ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΔΙΑΧΕΙΡΙΣΗ ΔΙΚΤΥΩΝ ΚΑΙ QoS ΜΕΤΑΔΟΣΗΣ ΓΙΑ ΙΔΙΩΤΙΚΟ ΔΙΚΤΥΟ WAN

Διπλωματική εργασία του
Μπατζιά Νικόλαου

Για την απόκτηση του πτυχίου Μηχανικών Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων

ΕΠΙΒΛΕΠΟΝΤΕΣ ΚΑΘΗΓΗΤΕΣ:

Αρσένης Σπυρίδων
Δασκαλοπούλου Ασπασία

Ιούλιος 2006

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 2

ΚΕΦΑΛΑΙΟ 3

ΚΕΦΑΛΑΙΟ 4

ΕΙΣΑΓΩΓΗ

Η ποιότητα της υπηρεσίας που παρέχεται μέσω ενός πληροφοριακού ή τηλεπικοινωνιακού συστήματος, αποτελεί ένα καίριο σημείο διαφοροποίησής της από άλλες ομότιμες υπηρεσίες. Στη σημερινή εποχή, η ποιότητα της προσφερόμενης υπηρεσίας αποτελεί ένα σημαντικό επιχείρημα στη διαδικασία προώθησής της. Στοιχεία όπως η αδιάλειπτη προσφορά της υπηρεσίας, η ταχύτητα εξυπηρέτησης και η ασφάλεια χρησιμοποιούνται κατά κόρον από τα τμήματα μάρκετινγκ των επιχειρήσεων. Τελικός σκοπός των επιχειρήσεων είναι τα τηλεπικοινωνιακά τους συστήματα να γίνουν κέντρα δημιουργίας εσόδων.

Το μέσο για να εξασφαλιστεί η εύρυθμη λειτουργία ενός πληροφοριακού συστήματος και συνακόλουθα η κάλυψη των ελάχιστων απαιτήσεων του πελάτη, σε σχέση με τις παρεχόμενες υπηρεσίες, είναι η σωστή διαχείριση του συστήματος. Στην παρούσα εργασία θα ασχοληθούμε με το θέμα της διαχείρισης δικτύων και των ενεργειών που πρέπει να γίνονται, ώστε να επιτυγχάνεται η επιθυμητή ποιότητα στις παρεχόμενες υπηρεσίες.

Πέρα από την παρουσίαση των μοντέλων και των πρωτοκόλλων διαχείρισης που υπάρχουν, θα ασχοληθούμε και με τη μελέτη πραγματικών δικτύων, αναλύοντας τον τρόπο με τον οποίο υλοποιούνται οι λειτουργίες διαχείρισης στο δίκτυο μίας μεγάλης επιχείρησης, την οποία επισκεφτήκαμε. Θα εστιάσουμε ιδιαίτερα στο πρόβλημα της διαθεσιμότητας και θα προτείνουμε έναν αλγόριθμο για τον εντοπισμό προβλημάτων που σχετίζονται με την απώλειά της. Για να ελέγξουμε τις αποδόσεις του αλγορίθμου, θα αναπτύξουμε μία προσομοίωση, μοντελοποιώντας κατάλληλα τα στοιχεία του δικτύου.

Η εργασία αποτελείται από τέσσερα κεφάλαια:

1) Η προβληματική της διαχείρισης επιχειρηματικού δικτύου μετάδοσης

2) Η διαχείριση σε πραγματικό περιβάλλον

3) Ανάλυση δικτύου βάση των εργαλείων και λειτουργιών διαχείρισης

4) Υλοποίηση αλγορίθμου για την εύρεση διαθεσιμότητας

Στο πρώτο κεφάλαιο παρουσιάζονται τα μοντέλα διαχείρισης δικτύων που υπάρχουν. Έμφαση δίνεται στα δύο επικρατέστερα – OSI και internet – τα οποία αναλύονται διεξοδικά. Επιπλέον, γίνεται μια λεπτομερής παρουσίαση των δύο πρωτοκόλλων που χρησιμοποιούνται από τα συγκεκριμένα μοντέλα διαχείρισης: του πρωτοκόλλου CMIP, που χρησιμοποιείται στο μοντέλο διαχείρισης OSI, και του πρωτοκόλλου SNMP, που χρησιμοποιείται στο μοντέλο διαχείρισης internet. Το κεφάλαιο κλείνει με μία σύγκριση των δύο μοντέλων διαχείρισης, συναρτήσει των πρωτοκόλλων που χρησιμοποιούν,

Στο δεύτερο κεφάλαιο αναφερόμαστε στη διαχείριση πραγματικών δικτύων. Αρχικά, παρουσιάζουμε τα βήματα τα οποία πρέπει να ακολουθήσουμε όταν έχουμε να μελετήσουμε ένα πραγματικό δίκτυο και στη συνέχεια τα εφαρμόζουμε στο δίκτυο της επιχείρησης, στην οποία αναφερθήκαμε παραπάνω. Επιπλέον, στο συγκεκριμένο

κεφάλαιο παρουσιάζουμε όλη τη διαδικασία εύρεσης, εγκατάστασης και χρήσης ενός λογισμικού για την παρατήρηση των διαχειριζόμενων αντικειμένων σε δρομολογητές του Πανεπιστημίου Θεσσαλίας.

Στο τρίτο κεφάλαιο κλιμακώνουμε προς τα κάτω το δίκτυο της επιχείρησης που επισκεφτήκαμε και αναλύουμε τις λειτουργίες διαχείρισης για ένα τέτοιο δίκτυο. Ιδιαίτερη έμφαση δίνουμε στην ανάλυση βλαβών που προκαλούν απώλεια της διαθεσιμότητας του συστήματος, προτείνοντας τον τρόπο με τον οποίο πρέπει να ενεργήσει το σύστημα διαχείρισης, προκειμένου να εντοπίσει την αιτία του προβλήματος. Ουσιαστικά στο κεφάλαιο αυτό γίνεται μία εισαγωγή στον αλγόριθμο ο οποίος θα παρουσιαστεί στο τέταρτο κεφάλαιο.

Στο τέταρτο κεφάλαιο, αρχικά, παρουσιάζουμε τον τρόπο με τον οποίο μοντελοποιήσαμε τα στοιχεία ενός δικτύου. Η μοντελοποίηση αυτή θα χρησιμοποιηθεί και στην προσομοίωση την οποία αναπτύξαμε. Στη συνέχεια, παρουσιάζουμε μερικές βασικές τεχνικές για τη συσχέτιση γεγονότων και τον αλγόριθμο για τον εντοπισμό βλαβών που προκαλούν προβλήματα διαθεσιμότητας. Τέλος, παρουσιάζουμε την προσομοίωση που αναπτύξαμε για την εφαρμογή του αλγορίθμου σε ένα εικονικό δίκτυο, το οποίο είναι ανάλογο με αυτό της επιχείρησης που επισκεφτήκαμε, και τις αποδόσεις του αλγορίθμου.

Στο παράρτημα βρίσκονται τα αντικείμενα του SNMP και του CMIP, τα οποία χρησιμοποιήσαμε κατά την εκπόνηση της συγκεκριμένης εργασίας.

Κεφάλαιο 1

Η ΠΡΟΒΛΗΜΑΤΙΚΗ ΤΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΔΙΚΤΥΟΥ ΜΕΤΑΔΟΣΗΣ

1.1 Μοντέλα διαχείρισης δικτύων [1], [2], [3]

Με τον όρο *διαχείριση δικτύων*, εννοούμε την ενεργή χρήση τους, που περιλαμβάνει όλες τις ενέργειες, λειτουργίες και διαδικασίες που χρησιμοποιούμε για να τα εποπτεύσουμε και να παρέμβουμε, ώστε να εξασφαλιστεί η σωστή χρήση και λειτουργία τους [4]. Όλες οι ενέργειες, οι λειτουργίες και οι διαδικασίες που προαναφέραμε, ορίζονται από τα *μοντέλα διαχείρισης*.

Υπάρχουν τέσσερα μοντέλα για τη διαχείριση δικτύων. Στην ενότητα αυτή θα γίνει μία συνοπτική παρουσίασή τους, ενώ στις επόμενες ενότητες θα εξετάσουμε με λεπτομέρειες τα δύο πιο διαδεδομένα: το μοντέλο Internet και το μοντέλο OSI (Open System Interconnection). Αρχικά παραθέτουμε μία σύντομη περιγραφή των βασικών μοντέλων διαχείρισης δικτύων.

Το μοντέλο OSI/TMN: Το μοντέλο διαχείρισης δικτύων OSI αναπτύχθηκε από τον οργανισμό ISO (International Standards Organization) και πραγματεύεται τη διαχείριση, τόσο τοπικών δικτύων όσο και δικτύων ευρείας περιοχής. Είναι το πιο πλήρες από όλα τα μοντέλα διαχείρισης και καλύπτει και τα επτά στρώματα του OSI. Στο συγκεκριμένο μοντέλο ακολουθείται αντικειμενοστραφής λογική και οι λειτουργίες διαχείρισης που προβλέπονται εκτελούνται από το πρωτόκολλο Common Management Information Protocol (CMIP). Βασικά μειονεκτήματα του μοντέλου OSI είναι οι μεγάλες απαιτήσεις σε μνήμη, οι οποίες δεκαπέντε χρόνια πριν καθιστούσαν σχεδόν απαγορευτική την υλοποίηση ενός συστήματος που θα λειτουργούσε σε πραγματικές συνθήκες, και η πολυπλοκότητά του.

Το μοντέλο διαχείρισης TMN σχεδιάστηκε από την International Telecommunication Union (ITU) για τη διαχείριση δικτύων τηλεπικοινωνιών. Στηρίζεται στο μοντέλο OSI, όπως αυτό περιγράφηκε παραπάνω, οπότε στη συνέχεια τα μοντέλα OSI και TMN δε θα διαχωρίζονται. Αξίζει να σημειωθεί ότι το TMN προχωράει πέρα από τη διαχείριση σε επίπεδο στοιχείων δικτύου, δικτύου και υπηρεσιών και φτάνει σε λειτουργίες που έχουν σχέση με το επιχειρηματικό περιβάλλον και την αύξηση των κερδών ενός οργανισμού.

Το μοντέλο Internet: Το μοντέλο διαχείρισης δικτύων Internet είναι ένα de facto βιομηχανικό πρότυπο. Υπεύθυνη για την ανάπτυξή του είναι η Internet Engineering Task Force (IETF). Το πρωτόκολλο που χρησιμοποιείται για την πραγματοποίηση των λειτουργιών που προβλέπονται από το συγκεκριμένο μοντέλο είναι το Simple Network Management Protocol (SNMP). Κύριο μέλημα κατά την ανάπτυξη του μοντέλου διαχείρισης Internet – και συνακόλουθα των τριών εκδόσεων του SNMP – ήταν η απλότητα. Αυτός είναι και ο βασικός λόγος που το SNMP είναι το ευρύτερα διαδεδομένο πρωτόκολλο για τη διαχείριση δικτύων. Μερικά από τα μειονέκτημα του μοντέλου διαχείρισης Internet είναι οι αδυναμίες του στον τομέα της ασφάλειας, η

έλλειψη πρόβλεψης για ανάλυση κόστους και η έλλειψη πλήρων λειτουργιών διαχείρισης και για τα επτά στρώματα του OSI.

Το κατανεμημένο μοντέλο διαχείρισης: Σε μεγάλα δίκτυα ο όγκος της πληροφορίας διαχείρισης είναι συνακόλουθα μεγάλος. Το αποτέλεσμα είναι να δημιουργούνται προβλήματα στον τρόπο με τον οποίο διαχειριζόμαστε και αναλύουμε την πληροφορία αυτή. Επιπλέον, αν έχουμε ένα κεντρικό σύστημα διαχείρισης μπορεί να παρουσιαστούν προβλήματα στο δίκτυο, εξαιτίας του μεγάλου όγκου πληροφορίας που κατευθύνεται στον κόμβο στον οποίο βρίσκεται ο διαχειριστής. Για να λυθούν τα παραπάνω προβλήματα, έχουν γίνει προσπάθειες για τη δημιουργία κατανεμημένων συστημάτων διαχείρισης.

Η κατανομή μπορεί να αναλυθεί σε δύο συνιστώσες: λειτουργική και γεωγραφική. Η λειτουργική κατανομή αναφέρεται στην ανάθεση συγκεκριμένης λειτουργίας σε κάθε διαχειριστή. Έτσι για παράδειγμα, μπορεί ένας διαχειριστής να ασχολείται με την καταγραφή των συναγερμών και ένας άλλος με την καταγραφή των αποδόσεων του συστήματος. Η γεωγραφική κατανομή αναφέρεται στη δυνατότητα ένας διαχειριστής να ασχολείται μόνο με ένα τμήμα του δικτύου. Έτσι για παράδειγμα, μπορεί να υπάρχει ένας διαχειριστής για κάθε γεωγραφικό διαμέρισμα της χώρας. Επίσης, μπορεί να υπάρχει και ένα δεύτερο επίπεδο διαχείρισης το οποίο αποτελείται από ένα διαχειριστή ο οποίος έχει τη δυνατότητα να ζητά στοιχεία από όλους τους διαχειριστές των γεωγραφικών διαμερισμάτων. Ως αντιπροσωπευτικό σύστημα του κατανεμημένου μοντέλου διαχείρισης αναφέρουμε το σύστημα Common Object Request Broker Architecture (CORBA). Το CORBA χρησιμοποιεί αντικειμενοστραφή λογική και προσφέρει δυνατότητες, τόσο λειτουργικής όσο και γεωγραφικής κατανομής.

Το Web-Based μοντέλο διαχείρισης: Το συγκεκριμένο μοντέλο διαχείρισης δικτύων βασίζεται, όπως δηλώνει και το όνομά του, σε web τεχνολογίες. Πρόκειται για μία ανερχόμενη τεχνολογία και υπάρχουν σε εξέλιξη σχετικές ερευνητικές προσπάθειες. Βασικές ιδέες είναι η χρησιμοποίηση ενός απλού φυλλομετρητή παγκοσμίου ιστού (browser) για την προσπέλαση των δεδομένων που μας ενδιαφέρουν και η χρήση του πρωτοκόλλου HTTP. Στη συνέχεια δε θα ασχοληθούμε με το Web-Based μοντέλο διαχείρισης. Παρόλα αυτά αναφέρουμε δύο αξιόλογες προσπάθειες που εντάσσονται στο μοντέλο αυτό. Η πρώτη είναι η Web-Based Enterprise Management (WBEM), η οποία βρίσκεται στη διαδικασία προτυποποίησης από τον οργανισμό Distributed Management Task Force (DMTF). Στόχος είναι η ανάπτυξη τεχνολογιών διαχείρισης σε περιβάλλον Internet για την ενοποίηση της διαχείρισης επιχειρησιακών περιβαλλόντων. Η δεύτερη είναι η ανάπτυξη του Java Management Extensions (JMX) από τη Sun Microsystems Inc. Βασικό σημείο σε αυτή την προσπάθεια είναι ότι το JMX μπορεί να ενσωματωθεί στα ήδη υπάρχοντα περιβάλλοντα διαχείρισης SNMP, οπότε οι επενδύσεις σε υπάρχοντα συστήματα μπορούν να διατηρηθούν.

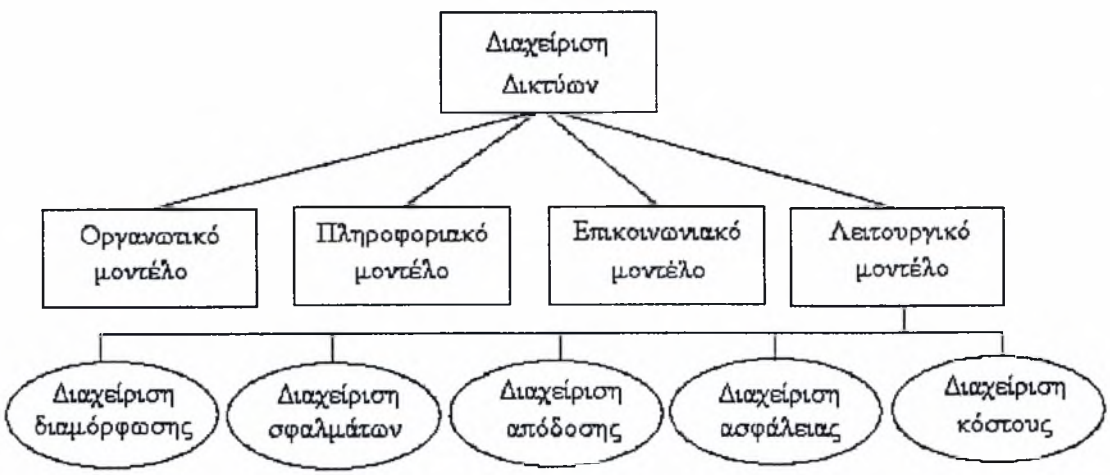
Στο συγκεκριμένο σημείο αξίζει να αναφέρουμε τα πρότυπα της IEEE για LAN και MAN.Τα πρότυπα αυτά αναφέρονται στα δύο πρώτα επίπεδα του OSI (φυσικό και ζεύξης). Πιο συγκεκριμένα, στη σειρά 802.x της IEEE υπάρχουν τα πρότυπα που ορίζονται για το φυσικό μέσο μετάδοσης και για τα πρωτόκολλα στο επίπεδο ζεύξης. Τα μοντέλα διαχείρισης OSI/CMIP και Internet/SNMP ακολουθούν πλήρως τις προδιαγραφές αυτές.

Με βάση όσα αναφέρθηκαν παραπάνω, είναι φανερό ότι τα δύο μεγάλα «αντίπαλα στρατόπεδα» στη διαχείριση δικτύων είναι το μοντέλο OSI/TMN και το μοντέλο Internet. Όπως είπαμε, το μοντέλο Internet χρησιμοποιήθηκε ευρύτερα, κυρίως εξαιτίας της απλότητάς του και των μικρών απαιτήσεών του σε μνήμη. Η χρήση του όμως έφερε στην επιφάνεια και τις πολλές του αδυναμίες, που έχουν σχέση με την έλλειψη πληρότητας των λειτουργιών διαχείρισης που προβλέπει. Από την άλλη, το μοντέλο διαχείρισης OSI μπορεί αρχικά να μη μπορούσε να εφαρμοστεί σε πραγματικά συστήματα, λόγω των μεγάλων απαιτήσεών του σε μνήμη, αλλά πλέον η αύξηση του μεγέθους των μνημών καθιστά αυτό τον περιορισμό παρελθόν. Επίσης, όπως είπαμε, το μοντέλο OSI/TMN είναι πλήρες και προβλέπει λειτουργίες που αναφέρονται σε επιχειρηματικό επίπεδο. Αυτοί είναι και οι λόγοι που το ενδιαφέρον για το μοντέλο διαχείρισης OSI γίνεται όλο και πιο έντονο, ιδιαίτερα από οργανισμούς που διαθέτουν μεγάλα δίκτυα. Στις ενότητες που ακολουθούν θα αναφερθούμε με λεπτομέρεια στα δύο μοντέλα και στις λειτουργίες τους, ώστε να γίνουν σαφέστερα τα πλεονεκτήματα και τα μειονεκτήματα του καθενός.

1.2 Ανάλυση της διαχείρισης με βάση μοντέλα και λειτουργίες [1]

Η διαχείριση δικτύων μπορεί να αναλυθεί με βάση τέσσερα μοντέλα. Τα μοντέλα αυτά είναι:

- Το οργανωτικό μοντέλο
- Το πληροφοριακό μοντέλο
- Το επικοινωνιακό μοντέλο
- Το λειτουργικό μοντέλο



Σχήμα 1.1: *Μοντέλα και λειτουργίες διαχείρισης*

Επιπλέον, το λειτουργικό μοντέλο αναλύεται περαιτέρω σε πέντε βασικές λειτουργίες διαχείρισης:

- Διαχείριση διαμόρφωσης
- Διαχείριση σφαλμάτων
- Διαχείριση απόδοσης
- Διαχείριση ασφάλειας
- Διαχείριση κόστους

Στην πραγματικότητα, τα παραπάνω μοντέλα αναφέρονται στο μοντέλο διαχείρισης OSI, αλλά, τουλάχιστον τα τρία πρώτα, βρίσκουν πλήρεις αντιστοιχίες και στο μοντέλο διαχείρισης Internet. Το τέταρτο μοντέλο (λειτουργικό) δεν ορίζεται άμεσα από τις προδιαγραφές του μοντέλου Internet, αλλά όπως θα δούμε, ορίζεται έμμεσα μέσω των λειτουργιών του SNMP. Στο σχήμα 1.1 φαίνεται η ανάλυση της διαχείρισης δικτύων σε μοντέλα και λειτουργίες διαχείρισης. Στη συνέχεια θα περιγράψουμε αναλυτικά τα προαναφερθέντα μοντέλα.

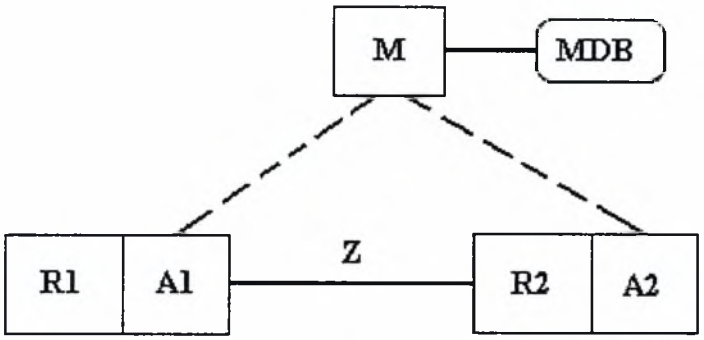
1.2.1 Οργανωτικό μοντέλο [1], [2]

Το οργανωτικό μοντέλο περιγράφει το στοιχεία από τα οποία αποτελείται ένα δίκτυο και τις μεταξύ τους σχέσεις. Τα στοιχεία ή αντικείμενα ενός δικτύου μπορούν να είναι δρομολογητές, hubs, ζεύξεις, τερματικά κ.τ.λ. και χωρίζονται σε δύο κατηγορίες. Στην πρώτη κατηγορία ανήκουν τα ***διαχειριζόμενα αντικείμενα*** στα οποία «κατοικεί» μία διεργασία που λέγεται ***πράκτορας*** και εκτελεί εντολές που της δίνονται από ένα ***διαχειριστή***. Στη δεύτερη κατηγορία ανήκουν τα μη διαχειριζόμενα αντικείμενα. Για τα αντικείμενα αυτά δεν υπάρχει διεργασία πράκτορα που να ενεργεί πάνω τους.

Ο διαχειριστής είναι η οντότητα η οποία αναλαμβάνει το μεγαλύτερο βάρος στη διαδικασία διαχείρισης. Μπορεί να ζητά από τον κάθε πράκτορα να του στείλει πληροφορίες σχετικές με το διαχειριζόμενο αντικείμενο στο οποίο «κατοικεί». Επίσης, όπως θα δούμε αναλυτικά σε επόμενες ενότητες, έχει τη δυνατότητα να ζητά από τους πράκτορες να αλλάζουν παραμέτρους στα διαχειριζόμενα αντικείμενα. Ο διαχειριστής επεξεργάζεται τα δεδομένα τα οποία του στέλνουν οι πράκτορες και αποθηκεύει τα αποτελέσματα σε μία βάση δεδομένων που διαθέτει. Είναι προφανές ότι εξαιτίας των υπολογισμών που εκτελεί ο διαχειριστής, πρέπει να «κατοικεί» σε ένα σύστημα που διαθέτει επεξεργαστική ισχύ ανάλογη των υπολογισμών που θέλουμε να κάνει.

Οι πράκτορες από την άλλη μεριά, αυτό που κάνουν είναι να εκτελούν τις εντολές του διαχειριστή. Επιπλέον, έχουν τη δυνατότητα να στέλνουν μηνύματα στο διαχειριστή για να τον ειδοποιήσουν για κάποιο γεγονός, όπως για παράδειγμα επανεκκίνηση του συστήματος στο οποίο βρίσκονται, χωρίς αυτό να τους έχει ζητηθεί.

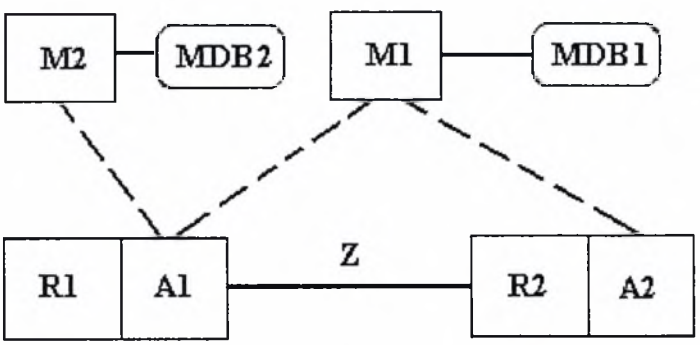
Πρέπει να σημειωθεί ότι στο κατανεμημένο μοντέλο οι πράκτορες αποκτούν σχετική αυτοδυναμία, καθώς έχουν τη δυνατότητα να εκτελέσουν εργασίες που αφορούν στον τομέα ευθύνης τους. Για παράδειγμα, αν σε μία συσκευή η ενεργή πύλη παρουσιάσει βλάβη, ο πράκτορας μπορεί να ενεργοποιήσει εναλλακτική πύλη επικοινωνίας.



Σχήμα 1.2: *Οργανωτικό μοντέλο δύο στρωμάτων*

Στο σχήμα 1.2 περιγράφεται το οργανωτικό μοντέλο για ένα απλό δίκτυο το οποίο αποτελείται από δύο δρομολογητές R1 και R2, στους οποίους «κατοικούν» οι πράκτορες A1 και A2 αντίστοιχα. Οι δρομολογητές δηλαδή είναι διαχειριζόμενα αντικείμενα, σε αντίθεση με τη ζεύξη Z που τους ενώνει. Ο διαχειριστής M «κατοικεί» σε ένα σύστημα το οποίο επικοινωνεί με τους δύο δρομολογητές με ζεύξεις, οι οποίες στο σχήμα σημειώνονται με διακεκομμένες γραμμές. Η MDB (Management Data Base) είναι η βάση στην οποία συγκεντρώνονται τα αποτελέσματα των υπολογισμών και των αναλύσεων που κάνει ο διαχειριστής. Σημειώνεται ότι στο παράδειγμά μας ακολουθούμε τη λογική του μοντέλου internet, αφού στο μοντέλο OSI και οι ζεύξεις αποτελούν διαχειριζόμενα αντικείμενα.

