπτική περιγραφή της διαδικασίας ανάπτυξης του λογισμικού ενός ΣΔΔ. Περισσότερη λεπτομέρεια θα ξέφευγε από τον σκοπό αυτού του βιβλίου, καθώς αναφέρεται σε προχωρημένα θέματα τεχνολογίας λογισμικού (software engineering).

Το πρώτο βήμα στην ανάπτυξη του λογισμικού του ΣΔΔ είναι η επιλογή του μοντέλου κύκλου ζωής (software life cycle [IEEE83] [AGRE86] [ΣΚΟΡ91]) του λογισμικού που θα ακολουθηθεί. Το μοντέλο κύκλου ζωής παρέχει οδηγίες για το ποιες εργασίες πρέπει να γίνουν για την ανάπτυξη του λογισμικού, με ποια σειρά και ποια είναι τα κριτήρια μετάβασης από τη μια εργασία (στάδιο) στην άλλη. Με λίγα λόγια, το μοντέλο κύκλου ζωής ορίζει μια μεθοδολογία ανάπτυξης λογισμικού από το στάδιο προσδιορισμού των απαιτήσεων μέχρι το τελικό προϊόν και την συντήρησή του.

Στην ανάπτυξη ενός ΣΔΔ, προφανώς μπορούν να χρησιμοποιηθούν οποιαδήποτε από τα συμβατικά μοντέλα κύκλου ζωής, όπως το πολύ κλασσικό "μοντέλο του καταρράκτη" (waterfall life cycle model [BOEH84] [RAMA84]). Όμως ο αντικειμενοστραφής χαρακτήρας που παραρουσιάζει η εφαρμογή (σύμφωνα με όσα αναφέρθηκαν στην προηγούμενη ενότητα) μας ωθεί στην υιοθέτηση του αντικειμενοστραφούς μοντέλου (object-oriented model) [BCOX87] [HEND90] [ΣΚΟΡ91]. Το μοντέλο αυτό παρουσιάζεται σε μορφή διαγράμματος στο σχήμα 8.7.



**Σχήμα 8.7 - Αντικειμενοστραφές Μοντέλο**

Κάθε φυσαλίδα αντιπροσωπεύει ένα στάδιο στην ζωή του λογισμικού, ενώ η μετάβαση γίνεται από κάτω προς τα πάνω. Σε σύγκριση με το κλασσικό μοντέλο του καταρράκτη, τα στάδια είναι περίπου τα ίδια ενώ οι διαφορές εντοπίζονται κυρίως σε δύο σημεία : (α) παρουσιάζει μεγάλη επικάλυψη μεταξύ των διαφόρων σταδίων, και (β) η ανάπτυξη του λογισμικού ακολουθεί μορφή πίδακα και οι οπισθοδρομήσεις γίνονται μόνο στο προηγούμενο στάδιο, εκτός από το τελευταίο στάδιο που μπορεί να οπισθοδρομήσει στην αρχή. Κάτι που δεν φαίνεται στον σχήμα 8.7 είναι το γεγονός ότι το μοντέλο επιτρέπει παράλληλη ανάπτυξη, ταυτόχρονα σε ανεξάρτητες ομάδες και κλάσεις. Επιπλέον, η δυνατότητα χρήσης επαναχρησιμοποιούμενων μονάδων (κλάσεις) μειώνει το κόστος, την δυσκολία και τον απαιτούμενο χρόνο στις διαδικασίες επέκτασης και συντήρησης. Το μικρό μέγεθος των φυσαλίδων του σχήματος 8.7 που αντιπροσωπεύουν αυτά τα δύο στάδια, υπογραμμίζουν το γεγονός αυτό που αποτελεί ένα από τα βασικότερα πλεονεκτήματα του μοντέλου αυτού. Καθώς βασικές απαιτήσεις (βλέπε παράγραφο 8.1) ενός ΣΔΔ είναι η δυνατότητα επέκτασης και προσαρμογής σε νέες ανάγκες διαχείρισης (συντήρησης λογισμικού), τα πλεονεκτήματα αυτά του αντικειμενοστραφούς μοντέλου ενισχύουν την επιλογή του για την ανάπτυξη ενός ΣΔΔ.

Η ανάλυση απαιτήσεων είναι το πρώτο στάδιο στο αντικειμενοστραφές μοντέλο κύκλου ζωής. Σε αυτό εντοπίζονται οι γενικές απαιτήσεις του συστήματος που θα αναπτυχθεί. Στην περίπτωση ανάπτυξης ενός ΣΔΔ, οδηγό σε αυτό το στάδιο μπορεί να αποτελέσει η παράγραφος 8.1, ενώ άλλες απαιτήσεις και περιορισμοί μπορούν να προστεθούν ανάλογα με τον σκοπό και το είδος του ΣΔΔ.

Στην επόμενη φάση, το στάδιο προδιαγραφών απαιτήσεων, το θέμα των απαιτήσεων αντιμετωπίζεται ειδικότερα και με μεγαλύτερη λεπτομέρεια. Τα ζητούμενα είναι δύο : οι απαιτήσεις από το σύστημα και οι απαιτήσεις από το λογισμικό, δηλαδή τι απαιτούμε να κάνει το σύστημα συνολικά και τι το λογισμικό ειδικότερα [ΣΚΟΡ91]. Η καταγραφή αυτών των απαιτήσεων ονομάζεται προδιαγραφή απαιτήσεων και συνήθως

παρουσιάζει σημαντική δυσκολία. Όπως φαίνεται από το σχήμα 8.7, η οπισθοδρόμηση στην προηγούμενη φάση είναι δυνατή και σκοπός της είναι ο επαναπροδιορισμός του προβλήματος. Η διαδικασία διατύπωσης του προβλήματος, συλλογής πληροφοριών για τις απαιτήσεις, επεξεργασίας και καταγραφής τους, ονομάζεται **ανάλυση** και μπορεί να γίνει με διάφορες μεθόδους. Προϊόντα της ανάλυσης είναι τα έγγραφα καταγραφής των απαιτήσεων : έγγραφο προδιαγραφών απαιτήσεων από το σύστημα (ΕΠΑΣ - περίγραμμα και περιγραφή δίνονται στα [FAIR85] [ΣΚΟΡ91]) και έγγραφο απαιτήσεων από το λογισμικό (ΕΠΑΛ - περίγραμμα και περιγραφή δίνονται στα [IEEE84] [ΣΚΟΡ91]).

Η πιο διαδεδομένη τεχνική ανάλυσης είναι **δομημένη ανάλυση** (structured analysis) [DEMA78] [GANE79] [GEOF87]. Η τεχνική επικεντρώνεται στην ροή των δεδομένων μεταξύ των μονάδων του λογισμικού, από την είσοδο προς την έξοδο του συστήματος. Η ροή αυτή καταγράφεται με γραφικό τρόπο στα **διαγράμματα ροής δεδομένων** (data flow diagrams) [YOUR89] [ΣΚΟΡ91]. Σε αυτά παρουσιάζονται, εκτός από τα δεδομένα και η ροή τους, οι βασικές λειτουργικές μονάδες λογισμικού που ονομάζονται και μετασχηματισμοί, καθώς "μετασχηματίζουν" τα δεδομένα εισόδου σε δεδομένα εξόδου. Παρόλο που η μέθοδος της δομημένης ανάλυσης είναι συνυφασμένη με τον διαδικασιακό παράδειγμα προγραμματισμό (procedural programming paradigm) και αποτελεί το πρώτο μέρος μιας τεχνικής που συμπληρώνεται με την δομημένη σχεδίαση (structured design) [GEOF87] [ROSS87], μπορεί να χρησιμοποιηθεί στο αντικειμενοστραφές μοντέλο (συνυφασμένο με το αντικειμενοστραφές παράδειγμα προγραμματισμού), ακόμη και αν ακολουθηθεί από αντικειμενοστραφής σχεδίαση (βλέπε παρακάτω). Βέβαια σε αυτή την περίπτωση, πρέπει πριν την φάση της σχεδίασης να απεικονιστούν οι λειτουργίες - οντότητες της δομημένης ανάλυσης σε κατάλληλες κλάσεις.

Μια άλλη μέθοδος ανάλυσης, περισσότερο συσχετισμένη εννοιολογικά με το αντικειμενοστραφές μοντέλο ζωής είναι η **αντικειμενοστραφής ανάλυση** (object-oriented analysis). Μαζί με την αντικειμενοστραφή σχεδίαση είναι μια σχετικά νέα μέθοδος και παρόλο που έχουν παρουσιαστεί διάφορες τεχνικές [MEYE88] [BAIL89] [COAA91] δεν έχει επικρατήσει καμμία εως τώρα. Τυποποίηση δεν έχει επιτευχθεί ακόμη, ενώ γίνονται προσπάθειες προς αυτή την κατεύθυνση. Επιπλέον των αμιγών αντικειμενοστραφών τεχνικών γίνεται προσπάθεια για έκδοση μιας νέας έκδοσης δομημένης ανάλυσης που να είναι κατάλληλη για αντικειμενοστραφή ανάλυση.

Πέρα των διαφορών που παρουσιάζουν οι διάφορες αντικειμενοστραφής τεχνικές ανάλυσης, όλες χρησιμοποιούν ως οντότητες κλάσεις και αντικείμενα, σε αντίθεση με τα δεδομένα και τους μετασχηματισμούς της δομημένης ανάλυσης. Πρέπει να οριστούν οι κλάσεις, οι δομές ιεραρχίας και η επικοινωνία μεταξύ των αντικειμένων με μηνύματα. Για κάθε κλάση ορίζονται τα κατηγορίματα και οι υπηρεσίες. Ανάλογα με την τεχνική, χρησιμοποιούνται διάφορες γραφικές μέθοδοι παρουσίασης όλων αυτών των στοιχείων και άλλες μέθοδοι καταγραφής των απαιτήσεων. Ολόκληρη, πάντως, η λογική χτίζεται γύρω από την έννοια του αντικειμένου (δεδομένα & υπηρεσίες) και όχι γύρω από τις λειτουργίες και τη ροή δεδομένων.

Στην περίπτωση ανάπτυξης του λογισμικού ενός ΣΔΔ, οδηγός στην φάση ανάλυσης μπορούν να είναι όσα αναφέρθηκαν στην παράγραφο 8.3. Βάση των λειτουργικών απαιτήσεων που αναφέρονται εκεί (εμπλουτισμένα με μεγαλύτερη λεπτομέρεια και πρόσθετα στοιχεία ανάλογα με τον χαρακτήρα του ΣΔΔ) μπορούν να οριστούν οι ανάλογες λειτουργίες (αν χρησιμοποιηθεί δομημένη ανάλυση) ή τα κατάλληλα αντικείμενα και κλάσεις (αν εφαρμοστεί αντικειμενοστραφής ανάλυση).

Η επιλογή αρχιτεκτονικής (βλέπε παράγραφο 8.2) βρίσκεται μεταξύ ανάλυσης και σχεδίασης. Αν γίνει στην φάση της ανάλυσης, διευκολύνει την διαδικασία ορισμού λειτουργιών ή κλάσεων. Η επιλογή πάντως πρέπει να γίνει μετά την αρχική φάση του μοντέλου (ανάλυση απαιτήσεων), καθώς η αρχιτεκτονική εξαρτάται από τους

περιορισμούς και απαιτήσεις του ΣΔΔ. Στην ιδανική περίπτωση, η αρχιτεκτονική του διαχειριστικού συστήματος πρέπει να αντικατοπτρίζει την δομή του οργανισμού που το χρησιμοποιεί και την λειτουργικότητα του δικτύου που θα διαχειρίζεται [LEIN93]. Για παράδειγμα εάν η οργάνωση της επιχείρησης που θα χρησιμοποιεί το ΣΔΔ είναι καθαρά κεντροποιημένη, τότε η επιλογή μιας κεντροποιημένης αρχιτεκτονικής είναι συνήθως η λογική απόφαση. Αντίθετα, αν η οργάνωση είναι ευρέως κατανεμημένη με πολλά ομότιμα επίπεδα διαχείρισης, μια αρχιτεκτονική με κατανεμημένη λογική είναι προτιμότερη. Ανάλογες αποφάσεις παίρνονται όταν υπάρχουν λογιστικές ή γεωγραφικές δομές στο διαχειριζόμενο δίκτυο και επιβάλουν κάποιους περιορισμούς στην διαχείριση.

Το επόμενο στάδιο είναι αυτό της **σχεδίασης**. Στο στάδιο αυτό χρησιμοποιούνται τα αποτελέσματα της ανάλυσης για να σχηματιστεί ένα ομοίωμα του τελικού συστήματος [ΣΚΟΡ91]. Δεδομένο, λοιπόν, σε αυτό το στάδιο είναι οι προδιαγραφές απαιτήσεων και ζητούμενο είναι ένα **σχέδιο λογισμικού**, δηλαδή ένα ομοίωμα του πραγματικού λογισμικού (του τελικού κώδικα, στον οποίο η φάση της σχεδίασης βρίσκεται κοντύτερα από την φάση της ανάλυσης). Στοιχεία του σχεδίου λογισμικού αποτελούν δομικές μονάδες λογισμικού σε μια αφηρημένη μορφή οι οποίες μπορούν, σε επόμενη φάση, να υλοποιηθούν και να δώσουν μονάδες κώδικα που ακολουθούν τις προδιαγραφές απαιτήσεων. Το σχέδιο χωρίζεται σε αρχιτεκτονικό ή προκαταρτικό και λεπτομερές, που περιγράφουν αντίστοιχα την μακροσκοπική και μικροσκοπική δομή του λογισμικού.

Η φάση της σχεδίασης είναι απόλυτα συσχετισμένη με το παράδειγμα προγραμματισμού που θα ακολουθηθεί στην επόμενη φάση της κωδικοποίησης. Αυτό σημαίνει, ότι στο αντικειμενοστραφές μοντέλο ζωής, στο οποίο χρησιμοποιείται το αντι-κειμενοστραφές μοντέλο προγραμματισμού, εάν δεν χρησιμοποιηθεί αντικειμενοστραφής σχεδίαση, δημιουργούνται προβλήματα στην κωδικοποίηση. Για παράδειγμα η χρήση δομημένης σχεδίασης, δίνει ένα σχέδιο λογισμικού του οποίου οι μονάδες πρέπει να αντιστοιχιστούν σε κατάλληλες κλάσεις/αντικείμενα και iεραρχικές δομές κλάσεων. Κάτι τέτοιο εισάγει πρόσθετες δυσκολίες σε μια ήδη δύσκολη διαδικασία και είναι προτιμότερο να αποφευχθεί. Έτσι εφόσον επιλεγεί το αντικειμενοστραφές μοντέλο κύκλου ζωής, έστω και αν για την φάση της ανάλυσης του ΣΔΔ χρησιμοποιηθεί δομημένη ανάλυση, στη φάση της σχεδίασης πρέπει να εφαρμοστεί αντικειμενοστραφής τεχνική.

Όπως και για την ανάλυση, έτσι και για την αντικειμενοστραφή σχεδίαση δεν έχουν αναπτυχθεί τυποποιήσεις, ενώ έχουν αναπτυχθεί αρκετές τεχνικές [BOOC86] [ΣΚΟΡ91] [COAD91]. Όλες οι μέθοδοι χρησιμοποιούν τα αποτελέσματα κάποιας μεθόδου ανάλυσης (κυρίως κάποιας αντίστοιχης αντικειμενοστραφούς ανάλυσης) και προσθέτουν λεπτομέριες που χρειάζονται για την συγκεκριμένη υλοποίηση. Αυτές περιλαμβάνουν στοιχεία διαπροσωπείας χρήστη, μηχανισμούς διαχείρισης διεργασιών, αποθήκευσης δεδομένων και άλλες λεπτομέριες υλοποίησης. Το σχέδιο λογισμικού που παράγεται ως τελικό προϊόν αυτής της φάσης έχει μορφή παρόμοια με τα αποτελέσματα της ανάλυσης : χρησιμοποιούνται διαγράμματα για να παρουσιάσουν τις κλάσεις, τις δομές iεραρχίας και την επικοινωνία με μνήματα, ψευδοκώδικας για την περιγραφή των υπηρεσιών των κλάσεων κτλ, αλλά παρουσιάζεται περισσότερη λεπτομέρια και η φιλοσοφία είναι πιο κοντά στην υλοποίηση. Ακόμη, στην φάση της σχεδίασης, αποφασίζεται αν θα επαχρησιμοποιηθούν κάποιες ήδη σχεδιασμένες κλάσεις (που μπορεί και να έχουν κωδικοποιηθεί).

Το επόμενο στάδιο είναι η φάση της **κωδικοποίησης**. Εδώ το σχέδιο λογισμικού μετατρέπεται σε κώδικα της συγκεκριμένης γλώσσας υλοποίησης. Συνήθως, υπάρχει μια αντίστοιχεία ένα προς ένα μεταξύ των στοιχείων του σχεδίου λογισμικού και μονάδων κώδικα. Γράφεται, λοιπόν, νέος κώδικας και επαναχρησιμοποιείται όσο το δυνατόν κώδικας που ήδη έχει γραφτεί. Η επικάλυψη μεταξύ των σταδίων σχεδίασης και κωδικοποίησης στο αντικειμενοστραφές μοντέλο, δείχνει ότι η κωδικοποίηση μπορεί να ξεκινήσει πριν τελειώσει η σχεδίαση. Πολλές τεχνικές αντικειμενοστραφούς σχεδίασης

καταφεύγουν στην κωδικοποίηση προτοτύπων (prototyping), υλοποιούν δηλαδή τις μέχρι τότε σχεδιασμένες κλάσεις, ελέγχουν την λειτουργία τους και ολοκληρώνουν την σχεδίαση μέσα από μια επαναλαμβανόμενη διαδικασία επανασχεδίασης-δημιουργίας προτοτύπου.

Μετά το στάδιο κωδικοποίησης ακολουθεί το στάδιο **ελέγχου** που χωρίζεται σε **φάση ελέγχου μονάδας λογισμικού**, όπου ελέγχονται οι μονάδες ξεχωριστά και **φάση ελέγχου του συστήματος**, όπου ελέγχεται η συνολική ορθότητα και λειτουργικότητα του συστήματος. Αν εντοπιστούν λάθη, προκαλείται, όπως φαίνεται και στο σχήμα 8.7, οπισθοδρόμηση στην φάση κωδικοποίησης. Ακολουθεί το στάδιο **χρήσης του προγράμματος** από το οποίο οπισθοδρόμηση γίνεται μόνο στην περίπτωση εντοπισμού λάθους που ξέφυγε από την φάση ελέγχου, η οποία μπορεί να φτάσει μέχρι την αρχή του πίδακα. Τέλος, τα στάδια **συντήρησης** και **επέκτασης** ενεργοποιούνται όταν υπάρχει ανάγκη συντήρησης ή επέκτασης αντίστοιχα, και προκαλούν οπισθοδρόμηση σε κάποια προηγούμενη φάση. Πρέπει να τονιστεί και πάλι ότι στο αντικειμενοστραφές μοντέλο κύκλου ζωής αυτά τα δύο στάδια απαιτούν λιγότερη προσπάθεια και χρόνο σε σχέση με άλλα μοντέλα και προσφέρουν μεγαλύτερη δυνατότητα επαναχρησιμοποίησης έτοιμων μονάδων.

## 8.6. Έμπειρα συστήματα

Στις παραγράφους που ακολουθούν θα εξετάσουμε την χρησιμοποίηση έμπειρων συστημάτων στην ανάπτυξη Συστημάτων Διαχείρισης Δικτύων.

### 8.6.1. Έμπειρα συστήματα και διαχείριση δικτύων

Η χρήση των έμπειρων συστημάτων στη διαχείριση δικτύων τα τελευταία χρόνια γίνεται σε όλο και μεγαλύτερο βαθμό. Και αυτό είναι αναμενόμενο αν σκεφτεί κανείς ότι το μέγεθος, η πολυπλοκότητα και η ανομοιομορφία των δικτύων αυξάνει ραγδαία, με αποτέλεσμα να γίνονται όλο και περισσότερες οι γνώσεις που χρειάζονται για την παρακολούθηση και συντήρηση αυτών.

Τα έμπειρα συστήματα, λοιπόν, καθώς έχουν τη δυνατότητα να αυτοματοποιούν την εξαγωγή αποφάσεων, χρησιμοποιώντας κανόνες που βασίζονται στη γνώση και την εμπειρία ειδικών, μπορούν να παίζουν σημαντικό ρόλο καθώς περιορίζουν το χρόνο και την κούραση που απαιτείται από το χρήστη του συστήματος.

Μέσω ενός έμπειρου συστήματος ο manager ενός δικτύου μπορεί να ενημερωθεί για βέλτιστες λύσεις που αφορούν για παράδειγμα, τον βαθμό χρησιμοποίησης κάποιων links ή την τοπολογία του δικτύου, ώστε να γίνεται σωστότερη κατανομή των πόρων του συστήματος και αποδοτικότερη χρησιμοποίηση αυτών.

Μπορεί ακόμη να παίρνει αυτόματα αναφορές για την κατάσταση του δικτύου, και να ενημερώνεται εγκαίρως για προβληματικές καταστάσεις που πιθανά να συμβούν στο μέλλον.

Αναμφισβήτητα όμως εκεί που τα έμπειρα συστήματα έχουν την πιο διαδεδομένη χρήση, είναι στη διαχείριση λαθών (Fault Management). Προβλήματα στην μετάδοση των πληροφοριών στο φυσικό επίπεδο, ελαττωματικά links, interfaces, bridges ή routers ή ακόμα και προβλήματα υπερχρησιμοποίησης κάποιων στοιχείων του δικτύου λόγω λανθασμένης δρομολόγησης, μπορούν να απομονωθούν και να διαχωριστούν από τα λεγόμενα side-effects που τα ίδια προκαλούν. Τα τελευταία μάλιστα είναι εκείνα που κάνουν τη διάγνωση ενός λάθους πολύ δύσκολη, καθώς λόγω της πολυπλοκότητας των

δικτύων, εμφανίζονται αλυσιδωτές ακολουθίες γεγονότων που πολλές φορές οδηγούν τη διαδικασία της διάγνωσης σε λάθος κατεύθυνση.

### 8.6.2. Τι κάνει ένα έμπειρο σύστημα

Αν επιχειρήσουμε να καταγράψουμε με αυστηρό τρόπο τις βασικές λειτουργίες ενός έμπειρου συστήματος, αυτές είναι τρείς :

- Προσδιορισμός του προβλήματος ή χαρακτηρισμός μιας κατάστασης ως προβληματικής
- Διάγνωση της πραγματικής ή της πιο πιθανής αιτίας του προβλήματος
- Προτάσεις για την αντιμετώπιση της κατάστασης, ή αυτόματη διόρθωση του λάθους.

### 8.6.3. Κατηγορίες έμπειρων συστημάτων

Ανάλογα με τις ικανότητές τους τα έμπειρα συστήματα κατατάσσονται στις εξής κατηγορίες :

- **True Expert Systems:** Παρέχουν αυτόματη διάγνωση λαθών και προτείνουν βέλτιστες λύσεις για την αντιμετώπιση αυτών.
- **Apprentice Systems:** Βοηθούν στην αντιμετώπιση προβλημάτων εκτελώντας διαγνωστικά τέστ και αναφέρωντας τα αποτελέσματα σε κάποιον ειδικό.
- **Job Aids:** είναι το μεγαλύτερο μέρος των έμπειρων συστημάτων που υπάρχουν σήμερα. Βοηθούν το χρήστη να αντιμετοπίσει προβληματικές καταστάσεις δίνοντάς του κάποιους κανόνες που πρέπει να ακολουθήσει. Με άλλα λόγια απλώς καθοδηγεί το χρήστη, χωρίς όμως να μπορεί να ενεργήσει χωρίς την παρουσία αυτού.

### 8.6.4. Στοιχεία που αποτελούν το έμπειρο σύστημα

Ένα έμπειρο σύστημα αποτελείται από τρία μέρη :

- Τη βάση γνώσης (Knowledge Base)
- Τη μονάδα που αξιοποιεί τη γνώση (Inference Engine)
- Τη διαπροσωπεία χρήστη (User Interface)

#### *Βάση Γνώσης*

Η βάση γνώσεως περιέχει γεγονότα, κανόνες που αφορούν τις σχέσεις μεταξύ γεγονότων, καθώς και στρατηγικές ή ευριστικές μεθόδους και ιδέες για την εξαγωγή συμπερασμάτων. Βέβαια, όλα τα παραπάνω προέρχονται από τις γνώσεις και την εμπειρία ενός ειδικού πάνω στο συγκεκριμένο θέμα, καθώς επίσης και από τον κορμό της γνώσης που σχετίζεται με το θέμα, που στο σύνολό του ονομάζεται Domain.

Για παράδειγμα για ένα δίκτυο υπολογιστών σάν domain θεωρούμε μια ακολουθία συσχετιζόμενων γεγονότων που εμφανίζονται μέσω πολύπλοκων ενεργειών από τους χρήστες του δικτύου.

Μια πραγματικά αποδοτική βάση γνώσης δηλαδή περιέχει πολύ περισσότερα στοιχεία από απλά γεγονότα. Όπως ακριβώς και ένας άνθρωπος έχει τη γνώση και την εμπειρία πάνω στη δουλειά του, έτσι και στο έμπειρο σύστημα είναι ενσωματωμένες τόσο οι λογικές διαδικασίες που χρειάζονται, όσο και η απαραίτητη εμπειρία.

Όλα τα παραπάνω ενσωματώνονται στη βάση γνώσης με τη μορφή κανόνων. Οι κανόνες αυτοί πρέπει να είναι όσο το δυνατό περιορισμένοι σε αριθμό, ενώ ταυτόχρονα να είναι ιδιαίτερα περιεκτικοί και αποτελεσματικοί. Κάτι τέτοιο βεβαίως είναι αρκετά δύσκολο, διότι ακόμη και ένας άνθρωπος ειδικός μπορεί μεν να καταλήξει πολύ γρήγορα σε κάποια συμπεράσματα, βασιζόμενος όμως στην εμπειρία που έχει αποχτήσει με την εργασία πολλών ετών και χωρίς ο ίδιος να έχει ουσιαστική επίγνωση του ΠΩΣ έφτασε στα συμπεράσματα αυτά. Είναι λοιπόν επόμενο ότι το να αναλύσει κανείς τα συμπεράσματα ενός ανθρώπου βήμα πρός βήμα είναι μια πολύ δύσκολη υπόθεση.

### *Inference Engine*

Η μονάδα αξιοποίησης της γνώσης ουσιαστικά είναι η διαδικασία που παράγει τη λύση σε κάποιο συγκεκριμένο πρόβλημα. Ο μηχανισμός λειτουργίας της inference engine μπορεί να είναι δύο ειδών :

- forward chaining
- backward chaining

Κατά την forward chaining η inference engine χρησιμοποιεί τα παρεχόμενα δεδομένα από το σύστημα ή το χρήστη, για να παίρνει τις αποφάσεις της. Με τον τρόπο αυτό μάλιστα δίνεται η δυνατότητα προσδιορισμού της πιθανότητας επιτυχίας κάποιας διάγνωσης.

Αντίθετα κατά την backward chaining αρχικά παράγεται κάποια υπόθεση που αφορά την αιτία κάποιου γεγονότος, και στη συνέχεια γίνεται η αναζήτηση των στοιχείων εκείνων που θα επιβεβαιώσουν, ή θα απορίψουν την αρχική υπόθεση. Με την απόρριψη αυτή ή επιβεβαίωση υποθέσεων η inference engine καταλήγει σε συμπεράσματα.

### *User Interface*

Η διαπροσωπεία χρήστη έχει σαν βασικό σκοπό την ομαλή επικοινωνία μεταξύ χρήστη και συστήματος, ενώ συγχρόνως προσφέρει στο χρήστη τη δυνατότητα να βλέπει μέσα από διαγράμματα την όλη κατάσταση του συστήματος.

## 8.6.5. Υλοποίηση εμπείρων συστημάτων

Τα έμπειρα συστήματα υλοποιούνται με εργαλεία που υποστηρίζουν declarative programming και όχι διαδικασιακό προγραμματισμό (procedural programming). Αυτό συμβαίνει διότι στον διαδικασιακό προγραμματισμό πρέπει να υλοποιηθούν αλγόριθμοι που να περιγράφουν με απόλυτη ακρίβεια το πρόβλημα σε όλες του τις διαστάσεις. Ετσι γιά παράδειγμα αν θέλαμε να περιγράψουμε ένα συγκεκριμένο πρόβλημα σε δίκτυα υπολογιστών, θα έπρεπε με βάση τα παραπάνω να δώσουμε την ακριβή σειρά των βημάτων για τον προσδιορισμό του προβλήματος, συμπεριλαμβάνοντας την αρχή της

διαδικασίας, τον αριθμό των μεταβλητών που θα χρησιμοποιήσουμε, καθώς και πολλά άλλα στοιχεία τα οποία είναι δύσκολο να προσδιοριστούν.

Αντίθετα στον declarative programming πρέπει απλώς να δηλώσουμε στον υπολογιστή τις συνθήκες εκείνες που πρέπει να υπάρχουν για να γίνει ο προσδιορισμός του προβλήματος.

## 8.7. Ασκήσεις

1. Αναφέρατε διάφορες αρχιτεκτονικές για τη σχεδίαση ενός διαχειριστικού συστήματος. Ποιες από αυτές είναι εφικτές, στην περίπτωση που θα χρησιμοποιηθεί το SNMP σαν πρωτόκολλο διαχείρισης;
2. Αναφέρατε νέες προσπάθειες στην κοινότητα TCP/IP που θα μπορούσαν να υποστηρίζουν επιπλέον αρχιτεκτονικές.

## 8.8. Βιβλιογραφία

- [AGRE86] Agresti, *New Paradigms for Software Development*, IEEE Computer Society Order Number 707, IEEE Computer Society Press, 1986.
- [ALMR92] Information processing systems - Open System Interconnection - *Systems Management: Alarm reporting function - International Organization for Standardization - International Standard 10164-4 - December 1992.*
- [ANAD91] Anand V. Rao 'An Introduction to Expert Systems in Network Management', DATAPRO September 1991.
- [ANDE91] D.R. Andersen, T.T. Bezoza, P.A. Johnson, M.R. Moroses, D.J. Sidor, *Application of Object-oriented Techniques to the OAM&P of Telecommunications Networks*, IEEE ICC '91, 1991.
- [BAIL89] S.C. Bailin, *An Object-oriented Requirements Specification Method*, CACM, Vol. 32, No 5, 1989.
- [BCOX87] B.J. Cox, *Object-oriented Programming : An Evolutionary Approach*, Addison Wesley, 1987.
- [BLCK92] Black, Uyless. *Network Management Standards: SNMP, CMOT, & OSI*. 1992. McGraw-Hill, Incorporated.
- [BOEH84] B. Boehm, *Software Life Cycle Factors, Handbook of Software Engineering*, Van Nostrand Reinhold, New York, 1984.
- [BOOC86] G.R. Booch, *An Object-oriented Development*, IEEE Transactions on Software Engineering, Vol. SE-12, No 2, 1986.

- [BOWE90] Bowen, Ken 'Prolog and Expert Systems', McGRAW-HILL, 1990.
- [BRAT89] Ivan Bratko 'PROLOG - Programming For Artificial Intelligence' Second Edition / Addison-Wesley, 1989.
- [CALL88] Paul H. Callahan 'Expert Systems for AT&T Switched Network Maintenance' 1988 AT&T.
- [CASE89] Jeffrey D. Case, James R. Davin, Mark S. Fedor, and Martin L. Schoffstall. *Network Management and the Design of SNMP*, ConneXions - The Interoperability Report, 3(3):22-26, March, 1989.
- [CASS89] Lillian N. Cassel, Graig Patridge, and Jil Westcott. *Network Management Architectures and Protocols : Problems and Approaches*, IEEE Journal on Selected Areas in Communications, Vol. 7, no. 7, Sep. 89.
- [CHAO90] C.W. Chao, P. Sarachik, B. Maglaris, R. Boorstyn, and D. Dimitrijevic, "Control of Multi-Domain Networks", Network Management and Control, Edited by A. Kershbaum *et al.*, Plenum Press, New York, 1990.
- [COAA91] P. Coad and E. Yourdon. *Object-oriented Analysis*. Yourdon Press, 1991.
- [COAD91] P. Coad and E. Yourdon. *Object-oriented Design*. Yourdon Press, 1991.
- [DEMA78] T. Demacro, *Structured Analysis and System Specification*, Yourdon Press, New York, 1978.
- [DIMI89] D.D. Dimitrijevic, B. Maglaris, R.R. Boorstyn, "Routing in Multiple Domain Networks", Proceedings of the IEEE INFOCOM-89, Ottawa, Canada, April 1989.
- [ERIC89] Ericson, Eric Ericson, Lisa T. & Minoli, Daniel., editors. *Expert Systems Applications in Integrated Network Management* (Artech House Telecom Engineering Library). 1989. Artech House, Incorporated.
- [EVNT91] Information processing systems - Open System Interconnection - *Systems Management: Event Report Management Function - International Organization for Standardization - International Standard 10164-5 - August 1991*.
- [FAIR85] R.E. Fairley, *Software Engineering Concepts*, McGraw-Hill, 1985.
- [GANE79] C. Gane and T. Sarson, Structured Systems Analysis : Tools and Techniques, Prentice-Hall, Egglewood Cliffs, N.J., 1979.
- [GEOF87] C. Geoff, Structured Systems Analysis and Design Methodology, Paradigm, 1987.
- [GUTT93] Michael K. Guttman. *Client/Server Computing : Emerging Trends, Solutions, and Strategies*, DATAPRO International, Communications & Networking Solutions, February 1993.

- [HALS92] Fred Halsall, *Data Communications, Computer Networks and Open Systems*, Chapter 12, Addison-Wesley, 1992.
- [HEGE93] Symposium on Integrated Network Management. Hegering, Heinz-Gerd, editor. Yemini, Yechiam, editor. 03/1993. Elsevier Science Publishing Company, Incorporated.
- [HELD92] Held, Gilbert. *Network Management: Techniques, Tools & Systems*. 1992. Wiley, John, & Sons, Incorporated.
- [HEND90] B. Henderson-Sellers and J.M. Edwards, *The Object-Oriented Systems Life Cycle*, CACM, Vol. 33, No. 9, 1990.
- [HERM90] James Herman, "Enterprise Management Vendors Shoot It Out", Data Communications International, November 1990.
- [IEEE83] ANSI/IEEE Std 729-1983, *Glossary of Software Engineering Terminology*, 1983.
- [IEEE84] ANSI/IEEE Std 830-1984, *Software Requirements Specifications*, IEEE Guide to Software, 1984.
- [KAUF91] Kauffels, Franz-Joachim. *Network Management: Problems, Standards, Strategies*. 1991. Addison-Wesley Publishing Company, Incorporated.
- [KRIS91] Krishman, I. & Zimmer, W. *Integrated Network Management, No. II*. 1991. Elsevier Science Publishing Company, Incorporated.
- [KRIS93] Krishnan, Iyengar. *Integrated Network Management*. 1993. McGraw-Hill, Incorporated.
- [LEIN93] Allan Leinwand, Karen Fang. *Network Management : A Practical Perspective*, Addison Wesley, 1993.
- [LOGC91] Information processing systems - Open System Interconnection - *Systems Management: Log Control Function - International Organization for Standardization - International Standard 10164-6* - August 1991.
- [MCGR90] J.D. McGregor and T.Korson, *Understanding Object-oriented : A Unifying Paradigm*, CACM, No 9, 1990.
- [MEAN89] Meandzja, B. & Westcott, J., editors. *Integrated Network Management: Proceedings of the IFIP TCG WG6.6 Symposium*, Boston , MA, 16-17 May, 1989, Vol. 1. Elsevier Science Publishing Company, Incorporated.
- [MEYE88] B. Meyer, *Object-oriented Software Construction*, Prentice Hall, 1988.
- [MULL89] Mark Mullin, *Object Oriented Program Design*, Addison-Wesley, 1989.
- [MULL93] Nathan J. Muller. *Object-oriented Networking*, DATAPRO International, Communications & Networking Solutions, February 1993.
- [NMIS91] Network Management Information Service, *OSI-Based Network Management*, DATAPRO International, 1991.

- [OBJM91] Information processing systems - Open System Interconnection - *Systems Management: Object Management Function - International Organization for Standardization - International Standard 10164-1* - August 1991.
- [OSIM89]. Information processing systems - Open System Interconnection - *OSI Management Framework - International Organization for Standardization - International Standard 7498/4* - April 1989.
- [RABI92] Sameh Rabie. *Integrated Network Management : Technologies and Implementation Experience*, Infocom '92, IEEE, 1992.
- [RAMA84] C. Ramamoorthy, A. Praksh, W. Tsai and Y. Usuda, *Software Engineering Problems and Perspectives*, Computer, Oct. 1984.
- [ROSS87] D.T. Ross, *Structured Analysis (SA) : A Language for Communicating ideas*, IEEE Transactions on Software Engineering, SE-3, No 1, 1987.
- [ROSE91] Marshall T. Rose, *The Simple Book: An Introduction to Management of TCP/IP - based Internets*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [SCAD91]. Information processing systems - Open System Interconnection - *Systems Management: Security audit trail function - International Organization for Standardization - International Standard 10164-8* - August 1991.
- [SCHA92] Schatt, Stan. *Understanding Network Management: Strategies & Solutions*. 1992. T A B Books.
- [SECU92]. Information processing systems - Open System Interconnection - *Systems Management: Security alarm reporting function - International Organization for Standardization - International Standard 10164-7* - May 1992.
- [ΣΚΟΡ91] Εμμ. Σκορδαλάκη, *Εισαγωγή στην Τεχνολογία Λογισμικού*, Εκδόσεις Συμμετρία, Αθήνα 1991.
- [SMFA] International Systems Organization, *Specific Management Functional Areas*, N4981, N875R, N4091, N4077, N3311.
- [SMIB92] Information technology - Open Systems Interconnection - *Structure of management information : Definition of management information* ISO/IEC 10165-2, 10-15-1992.
- [SMGO92] Information processing systems - Open Systems Interconnection - *Systems Management Overview - International Organization for Standardization - International Standard 10040* - November 1992.
- [ST2393] "The Simple Times", The Bi-Monthly Newsletter of SNMP Technology, Comment, and Events, Volume 2, Number 3, May/June 1993.
- [STAL93] Stallings, William. *SNMP, SMP, & CMIP: The Practical Guide to Network Management Standards*. 1993. Addison-Wesley Publishing Company, Incorporated.