Το παραπάνω οργανωτικό μοντέλο αποτελείται από δύο στρώματα: το πρώτο είναι αυτό των αντικειμένων του δικτύου και το δεύτερο αυτό του συστήματος διαχείρισης δικτύου, στο οποίο κατοικεί ο διαχειριστής. Όπως θα περιγράψουμε με λεπτομέρεια στο πληροφοριακό μοντέλο, ο διαχειριστής διαθέτει και μία βάση με τα διαχειριζόμενα αντικείμενα. Επίσης, βλέπουμε ότι έχουμε ένα διαχειριστή, ο οποίος έχει πρόσβαση σε όλους τους πράκτορες. Εδώ πρέπει να σημειωθεί ότι με τον όρο πρόσβαση δεν εννοούμε φυσική πρόσβαση, δηλαδή μία ζεύξη από τον διαχειριστή στον πράκτορα, αλλά εξουσιοδότηση για να επικοινωνήσει ο διαχειριστής με τον πράκτορα. Φυσικά, μπορεί να υπάρξουν και παραλλαγές πάνω σε αυτό το απλό οργανωτικό μοντέλο, μερικές εκ των οποίων φαίνονται στα παρακάτω σχήματα.

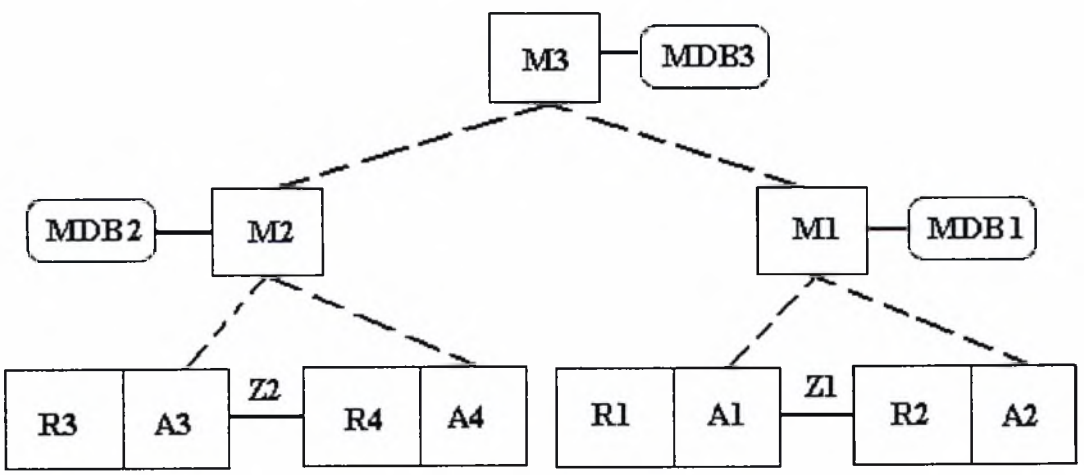


Σχήμα 1.3: *Οργανωτικό μοντέλο με δύο διαχειριστές*

Στο σχήμα 1.3 υπάρχουν δύο διαχειριστές. Ο M1 έχει πρόσβαση τόσο στον A1 όσο και στον A2, ενώ ο M2 έχει πρόσβαση μόνο στον A1. Φυσικά, ο σχεδιασμός θα μπορούσε να έχει γίνει έτσι ώστε ο M2 να έχει πρόσβαση και στους δύο πράκτορες. Ένας τέτοιος σχεδιασμός θα μπορούσε να αποσκοπεί στη λειτουργική κατανομή της διαχείρισης, όπως την περιγράψαμε στο κατανεμημένο μοντέλο διαχείρισης στην ενότητα 1.1.

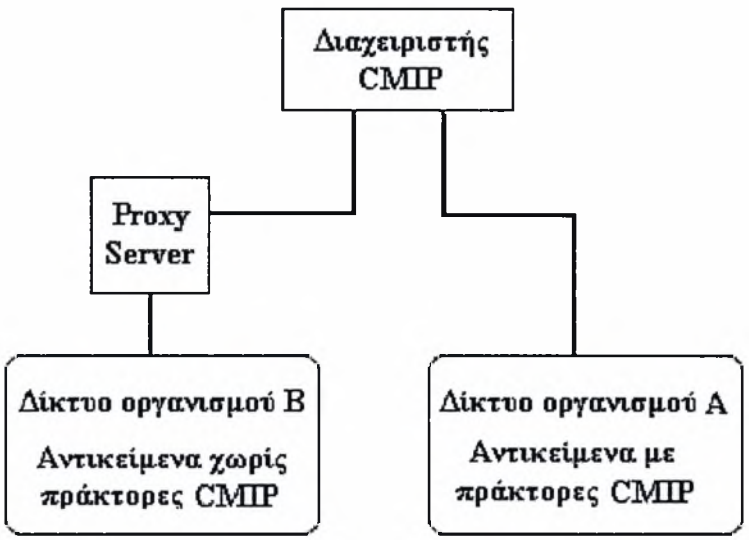
Στο σχήμα 1.4 παρουσιάζεται ένα οργανωτικό μοντέλο τριών στρωμάτων. Στην περίπτωση αυτή ο M1 και ο M2 διαχειρίζονται διαφορετικά αντικείμενα και βρίσκονται στο δεύτερο στρώμα. Στο τρίτο στρώμα βρίσκεται ο M3, ο οποίος επικοινωνεί τόσο με τον M2 όσο και με τον M1. Ουσιαστικά, ο M3 είναι ένας διαχειριστής διαχειριστών (Manager of Managers ή MoM). Τα οργανωτικά μοντέλα με περισσότερα από δύο στρώματα, όπως αναφέραμε στο κατανεμημένο μοντέλο διαχείρισης στην ενότητα 1.1, μπορούν να χρησιμοποιηθούν για την υλοποίηση της γεωγραφικής κατανομής των λειτουργιών διαχείρισης και την αποφυγή υπερφόρτωσης που μπορεί να παρατηρηθεί σε ένα κεντρικό σύστημα.

Μια ακόμη περίπτωση στην οποία βρίσκει εφαρμογή το οργανωτικό μοντέλο πολλαπλών στρωμάτων είναι όταν ο διαχειριστής λειτουργεί με βάση ένα συγκεκριμένο πρωτόκολλο και τα διαχειριζόμενα αντικείμενα δε διαθέτουν πράκτορες που να υποστηρίζουν αυτό το πρωτόκολλο. Η κατάσταση αυτή βέβαια μπορεί να είναι αρκετά περίπλοκη και καλό είναι να αποφεύγεται, όμως σε κάποιες περιπτώσεις υπάρχουν ήδη συστήματα τα οποία δεν μπορούν να αντικατασταθούν (legacy systems) και τα οποία πρέπει να διαχειριστούμε. Ας σκεφτούμε για παράδειγμα έναν οργανισμό Α ο οποίος διαχειρίζεται το δίκτυο του με βάση το πρωτόκολλο CMIP. Έστω ότι ο οργανισμός αυτός εξαγοράζει μία άλλη εταιρία Β, της οποίας το δίκτυο διαθέτει αντικείμενα χωρίς πράκτορες για το CMIP. Το ζητούμενο είναι να καταστεί δυνατή η διαχείριση του δικτύου της Β από το κεντρικό σύστημα διαχείρισης του Α.



Σχήμα 1.4: Οργανωτικό μοντέλο τριών στρωμάτων

Σε περιπτώσεις σαν αυτή που περιγράψαμε παραπάνω μία λύση είναι η εισαγωγή ενός proxy server μεταξύ του κεντρικού διαχειριστή και του « μη συμβατού δικτύου». Σκοπός του proxy server είναι να κάνει τις απαραίτητες μετατροπές, ώστε τα δεδομένα διαχείρισης από το «μη συμβατό δίκτυο» να πάρουν τέτοια μορφή που να μπορούν να γίνουν αντιληπτά από το διαχειριστή. Στο σχήμα 1.5 παρουσιάζεται το οργανωτικό μοντέλο για τη λύση του συγκεκριμένου προβλήματος που περιγράψαμε, με τη χρήση ενός «μεσάζοντα» proxy server.



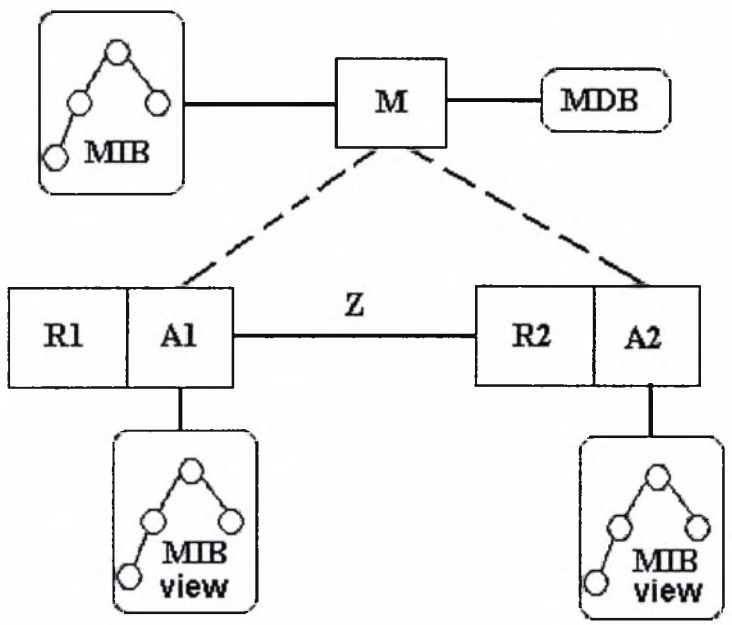
Σχήμα 1.5: Οργανωτικό μοντέλο παραδείγματος με proxy server

1.2.2 Πληροφοριακό μοντέλο [1], [2]

Το πληροφοριακό μοντέλο ασχολείται με τη δομή και τη μορφή με την οποία αποθηκεύεται η πληροφορία διαχείρισης. Δύο είναι οι κυρίαρχες έννοιες οι οποίες έχουν σχέση με το πληροφοριακό μοντέλο: η δομή της πληροφορίας διαχείρισης (***Structure of Management Information ή SMI***) και η βάση της πληροφορίας διαχείρισης (***Management Information Base ή MIB***).

Η SMI ορίζει αυστηρά το συντακτικό και τη σημασιολογία της πληροφορίας που αποθηκεύεται στη MIB. Με τον τρόπο αυτό είναι δυνατή η επικοινωνία του διαχειριστή με τους πράκτορες, αφού υπάρχει κοινή ερμηνεία για την πραγματική σημασία της πληροφορίας που ανταλλάσσεται μεταξύ τους. Η έννοια της *διαμοιραζόμενης γνώσης* ανάμεσα στο διαχειριστή και έναν πράκτορα συνδέεται άμεσα με τη SMI και αναφέρεται σε πληροφορία που είναι γνωστή τόσο στο διαχειριστή όσο και στον πράκτορα και που ερμηνεύεται με τον ίδιο τρόπο και από τους δύο. Είναι σαφές ότι αν οι δύο πλευρές δε διαθέτουν ένα κοινό τρόπο να ερμηνεύουν την πληροφορία διαχείρισης, δε μπορούν να συνεργαστούν αποτελεσματικά.

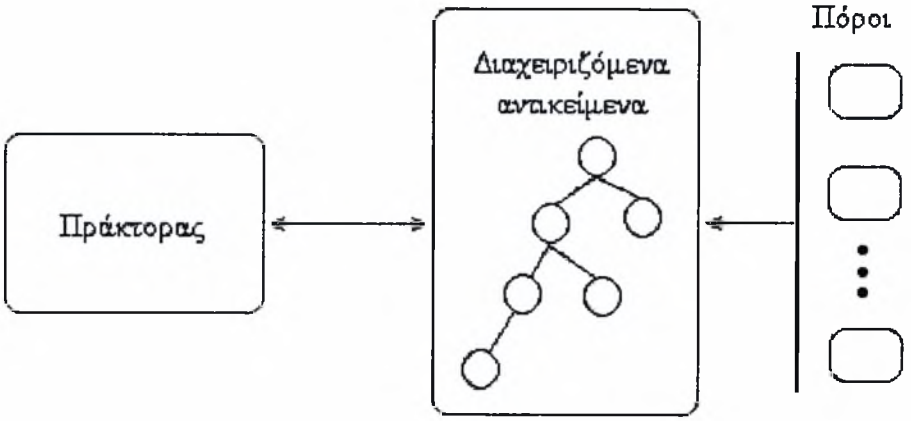
Η MIB χρησιμοποιείται τόσο από το διαχειριστή όσο και από τους πράκτορες για την αποθήκευση της πληροφορίας διαχείρισης. Σε αυτό το σημείο, πρέπει να γίνει σαφές ότι δεν υπάρχει μία MIB για όλες τις διεργασίες (πράκτορες και διαχειριστή), αλλά ότι ο διαχειριστής έχει μία δικιά του MIB και καθένας από τους πράκτορες έχει μία ξεχωριστή MIB. Στη MIB του διαχειριστή υπάρχει πληροφορία για όλα τα στοιχεία του δικτύου που διαχειρίζεται. Αντίθετα, κάθε πράκτορας έχει αποθηκευμένη στη MIB του πληροφορία που αφορά στο στοιχείο του δικτύου στο οποίο «κατοικεί». Λέμε ότι αυτή η τοπική πληροφορία που βρίσκεται στη MIB κάθε πράκτορα αποτελεί μία άποψη της MIB (MIB view). Τα παραπάνω παρουσιάζονται στο σχήμα 1.6.



Σχήμα 1.6: *Πληροφοριακό μοντέλο*

Ένα ακόμα σημείο το οποίο πρέπει να προσέξουμε είναι ότι η MIB του διαχειριστή δεν έχει σχέση με τη βάση δεδομένων στην οποία αποθηκεύονται τα αποτελέσματα των αναλύσεων που κάνει ο διαχειριστής. Η MDB είναι μία ξεχωριστή βάση δεδομένων.

Στο σχήμα 1.7 παρουσιάζεται η σχέση ανάμεσα στον πράκτορα, στα διαχειριζόμενα αντικείμενα που βρίσκονται στη MIB του και στα πραγματικά στοιχεία του δικτύου. Όπως είπαμε, ο διαχειριστής επικοινωνεί με τον πράκτορα και του στέλνει εντολές. Ο πράκτορας ερμηνεύει τις εντολές αυτές και τις στέλνει στα διαχειριζόμενα αντικείμενα. Τα διαχειριζόμενα αντικείμενα εκτελούν τις λειτουργίες που τους υπαγορεύονται και στέλνουν τις κατάλληλες πληροφορίες στον πράκτορα. Ο πράκτορας με τη σειρά του θα στείλει τις πληροφορίες που πρέπει στο διαχειριστή. Στο μοντέλο διαχείρισης OSI η επικοινωνία ανάμεσα στον πράκτορα και τη MIB του γίνεται μέσω λειτουργιών που λέγονται System Management Functions (SMF) και ορίζονται στο ISO 10164. Τα διαχειριζόμενα αντικείμενα (MIB του πράκτορα) αναφέρονται σε πραγματικούς πόρους του δικτύου, δεν είναι όμως αναγκαστικά «φυσικοί πόροι». Για παράδειγμα, ο αριθμός λάθος πακέτων που αποστέλλονται από μία διεπαφή αποτελεί ένα διαχειριζόμενο αντικείμενο της MIB.



Σχήμα 1.7: Πράκτορας, MIB και πραγματικοί πόροι

Πριν προχωρήσουμε σε περισσότερες λεπτομέρειες σε σχέση με την πληροφορία διαχείρισης, αξίζει να σημειωθεί ότι η οργάνωση της ακολουθεί την ***Abstract Syntax Notation One*** (***ASN.1***). Η συγκεκριμένη σημειογραφία στηρίζεται στη γραμματική Backus-Nauer. Επιπλέον, η ASN.1 δίνει τη δυνατότητα ορισμού λιστών, πινάκων, συνόλων αντικειμένων και σύνθετων τύπων δεδομένων που βασίζονται στους απλούς τύπους δεδομένων που παρέχει. Στο ISO 8824/X.208 ο ενδιαφερόμενος αναγνώστης μπορεί να βρει την τεκμηρίωση της ASN.1. Βέβαια, για λόγους απλότητας η πληροφορία διαχείρισης τόσο στο μοντέλο OSI, όσο και στο μοντέλο Internet κωδικοποιείται με τη χρήση ενός υποσυνόλου της ASN.1 και δε γίνεται πλήρης χρήση των δυνατοτήτων της. Το μεγάλο πλεονέκτημα της ASN.1 είναι ότι παρέχει τη δυνατότητα κωδικοποίησης της πληροφορίας διαχείρισης σε μία γλώσσα που είναι σχετικά κατανοητή από τον άνθρωπο με έναν αυστηρό τρόπο. Ο αλγόριθμος που χρησιμοποιείται για τη μετατροπή της ANS.1 σε κώδικα μηχανής ονομάζεται *Basic Encoding Rules (BER)* και ορίζεται στο ISO 8825/x.209.

Στο υπόλοιπο αυτής της ενότητας θα παρουσιάσουμε έννοιες, οι οποίες έχουν σχέση με την πληροφορία διαχείρισης δικτύων. Η παρουσίαση αυτή θα γίνει με βάση τα δύο κύρια μοντέλα διαχείρισης: το μοντέλο OSI και το μοντέλο Internet.

1.2.2.1 Οργάνωση πληροφορίας διαχείρισης στο μοντέλο OSI

Τα διαχειριζόμενα αντικείμενα στο μοντέλο OSI [2], [3], [16]

Τα διαχειριζόμενα αντικείμενα στο μοντέλο OSI μπορούν να οριστούν με βάση τα ακόλουθα πέντε χαρακτηριστικά:

- Object class
- Attributes
- Operations
- Behavior

- Notifications

Όπως έχουμε ήδη πει, το μοντέλο OSI είναι αντικειμενοστραφές. Έτσι κάθε διαχειριζόμενο αντικείμενο ανήκει σε μία κλάση (object class). Στην πραγματικότητα μία κλάση δηλώνει ένα σύνολο από διαχειριζόμενα αντικείμενα με κοινά γνωρίσματα και συμπεριφορά, τα οποία μπορούν να εκτελέσουν ίδιες λειτουργίες και να στείλουν ίδια μηνύματα. Κάθε διαχειριζόμενο αντικείμενο είναι ένα στιγμιότυπο της κλάσης στην οποία ανήκει. Για παράδειγμα μπορεί να έχουμε την κλάση hub. Στην κλάση αυτή ανήκουν όλα τα hub του δικτύου μας. Μερικά από τα γνωρίσματα (attributes) της κλάσης hub μπορεί να είναι ένα αναγνωριστικό (ID), ο κατασκευαστής και ο αριθμός των διεπαφών. Κάθε hub αποτελεί ένα ξεχωριστό στιγμιότυπο της κλάσης, οπότε τα γνωρίσματα που προαναφέρθηκαν παίρνουν τις ανάλογες τιμές.

Τα γνωρίσματα (attributes) ενός διαχειριζόμενου αντικείμενου αποτελούνται από ένα τύπο δεδομένων (π.χ. ακέραιος αριθμός) και μία τιμή. Ο τύπος ενός γνωρίσματος μπορεί να είναι, είτε απλός είτε σύνθετος. Με τον όρο σύνθετος εννοούμε ότι μπορεί να κατασκευαστεί με τέτοιο τρόπο, ώστε να αποτελείται από ένα σύνολο απλών τύπων δεδομένων. Για παράδειγμα, ο τύπος δεδομένων ενός γνωρίσματος μπορεί να αποτελείται από έναν ακέραιο αριθμό και μία συμβολοσειρά. Επιπλέον, υπάρχει η δυνατότητα σε ένα γνώρισμα να μην αποδώσουμε μία μοναδική τιμή αλλά ένα σύνολο τιμών. Όλα αυτά γίνονται με τη χρήση της ASN.1.

Τα γνωρίσματα μπορούν να ομαδοποιούνται και να σχηματίζουν ομάδες *γνωρισμάτων (attribute groups)*. Με κάθε ομάδα συνδέεται ένα μοναδικό αναγνωριστικό, ώστε να είναι εύκολη η αναφορά μας σε αυτή. Χρήση μίας ομάδας μπορούμε να έχουμε, για παράδειγμα, όταν ζητάμε να μας επιστραφούν οι τιμές όλων των γνωρισμάτων της. Τέλος, αναφέρουμε ότι με κάθε γνώρισμα συνδέονται δικαιώματα προσπέλασης (ανάγνωση, εγγραφή, ανάγνωση και εγγραφή).

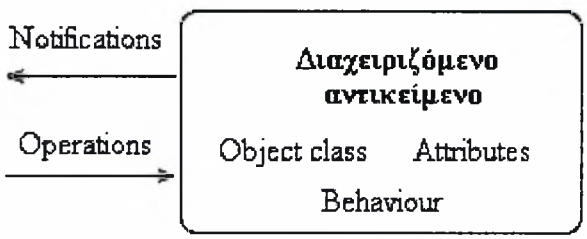
Τα διαχειριζόμενα αντικείμενα γίνονται αποδέκτες ενεργειών, οι οποίες παίρνουν τη μορφή λειτουργιών (operations) που πρέπει να εκτελεστούν. Για παράδειγμα, μία λειτουργία μπορεί να είναι η αλλαγή της τιμής ενός γνωρίσματος. Οι λειτουργίες μπορούν να χωριστούν σε δύο κατηγορίες.

Στην πρώτη κατηγορία λειτουργιών ανήκουν αυτές που έχουν σχέση με τα γνωρίσματα (*attribute-oriented operations*). Πιο συγκεκριμένα οι λειτουργίες αυτές είναι οι *set*, *get*, *replace*, add και *remove*. Οι set και get χρησιμοποιούνται αντίστοιχα για να θέτουμε και να διαβάζουμε την τιμή ενός γνωρίσματος. Η λειτουργία replace μπορεί να χρησιμοποιηθεί όταν θέλουμε να αντικαταστήσουμε την τιμή ενός γνωρίσματος με μία εξορισμού τιμή (replace by default). Μία τέτοια περίπτωση μπορεί να προκύψει, για παράδειγμα, αν δεν έχει δοθεί καμία τιμή στο γνώρισμα κατά τη δημιουργία του (οπότε θα είχε την τιμή NULL). Οι λειτουργίες add και remove χρησιμοποιούνται για την πρόσθεση/αφαίρεση ενός μέλους σε/από ένα σύνολο αντίστοιχα.

Στη δεύτερη κατηγορία λειτουργιών ανήκουν αυτές που έχουν σχέση με αντικείμενα (*object-oriented operations*). Πιο συγκεκριμένα οι λειτουργίες αυτές είναι οι *action*, *create* και *delete*. Με τη λειτουργία action είναι δυνατόν να δοθεί εντολή για την εκτέλεση μίας ενέργειας, αρκετά περίπλοκης, πάνω στο αντικείμενο. Για παράδειγμα, η λειτουργία action θα μπορούσε να χρησιμοποιηθεί για την εκτέλεση ενός ελέγχου σε κάποιο διαχειριζόμενο αντικείμενο. Η δημιουργία και η διαγραφή ενός στιγμιοτύπου μίας κλάσης γίνεται με τις λειτουργίες create και delete αντίστοιχα.

Κάθε αντικείμενο έχει μία συμπεριφορά (behaviour). Η συμπεριφορά ενός αντικειμένου ορίζει όλες εκείνες τις εσωτερικές διεργασίες που λαμβάνουν χώρα στο αντικείμενο με βάση τα ερεθίσματα που λαμβάνει. Η συμπεριφορά ενός αντικειμένου εξαρτάται από τα πακέτα των γνωρισμάτων τα οποία περιλαμβάνει – στα οποία θα αναφερθούμε αργότερα – και εξωτερικεύεται με την αποστολή ειδικών μηνυμάτων (notifications).

Τα notifications είναι μηνύματα τα οποία αποστέλλει το διαχειριζόμενο αντικείμενο στον πράκτορά χωρίς να του έχει ζητηθεί. Με τα μηνύματα αυτά γίνεται ενημέρωση για κάποιο γεγονός. Το γεγονός αυτό μπορεί να είναι μία δυσλειτουργία, ένα σφάλμα που παρουσιάστηκε ή μειωμένη απόδοση του συστήματος. Το μήνυμα αυτό θα αποσταλεί στη συνέχεια στο διαχειριστή με τη μορφή ενός event report. Τα παραπάνω συνοψίζονται στο σχήμα 1.8.



Σχήμα 1.8: Διαχειριζόμενα αντικείμενα στο μοντέλο OSI

Η μέθοδος που χρησιμοποιείται για τον ορισμό των διαχειριζόμενων αντικειμένων στο μοντέλο OSI αναφέρεται στη βιβλιογραφία ως ***Guidelines for Definition of Managed Object (GDMO)***. Η GDMO παρέχει ένα σύνολο από templates με βάση τα οποία μπορούμε να ορίσουμε τα πέντε χαρακτηριστικά των διαχειριζόμενων αντικειμένων που προαναφέρθηκαν, αλλά και τα πακέτα στα οποία θα αναφερθούμε παρακάτω. Αναλυτική παρουσίαση της GDMO γίνεται στο OSI 10165-4/ITU X.722.

Αντικειμενοστραφείς ιδιότητες [3], [1]

Η αντικειμενοστραφής φύση του μοντέλου OSI έχει ως αποτέλεσμα τα διαχειριζόμενα αντικείμενα να παρουσιάζουν ιδιότητες όπως η ενθυλάκωση, η αρθρωτή κατασκευή, η επεκτασιμότητα, η δυνατότητα επαναχρησιμοποίησης και η δημιουργία πολλαπλών σχέσεων μεταξύ τους. Οι ιδιότητες αυτές αναλύονται στη συνέχεια.

- **Ενθυλάκωση**

Μία βασική έννοια της αντικειμενοστραφούς τεχνολογίας είναι αυτή της ενθυλάκωσης. Ένα αντικείμενο περιέχει γνωρίσματα και μεθόδους/λειτουργίες. Οι τιμές των γνωρισμάτων θα μπορούσαν να προσπελαστούν εξωτερικά και να αλλάξουν. Επιπλέον, οι μέθοδοι ενός αντικειμένου θα μπορούσαν να κληθούν

εξωτερικά. Πολλές φορές όμως δεν είναι επιθυμητό να υπάρχει δυνατότητα εξωτερικής προσπέλασης σε κάποια γνωρίσματα ή μεθόδους ενός αντικειμένου. Αυτή είναι και η έννοια της ενθυλάκωσης. Το αντικείμενο περιέχει όλη την απαραίτητη πληροφορία, όμως μπορεί και την «κρύβει» από το εξωτερικό του περιβάλλον.