- [STAM92]. Stamatelopoulos F., Stathatos K., Karounos T., Maglaris B., "*Cerberus Network Management System*", ERSIM International Workshop, Crete, Greece, 1992.
- [STAT91] Information processing systems - Open System Interconnection - *Systems Management: State Management Function - International Organization for Standardization - International Standard 10164-2 - August 1991.*

## Κεφάλαιο 9

### 9. Ενοποιημένη Διαχείριση: Οι Αρχιτεκτονικές UNMA (της AT&T), EMA (της DEC), NetView και SystemView (της IBM) και OpenView (της HP)

#### Περιεχόμενα του Κεφαλαίου 9

- 9.0. Εισαγωγή
- 9.1. Χαρακτηριστικά των ολοκληρωμένων διαχειριστικών συστημάτων
- 9.2. Η αρχιτεκτονική Unified Network Management System (UNMA)
  - 9.2.1. Μια γενική άποψη της αρχιτεκτονικής
  - 9.2.2. To Network Management Protocol (NMP)
  - 9.2.3. Προσαρμοστικότητα της αρχιτεκτονικής UNMA
  - 9.2.4. Προϊόντα και υπηρεσίες βασιζόμενα στην UNMA αρχιτεκτονική
- 9.3. Η αρχιτεκτονική Enterprise Management Architecture (EMA)
  - 9.3.1. Περιγραφή της αρχιτεκτονικής EMA
  - 9.3.2. Διαχειριστικά προϊόντα της DEC
- 9.4. Η αρχιτεκτονική NetView
  - 9.4.1. Χαρακτηριστικά του NetView
  - 9.4.2. NetView Tools
- 9.5. Το περιβάλλον διαχείρισης OpenView
  - 9.5.1. Γενικά χαρακτηριστικά
  - 9.5.2. Η δομή του OpenView
  - 9.5.3. Κατανεμημένο περιβάλλον διαχείρισης (Distributed Management Environment).
- 9.6. Ολοκληρωμένη διαχείριση ετερογενών συστημάτων
- 9.7. Ασκήσεις
- 9.8. Βιβλιογραφία

#### 9.0. Εισαγωγή

Στα σημερινά μεγάλα ετερογενή δίκτυα, μπορεί κανείς να συναντήσει μηχανήματα, τα οποία προέρχονται από ένα πολύ μεγάλο αριθμό κατασκευαστών. Τα μηχανήματα αυτά, τις περισσότερες φορές - αν όχι όλες - μπορούν και επικοινωνούν μεταξύ τους. Για παράδειγμα, ένα DC Hayes modem μπορεί και επικοινωνεί με ένα U.S.Robotics

modem. Δυστυχώς όμως, το να βρεθεί κάποιο διαχειριστικό σύστημα, το οποίο να μπορεί να ελέγξει και τα δύο αυτά modem, είναι κάτι εντελώς διαφορετικό και όχι τόσο απλό.

Στο Κεφάλαιο αυτό θα εξετάσουμε τα διάφορα συστήματα διαχείρισης δικτύων, τα οποία μπορεί να βρει κανείς σήμερα στην αγορά. Για αρκετά από αυτά, μπορούμε να πούμε, ότι προσφέρουν δυνατότητες για ενοποιημένη διαχείριση (Integrated Management), από την άποψη, ότι μπορούν να διαχειριστούν μια μεγάλη σειρά από ετερογενή επικοινωνιακά συστήματα.

Στα συστήματα που θα εξετάσουμε περιλαμβάνονται, οι αρχιτεκτονικές UNMA της AT&T, EMA της DEC, NetView και SystemView της IBM και τέλος το OpenView της HP. Πριν, όμως προχωρήσουμε σ' αυτά θα παρουσιάσουμε μερικά από τα σημαντικότερα χαρακτηριστικά, που μπορεί να συναντήσει κανείς σε ένα ολοκληρωμένο διαχειριστικό σύστημα.

## **9.1. Χαρακτηριστικά των ολοκληρωμένων διαχειριστικών συστημάτων**

Θα παρουσιάσουμε, στην συνέχεια μερικά από τα απαραίτητα χαρακτηριστικά που πρέπει να κατέχει κάποιο διαχειριστικό σύστημα, προκειμένου αυτό να μπορεί να χαρακτηρισθεί σαν ολοκληρωμένο. Μπορεί βέβαια μερικά διαχειριστικά συστήματα να προσφέρουν έναν μεγαλύτερο αριθμό από δυνατότητες στον χρήστη τους, αλλά σχεδόν όλα προσφέρουν τον κορμό δυνατοτήτων, που θα αναπτύξουμε παρακάτω.

### **1. Password access**

Τα πιο πολλά διαχειριστικά συστήματα χρησιμοποιούν κωδικούς ασφαλείας, προκειμένου να απαγορεύουν την πρόσβαση σε μη υπεύθυνο προσωπικό. Πιο περίπλοκα συστήματα μάλιστα, προσφέρουν διαφορετικά επίπεδα δικαιωμάτων κατά την πρόσβαση, προκειμένου να επιτρέπουν και σε μη υπεύθυνο προσωπικό και χρήστες να έχει πρόσβαση σε κάποιο κομμάτι της διαχειριστικής πληροφορίας, που το ενδιαφέρει.

### **2. Color-coded graphics, text display**

Τα περισσότερα διαχειριστικά συστήματα χρησιμοποιούν κάποιο χρωματικό κώδικα, προκειμένου να φανερώνουν την σπουδαιότητα κάποιας κατάστασης, που παρουσιάζουν. Ένας συνηθισμένος κώδικας χρησιμοποιεί τα χρώματα: κόκκινο για συναγερμούς, που απαιτούν άμεση ενέργεια για την επίλυση του σχετικού προβλήματος, κίτρινο για σοβαρές καταστάσεις, που αν δεν αντιμετωπιστούν μπορεί να οδηγήσουν σε συναγερμούς, και τέλος πράσινο για φυσιολογικές καταστάσεις.

Όλα αυτά ισχύουν βέβαια σε περίπτωση που η διαπροσωπεία της διαχειριστικής εφαρμογής υποστηρίζει γραφικά. Μέχρι πρόσφατα, οι περισσότερες διαχειριστικές εφαρμογές για προσωπικούς υπολογιστές ήταν command-driven και text-based. Η εισαγωγή όμως, των Microsoft Windows επέτρεψε στους μηχανικούς λογισμικού να υλοποιήσουν διαχειριστικές εφαρμογές με γραφικά σε προσωπικούς υπολογιστές.

### **3. Status reporting**

Ένας από τους σκοπούς κλειδιά ενός διαχειριστικού συστήματος, είναι η ελάττωση του χρόνου, που ξοδεύει κάποιος τεχνικός, προκειμένου να ελέγξει κάποιο στοιχείο του δικτύου. Με τον τρόπο αυτό ελαττώνονται βέβαια και οι ώρες, που το στοιχείο αυτό μπορεί να μείνει εκτός λειτουργίας.

Τα διαχειριστικά συστήματα με κατάλληλα μηνύματα, μπορούν να ζητήσουν από το ίδιο στοιχείο να τους αναφέρει την κατάσταση στην οποία βρίσκεται, μέσω των τιμών κάποιων μεταβλητών. Ο τεχνικός βλέποντας τις τιμές αυτές πολλές φορές μπορεί να αποφασίσει, κατά πόσον το στοιχείο απαιτεί κάποια επισκευή, ή λειτουργεί κανονικά. Επίσης οι μετρήσεις αυτές μπορούν να αποθηκεύονται σε μια βάση δεδομένων για μετέπειτα χρήση.

#### 4. Performance monitoring

Η παρακολούθηση των επιδόσεων μπορεί να συμπεριληφθεί σαν μέρος της δυνατότητας του διαχειριστικού συστήματος για την δημιουργία αναφορών, σχετικά με την κατάσταση των στοιχείων του δικτύου, αλλά μπορεί να αποτελέσει και ανεξάρτητο ιδιαίτερο χαρακτηριστικό του διαχειριστικού συστήματος. Πολύ γενικά θα μπορούσαμε να πούμε (χωρίς να ακριβολογούμε), ότι η αναφορά καταστάσεως αφορά κυρίως τα μηχανήματα του δικτύου, ενώ η παρακολούθηση των επιδόσεων αφορά τους μηχανισμούς σύνδεσης και επίτευξης της επικοινωνίας μεταξύ των μηχανημάτων αυτών.

#### 5. Configuration management

Η διαχείριση διάρθρωσης παρέχει στον χρήστη ενός συστήματος διαχείρισης δικτύων την ικανότητα να μεταβάλλει την λειτουργική κατάσταση των διαφόρων κόμβων, οι οποίοι βέβαια μπορούν να βρεθούν σε διάφορες λειτουργικές καταστάσεις. Για παράδειγμα αναφέρουμε κόμβους, όπως multiport modems, multispeed modems, multiplexers.

Η διαχείριση διάρθρωσης συχνά χρησιμοποιείται για να ρυθμίσει κάποιο δίκτυο. Η ρύθμιση αυτή μπορεί να επιτευχθεί, είτε αυτόματα, με κατάλληλη περιγραφή, που έχει δοθεί στο μηχάνημα σε προηγούμενη χρονική στιγμή, είτε με χειρισμούς του διαχειριστή την ίδια χρονική στιγμή.

#### 6. Threshold setting, alarm generation

Η δυνατότητα για την δημιουργία κατώφλιών, όπως επίσης η δυνατότητα συσχετισμού αυτών με συναγερμούς, επιτρέπει σε ένα διαχειριστικό σύστημα να δίνει προειδοποιήσεις για καταστάσεις, οι οποίες αν μείνουν ανεξέλεγκτες μπορεί να οδηγήσουν σε σφάλματα στο δίκτυο. Για παράδειγμα, μπορεί ένας διαχειριστής να ορίσει κάποιο συναγερμό για κάποια γραμμή, που είναι συνδεδεμένη σε ένα modem. Το κατώφλι για το συναγερμό μπορεί να βρίσκεται πιο κάτω από το συνηθισμένο πλάτος του σήματος στη γραμμή και πιο πάνω από την ευαισθησία του modem. Αν το σήμα λοιπόν στην γραμμή αυτή πέσει κάτω από το όριο αυτό, τότε ένας συναγερμός θα εκδοθεί. Ο συναγερμός μπορεί να είναι κάποιο ηχητικό σήμα, κάποιος κόκκινος χρωματισμός, κατά την αναπαράσταση μιας γραμμής σε οθόνη γραφικών, κάποιο σχετικό μήνυμα σε εκτυπωτή, είτε συνδυασμός όλων των παραπάνω.

#### 7. Database maintenance, report generation

Τα περισσότερα διαχειριστικά συστήματα μπορούν γρήγορα και αξιόπιστα να ανακτήσουν δεδομένα, τα οποία αφορούν προηγούμενες επιδόσεις ή κατάστασεις λειτουργίας μηχανημάτων και γραμμών του δικτύου. Πολλά συστήματα μπορούν ακόμα να δημιουργούν, ανανεώνουν, αποθηκεύουν και ανακτούν δελτία βλάβης (trouble tickets), τα οποία καταγράφουν προβλήματα, όσο αναφορά συγκεκριμένα στοιχεία του δικτύου, καθώς και την ενέργεια που πραγματοποίηθηκε για την διόρθωσή τους.

## 8. Fault/problem management

Η διαχείριση σφαλμάτων και προβλημάτων μπορεί να θεωρηθεί ένα υπερσύνολο της ιδιότητας των διαχειριστικών συστημάτων, που περιγράφαμε παραπάνω για τους συναγερμούς. Στην διαχείριση σφαλμάτων και προβλημάτων ειδικά διαχειριστικά εργαλεία παρέχονται στους διαχειριστές, προκειμένου να έχουν την δυνατότητα αναγνώρισης και ανάλυσης σχετικών με το δίκτυο προβλημάτων. Τα εργαλεία αυτά συχνά εμπεριέχουν την δυνατότητα δημιουργίας κατωφλίων μαζί με την δυνατότητα συσχετισμού συναγερμών μ' αυτά.

Δύο επιπρόσθετα χαρακτηριστικά, που μπορεί να έχει κανείς με την διαχείριση σφαλμάτων/προβλημάτων είναι η δημιουργία κατωφλίων σε ένα σύνολο από μεταβλητές, και η δημιουργία δελτίων βλάβης. Σύμφωνα με το πρώτο, ο διαχειριστής μπορεί να συσχετίσει περισσότερες από μία μεταβλητές, έτσι ώστε οι συναγερμοί να βασίζονται σε υπέρβαση περισσότερων του ενός ορίων, και να έχουν κατ' αυτόν τον τρόπο πιο αληθινό χαρακτήρα. Μάλιστα, η χρησιμοποίηση τεχνητής νοημοσύνης στο σύστημα μπορεί να επιτρέψει την δημιουργία προειδοποίησεων πριν την έκδοση του συναγερμού.

Η δυνατότητα δημιουργίας κάποιων δελτίων βλάβης, που έχουν πολλά διαχειριστικά συστήματα, επιτρέπει στους διαχειριστές να χειρίζονται με μεγαλύτερη ευκρίνεια κάποια προβλήματα. Ας εξηγήσουμε όμως τι είναι ένα δελτίο βλάβης. Στην ουσία είναι μια αναφορά στην οποία αναφέρονται πέρα από το πρόβλημα, τα άτομα τα οποία ήρθαν σε επαφή με τους διαχειριστές για την επίλυση του προβλήματος, τα μηχανήματα τα οποία πιθανών χρειάζονται επισκευή προκειμένου να επιλυθεί το πρόβλημα, το προσδοκόμενο χρόνο επίσκεψης ειδικευμένου προσωπικού για την επισκευή κάποιων μηχανημάτων, και άλλες σχετικές πληροφορίες. Έτσι σε περίπτωση που για κάποιο λόγο χρειαστεί να λήψει κάποιος διαχειριστής ή αλλάξει κάποια βάρδια, ο αντικαταστάτης μπορεί να ασχοληθεί εξίσου καλά με το πρόβλημα.

Σε ορισμένα διαχειριστικά συστήματα, μόλις το πρόβλημα λυθεί το δελτίο βλάβης μπορεί να "κλείσει" και να αποσυρθεί σε κάποια βάση ιστορικών δεδομένων. Μ' αυτόν τον τρόπο κάποια χρονική στιγμή ο διαχειριστής μπορεί να ζητήσει πληροφορίες από μια τέτοια βάση δεδομένων, για την συχνότητα επανάληψης κάποιου προβλήματος, για κάποια λίστα προβλημάτων που δημιουργούνται από κάποιο κοινό αίτιο κ.ά.

## 9. Accounting/cost management

Ένα τελευταίο ενδιαφέρον χαρακτηριστικό ενός συστήματος διαχείρισης δικτύων, είναι η δυνατότητα ανάπτυξης μεθοδολογιών για τον καθορισμό του κόστους της επικοινωνίας, που επιτυγχάνεται διαμέσου του δικτύου. Από την στιγμή, που μια μεθοδολογία έχει αναπτυχθεί, οι ακριβείς υπολογισμοί για τον καθορισμό των μηνιαίων χρεώσεων για κάθε τμήμα μιας εταιρείας είναι μια επιπλεόν επίπονη διαδικασία. Μερικά διαχειριστικά συστήματα παρέχουν επίσης τις δυνατότητες εκείνες που χρειάζονται για να διευκολυνθεί η παραπάνω εργασία.

Λογιστική διαχείριση επιτυγχάνεται από διαχειριστικά συστήματα με χορήγηση κωδικών στους χρήστες των υπηρεσιών του δικτύου ή στους οργανισμούς που χρησιμοποιούν τις υπηρεσίες του δικτύου και στην συνέχεια με μέτρηση ανά κάθε κωδικό του χρόνου σύνδεσης με το δίκτυο, του αριθμού των πακέτων που διακομίστηκαν, και άλλων στατιστικών μεγεθών που προσδιορίζουν χρησιμοποίηση. Στην συνέχεια τα στοιχεία αυτά στην καλύτερη περίπτωση, μπορεί να χρησιμοποιούνται από το ίδιο το διαχειριστικό σύστημα για την εκτύπωση λογαριασμών, ή σε ενδιάμεσες καταστάσεις πολυπλοκότητας, μπορεί να προσδίδεται στα στοιχεία αυτά η κατάλληλη μορφή, ώστε να μπορούν να δωθούν σαν είσοδος σε κάποιο πρόγραμμα λογιστικών φύλλων.

Μερικά ακόμα χαρακτηριστικά, τα οποία συναντιούνται όμως πιο σπάνια σε διαχειριστικά συστήματα είναι τα παρακάτω.

### 1. Equipment, line facility inventory

Οι δυνατότητες αυτές επιτρέπουν στον διαχειριστή να κρατήσει βάσεις δεδομένων με πληροφορίες για τα στοιχεία του δικτύου και στην συνέχεια να μπορεί να δημιουργεί αναφορές με την βοήθεια των πληροφοριών αυτών. Οι βάσεις αυτές δεδομένων μπορούν να είναι κάποια εφαρμογή μεταξύ μιας απλής λίστας των στοιχείων του δικτύου, και μιας σύνθετης βάσης δεδομένων που επιτρέπει στους χρήστες να προγραμματίζουν μια σειρά από κριτήρια αποφάσεων για κάποια γεγονότα, όπως για παράδειγμα την έλλειψη κάποιων ανταλλακτικών από κάποια αποθήκη, απαραίτητων για την ομαλή λειτουργία του δικτύου.

Μια βασική τέτοια εφαρμογή επιτρέπει στον διαχειριστή να κρατά σε εγγραφές όλα τα μηχανήματα του δικτύου, τον κατασκευαστή τους, το σειριακό αριθμό τους, τον υπεύθυνο συντήρησης, το τηλέφωνο του υπεύθυνου συντήρησης και άλλες χρήσιμες πληροφορίες. Παρόμοιες εγγραφές πραγματοποιούνται και για τους συνδέσμους που χρησιμοποιούν τα παραπάνω στοιχεία, προκειμένου να επικοινωνούν μεταξύ τους. Αν ένας τέτοιος πίνακας συνδεθεί επιπλεόν με κάποιο δελτίο βλάβης τότε ο διαχειριστής μπορεί πια άνετα να λύσει, το οποιοδήποτε πρόβλημα άμεσα.

### 2. Directory, message routing

Μια τέτοια δυνατότητα, επιτρέπει στον διαχειριστή του συστήματος να εξετάζει τις εργασίες του υπολοίπου προσωπικού του δικτύου καθώς και τα αναγνωριστικά τους, και μέσω της δυνατότητας προώθησης μηνυμάτων να τους αφήνει κατάλληλα σημειώματα για άμεσες και επείγουσες εργασίες.

### 3. Customization

Μια και είναι αδύνατο, κάποιος κατασκευαστής διαχειριστικών συστημάτων να ικανοποιήσει όλους τους πιθανούς αγοραστές, πολλές κατασκευάστριες εταιρείες παρέχουν στα συστήματά τους μια περιορισμένη δυνατότητα προσαρμογής στις απαιτήσεις του χρήστη. Δηλ. προσφέρουν δυνατότητες μεταβολής κάποιων εξωτερικών (κυρίως) χαρακτηριστικών του συστήματος, έτσι ώστε αυτό να ανταποκρίνεται καλύτερα στις ανάγκες του χρήστη. Μερικές πιθανές αλλαγές είναι τα χρώματα της οθόνης, ο τρόπος παροχής εντολών στο σύστημα, η μορφή των αναφορών του συστήματος, η μορφή των επικεφαλίδων στις αναφορές του συστήματος κ.ά.

Έχοντας ολοκληρώσει την εξέταση των κυριότερων χαρακτηριστικών των συστημάτων διαχείρισης δικτύου, μπορούμε να συνεχίσουμε περιγράφοντας τις κυριότερες αρχιτεκτονικές, και τα σπουδαιότερα διαχειριστικά συστήματα που κυκλοφορούν στην αγορά.

## 9.2. Η αρχιτεκτονική Unified Network Management System (UNMA)

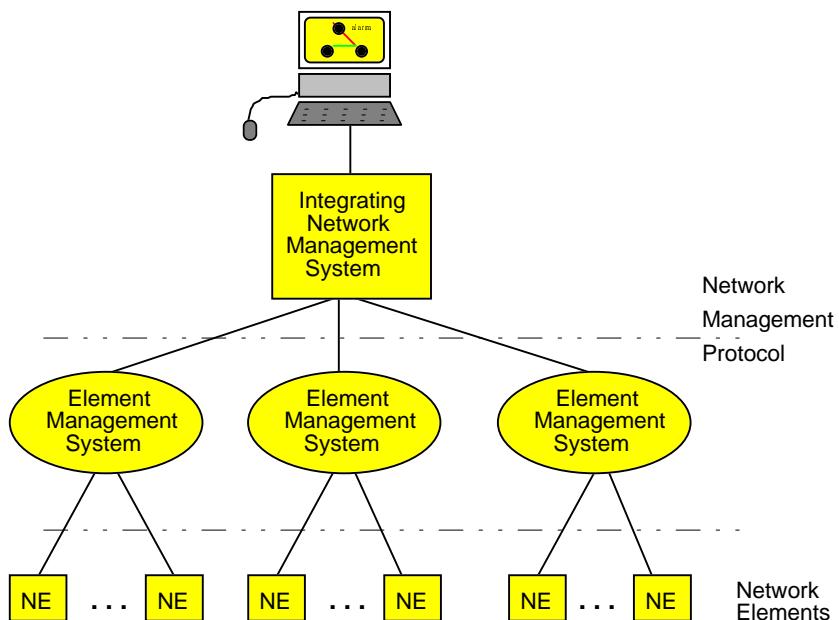
Θα ξεκινήσουμε με την αρχιτεκτονική Unified Network Management System (UNMA) της AT&T. Η AT&T ανακοίνωσε την αρχιτεκτονική αυτή τον Σεπτέμβριο του 1987 σαν ένα πλαίσιο για την επίτευξη ενοποιημένης από άκρη σε άκρη διαχείρισης δικτύων δεδομένων και φωνής σε ένα ετερογενές περιβάλλον.

Η φράση "από άκρη σε άκρη" είναι το πρώτο πλεονέχτημα της UNMA αρχιτεκτονικής, από την άποψη ότι περιλαμβάνει και διαχείριση παρεχομένων τηλεπικοινωνιακών υπηρεσίων, πέρα από την διαχείριση δικτύων δεδομένων. Το δεύτερο πλεονέχτημα είναι ότι η αρχιτεκτονική της AT&T είναι βασισμένη πάνω στην προσέγγιση των ανοικτών συστημάτων δηλ. στο πλαίσιο αναφοράς OSI, όσο αναφορά την διαχείριση.

Στην συνέχεια, θα περιγράψουμε την έκταση της αρχιτεκτονικής UNMA, έτσι ώστε να γίνουν κατανοητές οι περιοχές, καθώς και τα προϊόντα στα οποία είναι αυτή εφαρμόσιμη. Μαζί με τη γενική αρχιτεκτονική, θα αναλύσουμε και το σχετικό πρωτόκολλο, το οποίο χρησιμοποιείται για τη μεταφορά διαχειριστικής πληροφορίας, προκειμένου να επιτευχθεί η ενοποιημένη διαχείριση δικτύων δεδομένων και φωνής.

### 9.2.1. Μια γενική άποψη της αρχιτεκτονικής

Στο Σχήμα 9.1 φαίνεται η αρχιτεκτονική που προτείνει η AT&T σε μια δομή δένδρου. Χαμηλά στο δένδρο υπάρχουν διάφορα διαχειριζόμενα στοιχεία (network elements), όπως modems, multiplexers, DSUs/CSUs, PBXs, hosts, LANs κ.ά. Τα στοιχεία αυτά, τα οποία προέρχονται από διαφορετικούς κατασκευαστές, εξακολουθούν να χρησιμοποιούν ειδικά πρωτόκολλα (όχι υποχρεωτικά OSI πρωτόκολλα διαχείρισης) για την μεταβίβαση διαχειριστικής πληροφορίας στο διαχειριστικό σύστημα που προτείνει ο ίδιος κατασκευαστής. Κάθε ένα από αυτά τα διαχειριστικά συστήματα, τα οποία στην ουσία αποτελούν μια μονάδα επεξεργασίας, ορίζεται από την AT&T σαν Element Management System (ELM).



**Σχήμα 9.1 - Η αρχιτεκτονική UNMA**

Κάθε ELM είναι υπεύθυνο για την διαχείριση μιας ομάδας από διαχειριζόμενα στοιχεία. Προσφέρει "τοπικές" δυνατότητες διαχείρισης - αναγνώριση συναγερμών, επεξεργασία συναγερμών, συλλογή στοιχείων επιδόσεων - και επιπλέον με την σειρά του, μεταδίδει την διαχειριστική πληροφορία σε ένα κεντρικό σύστημα (Integrated Network Management System) με την βοήθεια του πρωτοκόλλου (Network Management Protocol), που καθορίζεται από την UNMA αρχιτεκτονική.

Κάθε ετερογενές δίκτυο περιλαμβάνει μια σειρά από ELMs, από την στιγμή που κάθε κατασκευαστής παρέχει διαφορετικό διαχειριστικό σύστημα για κάθε διαφορετική σειρά προϊόντων. Δεν είναι δηλαδή σπάνιο να βρει κανείς σε ένα δίκτυο διαφορετικό διαχειριστικό σύστημα για τους hosts, διαφορετικό για τα PBXs κ.ο.κ. Σύμφωνα με το UNMA κάθε ELM μπορεί και περνάει διαχειριστική πληροφορία σε ένα κεντρικό διαχειριστικό σύστημα, σύμφωνα με OSI πρωτόκολλα, ώστε να γίνει δυνατή η ενοποιημένη διαχείριση.

Το κεντρικό αυτό σύστημα διαχείρισης δικτύων (βλ. και παρακάτω AT&T Accumaster), παρέχει μια ενοποιημένη πρόσβαση στον διαχειριστή, που του επιτρέπει να ελέγξει το δίκτυο από άκρη σε άκρη (Unified User Interface, UUI), όπως επίσης και μια ευρεία περιοχή εφαρμογών. Το UUI προσφέρει στον διαχειριστή ένα σύνολο από: συντάξεις εντολών, εικονίδια, γραφικά, και διαχείριση παραθύρων, που του επιτρέπουν να έχει πρόσβαση σε κάθε EMS, ή σε εφαρμογές του κεντρικού διαχειριστικού συστήματος. Τέλος, συμπληρώνουμε ότι σύμφωνα με την UNMA αρχιτεκτονική δύο ενοποιημένα συστήματα διαχείρισης μπορούν να επικοινωνήσουν και μεταξύ τους.

Οι εφαρμογές οι οποίες υποστηρίζει η αρχιτεκτονική UNMA είναι οι παρακάτω:

## 1. Configuration and Name Management (Διαχείριση διάρθρωσης και ονομασίας των διαχειριζόμενων αντικειμένων)

Δεν μπορεί να αναπτυχθεί μια ενοποιημένη διαχειριστική εφαρμογή μέχρι να μπορούν να αναγνωρισθούν μοναδικά όλα τα ετερογενή διαχειριζόμενα αντικείμενα. Ένα πλαίσιο αναφοράς για την ονομασία των διαχειριζόμενων αντικειμένων, πρέπει να δίνει ανεξαρτησία σε κάθε εταιρεία να διακανονίζει μόνη της την ονομασία των προϊόντων της. Αυτό μπορεί να επιτευχθεί με μια ιεραρχική δομή στην ονομασία, που περιλαμβάνει το μοναδικό αναγνωριστικό της εταιρείας στην κορυφή της ιεραρχίας, και επιτρέπει στην ίδια την εταιρεία να χειρίστει την ονομασία των προϊόντων της.

Αλλά στην διαχείριση διάρθρωσης, πέρα από την μοναδική αναγνώριση των διαχειριζόμενων αντικειμένων, υπάρχουν και άλλα σοβαρά προβλήματα. Για παράδειγμα πληροφορία που αφορά την τοπολογία του δικτύου και την τοποθέτηση των διαχειριζόμενων αντικειμένων στο δίκτυο, είναι απαραίτητη για την δημιουργία χαρτών για την γραφική παράσταση του δικτύου. Η πληροφορία αυτή μάλιστα θα πρέπει να αντακλά κάθε αλλαγή της τοπολογίας, που οφείλεται σε βλάβη σε ανισοκατανομή του φορτίου, και άλλες αιτίες.

Επίσης είναι απαραίτητη πληροφορία, με ικανότητα αντίστοιχης ένος διαχειριζόμενου αντικειμένου σε κάποιο φιλικό όνομα (alias), στην τοποθεσία του, στην εταιρεία κατασκευής του και άλλα χρήσιμα στοιχεία.

## 2. Fault Management (Διαχείριση βλαβών)

Πολλοί είναι οι διαχειριστές, που δίνουν την υψηλότερη προτεραιότητα σ' αυτήν την εφαρμογή. Η εφαρμογή αυτή συλλεγεί συναγερμούς, τους ελέγχει και τους συσχετίζει, προκειμένου να δείξει κάποιες δυσλειτουργίες στο δίκτυο. Επιπλέον έχει δυνατότητες ελέγχου του δικτύου, διάγνωσης βλαβών, αυτόματης επαναφοράς του δικτύου μετά από βλάβη, όπου βέβαια αυτό είναι δυνατό, και διατήρησης αρχείων με παλαιότερα προβλήματα του δικτύου.

Από τα παραπάνω, πολύ σημαντική δυνατότητα σε ένα ετερογενές περιβάλλον είναι η συσχέτιση των συναγερμών, δεδομένου ότι ο διαχειριστής θα συλλεγεί ένα πλήθος συναγερμών από τα διάφορα ετερογενή διαχειριζόμενα αντικείμενα.

Η δυνατότητα διάγνωσης διαφόρων προβλημάτων σημαίνει ότι ο κεντρικός διαχειριστικός κόμβος και τα EMSs συμφωνούν πάνω στην δομή των μηνυμάτων των συναγερμών (π.χ. επίπεδο προβλήματος, είδος προβλήματος, και κατώφλια). Ο κεντρικός διαχειριστικός κόμβος, μάλιστα οφείλει να κάνει έξυπνη ερμηνεία των παραπάνω παραμέτρων. Όταν μάλιστα έχει αποκτηθεί αρκετή εμπειρία από την επεξεργασία τέτοιων συναγερμών, ο έλεγχος μπορεί να αποδοθεί και σε έμπειρα συστήματα.

### 3. Performance Management (Διαχείριση επιδόσεων)

Η εφαρμογή αυτή ουσιαστικά υποστηρίζει τον τελικό χρήστη, ο οποίος απαιτεί υψηλές απαίτησεις από το δίκτυο. Αυτές οι υψηλές απαίτησεις μπορούν να περιλαμβάνουν κάποιους μικρούς χρόνους απόκρισης ή μεγάλα ποσοστά για την διαθεσιμότητα και την αξιοπιστία του δικτύου.

Η διαχείριση επιδόσεων μετρά διαφορά χαρακτηριστικά του δικτύου, και ειδοποιεί τον διαχειριστή του δικτύου, σε περίπτωση, που κάποιο πρόβλημα διαφαίνεται μέσω των μετρήσεων. Προκειμένου ο κεντρικός διαχειριστικός κόμβος να μπορεί να επιτύχει από άκρη σε άκρη διαχείριση επιδόσεων θα πρέπει τόσο αυτός, όσο και τα EMSs να συμφωνούν στο τι χαρακτηριστικά πρέπει να μετρώνται και βέβαια με τι τεχνικές θα πρέπει να μετρώνται.

Μια αρχική εφαρμογή αρκεί να μετρά τις επιδόσεις σε ολόκληρο το δίκτυο και με κάποια κατώφλια και συναγερμούς να ειδοποιεί τον διαχειρίστη σε περίπτωση προβλημάτων. Σε πιο περίπλοκες υλοποιήσεις είναι δυνατόν μέσα από πολύπλοκους υπολογισμούς με την βοήθεια των μετρούμενων μεγεθών, να επιτευχθεί μια γρήγορη ειδοποίηση για πιθανή μείωση της ποιότητας των προσφερόμενων υπηρεσιών από το δίκτυο.

### 4. Accounting Management (Λογιστική Διαχείριση)

Η εφαρμογή αυτή παρέχει δυνατότητες χρέωσης των υπηρεσιών, που προσφέρει το δίκτυο στους συνδρομητές του. Η χρέωση αυτή μπορεί να περιλαμβάνει ένα σταθερό μέρος για την πρόσβαση στο δίκτυο, στο οποίο προστίθεται και ένα μεταβαλλόμενο μέρος, που εξαρτάται από τις αποστάσεις και από τον χρόνο χρησιμοποίησης του δικτύου. Πιο πολύπλοκοι αλγόριθμοι μπορούν να χρησιμοποιούν μια σταθερή χρέωση μέχρι την υπέρβαση κάποιου ορίου, οπότε οδηγούμαστε σε μεταβλητή χρέωση. Η λογιστική διαχείριση οφείλει να συλλέγει στοιχεία από ετερογενής συσκευές, και κυρίως από αυτές που εξασφαλίζουν την εγκατάσταση και το κλείσιμο μιας σύνδεσης. Στην εφαρμογή αυτή ανήκουν και λειτουργίες ταξινόμησης και δημιουργίας αρχείων με τα μηχανήματα του δικτύου.

### 5. Security Management (Διαχείριση ασφάλειας)

Αυτή είναι μία από τις σημαντικότερες εφαρμογές για την ομαλή λειτουργία του δικτύου. Επειδή πολλά συμφέροντα μπορεί να στηρίζονται πάνω στην ομαλή λειτουργία του δικτύου, μόνο αυστηρά επιλεγμένο προσωπικό πρέπει να μπορεί να αλλάζει την τοπολογία του δικτύου ή να βάζει σε λειτουργία ελέγχους του δικτύου, που εμποδίζουν την ομαλή λειτουργία του. Πολλοί όμως χρήστες θέλουν να έχουν πρόσβαση σε διαχειριστικές πληροφορίες, ώστε να αξιοποιούν καλύτερα το δίκτυο. Η πρόσβαση αυτών των χρηστών μπορεί να επιτευχθεί με κατάλληλα πολυεπίπεδα σχήματα παροχής δικαιωμάτων.

Ο χωρισμός ενός δικτύου σε υποδίκτυα, μερικές φορές ανήκει στην εφαρμογή αυτή. Στην περίπτωση αυτή, πιθανά είναι απαραίτητος διαφορετικός διαχειριστής για κάθε υποδίκτυο.

## 6. Network Planning (Προγραμματισμός)

Η εφαρμογή αυτή χρησιμοποιείται από τον διαχειριστή που θέλει να κάνει "what if" ερωτήσεις στο δίκτυο. Συχνά τα δίκτυα χρειάζονται αλλαγές ή επεκτάσεις για λειτουργικούς ή οργανωτικούς λόγους. Ο χρόνος εκτέλεσης της αλλαγής ή της επέκτασης δεν είναι τόσο καλά καθορισμένος, όπως σε προβλήματα των προγούμενων εφαρμογών. Μπορεί να ανήκει στο χρονικό διάστημα του επόμενου μήνα ή και του επόμενου χρόνου.

Το Network Planning στηρίζεται στις παραπάνω εφαρμογές. Για παράδειγμα η διαχείριση επιδόσεων μπορεί να του προσφέρει μια σειρά από στατιστικές πάνω στις επιδόσεις του δικτύου, όπου και γίνονται φανερές οι ανάγκες για αλλαγές και επεκτάσεις. Από την λογιστική διαχείριση και την διαχείριση διάρθρωσης δέχεται στοιχεία, που αφορούν την διάρθρωση του δικτύου. Στην συνέχεια το Network Planning υπολογίζει αν με την παρούσα διάρθρωση είναι δυνατή η επίτευξη των επιθυμητών επιδόσεων, διαφορετικά προτείνει την οικονομικότερη διάρθρωση για την επίτευξή τους καθώς και την κατάλληλη στιγμή για τις αλλαγές.