Η ενθυλάκωση από μία άποψη προσφέρει ένα επίπεδο αφαίρεσης. Η πληροφορία που δεν είναι ορατή έξω από τα όρια του αντικείμένου μπορεί να είναι «περιττή», με την έννοια ότι δεν μας ενδιαφέρει. Για παράδειγμα, μπορεί να είναι πληροφορία σχετική με εσωτερικές λειτουργίες του αντικειμένου. Αν το μόνο που μας ενδιαφέρει είναι η ύπαρξη μίας διεπαφής με το αντικείμενο, ώστε να μπορούμε να επικοινωνήσουμε μαζί του, τότε δεν χρειάζεται να έχουμε γνώση των διεργασιών που λαμβάνουν χώρα στο εσωτερικό του.

Την ενθυλάκωση μπορούμε επίσης να τη δούμε και σαν ένα μηχανισμό ασφάλειας. Είναι πιθανόν κάποια ευαίσθητη πληροφορία να πρέπει να είναι μη προσπελάσιμη από τον «εξωτερικό» κόσμο του αντικειμένου. Επίσης, μπορεί να οριστούν εσωτερικές λειτουργίες που εκτελούν ελέγχους για τη συνέπεια/ορθότητα της πληροφορίας της οποίας το αντικείμενο είναι δέκτης.

- **Αρθρωτή κατασκευή**

Όταν κατασκευάζεται ένα σύστημα διαχείρισης που στηρίζεται σε αντικειμενοστραφή τεχνολογία και οργανώνεται η πληροφορία διαχείρισης, ένα από τα πρώτα ερωτήματα που προκύπτουν έχει σχέση με τον κατάλληλο αριθμό των διαχειριζόμενων αντικειμένων που πρέπει να κατασκευαστούν. Επιπλέον, πρέπει να οριστούν ακριβώς οι λειτουργίες που θα εκτελεί το κάθε αντικείμενο. Άλλωστε, όπως θα δούμε και στη συνέχεια, στο μοντέλο OSI κάθε διαχειριζόμενο αντικείμενο αποτελείται από πακέτα που καθορίζουν και τις λειτουργίες του.

Από την προηγούμενη παράγραφο πρέπει να έχει γίνει ήδη ξεκάθαρο ότι, όταν κατασκευάζουμε διαχειριζόμενα αντικείμενα, ενδιαφερόμαστε ιδιαιτέρως για τον σαφή ορισμό των «ορίων» του κάθε αντικειμένου. Τα «όρια» ενός αντικειμένου αναφέρονται στην πληροφορία που ενθυλακώνει και στις λειτουργίες που εκτελεί. Στόχος είναι να μην υπάρχουν «επικαλύψεις» που θα μπορούσαν να δημιουργήσουν ασυνέπεια στην πληροφορία διαχείρισης ή δυσλειτουργίες στο σύστημα. Επίσης, η αρθρωτή κατασκευή ενός συστήματος οδηγεί σε ένα δομημένο αποτέλεσμα, λιγότερο περίπλοκο και περισσότερο κατανοητό και εύχρηστο.

Για να καταλάβουμε τη σημασία της αρθρωτής οργάνωσης της πληροφορίας διαχείρισης μπορούμε να δούμε το σύνολό της σαν έναν τοίχο. Τα διαχειριζόμενα αντικείμενα είναι τα τούβλα που αποτελούν τον τοίχο. Κάθε τούβλο καταλαμβάνει ένα συγκεκριμένο χώρο («όρια διαχειριζόμενου αντικειμένου»). Επίσης, κάθε τούβλο βρίσκεται σε φυσική επαφή με τα γειτονικά τούβλα. Αυτό μπορούμε να το παρομοιάσουμε με τις σχέσεις που υπάρχουν μεταξύ των αντικειμένων, στις οποίες θα αναφερθούμε παρακάτω. Αν υποθέσουμε ότι υπάρχουν διαφορετικά μεγέθη τούβλων, για την κάλυψη του ίδιου χώρου μπορούμε να χρησιμοποιήσουμε είτε ένα μεγάλο είτε δύο μικρά. Το τελευταίο αντιστοιχεί στην απόφαση για το πόση πληροφορία διαχείρισης θα ενθυλακώνει ένα αντικείμενο. Για παράδειγμα έστω ότι έχουμε ένα διαχειριζόμενο αντικείμενο το οποίο αντιστοιχεί σε ένα interface. Θα μπορούσαμε να επιλέξουμε αυτό το αντικείμενο να ενθυλακώνει με κάποιο τρόπο την πληροφορία για τον αριθμό των λάθος πακέτων που εισήλθαν στο interface και για τον αριθμό των λάθος πακέτων που εξήλθαν από αυτό. Εναλλακτικά θα μπορούσαμε

να φτιάξουμε δύο επιπλέον διαχειριζόμενα αντικείμενα: ένα που να δηλώνει τον αριθμό των λάθος εισερχόμενων πακέτων και ένα που να δηλώνει τον αριθμό των λάθος εξερχόμενων πακέτων.

Ένα ακόμα πλεονέκτημα της αρθρωτής οργάνωσης της πληροφορίας διαχείρισης – για το οποίο δεν υπάρχει φυσικό ανάλογο στο παράδειγμα με τον τοίχο – είναι η ευκολότερη επαναχρησιμοποίηση της. Στο θέμα αυτό θα αναφερθούμε στη συνέχεια. Τελειώνοντας την αναφορά μας στη συγκεκριμένη ιδιότητα, πρέπει να αναφέρουμε ότι ο στόχος για αρθρωτό ορισμό αντικειμένων δεν πρέπει να μας οδηγεί σε υπερβολές. Ο υπερβολικά μεγάλος αριθμός αντικειμένων, σε συνδυασμό με την πολυπλοκότητα των μεταξύ τους σχέσεων, μπορεί να έχει αρνητικές συνέπειες στην ευκολία διαχείρισης της πληροφορίας διαχείρισης.

- **Επεκτασιμότητα και δυνατότητα επαναχρησιμοποίησης**

Η προσαρμογή μίας υπάρχουσας τεχνολογίας, ώστε να μπορέσει να συνεργαστεί με νέες τεχνολογίες, πολλές φορές είναι αναπόφευκτη. Το πρόβλημα που δημιουργείται είναι ιδιαίτερα έντονο, όταν δεν υπάρχει δυνατότητα για την προσαρμογή αυτή ή όταν απαιτούνται πολύ μεγάλες αλλαγές στην υπάρχουσα τεχνολογία. Αυτό μπορεί να οδηγήσει σε δύο πιθανές αποφάσεις. Η πρώτη είναι ότι δε χρησιμοποιούμε νέες τεχνολογίες που δε μπορούν να συνεργαστούν με αυτή που διαθέτουμε ήδη. Προφανώς, κάτι τέτοιο μπορεί να μας στερήσει από πολύ σημαντικές ωφέλειες που παρέχουν οι τεχνολογίες αυτές. Η δεύτερη είναι ότι θα προσπαθήσουμε να τροποποιήσουμε την τεχνολογία που διαθέτουμε για να συνεργαστεί με τη νέα τεχνολογία. Ανάλογες τροποποιήσεις θα μπορούσαμε να κάνουμε – αν φυσικά έχουμε τη δυνατότητα – και στη νέα τεχνολογία. Βέβαια, αλλαγές μεγάλης κλίμακας ή, ακόμα χειρότερα, ανάπτυξη μίας νέας τεχνολογίας για την αντικατάσταση της υπάρχουσας, μεταφράζεται σε μεγάλο κόστος για τους οργανισμούς.

Η αντικειμενοστραφής τεχνολογία παρέχει τη δυνατότητα της *κληρονομικότητας* στην οποία θα αναφερθούμε παρακάτω. Η ιδέα είναι ότι μία κλάση μπορεί να κληρονομεί χαρακτηριστικά από μία άλλη κλάση. Με αυτό τον τρόπο μπορούμε να ορίσουμε μία καινούρια κλάση που επεκτείνει τις δυνατότητες μίας προϋπάρχουσας κλάσης. Έτσι δε χάνουμε χρόνο για να φτιάξουμε πάλι κάτι που έχουμε ήδη έτοιμο αλλά επαναχρησιμοποιούμε τμήματα που έχουμε έτοιμα για να κατασκευάσουμε κάτι που καλύπτει τις νέες μας ανάγκες. Η τακτική αυτή μας επιτρέπει να διατηρούμε στο ακέραιο τις παλιές δυνατότητες του συστήματος μας.

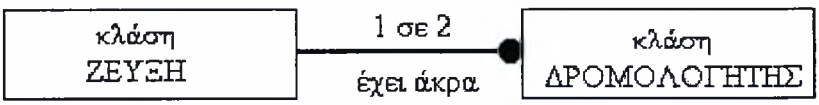
Για να κατανοήσουμε τη μεγάλη σημασία της δυνατότητας επαναχρησιμοποίησης αρκεί να σκεφτούμε την περίπτωση που στο δίκτυό μας εισάγουμε ένα νέο δρομολογητή. Έστω ότι αυτός ο δρομολογητής διαθέτει πράκτορα μίας νεότερης έκδοσης του πρωτοκόλλου διαχείρισης που χρησιμοποιούμε και ότι ο πράκτορας αυτός στέλνει ακριβώς τα ίδια μηνύματα με τους πράκτορες της παλαιότερης έκδοσης. Μοναδική του διαφορά είναι ότι υποστηρίζει ένα επιπλέον μήνυμα. Αυτό που μπορούμε να κάνουμε, για να εκμεταλλευτούμε τις δυνατότητες που μας παρέχει το νέο είδος μηνύματος, είναι να «χτίσουμε» πάνω στο υπάρχον σύστημα διαχείρισης που διαθέτουμε και να προσθέσουμε τη δυνατότητα χειρισμού του νέου μηνύματος. Προφανώς, αν υπάρχουν κλάσεις που αποτελούν τη «βάση» για το χειρισμό μηνυμάτων στο συγκεκριμένο πρωτόκολλο μπορούν να επεκταθούν για να εξυπηρετήσουν το σκοπό μας.

- **Σχέσεις ανάμεσα στα αντικείμενα**

Όταν μοντελοποιούμε ένα σύστημα με βάση την αντικειμενοστραφή προσέγγιση μία πολύ σημαντική παράμετρος που πρέπει να μελετηθεί είναι οι σχέσεις που προκύπτουν ανάμεσα στα αντικείμενα, ή για να είμαστε πιο ακριβείς, ανάμεσα στις κλάσεις των αντικειμένων. Οι σχέσεις αυτές μπορεί να μην είναι εμφανείς αλλά να υπονοούνται. Για παράδειγμα μπορεί να υπάρχει η ίδια πληροφορία σε δύο κλάσεις και να πρέπει να φροντίζουμε ώστε να διατηρείται συνέπεια ανάμεσα στις δύο «εκδόσεις» της πληροφορίας. Η περίπτωση αυτή υποδηλώνει ότι ανάμεσα στις δύο κλάσεις υπάρχει κάποια σχέση.

Ένα εργαλείο το οποίο χρησιμοποιείται για την απεικόνιση των κλάσεων και των μεταξύ τους σχέσεων είναι το μοντέλο οντοτήτων συσχετίσεων (entity relation ή E-R model). Μία πολύ απλή περίπτωση χρήσης του μοντέλου οντοτήτων συσχετίσεων περιγράφεται στη συνέχεια. Έστω ότι έχουμε ορίσει μία κλάση για τις ζεύξεις και μία κλάση για τους δρομολογητές. Είναι προφανές ότι μία ζεύξη ενός δικτύου ευρείας περιοχής έχει σε κάθε άκρο της ένα δρομολογητή. Έτσι κάθε ζεύξη «συνδέεται» με δύο δρομολογητές. Το γεγονός αυτό θα μπορούσε να αναπαρασταθεί με βάση το μοντέλο οντοτήτων συσχετίσεων όπως φαίνεται στο σχήμα 1.9.

Όπως βλέπουμε στη συγκεκριμένη περίπτωση έχουμε μία σχέση «από 1 σε 2». Φυσικά, το συγκεκριμένο μοντέλο παρέχει τη δυνατότητα μοντελοποίησης σχέσεων από 1 σε πολλά, όπου το πολλά είναι ένας αριθμός Ν. Αυτό ακριβώς δηλώνει και ο μαύρος κύκλος μπροστά από την κλάση ΔΡΟΜΟΛΟΓΗΤΗΣ. Επιπλέον, κάθε σχέση εκτός από την πληθικότητά της έχει και ένα όνομα. Στην περίπτωσή μας επιλέξαμε να ονομάσουμε τη σχέση *«έχει άκρα»*. Δεν πρέπει, εξαιτίας της απλότητας του παραδείγματος, να δημιουργηθεί η εντύπωση ότι η μοντελοποίηση με βάση το μοντέλο οντοτήτων συσχετίσεων είναι μία απλή υπόθεση. Τα πράγματα μπορεί να γίνουν ιδιαίτερα περίπλοκα όταν το πλήθος των συσχετιζόμενων κλάσεων είναι μεγάλο. Επιπλέον, υπάρχουν σχέσεις που το μοντέλο οντοτήτων συσχετίσεων αποτυγχάνει να απεικονίσει επαρκώς. Συνήθως οι σχέσεις αυτές αναφέρονται σε θέματα υλοποίησης («επικάλυψη» πληροφορίας και ανταλλαγή πληροφορίας μεταξύ των κλάσεων) και όχι σε «φυσικές» σχέσεις, όπως αυτή του παραδείγματος.



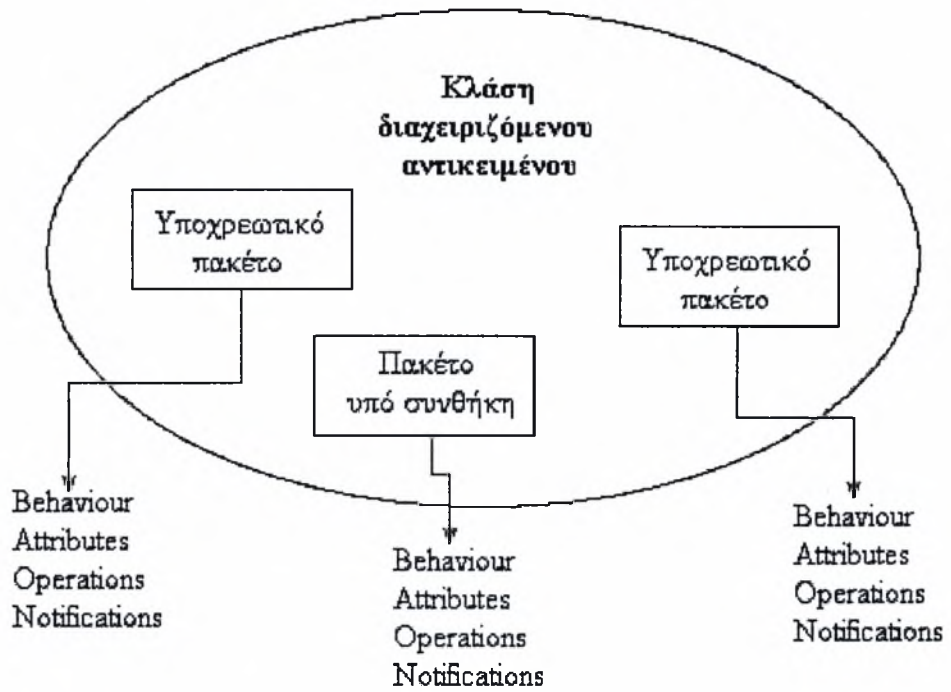
Σχήμα 1.9: Ένα απλό παράδειγμα σχέσης μεταξύ αντικειμένων

Πακέτα [2], [3], [1]

Μία πολύ σημαντική έννοια που σχετίζεται με τη δημιουργία κλάσεων διαχειριζόμενων αντικειμένων στο μοντέλο OSI είναι αυτή των πακέτων. Στην πραγματικότητα τα πακέτα είναι συνιστώντα μέρη των κλάσεων και περιέχουν μία συλλογή από χαρακτηριστικά της κλάσης (δηλαδή behavior, attributes, operations, notifications). Μία κλάση μπορεί να περιέχει ένα ή περισσότερα πακέτα. Από τη στιγμή που δημιουργηθεί ένα στιγμιότυπο της κλάσης, το στιγμιότυπο αυτό διαθέτει

όλα τα χαρακτηριστικά που περιέχονται στα πακέτα. Εδώ εμφανίζεται και η έννοια της δυνατότητας επαναχρησιμοποίησης, στην οποία έχουμε αναφερθεί, καθώς μπορούμε να ορίσουμε ένα πακέτο και στη συνέχεια να το χρησιμοποιήσουμε σε όποια κλάση θέλουμε.

Είναι σημαντικό να γίνει κατανοητό ότι από τη στιγμή που σχηματίζεται ένα στιγμιότυπο μίας κλάσης, τα χαρακτηριστικά που βρίσκονται στα πακέτα της «ενσωματώνονται» στο στιγμιότυπο. Αυτό σημαίνει ότι τα πακέτα από αυτό το σημείο και πέρα δεν έχει καμία αλληλεπίδραση με το στιγμιότυπο. Αν δεν συνέβαινε αυτό πολλά στιγμιότυπα θα «έδειχναν» στο ίδιο πακέτο και θα άλλαζαν, για παράδειγμα, τις τιμές των γνωρισμάτων που περιέχονται σε αυτό. Στην περίπτωση αυτή ανεπιθύμητες καταστάσεις, όπως η ασυνέπεια δεδομένων, θα ήταν αναπόφευκτες.



Σχήμα 1.10: *Μία κλάση διαχειριζόμενου αντικειμένου και τα πακέτα της*

Υπάρχουν δύο ειδών πακέτα: τα ***υποχρεωτικά πακέτα*** (***mandatory packages***) και τα ***πακέτα υπό συνθήκη*** (***conditional packages***). Τα υποχρεωτικά πακέτα περιέχουν χαρακτηριστικά που είναι πάντα παρόντα σε κάθε στιγμιότυπο μίας κλάσης που τα περιέχει. Αντίθετα, τα πακέτα υπό συνθήκη περιέχουν χαρακτηριστικά που είναι παρόντα σε ένα στιγμιότυπο, μόνο αν ισχύει κάποια συνθήκη.

Η συνθήκη που πρέπει να πληρείται, για να περιληφθούν τα γνωρίσματα ενός υπό συνθήκη πακέτου σε ένα στιγμιότυπο, ελέγχεται κατά τη δημιουργία του στιγμιοτύπου. Αν η εκτίμηση της συνθήκης είναι «true» τότε τα γνωρίσματα περιλαμβάνονται στο στιγμιότυπο. Αν η εκτίμηση της συνθήκης είναι «false» τότε τα γνωρίσματα μπορούν, είτε να περιληφθούν στο στιγμιότυπο, είτε όχι.

Μία κλάση διαχειριζόμενου αντικειμένου μπορεί να αποτελείται από μηδέν ή περισσότερα υποχρεωτικά πακέτα και από μηδέν ή περισσότερα υπό συνθήκη πακέτα. Το σχήμα 1.10 αναπαριστά μία τέτοια κλάση. Κάθε πακέτο αποτελείται από

behavior, attributes, operations και notifications. Σημειώνεται ότι κάποια από αυτά τα χαρακτηριστικά μπορεί να μην υπάρχουν σε ένα πακέτο.

Όπως ήδη αναφέραμε, ένα υπό συνθήκη πακέτο μπορεί να είναι παρόν ή όχι σε στιγμιότυπα της ίδιας κλάσης. Βέβαια, η απόφαση αυτή λαμβάνεται μόνο κατά τη στιγμή της δημιουργίας του στιγμιοτύπου. Αν ένα στιγμιότυπο κατά τη δημιουργία του περιέχει ένα υπό συνθήκη πακέτο, τότε το στιγμιότυπο αυτό θα φέρει τα γνωρίσματα του πακέτου σε όλη τη διάρκεια της ζωής του. Αν θέλουμε το στιγμιότυπο να περιέχει και άλλα πακέτα ή να αφαιρεθούν από αυτό κάποια πακέτα που ήδη περιέχει, τότε το στιγμιότυπο πρέπει να διαγραφεί και να ξαναδημιουργηθεί.

Αν ένα γνώρισμα περιέχεται σε δύο πακέτα και τα συγκεκριμένα πακέτα περιέχονται με τη σειρά τους στην ίδια κλάση, τότε στην κλάση αυτή δεν υπάρχει το ίδιο γνώρισμα δύο φορές, αλλά μόνο μία. Τέτοιες περιπτώσεις βέβαια χρήζουν ιδιαίτερης προσοχής κατά τη διαδικασία σχεδίασης, καθώς μπορεί να οδηγήσουν σε αντιφάσεις και ασυνέπεια. Για παράδειγμα, πρόβλημα μπορεί να δημιουργηθεί όταν συμπεριλάβουμε σε δύο πακέτα το ίδιο γνώρισμα (attribute) και στη μία περίπτωση δώσουμε δικαιώματα τροποποίησής του, ενώ στην άλλη όχι.

Δέντρα πληροφορίας διαχείρισης [3], [2], [1]

Στο μοντέλο OSI γίνεται χρήση δενδροειδών δομών για την οργάνωση της πληροφορίας διαχείρισης. Στη συνέχεια θα ασχοληθούμε με τρία δένδρα που χρησιμοποιούνται για το σκοπό αυτό. Το πρώτο δένδρο είναι το δένδρο κληρονομικότητας, το οποίο δείχνει τις σχέσεις υποκλάσης-υπερκλάσης. Το δεύτερο δέντρο χρησιμοποιείται για την ονομασία των στιγμιοτύπων των κλάσεων και ονομάζεται δέντρο ονοματολογίας. Το τρίτο δέντρο ονομάζεται δέντρο καταχώρισης και χρησιμοποιείται για την καταχώριση κλάσεων και πακέτων από μία κεντρική αρχή, ώστε να είναι δυνατή η καθολική χρήση τους.

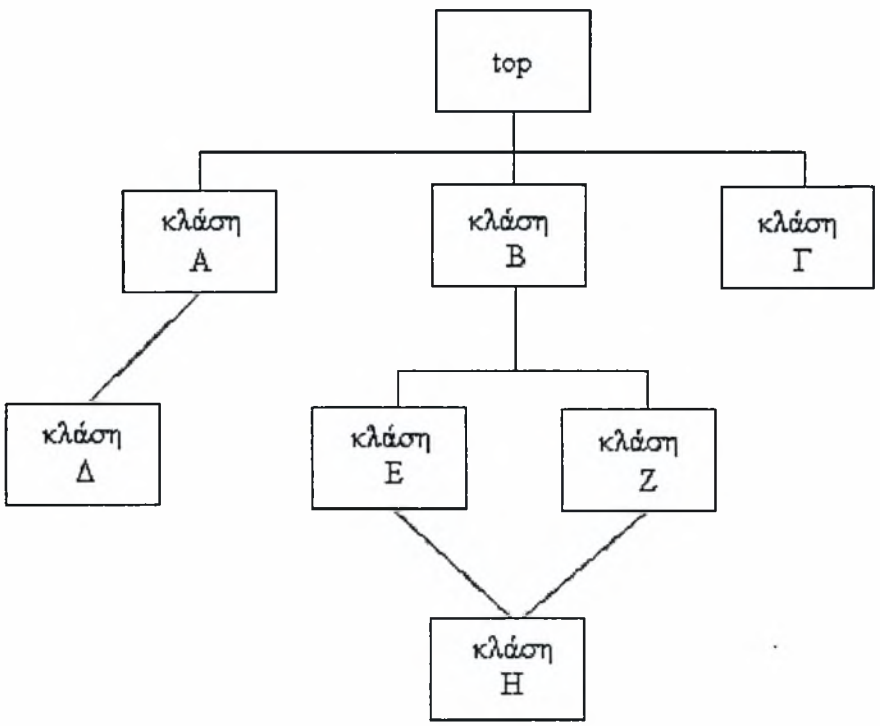
- **Δένδρο κληρονομικότητας**

Μία έννοια που είναι συνδεδεμένη με την αντικειμενοστραφή τεχνολογία είναι αυτή της κληρονομικότητας. Η κληρονομικότητα μας επιτρέπει να κατασκευάσουμε από μία κλάση Α μία νέα κλάση Β, η οποία έχει όλα τα χαρακτηριστικά της Α. Η κλάση Α ονομάζεται *υπερκλάση* και η κλάση Β *υποκλάση*. Λέμε ότι η κλάση Β *κληρονομεί* τα χαρακτηριστικά της Α και ότι μπορεί να την *επεκτείνει* καθώς υπάρχει η δυνατότητα να περιλάβει επιπλέον χαρακτηριστικά. Η διαδικασία επέκτασης μίας υπερκλάσης και η κατασκευή μίας υποκλάσης λέγεται *ειδίκευση*.

Στο μοντέλο OSI, όπως είπαμε, τα γνωρίσματα ομαδοποιούνται σε πακέτα. Σε μία υποκλάση επιτρέπεται μόνο να προστεθούν νέα πακέτα, υποχρεωτικά ή υπό συνθήκη, σε σχέση με την υπερκλάση της και όχι να αφαιρεθούν. Επιπλέον, υπάρχει η δυνατότητα μετατροπής ενός υπό συνθήκη πακέτου σε υποχρεωτικό αλλά το αντίθετο δεν επιτρέπεται. Στην κορυφή της ιεραρχίας κληρονομικότητας που δημιουργείται βρίσκεται η κλάση top, η οποία ορίζεται στο ITU Recommendation X.720. Στην κλάση top υπάρχουν τέσσερα γνωρίσματα, τα οποία κληρονομούνται σε όλες τις κλάσεις, αφού κάθε κλάση είναι άμεσα ή έμμεσα υποκλάση της top. Τα γνωρίσματα αυτά είναι: object class, name binding, packages και allomorphs. Τα δύο πρώτα γνωρίσματα είναι υποχρεωτικό να υπάρχουν σε κάθε κλάση.

Το γνώρισμα object class δηλώνει την κλάση στη οποία ανήκει ένα αντικείμενο. Μία από τις χρησιμότητες που μπορεί να έχει το γνώρισμα αυτό είναι στη διαδικασία filtering του CMIP, στην οποία θα αναφερθούμε σε άλλη ενότητα. Το δεύτερο υποχρεωτικό γνώρισμα – name binding – συνδέεται με την ονομασία ενός συγκεκριμένου στιγμιοτύπου.

Τα άλλα δύο γνωρίσματα αναφέρονται στην πραγματικότητα στην ύπαρξη δύο υπό συνθήκη πακέτων. Το υπό συνθήκη πακέτο packages υποδεικνύει τα πακέτα τα οποία περιέχονται σε μία κλάση, ενώ το υπό συνθήκη πακέτο allomorphs συνδέεται με την ιδιότητα του αλλομορφισμού. Την ιδιότητα αυτή θα αναλύσουμε λίγο παρακάτω, όμως εν συντομία μπορούμε να πούμε ότι αναφέρεται στη δυνατότητα να διαχειριζόμαστε ένα στιγμιότυπο μίας κλάσης σαν να ήταν στιγμιότυπο κάποιας άλλης.

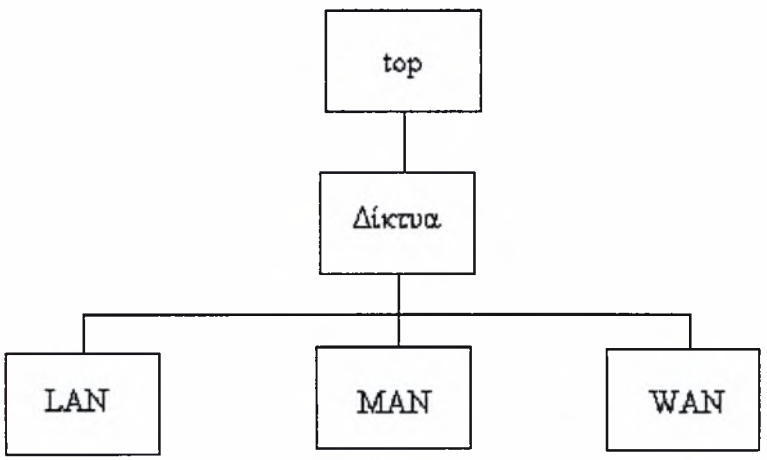


Σχήμα 1.11.α : Μορφή δέντρου κληρονομικότητας

Στο σχήμα 1.11.α παρουσιάζεται η μορφή που μπορεί να έχει ένα δέντρο κληρονομικότητας. Βλέπουμε ότι στην κορυφή της ιεραρχίας κληρονομικότητας βρίσκεται η κλάση top. Επίσης, διακρίνουμε διαφορετικές περιπτώσεις κληρονομικότητας. Η πρώτη περίπτωση είναι αυτή της *απλής κληρονομικότητας*. Στην περίπτωση αυτή μία κλάση έχει μόνο μία άμεση υπερκλάση. Για παράδειγμα, η κλάση Δ έχει μόνο μία άμεση υπερκλάση, την κλάση Α. Η δεύτερη περίπτωση είναι μία κλάση να κληρονομεί άμεσα τα χαρακτηριστικά δύο ή περισσότερων υπερκλάσεων. Στην περίπτωση αυτή λέμε ότι έχουμε *πολλαπλή κληρονομικότητα*. Στο σχήμα μας, για παράδειγμα, βλέπουμε ότι η κλάση Η έχει ως άμεσες υπερκλάσεις την κλάση Ε και την κλάση Ζ. Η τρίτη περίπτωση κληρονομικότητας είναι αυτή του *αλλομορφισμού*. Όπως είπαμε και νωρίτερα, η ιδιότητα του αλλομορφισμού μας δίνει τη δυνατότητα να διαχειριζόμαστε τα αντικείμενα μίας κλάσης σαν να ήταν

αντικείμενα κάποιας άλλης κλάσης. Η δυνατότητα αυτή αποκτά νόημα όταν οι δύο κλάσεις έχουν σχέση «προγόνου» «απογόνου». Για παράδειγμα, έστω ότι η κλάση Δ είναι μία νέα κλάση στο σύστημα που διαθέτουμε και στέλνει μηνύματα τα οποία δεν μπορούμε να χειριστούμε. Μία λύση είναι να χειριστούμε τα στιγμιότυπα της Δ σαν στιγμιότυπα της κλάσης Α. Φυσικά, κάτι τέτοιο έχει ως αποτέλεσμα να μην μπορούμε να εκμεταλλευτούμε τις δυνατότητες που δίνει η νέα κλάση.

Στο σχήμα 1.11.β δίνεται ένα συγκεκριμένο παράδειγμα κληρονομικότητας. Κάτω από την τάξη top βρίσκεται η τάξη Δίκτυα. Η τάξη αυτή έχει τρεις υποκλάσεις: LAN, MAN και WAN. Προφανώς ό,τι ισχύει για την κλάση Δίκτυα ισχύει και για τις υποκλάσεις της. Για παράδειγμα, όλα τα δίκτυα αποτελούνται από κόμβους και ζεύξεις, άρα τα χαρακτηριστικά κόμβοι και ζεύξεις βρίσκονται σε όλες τις κλάσεις του σχήματος. Αντίθετα, ένα χαρακτηριστικό που υπάρχει σε μία υποκλάση δεν είναι υποχρεωτικό να βρίσκεται και στην αντίστοιχη υπερκλάση. Για παράδειγμα, τα δίκτυα LAN μπορεί να έχουν hubs, όμως δε συμβαίνει το ίδιο με όλα τα δίκτυα.



Σχήμα 1.11.β: *Παράδειγμα δέντρου κληρονομικότητας*

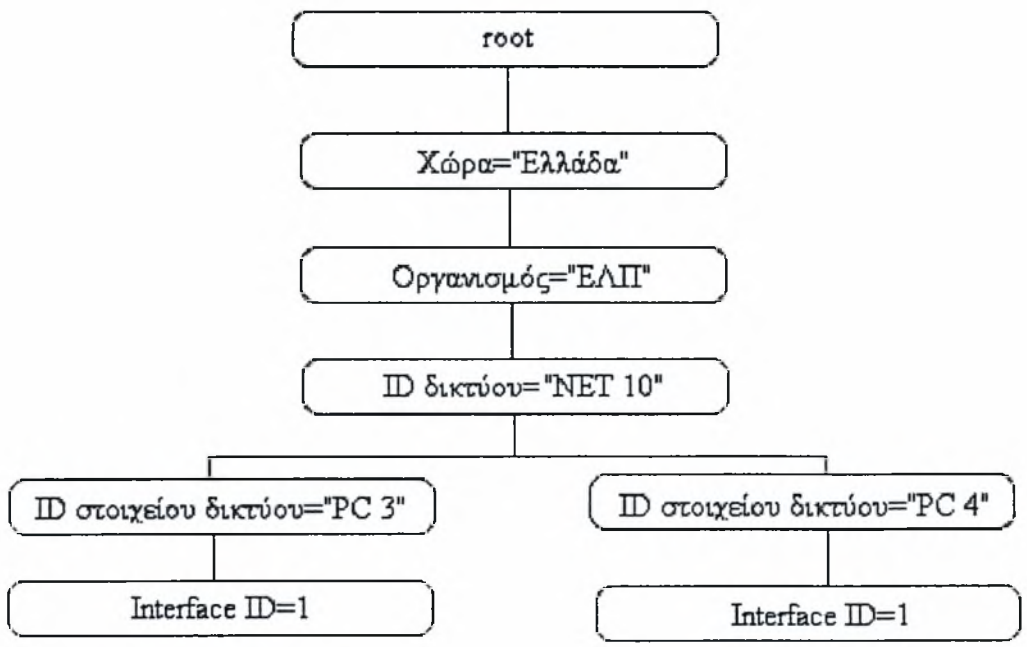
- **Δένδρο ονοματολογίας**

Το πληροφοριακό μοντέλο δεν παρέχει δυνατότητες μόνο για τον ορισμό των κλάσεων και το διαχωρισμό τους, αλλά και για το σαφή διαχωρισμό ανάμεσα σε διαφορετικά στιγμιότυπα της ίδιας κλάσης. Το δένδρο ονοματολογίας (naming tree) είναι το «εργαλείο» το οποίο χρησιμοποιείται για το διαχωρισμό των στιγμιοτύπων. Το δέντρο ονοματολογίας δείχνει επίσης ποιο αντικείμενο «περιέχεται» σε ποιο.

Στο σχήμα 1.12 δίνεται ένα παράδειγμα δέντρου ονοματολογίας. Στην κορυφή του δέντρου βρίσκεται το αντικείμενο root. Το αντικείμενο αυτό περιέχει όλα τα υπόλοιπα. Ακολούθως η Ελλάδα περιέχει έναν οργανισμό που λέγεται ΕΛΠ, ο οποίος διαθέτει ένα δίκτυο με ID NET 10 που αποτελείται από στοιχεία δικτύου κ.τ.λ. Είναι σαφές ότι η ιεραρχία που προκύπτει δεν απεικονίζει καμία σχέση κληρονομικότητας. Επιπλέον, όταν λέμε ότι ένα αντικείμενο στο δέντρο ονοματολογίας «περιέχει» τους κόμβους που βρίσκονται κάτω από αυτό, δεν υπονοούμε ότι υπάρχει αναγκαστικά μία φυσική αρχιτεκτονική σε πλήρη αντιστοιχία με το δέντρο.

Η αναφορά σε ένα συγκεκριμένο στιγμιότυπο μπορεί να γίνει με τρεις τρόπους. Ο πρώτος είναι το *σχετικό όνομα αναγνώρισης* (*relative distinguished* name ή *RDN*). Το σχετικό όνομα αναγνώρισης για το interface του "PC 4" είναι {Interface ID=1}. Βλέπουμε ότι το σχετικό όνομα αναγνώρισης δεν είναι μη διφορούμενο, καθώς υπάρχει και το interface του "PC 3" που έχει ακριβώς το ίδιο σχετικό όνομα αναγνώρισης.

Ένας άλλος τρόπος αναφοράς σε ένα συγκεκριμένο στιγμιότυπο είναι με βάση ένα *καθολικό όνομα* (*global* ή *distinguished name*). Το καθολικό όνομα δίνει τη δυνατότητα αναφοράς σε ένα συγκεκριμένο στιγμιότυπο, με μη διφορούμενο τρόπο. Για να προκύψει το καθολικό όνομα, αρκεί να αρχίσουμε από την κορυφή του δέντρου ονοματολογίας και να παραθέτουμε τα σχετικά ονόματα αναγνώρισης που βρίσκονται στο μονοπάτι που καταλήγει στο στιγμιότυπο που θέλουμε. Για παράδειγμα, το καθολικό όνομα του interface του PC 4 είναι: {Χώρα = "Ελλάδα", Οργανισμός = "ΕΛΠ", ID δικτύου = "NET 10", ID στοιχείου δικτύου = "PC 4", Interface ID = 1}.



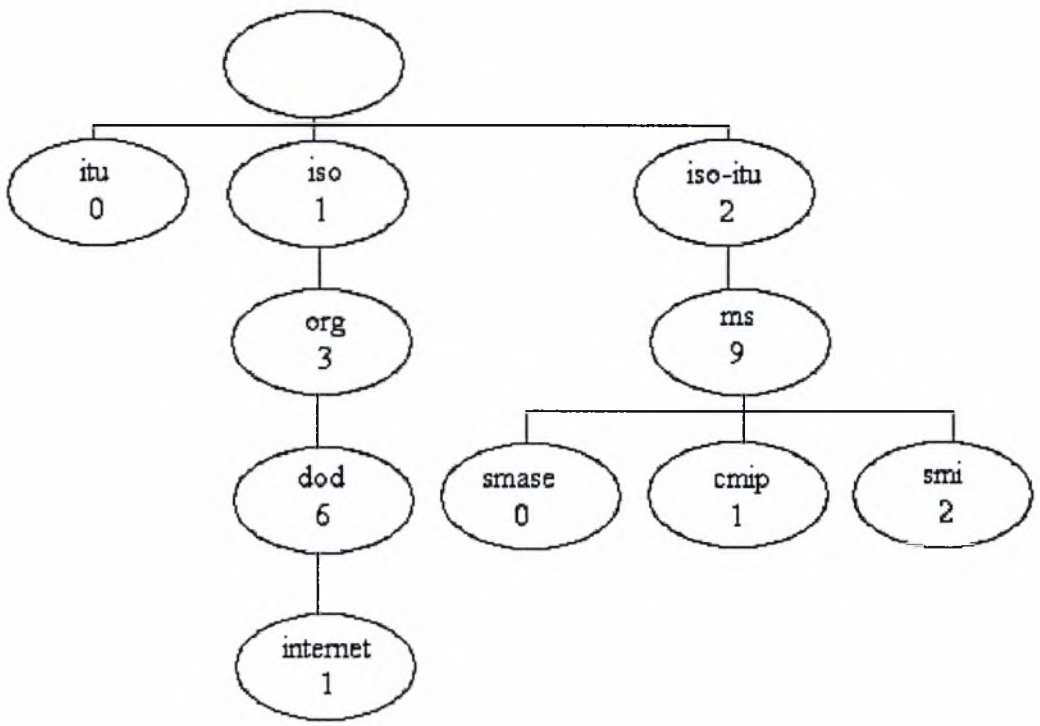
Σχήμα 1.12: *Παράδειγμα δέντρου ονοματολογίας*

Βέβαια, το καθολικό όνομα μπορεί να γίνει σε κάποιες περιπτώσεις ιδιαίτερα μεγάλο. Ένας τρίτος τρόπος αναφοράς σε ένα στιγμιότυπο προσπαθεί να λύσει αυτό το πρόβλημα, φροντίζοντας ταυτόχρονα το όνομα που προκύπτει να είναι μη διφορούμενο. Η βασική ιδέα πίσω από αυτό τον τρόπο είναι ότι αν συμφωνήσουμε εξαρχής στο πλαίσιο στο οποίο αναφερόμαστε (ουσιαστικά στον κόμβο του δέντρου ο οποίος αποτελεί σημείο αναφοράς για το σκοπό μας), μπορούμε να παραλείψουμε το πρώτο μέρος του καθολικού ονόματος. Το όνομα που προκύπτει λέγεται *τοπικό όνομα* (*local name*). Έστω, για παράδειγμα, ότι συμφωνούμε πως πλαίσιο αναφοράς

μας είναι το δίκτυο NET 10. Αν θέλουμε να αναφερθούμε στο ίδιο interface με πριν, το τοπικό όνομα είναι {ID στοιχείου δικτύου = "PC 4", Interface ID = 1}.

- **Δένδρο καταχώρισης**

Σκοπός της δημιουργίας του δέντρου καταχώρισης ήταν να βρεθεί ένας τρόπος για μη διφορούμενη αναφορά στις κλάσεις των διαχειριζόμενων αντικειμένων (όπως ορίζονται στο δέντρο κληρονομικότητας), στα ονόματα των διαχειριζόμενων αντικειμένων (που χρησιμοποιούνται στο δέντρο ονοματολογίας) στα γνωρίσματα και τις ομάδες τους, στους τύπους των action που ορίζονται, στα πακέτα και στα notifications.



Σχήμα *1.13:* *Μέρος του δέντρου καταχώρισης*

Η ιδέα είναι ότι σε κάθε κόμβο του δέντρου έχει καταχωρηθεί ένας αριθμός, ενώ όλοι οι κόμβοι του ίδιου επιπέδου που είναι αδέρφια, έχουν διαφορετικό αριθμό. Η ακολουθία των ακεραίων που προκύπτει από τη ρίζα του δέντρου μέχρι τον κόμβο που μας ενδιαφέρει, του δίνει ένα μοναδικό και μη διφορούμενο αναγνωριστικό.

Στο σχήμα 1.13 φαίνονται κάποιοι από τους κόμβους του δέντρου καταχώρησης. Τονίζεται ότι τα ονόματα στους κόμβους αναγράφονται μόνο για λόγους αναγνωσιμότητας και ευκολίας στην κατανόηση. Στην πραγματικότητα η αναφορά σε ένα κόμβο γίνεται με βάση την ακολουθία των ακεραίων που προαναφέραμε.

Βλέπουμε ότι οι κόμβοι του πρώτου επιπέδου του σχήματος 1.13 αντιστοιχούν σε κάποιες αρχές. Οι αρχές αυτές είναι ο ISO και η ITU. Κάτω από τον κόμβο 2 βρίσκονται πρότυπα που αναπτύχθηκαν και από τον ISO και από την ITU. Τα πρότυπα που έχουν σχέση με το internet βρίσκονται κάτω από τον κόμβο {1 3 6 1} και αναπτύσσονται από τον ISO. Οι κλάσεις, τα πακέτα, τα γνωρίσματα (attributes),

τα notifications και τα actions που περιγράφονται στα X.700/ISO 10165 βρίσκονται κάτω από τον κόμβο smi. Τελειώνοντας την αναφορά μας στο δέντρο καταχώρισης, αξίζει να αναφέρουμε ότι είναι αρκετά συνηθισμένο ή ίδια πληροφορία να βρίσκεται σε περισσότερους από έναν κόμβους του δέντρου. Για παράδειγμα, ας σκεφτούμε την περίπτωση που αναπτύσσεται ένα πρότυπο σε εθνικό επίπεδο. Το πρότυπο αυτό θα καταχωρηθεί σε ένα κόμβο για τα εθνικά πρότυπα. Αν στη συνέχεια το πρότυπο αυτό δημοσιευθεί και σαν recommendation της ITU, τότε θα πρέπει να καταχωρηθεί και σε ένα κόμβο σαν recommendation της ITU.

1.2.2.2 Οργάνωση πληροφορίας διαχείρισης στο μοντέλο Internet

Τα διαχειριζόμενα αντικείμενα στο μοντέλο Internet [1], [11]

Σύμφωνα με το RFC 1155 τα διαχειριζόμενα αντικείμενα στο μοντέλο Internet ορίζονται από τις ακόλουθες πέντε παραμέτρους:

- Object type (identifier και descriptor)
- Syntax
- Access
- Status
- Definition

Η παράμετρος object type ορίζει το είδος του διαχειριζόμενου αντικειμένου με βάση τις δύο συνιστώσες της: το object identifier και το object descriptor. Η συνιστώσα object descriptor είναι ένα όνομα σε μορφή κειμένου. Το όνομα αυτό, που είναι κατανοητό από τον άνθρωπο και μπορεί να απομνημονευθεί εύκολα, δηλώνει τον τύπο του διαχειριζόμενου αντικειμένου. Όπως είναι φυσικό ένα όνομα αντιστοιχίζεται σε ένα μοναδικό τύπο αντικειμένου. Η συνιστώσα object identifier είναι ένας αριθμός ο οποίος ορίζει μονοσήμαντα τον τύπο του διαχειριζόμενου αντικειμένου. Όπως είδαμε, η MIB έχει δενδροειδή δομή και ο object identifier ουσιαστικά δηλώνει τη θέση που έχει ο συγκεκριμένος τύπος αντικειμένου στο δέντρο. Τονίζεται ότι οι συνιστώσες του object type χρησιμοποιούνται απλώς για τον ορισμό του τύπου των διαχειριζόμενων αντικειμένων και όχι για τον ορισμό συγκεκριμένων στιγμιοτύπων. Όταν πρέπει να προσπελαστούν συγκεκριμένα στιγμιότυπα ενός τύπου διαχειριζόμενου αντικειμένου, χρησιμοποιούνται άλλες πληροφορίες όπως η IP διεύθυνση που συνδέεται με το συγκεκριμένο στιγμιότυπο.

Η παράμετρος Syntax ορίζει τη μορφή που έχει το διαχειριζόμενο αντικείμενο, δηλαδή αν είναι ακέραιος, συμβολοσειρά, διεύθυνση IP κ.τ.λ.

Η παράμετρος access δηλώνει τα δικαιώματα προσπέλασης που έχουμε σε ένα διαχειριζόμενο αντικείμενο. Έτσι ορίζεται ότι σε ένα αντικείμενο έχουμε δικαιώματα ανάγνωσης και εγγραφής (read-write), μόνο ανάγνωσης (read only) ή ότι δεν έχουμε δικαίωμα προσπέλασης. Ένα παράδειγμα χρήσης της τελευταίας περίπτωσης είναι όταν ορίζουμε ένα πίνακα. Μπορούμε να δίνουμε δικαιώματα ανάγνωσης και γραφής για τις εγγραφές του πίνακα αλλά για τον πίνακα αυτό καθαυτό μπορεί να μη δίνουμε κανένα δικαίωμα προσπέλασης.

Η παράμετρος definition είναι ένα κείμενο που σκοπό του έχει να περιγράψει με αρκετή λεπτομέρεια το διαχειριζόμενο αντικείμενο. Με τον τρόπο αυτό αποφεύγονται παρανοήσεις σε σχέση με την ακριβή λειτουργία του αντικειμένου. Η περιγραφή αυτή αποτελεί τη βάση για τη δημιουργία σημασιολογικών κανόνων – σε σχέση με την πληροφορία διαχείρισης – που ακολουθούνται από όλους τους κατασκευαστές.

Η παράμετρος status μπορεί να πάρει τρεις τιμές. Η τιμή *mandatory* δηλώνει ότι η υλοποίηση του συγκεκριμένου τύπου αντικειμένου είναι υποχρεωτική. Αντίθετα, η τιμή *optional* δηλώνει ότι η υλοποίηση του συγκεκριμένου τύπου αντικειμένου είναι προαιρετική. Τέλος, η τιμή obsolete δηλώνει ότι μετά τη δημιουργία του, ο συγκεκριμένος τύπος αντικειμένου δεν μπορεί να διαγραφεί.

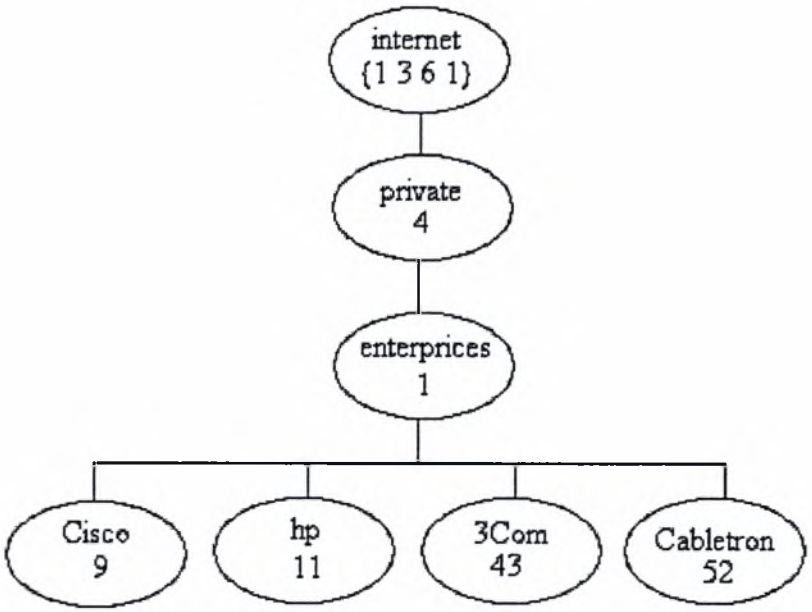
Στη συνέχεια παραθέτουμε την περιγραφή ενός διαχειριζόμενου αντικειμένου από το RFC 1213. Πρόκειται για το αντικείμενο sysLocation, όπως φαίνεται από το OBJECT-TYPE, το οποίο δηλώνει την τοποθεσία στην οποία βρίσκεται το «φυσικό» αντικείμενο (π.χ. ένας δρομολογητής) στο οποίο αναφερόμαστε, όπως περιγράφεται και στο DESCRIPTION. Η παράμετρος STATUS μας ενημερώνει ότι η υλοποίηση του συγκεκριμένου αντικειμένου είναι υποχρεωτική, ενώ βλέπουμε ότι μπορούμε να προσπελάσουμε το συγκεκριμένο αντικείμενο και για ανάγνωση και για εγγραφή. Αξίζει να σημειωθεί ότι ο object identifier στη συγκεκριμένη περίπτωση είναι το system 6, αφού το system μπορεί να αντικατασταθεί από τον αριθμό που το ορίζει αμφιμονοσήμαντα.

```
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}
```

Έχουμε ήδη αναφέρει πως η ASN.1 υποστηρίζει τη δημιουργία πινάκων. Τη δυνατότητα αυτή εκμεταλλευόμαστε για να φτιάξουμε ομάδες συσχετιζόμενων αντικειμένων τα οποία ονομάζονται ***aggregate objects***. Ένα παράδειγμα είναι το ipAddrTable, στο οποίο θα αναφερθούμε αργότερα, όπου αποθηκεύονται οι IP διευθύνσεις των interfaces του διαχειριζόμενου αντικειμένου στο οποίο αναφερόμαστε. Στο RFC 1212 υπάρχει η επίσημη τεκμηρίωση για τη δημιουργία τέτοιων αντικειμένων που αναφέρονται και σαν ***columnar objects***.

Δέντρο πληροφορίας διαχείρισης [1]

Στο μοντέλο internet χρησιμοποιείται το *δέντρο πληροφορίας διαχείρισης* (*management information tree ή MIT*), για το μη διφορούμενο ορισμό των διαχειριζόμενων αντικειμένων. Πρόκειται για το *δέντρο καταχώρισης* στο οποίο έχουμε ήδη αναφερθεί και μέρος του οποίου φαίνεται στο σχήμα 1.13. Στο σχήμα 1.14 παρουσιάζουμε ένα άλλο μέρος του δέντρου αυτού, που δείχνει τη θέση που καταλαμβάνουν κάποιοι κατασκευαστές. Κάθε κατασκευαστής αναπτύσσει νέους κόμβους, κάτω από τον κόμβο που του αντιστοιχεί.



Σχήμα 1.14: *Κόμβοι κατασκευαστών*

Βάση πληροφορίας διαχείρισης [14], [1], [13]

Το πρότυπο που χρησιμοποιείται πλέον για τη MIB είναι η MIB-II, όπως ορίζεται στο RFC 1213. Πρόκειται για ένα υπερσύνολο της MIB-I, ή απλώς MIB, όπως αυτή ορίζεται στο RFC 1156. Τόσο η MIB-I όσο και η MIB-II μπορούν να υλοποιηθούν με το SNMPv1. Μία διαφορά ανάμεσα στη MIB-I και τη MIB-II είναι ότι στη MIB-II το status ενός διαχειριζόμενου αντικειμένου μπορεί να πάρει μία ακόμα τιμή. Πρόκειται για την τιμή deprecated, που δηλώνει ότι η υλοποίηση του συγκεκριμένου αντικειμένου είναι υποχρεωτική στη MIB-II, αλλά το πιο πιθανό είναι να αφαιρεθεί στις επόμενες εκδόσεις της MIB.