### 9.2.2. To Network Management Protocol (NMP)

Στο Σχήμα 9.2 φαίνονται τα ISO/OSI πρωτόκολλα, τα οποία χρησιμοποιεί η AT&T για να δημιουργήσει τη στοίβα πρωτοκόλλων UNMA. Το πιο σημαντικό μεταξύ αυτών είναι βέβαια το CMIP το οποίο είναι πρωτόκολλο για ανταλλαγή διαχειριστικής πληροφορίας και αποτελεί διεθνές πρότυπο. Πέρα από τον καθορισμό αυτών των πρωτοκόλλων η AT&T καθορίζει επίσης τα διαχειριζόμενα αντικείμενα, με τα οποία πρέπει κάθε κατασκευαστής, να περιγράφει κάθε διαχειριζόμενο στοιχείο, προκειμένου να μπορεί το κεντρικό σύστημα να έχει πρόσβαση σ' αυτά.

Η δομή του Network Management Protocol (NMP) είναι σύμφωνη με το γενικό πλαίσιο διαχείρισης κατά OSI (IS 7498/4)[OSIM89] και καθορίζει τα επίπεδα 4 έως και 7 της στοίβας OSI. Είναι δε ανεξάρτητη των πρωτοκόλλων, που θα υλοποιήσουν τα χαμηλότερα επίπεδα. Ένα μειονέχτημα είναι ότι μερικά από τα πρωτόκολλα των επιπέδων 4 έως 7 δεν αποτελούν ακόμα διεθνή πρότυπα, οπότε μπορούν εύκολα να αλλάξουν.

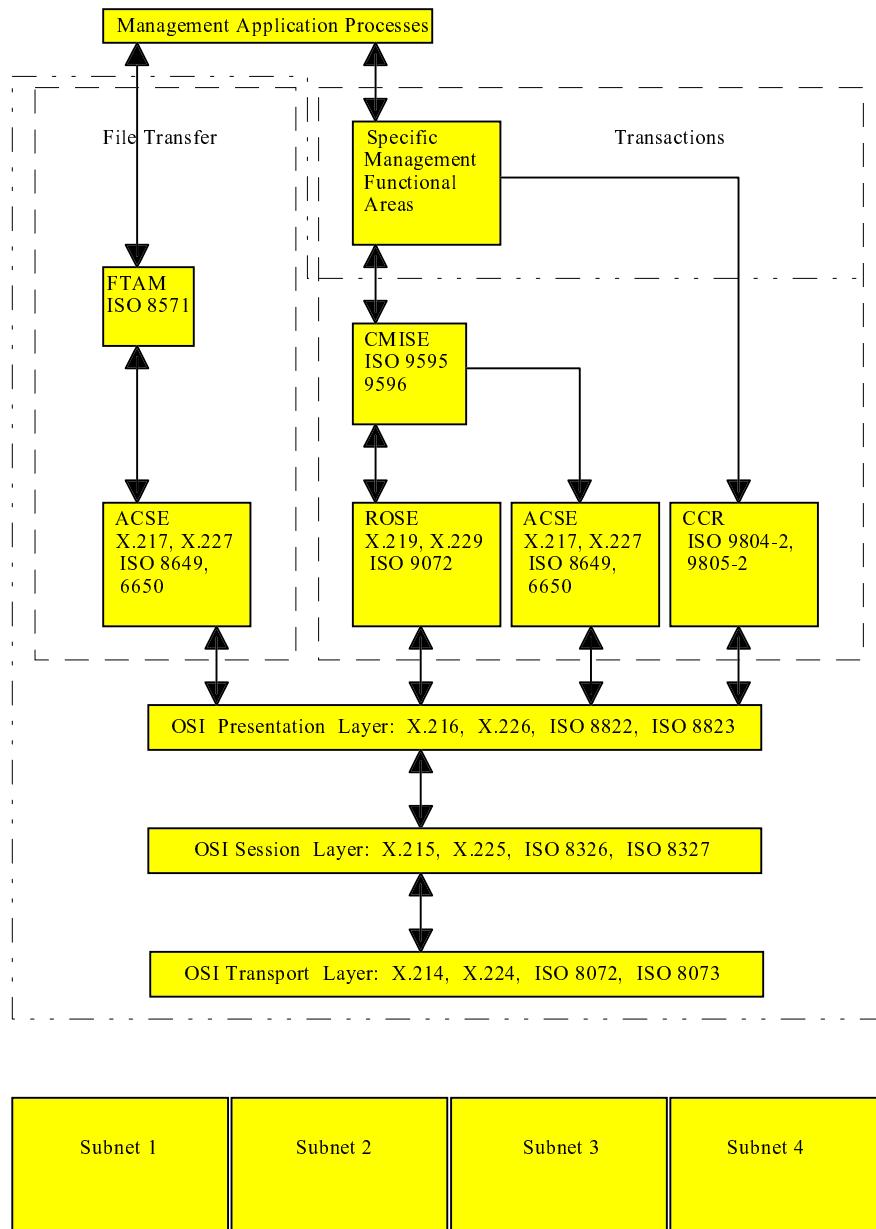
Όπως μπορούμε να δούμε και από το Σχήμα 9.2, η AT&T χωρίζει τις υπηρεσίες του επιπέδου εφαρμογής σε δύο κατηγορίες: transaction services και file transfer services.

Οι υπηρεσίες συναλλαγής στηρίζονται πάνω στα πρωτόκολλα CMIP, ROSE και ACSE, ενώ οι υπηρεσίες μεταφοράς αρχείων στηρίζονται στα πρωτόκολλα ACSE και FTAM. Επιπλέον, η AT&T έχει ορίσει διαχειριστικά μηνύματα για διαχείριση διάρθρωσης (configuration management) και διαχείριση σφαλμάτων (fault management). Οι προδιαγραφές του OSI για αυτές τις δύο λειτουργικές περιοχές είναι πιο κοντά σε προτυποποίηση, απ' ότι για τις υπόλοιπες περιοχές.

Για βαθύτερη ερμηνεία των πρωτοκόλλων μπορεί να ανατρέξει κανείς στα αντίστοιχα κεφάλαια, τα οποία περιγράφουν είτε διαχειριστικά πρωτόκολλα, είτε πρωτόκολλα επιπέδου εφαρμογής, είτε άλλα χαμηλότερα πρωτόκολλα.

### 9.2.3. Προσαρμοστικότητα της αρχιτεκτονικής UNMA

Η UNMA αρχιτεκτονική επιτρέπει στο Integrated Network Management System να μπορεί να χρησιμοποιεί έναν επεξεργαστή, ή να κατανέμει τις ανάγκες επεξεργασίας σε περισσότερους επεξεργαστές. Η αρχιτεκτονική μπορεί να χρησιμοποιηθεί σε κάποιο ιδιωτικό, δημόσιο ή υβριδικό δίκτυο. Επίσης, η αρχιτεκτονική επιτρέπει στα διαχειριζόμενα συστήματα να βρίσκονται είτε σε μια περιορισμένη, είτε σε μια ευρύτερη γεωγραφική περιοχή.



## Σχήμα 9.2 - Η στοίβα πρωτοκόλλων UNMA

Είναι κάτι παραπάνω από βέβαιο, ότι αν οι χρήστες χρησιμοποιήσουν δυνατότητες σαν τις παραπάνω, τότε μπορούν να χτίσουν ένα πολύ αξιόπιστο διαχειριστικό σύστημα.

Μια εφαρμογή των παραπάνω δυνατοτήτων, που δίνει η UNMA αρχιτεκτονική, είναι η τεχνική "N+1 sparing", σύμφωνα με την οποία κάθε μία από N κύριες εφαρμογές χρησιμοποιεί έναν επεξεργαστή, ενώ υπάρχει και ένας επιπλεόν επεξεργαστής για

περιπτώσεις βλαβών. Η τεχνική αυτή είναι μια οικονομική μέθοδος, για την επίτευξη υψηλής αξιοπιστίας σε ένα δίκτυο.

#### **9.2.4. Προϊόντα και υπηρεσίες βασιζόμενα στην UNMA αρχιτεκτονική**

Η AT&T προσφέρει μια σειρά από AT&T Element Management Systems (EMSs) τα οποία θεωρούνται ότι ακολουθούν την UNMA αρχιτεκτονική. Μερικά από αυτά φαίνονται στον Πίνακα 9.1.

Acculink Network Manager
Accunet T1.5 Information Manager, Release 1.1
Analysis 6510, Release 6.1.0
CompuLert (third-party application), Generic 2CMC, Issue 1
Computer Manager, Release 3.4
Comsphere Series 6820 NMS, Releases 1.1 and 1.2
Comsphere Series 6830 NMS, Releases 1.1 and 1.2
DataPhone II Level IV System Controller, Version 5.1
Integrated Access and Cross Connect System
StarGroup Network Manager
StarKeeper NMS, Release 3.0, G4.3
Systems Manager (Developing an Accumaster Integrator Alarm Interface/SNMP Application)
Tridom Clearlink Network Control System, Version 5.0
Trouble Tracker, Release 1, Version 2

#### **Πίνακας 9.1 - Προϊόντα διαχείρισης της AT&T**

Από αυτά εκείνο που αξίζει να εξετάσουμε είναι το Accumaster Integrator. Το σύστημα αυτό είναι ένας "manager of managers", που σχεδιάστηκε έτσι, ώστε να συσχετίζει συναγερμούς και άλλα δεδομένα, που λαμβάνει από διάφορα EMSs (βλ. και παραπάνω), έτσι ώστε να παρουσιάζει μια συνολική εικόνα για κάποια περιοχή του δικτύου.

Δυνατότητες, που προσφέρει η τελευταία έκδοση του Integrator είναι αυτές του Πίνακα 9.2.

Ο Integrator τρέχει σε ένα Sun SPARCstation client και ένα Sun 470 server (Unix System V). Η υποστήριξη του Integrator για την διαχείριση σφαλμάτων (fault management) και για την διαχείριση διάρθρωσης (configuration management) είναι παρόμοια με αυτήν άλλων διαχειριστικών συστημάτων (Sun's SunNet Manager, NCR's NCRNet Manager, HP's OpenView). Επιπλέον, όμως ο Integrator προσφέρει την δυνατότητα συσχέτισης συναγερμών από διαφορετικά EMSs, όπως επίσης την δυνατότητα υποστήριξης μηχανημάτων και υπηρεσιών δικτύων φωνής.

<b>Database partitioning</b>
Automation feature-a scripting feature which allows Accumaster Integrator to perform such tasks as testing and restoring a network element without intervention from the network manager
File import/export-allows database configuration information to be exchanged between Accumaster Integrator and a external database
<b>Trouble ticketing</b>
Audible alarms-with variable tones according to the cause of the alarm
ASCII terminal interface-allowing users to log on from an ASCII terminal

**Πίνακας 9.2 - Δυνατότητες του Accumaster Integrator**

### 9.3. Η αρχιτεκτονική Enterprise Management Architecture (EMA)

Η αρχιτεκτονική Enterprise Management Architecture (EMA) αποτελεί την πρόταση της Digital Equipment Corporation (DEC), για μια διαχείριση ετερογενών δικτύων υπολογιστών, βασισμένη σε πρότυπα. Μέσα από την πρόταση αυτή, η DEC υπόσχεται ενοποιημένη διαχείριση από άκρη σε άκρη, όχι μονάχα δικτύων υπολογιστών αλλά και ολοκληρωμένων κέντρων πληροφοριών με επικοινωνιακά συστήματα φωνής/fax/video, που χρησιμοποιούν διάφορα πρωτόκολλα και προέρχονται από πολλούς κατασκευαστές.

Η αρχιτεκτονική αποτελεί επίσης μια πλατφόρμα για το Digital's Telecommunications Network Management Program (TNMP). Το πρόγραμμα αυτό υποστηρίζει τις συστάσεις της CCITT για το μοντέλο Telecommunication Network Management (TNM model) και αποτελεί μια ανοικτή πλατφόρμα για την διαχείριση σύνθετων δημόσιων δικτύων, που περιλαμβάνουν κινητή τηλεφωνία, δεδομένα, ISDN, Signaling System 7 και άλλα στοιχεία, και η οποία ακολουθεί τα πρότυπα.

Συγκριτικά με την αρχιτεκτονική NetView της IBM, η DEC προσφέρει μια εντελώς διαφορετική φιλοσοφία σχεδίασης. Δηλ. ενώ το NetView είναι μια κεντροποιημένη και αυστηρά ιεραρχική αρχιτεκτονική, η αρχιτεκτονική EMA προσφέρει δυνατότητες και για κατανεμημένες, αλλά και για ιεραρχικές διευθετήσεις των στοιχείων διαχείρισης.

Η αρχιτεκτονική που ακολουθεί το OpenView είναι αντίθετα πιο κοντά σ' αυτή της DEC, αλλά οι πρώτες εκδόσεις της HP χρησιμοποιούσαν ένα flat file για την αποθήκευση των διαχειριστικών δεδομένων, και μόλις τώρα δημιουργούνται εκδόσεις με συστήματα αποθήκευσης βασιζόμενα στην SQL, ενώ η DEC από τις πρώτες εκδόσεις πρόσφερε αντικειμενοστραφή αποθήκη για τις διαχειριστικές πληροφορίες. Επίσης ενώ το OpenView απαιτεί περιβάλλον με παράθυρα, η EMA προσφέρει τρεις διαφορετικούς τρόπους πρόσβασης στο σύστημα διαχείρισης (από τη χρήση του command line μέχρι κάποιο πολύπλοκο GUI).

#### 9.3.1. Περιγραφή της αρχιτεκτονικής EMA

Η αρχιτεκτονική EMA στηρίζεται στο γενικό πλαίσιο, που περιγράφει την OSI διαχείριση. Ορίζει δύο βασικά στοιχεία: τη διαχειριζόμενη οντότητα (managed entity) και τον διευθυντή διαχείρισης (managing director). Τα στοιχεία αυτά είναι ανάλογα με

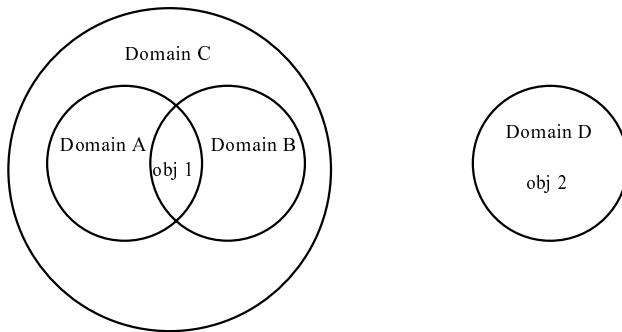
τα τα διαχειριζόμενα αντικείμενα (managed objects) και τα συστήματα διαχείρισης (management systems) του διεθνούς προτύπου IS 7498/4 [OSIM89], που καθορίζει το γενικό πλαίσιο της OSI διαχείρισης. Οι managing directors ανήκουν σε περιοχές (domains), οι οποίες μπορούν να περιλαμβάνουν άλλους managing directors, άλλες περιοχές, ή οποιοδήποτε άλλο δυνατό συνδυασμό. Οι διαχειριζόμενες οντότητες μπορούν να ανήκουν σε περισσότερες από μία περιοχές διαχειρίσης. Εχουν ένα μοναδικό όνομα, το οποίο ακολουθεί το Digital's Distributed Name Service (DECnet phase V). Στο Σχήμα 9.3 βλέπουμε μια πιθανή διευθέτηση διαχειριστικών περιοχών.

Στην συνέχεια θα εξετάσουμε κάθε μία από αυτές τις έννοιες, πιο αναλυτικά.

### 1. Διαχειριζόμενη οντότητα (Managed Entity)

Οι διαχειριστικές οντότητες είναι αντικείμενοστραφής πληροφορία που κυκλοφορεί μεταξύ των διαφόρων διαχειριστικών περιοχών. Παρόμοιες οντότητες ορανώνονται σε κλάσεις. Τα μέλη των κλάσεων αυτών μπορούν να διακριθούν με βάση τα παρακάτω:

- Χαρακτηριστικά, δηλ. ένα κοινό χαρακτηριστικό που το αντικείμενο κατέχει.



### Σχήμα 9.3 - Μια πιθανή διευθέτηση διαχειριστικών περιοχών

- Γεγονότα, για τα οποία το αντικείμενο μπορεί να εκδόσει συναγερμούς ή ειδοποιήσεις. Λειτουργίες, που το αντικείμενο μπορεί να εκτελέσει.
- Μια διαχειριστική οντότητα αποτελείται από δύο μέρη, το αντικείμενο και τον διαχειριστικό αντιπρόσωπο (agent).

### 2. Αντικείμενο

Το αντικείμενο είναι ένα υπαρκτό μηχάνημα, όπως ένα modem, ή μια λογική κατασκευή, όπως μια βάση δεδομένων. Με λίγα λόγια, αντικείμενο είναι, ότι θέλουμε να διαχειριστούμε.

Ένα αντικείμενο μπορεί να προέρχεται από διάφορους κατασκευαστές.

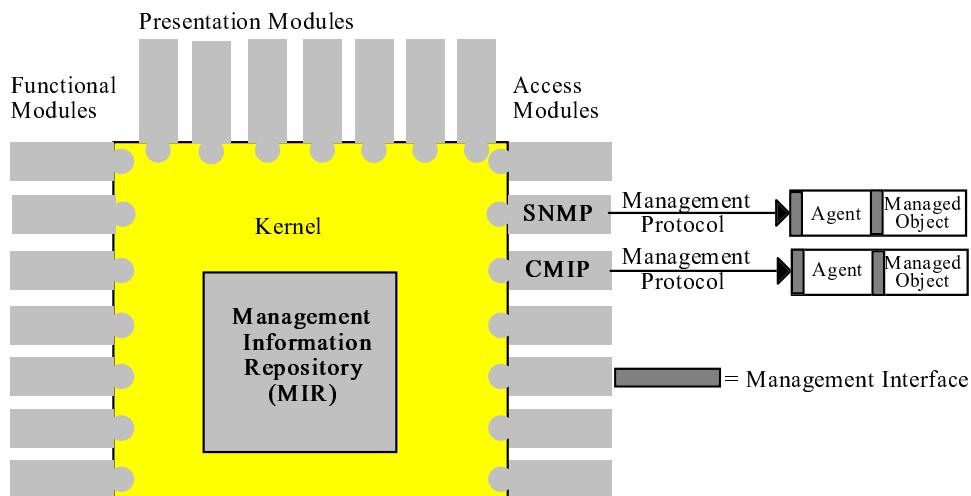
### 3. Διαχειριστικός αντιπρόσωπος (Agent)

Ο agent είναι πάντοτε μια λογική κατασκευή, δηλ. ένα κομψάτι λογισμικό. Αυτός ελέγχει την ροή της πληροφορίας μεταξύ του διαχειριζόμενου αντικειμένου και των διαχειριστών. Ενας agent μεταδίδει εντολές, αποκρίσεις και γεγονότα. Μια εντολή εκδίδεται όταν ο διαχειριστής ζητά κάποια πληροφορία ("Ποια είναι η ταχύτητά σου

μετάδοσης;"). Το αντικείμενο θα απαντήσει στην εντολή αυτή με μια απόκριση ("Η ταχύτητα μου μετάδοσης είναι 19.2Kbps"). Γεγονότα έχουμε, όταν ένα αντικείμενο στέλνει πληροφορία σε έναν διαχειριστή, χωρίς αυτή να είναι απόκριση σε κάποια εντολή ("Ο ένας σύνδεσμος έπεσε"). Τα γεγονότα αυτά είναι στην ουσία συναγερμοί. Το αντικείμενο και ο agent δεν είναι ανάγκη να βρίσκονται στο ίδιο σύστημα. Το μόνο απαραίτητο είναι ο διαχειριστής να έχει πρόσβαση στον agent. Ένας κοινός agent (Common Agent) μπορεί να αναλάβει την μεταφορά διαχειριστικής πληροφορίας μεταξύ πολλών διαχειριζόμενων αντικειμένων και ενός διαχειριστή.

#### 4. Διαχειριστής (Director)

Στο Σχήμα 9.4 βλέπουμε το βασικό μοντέλο, το οποίο περιγράφει την αρχιτεκτονική EMA. Στο ίδιο σχήμα παρατηρούμε ότι ο διαχειριστής αποτελείται από τέσσερα μέρη, τα οποία και θα εξετάσουμε στην συνέχεια.



**Σχήμα 9.4 - Το βασικό μοντέλο της αρχιτεκτονικής EMA**

##### (α) Management Modules

Σε αυτά ανήκουν ομάδες εφαρμογών που εξασφαλίζουν την λειτουργικότητα, την φιλική επικοινωνία με τον χρήστη, καθώς και την πρόσβαση σε διαχειριζόμενα αντικείμενα.

Στην κατηγορία των συναρτήσεων που εξασφαλίζουν την λειτουργικότητα του δικτύου (functional modules), ανήκουν ένα σύνολο από εφαρμογές γενικής φύσης. Για παράδειγμα, όταν ο διαχειριστής αρχικοποιεί την συλλογή ιστορικών στοιχείων, στην ουσία τρέχει το ανάλογο functional module. Κάθε τέτοια συνάρτηση μπορεί να καλεστεί από μια άλλη, ή κατ' ευθείαν από τον διαχειριστή μέσω κατάλληλης επικοινωνίας με το διαχειριστικό σύστημα.

Στην κατηγορία των συναρτήσεων που εξασφαλίζουν την φιλική επικοινωνία με τον χρήστη (presentation modules), ανήκουν λειτουργίες, οι οποίες με κάποιο τρόπο παρουσιάζουν διαχειριστικά στοιχεία στον τελικό χρήστη, ή δέχονται εντολές από

αυτόν. Ο τρόπος φιλικής επικοινωνίας με τον χρήστη μπορεί να επιτευχθεί είτε μέσω κάποιου command line, είτε και μέσω κάποιου Graphical User Interface (GUI).

Τέλος, στην κατηγορία των συναρτήσεων που εξασφαλίζουν την πρόσβαση σε διαχειριζόμενα αντικείμενα (access modules) ανήκουν όλα εκείνα τα σύνολα λειτουργιών, που επιτρέπουν στο διαχειριστικό σύστημα να έχει πρόσβαση στα ετερογενή διαχειριζόμενα αντικείμενα. Παράδειγμα τέτοιου συνόλου λειτουργιών είναι το SNMP, το οποίο επιτρέπει στο διαχειριστικό σύστημα να βλέπει αντικείμενα, τα οποία μπορούν να βρεθούν σε TCP/IP δίκτυα.

#### (β) The Interface

Η εφαρμογή αυτή αποτελείται από το λογισμικό εκείνο, το οποίο επεξεργάζεται την πληροφορία από τους διάφορους αντιπροσώπους και την παρουσιάζει στον διαχειριστή. Το λογισμικό αυτό περιέχει συγκεκριμένες συναρτήσεις με τις οποίες πρέπει να συνδέουν οι χρήστες και οι άλλες κατασκευάστριες εταιρείες τα αντικείμενά τους.

#### (γ) The executive

Η εφαρμογή αυτή εκτελεί όλες τις εντολές του διαχειριστή. Στέλνει όλες τις ερωτήσεις προς τα διαχειριστικά συστήματα, και δέχεται τις αποκρίσεις. Είναι κάτι σαν λειτουργικό σύστημα για το διαχειριστικό σύστημα, παρέχοντας κοινές υπηρεσίες για διαχείριση και ανάπτυξη λογισμικού. Επιπλέον χειρίζεται την επικοινωνία του διαχειριστικού συστήματος με άλλα διαχειριστικά συστήματα.

#### (δ) The Management Information Repository (MIR)

Το MIR είναι μια βάση δεδομένων, η οποία περιλαμβάνει πληροφορία για όλα τα διαχειριζόμενα αντικείμενα, τα οποία υπάρχουν στο δίκτυο. Οι απαραίτητες πληροφορίες για κάθε αντικείμενο περιλαμβάνουν:

- την κλάση του αντικειμένου
- την θέση του αντικειμένου
- τα χαρακτηριστικά του αντικειμένου
- τυχών άλλα ιδιαίτερα χαρακτηριστικά (π.χ. private data)

Το MIR χρησιμοποιεί αντικειμενοστραφής τεχνικές για την αποθήκευση της πληροφορίας. Ο διαχειριστής του δικτύου και οι χρήστες δεν μπορούν να έχουν άμεση πρόσβαση στην βάση δεδομένων αυτή, και μονάχα με άλλα functional modules μπορούν να αντλήσουν στοιχεία.

### 9.3.2. Διαχειριστικά προϊόντα της DEC

Τα προϊόντα, που είναι συμβατά με την αρχιτεκτονική αυτή, πουλιούνται από τη DEC, με το όνομα Polycenter. Παρακάτω θα εξετάσουμε τα σημαντικότερα από αυτά.

#### 1. DECmcc Management Stations

Το πρώτο DECmcc προϊόν εμφανίστηκε το Νοέμβριο του 1989. Τα προϊόντα αυτά, δηλ. τα DECmcc Site Management Station και το DECmcc Enterprise Management Station είναι ουσιαστικά παλαιότερα προϊόντα της DEC με ένα νέο κοινό user-interface. Την ίδια τακτική, όπως θα δούμε παρακάτω ακολούθησε και η IBM με τα NetView προϊόντα. Στην συνέχεια η DEC πρόσθεσε και τον DECmcc Director στα management station products προσανατολιζόμενη πλήρως προς την αρχιτεκτονική EMA.

Τα site management products αποτελούνται από τα DECmcc Basic Management System (BMS), NMCC/VAX ETHERnim, LAN Traffic Monitor, Remote Bridge Management Software, Terminal Server Manager και NMCC/DECnet Monitor. Τα παραπάνω προϊόντα επιτρέπουν την διαχείριση κάθε στοιχείου του δικτύου από έναν μοναδικό σταθμό εργασίας, αλλά από αυτά μόνο το BMS επιτρέπει μια συνολική εικόνα του δικτύου.

## 2. DECmcc Director

Το κύριο πρόγραμμα της DEC είναι το DECmcc Director software. Ο Director παρέχει έναν πυρήνα με το Management Information Repository (MIR), το executive και το interface. Επίσης περιέχει κάποια βασικά presentation, access και functional modules, που χρησιμοποιούνται για τον έλεγχο του περιβάλλοντος του χρήστη. Στο Σχήμα 9.4. φαίνονται καθαρά, όλα τα παραπάνω μέρη, σχετιζόμενα με το μοντέλο, που καθορίζει η αρχιτεκτονική EMA.

Η δομή της EMA παρέχει την βάση για κατανευμένη επεξεργασία. Ο Director έχει πλήρη πρόσβαση σε όλα τα μέρη του δικτύου, και μπορεί να καλέσει εφαρμογές σε όλες τις συνδεδεμένες στο δίκτυο πλατφόρμες. Πληροφορία για όλα τα στοιχεία του δικτύου υπάρχει και στο MIR. Έτσι μέσω του Director μια άλλη εφαρμογή μπορεί να βρει υποαπασχολούμενους κόμβους του δικτύου, και να τους αναθέσει στην συνέχεια κάποια εργασία, ή μέρος κάποιας εργασίας, τα αποτέλεσματα της οποίας, θα αναλάβει να διευθετήσει, και ίσως να επανασυνδέσει η ίδια εφαρμογή μετά το πέρας της εργασίας αυτής.

Τα presentation modules επιτρέπουν πρόσβαση στον Director από το command line, ή από οθόνη γραφικών. Τα access modules περιλαμβάνουν DECnet OSI Phase V, DECnet Phase IV, TCP/IP, Ethernet, bridge management και FDDI. Η DEC σκέφτεται και την προσθήκη access module για terminal servers. Τα functional modules είναι control, registration, domain, topology, alarms, historical data recording, performance analyzer. Σε επόμενη έκδοση προγραμματίζεται και η ύπαρξη fault diagnostic assistant functional module.

Συμπληρώνουμε ότι, είναι σημαντικό να μην μπερδεύει κανείς τον ορισμό του director στην αρχιτεκτονική EMA, με το DECmcc Director που είναι κάποιο συγκεκριμένο προϊόν της DEC.

## 3. TCP/IP SNMP Access Module

Το TCP/IP SNMP Access Module επιτρέπει την επικονωνία μεταξύ ενός TCP/IP δικτύου και ενός EMA-based δικτύου. Χρησιμοποιεί το SNMP για τον έλεγχο των στοιχείων, που τρέχουν τα TCP/IP πρωτόκολλα. Οι τελευταίες μάλιστα εκδόσεις επιτρέπουν το διάβασμα της MIB II, καθώς και το διάβασμα MIBs που αποτελούν προεκτάσεις των κατασκευαστών για την σωστή διαχείριση των TCP/IP προϊόντων τους. Ήδη MIBs από τα μέλη του DECmcc Strategic Vendor Program (Cabletron, Chipcom, Synoptics, Wellfleet) περιλαμβάνονται στα τελευταία προϊόντα.

Ο MIB - μεταφραστής, που χρησιμοποιείται επιτρέπει τον αυτόματο έλεγχο του ASN.1 συντακτικού και την παροχή άμεσων μηνυμάτων σε περιπτώσεις λάθους.

## 9.4. Η αρχιτεκτονική NetView

Μέχρι το 1986, η IBM πουλούσε μια σειρά από ξεχωριστά διαχειριστικά προϊόντα, όπως Network Communications Control Facility (NCCF), Network Problem Determination Application (NPDA), Network Logical Data Manager (NLDM), Network Management Productivity Facility (NMPF), και VTAM Node Control Application (VNCA). Το 1986 η IBM ανακοίνωσε την αρχιτεκτονική SNA Management Series (SNA/MS) και το προϊόν που ακολουθούσε την αρχιτεκτονική αυτή ήταν το NetView, το οποίο μπορεί να θεωρηθεί και σαν μια ολοκλήρωση όλων των παραπάνω προϊόντων.

Το NetView παρέχει μια κεντροποιημένη, ενοποιημένη διαχειριστική ικανότητα για SNA δίκτυα, όπως επίσης και κάποιο τρόπο πρόσβασης, ώστε να επιτευχθεί η διαχείριση των non-SNA προϊόντων.

### 9.4.1. Χαρακτηριστικά του NetView

Το NetView λειτουργεί σαν μια εφαρμογή στον επεξεργαστή VTAM σε υπολογιστές IBM S/370, στους οποίους περιλαμβάνονται και οι σειρές 308X και 43XX. Αφού το θεωρούμε σαν μια εφαρμογή του επεξεργαστή, το NetView είναι κάποιο focal point για τα SNA network management elements. Θυμίζουμε, ότι το focal point είναι ένα από τα τρία είδη στοιχείων δικτύου, που έχουν οριστεί από την IBM. Τα άλλα δύο είναι: τα entry points και τα service points.

Το focal point, όπως φαίνεται και από το όνομα του, παρέχει ένα κεντροποιημένο σύνολο από λειτουργίες διαχείρισης δικτύων, ενώ συγχωνεύει όλα τα δεδομένα που δέχεται από τα στοιχεία του δικτύου, λειτουργώντας σαν μια αποθήκη διαχειριστικών πληροφοριών.

Τα entry points λειτουργούν σαν σημεία ελέγχου, τα οποία παρέχουν στο focal point με διαχειριστικά δεδομένα για τα ίδια, όπως επίσης και για τα στοιχεία του δικτύου, που είναι συνδεδεμένα σ' αυτά. Παράδειγμα ενός entry point είναι το IBM 3174 control unit, το οποίο κάτω από τον έλεγχο του NetView μπορεί να παρέχει πληροφορίες για το ίδιο, καθώς και για τα τερματικά και τους εκτυπωτές που είναι συνδεδεμένα σ' αυτό.

Τα service points είναι σχεδιασμένα, ώστε να παρέχουν ένα παράθυρο επικοινωνίας με non-SNA προϊόντα. Το στοιχείο αυτό δέχεται πληροφορίες από non-SNA μηχανήματα, μεταφράζει τα δεδομένα στην συγκεκριμένη μορφή, που έχουν οι SNA διαχειριστικές πληροφορίες και οι οποίες ονομάζονται Network Management Vector Transports (NMVT) και δρομολογεί τις πληροφορίες αυτές στο focal point του δικτύου. Το NetView/PC, το οποίο είναι κάποιο πακέτο λογισμικού σχεδιασμένο να τρέχει σε IBM και άλλους συμβατούς προσωπικούς υπολογιστές είναι ένα παράδειγμα ενός service point.

Στο Σχήμα 9.5. δείχνουμε την σχέση μεταξύ focal points, entry points και service points. Στο Σχήμα 9.5. επίσης παρουσιάζεται η δυνατότητα, που έχουν άλλες κατασκευάστριες εταιρείες συμβατών με την IBM προϊόντων να υποστηρίζουν άμεσα το πρωτόκολλο του NetView και έτσι τα προϊόντα τους να γίνονται άμεσα entry points στο SNA δίκτυο.

Για non-SNA μηχανήματα, η δυνατότητα για έλεγχο και παρακολούθηση δίνεται μέσω της χρησιμοποίησης ενός ή περισσοτέρων entry points.

#### 9.4.2. NetView Tools

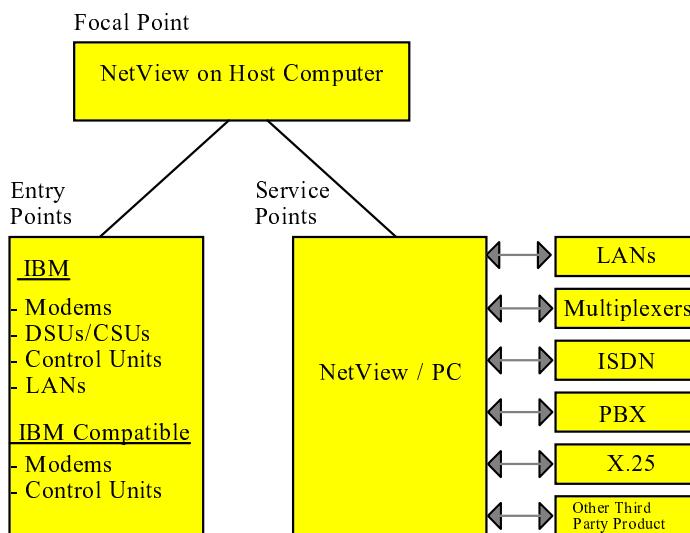
Το NetView αποτελείται από πέντε κύρια εργαλεία, όπως επίσης και από μια σειρά από προαιρετικά εργαλεία. Αυτά είναι στην ουσία τα προηγουμένως ανεξάρτητα προϊόντα της IBM, που τώρα αποτελούν κομμάτια του NetView. Παρουσιάζουμε κύρια και προαιρετικά εργαλεία στον Πίνακα 9.3.

NetView Command Facility
NetView Hardware Monitor
NetView Session Monitor
NetView Help Desk
NetView Status Monitor
NetView/PC
NetView Access Services
NetView File Transfer Program
NetView Distribution Manager
NetView Performance Monitor

**Πίνακας 9.3 - NetView Tools**

Από αυτά αρκετά σύνθετο είναι το NetView Performance Monitor (NPM) και ίσως αξιζει κανείς να ασχοληθεί μαζί του.

Συμπληρώνουμε ότι η αρχιτεκτονική SystemView είναι η τελευταία προσπάθεια της IBM στο χώρο της διαχείρισης.



**Σχήμα 9.5 - Η αρχιτεκτονική OpenView**

#### 9.5. Το περιβάλλον διαχείρισης OpenView

Η Hewlett-Packard (HP) ξεκίνησε το OpenView Project τον Μάρτιο 1988, θέτοντας πολύ σοβαρές βάσεις στον ανταγωνισμό για την δημιουργία ανοικτών διαχειριστικών συστημάτων. Παρακάτω θα εξετάσουμε τα θεμελιώδη χαρακτηριστικά του OpenView.

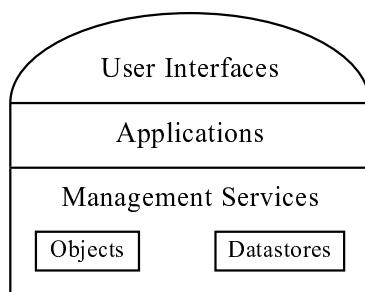
### 9.5.1. Γενικά χαρακτηριστικά

Το OpenView αποτελεί μια λύση για το πρόβλημα της διαχείρισης δικτύων υπολογιστών, και ταυτόχρονα, χρησιμοποιώντας πρωτόκολλα όπως το SNMP, το CMOP και το CMIP αποτελεί ένα σωστό δρόμο στην πορεία προς τα ανοικτά συστήματα. Το OpenView προσφέρει έναν Network Management Server πάνω από τον οποίο μπορεί κανείς να αναπτύξει ιδιαίτερες διαχειριστικές εφαρμογές. Τέτοιες διαχειριστικές εφαρμογές για παράδειγμα, θα μπορούσαν να αποτελέσουν proxy agents για το DECnet και το NetView της IBM.

Με το OpenView η HP ανάπτυξε μια νέα και στην ουσία της διαφορετική προσέγγιση στο πρόβλημα της διαχείρισης. Η καινούργια αρχιτεκτονική σχεδιάστηκε έτσι, σαν μια βάση για μια ανοικτή, κατανεμημένη λύση για το πρόβλημα της διαχείρισης. Η αρχιτεκτονική του OpenView ακολουθεί το γενικό πλαίσιο αναφοράς της OSI διαχείρισης, ενώ ταυτόχρονα έλαβε υπ' όψη και την διαχείριση των TCP/IP δικτύων. Το OpenView παρέχει ένα πλαίσιο για την ανάπτυξη αντικειμενοστραφών εφαρμογών.