Τα αντικείμενα τα οποία συσχετίζονται λογικά δημιουργούν ***ομάδες*** (***object groups***). Η ομαδοποίηση στην οποία αναφερόμαστε εδώ είναι διαφορετική από τα aggregate objects, στα οποία αναφερθήκαμε προηγουμένως, αφού εκεί τα αντικείμενα που αποτελούσαν τη συλλογή, ήταν του ίδιου τύπου. Στη MIB-II ορίζονται έντεκα ομάδες αντικειμένων. Τα αντικείμενα αυτά είναι κόμβοι κάτω από τον κόμβο-αντικείμενο MIB-II του δέντρου καταχώρησης. Το OBJECT IDENTIFIER του αντικειμένου MIB-II είναι 1.3.6.1.2.1, όπως φαίνεται και στο σχήμα 1.15.

Στη συνέχεια θα περιγράψουμε καθεμία από τις έντεκα ομάδες του σχήματος 1.15 ξεχωριστά.

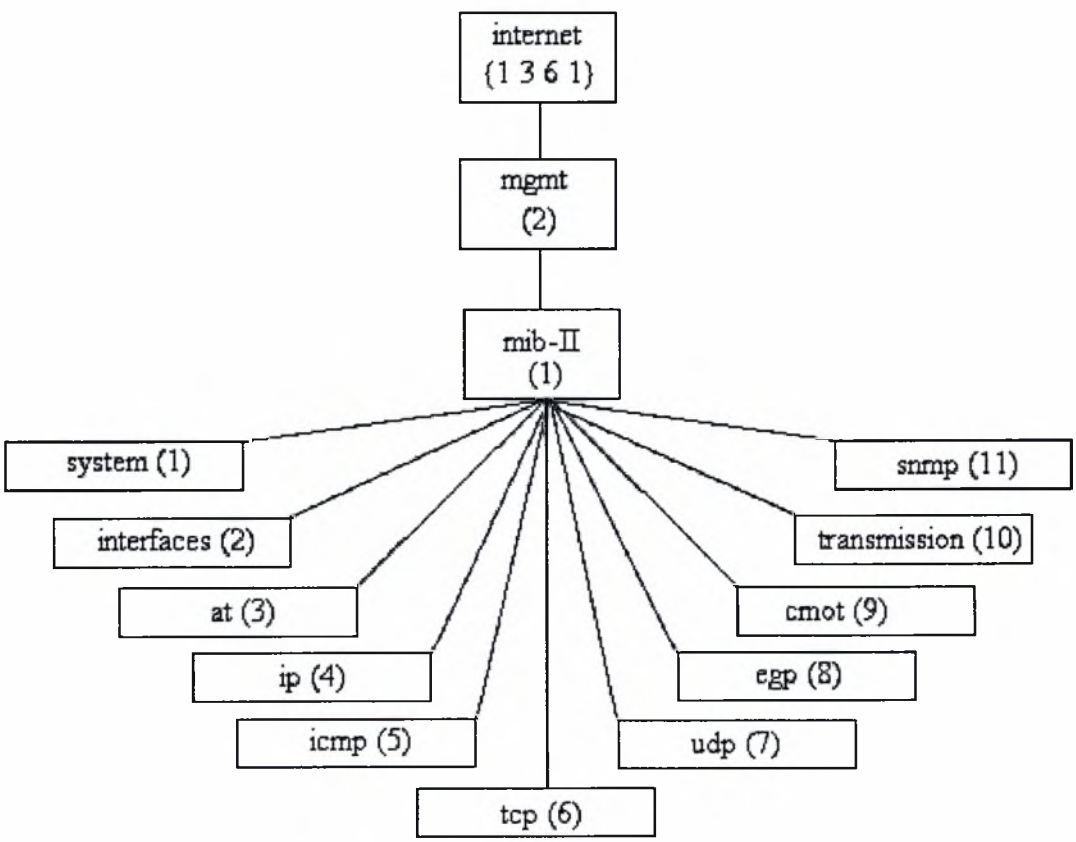
- **System Group**

Το System Group είναι μία από τις βασικότερες ομάδες διαχειριζόμενων αντικειμένων στη MIB-II. Τα στοιχεία της ομάδας αυτής είναι από αυτά που προσπελαύνονται πιο συχνά, όταν διαχειριζόμαστε ένα δίκτυο. Κατά τη διαδικασία «ανακάλυψης» όλων των στοιχείων ενός δικτύου από ένα *Σύστημα Διαχείρισης*

Δικτύου (*Network Management System ή NMS*), τα αντικείμενα του System Group είναι αυτά που θα μας δώσουν, μεταξύ άλλων, πληροφορίες σχετικές με το όνομα του στοιχείου και το ID του. Επιπλέον, στη συγκεκριμένη ομάδα υπάρχει αντικείμενο που δηλώνει τη φυσική τοποθεσία του στοιχείου του δικτύου για το οποίο ενδιαφερόμαστε, καθώς επίσης και κάποιο άλλο που περιέχει τα στοιχεία του ανθρώπου με τον οποίο πρέπει να επικοινωνήσουμε, αν παρουσιαστεί πρόβλημα στο συγκεκριμένο στοιχείο.

Το σύστημα διαχείρισης δικτύου – ουσιαστικά ο διαχειριστής – αντλεί τις πληροφορίες τις οποίες περιγράψαμε παραπάνω με τη χρήση του μηνύματος get-request. Στο μήνυμα αυτό, καθώς και στα άλλα μηνύματα του SNMP, θα αναφερθούμε με λεπτομέρειες όταν αναλύσουμε το επικοινωνιακό μοντέλο.

Η υλοποίηση του System Group είναι υποχρεωτική για όλα τα συστήματα, τόσο στην πλευρά του πράκτορα, όσο και στην πλευρά του διαχειριστή. Αποτελείται συνολικά από επτά αντικείμενα. Πέραν αυτών στα οποία έχουμε ήδη αναφερθεί, υπάρχει ένα ακόμα σημαντικό αντικείμενο στη συγκεκριμένη ομάδα. Πρόκειται για το sysUptime, που μας δίνει το χρόνο μου μεσολάβησε από την τελευταία επανεκκίνηση του στοιχείου.



Σχήμα 1.15: *Ομάδες αντικειμένων στη MIB 2*

- **Interfaces Group**

Το interfaces group, όπως δηλώνει και το όνομά του, περιέχει διαχειριζόμενα αντικείμενα με πληροφορίες που σχετίζονται με τις διεπαφές (interfaces) ενός

συστήματος. Η υλοποίηση της συγκεκριμένης ομάδας είναι υποχρεωτική για όλα τα συστήματα.

Ακριβώς κάτω από τον κόμβο του interfaces group, βρίσκονται δύο κόμβοι. Ο πρώτος ονομάζεται ifNumber και περιέχει τον αριθμό των διεπαφών που έχει το στοιχείο του δικτύου στο οποίο αναφερόμαστε. Ο δεύτερος κόμβος ονομάζεται ifTable. Ο ifTable είναι ένας πίνακας (aggregate object), ο οποίος σχηματικά μπορούμε να πούμε ότι διαθέτει μία γραμμή για κάθε διεπαφή. Μία διεπαφή αντιστοιχεί σε ένα αντικείμενο ifEntry, το οποίο είναι ένας κόμβος κάτω από τον ifTable. Το αντικείμενο ifEntry αναλύεται περαιτέρω σε 22 κόμβους–παιδιά.

Μεταξύ των πληροφοριών που μπορούμε να αντλήσουμε από τα αντικείμενα που είναι παιδιά του ifEntry, βρίσκεται και το μέγεθος της εισερχόμενης και της εξερχόμενης κίνησης σε μία διεπαφή. Επίσης, υπάρχει αντικείμενο το οποίο δηλώνει την κατάσταση στην οποία βρίσκεται η διεπαφή, δηλαδή up, down ή testing.

- **Address Translation Group**

Το address translation group αντιστοιχεί στον κόμβο at του σχήματος 1.14 και αποτελείται από έναν πίνακα που περιέχει την αντιστοιχία των διευθύνσεων δικτύου σε φυσικές διευθύνσεις. Το status του συγκεκριμένου αντικειμένου είναι deprecated, κάτι που δηλώνει ότι το πιο πιθανό είναι να μην υπάρχει στις επόμενες εκδόσεις της MIB. Ο λόγος για την αφαίρεση του συγκεκριμένου αντικειμένου είναι ότι, ήδη από τη MIB-II, η ομάδα κάθε πρωτοκόλλου διαθέτει το δικό της πίνακα μετάφρασης διευθύνσεων. Η υλοποίηση του address translation group στη MIB-II έγινε για λόγους συμβατότητας με τη MIB-I.

- **IP Group**

Όπως ξέρουμε στο Internet, ως πρωτόκολλο δικτύου, χρησιμοποιείται το πρωτόκολλο IP. Το IP group περιέχει πληροφορίες σχετικές με τις διάφορες παραμέτρους του πρωτοκόλλου. Επιπλέον, διαθέτει έναν πίνακα που μπορεί να αντικαταστήσει τον πίνακα μετάφρασης διευθύνσεων του address translation group. Οι δρομολογητές σε ένα δίκτυο περιοδικά ενημερώνουν τους πίνακες δρομολόγησης που διαθέτουν, εκτελώντας ένα αλγόριθμο. Οι πίνακες δρομολόγησης ορίζονται ως διαχειριζόμενα αντικείμενα που ανήκουν στη συγκεκριμένη ομάδα.

Το IP group παρέχει όλες τις πληροφορίες που είναι απαραίτητες στον κόμβο ενός δικτύου για να χειριστεί κίνηση IP. Ο κόμβος του δικτύου μπορεί να είναι, είτε ένα απλό τερματικό, είτε ένας δρομολογητής. Η υλοποίηση του IP group είναι υποχρεωτική.

Οι πληροφορίες που μπορούμε να πάρουμε από τα αντικείμενα του IP group είναι πολλές. Για παράδειγμα μπορούμε να βρούμε τον αριθμό IP πακέτων που απορρίφθηκαν, εξαιτίας λαθών που έγιναν στον αλγόριθμο «επανασυναρμολόγησης», ή τον αριθμό των IP πακέτων που απορρίφθηκαν, επειδή δεν υπήρχε χώρος στους buffers. Τα παραπάνω αντικείμενα είναι ιδιαίτερα χρήσιμα, όταν μελετούμε την απόδοση του δικτύου μας.

Ανάμεσα στα 23 αντικείμενα του IP group βρίσκονται και τρεις πίνακες: ο IP address table, ο IP routing table και ο IP address translation table. Ο IP address table περιέχει πληροφορίες σχετικές με τις IP διευθύνσεις που διαθέτει το στοιχείο του

δικτύου στο οποίο αναφερόμαστε. Για παράδειγμα, εκεί βρίσκονται όλες οι IP διευθύνσεις που μπορεί να έχει ένας δρομολογητής. Ο IP routing table περιέχει πληροφορίες δρομολόγησης. Για κάθε κόμβο-προορισμό, μπορούν να είναι αποθηκευμένες στον IP routing table μέχρι πέντε εναλλακτικές διαδρομές. Τέλος, στον IP address translation table βρίσκεται η αντιστοιχία των IP διευθύνσεων με τις φυσικές διευθύνσεις.

- **ICMP Group**

Το πρωτόκολλο ICMP (Internet Control Message Protocol) είναι μέρος της στοίβας πρωτοκόλλων TCP/IP. Στην πραγματικότητα πρόκειται για ένα πρωτόκολλο που απετέλεσε πρόδρομο του SNMP. Όλες οι παράμετροι που σχετίζονται με το ICMP βρίσκονται στο ICMP group. Η υλοποίηση της συγκεκριμένης ομάδας είναι υποχρεωτική.

Το ICMP group περιέχει 26 διαχειριζόμενα αντικείμενα και μας παρέχει τα στατιστικά των μηνυμάτων ελέγχου του πρωτοκόλλου ICMP. Όλα τα αντικείμενα της ομάδας αυτής είναι μετρητές, στους οποίους μπορούμε να έχουμε πρόσβαση μόνο για ανάγνωση. Ως παράδειγμα αντικειμένου αναφέρουμε το icmpOutEchos, που μας δείχνει τον αριθμό από pings που στάλθηκαν.

- **TCP Group**

Το TCP, όπως ξέρουμε, είναι ένα πρωτόκολλο επιπέδου μεταφοράς που χρησιμοποιείται όταν έχουμε επικοινωνία που βασίζεται σε σύνδεση. Το TCP group αποτελείται από δεκαπέντε αντικείμενα, τα οποία περιέχουν όλες της πληροφορίες που σχετίζονται με το TCP. Ενδεικτικά αναφέρουμε ότι υπάρχει αντικείμενο που δηλώνει τον αριθμό των αποτυχημένων προσπαθειών που έγιναν για τη δημιουργία σύνδεσης.

Η υλοποίηση της συγκεκριμένης ομάδας διαχειριζόμενων αντικειμένων είναι υποχρεωτική. Τα αντικείμενα που αναφέρονται σε μία σύνδεση υπάρχουν μόνο κατά το χρονικό διάστημα που η συγκεκριμένη σύνδεση υφίσταται. Στο αντικείμενο tcpConnTable – το οποίο είναι ένας πίνακας – υπάρχουν αντικείμενα που δηλώνουν τις IP διευθύνσεις και τις θύρες των άκρων κάθε TCP σύνδεσης.

- **UDP Group**

Το UDP είναι ένα πρωτόκολλο επιπέδου μεταφοράς, το οποίο χρησιμοποιείται όταν έχουμε επικοινωνία που δεν είναι προσανατολισμένη σε σύνδεση. Οι πληροφορίες που σχετίζονται με το συγκεκριμένο πρωτόκολλο βρίσκονται στα πέντε αντικείμενα του UDP group.

Η υλοποίηση του UDP group είναι υποχρεωτική. Τα αντικείμενα της ομάδας αυτής δίνουν, μεταξύ άλλων, πληροφορίες για τον αριθμό των datagrams που στέλνονται ή λαμβάνονται και για το πόσα από αυτά που ελήφθησαν είχαν λάθη.

- **EGP Group**

Το EGP group περιέχει πληροφορίες σχετικές με το πρωτόκολλο EGP (external gateway protocol). Η υλοποίηση της συγκεκριμένης ομάδας, η οποία αποτελείται από έξι αντικείμενα, είναι υποχρεωτική.

Μεταξύ των πληροφοριών που μπορούμε να πάρουμε από τα αντικείμενα του EGP group βρίσκεται ο αριθμός των απεσταλμένων και των ληφθέντων μηνυμάτων EGP. Επίσης, αν ενδιαφερόμαστε να εξετάσουμε τις αποδόσεις του συστήματος, υπάρχουν αντικείμενα στα οποία αποθηκεύεται ο αριθμός των λάθος EGP μηνυμάτων που ελήφθησαν.

- **CMOT Group**

Τα αντικείμενα του CMOT group δεν έχουν οριστεί στη MIB-II. Στην πραγματικότητα ο συγκεκριμένος κόμβος δημιουργήθηκε στο δέντρο της MIB-II ώστε να δεσμευτεί χώρος για μελλοντική χρήση.

- **Transmission Group**

Το transmission group δημιουργήθηκε στη MIB-II με σκοπό να υπάρχει ένας κόμβος στο δέντρο της, που θα είναι δεσμευμένος για μελλοντική χρήση. Στόχος ήταν κάτω από τον κόμβο αυτό να δημιουργηθούν αντικείμενα, που θα περιείχαν πληροφορίες για παραμέτρους που σχετίζονται με τη μετάδοση. Από το Μάρτιο του 1991, που εκδόθηκε το RFC 1213 στο οποίο ορίζεται η MIB-II, έχουν προστεθεί αρκετά αντικείμενα κάτω από τον κόμβο του transmission group.

- **SNMP Group**

Το SNMP είναι, όπως έχουμε ήδη πει, το πρωτόκολλο διαχείρισης δικτύου που χρησιμοποιείται στο μοντέλο internet. Το SNMP group περιέχει τα αντικείμενα εκείνα που μπορούν να μας δώσουν πληροφορίες σχετικές με το πρωτόκολλο SNMP. Η υλοποίησή του είναι υποχρεωτική.

Η συγκεκριμένη ομάδα έχει 30 αντικείμενα, δύο εξ' αυτών όμως (το 7 και το 23) δε χρησιμοποιούνται. Μερικές από τις πληροφορίες που μας παρέχουν τα αντικείμενα του SNMP group έχουν σχέση με τον αριθμό όλων των ειδών μηνυμάτων SNMP που εισέρχονται και εξέρχονται σε/από ένα σύστημα. Με τα είδη των μηνυμάτων που υπάρχουν στο SNMP θα ασχοληθούμε όταν αναλύσουμε το επικοινωνιακό μοντέλο.

1.2.3 Επικοινωνιακό μοντέλο [1]

Το επικοινωνιακό μοντέλο ασχολείται με τα μηνύματα τα οποία ανταλλάσσονται κατά τη διαδικασία διαχείρισης ενός δικτύου. Έχουμε ήδη αναφέρει ότι για να είναι

δυνατή η επικοινωνία ενός διαχειριστή με έναν πράκτορα πρέπει να υπάρχει κοινή γνώση κάποιων πραγμάτων και από τις δύο πλευρές. Προφανώς, το πρώτο πράγμα το οποίο πρέπει να είναι γνωστό και στους δύο είναι το πρωτόκολλο το οποίο χρησιμοποιείται για να πραγματοποιηθεί η επικοινωνία. Εννοείται, ότι τόσο ο πράκτορας, όσο και ο διαχειριστής, πρέπει να υποστηρίζουν το πρωτόκολλο αυτό. Με βάση το μηνύματα που υποστηρίζει το συγκεκριμένο πρωτόκολλο ο διαχειριστής υπαγορεύει στον πράκτορα τις ενέργειες που πρέπει να κάνει.

Στη συνέχεια της ενότητας αυτής θα αναφερθούμε στα μηνύματα που υπάρχουν στο SNMP και στο CMIP, τα δύο πρωτόκολλα που χρησιμοποιούνται στα μοντέλα διαχείρισης internet και OSI αντίστοιχα. Πρέπει να σημειωθεί ότι μέρος του επικοινωνιακού μοντέλου μπορεί να θεωρηθεί και ο τρόπος με τον οποίο διασφαλίζεται η πρόσβαση σε μία πληροφορία, μόνο από εξουσιοδοτημένες οντότητες, αφού, για παράδειγμα, στο SNMP ορίζονται κοινότητες οντοτήτων και η επικοινωνία είναι δυνατή μόνο μεταξύ των μελών μίας κοινότητας. Σε αυτά τα θέματα θα αναφερθούμε όταν αναλύσουμε τη λειτουργία της ασφάλειας και όχι εδώ.

1.2.3.1 Τα μηνύματα στο CMIP [15], [3], [2]

Το CMISE (Common Management Information Service Element) παρέχει όλες εκείνες της υπηρεσίες που υλοποιούνται από τα μηνύματα του CMIP. Το σχήμα 1.16 δείχνει τη μορφή που έχει ένα CMIP PDU.

Invoke ID	Operation Value	Managed/ Base Object Class	Managed/ Base Object Instance	Information

Σχήμα 1.16: Ένα PDU του CMIP

Το πεδίο Invoke ID χρησιμοποιείται ως αναγνωριστικό το οποίο χρησιμεύει στην αντιστοίχιση του PDU της απάντησης, με αυτό της ερώτησης. Το πεδίο Operation Value χρησιμοποιείται για να δηλώσει τη λειτουργία που περιέχεται στο PDU. Για παράδειγμα, ένα PDU που περιέχει τη λειτουργία get, έχει την τιμή 3 στο

Υπηρεσία	**Operation Value Με / Χωρίς επιβεβαίωση**	**Περιγραφή**
M-EVENT REPORT	0/1	Αποστολή notification
Multiple responses	2	Δεν είναι υπηρεσία CMISE, αλλά χρησιμοποιείται με το scope
M-GET	3	Ανάκληση γνωρισμάτων και τιμών από τα διαχειριζόμενα αντικείμενα
M-SET	4/5	Θέτουμε ή αλλάζουμε τιμές στα γνωρίσματα των αντικειμένων

M-ACTION	6/7	Ορισμός για μία συγκεκριμένη δράση πάνω σε ένα ή περισσότερα αντικείμενα
M-CREATE	8	Δημιουργία ενός διαχειριζόμενου αντικειμένου
M-DELETE	9	Διαγραφή ενός ή περισσότερων διαχειριζόμενων αντικειμένων
M-CANCEL GET	10	Ακύρωση ενός M-GET που είχε σταλεί νωρίτερα

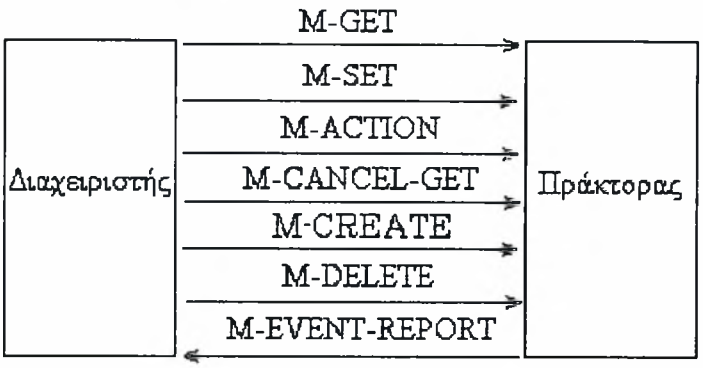
Πίνακας 1.1:*Υπηρεσίες του CMISE και οι αντίστοιχες Operation Values στο CMIP*

συγκεκριμένο πεδίο. Ο πίνακας 1.1 περιέχει τις υπηρεσίες του CMISE και τις αντίστοιχες τιμές του πεδίου Operation Value στο PDU του CMIP. Τα πεδία Managed/Base Object Class και Managed/Base Object Instance αναφέρονται στην κλάση και στο στιγμιότυπο που θέλουμε να εφαρμοστεί η λειτουργία. Ο όρος Base Object έχει σχέση με την παράμετρο scope, η χρήση της οποίας συνδέεται με την ανάκληση πολλών αντικειμένων, χρησιμοποιώντας κάποιο σαν αντικείμενο αναφοράς. Στην παράμετρο scope θα αναφερθούμε αναλυτικά στη συνέχεια. Τέλος, το πεδίο Information είναι στην πραγματικότητα μία ομάδα από πεδία, τα οποία περιέχουν δεδομένα σχετικά με την εκάστοτε λειτουργία.

Στον πίνακα 1.1 βλέπουμε ότι οι υπηρεσίες M-EVENT REPORT, M-SET και M-ACTION του CMISE μπορούν να χρησιμοποιηθούν είτε με επιβεβαίωση είτε χωρίς. Αντίθετα, στην περίπτωση των υπόλοιπων υπηρεσιών η αποστολή επιβεβαίωσης είναι υποχρεωτική. Επιπλέον, οι λειτουργίες M-GET, M-SET, M-ACTION και M-DELETE, όπως θα δούμε και αργότερα, μπορούν να εφαρμοστούν ταυτόχρονα σε περισσότερα από ένα αντικείμενα. Στο σχήμα 1.17 φαίνονται τα μηνύματα στο CMIP, τα οποία όπως βλέπουμε είναι σε πλήρη αντιστοιχία με τις υπηρεσίες που προσφέρει το CMISE. Η αρχή του βέλους δείχνει τον αποστολέα του μηνύματος και το τέλος τον παραλήπτη.

Επίσης, πρέπει να αναφέρουμε ότι η δημιουργία σύνδεσης για να επικοινωνήσουν ο διαχειριστής με τον πράκτορα στο CMIP, γίνεται με τη χρήση της υπηρεσίας ***A-ASSOCIATE*** του CMISE, όπως αυτή ορίζεται στο ISO 8649. Στο ίδιο ISO ορίζονται και οι υπηρεσίες ***A-RELEASE*** και ***A-ABORT*** για την κανονική και τη μη κανονική διακοπή της σύνδεσης.

Στη συνέχεια, θα αναλύσουμε πότε χρησιμοποιείται η κάθε υπηρεσία – ουσιαστικά πότε αποστέλλεται το αντίστοιχο μήνυμα CMIP – καθώς επίσης και τις σχετικές παραμέτρους. Οι παράμετροι αυτές θα παρουσιαστούν και σε πίνακες. Στην πρώτη στήλη του κάθε πίνακα αναφέρονται τα ονόματα των παραμέτρων. Στη δεύτερη στήλη φαίνεται αν μία παράμετρος είναι παρούσα στο μήνυμα-ερώτηση (request). Στην τρίτη στήλη φαίνεται η ύπαρξη ή όχι των παραμέτρων στην απόκριση (response/confirmation). Λάθη μπορούν να προκληθούν, είτε εξαιτίας της ύπαρξης μη έγκυρων δεδομένων στο μήνυμα request, είτε εξαιτίας ενός προβλήματος που παρουσιάστηκε κατά την εκτέλεση της λειτουργίας που ζητήθηκε.



Σχήμα 1.17: *Μηνύματα στο CMIP*

Τα σύμβολα που χρησιμοποιούμε για να δηλώσουμε το status κάθε παραμέτρου είναι τρία: Υ, Ε, Σ. Το Υ δηλώνει την υποχρεωτική παρουσία της συγκεκριμένης παραμέτρου. Το Ε δηλώνει ότι είναι επιλογή του χρήστη (χρηστής μπορεί να είναι και μία εφαρμογή), αν θα περιλαμβάνεται αυτή η παράμετρος στο μήνυμα που στέλνεται. Απουσία μίας παραμέτρου που σημειώνεται με Ε δε θεωρείται λάθος. Το Σ δηλώνει ότι η παρουσία μίας παραμέτρου καθορίζεται από την εκπλήρωση μίας συνθήκης. Μερικές από τις παραμέτρους της τρίτης στήλης μπορεί να ακολουθούνται από το σύμβολο της ισότητας (=). Αυτό δηλώνει ότι η τιμή της παραμέτρου στο μήνυμα-απόκριση πρέπει να είναι ίδια με αυτή που βρίσκεται στο μήνυμα-ερώτηση. Η παύλα δηλώνει ότι η συγκεκριμένη παράμετρος δεν υπάρχει σε αυτό τον τύπο μηνύματος. Πρέπει να σημειωθεί ότι οι παράμετροι που θα παρουσιαστούν *δεν* αντιστοιχούν υποχρεωτικά σε κάποιο πεδίο του PDU που ανταλλάσσουν ο πράκτορας με το διαχειριστή. Στην πραγματικότητα δηλώνουν απλώς τα δεδομένα που πρέπει να παρέχονται, ώστε να πραγματοποιηθεί η λειτουργία που μας ενδιαφέρει.