Όπως φαίνεται και στο Σχήμα 9.6. το OpenView ακολουθεί μια "building block" αρχιτεκτονική για την διαχείριση χρησιμοποιώντας τρία κυρίως κομμάτια. Αυτά είναι:

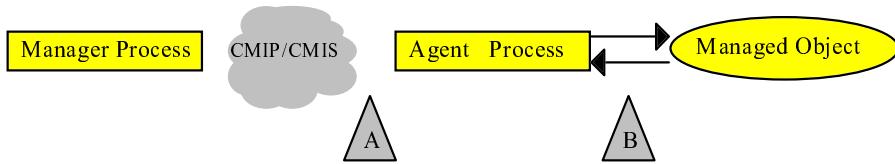
- Management User Interface
- Management Applications
- Management Services



**Σχήμα 9.6 - Λειτουργικό μοντέλο του OpenView**

Τα κομμάτια αυτά είναι διασυνδεδεμένα με πρότυπα application programming interfaces (APIs).

Το OpenView είναι βασισμένο στο OSI Systems Management Model, το οποίο φαίνεται και από το σχήμα 9.9. Οι διαδικασίες του manager και του agent ανταλλάσσουν πληροφορίες χρησιμοποιώντας το πρωτόκολλο Common Management Information Protocol (CMIP). Το μόνο καλά ορισμένο σημείο, όπου προγράμματα και υπηρεσίες συνδέονται είναι το σημείο A, ενώ το σημείο B δεν είναι καλά ορισμένο.



### Σχήμα 9.7 - OSI Systems Management Model

Η OSI Structure of Management Information (SMI), χρησιμοποιείται για να ορίσει αντικείμενα τα οποία διαχειρίζονται στο σημείο A. Το OpenView επιπλέον, επειδή είναι ένα object-oriented περιβάλλον, επιτρέπει την δημιουργία νέων αντικειμένων με τον επαναορισμό ήδη ορισμένων αντικειμένων. Η ιδιότητα της κληρονομικότητας, επιτρέπει στο νέο αντικείμενο να έχει και (καθορισμένα) χαρακτηριστικά του ήδη ορισμένου αντικειμενου. Τα αντικείμενα του OpenView είναι επίσης αλλομορφικά. Για παράδειγμα ένα Hayes modem, μπορεί να αντιμετωπιστεί σαν Hayes modem από κάποια διαχειριστική εφαρμογή, ενώ κάποια άλλη να το αντιμετωπίσει σαν ένα general-purpose modem.

Το πλαίσιο αναφοράς της HP επεκτείνει το OSI πλαίσιο αναφοράς για την διαχείριση στα παρακάτω τέσσερα σημεία:

1. Το OpenView επεκτείνει την object-oriented αντιμετώπιση του προβλήματος και πέρα από τα διαχειρίζόμενα αντικείμενα. Παρέχει μια επεκτάσιμη MIB, ενώ επιτρέπει και οι managers να φαίνονται σαν αντικείμενα σε άλλους managers. Ο συνδυασμός όλων των τεχνικών, επιτρέπει την πιο αποδοτική ανάπτυξη εφαρμογών με object-oriented προγραμματιστικές μεθόδους.
2. Το OpenView επιπλέον προσθέτει δυνατότητες κατανεμημένης επικοινωνίας με environment, supervisor, postmaster services. Οι κατανεμημένες επικοινωνίες παρέχουν ένα πραγματικά κατανεμημένο περιβάλλον εργασίας.
3. Το OpenView διαιρεί την διαδικασία του manager σε ξεχωριστές εφαρμογές και user interfaces, επιτρέποντας τον διαχωρισμό των εφαρμογών και την οδηγών των interfaces.
4. Το πλαίσιο OpenView επίσης παρέχει την δυνατότητα για αποθήκευση διαχειριζόμενων αντικειμένων, δηλ. περιοχές αποθήκευσης ευσταθών MIBs.

#### 9.5.2. Η δομή του OpenView

Η Hewlett-Packard αρχικά δημιούργησε δύο μοντέλα για την επίδειξη του OpenView, ένα organizational model, και ένα operational model.

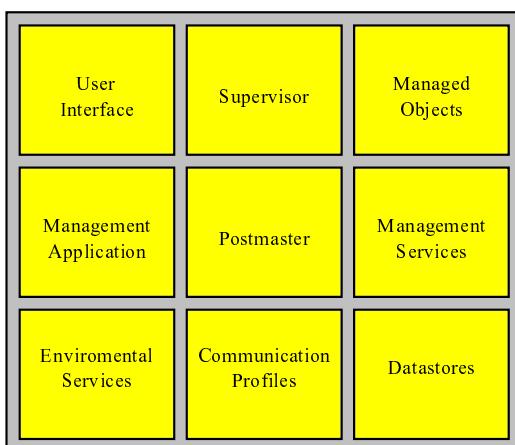
Το organizational model, το οποίο φαίνεται στο Σχήμα 9.6 είναι μια υψηλού επιπέδου αναπαράσταση των βασικών λειτουργιών της διαχείρισης και των σχέσεων μεταξύ τους. Το organizational model αποτελείται από τρία μέρη: user interfaces, management applications και management services. Το τελευταίο μέρος διαιρείται στα: objects και datastores. Το μοντέλο αυτό είναι χρήσιμο για υψηλού επιπέδου σχεδιάσεις διαχειριστικών λύσεων. Οι σχεδιαστές των διαχειριστικών συστημάτων μπορούν να χρησιμοποιήσουν το μοντέλο αυτό, προκειμένου να αντλήσουν την άποψη του χρήστη.

Τέλος, όλο το μοντέλο είναι ενδεικτικό της αντικειμενοστραφούς φιλοσοφίας προγραμματισμού. Τα βασικά κομμάτια του μοντέλου αυτού είναι:

- User Interfaces
- Management Applications
- Management Services
- Objects
- Data stores

Το operational model, που δείχνουμε στο Σχήμα 9.8, χρησιμοποιείται για την εξέταση των αλληλεπιδράσεων μεταξύ των εννέα κυρίων μερών του περιβάλλοντος διαχείρισης OpenView. Αντικείμενα και διαχειριστές μπορούν να εξεταστούν αφηρημένα. Το μοντέλο αυτό παρουσιάζει, πως βλέπει το λειτουργικό σύστημα μια εφαρμογή. Γενικά, κάθε εφαρμογή σπάει σε μικρότερες εργασίες, από τις οποίες μία πραγματοποιείται κάθε φορά. Τα βασικά κομμάτια του μοντέλου αυτού είναι:

- User Interface
- Supervisor
- Objects
- Applications
- Postmaster
- Management Services
- Environmental Services
- Communication Profiles
- Data stores



**Σχήμα 9.8 - Μοντέλο λειτουργίας του OpenView**

### 9.5.3. Κατανεμημένο περιβάλλον διαχείρισης (Distributed Management Environment)

Το κατανεμημένο περιβάλλον διαχείρισης (Distributed Management Environment, DME) ορίζει μια πλατφόρμα για λογισμικό διαχείρισης σε ένα κατανεμημένο ετερογενές περιβάλλον. Το DME είναι ένα σύνολο από κατανεμημένες διαχειριστικές υπηρεσίες, στις οποίες είναι δυνατή η πρόσβαση μέσω καλά ορισμένων APIs, και οι οποίες επιτρέπουν στους χρήστες να διαχειρίζονται τα κατανεμημένα ετερογενή περιβάλλοντά τους.

Το Σεπτέμβριο του 1991, η OSF (Open Software Foundation) αποφάσισε να ενοποιήσει τεχνολογίες από τις Hewlett-Packard, Tivoli Systems, Groupe Bull, IBM κ.ά. προκειμένου να προσφέρει την καλύτερη ολοκληρωμένη λύση για το DME. Οι υπηρεσίες που στηρίζονται σε πρωτόκολλα επιλέχθηκαν από την HP. Οι υπηρεσίες αυτές επιτρέπουν στις διαχειριστικές εφαρμογές να χρησιμοποιούν είτε το SNMP, είτε το CMIP.

## 9.6. Ολοκληρωμένη διαχείριση ετερογενών συστημάτων

Δύο γενικές κατευθύνσεις εμφανίζονται και προδιαγράφονται για το άμεσο μέλλον. Ολοκληρωμένη πολυεπίπεδη διαχείριση με δυνατότητα πρόσβασης σε διαχειριστές υποδικτύων (**integrated enterprise network management**), και η ανάπτυξη διαχειριστικών συστημάτων με ιεραρχική παρέμβαση σε ετερογενή περιβάλλοντα (**multivendor network management**). Πιο συγκεκριμένα:

1. Εμφανίζονται επιτακτικά οι ανάγκες για ολοκλήρωση πολυεπίπεδων λειτουργιών διαχείρισης από το φυσικό επίπεδο έως το επίπεδο εφαρμογών - υπηρεσιών, και παροχή στο χρήστη - διαχειριστή υποδικτύων πληροφορικών συστημάτων (enterprise networks) δυνατοτήτων παρακολούθησης και ελέγχου στο σύστημά του. Οι δυνατότητες αυτές προϋποθέτουν την ύπαρξη overlay networks (δικτύων προωθημένης τεχνολογίας πάνω στα βασικά τηλεπικοινωνιακά δίκτυα κορμού) με παροχή υπηρεσιών αυτο-διαχείρισης σαν προστιθέμενη αξία στον χρήστη (enterprise) ενός νοητού υποσυνόλου (**Virtual Private Network - VPN**). Σαν παράδειγμα, αναφέρουμε το overlay δίκτυο ψηφιακών γραμμών δεδομένων του ΟΤΕ, το Hellascom (βλέπε Ενότητα 6.4) που παρέχει δυνατότητα δημιουργίας VPN σε μεγάλους χρήστες. Η διαχείριση των πολλαπλών VPN πάνω στο Hellascom μοιράζεται στους διαχειριστές, αλλά με αδιαφάνεια ως προς τους λοιπούς χρήστες. Το κεντρικό διαχειριστικό σύστημα του Hellascom έχει τον ρόλο της διαιτησίας και τη τελική ευθύνη λειτουργίας στο φυσικό επίπεδο. Σε ένα γενικότερο επίπεδο, τα overlay networks πρέπει να συνεργάζονται με τα κεντρικά διαχειριστικά συστήματα του Τηλεπικοινωνιακού Φορέα παροχής της γενικής υποδομής (π.χ. το Hellascom, το X.25 Hellaspac το μελλοντικό δίκτυο ISDN, το δίκτυο κοινής σηματοδοσίας CCS7 κλπ. με το Εθνικό Δίκτυο του ΟΤΕ). Με την κατανεμημένη πολυεπίπεδη διαχείριση θα γεφυρωθεί η απόσταση ανάμεσα σε διαχείριση Τηλεπικοινωνιακών Δικτύων (Operation Systems), Δικτύων Υπολογιστών (πχ. SNMP, CMIP/CMISE), Λειτουργικών Συστημάτων κόμβων (Operating Systems) και Εφαρμογών (π.χ. παρέμβαση Κεντρικού Διαχειριστή Τραπεζικού Δικτύου σε επίπεδο θυρίδας Υποκαταστήματος για εντοπισμό λαθών και ανανέωση ή αναβάθμιση λογισμικού εφαρμογών).

2. Το μεγαλύτερο πρόβλημα διαχείρισης δικτύων υπολογιστών προκύπτει από την συνύπαρξη σε κοινό δίκτυο (enterprise network) υπολογιστικών και δικτυακών κόμβων διαφορετικών προμηθευτών (multivendor), με μη συμβατά διαχειριστικά συστήματα. Είναι χαρακτηριστικό ότι η λύση που προωθείται από την ISO/OSI έχει καθυστερήσει (ίσως ανεπανόρθωτα) από γραφειοκρατικές περιπέτειες και αντικρουόμενα συμφέροντα

των προμηθευτών στο διεθνή στοίβο. Σαν αποτέλεσμα, χρήστες και προμηθευτές έχουν προωθήσει τα de-facto standards όπως το TCP/IP και το διαχειριστικό πρωτόκολλο SNMP. Όπως αναφέρθηκε στο Κεφάλαιο 4 και 5, το SNMP έχει πολλούς σοβαρούς περιορισμούς, με κύριες αιχμές την αδυναμία πολυεπίπεδης διαχείρισης και την έλλειψη ασφαλείας (security). Σαν απάντηση, η κοινότητα TCP/IP προχωρεί στις προδιαγραφές SNMPv2 (RFCs1441-1452) με δυνατότητες κρυπτογράφησης μηνυμάτων, ιεραρχική αρχιτεκτονική με εγκατάσταση remote monitoring agents (περιφερειακούς κόμβους που καταχωρούν στα MIB τους στοιχεία για υποδίκτυα - subnetworks), πρόβλεψη βελτιωμένων MIB σε διάφορα στοιχεία δικτύου που δεν διαχειρίζεται το παρόν SNMP κλπ. Γίνεται παράλληλα προσπάθεια σύγκλισης των προτύπων SNMP και CMIP/CMISE στη λογική δικτύων με πρωτόκολλα OSI στα ανώτερα επίπεδα και δίκτυο μεταφοράς TCP/IP (ISODE version 8.0). Η γενική κατεύθυνση είναι να αναπτυχθούν MIB agents με πολλές δυνατότητες και ιεραρχική σχέση προς managers που είναι ταυτόχρονα agents κάτω από manager-of-managers. Η λύση αυτή επιτρέπει (1) τη διαχείριση πολλαπλών υποπεριοχών (domains) με διαφορετικές προδιαγραφές (multivendor), και (2) την απλούστευση των διαχειριστικών λειτουργιών με τον ιεραρχικό διαχωρισμό του δικτύου (συλλογή, επεξεργασία και αποθήκευση μεγάλου και ετερογενούς όγκου πληροφοριών).

Τα δίκτυα υπολογιστών ξεφεύγουν όλο και περισσότερο από την μορφή του κεντρικού ομοιογενούς δικτύου κορμού (single integrated backbone) και προσανατολίζονται προς τη δομή των διασυνδεδεμένων υποδικτύων. Μια τέτοια ιεραρχική δομή επιτρέπει μεγάλη ανάπτυξη στην έκταση ενός τέτοιου δικτύου και αύξηση των συστημάτων που συνδέει. Λόγω της μεγάλης πολυπλοκότητας και των καθυστερήσεων που μπορεί να εισάγει η μεγάλη έκταση, ρεαλιστική διαχείριση μπορεί να επιτευχθεί μόνο με εφαρμογή της φιλοσοφίας "διαίρει και βασίλευε". Έτσι είναι κοινός τόπος η ύπαρξη τοπικών διαχειριστικών συστημάτων στα υποδίκτυα. Επιπλέον, η πληθώρα κατασκευαστών και συστημάτων οδηγεί στην δημιουργία ετερογενών υποδικτύων που χωρίζονται σε αρκετά διαφοροποιημένα τμήματα. Το γεγονός αυτό εισάγει νέους τοπικούς διαχειριστές (εξαρτώμενο από τον κάθε κατασκευαστή) που ευθύνονται για την διαχείριση των μικρών ομογενών τμημάτων μέσα στα υποδίκτυα και πολλές φορές χρησιμοποιούν μητυποποιημένα πρωτόκολλα διαχείρισης ή μη-τυποποιημένες επεκτάσεις τους.

Έτσι η εικόνα των διαχειριστικών συστημάτων στα σύγχρονα δίκτυα παρουσιάζεται κατακερματισμένη και ετερογενής. Η ανάγκη ολοκληρωμένης διαχείρισης (integrated network management) από ένα κεντρικό διαχειριστή είναι προφανής. Η ύπαρξη ενός τέτοιου συστήματος προσφέρει πολλά πλεονεκτήματα όπως :

- Μείωση του κόστους εξοπλισμού και των απαιτήσεων σε χώρο, λιγότερη εκπαίδευση προσωπικού και μειωμένο κόστος συντήρησης. Δεν χρειάζονται πια τοπικές κονσόλες και χειριστές (ή χρειάζονται σε πολύ μικρότερο βαθμό) καθώς η διαχείριση γίνεται με περισσότερο κεντροποιημένο τρόπο ή μέσω μερικών επιπέδων ιεραρχίας με ένα χειριστή σε κάθε κόμβο του δέντρου ιεραρχίας. Για παράδειγμα, ας θεωρήσουμε ένα σύστημα όπου ξεχωριστά, σχετικά με τον κατασκευαστή (vendor specific) συστήματα διαχείρισης αναλαμβάνουν την διαχείριση των διαφορετικών τμημάτων. Έτσι, χρειάζεται μια κονσόλα και ένας χειριστής για το σύστημα διαχείρισης (ΣΔ) του TCP/IP τμήματος του δικτύου, ένας άλλος χειριστής για το ΣΔ των modems, ένας άλλος για το ΣΔ των μεταγωγέων στο τμήμα όπου χρησιμοποιείται μεταγωγή πακέτων κ.ο.κ. Αν ένα ολοκληρωμένο σύστημα (σχήμα 9.9) ενοποιήσει όλα αυτά τα συστήματα ένας μόνο χειριστής και μόνο μια κονσόλα (αυτή του ολοκληρωμένου συστήματος) απαιτούνται για ολοκληρωμένη διαχείριση όλων των πόρων του δικτύου.
- Παροχή ενός κοινού διαχειριστικού περιβάλλοντος διαχείρισης. Μέσα σε αυτό το περιβάλλον τα στοιχεία των ετερογενών τμημάτων και των διαφορετικών επιπέδων ιεραρχίας του δικτύου ακολουθούν μια ενοποιημένη αναπαράσταση (μπορεί να εφαρμοστούν αντικειμενοστραφής τεχνικές ή όχι) και η ανάπτυξη και η συντήρηση

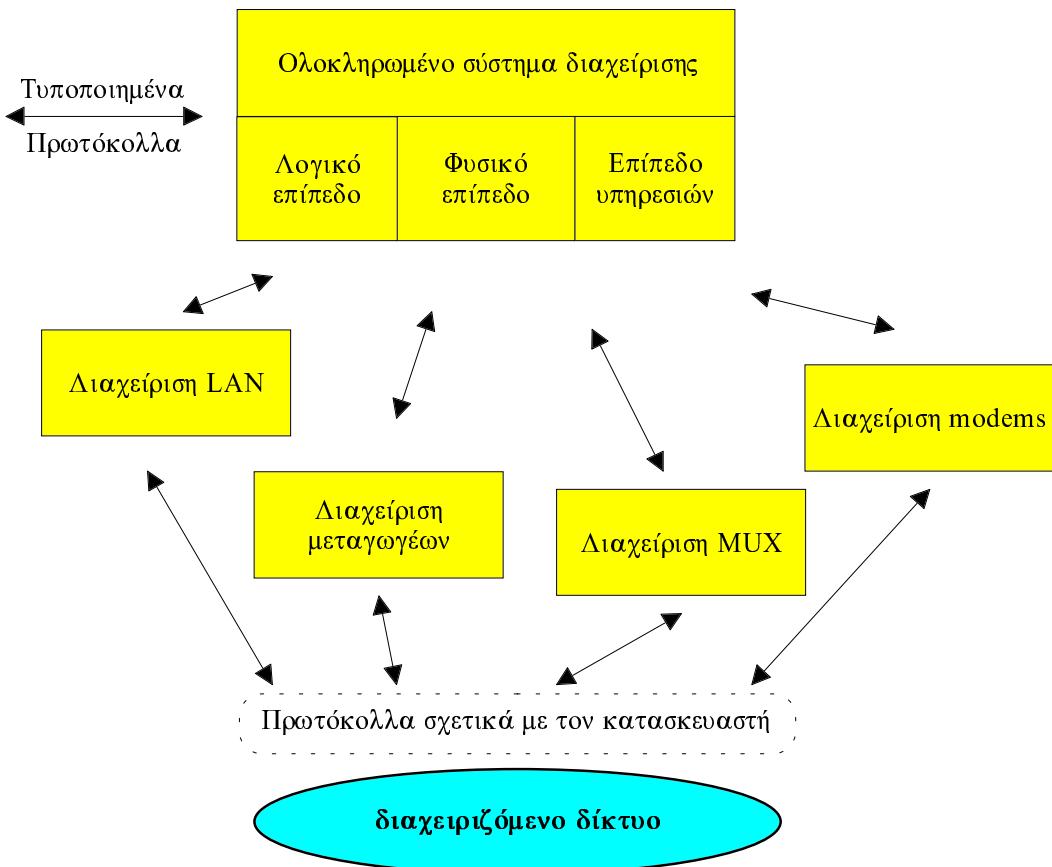
ολοκληρωμένων εφαρμογών γίνεται σημαντικά ευκολότερη. Επιπλέον, η χρήση προτύπων όπως τυποποιημένων πρωτοκόλλων διαχείρισης έιναι δυνατή, κάτι που κάνει ευκολότερη την επέκταση και προσαρμογή του συστήματος. Για παράδειγμα, όπως φαίνεται στο σχήμα 9.9, ο ολοκληρωμένος διαχειριστής μπορεί να προσφέρει τη δυνατότητα επικοινωνίας με τυποποιημένο πρωτόκολλο (SNMP, CMIP κτλ). Κάτι τέτοιο επιτρέπει την οργάνωση τέτοιων ολοκληρωμένων διαχειριστών σε ιεραρχικές ή δικτυωμένες αρχιτεκτονικές που προσφέρουν σημαντικά πλεονεκτήματα σε multi-domain δίκτυα και σε περιπτώσεις όπου απομόνωση τμημάτων επιβάλλεται (βλέπε ενότητα 8.2).

- Δυνατότητα ολοκληρωμένων λειτουργιών. Πολλές λειτουργίες σε ένα δίκτυο επιδρούν σε πολλά επίπεδα (φυσικό επίπεδο, λογικό επίπεδο, κτλ) ή σε παραπάνω του ενός στοιχεία του δικτύου που πιθανόν να διαχειρίζονται από διαφορετικά τοπικά διαχειριστικά συστήματα. Για παράδειγμα η παροχή και η συντήρηση μιας μισθωμένης γραμμής σε ένα τηλεπικοινωνιακό δίκτυο απαιτεί αλλαγές σε ένα σύνολο συστημάτων λογικού και υλικού, όπως αλλαγές στην δρομολόγηση, στα συστήματα μετάδοσης και μεταγωγής, σε βάσεις δεδομένων, σε στοιχεία λειτουργικού συστήματος κτλ. Ολόκληρη αυτή η διαδικασία μπορεί να επιταχυνθεί και να απλοποιηθεί μέσω ενός ολοκληρωμένου συστήματος διαχείρισης, μειώνοντας έτσι την ανθρώπινη προσπάθεια, την πιθανότητα λάθους, το κόστος και ταυτόχρονα αυξάνοντας την ποιότητα και το χρόνο απόκρισης στην παροχή υπηρεσίας.
- Δυνατότητα ανάπτυξης ολοκληρωμένων ευφυών εφαρμογών που επεκτείνονται σε πολλά επίπεδα. Η ικανότητα ενός ολοκληρωμένου ΔΣ να συγκεντρώνει διαχειριστικές πληροφορίες από διαφορετικά ετερογενή στοιχεία του δικτύου και να τα χειρίζεται με ενοποιημένο τρόπο, ανοίγει τον δρόμο προς την ανάπτυξη τέτοιων εφαρμογών. Έτσι για παράδειγμα, μπορούν να αναπτυχθούν εφαρμογές ολοκληρωμένης διαχείρισης σφαλμάτων ή απόδοσης που αντλούν πληροφορίες διάρθρωσης, συναγερμούς, και πληροφορίες κατάστασης από ολόκληρο το δίκτυο. Πληροφορίες από διαφορετικά επίπεδα επιτρέπουν σε τέτοιες εφαρμογές καλύτερη ανάλυση των σφαλμάτων, διάγνωση των αιτιών και πρόταση λύσεων.
- Προστασία των επενδύσεων σε ένα δίκτυο. Συστήματα που βασίζονται σε πρότυπα και τυποποιήσεις προσφέρουν ευελιξία στην διαδικασία πρόσθεσης ή αντικατάστασης συσκευών χωρίς η επίδραση στις διαχειριστικές εφαρμογές να είναι σημαντική.

Από τις αρχιτεκτονικές που αναφέρθηκαν στο κεφάλαιο 8, η ιδανικότερη αλλά και η λιγότερο ρεαλιστική είναι η κεντροποιημένη αρχιτεκτονική (και η παραλλαγή της πλατφόρμας διαχείρισης). Είναι ιδανική γιατί περιορίζει στο ελάχιστο τα διαχειριστικά συστήματα που χρησιμοποιούνται και η άντληση πληροφοριών είναι άμεση. Ο κεντρικός ολοκληρωμένος διαχειριστής επικοινωνεί απευθείας με τα ετερογενή στοιχεία του δικτύου και όχι μέσω των τοπικών διαχειριστών, ή μιας ιεραρχίας τέτοιων συστημάτων. Επιπλέον ο έλεγχος είναι καλύτερος, καθώς ο κεντρικός διαχειριστής δίνει εντολές στα στοιχεία και όχι στους τοπικούς διαχειριστές. Το αποτέλεσμα της στρατηγικής αυτής είναι μειώση στο ελάχιστο των απαιτουμένων συστημάτων διαχείρισης και αύξηση της απόδοσης λόγω της αμεσότητάς της. Το σημαντικότερο πρόβλημα είναι ότι τα σχετικά με τους κατασκευαστές συστήματα διαχείρισης είναι κλειστά συστήματα που εισάγουν ανυπέρβλητα εμπόδια στην διαχείριση των αντίστοιχων στοιχείων μέσω ενός ενοποιημένου διαχειριστή.

Η κατανεμημένη, η ιεραρχική, καθώς και η παραλλαγή τους, δικτυωμένη αρχιτεκτονική αποτελούν πιο ρεαλιστικές λύσεις για το πρόβλημα της ενοποιημένης διαχείρισης. Άλλωστε, οι αρχιτεκτονικές αυτές πρωτοεμφανίστηκαν ως προσπάθειες ολοκληρωμένης διαχείρισης. Η επιλογή μιας από αυτές εξαρτάται από τις

συγκεκριμένες απαιτήσεις του ΣΔΔ, τον τύπο και την μορφή του διαχειριζόμενου δικτύου.



**Σχήμα 9.9 - Ολοκληρωμένο ΣΔΔ**

## 9.7. Ασκήσεις

- [1]. Συγκρίνατε τα συστήματα ενοποιημένης διαχείρισης UNMA και OpenView. Σε τι είδους δικτύου θα χρησιμοποιούσατε το καθένα από αυτά;
- [2]. Αναφέρατε χαρακτηριστικά τα οποία θα θέλατε να προσφέρει το διαχειριστικό σύστημα το οποίο θα χρησιμοποιούσατε. Ποια θεωρείτε πολύ σημαντικά από αυτά; Πιστεύετε ότι ένα διαχειριστικό σύστημα με όλα τα παραπάνω χαρακτηριστικά θα ήταν αρκετό για τη διαχείριση ενός αριθμού από διασυνδεδεμένα τοπικά δίκτυα; Μήπως θα ήταν απαραίτητο επιπλέον υλικό ή λογισμικό;
- [3]. Βρείτε στοιχεία (π.χ. από περιοδικά, αντιπροσώπους, κ.τ.λ.) για κάποιο σύστημα ενοποιημένης διαχείρισης δικτύων (π.χ. από τη SUN Microsystems Inc.) το οποίο δεν αναφέρετε στις σημειώσεις. Φτιάξτε μια έκθεση αξιολόγησης του συγκεκριμένου πακέτου.

## 9.8. Βιβλιογραφία

- [BRIN88] Brinsfield J.G., Unified Network Management Architecture (UNMA), AT&T Laboratories, 1988.
- [CHAO90] C.W. Chao, P. Sarachik, B. Maglaris, R. Boorstyn, and D. Dimitrijevic, "Control of Multi-Domain Networks", Network Management and Control, Edited by A. Kershbaum *et al.*, Plenum Press, New York, 1990.
- [DIMI89] D.D. Dimitrijevic, B. Maglaris, R.R. Boorstyn, "Routing in Multiple Domain Networks", Proceedings of the IEEE INFOCOM-89, Ottawa, Canada, April 1989.
- [EMA\_92] Digital Equipment Corp. Enterprise Management Architecture, DATAPRO, Datapro International Network Management, Standards, Protocols and Architectures, January 1992.
- [HEGE93] Symposium on Integrated Network Management. Hegering, Heinz-Gerd, editor. Yemini, Yechiam, editor. 03/1993. Elsevier Science Publishing Company, Incorporated.
- [HELD92] Gilbert Held, Network Management, Techniques, Tools and Systems, John Wiley & Sons, England, 1992.
- [HERM] James Herman, "MultiVendor Network Management: Part I : Theory, Part II : Products and Architectures".
- [HERM90] James Herman, "Enterprise Management Vendors Shoot It Out", Data Communications International, November 1990.
- [HPOV92] Hewlett-Packard OpenView, DATAPRO, Datapro International Network Management, Standards, Protocols and Architectures, January 1992.
- [IBM\_92] IBM System View Management Architecture, DATAPRO, Datapro International Network Management, Standards, Protocols and Architectures, January 1992.
- [KAUF91] Kauffels, Franz-Joachim. *Network Management: Problems, Standards, Strategies*. 1991. Addison-Wesley Publishing Company, Incorporated.
- [KRIS91] Krishman, I. & Zimmer, W. *Integrated Network Management, No. II*. 1991. Elsevier Science Publishing Company, Incorporated.
- [KRIS93] Krishnan, Iyengar. *Integrated Network Management*. 1993. McGraw-Hill, Incorporated.
- [OPEN92] HP OpenView, Technical Evaluation Guide, HEWLETT PACKARD, Release 3.0, USA, 1992.
- [OSIM89] Information processing systems - Open System Interconnection - *OSI Management Framework* - International Organization for Standardization - International Standard 7498/4 - April 1989.

- [RABI92] Sameh Rabie. *Integrated Network Management : Technologies and Implementation Experience*, Infocom '92, IEEE, 1992.
- [UNMA92] AT&T Unified Network Management Architecture (UNMA) and Accumaster, DATAPRO, Datapro International Network Management, Standards, Protocols and Architectures, January 1992.

## Κεφάλαιο 10

### 10. Δίκτυο Διαχείρισης Τηλεπικοινωνιών - Telecommunications Management Network (TMN)

#### Περιεχόμενα του Κεφαλαίου 10

- 10.0. Εισαγωγή
- 10.1. Τα TMN πρότυπα στο Κοινοτικό Πρόγραμμα NETMAN (RACE R1024)
- 10.2. Λειτουργικό μοντέλο του TMN
- 10.3. Σημεία αναφοράς (Reference Points)
- 10.4. Φυσική Αρχιτεκτονική του TMN και σημεία διασύνδεσης
- 10.5. Λειτουργική δομή ενός Operations System
- 10.6. Βάση Πληροφορίας Διαχείρισης
- 10.7. Φυσική Αρχιτεκτονική του TMN
- 10.8. Βιβλιογραφία

#### 10.0. Εισαγωγή

Το πρότυπο αναφοράς TMN (Telecommunications Management Network) ορίστηκε για νέα δίκτυα ευρείας ζώνης (Integrated Broadband Communications - IBC ή Broadband ISDN - BISDN) βασισμένα στις τεχνικές μεταγωγής Asynchronous Transfer Mode (ATM). Είναι αξιοσημείωτο πως η προδιαγραφή ATM (cell relay) προχωρεί ταυτόχρονα σε τρία επίπεδα:

1. **Το επίπεδο μεταφοράς και μεταγωγής δεδομένων.** Βασίζεται στη χρήση σύγχρονης οπτικής ζεύξης (Synchronous Digital Hierarchy - SDH) και μεταγωγή μικρών πακέτων 53 bytes (cells) σε ταχύτητες  $\geq 155$  Mbits/sec ανά γραμμή εισόδου/εξόδου σε μεταγωγείς ATM.
2. **Το επίπεδο σηματοδοσίας (signalling).** Βασίζεται στη δημιουργία και συντήρηση νοητών κυκλωμάτων (virtual circuits, virtual paths) ανά ζεύγος τελικών χρηστών. Η διακίνηση και επεξεργασία των αναγκαίων σημάτων δεν γίνεται με την κλασική μέθοδο της ψηφιακής τηλεφωνίας Common Channel Signalling 7 (CCS 7) αλλά προτείνονται μοντέλα object oriented [MINZ89], [MINZ91].

---

\* Σε αντίθεση με το μονολιθικό μοντέλο κλήσης που ορίζεται από το CCS.7 της CCITT (ITU-T), προτείνονται μοντέλα που αποσυνθέτουν την κλήση σε στοιχειώδη αντικείμενα (Call Elementary Objects), όπως σύνδεση, υπηρεσία (φωνή, video), τρόπο πρόσβασης στο δίκτυο, άκρα σύνδεσης, κ.ά. Οι χρήστες έχουν τη δυνατότητα να ορίζουν και να αλλάζουν τα ανωτέρω

3. **Το επίπεδο διαχείρισης.** Προβλέπει τη διακίνηση μηνυμάτων διαχείρισης μέσα από νοητό ή πραγματικό **Δίκτυο Διαχείρισης Τηλεπικοινωνιών (Telecommunications Management Network - TMN)**. Οι λειτουργίες διαχείρισης αφορούν ζητήματα παρακολούθησης και ελέγχου του τηλεπικοινωνιακού συστήματος από εξωτερικό διαχειριστικό σύστημα, με στόχους π.χ. τον βραχυπρόθεσμο και μακροπρόθεσμο σχεδιασμό (*planning, design & installation*), τη διαχείριση πόρων του δικτύου (*provisioning*), την συντήρηση του δικτύου και τον εντοπισμό και διόρθωση βλαβών (*maintenance*), την μακροσκοπική παρακολούθηση επιδόσεων (*performance management*), την λογιστική παρακολούθηση (*accounting management*), τον έλεγχο μηχανισμών ασφαλείας (*security management*) και την εξυπηρέτηση πελατών με πρόσβαση σε υπολειτουργίες του TMN (*customer query & control*). Οι λειτουργίες του TMN αποτελούν ολοκλήρωση των κλασσικών λειτουργιών διαχείρισης ενός Τηλεπικοινωνιακού Φορέα που κωδικοποιούνται με τα αρχικά **OA&M (Operations, Administration & Maintenance)**. Διαφέρουν από τις λειτουργίες σηματοδοσίας οι οποίες αφορούν τη διαχείριση κλήσεων συνδρομητών σε πραγματικό χρόνο.

Με τις προδιαγραφές TMN είναι ίσως η πρώτη φορά που οι Τηλεπικοινωνιακοί Οργανισμοί συνδέουν την συμβατότητα **μεταφοράς δεδομένων** και **σηματοδοσίας** με προδιαγραφές **διαχείρισης**, ενοποιημένες σε κοινό **τυποποιημένο ανοικτό πλαίσιο**. Τα πρότυπα TMN μπορεί να θεωρηθούν σαν ένα πλαίσιο αναφοράς, που θα εντάξουν τα πρωτόκολλα διαχείρισης OSI (CMIP/CMIS) σαν μέσο μεταφοράς και επεξεργασίας της πληροφορίας διαχείρισης. Αντίθετα, τα πρωτόκολλα διαχείρισης SNMP, προερχόμενα από το χώρο της Πληροφορικής, δεν έχουν ακόμα ενταχθεί στην πρόβληματική των Τηλεπικοινωνιακών Φορέων όπως εκφράζονται στις Επιτροπές της CCITT (ITU-T), στα Κοινοτικά Ευρωπαϊκά Προγράμματα (RACE) και στις ΗΠΑ (Bellcore, Regional Bell Operating Companies).

## 10.1. Τα TMN πρότυπα στο Κοινοτικό Πρόγραμμα NETMAN RACE (R1024) [SMIT93]

Στο Κοινοτικό Πρόγραμμα RACE και τις αντίστοιχες Επιτροπές της CCITT έχει προκύψει μια συστηματική προσέγγιση στη διαχείριση τηλεπικοινωνιακών δικτύων, με μικρότερα λειτουργικά έξοδα. Η ιδέα του TMN (Telecommunications Management Network) παρέχει ένα πλαίσιο, κάτω από το οποίο μπορεί να δοθεί μια καλύτερη λύση στο πρόβλημα της διαχείρισης των τηλεπικοινωνιακών δικτύων. Έχει αναπτύχθει παράλληλα με τις επίσημες ή ad hoc τυποποιήσεις διαχείρισης δικτύων **τηλεπληροφορικής** (CMIP/SNMP) και τείνει να υιοθετηθεί από τους Δημόσιους Τηλεπικοινωνιακούς Οργανισμούς (Public Network Operators - PNOs). Το TMN αποτελεί ένα υπερσύνολο των πρωτοκόλλων OSI (CMIP) με στόχο τη διαχείριση μεγαλύτερων και πολυπλοκότερων δικτύων τηλεπικοινωνιακών υποδομών μέσω του TMN.