- **Η υπηρεσία M-Event Report**

Η υπηρεσία M-Event Report χρησιμοποιείται για την αναφορά ενός γεγονότος. Πρόκειται για ένα μήνυμα που αποστέλλεται από τα διαχειριζόμενα αντικείμενα προς το σύστημα διαχείρισης. Πιο συγκεκριμένα λέμε ότι τα διαχειριζόμενα αντικείμενα στέλνουν στους πράκτορες notifications, τα οποία οι πράκτορες με τη σειρά τους στέλνουν στο διαχειριστή σαν M-Event-Reports. Τα Event Reports μπορούν να χρησιμοποιηθούν για να αναφερθεί η αλλαγή κατάστασης ενός διαχειριζόμενου αντικειμένου ή ένα λάθος που συνέβηκε. Η συγκεκριμένη υπηρεσία μπορεί να χρησιμοποιηθεί με ή χωρίς επιβεβαίωση.

Ο πίνακας 1.2 περιέχει τις παραμέτρους τις υπηρεσίας M-Event Report. Η παράμετρος *invoke identifier* χρησιμοποιείται ως αναγνωριστικό, ώστε να είναι δυνατό να αναφερόμαστε με αυτή στο συγκεκριμένο notification. Όπως βλέπουμε η συγκεκριμένη παράμετρος βρίσκεται και στο μήνυμα της επιβεβαίωσης και πρέπει να έχει την ίδια τιμή που είχε στο αρχικό μήνυμα. Με τον τρόπο αυτό επιτυγχάνεται η συσχέτιση αρχικού μηνύματος (request) – επιβεβαίωσης, με αποτέλεσμα να μπορούν να σταλούν πολλά requests ταυτόχρονα. Αν δεν υπήρχε η παράμετρος invoke identifier, θα έπρεπε κάθε φορά που στέλνεται ένα event report να αναμένεται η επιβεβαίωσή του πριν σταλεί ένα άλλο, καθώς αν αποστέλλονταν δύο μαζί δεν θα υπήρχε δυνατότητα να βρούμε ποια επιβεβαίωση αντιστοιχεί σε κάθε event report.

Όνομα παραμέτρου	**Req/Ind**	**Rsp/Conf**
Invoke identifier	Y	Y(=)
Mode	Y	-
Managed object class	Y	E
Managed object instance	Y	E
Event type	Y	Σ(=)
Event time	E	-
Event information	E	-
Current time	-	E
Event reply	-	Σ
Errors	-	Σ

Πίνακας 1.2: *Παράμετροι της υπηρεσίας M-Event Report*

Η παράμετρος *mode* καθορίζει τις δύο μορφές με τις οποίες μπορεί να χρησιμοποιηθεί η υπηρεσία: α) με επιβεβαίωση β) χωρίς επιβεβαίωση.

Η παράμετρος *managed object class* περιέχει την κλάση που ανήκει το διαχειριζόμενο αντικείμενο στο οποίο συνέβηκε το αναφερόμενο γεγονός.

Η παράμετρος *managed object instance* προσδιορίζει το συγκεκριμένο στιγμιότυπο της κλάσης στο οποίο συνέβηκε το αναφερόμενο γεγονός.

Η παράμετρος *event type* χρησιμοποιείται για τον προσδιορισμό του τύπου του γεγονότος που συνέβηκε. Παραδείγματα γεγονότων είναι ένας συναγερμός για διακοπή επικοινωνίας, η αλλαγή κατάστασης ενός αντικειμένου και η δημιουργία ενός αντικειμένου. Οι πιθανοί τύποι γεγονότων έχουν σχέση με την κλάση στην οποία αναφερόμαστε. Αν στην παράμετρο αυτή υπάρχει ένας τύπος γεγονότος που δεν ορίζεται για τη συγκεκριμένη κλάση, τότε θα αναφερθεί σφάλμα τύπου γεγονότος (στην περίπτωση που η υπηρεσία χρησιμοποιείται με επιβεβαίωση).

Η παράμετρος *event time* περιέχει την ώρα κατά την οποία συνέβηκε το γεγονός που αναφέρεται.

Η παράμετρος *event information* περιέχει πληροφορίες σχετικές με το γεγονός που συνέβηκε.

Η παράμετρος *current time* μπορεί να βρίσκεται στο μήνυμα της επιβεβαίωσης, αν φυσικά δεν παρουσιάστηκε κάποιο σφάλμα. Περιέχει την ώρα δημιουργίας της επιβεβαίωσης.

Η παράμετρος *event reply* περιέχει την απόκριση στο μήνυμα event report. Μπορεί να περιέχεται στην επιβεβαίωση όταν δεν υπάρχει κάποιο σφάλμα.

Η παράμετρος *errors* χρησιμοποιείται όταν στέλνεται επιβεβαίωση στο event report και έχει συμβεί κάποιο σφάλμα. Υπάρχουν πολλοί τύποι σφαλμάτων που μπορούν να συμβούν. Στη συνέχεια αναφέρουμε τα σφάλματα που μπορούν να συμβούν κατά τη λειτουργία M-Event Report. Τα σφάλματα των άλλων λειτουργιών είναι παρόμοια με αυτά της M-Event Report. Φυσικά, σε καθεμία από αυτές ορίζονται επιπλέον σφάλματα ή αφαιρούνται κάποια από αυτά που αναφέρουμε, ανάλογα με τις παραμέτρους που διαθέτει.

- Duplicate invocation: η παράμετρος invoke identifier που χρησιμοποιήθηκε είναι ήδη δεσμευμένη από κάποιο άλλο notification ή λειτουργία.

- Invalid argument value: η τιμή που δόθηκε στην παράμετρο event information ήταν μη επιτρεπτή.

- Mistyped argument: δεν είχε συμφωνηθεί κατά τη δημιουργία της σύνδεσης, η ύπαρξη μίας από τις παραμέτρους που βρίσκονταν στο μήνυμα.

- No such argument: η πληροφορία που βρίσκονταν στη παράμετρο event information ήταν μη κατανοητή.

- No such event type: στην παράμετρο event type υπάρχει κάποιος τύπος γεγονότος που δεν είναι αναγνωρίσιμος.

- No such object class: η κλάση που περιέχεται στην παράμετρο managed object class δεν είναι αναγνωρίσιμη.

- No such object instance: το στιγμιότυπο που περιέχεται στην παράμετρο managed object instance είναι μη αναγνωρίσιμο.

- Processing failure: παρουσιάστηκε σφάλμα κατά την επεξεργασία του μηνύματος.

- Resource limitation: δεν έγινε επεξεργασία του μηνύματος, εξαιτίας περιορισμένων διαθέσιμων πόρων.

- Unrecognized operation: ύπαρξη λειτουργίας που δεν είχε συμφωνηθεί κατά τη δημιουργία της σύνδεσης.

➢ **Η υπηρεσία M-Get**

Η υπηρεσία get χρησιμοποιείται για την ανάκληση των τιμών των γνωρισμάτων από ένα ή περισσότερα αντικείμενα. Το αντίστοιχο μήνυμα CMIP στέλνεται από το διαχειριστή στους πράκτορες. Η ύπαρξη επιβεβαίωσης-απάντησης στη συγκεκριμένη υπηρεσία είναι υποχρεωτική. Στον πίνακα 1.3 περιέχονται οι παράμετροι της υπηρεσίας M-Get.

Η παράμετρος *scope* χρησιμοποιείται για τον ορισμό ενός υποδέντρου της MIB, στο οποίο θα εκτελεστεί η λειτουργία που δηλώνει το μήνυμα. Η λειτουργία μπορεί να εκτελεστεί σε ολόκληρο το υποδέντρο ή σε συγκεκριμένους κόμβους του. Πιο συγκεκριμένα η λειτουργία μπορεί να εκτελεστεί:

- μόνο για το Base Object

- για το ν-οστό επίπεδο κάτω από το Base Object

- για το Base Object και το υποδέντρο του μέχρι και το ν-οστό επίπεδο (μαζί με αυτό)

- για το Base object και ολόκληρο το υποδέντρο του

Οι παράμετροι *base object class* και b*ase object instance* ορίζουν την κλάση και το στιγμιότυπο του αντικειμένου που είναι η ρίζα του υποδέντρου για την εφαρμογή της παραμέτρου scope.

Η παράμετρος *linked identifier* συνδέεται με την παράμετρο scope. Αν κάνουμε χρήση της παραμέτρου scope θα πρέπει να αναμένουμε πολλά μηνύματα επιβεβαίωσης. Τα μηνύματα αυτά θα έχουν όλα τον ίδιο invoke identifier. Ο linked identifier είναι το αναγνωριστικό για να ξεχωρίζουμε αυτά τα μηνύματα μεταξύ τους.

Η παράμετρος *filter* χρησιμοποιείται σε συνδυασμό με την παράμετρο scope. Συγκεκριμένα, ορίζει την εκτέλεση κάποιου ελέγχου πριν την πραγματοποίηση της λειτουργίας που υπαγορεύει το μήνυμα. Η λειτουργία αυτή εκτελείται μόνο στα αντικείμενα που το αποτέλεσμα του ελέγχου είναι «ΑΛΗΘΕΣ». Αντίθετα, δεν εκτελείται σε όσα αντικείμενα ο έλεγχος δώσει αποτέλεσμα «ΨΕΥΔΕΣ».

Η παράμετρος *access control* δεν ορίζεται λεπτομερώς από το CMIS. Στην πραγματικότητα, εδώ αποθηκεύονται πληροφορίες που έχουν σχέση με την ασφάλεια. Η ακριβής τους μορφή εξαρτάται από το μηχανισμό που χρησιμοποιείται.

Η παράμετρος *synchronization* χρησιμοποιείται σε συνδυασμό με την παράμετρο scope. Ορίζει τον τρόπο με τον οποίο συγχρονίζονται οι ανακλήσεις τιμών πολλών αντικειμένων. Πιο συγκεκριμένα ορίζει ότι οι ανακλήσεις μπορούν να γίνονται με δύο τρόπους που περιγράφονται στη συνέχεια.

-Atomic: Ελέγχεται αν είναι δυνατό να πραγματοποιηθούν όλες οι ανακλήσεις δεδομένων που πρέπει. Αν είναι, τότε πραγματοποιούνται. Αν έστω και μία ανάκληση δεν μπορεί να γίνει, τότε δεν πραγματοποιείται καμία.

-Best effort: Δοκιμάζεται αν μπορούν να πραγματοποιηθούν οι επιθυμητές ανακλήσεις δεδομένων. Κάθε ανάκληση που μπορεί να γίνει, πραγματοποιείται ανεξάρτητα από το αν οι άλλες μπορούν. Αν η παράμετρος Synchronization δε χρησιμοποιείται, τότε ο τρόπος συγχρονισμού είναι ο best effort.

Όνομα παραμέτρου	**Req/Ind**	**Rsp/Conf**
Invoke identifier	Y	Y(=)
Linked identifier	-	Σ
Base object class	Y	-
Base object instance	Y	-
Scope	E	-
Filter	E	-
Access control	E	-
Synchronization	E	-
Attribute identifier list	E	-
Managed object class	-	Σ
Managed object instance	-	Σ
Attribute list	-	Σ
Current time	-	E
Errors	-	Σ

Πίνακας 1.3: *Παράμετροι της υπηρεσίας M-Get*

Η παράμετρος *attribute identifier list* περιέχει ένα σύνολο από αναγνωριστικά γνωρισμάτων, από τα οποία θέλουμε να ανακτήσουμε τις τιμές. Αν η παράμετρος αυτή δε χρησιμοποιείται, ανακτώνται οι τιμές όλων των γνωρισμάτων ενός αντικειμένου.

Οι παράμετροι *managed object class* και *managed object instance* βρίσκονται στο μήνυμα επιβεβαίωσης και δηλώνουν την κλάση και το στιγμιότυπο του αντικειμένου από το οποίο στάλθηκε η επιβεβαίωση. Μπορούν να βρίσκονται και στην επιβεβαίωση ορθής εκτέλεσης της λειτουργίας και στην επιβεβαίωση σφάλματος.

Η παράμετρος *attribute list* περιέχει τα αναγνωριστικά και τις τιμές των γνωρισμάτων, μετά την εκτέλεση της λειτουργίας που ζητήθηκε. Περιέχεται στην επιβεβαίωση ορθής εκτέλεσης της λειτουργίας.

Οι παράμετροι *invoke identifier*, *current time* και *errors* έχουν ήδη αναλυθεί.

- **Η υπηρεσία M-Set**

Η υπηρεσία M-Set χρησιμοποιείται όταν δίνεται εντολή για την αλλαγή τις τιμής ενός γνωρίσματος ενός αντικειμένου. Η συγκεκριμένη υπηρεσία μπορεί να χρησιμοποιηθεί, είτε με επιβεβαίωση, είτε χωρίς. Το αντίστοιχο μήνυμα CMIP στέλνεται από το διαχειριστή στους πράκτορες. Στον πίνακα 1.4 περιέχονται οι παράμετροι της υπηρεσίας M-Set.

Όνομα παραμέτρου	**Req/Ind**	**Rsp/Conf**
Invoke identifier	Y	Y(=)
Linked identifier	-	Σ
Mode	Y	-
Base object class	Y	-
Base object instance	Y	-
Scope	E	-
Filter	E	-
Access control	E	-
Synchronization	E	-
Managed object class	-	Σ
Managed object instance	-	Σ
Attribute list	Y	Σ
Current time	-	E
Errors	-	Σ

Πίνακας 1.4: *Παράμετροι της υπηρεσίας M-Set*

Οι παράμετροι της M-Set έχουν την ίδια σημασιολογία με αυτές της M-Get που αναλύσαμε παραπάνω. Διαφορετική είναι μόνο η χρήση της παραμέτρου *attribute list*, η οποία όπως βλέπουμε βρίσκεται και στο μήνυμα που αποστέλλεται αρχικά και στο μήνυμα της επιβεβαίωσης. Στο αρχικό μήνυμα, η παράμετρος αυτή περιέχει ένα σύνολο από αναγνωριστικά γνωρισμάτων και τις τιμές που επιθυμούμε να πάρουν τα

γνωρίσματα αυτά. Στο μήνυμα επιβεβαίωσης, η λειτουργία της παραμέτρου αυτής είναι ίδια με αυτή που αναλύσαμε στην υπηρεσία M-Get.

- **Η υπηρεσία M-Action**

Η υπηρεσία M-Action χρησιμοποιείται όταν δίνεται εντολή για την εκτέλεση μίας πράξης πάνω σε ένα αντικείμενο. Η πράξη αυτή πρέπει να ορίζεται στο πληροφοριακό μοντέλο, δηλαδή πρέπει να υπάρχει στον ορισμό της κλάσης στην οποία ανήκει το συγκεκριμένο αντικείμενο. Η συγκεκριμένη υπηρεσία μπορεί να χρησιμοποιηθεί, είτε με επιβεβαίωση, είτε χωρίς. Ένα χαρακτηριστικό που συναντάμε σε αυτή την υπηρεσία, και όχι στις άλλες, είναι ότι μπορεί να στέλνονται διαδοχικές επιβεβαιώσεις. Για παράδειγμα, αν η πράξη που ζητήθηκε να γίνει πάνω σε ένα διαχειριζόμενο αντικείμενο είναι ένας έλεγχος που αποτελείται από πολλά βήματα, μπορεί η υπηρεσία M-Action να έχει προγραμματιστεί να στέλνει μία επιβεβαίωση για κάθε βήμα. Το αντίστοιχο μήνυμα της λειτουργίας M-Action στο CMIP στέλνεται από το διαχειριστή στους πράκτορες. Στον πίνακα 1.5 περιέχονται οι παράμετροι της υπηρεσίας M-Action.

Όνομα παραμέτρου	**Req/Ind**	**Rsp/Conf**
Invoke identifier	Y	Y(=)
Linked identifier	-	Σ
Mode	Y	-
Base object class	Y	-
Base object instance	Y	-
Scope	E	-
Filter	E	-
Managed object class	-	Σ
Managed object instance	-	Σ
Access control	E	-
Synchronization	E	-
Action type	Y	Σ(=)
Action information	E	-
Current time	-	E
Action reply	-	Σ
Errors	-	Σ

Πίνακας 1.5: *Παράμετροι της υπηρεσίας M-Action*

Οι περισσότερες από τις παραμέτρους του πίνακα 1.5 έχουν αναλυθεί παραπάνω. Εξαίρεση αποτελούν οι παράμετροι action type, action information και action reply, με τις οποίες θα ασχοληθούμε εδώ.

Η παράμετρος *action type* περιέχει τον τύπο δράσης στον οποίο αναφερόμαστε. Αν στην επιβεβαίωση περιέχεται η παράμετρος action reply, τότε περιέχεται υποχρεωτικά και η action type. Σε διαφορετική περίπτωση μπορεί και να μην περιέχεται.

Η παράμετρος *action information* περιέχει επιπλέον πληροφορίες, που είναι απαραίτητες, για τον καθορισμό της φύσης και των τελουμένων της συγκεκριμένης δράσης. Τα συντακτικό και η σημασιολογία της συγκεκριμένης παραμέτρου καθορίζοντα από την ίδια τη δράση.

Η παράμετρος *action reply* περιέχει την απάντηση που στέλνεται για το μήνυμα action. Μπορεί να περιέχεται στην επιβεβαίωση για το μήνυμα action, αν δεν έχει παρουσιαστεί κάποιο σφάλμα κατά τη διάρκεια εκτέλεσης της M-Action.

- **Η υπηρεσία M-Create**

Η υπηρεσία M-Create χρησιμοποιείται για τη δημιουργία ενός στιγμιοτύπου κάποιου διαχειριζόμενου αντικείμενου. Το στιγμιότυπο αυτό ακολουθεί το σχήμα μίας κλάσης. Επιπλέον, η M-Create είναι υπεύθυνη για την σύνδεση του στιγμιοτύπου με ένα μοναδικό αναγνωριστικό και συνακόλουθα με ένα κόμβο στη MIB, καθώς επίσης και για την απόδοση αρχικών τιμών στα γνωρίσματα, όπου αυτό απαιτείται. Η συγκεκριμένη υπηρεσία χρησιμοποιείται μόνο με επιβεβαίωση.

Στον πίνακα 1.6 παρουσιάζονται οι παράμετροι της M-Create. Η παράμετρος invoke identifier έχει την ίδια σημασία που αναφέραμε και στις άλλες υπηρεσίες.

Η παράμετρος *managed object class*, περιέχει την κλάση στην οποία ανήκει το δημιουργούμενο στιγμιότυπο.

Η παράμετρος *managed object instance* καθορίζει το όνομα του στιγμιοτύπου και το αναγνωριστικό του. Αν δεν υπάρχει στο μήνυμα ούτε αυτή η παράμετρος, αλλά ούτε και η παράμετρος superior object instance, τότε το σύστημα είναι αυτό που δίνει ένα αναγνωριστικό στο συγκεκριμένο στιγμιότυπο. Στην περίπτωση αυτή, αν το στιγμιότυπο δημιουργηθεί επιτυχώς, η επιβεβαίωση περιέχει την παράμετρο. Σε αντίθετη περίπτωση η παρουσία της παραμέτρου στην επιβεβαίωση δεν είναι απαραίτητη.

Όνομα παραμέτρου	**Req/Ind**	**Rsp/Conf**
Invoke identifier	Y	Y(=)
Managed object class	Y	Σ
Managed object instance	E	Σ
Superior object instance	E	-
Access control	E	-
Reference object instance	E	-
Attribute list	E	Σ
Current time	-	E
Errors	-	Σ

Πίνακας 1.6: *Παράμετροι της υπηρεσίας M-Create*

Η παράμετρος *superior object instance* χρησιμοποιείται για να δηλώσει το υπάρχον στιγμιότυπο, το οποίο ιεραρχικά βρίσκεται πάνω από το στιγμιότυπο που δημιουργείται. Αν η παράμετρος αυτή υπάρχει στο αρχικό μήνυμα, τότε η παράμετρος managed object instance δεν θα υπάρχει.

Η παράμετρος *attribute list* περιέχει ένα σύνολο από αναγνωριστικά γνωρισμάτων, συνοδευόμενα από τις τιμές που θα έχουν τα αντίστοιχα γνωρίσματα στο νέο στιγμιότυπο. Η επιβεβαίωση μπορεί να περιέχει αυτή την παράμετρο, αν η δημιουργία του στιγμιοτύπου είναι επιτυχής. Στην περίπτωση αυτή η attribute list περιέχει τα αναγνωριστικά και τις τιμές όλων των γνωρισμάτων (και αυτών που περιέχουν default τιμές).

Η παράμετρος *reference object instance*, περιέχει ένα στιγμιότυπο αντικειμένου που ανήκει στην ίδια κλάση με το δημιουργούμενο. Αν στην attribute list του αρχικού μηνύματος δεν περιέχονται κάποιες τιμές γνωρισμάτων, τότε αυτές τίθενται στην τιμή του αντίστοιχου γνωρίσματος του reference object instance. Οι υπόλοιπες παράμετροι της υπηρεσίας έχουν αναλυθεί παραπάνω.

- **Η υπηρεσία M-Delete**

Η υπηρεσία M-Delete χρησιμοποιείται για τη διαγραφή ενός στιγμιοτύπου κάποιου διαχειριζόμενου αντικειμένου και για τη συνακόλουθη αποδέσμευση του αναγνωριστικού του. Το αντίστοιχο μήνυμα στο CMIP στέλνεται από το διαχειριστή στον πράκτορα. Η συγκεκριμένη υπηρεσία χρησιμοποιείται με επιβεβαίωση.

Όνομα παραμέτρου	**Req/Ind**	**Rsp/Conf**
Invoke identifier	Y	Y
Linked identifier	-	Σ
Base object class	Y	-
Base object instance	Y	-
Scope	E	-
Filter	E	-
Access control	E	-
Synchronization	E	-
Managed object class	-	Σ
Managed object instance	-	Σ
Current time	-	E
Errors	-	Σ

Πίνακας 1.7: *Παράμετροι της υπηρεσίας M-Delete*

Η σημασιολογία των παραμέτρων της υπηρεσίας M-Delete είναι ίδια με των άλλων υπηρεσιών και έχει αναλυθεί παραπάνω.

- **Η υπηρεσία M-Cancel Get**

Η υπηρεσία M-Cancel Get χρησιμοποιείται για την ακύρωση μίας υπηρεσίας M-Get, η οποία είχε ζητηθεί προηγουμένως. Το αντίστοιχο μήνυμα στο CMIP στέλνεται από το διαχειριστή στον πράκτορα. Η συγκεκριμένη υπηρεσία χρησιμοποιείται με επιβεβαίωση. Για να κατανοήσουμε τη χρησιμότητά της, αρκεί να σκεφτούμε δύο

παραδείγματα. Στο πρώτο, ο διαχειριστής στέλνει μήνυμα get για την ανάκληση δεδομένων από πολλά αντικείμενα, αλλά η ανάκληση αυτή παίρνει υπερβολικά πολύ χρόνο. Στο δεύτερο παράδειγμα, τα δεδομένα που τελικά είναι απαραίτητα στο διαχειριστή συγκεντρώνονται χωρίς να ληφθούν απαντήσεις από όλα τα αντικείμενα. Προφανώς, επιπλέον απαντήσεις μόνο πρόσθετο φόρτο θα προκαλέσουν.

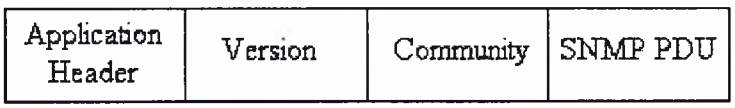
Όνομα παραμέτρου	**Req/Ind**	**Rsp/Conf**
Invoke identifier	Y	Y(=)
Get invoke identifier	Y	-
Errors	-	Σ

Πίνακας 1.8: *Παράμετροι της υπηρεσίας M-Cancel-Get*

Στον πίνακα 1.8 φαίνονται οι παράμετροι τις υπηρεσίας M-Cancel Get. Η παράμετρος get invoke identifier περιέχει το αναγνωριστικό της υπηρεσίας Get που ζητείται να ακυρωθεί.

1.2.3.2 Τα μηνύματα στο SNMP [1], [7], [12]

Το SNMP είναι ένα πρωτόκολλο επιπέδου εφαρμογής, που λειτουργεί πάνω από δίκτυα UDP/IP. Υπάρχουν τρεις εκδόσεις του πρωτοκόλλου η SNMPv1, η SNMPv2 και η SNMPv3. Η SNMPv1 ορίζει πέντε τύπους μηνυμάτων: get-request, get-next-request, set-request, get-response και trap. Στη δεύτερη έκδοση του πρωτοκόλλου ορίζονται δύο επιπλέον μηνύματα: το μήνυμα inform και το μήνυμα get-bulk-request. Στην τρίτη έκδοση του πρωτοκόλλου δεν ορίζονται νέοι τύποι μηνυμάτων. Όλα τα μηνύματα του SNMP λαμβάνονται στη θύρα 161. Εξαίρεση αποτελεί το μήνυμα trap που λαμβάνεται στη θύρα 162.



Σχήμα 1.18: *Ένα μήνυμα SNMP στο επίπεδο εφαρμογής*

Στο σχήμα 1.18 φαίνεται η μορφή που έχει ένα μήνυμα SNMP στο επίπεδο εφαρμογής. Πέρα από το πεδίο που αποτελεί την κεφαλίδα που προστίθεται στο επίπεδο εφαρμογής (*Application Header*), βλέπουμε ότι το μήνυμα περιέχει τρία ακόμα πεδία.

Το πεδίο *Version* χρησιμοποιείται για να δηλωθεί η έκδοση του πρωτοκόλλου. Αν ένα μήνυμα SNMP συγκεκριμένης έκδοσης καταλήξει σε παραλήπτη που δεν την υποστηρίζει, τότε το μήνυμα αυτό απορρίπτεται.