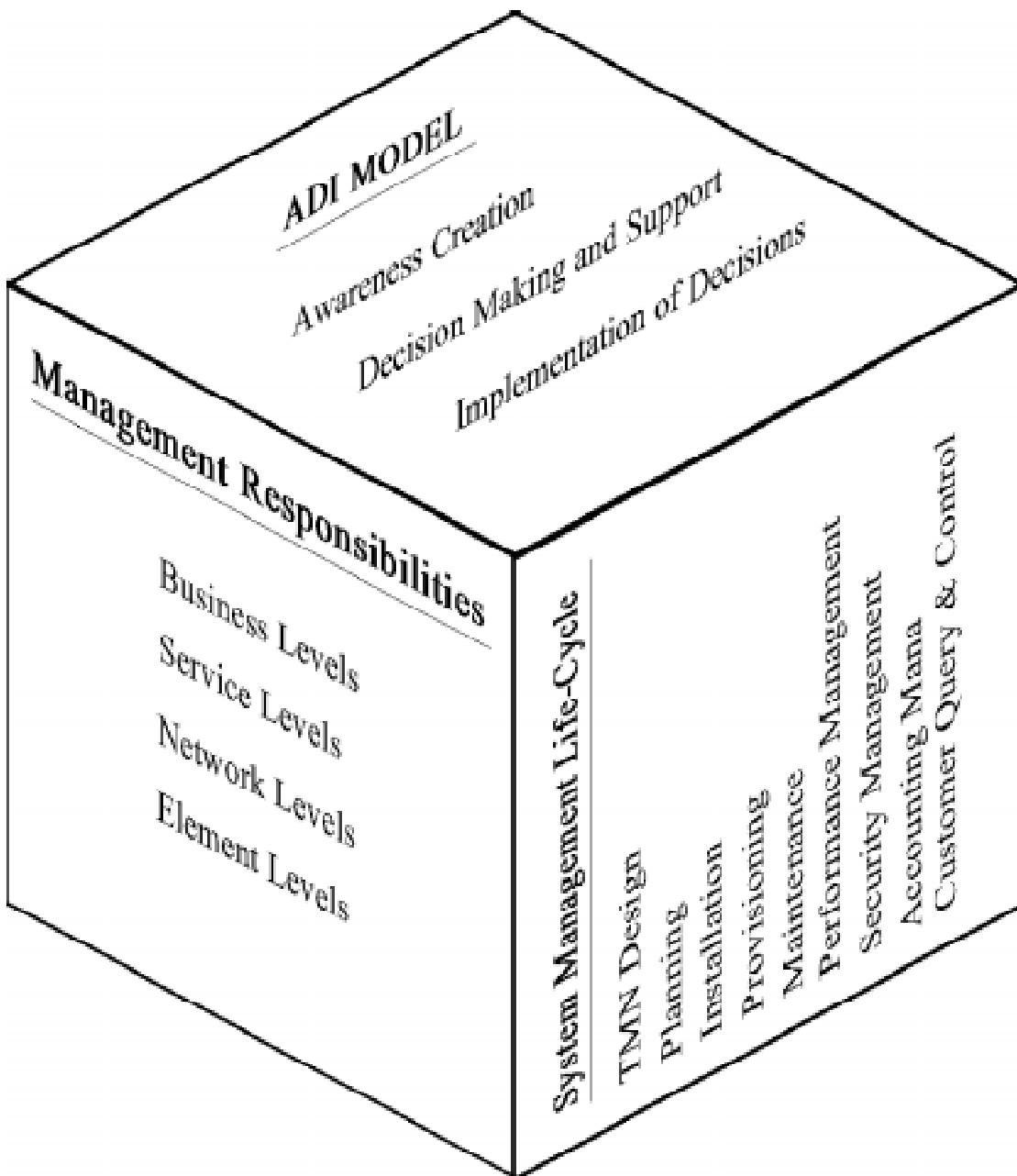
Το TMN είναι ένα **λογικό δίκτυο** το οποίο έχει προσβάσεις στο τηλεπικοινωνιακό δίκτυο, προκειμένου να αντλεί και να στέλνει πληροφορίες από και προς αυτό και ταυτόχρονα να το ελέγχει. Μπορεί επίσης να χρησιμοποιεί κομμάτια του τηλεπικοινωνιακού δικτύου, προκειμένου να εξασφαλίζει την μεταφορά της πληροφορίας, που επιθυμεί. Για παράδειγμα, πληροφορία TMN μπορεί να υποστηριχθεί

---

αντικείμενα στη διάρκεια της κλήσης (π.χ. πρόσθεση νέου συνομιλητή, αλλαγή ποιότητας μετάδοσης πληροφορίας) με την ανταλλαγή προγραμμάτων transactions με το δίκτυο και τους άλλους συνδιαλεγόμενους συνδρομητές.

από το Embedded Control Channel (ECC) σε Synchronous Digital Hierarchy(SDH) οπτικά δίκτυα.

Κλειδί στην προσέγγιση του TMN είναι τα διάφορα μοντέλα που αναπτύχθηκαν κάτω από διάφορους βαθμούς αφαίρεσης και τα οποία μπορούν να βοηθήσουν τόσο τους σχεδιαστές, όσο και τους μελλοντικούς χρήστες να αντιληφθούν το εύρος και τη φύση των μελλοντικών συστημάτων διαχείρισης τηλεπικοινωνιακών δικτύων ευρείας ζώνης.

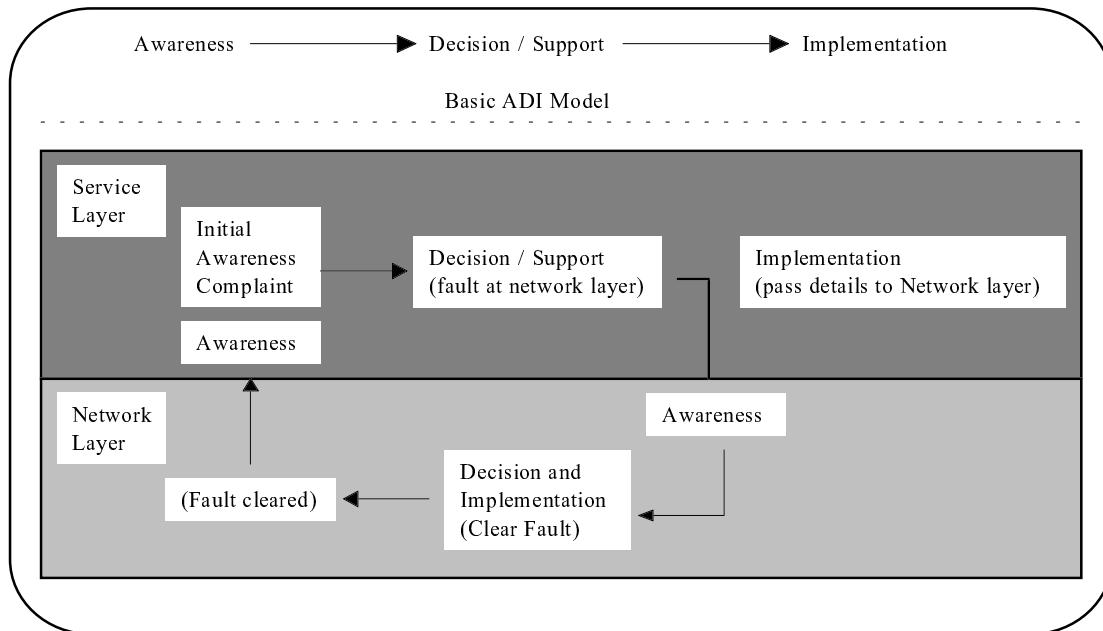


**Σχήμα 10.1 - Μοντέλο κύβου NETMAN**

Στα πλαίσια του προγράμματος RACE NETMAN (R1024) αναπτύχθηκε το μοντέλο του κύβου (Σχήμα 10.1), το οποίο παρουσιάζει τρεις συνδυασμένες πλευρές της διαχείρισης, οι οποίες δομούν τις απαιτήσεις του μοντέλου αναφοράς λειτουργικότητας του TMN. Οι πλευρές αυτές είναι : το μοντέλο **ADI**, το μοντέλο **Management Responsibilities**, καθώς και η πλευρά κύκλου ζώνης της διαχείρισης συστημάτων - **System Management Life-Cycle**.

Η πλευρά ***ADI (Awareness Creation (A), Decision Making and Support (D) και Decision Implementation (I))***. Καθορίζει ότι κάθε διαχειριστικό σύστημα έχει να εκτελέσει ένα σύνολο από Α λειτουργίες, Δ λειτουργίες και Ι λειτουργίες. Η επεξεργασία μεταφέρεται λογικά από την μία λειτουργία στην άλλη σαν μια σειρά από γεγονότα (από το Α στο Δ στο Ι). Ενώ, πολλές τέτοιες λειτουργίες μπορεί να εκτελούνται παράλληλα στο σύστημα.

Ένα παράδειγμα εφαρμογής του μοντέλου ADI φαίνεται στο σχήμα 10.2.

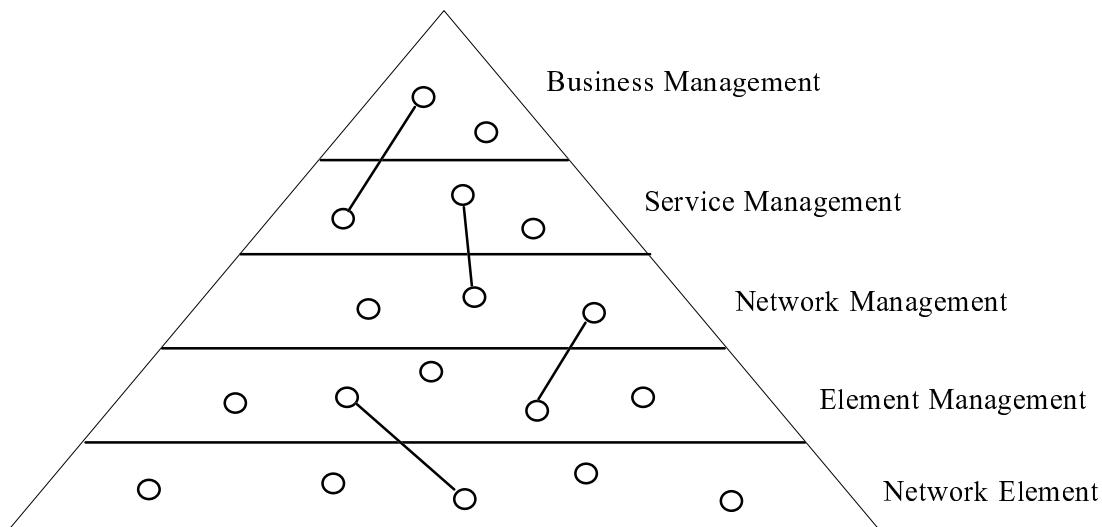


**Σχήμα 10.2 - Εφαρμογή του μοντέλου ADI**

Το ***Management Responsibilities model*** είναι ένα πολυεπίπεδο μοντέλο και απεικονίζει γενικά παραδεκτές αρχές της συμπεριφοράς των συστημάτων και της θεωρίας διαχείρισης. Τα επίπεδα στο Responsibility model είναι τα εξής (βλ. και Σχήμα 10.3):

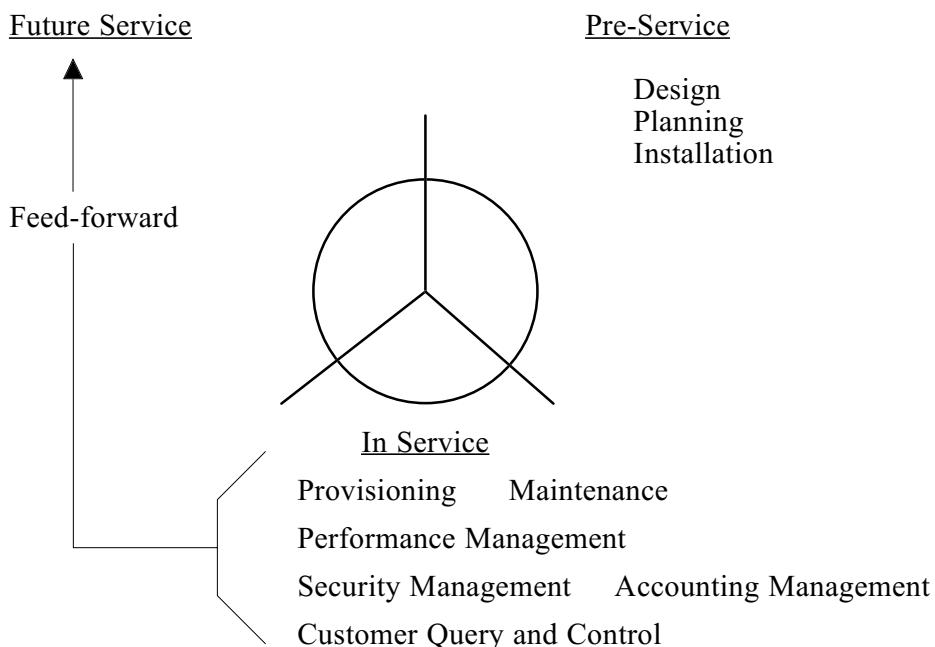
- **Business management layer (Επίπεδο Διαχείρισης Επιχείρησης):**  
Αφορά στρατηγικές αποφάσεις στα πλαίσια μιας επιχείρησης
- **Service management layer (Επίπεδο Διαχείρισης Υπηρεσιών):**  
Αφορά συμφωνητικά παροχής τηλεπικοινωνιακών υπηρεσιών
- **Network management layer (Επίπεδο Διαχείρισης Δικτύου):**  
Αφορά συνολική παρακολούθηση και έλεγχο του τηλεπικοινωνιακού δικτύου

- **Subnetwork management layer (Επίπεδο Διαχείρισης Υποδικτύου):**  
Αφορά παρακολούθηση και έλεγχο τμημάτων του τηλεπικοινωνιακού δικτύου
- **Element management layer (Επίπεδο Διαχείρισης Στοιχείων Δικτύου):**  
Αφορά παρακολούθηση και έλεγχο στοιχείων του τηλεπικοινωνιακού δικτύου



**Σχήμα 10.3 - NETMAN Responsibility model**

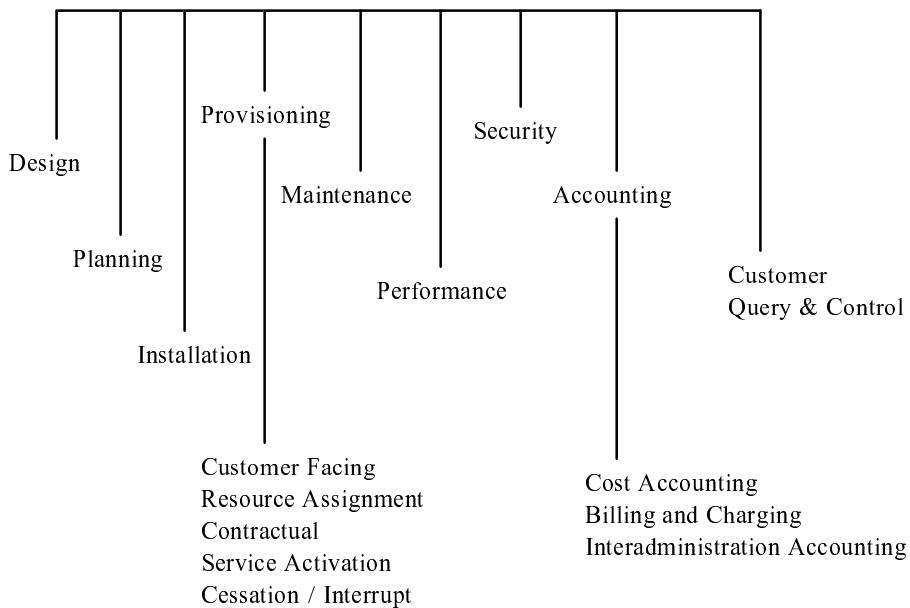
Η πλευρά κύκλου ζωής της διαχείρισης συστημάτων (Σχήμα 10.4) έχει σαν σκοπό της την ταξινόμηση των διαφόρων εργασιών, οι οποίες είναι μέρος του συνολικού εύρους εφαρμογής της διαχείρισης τηλεπικοινωνιακών συστημάτων ευρείας ζώνης. Η ταξινόμηση των εργασιών αυτών σε κλάσεις οδηγεί στον καθορίσμο των λειτουργικών περιοχών του TMN, Telecommunications Management Functional Areas-TMFAs.



## Σχήμα 10.4 - Μοντέλο κύκλου ζωής TMN

Στην συνέχεια θα ορίσουμε την λειτουργικότητα που πρέπει να προσφέρει το TMN. Αυτό σκοπό έχει να παρέχει ένα ευρύ φάσμα από διαχειριστικές λειτουργίες, οι οποίες καλύπτουν περιοχές, όπως τον προγραμματισμό, την λειτουργικότητα και την συντήρηση των τηλεπικοινωνιακών δικτύων. Μια καλή προσέγγιση είναι οι πέντε λειτουργικές περιοχές (Configuration management, Fault management, Accounting management, Performance management, Security management), που καθορίζονται στα πρότυπα του ISO και τις συστάσεις της CCITT. Άλλα, μέσα από την θεώρηση, ότι η λειτουργικότητα του TMN πρέπει να διευθετεί και λειτουργίες pre-service και future service, οι πέντε λειτουργικές περιοχές της διαχείρισης δικτύων μπορούν να επεκτάθουν στις ακόλουθες εννέα (Σχήμα 10.5):

- *Design (Σχεδιασμός δικτύου)*
- *Planning (Μακροπρόθεσμος σχεδιασμός τηλεπικοινωνιών)*
- *Installation (Εγκατάσταση)*
- *Provisioning (Λιαχείριση πόρων)*
- *Maintenance (Συντήρηση)*
- *Performance Management (Λιαχείριση επιδόσεων)*
- *Security Management (Λιαχείριση ασφαλείας)*
- *Accounting Management (Λογιστική παρακολούθηση)*
- *Customer Query & Control (Εξυπηρέτηση πελατών με πρόσβαση στο TMN)*



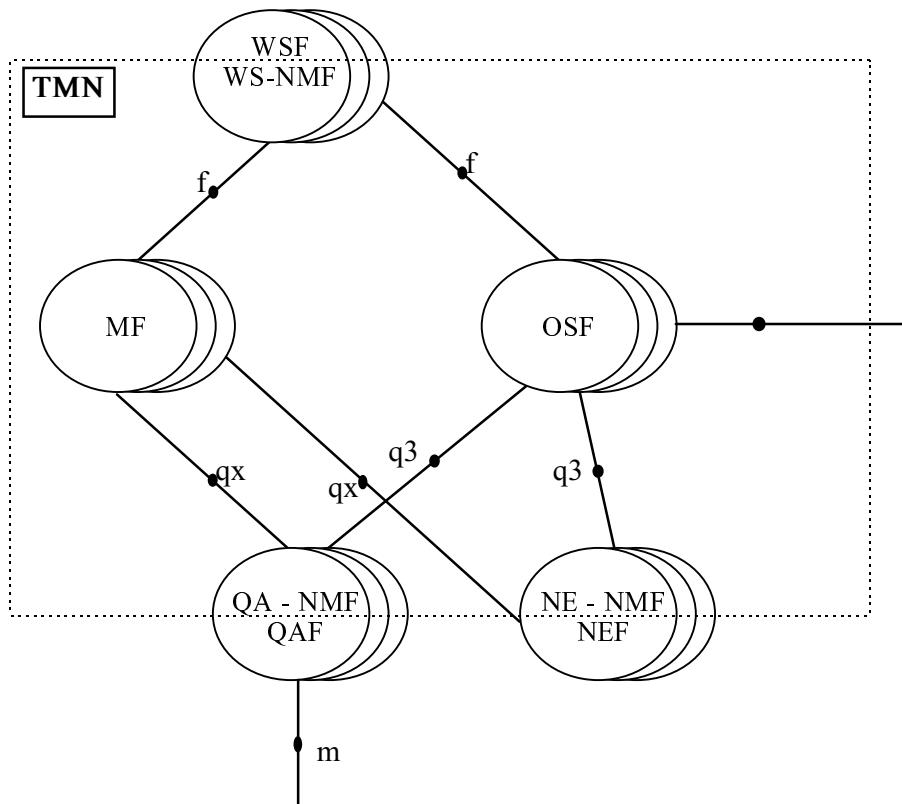
**Σχήμα 10.5 - Telecommunications Management Functional Areas-TMFAs**

## 10.2. Λειτουργικό Μοντέλο του TMN

Η λειτουργικότητα του TMN παρέχει τα μέσα για τη μεταφορά και την επεξεργασία πληροφορίας σχετικής με τη διαχείριση των τηλεπικοινωνιακών δικτύων. Αποτελείται από μονάδες λειτουργιών, οι οποίες είναι οι παρακάτω:

- Operations systems function (OSF) blocks
- Mediation function (MF) blocks
- Data communication function (DCF) blocks
- Network element - Network management function (NEF) blocks
- Workstation function (WF) blocks
- Q adaptor function (QAF) blocks

και οι οποίες υποστηρίζουν τις λειτουργίες διαχείρισης του TMN.



**Σχήμα 10.6 - TMN μονάδες λειτουργιών και σημεία αναφοράς**

### *Operations systems function (OSF) blocks*

Οι TMN λειτουργίες διαχείρισης και σχεδιασμού προσφέρονται από OSFs. Υπάρχει ανάγκη από πολλούς τύπους OSFs προκειμένου να διαχειριστούμε και να σχεδιάσουμε τα σημερινά τηλεπικοινωνιακά δίκτυα και τις ανάλογες υπηρεσίες. Σύμφωνα με την ITU-T M.3010 υπάρχουν τέσσερα διαφορετικά OSFs προκειμένου να υποστηρίζουν τα παρακάτω επίπεδα διαχείρισης:

- **Business management layer (Επίπεδο Διαχείρισης Επιχείρησης):**
- **Service management layer (Επίπεδο Διαχείρισης Υπηρεσιών):**
- **Network management layer (Επίπεδο Διαχείρισης Δικτύου):**
- **Subnetwork management layer (Επίπεδο Διαχείρισης Υποδικτύου):**

Ο παραπάνω διαχωρισμός βοηθά το σχεδιαστή δικτύων διαχείρισης να καταλάβει τη διαφορά ανάμεσα στη διαχείριση φυσικών οντότητων (π.χ. στοιχείων του δικτύου, πόρων, κ.ά.) και στη διαχείριση νοητών οντοτήτων (π.χ. υπηρεσιών, ασφάλειας, συμβολαίων, σχεδιασμού εργασιών, κ.ά.).

### ***Mediation function (MF) blocks***

Οι μονάδες αυτές λειτουργιών επιτρέπουν στις υπόλοιπες TMN μονάδες λειτουργιών να επικοινωνούν μεταξύ τους παρ' ότι προσφέρουν διαφορετικά σημεία αναφοράς ή σημεία διασύνδεσης. Με άλλα λόγια, μια MF μονάδα προσφέρει ένα σύνολο από λειτουργίες διασύνδεσης/αναμετάδοσης.

### ***Data communication function (DCF) blocks***

Η μονάδα αυτή λειτουργιών χρησιμοποιείται από τη TMN λειτουργία επικοινωνίας με σκοπό την ανταλλαγή πληροφορίας μεταξύ TMN μονάδων λειτουργιών. Ο κύριος ρόλος της DCF είναι να παρέχει μεταφορά της πληροφορίας για τις επικοινωνίες μεταξύ: OS/OS, OS/NE, NE/NE, WS/OS, και WS/NE. Η μονάδα λειτουργιών DCF μπορεί να υποστηριχθεί από τις υπηρεσίες μεταφοράς πολλών διαφορετικών τύπων υποδικτύων. Αυτά μπορεί να περιλαμβάνουν γραμμές σημείο-προς-σημείο, τοπικά δίκτυα υπολογιστών, δίκτυα ευρείας ζώνης, κ.ά.

### ***Network element - Network management function (NEF) blocks***

Στοιχεία δικτύου είναι οι μεταγωγείς, ψηφιακά συστήματα cross connects, add-drop πολυπλέκτες, πολυπλέκτες, ψηφιακοί loop carriers. Η εξέλιξη των συστημάτων αυτών είναι τέτοια ώστε πολλές λειτουργίες OSF και MF έχουν εισαχθεί μέσα στα στοιχεία αυτά.

Μερικά παραδείγματα NE-NMFs αποτελούν οι παρακάτω λειτουργίες:

- protocol conversion
- address mapping
- message conversion
- routing
- data collection and storage (π.χ. performance, accounting)
- data backup
- self-healing
- self-testing
- self-fault localization
- NE level alarm analysis
- operations data transport

### ***Workstation function (WF) blocks***

Η λειτουργία αυτή παρέχει τα μέσα για την επεξήγηση πληροφορίας διαχείρισης σε χρήστες με τη μετάφραση της διαχειριστικής πληροφορίας από "F" μορφές διασύνδεσης σε "G" μορφές διασύνδεσης.

### ***Q adaptor function (QAF) blocks***

Η λειτουργία Q adaptor QA NMF παρέχει μετάφραση/μετατροπή μεταξύ ενός TMN σημείου αναφοράς και ενός όχι-TMN σημείου αναφοράς. Μια τέτοιου είδους λειτουργία επιτρέπει τη διαχείριση όχι-TMN στοιχείων δικτύων μέσω του TMN περιβάλλοντος.

### **1.3. Σημεία αναφοράς (reference points)**

Σημείο αναφοράς ονομάζουμε ένα νοητό σημείο στο οποίο λαμβάνει χώρα ανταλλαγή πληροφορίας μεταξύ μη αλληλοκαλυπτόμενων μονάδων λειτουργιών. Το σημείο αναφοράς θα αποτελέσει σημείο διασύνδεσης στην περίπτωση που οι δύο μονάδες λειτουργιών εισαχθούν σε διαφορετικά φυσικά συστήματα.

Σημεία αναφοράς που συναντά κανείς στο TMN είναι τα παρακάτω:

#### q σημείο αναφοράς

Αυτό συνδέει τις OSF, MF, NEF και QAF μονάδες λειτουργιών μεταξύ τους είτε άμεσα, είτε μέσω της DCF μονάδας λειτουργιών. Μέσα στην κλάση των q σημείων αναφοράς, το q3 συνδέει μονάδες NEF λειτουργιών με OSF μονάδες λειτουργιών, MF με OSF μονάδες λειτουργιών, QAF με OSF μονάδες λειτουργιών, OSF με OSF μονάδες λειτουργιών. Το qx σημείο αναφοράς συνδέει MF με MF μονάδες λειτουργιών, MF με NEF μονάδες λειτουργιών και MF με QAF μονάδες λειτουργιών.

#### f σημείο αναφοράς

Το f σημείο αναφοράς συνδέει μονάδα OSF και MF λειτουργιών με μονάδα WSF λειτουργιών.

#### x σημείο αναφοράς

Το x σημείο αναφοράς συνδέει OSF μονάδες λειτουργιών που βρίσκονται σε διαφορετικά TMNs ή μια OSF μονάδα λειτουργιών που βρίσκεται σε ένα TMN και της ανάλογης μονάδας OSF λειτουργιών που βρίσκεται σε ένα όχι-TMN περιβάλλον.

#### g σημείο αναφοράς

Το g σημείο αναφοράς δεν θεωρείται ότι αποτελεί μέρος του TMN παρ' ότι μεταφέρει TMN πληροφορία. Τα όχι-TMN g σημεία αναφοράς θεωρούνται τοποθετημένα έξω από το TMN μεταξύ της WSF μονάδας λειτουργιών και του χειριστή.

#### m σημείο αναφοράς

Τα μ σημεία αναφοράς είναι επίσης τοποθετημένα έξω από το TMN μεταξύ μονάδας λειτουργιών QAF και όχι-TMN διαχειρίζομενων οντοτήτων. Αυτό θα επιτρέψει τη διαχείριση όχι-TMN NEs μέσα από το TMN περιβάλλον.

## 10.4. Φυσική αρχιτεκτονική του TMN και σημεία διασύνδεσης

Οι TMN λειτουργίες μπορούν να υλοποιηθούν με μια ποικιλία από φυσικές διαρθρώσεις. Στο ακόλουθο σχήμα βλέπουμε μια απλοποιημένη φυσική αρχιτεκτονική για το TMN. Μια φυσική αρχιτεκτονική του TMN παρέχει τα μέσα για τη μεταφορά και την επεξεργασία πληροφορίας σχετικής με τη διαχείριση τηλεπικοινωνιακών δικτύων. Μια φυσική αρχιτεκτονική αποτελείται από τα ακόλουθα φυσικά τμήματα.

- Operations systems (OSs)
- A data communications network (DCN)
- Mediation devices (MDs)
- Workstations (WSs)
- Network elements (NEs)
- Q adaptors (Qas)

Λόγω της πολυπλοκότητας των σημερινών τηλεπικοινωνιακών δικτύων, σε κάποιες περιπτώσεις είναι πιθανό να μην εμφανίζονται κάποια από τα παραπάνω φυσικά τμήματα (MDs, QAs). Επίσης αξίζει να παρατηρήσουμε στο σχήμα ότι τα σημεία αναφοράς μετατρέπονται σε σημεία διασύνδεσης (interfaces), όταν οι μονάδες λειτουργιών που συνδέονται ανήκουν σε διαφορετικά φυσικά τμήματα.

### ***TMN σημεία διασύνδεσης (interfaces)***

Οι διασύνδεσεις μεταξύ NEs, OSs, WSs, QAs, MDs μέσω ενός δικτύου επικοινωνίας δεδομένων παρέχονται μέσα από καλά ορισμένα σημεία διασύνδεσης. Τα σημεία διασύνδεσης αυτά εξασφαλίζουν τη διασυνδεσμότητα συστημάτων προκειμένου αυτά να ολοκληρώσουν μια TMN λειτουργία διαχείρισης.

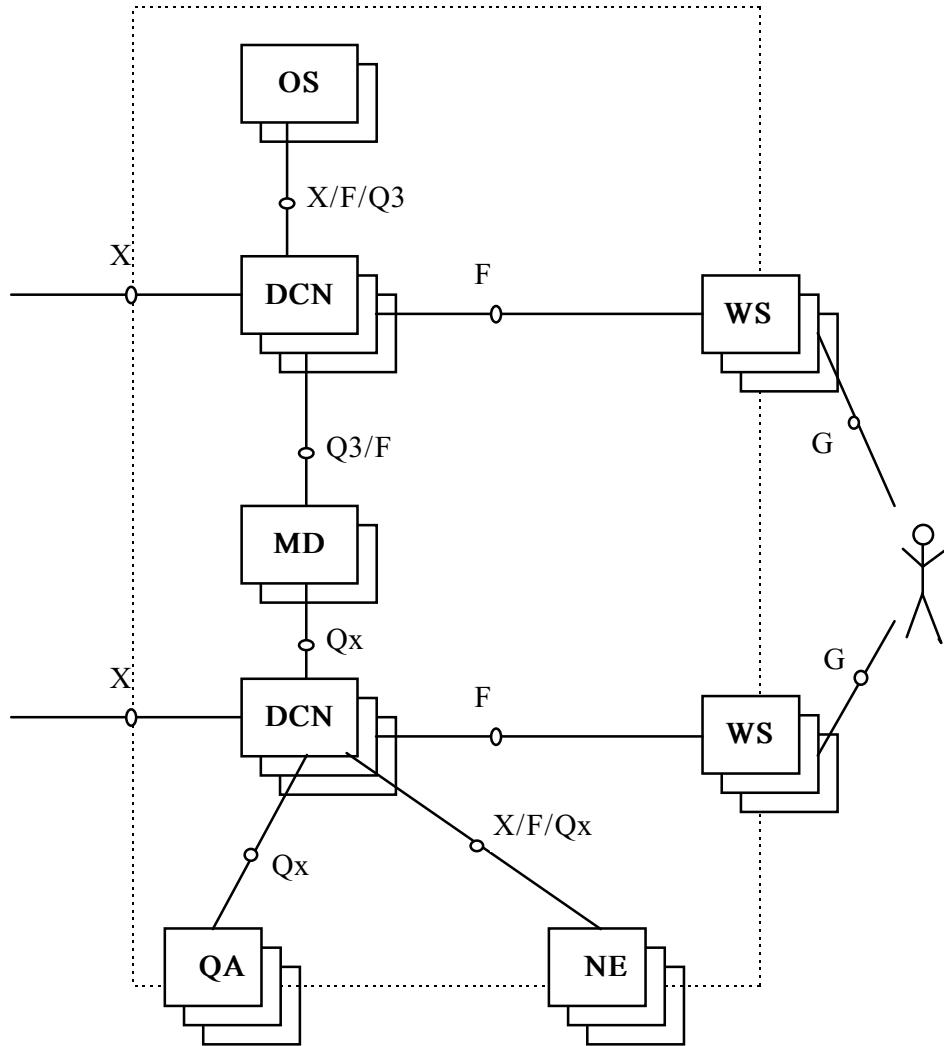
Αφού τα TMN σημεία διασύνδεσης βασίζονται στις γενικότερες έννοιες του OSI μοντέλου αναφοράς πρωτόκολλων (βλέπε ISO 7498), τα πρωτόκολλα επικοινωνίας που θα καθοριστούν για τα σημεία διασύνδεσης αυτά θα πρέπει να συμμορφώνονται με τις ανάλογες συστάσεις, εκτός και εάν οικονομικοί λόγοι δεν το επιτρέπουν.

### **Qx σημείο διασύνδεσης**

Ένα Qx σημείο διασύνδεσης το οποίο υποστηρίζει ένα περιορισμένο σύνολο λειτουργιών χρησιμοποιώντας μια απλή στοίβα πρωτοκόλλων είναι κατάλληλο για στοιχεία δικτύων τα οποία απαιτούν λίγες OAM λειτουργίες και χρησιμοποιούνται με τεράστια

συχνότητα. Μια εφαρμογή του Qx σημείου διασύνδεσης θα ήταν η αμφίδρομη μεταφορά πληροφορίας σχετικής με απλά γεγονότα, όπως αλλαγές σε καταστάσεις, loopback ελέγχους, κ.ά. Πρωτόκολλα που επιλέγονται για τέτοιες εφαρμογές απλά απαιτούν κάποιες λειτουργίες από τα δύο πρώτα επίπεδα της στοίβας OSI.

Qx σημεία διασύνδεσης με σύνθετη στοίβα πρωτοκόλλων μπορούν να υποστηρίζουν μεγάλα σύνολα από OAM λειτουργίες και απαιτούν επιπρόσθετες υπηρεσίες από τα επίπεδα 3 έως και 7 της στοίβας OSI.



**Σχήμα 10.7 - TMN Φυσική αρχιτεκτονική**

#### Q3 σημείο διασύνδεσης

Το Q3 σημείο διασύνδεσης υποστηρίζει το πιο σύνθετο σύνολο λειτουργιών και απαιτεί πολλές υπηρεσίες πρωτοκόλλων προκειμένου να το υποστηρίξει. Οι απαιτήσεις σε πρωτόκολλα για κάθε σύνολο OAM λειτουργιών πρέπει να υποστηρίζετε με επιλογές πρωτοκόλλων από τα επίπεδα 1 έως 7 του OSI μοντέλου αναφοράς πρωτόκολλων. Η παράλειψη κάποιων από αυτά μπορεί να είναι απαραίτητη για λόγους οικονομίας ή επιδόσεων. Αυτή τη στιγμή συστάσεις πρωτόκολλων για τη διασύνδεση OS με NE δίνονται στα ANSI T1-204-1989 και T1-208-1989.

### X σημείο διασύνδεσης

Το X σημείο διασύνδεσης υποστηρίζει ένα σύνολο από OS σε OS λειτουργίες μεταξύ TMNs ή μεταξύ ενός TMN και ενός άλλου είδους δικτύου διαχείρισης και απαιτεί πολλές υπηρεσίες πρωτόκολλων προκειμένου να υποστηρίζει αυτό το σύνολο λειτουργιών.

### F σημείο διασύνδεσης

Το F σημείο διασύνδεσης υποστηρίζει ένα σύνολο από λειτουργίες για τη σύνδεση σταθμών εργασίας σε φυσικά τμήματα που εμπεριέχουν OSF ή MF μονάδες λειτουργιών μέσω ενός δικτύου επικοινωνίας δεδομένων.

Όπου είναι αυτό δυνατό τα Q3, Qx, και X σημεία θα διασύνδεσης θα υποστηρίζονται από τα CMIP/CMIS.

## 10.5. Λειτουργική δομή ενός Operations System (OS) \* [SMIT93]

Ένα OS αποτελείται από δύο κύρια τμήματα, τις Εφαρμογές Διαχείρισης και τη Βάση Πληροφορίας. Η Βάση Πληροφορίας αποτελεί την πραγματοποίηση της δυνατότητας αποθήκευσης πληροφορίας του TMN. Λεν πρέπει να την συγχέουμε με την MIB η οποία είναι μια νοητή αποθήκη όλων των πληροφοριών του TMN και η οποία μπορεί να πραγματοποιηθεί με βάσης πληροφορίας τοποθετημένες σε OS, MD και NEs.

Οι Εφαρμογές Διαχείρισης αποτελούνται από Γενικές Εφαρμογές Διαχείρισης οι οποίες εξυπηρετούν μια σειρά Ειδικές Εφαρμογές Διαχείρισης. Μονάχα οι τελευταίες είναι εκείνες με τις οποίες επικοινωνεί ο χρήστης. Οι Γενικές Εφαρμογές Διαχείρισης είναι κοινές TMN λειτουργίες. Παραδείγματα Γενικών Λειτουργιών Διαχείρισης αποτελούν η διαχείριση διάρθρωσης (configuration management), η διαχείριση γεγονότων (event management), καθώς και εφαρμογές υποδομής όπως διαχείριση της βάσης δεδομένων, διαχείριση των επικοινωνιών, υπηρεσίες επικοινωνίας με τον χρήστη, υπηρεσίες ασφάλειας κ.ά. Η πρώτη ομάδα προσφέρει βασικές εφαρμογές, πάνω στις οποίες θα στηριχθούν οι Ειδικές Εφαρμογές Διαχείρισης, ενώ η δεύτερη ομάδα παρέχει πιο στοιχειώδεις και χαμηλότερου επιπέδου εφαρμογές.