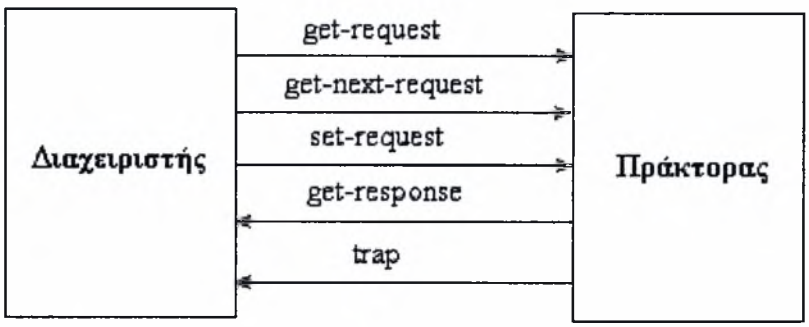
Το πεδίο *Community* χρησιμοποιείται για να δηλωθεί η κοινότητα στην οποία ανήκει ο αποστολέας. Στην έννοια της κοινότητας θα αναφερθούμε αναλυτικότερα όταν αναλύσουμε την ασφάλεια στο SNMP. Σε αυτό το σημείο θα πούμε μόνο, ότι

για να επικοινωνήσουν ένας πράκτορας με ένα διαχειριστή, θα πρέπει και οι δύο να ανήκουν στην ίδια κοινότητα.

Τέλος, το πεδίο *SNMP PDU* είναι στην ουσία το μήνυμα SNMP. Στη συνέχεια θα αναφερθούμε στον τρόπο με τον οποίο χρησιμοποιείται το κάθε μήνυμα. Επίσης, θα αναλύσουμε τη μορφή που έχει το πεδίο SNMP PDU ανάλογα με τον τύπο του μηνύματος στον οποίο βρίσκεται.

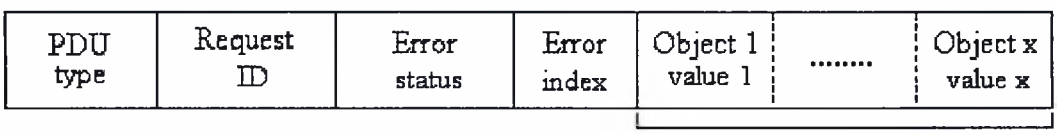
Τα μηνύματα στο SNMPv1

Στο σχήμα 1.19 παρουσιάζονται τα μηνύματα στο SNMPv1. Η αρχή του βέλους δηλώνει τον αποστολέα του μηνύματος και το τέλος του τον παραλήπτη. Στο κείμενο που ακολουθεί εξηγείται η λειτουργία κάθε μηνύματος.



Σχήμα 1.19: *Τα μηνύματα στο SNMPv1*

Get-request: Το συγκεκριμένο μήνυμα αποστέλλεται από το διαχειριστή σε ένα πράκτορα. Ο διαχειριστής ζητά από τον πράκτορα να του στείλει τις τιμές από διαχειριζόμενα αντικείμενα που βρίσκονται στη MIB του δεύτερου. Το PDU του μηνύματος get-request παρουσιάζεται στο σχήμα 1.20.



Σχήμα 1.20: Get, get-next, set, και get-response PDU

Το πεδίο *PDU type* ορίζει τον τύπο του μηνύματος στον οποίο αναφερόμαστε. Το πεδίο *Request ID* είναι ένα αναγνωριστικό, το οποίο χρησιμοποιείται για τη συσχέτιση του συγκεκριμένου μηνύματος με την απάντηση που θα ληφθεί.

Το πεδίο *Error status* χρησιμοποιείται για την αποθήκευση ενός ακεραίου που δηλώνει ότι συνέβηκε κάποιο σφάλμα.

Το πεδίο *Error index* χρησιμοποιείται για την παροχή επιπλέον πληροφοριών σχετικά με το λάθος που προκλήθηκε. Στην πραγματικότητα το πεδίο αυτό

συμπληρώνεται μόνο κατά την αποστολή του μηνύματος get-response. Όλα τα άλλα μηνύματα θέτουν την τιμή NULL στο πεδίο αυτό.

Όπως είπαμε, στο SNMP ένα διαχειριζόμενο αντικείμενο είναι στην ουσία μία μεταβλητή, η οποία έχει μία τιμή. Τα επόμενα πεδία στο σχήμα 1.20 δηλώνουν ακριβώς αυτό, δηλαδή μία μεταβλητή και την τιμή που της αντιστοιχεί. Η τιμή της μεταβλητής αγνοείται στα μηνύματα get-request και get-next-request.

Το PDU το σχήματος 1.20 ισχύει για όλα τα μηνύματα στο SNMPv1, εκτός του trap.

Get-next-request: Το μήνυμα get-next-request στέλνεται από το διαχειριστή στον πράκτορα. Η λειτουργία του είναι όμοια με αυτή του μηνύματος get-next, με τη διαφορά ότι ο διαχειριστής ζητά την τιμή του αντικειμένου που είναι το επόμενο από αυτό του οποίου ο OBJECT IDENTIFIER περιέχεται στο μήνυμα.

Το μήνυμα αυτό είναι ιδιαίτερα χρήσιμο όταν ο διαχειριστής θέλει να μάθει τις τιμές aggregate objects, δηλαδή αντικειμένων που βρίσκονται στον ίδιο πίνακα, των οποίων δεν ξέρει το πλήθος. Στην περίπτωση αυτή μπορεί να στείλει αρχικά στον πράκτορα ένα μήνυμα get-next-request με το OBJECT IDENTIFIER ολόκληρου του πίνακα. Στην απάντηση που θα λάβει θα περιέχεται το OBJECT IDENTIFIER και η τιμή του πρώτου αντικειμένου που βρίσκεται στον πίνακα. Στη συνέχεια με διαδοχικά get-next-request, όπου στο καθένα θα περιέχεται το OBJECT IDENTIFIER από την προηγούμενη απάντηση που του είχε έρθει, ο διαχειριστής μπορεί να μάθει τις τιμές όλων των αντικειμένων του πίνακα.

Set- request: Το μήνυμα set-request στέλνεται από το διαχειριστή στον πράκτορα Χρησιμοποιείται όταν ο διαχειριστής στέλνει εντολή στον πράκτορα να θέσει μία τιμή σε μία συγκεκριμένη μεταβλητή.

Get-response: Το μήνυμα get- response αποστέλλεται από τον πράκτορα στο διαχειριστή σαν απάντηση στα μηνύματα get-request και get-next-request. Περιέχει τις τιμές των μεταβλητών που είχε ζητήσει ο διαχειριστής.

Trap: Το μήνυμα trap αποστέλλεται από τον πράκτορα στο διαχειριστή για να τον ενημερώσει για κάποιο γεγονός που συνέβηκε. Το PDU του μηνύματος trap είναι διαφορετικό από αυτό των άλλων μηνυμάτων και παρουσιάζεται στο σχήμα 1.21. Στο κείμενο που ακολουθεί εξηγείται η σημασία των πεδίων του trap PDU.

PDU Type	Enterprise	Agent address	Generic trap type	Specific trap code	Time stamp	Object 1 Value 1	.....	Object x Value x

Σχήμα 1.21: SNMPv1 trap PDU

Η σημασία του πεδίου *PDU Type* έχει ήδη εξηγηθεί, ενώ το πεδίο *Enterprise* καθορίζει το αντικείμενο από το οποίο προέρχεται το trap.

Το πεδίο *Agent address* περιέχει τη διεύθυνση του αντικειμένου στο οποίο «κατοικεί» ο πράκτορας που έστειλε το trap.

Το πεδίο *Generic trap type* περιέχει τον τύπο του trap που στέλνεται. Ορίζονται επτά τύποι generic traps:

- coldStart trap (0): Το trap αυτό δηλώνει ότι η οντότητα που το έστειλε επαναρχικοποιήθηκε με τρόπο που μπορεί να άλλαξε τις υπάρχουσες ρυθμίσεις.

- warmStart trap (1): Το trap αυτό δηλώνει ότι η οντότητα που το έστειλε επαναρχικοποιήθηκε, αλλά με τρόπο που δεν επηρέασε τις υπάρχουσες ρυθμίσεις.

- linkDown trap (2): Το trap αυτό δηλώνει ότι η οντότητα που το έστειλε δεν «βλέπει» πλέον μία από τις ζεύξεις που «έβλεπε» πριν.

- linkUp trap (3): Το trap αυτό δηλώνει ότι η οντότητα που το έστειλε «βλέπει» ξανά μία από τις ζεύξεις που είχε χάσει.

- authenticationFailure trap (4): Το trap αυτό δηλώνει ότι η οντότητα που το έστειλε έλαβε ένα μήνυμα το οποίο δεν κατάφερε να αυθεντικοποιηθεί.

- egpNeighborLoss trap (5): Το trap αυτό δηλώνει ότι η οντότητα που το έστειλε «έχασε» έναν από τους EGP γείτονές της.

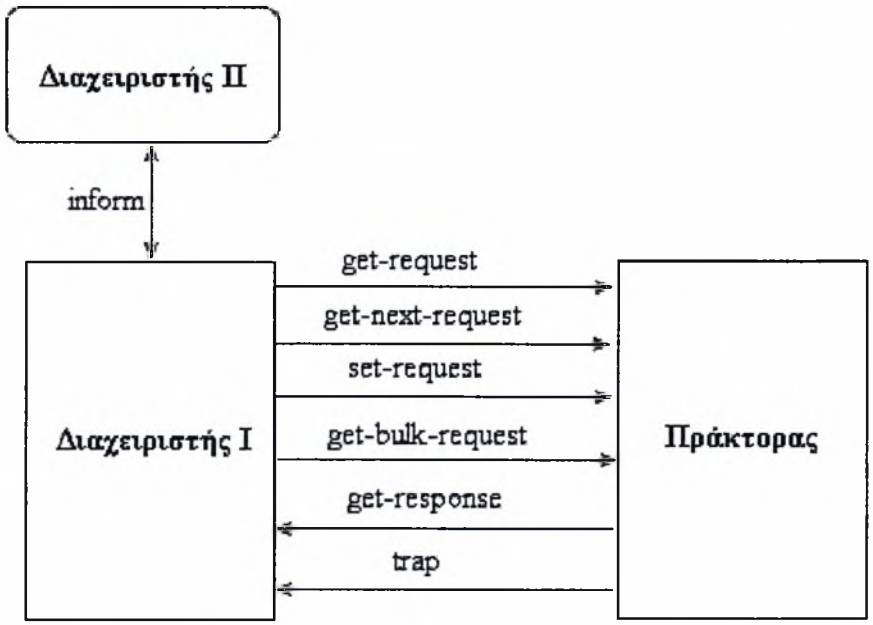
- EnterpriseSpecific trap (6): Το trap trap αυτό δηλώνει ότι έχει συμβεί ένα άλλο γεγονός που δεν ανήκει στις προηγούμενες κατηγορίες. Το γεγονός αυτό δηλώνεται στο πεδίο *Specific trap code.*

Το πεδίο *Time stamp* δηλώνει το χρόνο που πέρασε από την τελευταία αρχικοποίηση του αντικειμένου από το οποίο στάλθηκε το μήνυμα.

Τέλος στα πεδία *Object type* και *Value*, περιέχονται αντικείμενα και οι τιμές τους, τη στιγμή που στάλθηκε το trap

Τα μηνύματα στο SNMPv2

Στο σχήμα 1.22 παρουσιάζονται το μηνύματα που υπάρχουν στο SNMPv2. Παρατηρούμε ότι υπάρχουν μόνο δύο νέα μηνύματα σε σχέση με το SNMPv1: το get-bulk-request και το inform. Στη συνέχεια θα αναλύσουμε μόνο αυτά τα μηνύματα, καθώς τα υπόλοιπα λειτουργούν με τον ίδιο τρόπο που λειτουργούσαν και στην πρώτη έκδοση του πρωτοκόλλου.



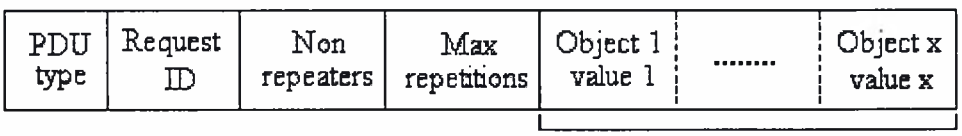
Σχήμα 1.22: Τα μηνύματα στα SNMPv2 και SNMPv3

Get-bulk-request: Το μήνυμα get-bulk-request στέλνεται από το διαχειριστή στον πράκτορα. Η λειτουργία του είναι όμοια με αυτή του μηνύματος get, με τη διαφορά ότι μπορεί να χρησιμοποιηθεί για την ανάκληση ογκωδών δεδομένων. Επισημαίνουμε ότι το CMIP προσέφερε αυτή τη δυνατότητα και ότι αυτό ήταν ένα από τα σημεία στα οποία υπερτερούσε έναντι του SNMP. Με το μήνυμα αυτό μπορεί να επιταχυνθεί κατά πολύ η διαδικασία ανάκλησης δεδομένων από πίνακα με την εντολή get-next, όπως αυτή περιγράφηκε παραπάνω. Στην πραγματικότητα αυτό που κάνει το συγκεκριμένο μήνυμα είναι να ζητά από τον πράκτορα να στείλει τις τιμές για περισσότερα του ενός αντικείμενα. Σημειώνεται ότι με το ίδιο μήνυμα μπορεί να ζητηθούν οι τιμές και από aggregate και από μη aggregate objects. Στο σχήμα 1.23 παρουσιάζονται τα πεδία του get-bulk-request PDU.

Η σημασία κάποιων από τα πεδία του get-bulk-request PDU έχει εξηγηθεί όταν αναφερθήκαμε σε άλλα μηνύματα. Τα δύο πεδία τα οποία δεν έχουμε ξανασυναντήσει μέχρι τώρα είναι το Non repeaters και το Max repetitions.

Το πεδίο *Non repeaters* χρησιμοποιείται για να οριστούν τα μη aggregate αντικείμενα που ζητά ο διαχειριστής.

Το πεδίο *Max repetitions* χρησιμοποιείται για να δηλώσει το μέγιστο αριθμό φορών που ο διαχειριστής θέλει να γίνει ανάκληση στα aggregate αντικείμενα, δηλαδή σε όσα δε δηλώθηκαν στο πεδίο Non repeaters.

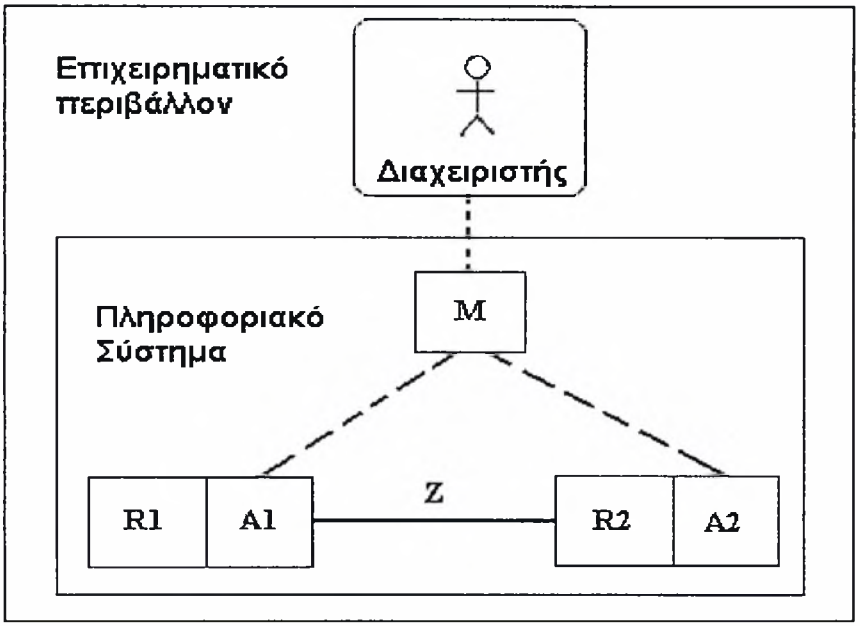


Σχήμα 1.23: Get-bulk-request PDU

Inform: Το μήνυμα inform αποστέλλεται από ένα διαχειριστή σε ένα άλλο. Σκοπός του είναι να ενημερωθεί ο δεύτερος διαχειριστής για ένα γεγονός. Το μήνυμα αυτό δίνει τη δυνατότητα για τη δημιουργία οργανωτικών μοντέλων πολλών επιπέδων, στα οποία έχουμε αναφερθεί όταν αναλύσαμε το οργανωτικό μοντέλο. Τα πεδία του συγκεκριμένου μηνύματος είναι αυτά που φαίνονται στο σχήμα 1.20.

1.2.4 Λειτουργικό μοντέλο [1], [2]

Το λειτουργικό μοντέλο εισάγει τη σχέση που έχει ο χρήστης/διαχειριστής με ένα δίκτυο, θέτοντας απαιτήσεις, μεταξύ άλλων, σχετικά με την απόδοση του δικτύου. Επίσης, προχωρά ένα βήμα παραπέρα και τοποθετεί το διαχειριστή και όλο το πληροφοριακό σύστημα σε ένα επιχειρηματικό περιβάλλον. Το ζητούμενο είναι η αύξηση των εσόδων του οργανισμού στον οποίο ανήκει το σύστημα και η ανάλυσή του με βάση το κόστος. Όλα αυτά απεικονίζονται στο σχήμα 1.24.



Σχήμα 1.24: Πληροφοριακό σύστημα και επιχειρηματικό περιβάλλον

Πρέπει να σημειώσουμε ότι το λειτουργικό μοντέλο ορίζεται μόνο για το μοντέλο διαχείρισης OSI και ότι στο μοντέλο διαχείρισης internet το SNMP μπορεί να υλοποιήσει μόνο ένα περιορισμένο αριθμό από τις προβλεπόμενες λειτουργίες. Στη συνέχεια θα αναλύσουμε τα πέντε υπομοντέλα του λειτουργικού μοντέλου: τη διαχείριση διαμόρφωσης, τη διαχείριση σφαλμάτων, τη διαχείριση απόδοσης, τη διαχείριση ασφάλειας και τη διαχείριση κόστους.

1.2.4.1 Υπομοντέλα λειτουργικού μοντέλου [2], [1], [3]

- **Διαχείριση διαμόρφωσης**

Στη διαχείριση διαμόρφωσης ασχολούμαστε με το σχεδιασμό του δικτύου, την εγκατάστασή του και τη ρύθμιση των στοιχείων του. Η ρύθμιση των παραμέτρων των στοιχείων του δικτύου πρέπει να γίνεται έτσι ώστε έτσι ώστε να ικανοποιούνται οι απαιτήσεις του πελάτη. Η διαχείριση διαμόρφωσης μπορεί να αναλυθεί στις παρακάτω λειτουργίες:

- ✓ **Σχεδιασμός και μηχανική δικτύου – Network planning and engineering**

Η λειτουργία αυτή περιλαμβάνει το σχεδιασμό του δικτύου με τέτοιο τρόπο που να ικανοποιούνται οι απαιτήσεις που έχουν τεθεί. Πέρα από την τοπολογία του δικτύου και την απόφαση για την τεχνολογία που θα χρησιμοποιηθεί (δρομολογητές, πρωτόκολλα, τερματικά, μισθωμένες ή μη ζεύξεις), πρέπει να γίνει και κοστολόγηση της κατασκευής του δικτύου. Ο σχεδιασμός πρέπει να λαμβάνει υπόψη και επεκτάσεις που μπορεί να γίνουν στο μέλλον ή την παροχή νέων υπηρεσιών.

Για τα υπάρχοντα δίκτυα, η συγκεκριμένη λειτουργία αναφέρεται στις επεκτάσεις και τις αναβαθμίσεις που μπορούν να γίνουν και χρησιμοποιεί δεδομένα από τη διαχείριση αποδόσεων – για παράδειγμα για την απόφαση αναβάθμισης μίας γραμμής– και λαθών – για παράδειγμα για την αντικατάσταση παλιών μηχανημάτων.

- ✓ **Εγκατάσταση – Installation**

Η λειτουργία αυτή περιλαμβάνει την εγκατάσταση του δικτύου. Η εγκατάσταση μπορεί να αναφέρεται, τόσο σε εγκατάσταση υλικού όσο και σε εγκατάσταση λογισμικού. Επίσης, πρέπει να γίνει σαφές ότι με τον όρο εγκατάσταση δεν εννοούμε αναγκαστικά την εξαρχής δημιουργία ενός δικτύου. Μπορεί να αναφερόμαστε σε μία απλή επέκταση ενός υπάρχοντος δικτύου. Η συγκεκριμένη λειτουργία περιέχει και την παράδοση του έργου, οπότε πρέπει να γίνουν και οι απαραίτητοι έλεγχοι πριν την παράδοση. Οι έλεγχοι αυτοί δεν αναφέρονται μόνο στη δυνατότητα του δικτύου να λειτουργεί σα σύνολο, αλλά και στη δυνατότητα κάθε στοιχείου του δικτύου να λειτουργεί σύμφωνα με τις απαιτήσεις που έχουν τεθεί εξαρχής.

- ✓ **Σχεδιασμός υπηρεσιών και διαπραγματεύσεις – Service planning and negotiation**

Η λειτουργία αυτή έχει να κάνει με το σχεδιασμό νέων υπηρεσιών και την αναβάθμιση παλαιότερων. Για να σχεδιαστούν οι νέες υπηρεσίες, λαμβάνονται υπόψη οι απαιτήσεις τους σε πόρους και οι δυνατότητες που έχει το δίκτυο να τις καλύψει. Επίσης, μπορεί να αποφασιστεί η κατάργηση κάποιας παλιάς υπηρεσίας. Στη συγκεκριμένη λειτουργία συγκαταλέγονται και δράσεις που πρέπει να ληφθούν σε επιχειρηματικό επίπεδο. Το πρώτο βήμα για το σχεδιασμό μία υπηρεσίας είναι η πλήρης κατανόηση της ανάγκης που θα καλυφθεί από την υπηρεσία αυτή. Αν υπάρχουν πολλοί πιθανοί χρήστες της υπηρεσίας αυτής – οι οποίοι βέβαια είναι διατεθειμένοι να πληρώσουν – θα πρέπει να τους γνωστοποιήσουμε τη νέα υπηρεσία μέσω διαφήμισης. Επιπλέον η συγκεκριμένη λειτουργία περιλαμβάνει και επίλυση

νομικών θεμάτων που μπορεί να προκύψουν από τη δημιουργία μίας νέας υπηρεσίας. Για παράδειγμα, αν για να υλοποιηθεί η νέα υπηρεσία πρέπει να σκάψουμε σε δημόσιο χώρο για να εγκαταστήσουμε μία οπτική ίνα, τότε πρέπει να εξασφαλιστούν εκ των πρότερων οι απαραίτητες άδειες.

- ✓ **Προμήθεια – Provisioning**

Η λειτουργία αυτή αναφέρεται στην προμήθεια νέων υλικών πόρων (μηχανήματα, ανταλλακτικά) που θα χρησιμοποιηθούν στο δίκτυο. Για να γίνει μία σωστή εκτίμηση των μηχανημάτων που θα χρειαστούμε στο μέλλον, πρέπει να γίνει μελέτη των ήδη υπαρχόντων στατιστικών. Για παράδειγμα, μπορεί να έχουμε στατιστικά που μας λένε πόσο συχνά χρειάζεται αντικατάσταση ένας τύπος μηχανήματος, οπότε μπορούμε να εκτιμήσουμε πόσα τέτοια μηχανήματα θα χρειαστούμε μέσα στον επόμενο χρόνο και να τα προμηθευτούμε πριν συμβούν οι βλάβες. Κάτι τέτοιο θα εξοικονομούσε πολύτιμο χρόνο κατά την αποκατάσταση της βλάβης.

- ✓ **Κατάσταση και έλεγχος – Status and control**

Η λειτουργία αυτή σχετίζεται με τον έλεγχο και την καταγραφή της κατάστασης των στοιχείων του δικτύου, δηλαδή αν λειτουργούν κανονικά ή όχι. Επίσης με βάση τη λειτουργία αυτή μπορούμε, για παράδειγμα, να αποδώσουμε διαφορετικό βαθμό προτεραιότητας σε διαφορετικές υπηρεσίες του δικτύου. Αυτό θα μεταφράζονταν σε ρύθμιση των στοιχείων του δικτύου, έτσι ώστε να αποστέλλουν πρώτα τα πακέτα των υπηρεσιών με τη μεγαλύτερη προτεραιότητα.

- ✓ **Καταγραφή**

Μία βασική λειτουργία κατά τη διαχείριση ενός δικτύου είναι η πλήρης καταγραφή του. Η καταγραφή αναφέρεται στην τοπολογία του, στον υπάρχοντα υλικό εξοπλισμό, στα χρησιμοποιούμενα πρωτόκολλα, στις μισθωμένες ζεύξεις κ.τ.λ. Ένα βασικό σημείο αυτής της λειτουργίας είναι η συνεχής ενημέρωση των αρχείων όπου κρατάμε τα στοιχεία που αναφέρονται στο δίκτυο. Στη λειτουργία της καταγραφής θα αναφερθούμε ιδιαιτέρως σε άλλο κεφάλαιο.

- ➢ **Διαχείριση σφαλμάτων**

Η διαχείριση σφαλμάτων ασχολείται με τον εντοπισμό και την απομόνωση μη φυσιολογικών καταστάσεων που επηρεάζουν δυσμενώς τη λειτουργία του δικτύου. Οι λειτουργίες που περιλαμβάνει η διαχείριση σφαλμάτων αναλύονται στη συνέχεια.

- ✓ **Βεβαίωση αξιοπιστίας, διαθεσιμότητας και βιωσιμότητας – Reliability, availability and survivability (RAS) quality assurance**

Στη λειτουργία αυτή περιλαμβάνεται η απόφαση για το ελάχιστο όριο αξιοπιστίας και διαθεσιμότητας που έχει το δίκτυο. Τα όρια αυτά τίθενται ξεχωριστά για κάθε στοιχείο του δικτύου και συνολικά για όλο το δίκτυο.

✓ **Παρακολούθηση συναγερμών – Alarm surveillance**

Η λειτουργία αυτή παρέχει τη δυνατότητα καταγραφής ενός σφάλματος σε πραγματικό χρόνο και αναφοράς του στο διαχειριστή. Επίσης, τα σφάλματα μπορούν να καταγραφούν σε ένα αρχείο log για ανάλυση στο μέλλον. Μέρος της λειτουργίας αυτής είναι ο καθορισμός μίας γενικότερης πολιτικής για το ποια γεγονότα που συμβαίνουν το δίκτυο πρέπει να προκαλούν συναγερμούς. Επίσης, εδώ καθορίζεται και η πολιτική σύμφωνα με την οποία συσχετίζονται οι συναγερμοί.