Άλλες λειτουργίες ενός OS είναι οι παρακάτω:

### Dialogue Manager (TMN-DM)

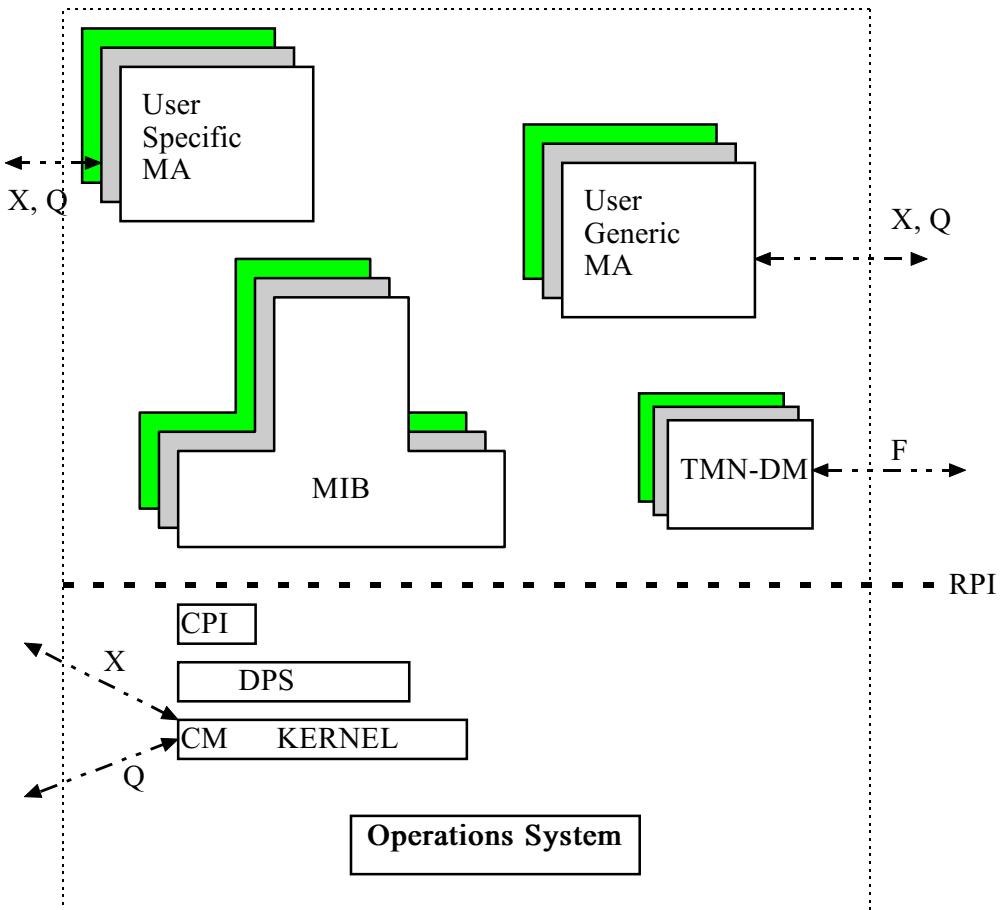
Η λειτουργία αυτή υποστηρίζει το διάλογο κατά τη διασύνδεση με το χρήστη.

### Management Information Base Function (MIBFs)

---

\* Περισσότερες λεπτομέρειες για τη δομή λειτουργίας ενός Operations System (OS) μπορεί να βρει κανείς στο deliverable GUIDELINE Deliverable ME8: "TMN Implementation Architecture," 03/DOW/SAR/DS/B/012/b3, RACE Project R1024 NETMAN, August 1992.

Αυτή είναι μια λειτουργία διαχείρισης της αποθήκης πληροφορίας διαχείρισης. Παραδείγματα MIBFs είναι μια View Manager Function, μια Object Manipulation & Management Function και μια TMN Directory Function.



### Σχήμα 10.8 - Λειτουργική δομή ενός Operations System

Οι παραπάνω λειτουργίες (προσανατολισμένες προς την εφαρμογή) θα υποστηριχθούν από ένα αριθμό λειτουργιών οι οποίες ονομάζονται TMN platform. Παραδείγματα TMN platform λειτουργιών είναι οι παρακάτω:

- Run Time Platform Interface (RPI)
- Computing Platform Interface (CPI)
- TMN Platform Kernel
- Communications Management functions, και
- Distributed Processing Support (DPS)

## 10.6. Βάση Πληροφορίας Διαχείρισης

Η MIB αποτελεί τη νοητή αποθήκη όλων των πληροφοριών που κρατά το TMN μαζί με τις πληροφορίες για το ίδιο το TMN. Μια βάση πληροφορίας είναι απαραίτητη προκειμένου να αποθηκεύονται πληροφορίες για τη διάρθρωση του δικτύου και των συστημάτων, τους πελάτες και τις υπηρεσίες, τις παρούσες επιδόσεις του δικτύου καθώς και πληροφορίες από το παρελθόν, παραμέτρους ασφάλειας και λογιστικές πληροφορίες. Θα μπορούσαμε να περιγράψουμε την MIB αυτή σαν μια συλλογή από βάσεις δεδομένων που κρατιούνται από υπολογιστές που είναι σχετικοί με το TMN. Η συλλογή αυτή θα μπορούσε να περιλαμβάνει μεγάλες βάσεις δεδομένων με τα αρχεία των πελατών, των υπηρεσιών και λεπτομέρειες για τη χρήση, μικρότερες βάσεις δεδομένων με περιφερειακές πληροφορίες επιδόσεων και διάρθρωσης και πίνακες στη μνήμη με τα στοιχεία του δικτύου ώστε να είναι δυνατές άμεσες αποφάσεις για επαναδρομολόγηση κ.ά.

Μερικές απαιτήσεις κλειδιά από την αρχιτεκτονική όλης της πληροφορίας αυτής είναι οι παρακάτω [SMIT93]:

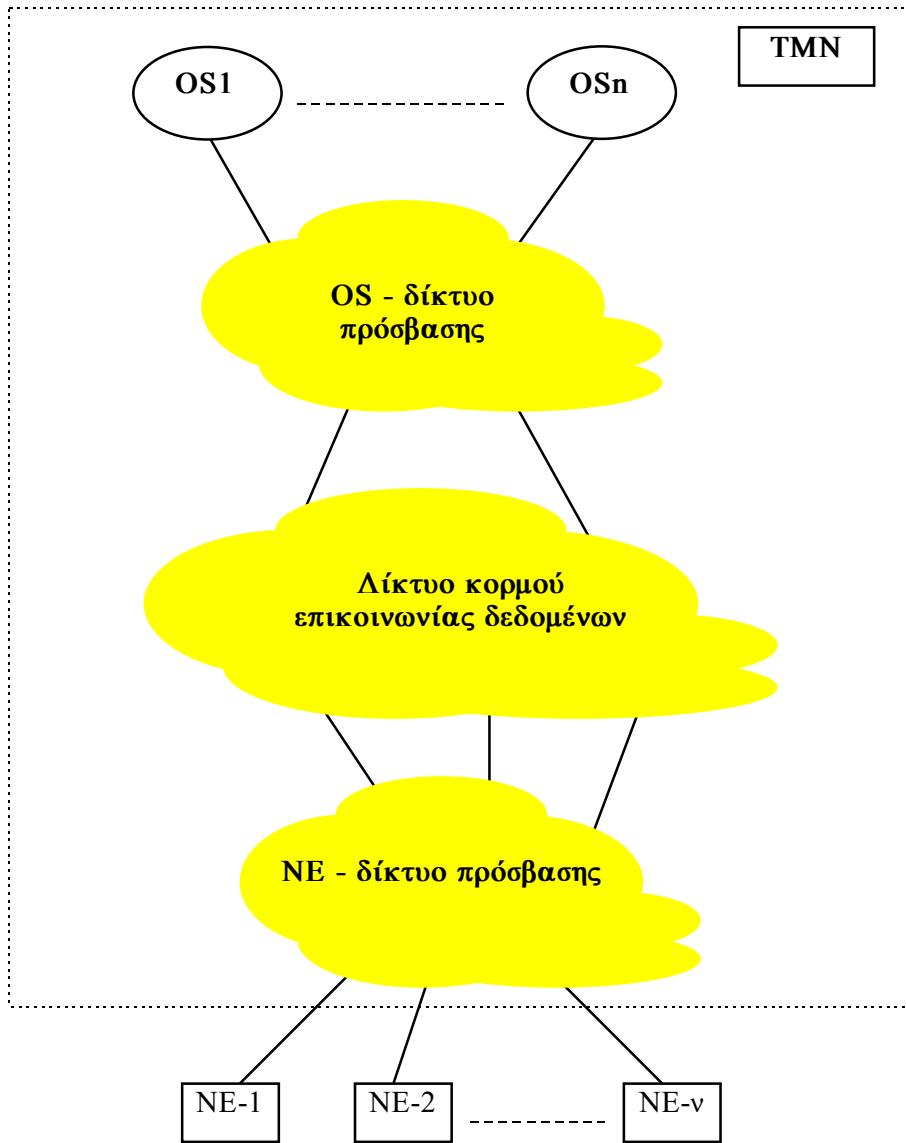
- Η ανάγκη για τεχνικές και μεθόδους ορισμού ενός **εννοιολογικού σχήματος (conceptual schema)**, το οποίο θα υποστηρίζει όλες τις πλευρές των υπηρεσιών και των λειτουργιών του δικτύου προς όλους τους χρήστες του TMN. Αυτές οι προσεγγίσεις θα πρέπει να υποστηρίζουν την εξελικτική αναβάθμιση των συστημάτων διαχείρισης δικτύων.
- Επειδή η ανάπτυξη ενός σύνθετου conceptual schema είναι ιδιαίτερα δύσκολη, η αρχιτεκτονική της πληροφορίας πρέπει να υποστηρίζει μια **διαχωρισμός των ενδιαφερόντων (separation of concerns)** με ένα τρόπο χρήσιμο για το πλαίσιο αναφοράς του TMN.
- Θα απαιτηθούν μεγάλες βάσεις αποθήκευσης **πλούσιον περιεχομένου γνώσεων (knowledge rich)** προκειμένου να υποστηρίζουν, για παράδειγμα, εφαρμογές υποστήριξης αποφάσεων. Αυτές πιθανώς θα απαιτήσουν τον ορισμό σχημάτων μεγαλής πολυπλοκότητας.
- Τέλος, απαιτούνται MIBs με τεχνικές για διαχείριση **μεγάλων όγκων πληροφορίας (large volumes of information)**. Οι MIBs πρέπει να υλοποιούνται με τέτοιο τρόπο ώστε οι επιδόσεις και οι απαιτήσεις πρόσθιασης να ικανοποιούνται όταν τα δεδομένα αποθήκευονται σε βάσεις δεδομένων οι οποίες γενικά θα είναι κατανευμημένες και ετερογενείς.

## 10.7. Φυσική αρχιτεκτονική του TMN

Ένας μεγάλος αριθμός από μονοπάτια είναι απαραίτητος προκειμένου να προσφερθεί επικοινωνία μεταξύ OSs, NEs και WSs στα σημερινά μεγάλα ψηφιακά τηλεπικοινωνιακά δίκτυα. Δεν είναι πάντοτε οικονομικό να συνδέεις όλα τα NEs σε ένα PSN ή σε ένα OS άμεσα με αποκλειστικές ευκολίες μετάδοσης. Για το λόγο αυτό

ένα NE μπορεί να χρησιμοποιηθεί προκειμένου να συλλέγει πληροφορίες διαχείρισης από άλλα NEs.

Η λειτουργία επικοινωνίας δεδομένων του TMN θα υλοποιηθεί με το σχεδιασμό ενός δίκτυου επικοινωνίας δεδομένων (DCN). Από τη στιγμή που αυτό μπορεί να είναι ένα πολύ μεγάλο δίκτυο μπορεί να διαιρεθεί σε τρία μικρότερα σύμφωνα με το παρακάτω σχήμα [SAHI94]. Τα υποδίκτυα αυτά είναι το OS-δίκτυο πρόσβασης, τα λίκτυα κορμού του δίκτυου επικοινωνίας δεδομένων, και δύο δίκτυα πρόσβασης, τα οποία συνδέουν τα OSs και τα NEs στο δίκτυο κορμού. Αυτά ονομάζονται OS-δίκτυο πρόσβασης και NE-δίκτυο πρόσβασης.



**Σχήμα 6.9 - Μοντέλο αρχιτεκτονικής υλοποίησης TMN**

### ***OS-δίκτυο πρόσβασης***

Το OS-δίκτυο πρόσβασης παρέχει επικοινωνιακά μονοπάτια μεταξύ OSs και συνδέει τα OSs πάνω στο δίκτυο κορμού επικοινωνίας δεδομένων. Το απλούστερο OS-δίκτυο

πρόσβασης χρησιμοποιεί μισθωμένες γραμμές σημείο-προς-σημείο προκειμένου να παρέχει τις αναγκαίες επικοινωνίες μεταξύ OSs και μεταξύ OSs και NEs. Μια πιο σύνθετη αρχιτεκτονική θα χρησιμοποιούσε για δίκτυο πρόσβασης κάποιο τοπικό δίκτυο υπολογιστών υψηλών-ταχυτήτων ή ένα FDDI προκειμένου να συνδέσει τα OSs στο δίκτυο κορμού επικοινωνίας δεδομένων.

### **NE-δίκτυο πρόσβασης**

Το NE-δίκτυο πρόσβασης συνδέει τα NEs με το δίκτυο κορμού επικοινωνίας δεδομένων και παρέχει δυνατότητα επικοινωνίας μεταξύ των NEs. Το δίκτυο πρόσβασης αυτό χρησιμοποιεί ***embedded operations channels*** προκειμένου να ελαχιστοποιήσουν το κόστος πρόσβασης των NEs στο δίκτυο κορμού. Η έννοια αυτή χρησιμοποιείται για τη κοινή χρήση πόρων του δικτύου (ευκολίες μετάδοσης, NEs, διασυνδέσεις) μεταξύ διαφορετικών εφαρμογών, για την ελαχιστοποίηση του κόστους μεταφοράς πληροφορίας διαχείρισης. Gateway NEs και Intermediate NEs μπορεί να χρησιμοποιηθούν προκειμένου να σχεδιαστεί ένα NE-δίκτυο πρόσβασης.

### **Δίκτυο κορμού επικοινωνίας δεδομένων**

Το δίκτυο κορμού επικοινωνίας δεδομένων μπορεί να είναι ένα ιδιωτικό δίκτυο αποκλειστικών γραμμών, ένα δίκτυο μεταγωγής κυκλώματος, ένα δίκτυο μεταγωγής πακέτου ή ένας συνδυασμός των παραπάνω δικτύων.

## 10.8. Βιβλιογραφία

- [AIDA94] Telecommunications Network Management into the 21st Century: Techniques, Standards, Technologies and Applications, Ed. S. Aidarous, T. Plevyak, IEEE Press, 1994.
- [MINZ89] S. Minzer, D. Spears, "New Directions in Signalling for Broadband ISDN," IEEE Communications Magazine, Vol. 27, No. 2, Feb. 1989.
- [MINZ91] S. Minzer, "A Signalling Protocol for Complex Multimedia Services," IEEE JSAC, Vol. 9, No. 9, Dec. 91.
- [ΜΠΙΛΗ94] Ευρ. Μπιλής, "Εισαγωγή στο TMN.,," Σημειώσεις, Αθήνα 1994.
- [RACE92] RACE Project NETMAN R1024, "Tele-communications Management Specifications," Deliverable 6, 1992.
- [SMIT93] R. Smith, E.H. Mamdami and J. Callaghan, The Management of Telecommunications Networks, Ellis Horwood, 1993.
- [TMN\_92] "Telecommunications Management Conceptual Models," S. Plagemann and T. Turner, Broadcom, 1992.



# Παράρτημα A

## Το Περιβάλλον Διαχείρισης του Carnegie Mellon University (CMU)

Ένα από τα πλεονεκτήματα του SNMP είναι ότι υπάρχουν public domain υλοποιήσεις διαχειριστικών συστημάτων. Παράδειγμα τέτοιου public domain λογισμικού αποτελούν τα συστήματα που έχουν αναπτυχθεί στο Carnegie Mellon University και στο MIT. Στο παράρτημα αυτό δίνεται μια σύντομη περιγραφή του συστήματος του CMU, το οποίο αποτελείται από τις υλοποιήσεις του SNMP agent και μιας βιβλιοθήκης μικρών και απλών διαχειριστικών εφαρμογών.

Η περιγραφή αναφέρεται στην έκδοση 1.0 του CMU SNMP distribution. Αυτή περιλαμβάνει την SNMP/ASN.1 βιβλιοθήκη, μερικές απλές διαχειριστικές εφαρμογές και την αντίστοιχη τεκμηρίωση. Ο SNMP agent και όλες οι εφαρμογές δίνονται σε μορφή κώδικα C, ενώ έχουν καταβληθεί προσπάθειες στην υλοποίηση ώστε ο κώδικας να είναι αποδοτικός, γρήγορος και μεταφέρσιμος. Για παράδειγμα ο κώδικας των εφαρμογών (της έκδοσης 1.0 πάντα) μπορεί να μεταγλωτιστεί και να τρέξει στα ακόλουθα συστήματα : IBM PC/RT που τρέχει το σύστημα ACIS Release 3, Sun3/60 με SUNOS 3.5, DEC microVax με Ultrix 2.2 και DECStation 3100's που Ultrix 3.0. Αναμένεται ότι τρέχει σε κάθε μηχάνημα που υποστηρίζει τον μηχανισμό των sockets του Berkeley Unix. Ήδη τρέχει σε ένα πλήθος μηχανημάτων του δικτύου του EMPI (για παράδειγμα ο theseas και ο phgasos).

Ο SNMP agent εκτελείται με την εντολή snmpd. Οι εφαρμογές είναι : snmpget, snmpgetnext, snmpwalk, snmptrapd, snmptrap, snmpstatus, snmpnetstat, snmpptest. Η περιγραφή της MIB που χρησιμοποιούν οι εφαρμογές δεν περιέχεται μέσα στα εκτελέσιμα αρχεία αλλά μέσα στο αρχείο mib.txt, ένα αρχείο text γραμμένο σε μορφή ASN.1 σύμφωνα με το RFC1066. Η συντακτική ανάλυση (parsing), και η εκτύπωση των μεταβλητών και των αντίστοιχων object identifiers οδηγούνται από αυτό το αρχείο, που διαβάζεται όταν ξεκινάει ο agent. Αυτό επιτρέπει την εύκολη επέκταση και προσθήκη νέων τύπων MIB καθώς ένα αρχείο περιγραφής σύμφωνο με το RFC1066, αρκεί να προστεθεί.

Ακολουθεί περιγραφή των εφαρμογών που αναφέρθηκαν. Για κάθε μια δίνεται μια σύντομη περιγραφή της λειτουργίας και της σύνταξης της. Περισσότερες πληροφορίες μπορούν να βρεθούν μέσω της εντολής man του UNIX (για παράδειγμα man snmpget).

- **snmpget** : ουσιαστικά υλοποιεί την εντολή-αίτηση GET του SNMP. Επικοινωνεί μέσω του πρωτοκόλου SNMP με τον ζητούμενο κόμβο (τον αντίστοιχο SNMP agent - είτε είναι του CMU είτε όχι) και διαβάζει την(ις) ζητούμενη(ες) μεταβλητή(ές). Η σύνταξη είναι :

```
snmpget host community variable-name [variable-name]...
```

Η παράμετρος host είναι το όνομα ή η ip διεύθυνση του κόμβου, η παράμετρος community είναι το community name (συνήθως public) και τα variable-name είναι τα ονόματα των ζητούμενων μεταβλητών. Παράδειγμα χρήσης :

```
snmpget theseas.ntua.gr public system.sysdescr.0
system.sysUpTime.0
```

Με αυτή την εντολή τυπώνονται οι τιμές των μεταβλητών sysdescr και sysUpTime του system group της MIB του theseas.ntua.gr :

```
Name: system.sysDescr.0
OCTET STRING- (ascii): Unix 4.3BSD
Name: system.sysUpTime.0
Timeticks: (60943871) 7 days, 1:17:18
```

- **snmpgetnext** : ουσιαστικά υλοποιεί την εντολή-αίτηση GET NEXT του SNMP. Επικοινωνεί μέσω του πρωτοκόλου SNMP με τον ζητούμενο κόμβο (τον αντίστοιχο SNMP agent - είτε έιναι του CMU είτε όχι) και διαβάζει την λεξικογραφικά επόμενη λέξη της(ων) ζητούμενης(ων) μεταβλητής(ων). Η σύνταξη είναι :

```
snmpgetnext host community variable-name [variable-name]...
```

Η παράμετρος host είναι το όνομα ή η ip διεύθυνση του κόμβου, η παράμετρος community είναι το community name (συνήθως public) και τα variable-name είναι τα ονόματα των ζητούμενων μεταβλητών. Παράδειγμα χρήσης :

```
snmpgetnext theseas.ntua.gr public system.sysdescr.0
system.sysUpTime.0
```

Με αυτή την εντολή τυπώνονται οι τιμές των μεταβλητών που είναι λεξικογραφικά επόμενες των sysdescr και sysUpTime του system group της MIB του theseas.ntua.gr :

```
Name: system.sysObjectID.0
OBJECT IDENTIFIER:
.iso.org.dod.internet.private.enterprises.cmu.systems.cmuSNMP
Name: interfaces.ifNumber.0
INTEGER: 13
```

- **snmpwalk** : Η εντολή αυτή χρησιμοποιεί συνεχόμενα SNMP GET NEXT για να διαβάσει και να τυπώσει ολόκληρο το δέντρο της MIB του ζητούμενου κόμβου. Η σύνταξη είναι :

```
snmpwalk host community [variable-name]
```

Οι παράμετροι host και community είναι ανάλογοι αυτών των προηγούμενων εντολών. Η τρίτη παράμετρος είναι το όνομα μιας μεταβλητής και είναι προαιρετική. Αν δεν δοθεί ολόκληρη η MIB διαβάζεται. Αν δοθεί ορίζει από ποιό σημείο του δέντρου ξεκινάνε τα GET NEXT. Όλες οι μεταβλητές στο υπόδεντρο με ρίζα αυτή την μεταβλητή τυπώνονται. Για παράδειγμα στην εντολή :

```
snmpwalk theseas.ntua.gr public system
```

το αποτέλεσμα είναι ολόκληρο το system group της MIB :

```
Name: system.sysDescr.0
OCTET STRING- (ascii): Unix 4.3BSD
Name: system.sysObjectID.0
OBJECT IDENTIFIER:
.iso.org.dod.internet.private.enterprises.cmu.systems.cmuSNMP
Name: system.sysUpTime.0
Timeticks: (61037964) 7 days, 1:32:59
```

- **snmptrapd** : Ο agent snmpd δεν καταγράφει τα SNMP traps. Η εφαρμογή snmptrapd κάνει αυτή ακριβώς την δουλειά. Τρέχει και αυτή υπό την μορφή deamon και περιμένει στο port 162 να δεχθεί traps. Αν δεχθεί τα καταγράφει στο sys-log (βλέπε man 8 sys-log), εκτός αν έχει δοθεί η παράμετρος -p, οπότε τυπώνεται στο standard output. Μόνο με root username μπορεί κανείς να τρέξει αυτή την εφαρμογή ώστε να μπορεί να ανοίξει το port 162.

- **snmptrap** : Η εφαρμογή αυτή επιτρέπει την αποστολή trap και έχει την ακόλουθη σύνταξη :

```
snmptrap host community trap-type specific-type device-
description [ -a agent-addr ]
```

Οι παράμετροι host, community είναι ανάλογες με αυτές των προηγούμενων εντολών. Η trap-type και η specific-type είναι integers και ορίζουν το είδος του trap που θα σταλεί. Η device-description είναι μια περιγραφή (κείμενο) που χρησιμοποιείται σαν τιμή της μεταβλητής system.sysDescr.0 που μπαίνει στο πακέτο του trap. Τέλος, η προαιρετική παράμετρος -a agent-addr ορίζει την διεύθυνση από την οποία αναφέρει το trap ότι στάλθηκε. Αν δεν δοθεί αυτή η παράμετρος, το trap αναφέρει τη διεύθυνση του μηχανήματος από το οποίο εκτελέστηκε η εντολή. Για παράδειγμα η εντολή :

```
snmptrap theseas.ntua.gr public 0 0 'SUN 3/60: SUNOS4.0'
```

στέλνει ένα Cold Start trap στον theseas.ntua.gr. Τα traps που ορίζονται είναι :

0	coldStart
1	warmStart
2	linkDown
3	linkUp
4	authenticationFailure
5	egpNeighborLoss
6	enterpriseSpecific

- **snmpstatus** : είναι μια εφαρμογή που επιστρέφει κάποιες γενικές πληροφορίες για ένα κόμβο του δικτύου. Αυτές είναι : η IP διεύθυνση του κόμβου, μια περιγραφή του κόμβου (sysDescr.0), ο χρόνος λειτουργίας του (sysUpTime.0), το άθροισμα όλων των πακέτων που έχει δεχθεί από όλα τα interfaces (ifInUCastPkts.\* + ifInNUCastPkts.\*), το άθροισμα όλων των πακέτων που έστειλε από όλα τα interfaces (ifOutUCastPkts.\* + ifOutNUCastPkts.\*), τον αριθμό των IP πακέτων εισόδου (ipInReceives.0) και τον αριθμό IP πακέτων εξόδου (ipOutRequests.0). Η σύνταξη της εντολής είναι :

```
snmpstatus host [community]
```

Οι παράμετροι έχουν την γνωστή έννοια, ενώ το community name είναι προαιρετικό. Αν δεν δοθεί χρησιμοποιείται το public. Για παράδειγμα η εντολή :

```
snmpstatus theseas.ntua.gr
```

Τυπώνει τις παρακάτω πληροφορίες για τον theseas.ntua.gr :

```
[147.102.1.1]=>[Unix 4.3BSD] Up: 7 days, 2:17:38
Recv/Trans packets: Interfaces: 6780077/6329722 | IP: 6608548/0
```

- **snmpnetstat** : Η εφαρμογή αυτή παρουσιάζει συμβολικά τις τιμές διαφόρων μεγεθών συσχετισμένων με το δίκτυο, που αντλούνται από το δοσμένο σύστημα χρησιμοποιώντας το πρωτόκολλο SNMP. Υποστηρίζει τέσσερις διαφορετικές μορφές ανάλογα με τον αριθμό παραμέτρων που δίνονται :

```
snmpnetstat host community [ -an ]
snmpnetstat host community [ -inrs ]
snmpnetstat host community [ -n ] [ -I interface ] interval
snmpnetstat host community [ -p protocol ]
```

Η πρώτη μορφή τυπώνει μια λίστα των ενεργών sockets. Η δεύτερη παρουσιάζει άλλες πληροφορίες, ανάλογα με τις παραμέτρους που περιγράφονται παρακάτω. Η τρίτη μορφή προσφέρει την δυνατότητα συνεχής παρακαλούθησης (με συγκεκριμένο μεσοδιάστημα) πληροφορίες για την κίνηση πακέτων στο συγκεκριμένο interface. Τέλος, η τέταρτη μορφή δίνει πληροφορίες για το συγκεκριμένο πρωτόκολλο. Οι παράμετροι host, community έχουν τη γνωστή έννοια. Με την παράμετρο -a η εντολή δείχνει την κατάσταση όλων των sockets, με -i δείχνει την κατάσταση των interfaces, με -I interface δίνει πληροφορίες μόνο για το συγκεκριμένο interface, με -n παρουσιάζει τις διευθύνσεις αριθμητικά, με -r δείχνει πληροφορίες για το συγκεκριμένο πρωτόκολλο, με -s δείχνει στατιστικά σχετικά με το πρωτόκολλο και με -t δείχνει τα routing tables. Αναλυτικότερη περιγραφή δίνεται με το man snmpnetstat.

- **snmptest** : Είναι μια αρκετά ευέλικτη εφαρμογή που επιτρέπει παρακολούθηση και διαχείριση πληροφοριών ενός κόμβου με SNMP agent. Αφού τρέξει (παίρνει μόνο δύο παραμέτρους : host και community), δέχεται εντολές SET ή GET, επιτρέποντας έτσι στον χρήστη να διαβάσει και να γράψει μεταβλητές της MIB. Η εφαρμογή παρουσιάζει ένα command line και περιμένει το όνομα μιας ή παραπάνω μεταβλητών και αρχικά βρίσκεται σε κατάσταση διαβάσματος (GET). Με τις εντολές \$G, \$N και \$S, η κατάσταση αλλάζει σε GET, GET NEXT και SET αντίστοιχα. Αναλυτική παρουσίαση με λεπτομέριες χρήσης δίνονται με τα man snmptest.

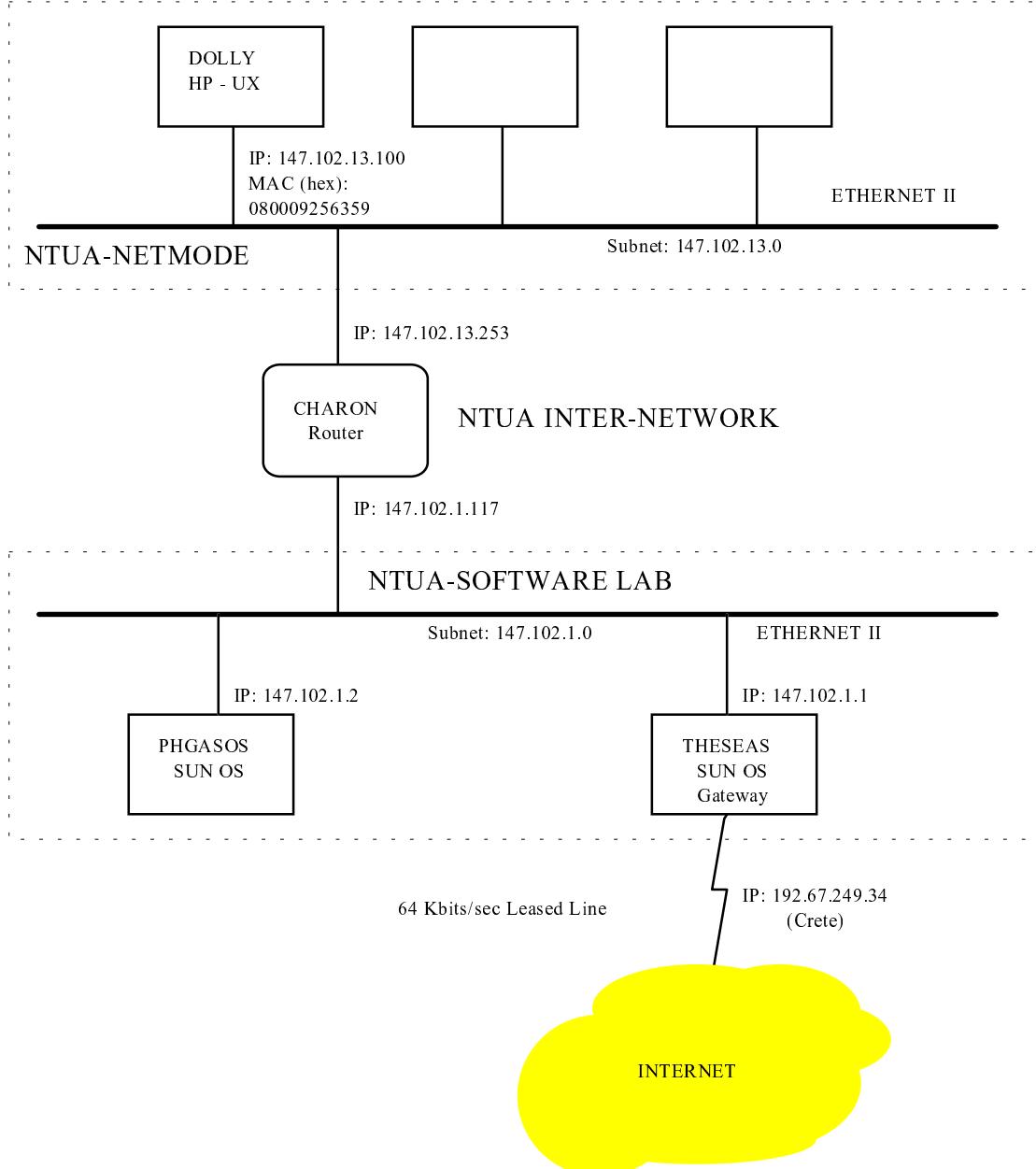
Η έκδοση 1.0 (και οι πιθανές επόμενες εκδόσεις) μπορούν να μεταφερθούν με anonymous FTP από τον host : lancaster.andrew.cmu.edu (128.2.13.21). Ολόκληρη η έκδοση περιέχεται στο αρχείο pub/cmu-snmp1.0.tar, ενώ στο αρχείο pub/mibs περιλαμβάνονται αρχεία με ορισμούς MIB. Αυτά είναι text αρχεία γραμμένα σε μορφή ASN.1 που χρησιμοποιείται στην RFC1066 MIB όπως προαναφέρθηκε.

## Παράρτημα Β

### MIB-II (RFC1213)

#### B.1 ΠΑΡΑΔΕΙΓΜΑ ΥΛΟΠΟΙΗΣΗΣ MIB-II

Ακολουθεί το αρχείο MIB στον κόμβο dolly.ntua.gr. Μερικά από τα στοιχεία που αναφέρονται στο δίκτυο του ΕΜΠ, παρουσιάζονται στο παρακάτω σχήμα. Την στιγμή της μέτρησης υπήρχαν δύο ανοικτές TCP συνδέσεις (rlogin) μεταξύ κόμβων UNIX. Η μία από αυτές ήταν εγκαταστημένη μεταξύ του κόμβου dolly.ntua.gr και του κόμβου phgasos.ntua.gr, όπου έτρεχε η εφαρμογή snmpwalk του περιβάλλοντος διαχείρισης CMU (βλ. και παράρτημα A). Στο MIB αναφέρεται ως 147.102.13.100.1022 (TCP port 1022 του κόμβου dolly) με 147.102.1.2.513 (TCP port 513 του κόμβου phgasos). Η δεύτερη ήταν εγκαταστημένη μεταξύ του κόμβου dolly.ntua.gr και του κόμβου theseas.ntua.gr, όπου έτρεχε η εφαρμογή ηλεκτρονικού ταχυδρομείου. Στο MIB αναφέρεται ως 147.102.13.100.1023 (TCP port 1023 του κόμβου dolly) με 147.102.1.1.513 (TCP port 513 του κόμβου Theseas).



## To Αρχείο MIB II στον Κόμβο dolly.ntua.gr

```

Name: system.sysDescr.0
OCTET STRING- (ascii): HP-UX dolly A.08.07 A 9000/710 2009778770
Name: system.sysObjectID.0
OBJECT IDENTIFIER:
    .iso.org.dod.internet.private.enterprises.11.2.3.2.5
Name: system.sysUpTime.0
Timeticks: (28590694) 3 days, 7:25:06
Name: system.sysContact.0
OCTET STRING- (hex): Y.Scopoulis
Name: system.sysName.0
OCTET STRING- (ascii): dolly.ntua.gr
Name: system.sysLocation.0
OCTET STRING- (hex): NETMODE
Name: system.sysServices.0
INTEGER: 72
Name: interfaces.ifNumber.0
INTEGER: 4
Name: interfaces.ifTable.ifEntry.ifIndex.1
INTEGER: 1
Name: interfaces.ifTable.ifEntry.ifIndex.2
INTEGER: 2
Name: interfaces.ifTable.ifEntry.ifIndex.3
INTEGER: 3
Name: interfaces.ifTable.ifEntry.ifIndex.4
INTEGER: 4
Name: interfaces.ifTable.ifEntry.ifDescr.1
OCTET STRING- (ascii): ni0 Hewlett Packard Network Interface Pseudo
Driver
Name: interfaces.ifTable.ifEntry.ifDescr.2
OCTET STRING- (ascii): nil Hewlett Packard Network Interface Pseudo
Driver
Name: interfaces.ifTable.ifEntry.ifDescr.3
OCTET STRING- (ascii): lo0 Hewlett-Packard Software Loopback

Name: interfaces.ifTable.ifEntry.ifDescr.4
OCTET STRING- (ascii): lan0 Hewlett-Packard LAN Interface Hw Rev 0
Name: interfaces.ifTable.ifEntry.ifType.1
INTEGER: other(1)
Name: interfaces.ifTable.ifEntry.ifType.2
INTEGER: other(1)
Name: interfaces.ifTable.ifEntry.ifType.3
INTEGER: 24
Name: interfaces.ifTable.ifEntry.ifType.4
INTEGER: ethernet-csmacd(6)
Name: interfaces.ifTable.ifEntry.ifMtu.1
INTEGER: 0
Name: interfaces.ifTable.ifEntry.ifMtu.2
INTEGER: 0
Name: interfaces.ifTable.ifEntry.ifMtu.3
INTEGER: 1536
Name: interfaces.ifTable.ifEntry.ifMtu.4
INTEGER: 1500
Name: interfaces.ifTable.ifEntry.ifSpeed.1
Gauge: 1000000
Name: interfaces.ifTable.ifEntry.ifSpeed.2
Gauge: 1000000
Name: interfaces.ifTable.ifEntry.ifSpeed.3
Gauge: 10000000
Name: interfaces.ifTable.ifEntry.ifSpeed.4
Gauge: 10000000
Name: interfaces.ifTable.ifEntry.ifPhysAddress.1
OCTET STRING- (hex):
Name: interfaces.ifTable.ifEntry.ifPhysAddress.2
OCTET STRING- (hex):
Name: interfaces.ifTable.ifEntry.ifPhysAddress.3
OCTET STRING- (hex):
Name: interfaces.ifTable.ifEntry.ifPhysAddress.4

```

OCTET STRING- (hex): 08 00 09 25 63 59  
Name: interfaces.ifTable.ifEntry.ifAdminStatus.1  
INTEGER: up(1)  
Name: interfaces.ifTable.ifEntry.ifAdminStatus.2  
INTEGER: up(1)  
Name: interfaces.ifTable.ifEntry.ifAdminStatus.3  
INTEGER: up(1)  
Name: interfaces.ifTable.ifEntry.ifAdminStatus.4  
INTEGER: up(1)  
Name: interfaces.ifTable.ifEntry.ifOperStatus.1  
INTEGER: down(2)  
Name: interfaces.ifTable.ifEntry.ifOperStatus.2  
INTEGER: down(2)  
Name: interfaces.ifTable.ifEntry.ifOperStatus.3  
INTEGER: up(1)  
Name: interfaces.ifTable.ifEntry.ifOperStatus.4  
INTEGER: up(1)  
Name: interfaces.ifTable.ifEntry.ifLastChange.1  
Timeticks: (0) 0:00:00  
Name: interfaces.ifTable.ifEntry.ifLastChange.2  
Timeticks: (0) 0:00:00  
Name: interfaces.ifTable.ifEntry.ifLastChange.3  
Timeticks: (0) 0:00:00  
Name: interfaces.ifTable.ifEntry.ifLastChange.4  
Timeticks: (947274806) 109 days, 15:19:08  
Name: interfaces.ifTable.ifEntry.ifInOctets.1  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInOctets.2  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInOctets.3  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInOctets.4  
Counter: 43623334  
Name: interfaces.ifTable.ifEntry.ifInUcastPkts.1  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInUcastPkts.2  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInUcastPkts.3  
Counter: 3596  
Name: interfaces.ifTable.ifEntry.ifInUcastPkts.4  
Counter: 57193  
Name: interfaces.ifTable.ifEntry.ifInNUcastPkts.1  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInNUcastPkts.2  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInNUcastPkts.3  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInNUcastPkts.4  
Counter: 24399  
Name: interfaces.ifTable.ifEntry.ifInDiscards.1  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInDiscards.2  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInDiscards.3  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInDiscards.4  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInErrors.1  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInErrors.2  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInErrors.3  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInErrors.4  
Counter: 11  
Name: interfaces.ifTable.ifEntry.ifInUnknownProtos.1  
Counter: 0  
Name: interfaces.ifTable.ifEntry.ifInUnknownProtos.2  
Counter: 0

```
Name: interfaces.ifTable.ifEntry.ifInUnknownProtos.3
Counter: 0
Name: interfaces.ifTable.ifEntry.ifInUnknownProtos.4
Counter: 14503
Name: interfaces.ifTable.ifEntry.ifOutOctets.1
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutOctets.2
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutOctets.3
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutOctets.4
Counter: 2455196
Name: interfaces.ifTable.ifEntry.ifOutUcastPkts.1
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutUcastPkts.2
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutUcastPkts.3
Counter: 3596
Name: interfaces.ifTable.ifEntry.ifOutUcastPkts.4
Counter: 33592
Name: interfaces.ifTable.ifEntry.ifOutNUcastPkts.1
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutNUcastPkts.2
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutNUcastPkts.3
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutNUcastPkts.4
Counter: 43
Name: interfaces.ifTable.ifEntry.ifOutDiscards.1
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutDiscards.2
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutDiscards.3
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutDiscards.4
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutErrors.1
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutErrors.2
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutErrors.3
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutErrors.4
Counter: 0
Name: interfaces.ifTable.ifEntry.ifOutQLen.1
Gauge: 0
Name: interfaces.ifTable.ifEntry.ifOutQLen.2
Gauge: 0
Name: interfaces.ifTable.ifEntry.ifOutQLen.3
Gauge: 0
Name: interfaces.ifTable.ifEntry.ifOutQLen.4
Gauge: 27
Name: interfaces.ifTable.ifEntry.ifSpecific.1
OBJECT IDENTIFIER: .0.0
Name: interfaces.ifTable.ifEntry.ifSpecific.2
OBJECT IDENTIFIER: .0.0
Name: interfaces.ifTable.ifEntry.ifSpecific.3
OBJECT IDENTIFIER: .0.0
Name: interfaces.ifTable.ifEntry.ifSpecific.4
OBJECT IDENTIFIER: .0.0
Name: at.atTable.atEntry.atIfIndex.4.1.147.102.13.253
INTEGER: 4
Name: at.atTable.atEntry.atPhysAddress.4.1.147.102.13.253
OCTET STRING- (hex): 00 00 0C 01 06 71
Name: at.atTable.atEntry.atNetAddress.4.1.147.102.13.253
IpAddress: 147.102.13.253
Name: ip.ipForwarding.0
INTEGER: host(2)
Name: ip.ipDefaultTTL.0
INTEGER: 255
```

Name: ip.ipInReceives.0  
Counter: 70349  
Name: ip.ipInHdrErrors.0  
Counter: 0  
Name: ip.ipInAddrErrors.0  
Counter: 0  
Name: ip.ipForwDatagrams.0  
Counter: 0  
Name: ip.ipInUnknownProtos.0  
Counter: 1  
Name: ip.ipInDiscards.0  
Counter: 0  
Name: ip.ipInDelivers.0  
Counter: 70355  
Name: ip.ipOutRequests.0  
Counter: 37102  
Name: ip.ipOutDiscards.0  
Counter: 0  
Name: ip.ipOutNoRoutes.0  
Counter: 8  
Name: ip.ipReasmTimeout.0  
INTEGER: 30  
Name: ip.ipReasmReqds.0  
Counter: 0  
Name: ip.ipReasmOKs.0  
Counter: 0  
Name: ip.ipReasmFails.0  
Counter: 0  
Name: ip.ipFragOKs.0  
Counter: 0  
Name: ip.ipFragFails.0  
Counter: 0  
Name: ip.ipFragCreates.0  
Counter: 0  
Name: ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.127.0.0.1  
IpAddress: 127.0.0.1  
Name: ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.147.102.13.100  
IpAddress: 147.102.13.100  
Name: ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.127.0.0.1  
INTEGER: 3  
Name: ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.147.102.13.100  
INTEGER: 4  
Name: ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask.127.0.0.1  
IpAddress: 255.0.0.0  
Name: ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask.147.102.13.100  
IpAddress: 255.255.255.0  
Name: ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.127.0.0.1  
INTEGER: 0  
Name: ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.147.102.13.100  
INTEGER: -1821966337  
Name: ip.ipAddrTable.ipAddrEntry.5.127.0.0.1  
INTEGER: -1  
Name: ip.ipAddrTable.ipAddrEntry.5.147.102.13.100  
INTEGER: -1  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteDest.0.0.0.0  
IpAddress: 0.0.0.0  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteDest.127.0.0.1  
IpAddress: 127.0.0.1  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteDest.147.102.13.0  
IpAddress: 147.102.13.0  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteIfIndex.0.0.0.0  
INTEGER: 4  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteIfIndex.127.0.0.1  
INTEGER: 3  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteIfIndex.147.102.13.0  
INTEGER: 4  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric1.0.0.0.0  
INTEGER: -1  
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric1.127.0.0.1

```
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric1.147.102.13.0
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric2.0.0.0.0
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric2.127.0.0.1
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric2.147.102.13.0
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric3.0.0.0.0
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric3.127.0.0.1
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric3.147.102.13.0
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric4.0.0.0.0
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric4.127.0.0.1
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteMetric4.147.102.13.0
INTEGER: -1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.0.0.0.0
IpAddress: 147.102.13.253
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.127.0.0.1
IpAddress: 127.0.0.1
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.147.102.13.0
IpAddress: 147.102.13.100
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteType.0.0.0.0
INTEGER: remote(4)
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteType.127.0.0.1
INTEGER: direct(3)
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteType.147.102.13.0
INTEGER: direct(3)
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteProto.0.0.0.0
INTEGER: local(2)
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteProto.127.0.0.1
INTEGER: local(2)
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteProto.147.102.13.0
INTEGER: local(2)
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteAge.0.0.0.0
INTEGER: 285772
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteAge.127.0.0.1
INTEGER: 285773
Name: ip.ipRoutingTable.ipRouteEntry.ipRouteAge.147.102.13.0
INTEGER: 285773
Name: ip.ipRoutingTable.ipRouteEntry.11.0.0.0.0
IpAddress: 0.0.0.0
Name: ip.ipRoutingTable.ipRouteEntry.11.127.0.0.1
IpAddress: 255.0.0.0
Name: ip.ipRoutingTable.ipRouteEntry.11.147.102.13.0
IpAddress: 255.255.255.0
Name:
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaIfIndex.4.147.102.
13.253
INTEGER: 4
Name:
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaPhysAddress.4.147.
102.13.253
OCTET STRING- (hex): 00 00 0C 01 06 71
Name:
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaNetAddress.4.147.1
02.13.253
IpAddress: 147.102.13.253
Name:
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.4.147.102.13.
253
INTEGER: 3
Name: icmp.icmpInMsgs.0
Counter: 1
Name: icmp.icmpInErrors.0
```

```
Counter: 0
Name: icmp.icmpInDestUnreachs.0
Counter: 1
Name: icmp.icmpInTimeExcds.0
Counter: 0
Name: icmp.icmpInParmProbs.0
Counter: 0
Name: icmp.icmpInSrcQuenches.0
Counter: 0
Name: icmp.icmpInRedirects.0
Counter: 0
Name: icmp.icmpInEchos.0
Counter: 0
Name: icmp.icmpInEchoReps.0
Counter: 0
Name: icmp.icmpInTimestamps.0
Counter: 0
Name: icmp.icmpInTimestampReps.0
Counter: 0
Name: icmp.icmpInAddrMasks.0
Counter: 0
Name: icmp.icmpInAddrMaskReps.0
Counter: 0
Name: icmp.icmpOutMsgs.0
Counter: 11
Name: icmp.icmpOutErrors.0
Counter: 0
Name: icmp.icmpOutDestUnreachs.0
Counter: 11
Name: icmp.icmpOutTimeExcds.0
Counter: 0
Name: icmp.icmpOutParmProbs.0
Counter: 0
Name: icmp.icmpOutSrcQuenches.0
Counter: 0
Name: icmp.icmpOutRedirects.0
Counter: 0
Name: icmp.icmpOutEchos.0
Counter: 0
Name: icmp.icmpOutEchoReps.0
Counter: 0
Name: icmp.icmpOutTimestamps.0
Counter: 0
Name: icmp.icmpOutTimestampReps.0
Counter: 0
Name: icmp.icmpOutAddrMasks.0
Counter: 0
Name: icmp.icmpOutAddrMaskReps.0
Counter: 0
Name: tcp.tcpRtoAlgorithm.0
INTEGER: vanj(4)
Name: tcp.tcpRtoMin.0
INTEGER: 1000
Name: tcp.tcpRtoMax.0
INTEGER: 64000
Name: tcp.tcpMaxConn.0
INTEGER: -1
Name: tcp.tcpActiveOpens.0
Counter: 46
Name: tcp.tcpPassiveOpens.0
Counter: 189
Name: tcp.tcpAttemptFails.0
Counter: 0
Name: tcp.tcpEstabResets.0
Counter: 2
Name: tcp.tcpCurrEstab.0
Gauge: 2
Name: tcp.tcpInSegs.0
Counter: 59711
```

```
Name: tcp.tcpOutSegs.0
Counter: 17958
Name: tcp.tcpRetransSegs.0
Counter: 11
Name: tcp.tcpConnTable.0
Counter: 0
Name: tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.7.0.0.0.0.0
INTEGER: listen(2)
Name: tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.9.0.0.0.0.0
INTEGER: listen(2)
Name: tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.13.0.0.0.0.0
INTEGER: listen(2)
Name: tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.19.0.0.0.0.0
INTEGER: listen(2)
Name: tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.21.0.0.0.0.0
INTEGER: listen(2)
Name: tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.23.0.0.0.0.0
INTEGER: listen(2)
Name: tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.25.0.0.0.0.0
INTEGER: listen(2)
Name: tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.37.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.111.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.512.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.513.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.514.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.1096.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.1260.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.5696.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.5997.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.6000.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.0.0.0.0.6111.0.0.0.0.0
INTEGER: listen(2)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnState.147.102.13.100.1022.147.10
2.1.2.513
INTEGER: established(5)
tcp.tcpConnTable.tcpConnEntry.tcpConnState.147.102.13.100.1023.147.10
2.1.1.513
INTEGER: established(5)
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.7.0.0.0.0.0
IpAddress: 0.0.0.0
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.9.0.0.0.0.0
IpAddress: 0.0.0.0
Name:
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.13.0.0.0.0.0
0
IpAddress: 0.0.0.0
```

---

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.19.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.21.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.23.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.25.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.37.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.111.0.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.512.0.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.513.0.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.514.0.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.1096.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.1260.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.5696.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.5997.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.6000.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.0.0.0.0.6111.0.0.0.  
.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.147.102.13.100.1022  
.1023.147.102.1.2.513  
IpAddress: 147.102.13.100  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalAddress.147.102.13.100.1023  
.147.102.1.1.513  
IpAddress: 147.102.13.100

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.7.0.0.0.0.0  
INTEGER: 7  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.9.0.0.0.0.0  
INTEGER: 9  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.13.0.0.0.0.0  
INTEGER: 13  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.19.0.0.0.0.0  
INTEGER: 19  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.21.0.0.0.0.0  
INTEGER: 21  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.23.0.0.0.0.0  
INTEGER: 23  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.25.0.0.0.0.0  
INTEGER: 25  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.37.0.0.0.0.0  
INTEGER: 37  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.111.0.0.0.0.0  
INTEGER: 111  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.512.0.0.0.0.0  
INTEGER: 512  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.513.0.0.0.0.0  
INTEGER: 513  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.514.0.0.0.0.0  
INTEGER: 514  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.1096.0.0.0.0.0  
INTEGER: 1096  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.1260.0.0.0.0.0  
INTEGER: 1260  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.5696.0.0.0.0.0  
INTEGER: 5696  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.5997.0.0.0.0.0  
INTEGER: 5997  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.6000.0.0.0.0.0  
INTEGER: 6000  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.0.0.0.0.6111.0.0.0.0.0  
INTEGER: 6111  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.147.102.13.100.1023.14  
7.102.1.2.513  
INTEGER: 1022  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnLocalPort.147.102.13.100.1023.14  
7.102.1.1.513  
INTEGER: 1023  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.7.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.9.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.13.0.0.0.0.0  
IpAddress: 0.0.0.0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.19.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.21.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.23.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.25.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.37.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.111.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.512.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.513.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.514.0.0.0.0.0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.1096.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.1260.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.5696.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.5997.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.6000.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.0.0.0.0.6111.0.0.0.0.  
0  
IpAddress: 0.0.0.0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.147.102.13.100.1022.1  
47.102.1.2.513  
IpAddress: 147.102.1.2  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemAddress.147.102.13.100.1023.1  
47.102.1.1.513  
IpAddress: 147.102.1.1  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.7.0.0.0.0.0  
INTEGER: 0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.9.0.0.0.0.0  
INTEGER: 0  
Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.13.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.19.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.21.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.23.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.25.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.37.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.111.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.512.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.513.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.514.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.1096.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.1260.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.5696.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.5997.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.6000.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.0.0.0.0.6111.0.0.0.0.0  
INTEGER: 0

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.147.102.13.100.1022.147.  
102.1.2.513  
INTEGER: 513

Name:  
tcp.tcpConnTable.tcpConnEntry.tcpConnRemPort.147.102.13.100.1023.147.  
102.1.1.513  
INTEGER: 513

Name: tcp.tcpInErrs.0  
Counter: 0

Name: tcp.tcpOutRsts.0  
Counter: 2

Name: udp.udpInDatagrams.0  
Counter: 1866

Name: udp.udpNoPorts.0  
Counter: 9686

Name: udp.udpInErrors.0  
Counter: 0

Name: udp.udpOutDatagrams.0  
Counter: 1876

Name: udp.5.1.1.0.0.0.0.7  
IpAddress: 0.0.0.0

Name: udp.5.1.1.0.0.0.0.9  
IpAddress: 0.0.0.0

Name: udp.5.1.1.0.0.0.0.13

```
IpAddress: 0.0.0.0
Name: udp.5.1.1.0.0.0.0.0.19
IpAddress: 0.0.0.0
Name: udp.5.1.1.0.0.0.0.0.37
IpAddress: 0.0.0.0
Name: udp.5.1.1.0.0.0.0.0.69
IpAddress: 0.0.0.0
Name: udp.5.1.1.0.0.0.0.111
IpAddress: 0.0.0.0
Name: udp.5.1.1.0.0.0.0.161
IpAddress: 0.0.0.0
Name: udp.5.1.1.0.0.0.0.177
IpAddress: 0.0.0.0
Name: udp.5.1.1.0.0.0.0.514
IpAddress: 0.0.0.0
Name: udp.5.1.1.0.0.0.1028
IpAddress: 0.0.0.0
Name: udp.5.1.2.0.0.0.0.7
INTEGER: 7
Name: udp.5.1.2.0.0.0.0.9
INTEGER: 9
Name: udp.5.1.2.0.0.0.0.13
INTEGER: 13
Name: udp.5.1.2.0.0.0.0.19
INTEGER: 19
Name: udp.5.1.2.0.0.0.0.37
INTEGER: 37
Name: udp.5.1.2.0.0.0.0.69
INTEGER: 69
Name: udp.5.1.2.0.0.0.0.111
INTEGER: 111
Name: udp.5.1.2.0.0.0.0.161
INTEGER: 161
Name: udp.5.1.2.0.0.0.0.177
INTEGER: 177
Name: udp.5.1.2.0.0.0.0.514
INTEGER: 514
Name: udp.5.1.2.0.0.0.0.1028
INTEGER: 1028
Name: snmp.snmpInPkts.0
Counter: 694
Name: snmp.snmpOutPkts.0
Counter: 694
Name: snmp.snmpInBadVersions.0
Counter: 0
Name: snmp.snmpInBadCommunityNames.0
Counter: 0
Name: snmp.snmpInBadCommunityUses.0
Counter: 0
Name: snmp.snmpInASNParseErrs.0
Counter: 0
Name: snmp.snmpInBadTypes.0
Counter: 0
Name: snmp.snmpInTooBigs.0
Counter: 0
Name: snmp.snmpInNoSuchNames.0
Counter: 0
Name: snmp.snmpInBadValues.0
Counter: 0
Name: snmp.snmpInReadOnlys.0
Counter: 0
Name: snmp.snmpInGenErrs.0
Counter: 0
Name: snmp.snmpInTotalReqVars.0
Counter: 705
Name: snmp.snmpInTotalSetVars.0
Counter: 0
Name: snmp.snmpInGetRequests.0
Counter: 0
```

```
Name: snmp.snmpInGetNexts.0
Counter: 709
Name: snmp.snmpInSetRequests.0
Counter: 0
Name: snmp.snmpInGetResponses.0
Counter: 0
Name: snmp.snmpInTraps.0
Counter: 0
Name: snmp.snmpOutTooBigs.0
Counter: 0
Name: snmp.snmpOutNoSuchNames.0
Counter: 0
Name: snmp.snmpOutBadValues.0
Counter: 0
Name: snmp.snmpOutReadOnlys.0
Counter: 0
Name: snmp.snmpOutGenErrs.0
Counter: 0
Name: snmp.snmpOutGetRequests.0
Counter: 0
Name: snmp.snmpOutGetNexts.0
Counter: 0
Name: snmp.snmpOutSetRequests.0
Counter: 0
Name: snmp.snmpOutGetResponses.0
Counter: 0
Name: snmp.snmpOutTraps.0
Counter: 0
Name: snmp.snmpEnableAuthTraps.0
INTEGER : 1
```

## B.2 ΟΡΙΣΜΟΣ MIB-II (RFC1213)

Κατωτέρω παρουσιάζεται το πλήρες κείμενο του RFC1213 που ορίζει το αρχείο MIB-II του SNMP. Η αντιστοίχιση των μεταβλητών του Παραδείγματος στο Μέρος B1 είναι ο καλύτερος τρόπος για τη κατανόηση των σχετικών εννοιών.

```
-- groups in MIB-II

system      OBJECT IDENTIFIER ::= { mib-2 1 }
interfaces   OBJECT IDENTIFIER ::= { mib-2 2 }
at          OBJECT IDENTIFIER ::= { mib-2 3 }
ip          OBJECT IDENTIFIER ::= { mib-2 4 }
icmp         OBJECT IDENTIFIER ::= { mib-2 5 }
tcp          OBJECT IDENTIFIER ::= { mib-2 6 }
udp          OBJECT IDENTIFIER ::= { mib-2 7 }
egp          OBJECT IDENTIFIER ::= { mib-2 8 }

-- historical (some say hysterical)
-- cmot        OBJECT IDENTIFIER ::= { mib-2 9 }

transmission OBJECT IDENTIFIER ::= { mib-2 10 }
snmp         OBJECT IDENTIFIER ::= { mib-2 11 }

-- the System group

-- Implementation of the System group is mandatory for all
-- systems. If an agent is not configured to have a value
-- for any of these variables, a string of length 0 is
-- returned.

sysDescr    OBJECT-TYPE
            SYNTAX  DisplayString (SIZE (0..255))
            ACCESS  read-only
            STATUS  mandatory
            DESCRIPTION
                    "A textual description of the entity. This
value
                    should include the full name and version
                    identification of the system's hardware type,
                    software operating-system, and networking
                    software. It is mandatory that this only
contain
                    printable ASCII characters."
            ::= { system 1 }

sysObjectID OBJECT-TYPE
            SYNTAX  OBJECT IDENTIFIER
            ACCESS  read-only
            STATUS  mandatory
            DESCRIPTION
                    "The vendor's authoritative identification of
the
                    network management subsystem contained in the
                    entity. This value is allocated within the SMI
```

enterprises subtree (1.3.6.1.4.1) and provides  
 an easy and unambiguous means for determining  
 `what kind of box' is being managed. For example, if  
 vendor 'Flintstones, Inc.' was assigned the  
 subtree 1.3.6.1.4.1.4242, it could assign the  
 identifier 1.3.6.1.4.1.4242.1.1 to its 'Fred  
 Router'."  
`::= { system 2 }`

**sysUpTime** OBJECT-TYPE  
 SYNTAX TimeTicks  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "The time (in hundredths of a second) since the  
 last network management portion of the system was  
 re-initialized."  
`::= { system 3 }`

**sysContact** OBJECT-TYPE  
 SYNTAX DisplayString (SIZE (0..255))  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 "The textual identification of the contact  
 person for this managed node, together with  
 information on how to contact this person."  
`::= { system 4 }`

**sysName** OBJECT-TYPE  
 SYNTAX DisplayString (SIZE (0..255))  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 "An administratively-assigned name for this  
 managed node. By convention, this is the  
 node's fully-qualified domain name."  
`::= { system 5 }`

**sysLocation** OBJECT-TYPE  
 SYNTAX DisplayString (SIZE (0..255))  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 "The physical location of this node (e.g.,  
 'telephone closet, 3rd floor')."  
`::= { system 6 }`

**sysServices** OBJECT-TYPE  
 SYNTAX INTEGER (0..127)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "A value which indicates the set of services  
 that this entity primarily offers.  
 The value is a sum. This sum initially takes  
 the value zero. Then, for each layer, L, in the  
 range 1 through 7, that this node performs  
 transactions  
 For

for, 2 raised to (L - 1) is added to the sum.

routing example, a node which performs primarily functions would have a value of 4 ( $2^{(3-1)}$ ). In contrast, a node which is a host offering application services would have a value of 72 ( $2^{(4-1)} + 2^{(7-1)}$ ). Note that in the context of the Internet suite of protocols, values should be calculated accordingly:

```
layer functionality
 1 physical (e.g., repeaters)
 2 datalink/subnetwork (e.g., bridges)
 3 internet (e.g., IP gateways)
 4 end-to-end (e.g., IP hosts)
 7 applications (e.g., mail relays)
```

and For systems including OSI protocols, layers 5 and 6 may also be counted."

```
::= { system 7 }
```

-- the Interfaces group

-- Implementation of the Interfaces group is mandatory for -- all systems.

```
ifNumber OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
    "The number of network interfaces (regardless
of
    their current state) present on this system."
::= { interfaces 1 }
```

-- the Interfaces table

entity's -- The Interfaces table contains information on the -- interfaces. Each interface is thought of as being -- attached to a 'subnetwork'. Note that this term should -- not be confused with 'subnet' which refers to an -- addressing partitioning scheme used in the Internet suite -- of protocols.

```
ifTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF IfEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
    "A list of interface entries. The number of
     entries is given by the value of ifNumber."
::= { interfaces 2 }
```

```
ifEntry OBJECT-TYPE
  SYNTAX  IfEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
    "An interface entry containing objects at the
     subnetwork layer and below for a particular
     interface."
INDEX   { ifIndex }
```

```

 ::= { ifTable 1 }

IfEntry ::= 
SEQUENCE {
    ifIndex
        INTEGER,
    ifDescr
        DisplayString,
    ifType
        INTEGER,
    ifMtu
        INTEGER,
    ifSpeed
        Gauge,
    ifPhysAddress
        PhysAddress,
    ifAdminStatus
        INTEGER,
    ifOperStatus
        INTEGER,
    ifLastChange
        TimeTicks,
    ifInOctets
        Counter,
    ifInUcastPkts
        Counter,
    ifInNUcastPkts
        Counter,
    ifInDiscards
        Counter,
    ifInErrors
        Counter,
    ifInUnknownProtos
        Counter,
    ifOutOctets
        Counter,
    ifOutUcastPkts
        Counter,
    ifOutNUcastPkts
        Counter,
    ifOutDiscards
        Counter,
    ifOutErrors
        Counter,
    ifOutQLen
        Gauge,
    ifSpecific
        OBJECT IDENTIFIER
}

ifIndex OBJECT-TYPE
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each interface. Its value
     ranges between 1 and the value of ifNumber.
The
at
entity's
value for each interface must remain constant
least from one re-initialization of the
network management system to the next re-
initialization."
 ::= { ifEntry 1 }

ifDescr OBJECT-TYPE
SYNTAX  DisplayString (SIZE (0..255))
ACCESS  read-only
STATUS  mandatory

```

---

the  
of  
version

DESCRIPTION  
"A textual string containing information about  
interface. This string should include the name  
the manufacturer, the product name and the  
of the hardware interface."  
 ::= { ifEntry 2 }

ifType OBJECT-TYPE  
SYNTAX INTEGER {  
other(1), -- none of the following  
regular1822(2),  
hdh1822(3),  
ddn-x25(4),  
rfc877-x25(5),  
ethernet-csmacd(6),  
iso88023-csmacd(7),  
iso88024-tokenBus(8),  
iso88025-tokenRing(9),  
iso88026-man(10),  
starLan(11),  
proteon-10Mbit(12),  
proteon-80Mbit(13),  
hyperchannel(14),  
fddi(15),  
lapb(16),  
sdlc(17),  
ds1(18), -- T-1  
e1(19), -- european equiv. of  
T-1  
basicISDN(20),  
primaryISDN(21), -- proprietary serial  
propPointToPointSerial(22),  
ppp(23),  
softwareLoopback(24),  
eon(25), -- CLNP over IP [11]  
ethernet-3Mbit(26),  
nsip(27), -- XNS over IP  
slip(28), -- generic SLIP  
ultra(29), -- ULTRA technologies  
ds3(30), -- T-3  
sip(31), -- SMDS  
frame-relay(32)  
}  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The type of interface, distinguished according  
to  
the physical/link protocol(s) immediately  
'below'  
the network layer in the protocol stack."  
 ::= { ifEntry 3 }

ifMtu OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The size of the largest datagram which can be  
sent/received on the interface, specified in  
octets. For interfaces that are used for  
transmitting network datagrams, this is the  
size  
sent  
of the largest network datagram that can be  
on the interface."

```

        ::= { ifEntry 4 }

ifSpeed OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "An estimate of the interface's current
bandwidth
in bits per second. For interfaces which do
not
vary in bandwidth or for those where no
accurate
estimation can be made, this object should
contain
the nominal bandwidth."
        ::= { ifEntry 5 }

ifPhysAddress OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The interface's address at the protocol layer
immediately `below' the network layer in the
protocol stack. For interfaces which do not
have
such an address (e.g., a serial line), this
object
should contain an octet string of zero length."
        ::= { ifEntry 6 }

ifAdminStatus OBJECT-TYPE
    SYNTAX INTEGER {
        up(1),           -- ready to pass packets
        down(2),
        testing(3)      -- in some test mode
    }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The desired state of the interface. The
testing(3) state indicates that no operational
packets can be passed."
        ::= { ifEntry 7 }

ifOperStatus OBJECT-TYPE
    SYNTAX INTEGER {
        up(1),           -- ready to pass packets
        down(2),
        testing(3)      -- in some test mode
    }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The current operational state of the
interface.
The testing(3) state indicates that no
operational
packets can be passed."
        ::= { ifEntry 8 }

ifLastChange OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The value of sysUpTime at the time the
interface
entered its current operational state. If the

```

```
current state was entered prior to the last re-
initialization of the local network management
subsystem, then this object contains a zero
value."
 ::= { ifEntry 9 }

ifInOctets OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of octets received on the
         interface, including framing characters."
 ::= { ifEntry 10 }

ifInUcastPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of subnetwork-unicast packets
         delivered to a higher-layer protocol."
 ::= { ifEntry 11 }

ifInNUcastPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of non-unicast (i.e., subnetwork-
         broadcast or subnetwork-multicast) packets
         delivered to a higher-layer protocol."
 ::= { ifEntry 12 }

ifInDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of inbound packets which were
chosen
        to be discarded even though no errors had been
detected to prevent their being deliverable to
a
        higher-layer protocol. One possible reason for
discarding such a packet could be to free up
buffer space."
 ::= { ifEntry 13 }

ifInErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of inbound packets that contained
errors preventing them from being deliverable
to a
        higher-layer protocol."
 ::= { ifEntry 14 }

ifInUnknownProtos OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of packets received via the
interface
        which were discarded because of an unknown or
unsupported protocol."
 ::= { ifEntry 15 }
```

```

ifOutOctets OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of octets transmitted out of
the
    interface, including framing characters."
 ::= { ifEntry 16 }

ifOutUcastPkts OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of packets that higher-level
    protocols requested be transmitted to a
    subnetwork-unicast address, including those
that
    were discarded or not sent."
 ::= { ifEntry 17 }

ifOutNUcastPkts OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of packets that higher-level
    protocols requested be transmitted to a non-
    unicast (i.e., a subnetwork-broadcast or
    subnetwork-multicast) address, including those
    that were discarded or not sent."
 ::= { ifEntry 18 }

ifOutDiscards OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The number of outbound packets which were
chosen
    to be discarded even though no errors had been
    detected to prevent their being transmitted.
One
    possible reason for discarding such a packet
could
    be to free up buffer space."
 ::= { ifEntry 19 }

ifOutErrors OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The number of outbound packets that could not
be
    transmitted because of errors."
 ::= { ifEntry 20 }

ifOutQLen OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The length of the output packet queue (in
    packets)."
 ::= { ifEntry 21 }

ifSpecific OBJECT-TYPE

```

```
SYNTAX OBJECT IDENTIFIER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "A reference to MIB definitions specific to the
particular media being used to realize the
interface. For example, if the interface is
realized by an ethernet, then the value of this
object refers to a document defining objects
specific to ethernet. If this information is
not
present, its value should be set to the OBJECT
IDENTIFIER { 0 0 }, which is a syntactically
valid
object identifier, and any conformant
implementation of ASN.1 and BER must be able to
generate and recognize this value."
 ::= { ifEntry 22 }

-- the Address Translation group

-- Implementation of the Address Translation group is
-- mandatory for all systems. Note however that this group
-- is deprecated by MIB-II. That is, it is being included
-- solely for compatibility with MIB-I nodes, and will most
-- likely be excluded from MIB-III nodes. From MIB-II and
-- onwards, each network protocol group contains its own
-- address translation tables.

-- The Address Translation group contains one table which
is
tables
-- the union across all interfaces of the translation
into
term,
tables
-- for converting a NetworkAddress (e.g., an IP address)
-- a subnetwork-specific address. For lack of a better
-- this document refers to such a subnetwork-specific
-- as a 'physical' address.

-- Examples of such translation tables are: for broadcast
-- media where ARP is in use, the translation table is
-- equivalent to the ARP cache; or, on an X.25 network
where
-- non-algorithmic translation to X.121 addresses is
-- required, the translation table contains the
-- NetworkAddress to X.121 address equivalences.

atTable OBJECT-TYPE
SYNTAX SEQUENCE OF AtEntry
ACCESS not-accessible
STATUS deprecated
DESCRIPTION
    "The Address Translation tables contain the
NetworkAddress to 'physical' address
equivalences.
for
DDN-X.25
are
table
Some interfaces do not use translation tables
determining address equivalences (e.g.,
has an algorithmic method); if all interfaces
of this type, then the Address Translation
is empty, i.e., has zero entries."
 ::= { at 1 }
```

```

atEntry OBJECT-TYPE
  SYNTAX  AtEntry
  ACCESS  not-accessible
  STATUS  deprecated
  DESCRIPTION
    "Each entry contains one NetworkAddress to
     `physical' address equivalence."
  INDEX   { atIfIndex,
            atNetAddress }
  ::= { atTable 1 }

AtEntry ::= 
SEQUENCE {
  atIfIndex
    INTEGER,
  atPhysAddress
    PhysAddress,
  atNetAddress
    NetworkAddress
}

atIfIndex OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  deprecated
  DESCRIPTION
    "The interface on which this entry's
equivalence
is effective. The interface identified by a
particular value of this index is the same
interface as identified by the same value of
ifIndex."
  ::= { atEntry 1 }

atPhysAddress OBJECT-TYPE
  SYNTAX  PhysAddress
  ACCESS  read-write
  STATUS  deprecated
  DESCRIPTION
    "The media-dependent `physical' address.

zero
Setting this object to a null string (one of
length) has the effect of invalidating the
corresponding entry in the atTable object.

That
is, it effectively dissasociates the interface
identified with said entry from the mapping
identified with said entry. It is an
implementation-specific matter as to whether
the
agent removes an invalidated entry from the
table.

prepared
Accordingly, management stations must be
to receive tabular information from agents that
corresponds to entries not currently in use.
Proper interpretation of such entries requires
examination of the relevant atPhysAddress
object."
  ::= { atEntry 2 }

atNetAddress OBJECT-TYPE
  SYNTAX  NetworkAddress
  ACCESS  read-write
  STATUS  deprecated
  DESCRIPTION
    "The NetworkAddress (e.g., the IP address)
corresponding to the media-dependent `physical'
address."

```

```
 ::= { atEntry 3 }

-- the IP group

-- Implementation of the IP group is mandatory for all
-- systems.

ipForwarding OBJECT-TYPE
    SYNTAX  INTEGER {
                forwarding(1), -- acting as a gateway
                not-forwarding(2) -- NOT acting as a
gateway
                }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The indication of whether this entity is
        as an IP gateway in respect to the forwarding
        datagrams received by, but not addressed to,
        entity. IP gateways forward datagrams. IP
        do not (except those source-routed via the
        host).
        Note that for some managed nodes, this object
        may
        take on only a subset of the values possible.
        Accordingly, it is appropriate for an agent to
        return a `badValue' response if a management
        station attempts to change this object to an
        inappropriate value."
    ::= { ip 1 }

ipDefaultTTL OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The default value inserted into the
        field of the IP header of datagrams originated
        this entity, whenever a TTL value is not
        supplied
        by the transport layer protocol."
    ::= { ip 2 }

ipInReceives OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of input datagrams received
from
        interfaces, including those received in error."
    ::= { ip 3 }

ipInHdrErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of input datagrams discarded due to
        errors in their IP headers, including bad
        checksums, version number mismatch, other
format
```

```

errors,      time-to-live      exceeded,      errors
discovered
          in processing their IP options, etc."
 ::= { ip 4 }

ipInAddrErrors OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
          "The number of input datagrams discarded
because
          the IP address in their IP header's destination
          field was not a valid address to be received at
          this entity. This count includes invalid
          addresses (e.g., 0.0.0.0) and addresses of
          unsupported Classes (e.g., Class E). For
entities
          which are not IP Gateways and therefore do not
forward datagrams, this counter includes
datagrams
          discarded because the destination address was
not
          a local address."
 ::= { ip 5 }

ipForwDatagrams OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
          "The number of input datagrams for which this
entity was not their final IP destination, as a
result of which an attempt was made to find a
route to forward them to that final
destination.
          In entities which do not act as IP Gateways,
this
          counter will include only those packets which
were
          Source-Routed via this entity, and the Source-
Route option processing was successful."
 ::= { ip 6 }

ipInUnknownProtos OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
          "The number of locally-addressed datagrams
received successfully but discarded because of
an
          unknown or unsupported protocol."
 ::= { ip 7 }

ipInDiscards OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
          "The number of input IP datagrams for which no
problems were encountered to prevent their
continued processing, but which were discarded
(e.g., for lack of buffer space). Note that
this
          counter does not include any datagrams
discarded
          while awaiting re-assembly."
 ::= { ip 8 }

```

---

```
ipInDelivers OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of input datagrams
successfully
delivered to IP user-protocols (including
ICMP)."
 ::= { ip 9 }

ipOutRequests OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of IP datagrams which local
IP
in
counter
user-protocols (including ICMP) supplied to IP
requests for transmission. Note that this
does not include any datagrams counted in
ipForwDatagrams."
 ::= { ip 10 }

ipOutDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of output IP datagrams for which no
problem was encountered to prevent their
transmission to their destination, but which
were
discarded (e.g., for lack of buffer space).
Note
that this counter would include datagrams
counted
in ipForwDatagrams if any such packets met this
(discretionary) discard criterion."
 ::= { ip 11 }

ipOutNoRoutes OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of IP datagrams discarded because
no
route could be found to transmit them to their
any
destination. Note that this counter includes
this
packets counted in ipForwDatagrams which meet
any
`no-route' criterion. Note that this includes
any
datagarms which a host cannot route because all
of
its default gateways are down."
 ::= { ip 12 }

ipReasmTimeout OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The maximum number of seconds which received
fragments are held while they are awaiting
```

```

        reassembly at this entity."
 ::= { ip 13 }

ipReasmReqds OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of IP fragments received which
needed
        to be reassembled at this entity."
 ::= { ip 14 }

ipReasmOKs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of IP datagrams successfully re-
        assembled."
 ::= { ip 15 }

ipReasmFails OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of failures detected by the IP re-
        assembly algorithm (for whatever reason: timed
        out, errors, etc). Note that this is not
        necessarily a count of discarded IP fragments
        since some algorithms (notably the algorithm in
        RFC 815) can lose track of the number of
fragments
        by combining them as they are received."
 ::= { ip 16 }

ipFragOKs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of IP datagrams that have been
        successfully fragmented at this entity."
 ::= { ip 17 }

ipFragFails OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of IP datagrams that have been
        discarded because they needed to be fragmented
at
        this entity but could not be, e.g., because
their
        Don't Fragment flag was set."
 ::= { ip 18 }

ipFragCreates OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of IP datagram fragments that have
        been generated as a result of fragmentation at
        this entity."
 ::= { ip 19 }

-- the IP address table

```

```
-- The IP address table contains this entity's IP
addressing
-- information.

ipAddrTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IpAddrEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "The table of addressing information relevant
to
        this entity's IP addresses."
 ::= { ip 20 }

ipAddrEntry OBJECT-TYPE
    SYNTAX  IpAddrEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "The addressing information for one of this
entity's IP addresses."
    INDEX   { ipAdEntAddr }
 ::= { ipAddrTable 1 }

IpAddrEntry ::=

SEQUENCE {
    ipAdEntAddr
        InetAddress,
    ipAdEntIfIndex
        INTEGER,
    ipAdEntNetMask
        InetAddress,
    ipAdEntBcastAddr
        INTEGER,
    ipAdEntReasmMaxSize
        INTEGER (0..65535)
}

ipAdEntAddr OBJECT-TYPE
    SYNTAX  InetAddress
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The IP address to which this entry's
addressing
        information pertains."
 ::= { ipAddrEntry 1 }

ipAdEntIfIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The index value which uniquely identifies the
interface to which this entry is applicable.
The
        interface identified by a particular value of
this
        index is the same interface as identified by
the
        same value of ifIndex."
 ::= { ipAddrEntry 2 }

ipAdEntNetMask OBJECT-TYPE
    SYNTAX  InetAddress
    ACCESS  read-only
    STATUS  mandatory
```

```

        DESCRIPTION
of           "The subnet mask associated with the IP address
           this entry. The value of the mask is an IP
           address with all the network bits set to 1 and
all           the hosts bits set to 0."
           ::= { ipAddrEntry 3 }

ipAdEntBcastAddr OBJECT-TYPE
SYNTAX  INTEGER
ACCESS  read-only
STATUS   mandatory
DESCRIPTION
IP           "The value of the least-significant bit in the
           broadcast address used for sending datagrams on
           the (logical) interface associated with the IP
           address of this entry. For example, when the
           Internet standard all-ones broadcast address is
           used, the value will be 1. This value applies
to           both the subnet and network broadcasts
addresses
           used by the entity on this (logical)
interface."           ::= { ipAddrEntry 4 }

ipAdEntReasmMaxSize OBJECT-TYPE
SYNTAX  INTEGER (0..65535)
ACCESS  read-only
STATUS   mandatory
DESCRIPTION
fragmented      "The size of the largest IP datagram which this
           entity can re-assemble from incoming IP
           datagrams received on this interface."
           ::= { ipAddrEntry 5 }

-- the IP routing table

-- The IP routing table contains an entry for each route
-- presently known to this entity.

ipRouteTable OBJECT-TYPE
SYNTAX  SEQUENCE OF IpRouteEntry
ACCESS  not-accessible
STATUS   mandatory
DESCRIPTION
           "This entity's IP Routing table."
           ::= { ip 21 }

ipRouteEntry OBJECT-TYPE
SYNTAX  IpRouteEntry
ACCESS  not-accessible
STATUS   mandatory
DESCRIPTION
           "A route to a particular destination."
INDEX    { ipRouteDest }
           ::= { ipRouteTable 1 }

IpRouteEntry ::==
SEQUENCE {
    ipRouteDest
        InetAddress,
    ipRouteIfIndex
        INTEGER,
    ipRouteMetric1
        INTEGER,
}

```

```
        ipRouteMetric2
            INTEGER,
        ipRouteMetric3
            INTEGER,
        ipRouteMetric4
            INTEGER,
        ipRouteNextHop
            IpAddress,
        ipRouteType
            INTEGER,
        ipRouteProto
            INTEGER,
        ipRouteAge
            INTEGER,
        ipRouteMask
            IPAddress,
        ipRouteMetric5
            INTEGER,
        ipRouteInfo
            OBJECT IDENTIFIER
    }

ipRouteDest OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-write
    STATUS   mandatory
    DESCRIPTION
        "The destination IP address of this route. An
         entry with a value of 0.0.0.0 is considered a
         default route. Multiple routes to a single
         destination can appear in the table, but access
to
table-
such multiple entries is dependent on the
access mechanisms defined by the network
management protocol in use."
 ::= { ipRouteEntry 1 }

ipRouteIfIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS   mandatory
    DESCRIPTION
        "The index value which uniquely identifies the
         local interface through which the next hop of
this
identified
route should be reached. The interface
by a particular value of this index is the same
interface as identified by the same value of
ifIndex."
 ::= { ipRouteEntry 2 }

ipRouteMetric1 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS   mandatory
    DESCRIPTION
        "The primary routing metric for this route.
The
semantics of this metric are determined by the
routing-protocol specified in the route's
ipRouteProto value. If this metric is not
used,
its value should be set to -1."
 ::= { ipRouteEntry 3 }

ipRouteMetric2 OBJECT-TYPE
    SYNTAX  INTEGER
```

```

        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                  "An alternate routing metric for this route.
The
semantics of this metric are determined by the
routing-protocol specified in the route's
ipRouteProto value. If this metric is not
used,
its value should be set to -1."
 ::= { ipRouteEntry 4 }

ipRouteMetric3 OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
    "An alternate routing metric for this route.
The
semantics of this metric are determined by the
routing-protocol specified in the route's
ipRouteProto value. If this metric is not
used,
its value should be set to -1."
 ::= { ipRouteEntry 5 }

ipRouteMetric4 OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
    "An alternate routing metric for this route.
The
semantics of this metric are determined by the
routing-protocol specified in the route's
ipRouteProto value. If this metric is not
used,
its value should be set to -1."
 ::= { ipRouteEntry 6 }

ipRouteNextHop OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
    "The IP address of the next hop of this route.
    (In the case of a route bound to an interface
    which is realized via a broadcast media, the
value
of this field is the agent's IP address on that
interface.)"
 ::= { ipRouteEntry 7 }

ipRouteType OBJECT-TYPE
  SYNTAX  INTEGER {
            other(1),          -- none of the following
            invalid(2),         -- an invalidated route
            direct(3),          -- route to directly
                                -- connected (sub-)network
                                -- route to a
non-local
            indirect(4)          --
host/network/sub-network
          }
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
    "The type of route. Note that the values

```

of direct(3) and indirect(4) refer to the notion direct and indirect routing in the IP architecture. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipRouteTable object. That is, it effectively disassociates the destination identified with said entry from the route identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipRouteType object."

```
 ::= { ipRouteEntry 8 }

ipRouteProto OBJECT-TYPE
    SYNTAX  INTEGER {
                other(1),          -- none of the following
                --                      non-protocol
                --          e.g.,      manually
                local(2),           -- entries
                -- set via a network
                netmgmt(3),         -- management protocol
                -- obtained via ICMP,
                icmp(4),            -- e.g., Redirect
                -- the remaining values are
                -- all gateway routing
                -- protocols
                egp(5),
                ggp(6),
                hello(7),
                rip(8),
                is-is(9),
                es-is(10),
                ciscoIgrp(11),
                bbnSpfIgp(12),
                ospf(13),
                bgp(14)
            }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The routing mechanism via which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols."
 ::= { ipRouteEntry 9 }

ipRouteAge OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
```

last  
implied  
protocol

"The number of seconds since this route was updated or otherwise determined to be correct. Note that no semantics of 'too old' can be except through knowledge of the routing by which the route was learned."  
 $::= \{ \text{ipRouteEntry} \ 10 \ }$

the  
systems  
by  
correspondent

ipRouteMask OBJECT-TYPE  
SYNTAX IpAddress  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
"Indicate the mask to be logical-ANDED with the destination address before being compared to value in the ipRouteDest field. For those that do not support arbitrary subnet masks, an agent constructs the value of the ipRouteMask determining whether the value of the ipRouteDest field belong to a class-A, B, or C network, and then using one of:

mask	network
255.0.0.0	class-A
255.255.0.0	class-B
255.255.255.0	class-C

If the value of the ipRouteDest is 0.0.0.0 (a default route), then the mask value is also 0.0.0.0. It should be noted that all IP routing subsystems implicitly use this mechanism."  
 $::= \{ \text{ipRouteEntry} \ 11 \ }$

The  
used,  
responsible  
If  
should  
is

ipRouteMetric5 OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
"An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1."  
 $::= \{ \text{ipRouteEntry} \ 12 \ }$

ipRouteInfo OBJECT-TYPE  
SYNTAX OBJECT IDENTIFIER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"A reference to MIB definitions specific to the particular routing protocol which is responsible for this route, as determined by the value specified in the route's ipRouteProto value. If this information is not present, its value be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier, and any

be conformant implementation of ASN.1 and BER must able to generate and recognize this value."

::= { ipRouteEntry 13 }

-- the IP Address Translation table

-- The IP address translation table contain the IpAddress to -- `physical' address equivalences. Some interfaces do not -- use translation tables for determining address -- equivalences (e.g., DDN-X.25 has an algorithmic method); -- if all interfaces are of this type, then the Address -- Translation table is empty, i.e., has zero entries.

ipNetToMediaTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpNetToMediaEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION "The IP Address Translation table used for mapping from IP addresses to physical addresses."

::= { ip 22 }

ipNetToMediaEntry OBJECT-TYPE

SYNTAX IpNetToMediaEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION "Each entry contains one IpAddress to `physical' address equivalence."

INDEX { ipNetToMediaIfIndex, ipNetToMediaNetAddress }

::= { ipNetToMediaTable 1 }

IpNetToMediaEntry ::=

SEQUENCE {

ipNetToMediaIfIndex INTEGER,

ipNetToMediaPhysAddress PhysAddress,

ipNetToMediaNetAddress IPAddress,

ipNetToMediaType INTEGER

}

ipNetToMediaIfIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION "The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex."

::= { ipNetToMediaEntry 1 }

ipNetToMediaPhysAddress OBJECT-TYPE

SYNTAX PhysAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION "The media-dependent `physical' address."

```

 ::= { ipNetToMediaEntry 2 }

 ipNetToMediaNetAddress OBJECT-TYPE
   SYNTAX  IpAddress
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
     "The IPAddress corresponding to the media-
      dependent `physical' address."
 ::= { ipNetToMediaEntry 3 }

 ipNetToMediaType OBJECT-TYPE
   SYNTAX  INTEGER {
     other(1),           -- none of the following
     invalid(2),         -- an invalidated mapping
     dynamic(3),
     static(4)
   }
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
     "The type of mapping.

Setting this object to the value invalid(2) has
the effect of invalidating the corresponding
entry
effectively
said
entry.

It is an implementation-specific matter as to
whether the agent removes an invalidated entry
from the table. Accordingly, management
stations
such
entries requires examination of the relevant
ipNetToMediaType object."
 ::= { ipNetToMediaEntry 4 }

-- additional IP objects

 ipRoutingDiscards OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
     "The number of routing entries which were
      chosen
      One
      could
      to be discarded even though they are valid.
      possible reason for discarding such an entry
      be to free-up buffer space for other routing
      entries."
 ::= { ip 23 }

-- the ICMP group

-- Implementation of the ICMP group is mandatory for all
-- systems.

 icmpInMsgs OBJECT-TYPE
   SYNTAX  Counter

```

---

```
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The total number of ICMP messages which the
entity received. Note that this counter
includes
        all those counted by icmpInErrors."
::= { icmp 1 }

icmpInErrors OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of ICMP messages which the entity
received but determined as having ICMP-specific
errors (bad ICMP checksums, bad length, etc.)."
::= { icmp 2 }

icmpInDestUnreachs OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of ICMP Destination Unreachable
messages received."
::= { icmp 3 }

icmpInTimeExcds OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of ICMP Time Exceeded messages
received."
::= { icmp 4 }

icmpInParmProbs OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of ICMP Parameter Problem messages
received."
::= { icmp 5 }

icmpInSrcQuenches OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of ICMP Source Quench messages
received."
::= { icmp 6 }

icmpInRedirects OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of ICMP Redirect messages
received."
::= { icmp 7 }

icmpInEchos OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
```

```

DESCRIPTION
    "The number of ICMP Echo (request) messages
     received."
 ::= { icmp 8 }

icmpInEchoReps OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of ICMP Echo Reply messages
received."
 ::= { icmp 9 }

icmpInTimestamps OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of ICMP Timestamp (request)
messages
     received."
 ::= { icmp 10 }

icmpInTimestampReps OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of ICMP Timestamp Reply messages
received."
 ::= { icmp 11 }

icmpInAddrMasks OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of ICMP Address Mask Request
messages
     received."
 ::= { icmp 12 }

icmpInAddrMaskReps OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of ICMP Address Mask Reply messages
received."
 ::= { icmp 13 }

icmpOutMsgs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of ICMP messages which this
entity attempted to send. Note that this
counter
     includes all those counted by icmpOutErrors."
 ::= { icmp 14 }

icmpOutErrors OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of ICMP messages which this entity
did

```

not send due to problems discovered within ICMP such as a lack of buffers. This value should include errors discovered outside the ICMP such as the inability of IP to route the datagram. In some implementations there may be types of error which contribute to this counter's value."  
 ::= { icmp 15 }

icmpOutDestUnreachs OBJECT-TYPE  
 SYNTAX Counter  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "The number of ICMP Destination Unreachable messages sent."  
 ::= { icmp 16 }

icmpOutTimeExcds OBJECT-TYPE  
 SYNTAX Counter  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "The number of ICMP Time Exceeded messages sent."  
 ::= { icmp 17 }

icmpOutParmProbs OBJECT-TYPE  
 SYNTAX Counter  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "The number of ICMP Parameter Problem messages sent."  
 ::= { icmp 18 }

icmpOutSrcQuenches OBJECT-TYPE  
 SYNTAX Counter  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "The number of ICMP Source Quench messages sent."  
 ::= { icmp 19 }

icmpOutRedirects OBJECT-TYPE  
 SYNTAX Counter  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "The number of ICMP Redirect messages sent.  
For a host, this object will always be zero, since hosts do not send redirects."  
 ::= { icmp 20 }

icmpOutEchos OBJECT-TYPE  
 SYNTAX Counter  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "The number of ICMP Echo (request) messages sent."

```

        ::= { icmp 21 }

icmpOutEchoReps OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of ICMP Echo Reply messages sent."
    ::= { icmp 22 }

icmpOutTimestamps OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of ICMP Timestamp (request)
messages
sent."
    ::= { icmp 23 }

icmpOutTimestampReps OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of ICMP Timestamp Reply messages
sent."
    ::= { icmp 24 }

icmpOutAddrMasks OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of ICMP Address Mask Request
messages
sent."
    ::= { icmp 25 }

icmpOutAddrMaskReps OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of ICMP Address Mask Reply messages
sent."
    ::= { icmp 26 }

-- the TCP group

-- Implementation of the TCP group is mandatory for all
-- systems that implement the TCP.

-- Note that instances of object types that represent
-- information about a particular TCP connection are
-- transient; they persist only as long as the connection
-- in question.

tcpRtoAlgorithm OBJECT-TYPE
    SYNTAX INTEGER {
        other(1),      -- none of the following
        constant(2),   -- a constant rto
        rsre(3),       -- MIL-STD-1778, Appendix B
        vanj(4)        -- Van Jacobson's algorithm
    }
    ACCESS read-only

```

[10]

---

```
        STATUS mandatory
        DESCRIPTION
            "The algorithm used to determine the timeout
value
            used for retransmitting unacknowledged octets."
::= { tcp 1 }

tcpRtoMin OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The minimum value permitted by a TCP
semantics
        implementation for the retransmission timeout,
        measured in milliseconds. More refined
algorithm
        for objects of this type depend upon the
        used to determine the retransmission timeout.
In
        In particular, when the timeout algorithm is
rsre(3),
        an object of this type has the semantics of the
        LBOUND quantity described in RFC 793."
::= { tcp 2 }

tcpRtoMax OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The maximum value permitted by a TCP
semantics
        implementation for the retransmission timeout,
        measured in milliseconds. More refined
algorithm
        for objects of this type depend upon the
        used to determine the retransmission timeout.
In
        In particular, when the timeout algorithm is
rsre(3),
        an object of this type has the semantics of the
        UBOUND quantity described in RFC 793."
::= { tcp 3 }

tcpMaxConn OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The limit on the total number of TCP
connections
        the entity can support. In entities where the
        maximum number of connections is dynamic, this
        object should contain the value -1."
::= { tcp 4 }

tcpActiveOpens OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of times TCP connections have made
a
        direct transition to the SYN-SENT state from
the
        CLOSED state."
::= { tcp 5 }
```

```

tcpPassiveOpens OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times TCP connections have made
a           direct transition to the SYN-RCVD state from
the           LISTEN state."
::= { tcp 6 }

tcpAttemptFails OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times TCP connections have made
a           direct transition to the CLOSED state from
either           the SYN-SENT state or the SYN-RCVD state, plus
the           number of times TCP connections have made a
direct           transition to the LISTEN state from the
SYN-RCVD           state."
::= { tcp 7 }

tcpEstabResets OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times TCP connections have made
a           direct transition to the CLOSED state from
either           the ESTABLISHED state or the CLOSE-WAIT state."
::= { tcp 8 }

tcpCurrEstab OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of TCP connections for which the
current state is either ESTABLISHED or CLOSE-
WAIT."
::= { tcp 9 }

tcpInSegs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of segments received,
including           those received in error. This count includes
                           segments received on currently established
                           connections."
::= { tcp 10 }

tcpOutSegs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory

```

```
DESCRIPTION
      "The total number of segments sent, including
      those on current connections but excluding
those
      containing only retransmitted octets."
 ::= { tcp 11 }

tcpRetransSegs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
      "The total number of segments retransmitted -
that
      is, the number of TCP segments transmitted
      containing one or more previously transmitted
      octets."
 ::= { tcp 12 }

-- the TCP Connection table

-- The TCP connection table contains information about this
-- entity's existing TCP connections.

tcpConnTable OBJECT-TYPE
SYNTAX SEQUENCE OF TcpConnEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
      "A table containing TCP connection-specific
      information."
 ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
SYNTAX TcpConnEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
      "Information about a particular current TCP
      connection. An object of this type is
transient,
      in that it ceases to exist when (or soon after)
      the connection makes the transition to the
CLOSED
      state."
INDEX { tcpConnLocalAddress,
        tcpConnLocalPort,
        tcpConnRemAddress,
        tcpConnRemPort }
 ::= { tcpConnTable 1 }

TcpConnEntry ::=
SEQUENCE {
    tcpConnState
        INTEGER,
    tcpConnLocalAddress
       IpAddress,
    tcpConnLocalPort
        INTEGER (0..65535),
    tcpConnRemAddress
       IpAddress,
    tcpConnRemPort
        INTEGER (0..65535)
}

tcpConnState OBJECT-TYPE
SYNTAX INTEGER {
    closed(1),
```

```

        listen(2),
        synSent(3),
        synReceived(4),
        established(5),
        finWait1(6),
        finWait2(7),
        closeWait(8),
        lastAck(9),
        closing(10),
        timeWait(11),
        deleteTCB(12)
    }
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
    "The state of this TCP connection.

The only value which may be set by a management
station is deleteTCB(12). Accordingly, it is
appropriate for an agent to return a `badValue'
response if a management station attempts to
set
this object to any other value.

If a management station sets this object to the
value deleteTCB(12), then this has the effect
of
deleting the TCB (as defined in RFC 793) of the
corresponding connection on the managed node,
resulting in immediate termination of the
connection.

As an implementation-specific option, a RST
segment may be sent from the managed node to
the
other TCP endpoint (note however that RST
segments
are not sent reliably)."
::= { tcpConnEntry 1 }

tcpConnLocalAddress OBJECT-TYPE
SYNTAX  IpAddress
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The local IP address for this TCP connection.

In
the case of a connection in the listen state
which
is willing to accept connections for any IP
interface associated with the node, the value
0.0.0.0 is used."
::= { tcpConnEntry 2 }

tcpConnLocalPort OBJECT-TYPE
SYNTAX  INTEGER (0..65535)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The local port number for this TCP
connection."
::= { tcpConnEntry 3 }

tcpConnRemAddress OBJECT-TYPE
SYNTAX  IpAddress
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The remote IP address for this TCP
connection."

```

```
 ::= { tcpConnEntry 4 }

tcpConnRemPort OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The remote port number for this TCP
connection."
 ::= { tcpConnEntry 5 }

-- additional TCP objects

tcpInErrs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of segments received in error
(e.g., bad TCP checksums)."
 ::= { tcp 14 }

tcpOutRsts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of TCP segments sent containing the
RST flag."
 ::= { tcp 15 }

-- the UDP group

-- Implementation of the UDP group is mandatory for all
-- systems which implement the UDP.

udpInDatagrams OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of UDP datagrams delivered to
UDP users."
 ::= { udp 1 }

udpNoPorts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of received UDP datagrams for
which there was no application at the
destination
port."
 ::= { udp 2 }

udpInErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of received UDP datagrams that
could
not be delivered for reasons other than the
lack
of an application at the destination port."
 ::= { udp 3 }
```

```

    udpOutDatagrams OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The total number of UDP datagrams sent from
this
entity."
::= { udp 4 }

-- the UDP Listener table

-- The UDP listener table contains information about this
-- entity's UDP end-points on which a local application is
-- currently accepting datagrams.

udpTable OBJECT-TYPE
    SYNTAX SEQUENCE OF UdpEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A table containing UDP listener information."
::= { udp 5 }

udpEntry OBJECT-TYPE
    SYNTAX UdpEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Information about a particular current UDP
listener."
INDEX { udpLocalAddress, udpLocalPort }
::= { udpTable 1 }

UdpEntry ::=

SEQUENCE {
    udpLocalAddress
        IpAddress,
    udpLocalPort
        INTEGER (0..65535)
}

udpLocalAddress OBJECT-TYPE
    SYNTAX IpAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The local IP address for this UDP listener.
In
the case of a UDP listener which is willing to
accept datagrams for any IP interface
associated
with the node, the value 0.0.0.0 is used."
::= { udpEntry 1 }

udpLocalPort OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The local port number for this UDP listener."
::= { udpEntry 2 }

-- the EGP group

-- Implementation of the EGP group is mandatory for all
-- systems which implement the EGP.

```

```
egpInMsgs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of EGP messages received without
         error."
    ::= { egp 1 }

egpInErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of EGP messages received that
proved
         to be in error."
    ::= { egp 2 }

egpOutMsgs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of locally generated EGP
         messages."
    ::= { egp 3 }

egpOutErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of locally generated EGP messages
not
         sent due to resource limitations within an EGP
         entity."
    ::= { egp 4 }

-- the EGP Neighbor table

-- The EGP neighbor table contains information about this
-- entity's EGP neighbors.

egpNeighTable OBJECT-TYPE
    SYNTAX SEQUENCE OF EgpNeighEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The EGP neighbor table."
    ::= { egp 5 }

egpNeighEntry OBJECT-TYPE
    SYNTAX EgpNeighEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Information about this entity's relationship
with
         a particular EGP neighbor."
    INDEX { egpNeighAddr }
    ::= { egpNeighTable 1 }

EgpNeighEntry ::= 
    SEQUENCE {
        egpNeighState
        INTEGER,
```

```

        egpNeighAddr
            IpAddress,
        egpNeighAs
            INTEGER,
        egpNeighInMsgs
            Counter,
        egpNeighInErrs
            Counter,
        egpNeighOutMsgs
            Counter,
        egpNeighOutErrs
            Counter,
        egpNeighInErrMsgs
            Counter,
        egpNeighOutErrMsgs
            Counter,
        egpNeighStateUps
            Counter,
        egpNeighStateDowns
            Counter,
        egpNeighIntervalHello
            INTEGER,
        egpNeighIntervalPoll
            INTEGER,
        egpNeighMode
            INTEGER,
        egpNeighEventTrigger
            INTEGER
    }

egpNeighState OBJECT-TYPE
    SYNTAX  INTEGER {
        idle(1),
        acquisition(2),
        down(3),
        up(4),
        cease(5)
    }
    ACCESS  read-only
    STATUS   mandatory
    DESCRIPTION
        "The EGP state of the local system with respect
to
        this entry's EGP neighbor. Each EGP state is
        represented by a value that is one greater than
        the numerical value associated with said state
in
        RFC 904."
    ::= { egpNeighEntry 1 }

egpNeighAddr OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-only
    STATUS   mandatory
    DESCRIPTION
        "The IP address of this entry's EGP neighbor."
    ::= { egpNeighEntry 2 }

egpNeighAs OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS   mandatory
    DESCRIPTION
        "The autonomous system of this EGP peer. Zero
        should be specified if the autonomous system
        number of the neighbor is not yet known."
    ::= { egpNeighEntry 3 }

egpNeighInMsgs OBJECT-TYPE
    SYNTAX  Counter

```

---

```
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of EGP messages received without
error
from this EGP peer."
 ::= { egpNeighEntry 4 }

egpNeighInErrs OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of EGP messages received from this
EGP
peer that proved to be in error (e.g., bad EGP
checksum)."
 ::= { egpNeighEntry 5 }

egpNeighOutMsgs OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of locally generated EGP messages
to
this EGP peer."
 ::= { egpNeighEntry 6 }

egpNeighOutErrs OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of locally generated EGP messages
not
sent to this EGP peer due to resource
limitations
within an EGP entity."
 ::= { egpNeighEntry 7 }

egpNeighInErrMsgs OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of EGP-defined error messages
received
from this EGP peer."
 ::= { egpNeighEntry 8 }

egpNeighOutErrMsgs OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of EGP-defined error messages sent
to
this EGP peer."
 ::= { egpNeighEntry 9 }

egpNeighStateUps OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The number of EGP state transitions to the UP
state with this EGP peer."
 ::= { egpNeighEntry 10 }
```

```

egpNeighStateDowns OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
    "The number of EGP state transitions from the
UP
    state to any other state with this EGP peer."
    ::= { egpNeighEntry 11 }

egpNeighIntervalHello OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
    "The interval between EGP Hello command
    retransmissions (in hundredths of a second).
This
    represents the t1 timer as defined in RFC 904."
    ::= { egpNeighEntry 12 }

egpNeighIntervalPoll OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
    "The interval between EGP poll command
    retransmissions (in hundredths of a second).
This
    represents the t3 timer as defined in RFC 904."
    ::= { egpNeighEntry 13 }

egpNeighMode OBJECT-TYPE
  SYNTAX  INTEGER { active(1), passive(2) }
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
    "The polling mode of this EGP entity, either
    passive or active."
    ::= { egpNeighEntry 14 }

egpNeighEventTrigger OBJECT-TYPE
  SYNTAX  INTEGER { start(1), stop(2) }
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
    "A control variable used to trigger operator-
    initiated Start and Stop events. When read,
this
    variable always returns the most recent value
that
    egpNeighEventTrigger was set to. If it has not
    been set since the last initialization of the
    network management subsystem on the node, it
    returns a value of 'stop'.

on
    When set, this variable causes a Start or Stop
    event on the specified neighbor, as specified
acquisition
    pages 8-10 of RFC 904. Briefly, a Start event
    causes an Idle peer to begin neighbor
peer
    and a non-Idle peer to reinitiate neighbor
    acquisition. A stop event causes a non-Idle
    to return to the Idle state until a Start event
    occurs, either via egpNeighEventTrigger or
    otherwise."
    ::= { egpNeighEntry 15 }

```

```
-- additional EGP objects

egpAs OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS   mandatory
    DESCRIPTION
        "The autonomous system number of this EGP
entity."
    ::= { egp 6 }

-- the Transmission group

-- Based on the transmission media underlying each
interface
-- on a system, the corresponding portion of the
Transmission
-- group is mandatory for that system.

-- When Internet-standard definitions for managing
-- transmission media are defined, the transmission group
is
-- used to provide a prefix for the names of those objects.

-- Typically, such definitions reside in the experimental
-- portion of the MIB until they are "proven", then as a
-- part of the Internet standardization process, the
-- definitions are accordingly elevated and a new object
-- identifier, under the transmission group is defined. By
-- convention, the name assigned is:
--
--     type OBJECT IDENTIFIER      ::= { transmission number
}
--
-- where "type" is the symbolic value used for the media in
-- the ifType column of the ifTable object, and "number" is
-- the actual integer value corresponding to the symbol.

-- the SNMP group

-- Implementation of the SNMP group is mandatory for all
-- systems which support an SNMP protocol entity. Some of
-- the objects defined below will be zero-valued in those
-- SNMP implementations that are optimized to support only
-- those functions specific to either a management agent or
-- a management station. In particular, it should be
-- observed that the objects below refer to an SNMP entity,
-- and there may be several SNMP entities residing on a
-- managed node (e.g., if the node is hosting acting as
-- a management station).

snmpInPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS   mandatory
    DESCRIPTION
        "The total number of Messages delivered to the
        SNMP entity from the transport service."
    ::= { snmp 1 }

snmpOutPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS   mandatory
    DESCRIPTION
```

```

        "The total number of SNMP Messages which were
        passed from the SNMP protocol entity to the
        transport service."
 ::= { snmp 2 }

snmpInBadVersions OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
        "The total number of SNMP Messages which were
        delivered to the SNMP protocol entity and were
for
        an unsupported SNMP version."
 ::= { snmp 3 }

snmpInBadCommunityNames OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
        "The total number of SNMP Messages delivered to
        the SNMP protocol entity which used a SNMP
        community name not known to said entity."
 ::= { snmp 4 }

snmpInBadCommunityUses OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
        "The total number of SNMP Messages delivered to
        the SNMP protocol entity which represented an
SNMP
        operation which was not allowed by the SNMP
        community named in the Message."
 ::= { snmp 5 }

snmpInASNParseErrs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
        "The total number of ASN.1 or BER errors
        encountered by the SNMP protocol entity when
        decoding received SNMP Messages."
 ::= { snmp 6 }

-- { snmp 7 } is not used

snmpInTooBigs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
        "The total number of SNMP PDUs which were
        delivered to the SNMP protocol entity and for
        which the value of the error-status field is
        `tooBig'."
 ::= { snmp 8 }

snmpInNoSuchNames OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
        "The total number of SNMP PDUs which were
        delivered to the SNMP protocol entity and for
        which the value of the error-status field is

```

```
        `noSuchName'."  
 ::= { snmp 9 }  
  
snmpInBadValues OBJECT-TYPE  
    SYNTAX Counter  
    ACCESS read-only  
    STATUS mandatory  
    DESCRIPTION  
        "The total number of SNMP PDUs which were  
        delivered to the SNMP protocol entity and for  
        which the value of the error-status field is  
        `badValue'."  
 ::= { snmp 10 }  
  
snmpInReadOnlys OBJECT-TYPE  
    SYNTAX Counter  
    ACCESS read-only  
    STATUS mandatory  
    DESCRIPTION  
        "The total number valid SNMP PDUs which were  
        delivered to the SNMP protocol entity and for  
        which the value of the error-status field is  
        `readOnly'. It should be noted that it is a  
        protocol error to generate an SNMP PDU which  
        contains the value `readOnly' in the  
error-status  
means  
field, as such this object is provided as a  
of detecting incorrect implementations of the  
SNMP."  
 ::= { snmp 11 }  
  
snmpInGenErrs OBJECT-TYPE  
    SYNTAX Counter  
    ACCESS read-only  
    STATUS mandatory  
    DESCRIPTION  
        "The total number of SNMP PDUs which were  
        delivered to the SNMP protocol entity and for  
        which the value of the error-status field is  
        `genErr'."  
 ::= { snmp 12 }  
  
snmpInTotalReqVars OBJECT-TYPE  
    SYNTAX Counter  
    ACCESS read-only  
    STATUS mandatory  
    DESCRIPTION  
        "The total number of MIB objects which have  
been  
entity  
Get-Request  
        retrieved successfully by the SNMP protocol  
as the result of receiving valid SNMP  
and Get-Next PDUs."  
 ::= { snmp 13 }  
  
snmpInTotalSetVars OBJECT-TYPE  
    SYNTAX Counter  
    ACCESS read-only  
    STATUS mandatory  
    DESCRIPTION  
        "The total number of MIB objects which have  
been  
entity  
Set-Request  
        altered successfully by the SNMP protocol  
as the result of receiving valid SNMP  
PDUs."  
 ::= { snmp 14 }
```

```

 ::= { snmp 14 }

snmpInGetRequests OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of SNMP Get-Request PDUs
which
    have been accepted and processed by the SNMP
    protocol entity."
 ::= { snmp 15 }

snmpInGetNexsts OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of SNMP Get-Next PDUs which
have
    been accepted and processed by the SNMP
protocol
    entity."
 ::= { snmp 16 }

snmpInSetRequests OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of SNMP Set-Request PDUs
which
    have been accepted and processed by the SNMP
    protocol entity."
 ::= { snmp 17 }

snmpInGetResponses OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of SNMP Get-Response PDUs
which
    have been accepted and processed by the SNMP
    protocol entity."
 ::= { snmp 18 }

snmpInTraps OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of SNMP Trap PDUs which have
been accepted and processed by the SNMP
protocol
    entity."
 ::= { snmp 19 }

snmpOutTooBigs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of SNMP PDUs which were
generated by the SNMP protocol entity and for
    which the value of the error-status field is
    `tooBig.'"
 ::= { snmp 20 }

snmpOutNoSuchNames OBJECT-TYPE

```

```
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of SNMP PDUs which were
     generated by the SNMP protocol entity and for
     which the value of the error-status is
     `noSuchName'.""
 ::= { snmp 21 }

snmpOutBadValues OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of SNMP PDUs which were
         generated by the SNMP protocol entity and for
         which the value of the error-status field is
         `badValue'."
 ::= { snmp 22 }

-- { snmp 23 } is not used

snmpOutGenErrs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of SNMP PDUs which were
         generated by the SNMP protocol entity and for
         which the value of the error-status field is
         `genErr'."
 ::= { snmp 24 }

snmpOutGetRequests OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of SNMP Get-Request PDUs
which
entity." have been generated by the SNMP protocol
 ::= { snmp 25 }

snmpOutGetNexsts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of SNMP Get-Next PDUs which
have
been generated by the SNMP protocol entity."
 ::= { snmp 26 }

snmpOutSetRequests OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of SNMP Set-Request PDUs
which
entity." have been generated by the SNMP protocol
 ::= { snmp 27 }

snmpOutGetResponses OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
```

```
        STATUS mandatory
        DESCRIPTION
            "The total number of SNMP Get-Response PDUs
which
            have been generated by the SNMP protocol
entity."
        ::= { snmp 28 }

snmpOutTraps OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of SNMP Trap PDUs which have
            been generated by the SNMP protocol entity."
    ::= { snmp 29 }

snmpEnableAuthenTraps OBJECT-TYPE
    SYNTAX INTEGER { enabled(1), disabled(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Indicates whether the SNMP agent process is
            permitted to generate authentication-failure
            traps. The value of this object overrides any
            configuration information; as such, it provides
a
            means whereby all authentication-failure traps
may
            be disabled.

Note that it is strongly recommended that this
object be stored in non-volatile memory so that
it
            remains constant between re-initializations of
the
            network management system."
    ::= { snmp 30 }
```