✓ **Εντοπισμός σφαλμάτων – Fault localization**

Από τη στιγμή που αναφερθεί ένα σφάλμα, πρέπει να γίνει ακριβής εντοπισμός του πάνω στο δίκτυο. Αυτό είναι το αντικείμενο της συγκεκριμένης λειτουργίας.

✓ **Επιδιόρθωση σφαλμάτων – Fault correction**

Η λειτουργία αυτή περιλαμβάνει την επιδιόρθωση ενός σφάλματος. Η επιδιόρθωση μπορεί να γίνει αυτοματοποιημένα ή χειρονακτικά. Επίσης, μπορεί να πρέπει να έρθουμε σε συνεννόηση με τον πελάτη, πριν προχωρήσουμε στην επιδιόρθωση. Μία συνιστώσα της συγκεκριμένης λειτουργίας είναι και η χάραξη μίας γενικότερης πολιτικής για την αντιμετώπιση σφαλμάτων, για παράδειγμα η ύπαρξη ανταλλακτικών σε κάποιους κόμβους του δικτύου.

✓ **Έλεγχοι – Testing**

Η λειτουργία αυτή περιλαμβάνει ελέγχους ρουτίνας που πρέπει να γίνονται στο δίκτυο, για τον εντοπισμό σφαλμάτων. Επιπλέον, περιλαμβάνει και τους ελέγχους που γίνονται μετά την επιδιόρθωση ενός σφάλματος.

✓ **Διαχείριση προβλημάτων – Trouble administration**

Η λειτουργία αυτή σχετίζεται και με το επιχειρηματικό περιβάλλον. Πιο συγκεκριμένα περιλαμβάνει τις αναφορές που ανταλλάσσονται ανάμεσα στον πελάτη και τον οργανισμό, όταν υπάρχει πρόβλημα στο δίκτυο. Επίσης, προβλέπει την ύπαρξη μηχανισμού καταγραφής προβλημάτων που αναφέρει ο πελάτης, ή που εντοπίζονται, για μελλοντική επεξεργασία.

➢ **Διαχείριση απόδοσης**

Η διαχείριση απόδοσης πραγματεύεται τη συλλογή και παρακολούθηση των στατιστικών μεγεθών που δηλώνουν πόσο «καλά» λειτουργεί το δίκτυο. Η απόδοση μπορεί να αναφέρεται στη διαθεσιμότητα, σε καθυστερήσεις, σε σφάλματα μετάδοσης, στον αριθμό των πακέτων που πρέπει να επανεκπεμπούν κ.τ.λ. Οι λειτουργίες που περιέχει η διαχείριση απόδοσης περιγράφονται στη συνέχεια.

✓ **Βεβαίωση ποιότητας απόδοσης – Performance quality assurance**

Η λειτουργία αυτή πραγματεύεται τα κριτήρια που έχει θέσει ο πελάτης σε σχέση με την απόδοση. Για να επιτευχθούν οι απαιτήσεις του πελάτη πρέπει να καθοριστούν απαιτήσεις σε επίπεδο στοιχείων, δικτύου και υπηρεσιών. Οι απαιτήσεις αυτές συχνά εκφράζονται μέσα από μία συμφωνία εγγυημένου επιπέδου υπηρεσιών (SLA–Service Level Agreement). Για την περιγραφή του επιπέδου υπηρεσιών χρησιμοποιούνται κριτήρια ποιότητας, όπως η διαθεσιμότητα και οι καθυστερήσεις.

✓ **Καταγραφή απόδοσης – Performance monitoring**

Η λειτουργία αυτή ασχολείται με τη συνεχή καταγραφή των μεγεθών που δείχνουν την απόδοση του δικτύου. Η ύπαρξη μίας πολιτικής τόσο για την καταγραφή, όσο και για την αποθήκευση των δεδομένων αυτών είναι απαραίτητη.

✓ **Έλεγχος απόδοσης – Performance control**

Η λειτουργία αυτή ασχολείται με τον έλεγχο παραμέτρων που επηρεάζουν την απόδοση. Περιλαμβάνει την παραμετροποίηση κατωφλίων που σχετίζονται με την απόδοση και την τοποθέτηση μετρητών. Για παράδειγμα, ο έλεγχος της δρομολόγησης είναι μέρος του ελέγχου της κίνησης στο δίκτυο. Η κίνηση στις ζεύξεις του δικτύου επηρεάζει την καθυστέρηση των πακέτων, δηλαδή την απόδοση, και όταν περνά ένα κατώφλι τα αποτελέσματα είναι ιδιαίτερα δυσμενή.

✓ **Ανάλυση απόδοσης – Performance analysis**

Η λειτουργία αυτή ασχολείται με την ανάλυση των δεδομένων απόδοσης που έχουν καταγραφεί. Η ανάλυση των δεδομένων αυτών μπορεί να οδηγήσει σε προτάσεις για τη βελτίωση του δικτύου, όπως αναβαθμίσεις γραμμών και μηχανημάτων και πιθανές επεκτάσεις. Επίσης, μπορεί να χρησιμοποιηθεί για την πρόβλεψη της κίνησης στο δίκτυο κάτω από συγκεκριμένες συνθήκες, με βάση όσα είναι ήδη γνωστά.

➢ **Διαχείριση ασφάλειας**

Η διαχείριση ασφάλειας ασχολείται με την ασφάλεια στην επικοινωνία μεταξύ συστημάτων, μεταξύ πελάτη και συστήματος και μεταξύ υπαλλήλων του οργανισμού και συστήματος. Βασικές συνιστώσες της ασφάλειας είναι η αυθεντικοποίηση, η εμπιστευτικότητα, η ακεραιότητα δεδομένων και η μη αποποίηση. Στη συνέχεια αναφέρουμε τις λειτουργίες της διαχείρισης ασφάλειας.

✓ **Πρόληψη – Prevention**

Η λειτουργία αυτή αναφέρεται στην αποτροπή παραβίασης κάποιου από τους κανόνες ασφαλείας που προβλέπονται στην πολιτική ασφαλείας του οργανισμού. Συνήθως μιλάμε για αποτροπή μη εξουσιοδοτημένης προσπέλασης σε συγκεκριμένο χώρο ή δεδομένα. Μέρος της λειτουργίας αυτής είναι η φύλαξη των χώρων σε φυσικό επίπεδο και η ανάλυση επικινδυνότητας του προσωπικού του οργανισμού. Επιπλέον,

μέρος της πρόληψης κάποιας παραβίασης των κανόνων ασφαλείας είναι και η τοποθέτηση συστημάτων που εξασφαλίζουν το δίκτυο της επιχείρησης από εξωτερικούς κινδύνους. Το ποιο χαρακτηριστικό ίσως παράδειγμα είναι η τοποθέτηση αναχωμάτων ασφαλείας (firewalls), τα οποία προσπαθούν να ανακόψουν επιθέσεις που προέρχονται από τον «έξω κόσμο».

✓ **Ανίχνευση – Detection**

Η λειτουργία της ανίχνευσης μίας παραβίασης των κανόνων ασφαλείας μπορεί να πάρει δύο μορφές. Στην πρώτη περίπτωση δρα προληπτικά, αφού ανιχνεύεται η προσπάθεια κάποιου να παραβιάσει τους κανόνες ασφαλείας. Στη δεύτερη περίπτωση η παραβίαση ανιχνεύεται αφού έχει γίνει. Μέσα για την ανίχνευση μίας παραβίασης ασφαλείας μπορούν να γίνουν η ανάλυση κίνησης και η παρατήρηση ασυνήθιστης συμπεριφοράς των χρηστών που δε συμφωνεί με τα υπάρχοντα πρότυπα [5].

Στην πρώτη περίπτωση, για παράδειγμα, μπορεί να έχουμε ένα δίκτυο επιχείρησης στο οποίο μετά της δώδεκα το βράδυ και πριν τις έξι το πρωί δεν υπάρχει κίνηση, καθώς η επιχείρηση είναι κλειστή. Αν μία μέρα παρατηρηθεί κίνηση από τις τρεις μέχρι τις τέσσερις το πρωί, θα πρέπει να υποψιαστούμε ότι κάποιος χρησιμοποιεί το δίκτυο χωρίς να έχει εξουσιοδότηση.

Για να κατανοήσουμε την περίπτωση στην οποία έχουμε ασυνήθιστη συμπεριφορά χρηστών θα αναφέρουμε το εξής παράδειγμα: έστω ότι διαθέτουμε ένα σύστημα που καταγράφει πόσες φορές ένας χρήστης δίνει λανθασμένο κωδικό πριν κάνει login στο λογαριασμό του. Αν στη συντριπτική πλειοψηφία των περιπτώσεων κάποιος μπαίνει στο σύστημα κάνοντας το πολύ δύο φορές λάθος και καταγράψουμε μία περίπτωση στην οποία κάποιος μπήκε αφού έκανε λάθος πεντακόσιες φορές, θα πρέπει να σκεφτούμε ότι μπορεί να εκδηλώθηκε παραβίαση της ασφάλειας του συστήματος. Η συμπεριφορά αυτή ξεφεύγει από τα συνηθισμένα πρότυπα και μπορεί να δηλώνει ότι κάποιος προσπαθούσε να σπάσει ένα κωδικό και να εισέλθει στο σύστημα.

✓ **Περιορισμός και ανάκαμψη – Containment and recovery**

Η λειτουργία αυτή έχει να κάνει με τη λήψη προληπτικών μέτρων, τα οποία θα αποτρέψουν μεγάλης έκτασης ζημιές, αν συμβεί κάποια παραβίαση στους κανόνες ασφαλείας. Επίσης, προβλέπει τι πρέπει να γίνει, αφού εκδηλωθεί η παραβίαση των κανόνων ασφαλείας, ώστε οι ζημιές που θα προκληθούν να μειωθούν στο ελάχιστο και το σύστημα να μπορέσει να λειτουργήσει απρόσκοπτα [5].

Ως παράδειγμα αναφέρουμε τα προγράμματα προστασίας από ιούς. Έστω ότι λάβαμε με e-mail ένα αρχείο το οποίο είναι μολυσμένο με κάποιον ιό. Κάποια προγράμματα για προστασία από τους ιούς δε θα μας αφήσουν να ανοίξουμε το αρχείο, αποτρέποντας με τον τρόπο αυτό τη μόλυνση του συστήματός μας. Αν υποθέσουμε ότι ανοίξαμε το αρχείο και τελικά το σύστημά μας μολύνθηκε, τα προγράμματα αυτά βοηθούν στον περιορισμό των δυσμενών επιπτώσεων. Για να γίνει κατανοητή η περίπτωση αυτή, ας υποθέσουμε ότι ο ιός από τον οποίο μολυνθήκαμε προκαλεί τη συνεχή εκπομπή πακέτων από συγκεκριμένη θύρα του τερματικού μας, με αποτέλεσμα αν μολύνθηκαν πολλά τερματικά να φορτώνεται το δίκτυο με άχρηστα δεδομένα και να είναι υπαρκτός ο κίνδυνος κατάρρευσης. Υπάρχουν προγράμματα που μπορούν να μπλοκάρουν την πρόσβαση του ιού στη

συγκεκριμένη θύρα (μόλις τον εντοπίσουν) και έτσι να αποτραπεί η κατάρρευση του δικτύου.

✓ **Διαχείριση ασφάλειας – Security administration**

Η λειτουργία αυτή περιλαμβάνει τη χάραξη των γενικότερων πολιτικών ασφαλείας του οργανισμού που συνοψίζει ποιος έχει πρόσβαση πού και για ποιες εργασίες. Υλοποιείται με τον καθορισμό ομάδων χρηστών, ομάδων μηχανημάτων και δικαιωμάτων (δικαίωμα μόνο για ανάγνωση, δικαίωμα μετατροπής, πλήρη δικαιώματα). Επίσης, περιλαμβάνει την ανάλυση των δεδομένων που έχουν σχέση με την ασφάλεια, τη διαχείριση των passwords και των μηχανισμών ελέγχου προσπέλασης στα δεδομένα.

➢ **Διαχείριση κόστους**

Η διαχείριση κόστους ασχολείται με τη μέτρηση του κόστους για την παροχή μίας υπηρεσίας, με την πολιτική χρέωσης του πελάτη και με την εύρεση της φόρμουλας που θα αυξήσει στο μέγιστο δυνατό τα κέρδη του οργανισμού. Τα σύγχρονα λογιστικά πρότυπα αναλύουν τις μονάδες της επιχείρησης σε κέντρα εσόδων και κόστους. Το κόστος των τηλεπικοινωνιών πρέπει να επιμεριστεί ανά μονάδα. Η διαχείριση κόστους αφορά στη χρήση των τηλεπικοινωνιακών πόρων δηλαδή Η/Υ, τηλεφώνων, γραμμών δεδομένων κ.τ.λ. Επίσης, μπορεί να επεκταθεί στη χρήση περιφερειακών συσκευών, όπως οι εκτυπωτές και οι σαρωτές [4]. Οι λειτουργίες της διαχείρισης κόστους περιγράφονται στη συνέχεια.

✓ **Μέτρηση χρήσης – Usage measurement**

Η μέτρηση της χρήσης των πόρων του δικτύου ή μίας υπηρεσίας είναι το αντικείμενο αυτής της λειτουργίας. Η καταγραφή της χρήσης γίνεται πάντα σε σχέση με ένα συγκεκριμένο χρήστη. Αυτό είναι το πρώτο βήμα για να γίνει στη συνέχεια η χρέωση.

✓ **Κοστολόγηση/Τιμολόγηση – Tariffing/Pricing**

Η λειτουργία αυτή αναφέρεται στη χρέωση του χρήστη και περιλαμβάνει μία γενικότερη πολιτική κοστολόγησης. Η πολιτική αυτή αποφασίζεται με βάση κριτήρια σε επίπεδο επιχείρησης.

✓ **Είσπραξη και δημοσιονομία – Collections and finance**

Η λειτουργία αυτή αναφέρεται στη διαδικασία συλλογής των χρημάτων από τους πελάτες. Περιλαμβάνει την κοινοποίηση του λογαριασμού στον πελάτη, η οποία μπορεί να γίνει, για παράδειγμα, μέσω ταχυδρομείου ή ηλεκτρονικά. Επίσης, καθορίζει τον τρόπο με τον οποίο θα γίνει η εξόφληση του λογαριασμού.

✓ **Έλεγχος επιχειρηματικού περιβάλλοντος – Enterprise control**

Η λειτουργία αυτή αναφέρεται στο επιχειρηματικό περιβάλλον. Ασχολείται με την ορθή διαχείριση των οικονομικών του οργανισμού και καθορίζει τη ροή κεφαλαίων μεταξύ του οργανισμού, των ιδιοκτητών του και των πιστωτών του. Προφανής στόχος είναι η αύξηση των κερδών και η μείωση του κόστους.

1.2.4.2 Αντιστοιχίες του λειτουργικού μοντέλου στο SNMP

Όπως έχουμε ήδη αναφέρει, το λειτουργικό μοντέλο δεν ορίζεται για το μοντέλο διαχείρισης του internet. Παρόλα αυτά, κάποιες από της λειτουργίες που προβλέπονται στο μοντέλο OSI μπορούν να υλοποιηθούν με το SNMP. Στη συνέχεια αναφέρουμε τις αντιστοιχίες αυτές.

Διαχείριση διαμόρφωσης: Η διαχείριση διαμόρφωσης στο μοντέλο διαχείρισης internet γίνεται με το μήνυμα set-request του SNMP. Όπως είδαμε, με το συγκεκριμένο μήνυμα ο διαχειριστής μπορεί να αρχικοποιήσει ή να μεταβάλλει τις τιμές των διαχειριζόμενων αντικειμένων. Έτσι, το μήνυμα αυτό χρησιμοποιείται, για παράδειγμα, για να συμπληρωθούν οι πίνακες διευθύνσεων και δρομολόγησης. Υπενθυμίζεται ότι στο μήνυμα set-request μπορούν να υπάρξουν περισσότερα του ενός ζεύγη μεταβλητής-τιμής, οπότε με ένα set-request ο διαχειριστής μπορεί να δώσει εντολή για την αλλαγή περισσότερων της μίας μεταβλητής.

Διαχείριση σφαλμάτων: Η διαχείριση σφαλμάτων στο μοντέλο διαχείρισης internet γίνεται με το μήνυμα trap του SNMP, το οποίο αποστέλλει ο πράκτορας στο διαχειριστή.

Διαχείριση απόδοσης: Η διαχείριση απόδοσης στο μοντέλο διαχείρισης internet, γίνεται ουσιαστικά με το μήνυμα get-request του SNMP. Σκοπός του διαχειριστή στη συγκεκριμένη περίπτωση είναι να αντλήσει πληροφορίες, σχετικές με την απόδοση, οι οποίες βρίσκονται στη MIB του πράκτορα. Για παράδειγμα, υπάρχει αντικείμενο που δηλώνει τον αριθμό τον πακέτων που εισήλθαν σε ένα interface. Ο διαχειριστής διαθέτει την επεξεργαστική ισχύ που χρειάζεται για να κάνει την ανάλυση των δεδομένων που λαμβάνει. Ανάλογος είναι και ο τρόπος που γίνεται η διαχείριση απόδοσης στο OSI. Βέβαια, πρέπει να επισημανθεί ότι εκεί τα διαχειριζόμενα αντικείμενα στέλνουν notifications που μπορεί να περιέχουν και αποτελέσματα ελέγχων που έκαναν τα ίδια, οπότε ο διαχειριστής δεν είναι υποχρεωμένος να στέλνει πάντα μηνύματα get για κάθε πληροφορία που θέλει να πάρει.

Διαχείριση ασφάλειας: Το SNMP προβλέπει μηχανισμούς ασφαλείας μόνο για την πληροφορία διαχείρισης και όχι για το σύνολο της πληροφορίας που διακινείται μέσω του δικτύου. Στους μηχανισμούς ασφαλείας του SNMP θα αναφερθούμε στη συνέχεια.

Διαχείριση κόστους: Όπως έχουμε ήδη αναφέρει το πρώτο βήμα για τη διαχείριση κόστους είναι να δούμε ποιος χρησιμοποιεί τι. Για να βρούμε πόση κίνηση προκαλεί στο δίκτυο μία εφαρμογή, μπορούμε να χρησιμοποιήσουμε τη λειτουργία get του

SNMP και να μάθουμε πόσα είναι τα πακέτα σε μία ζεύξη που αντιστοιχούν στο συγκεκριμένο τύπο κίνησης που προκαλεί η εφαρμογή αυτή. Στη συνέχεια, διαιρώντας τον αριθμό των πακέτων πολλαπλασιασμένο με το μέσο μέγεθος πακέτου με τη συνολική κίνηση στη ζεύξη, μαθαίνουμε το ποσοστό του φόρτου που προκαλείται στη συγκεκριμένη ζεύξη από την εφαρμογή. Φυσικά, τα πράγματα περιπλέκονται όταν πολλές εφαρμογές προκαλούν κίνηση ίδιου τύπου.

Η ασφάλεια στο SNMP [5], [1], [20]

Όπως αναφέραμε το SNMP προσφέρει κάποιους μηχανισμούς ασφαλείας, οι οποίοι όμως αφορούν μόνο στην πληροφορία διαχείρισης. Στην πρώτη έκδοση του πρωτοκόλλου (**SNMPv1**), η βασική έννοια που σχετίζεται με την ασφάλεια είναι η ***κοινότητα***. Επιπλέον, οι έννοιες του δικαιώματος προσπέλασης και της άποψης της MIB έρχονται να συμπληρώσουν την ασφάλεια που προσφέρεται από την κοινότητα. Στη συνέχεια αναλύουμε τις τρεις αυτές έννοιες.

- **Κοινότητα**

Ένας διαχειριστής μπορεί να επικοινωνήσει με ένα πράκτορα μόνο αν ανήκουν στην ίδια κοινότητα. Η κοινότητα χαρακτηρίζεται από ένα όνομα, το community string, και μέσω αυτού ουσιαστικά πραγματοποιείται η αυθεντικοποίηση των μελών της. Πολλοί πράκτορες μπορεί να ανήκουν στην ίδια κοινότητα, οπότε ένας διαχειριστής που ανήκει επίσης στην ίδια κοινότητα, μπορεί να επικοινωνεί με όλους. Επίσης, ένας πράκτορας μπορεί να ανήκει σε περισσότερες από μία κοινότητες, άρα μπορεί να προσπελαστεί από περισσότερους του ενός διαχειριστές. Με βάση τα παραπάνω είναι φανερό, ότι η επικοινωνία μεταξύ πρακτόρων και διαχειριστών μπορεί να πάρει τη μορφή ένας με πολλούς, πολλοί με έναν και πολλοί με πολλούς.

- **Άποψη της MIB**

Όπως έχουμε ήδη αναφέρει, ένας πράκτορας δεν έχει τη συνολική εικόνα των διαχειριζόμενων αντικειμένων του δικτύου. Αυτό που «βλέπει» είναι ένα υποσύνολο της MIB, ή όπως λέμε μία άποψη της MIB.

- **Δικαίωμα προσπέλασης**

Τα μέλη μίας κοινότητας δεν είναι απαραίτητο να έχουν πλήρη δικαιώματα προσπέλασης στο διαχειριζόμενα αντικείμενα που μπορούν να «δουν». Για παράδειγμα, μπορεί να ορίζεται ότι τα δικαιώματά τους είναι μόνο για ανάγνωση ή για εγγραφή και ανάγνωση. Επιπλέον, ο ορισμός κάθε διαχειριζόμενου αντικειμένου αναφέρει αν το αντικείμενο αυτό μπορεί να προσπελαστεί μόνο ανάγνωση, για ανάγνωση και εγγραφή ή αν δεν μπορεί να προσπελαστεί καθόλου.

Είναι προφανές ότι οι μηχανισμοί ασφαλείας που υποστηρίζει το SNMPv1 είναι ελλιπείς. Για το λόγο αυτό, το ***SNMPv3*** σχεδιάστηκε έτσι ώστε να υποστηρίζει επιπρόσθετους μηχανισμούς ασφαλείας. Οι μηχανισμοί αυτοί αναλύονται στη συνέχεια.

- **Εξουσιοδοτημένες και μη εξουσιοδοτημένες μηχανές**

Σε κάθε μετάδοση μηνύματος SNMP, μία από τις δύο οντότητες, είτε ο αποστολέας είτε ο δέκτης, ανακηρύσσεται σε εξουσιοδοτημένη μηχανή SNMP. Η άλλη οντότητα είναι μία μη εξουσιοδοτημένη μηχανή. Ως εξουσιοδοτημένη μηχανή ορίζεται η οντότητα που αποστέλλει τα μηνύματα trap και response, ή αυτή που παραλαμβάνει τα μηνύματα get-request, set-request, get-next-request, get-bulk-request και inform. Είναι φανερό, πως στη επικοινωνία ενός διαχειριστή με έναν πράκτορα, ο πράκτορας είναι η εξουσιοδοτημένη μηχανή.

Ο υπολειπόμενος χρόνος του μηνύματος προσδιορίζεται από ένα ρολόι που υπάρχει στην εξουσιοδοτημένη μηχανή SNMP. Όταν μία εξουσιοδοτημένη μηχανή στέλνει ένα μήνυμα, περιλαμβάνει σε αυτό την τρέχουσα τιμή του ρολογιού της, ώστε η μη εξουσιοδοτημένη μηχανή να συγχρονιστεί χαλαρά με αυτή. Επιπλέον, όταν μία μη εξουσιοδοτημένη μηχανή στέλνει ένα μήνυμα , αυτό περιλαμβάνει την εκτίμηση της τιμής του χρόνου του αποδέκτη. Με τη διατήρηση αυτού του χαλαρού συγχρονισμού, παρέχεται προστασία από επιθέσεις τύπου ***επανεκπομπής μηνυμάτων***.

- **Κλειδιά αυθεντικοποίησης και εμπιστευτικότητας**

Στο SNMPv3 υπάρχει η δυνατότητα υποστήριξης μηχανισμών ***εμπιστευτικότητας*** και ***αυθεντικοποίησης***. Για να γίνει αυτό, πρέπει μία εξουσιοδοτημένη και μία μη εξουσιοδοτημένη μηχανή να γνωρίζουν τα ίδια μυστικά κλειδιά αυθεντικοποίησης και εμπιστευτικότητας/κρυπτογράφησης.

Κάθε οντότητα πρέπει να διατηρεί ένα κλειδί αυθεντικοποίησης και ένα κλειδί κρυπτογράφησης. Αυτά τα κλειδιά δεν αποθηκεύονται στη MIB και δεν είναι προσπελάσιμα μέσω του SNMP. Στην πραγματικότητα, αυτό που γίνεται είναι ότι ένας χρήστης εισάγει το συνθηματικό του και από αυτό παράγονται και τα δύο κλειδιά. Έτσι, δε χρειάζεται να αποθηκεύονται οι τιμές των κλειδιών του χρήστη.

Ένα τοπικό κλειδί καθορίζεται ως ένα μυστικό κλειδί που γνωρίζουν ένας χρήστης και μία εξουσιοδοτημένη μηχανή SNMP. Ένα μοναδικό μυστικό κλειδί ενός χρήστη αντιστοιχίζεται σε διαφορετικά τοπικά κλειδιά, μέσω μίας μονόδρομης συνάρτησης σύνοψης. Το τοπικό κλειδί που προκύπτει αντιστοιχίζεται σε ένα πράκτορα με ασφαλή τρόπο. Επειδή ακριβώς χρησιμοποιείται μονόδρομη συνάρτηση σύνοψης (SHA-1 ή MD-5), είναι αδύνατο για κάποιον που πραγματοποιεί επίθεση να μάθει το κλειδί του χρήστη, ακόμα και αν γνωρίζει το τοπικό κλειδί. Με βάση τα παραπάνω είναι προφανές ότι ισχύουν τα εξής:

i) Κάθε χρήστης έχει ένα μοναδικό κλειδί, έτσι ακόμα και αν το κλειδί αυτό εκτεθεί σε κίνδυνο, τα κλειδιά των υπόλοιπων χρηστών δεν αποκαλύπτονται.

ii) Τα κλειδιά που έχουν διαφορετικοί πράκτορες για ένα συγκεκριμένο χρήστη είναι διαφορετικά. Άρα, αν ένας πράκτορας εκτεθεί σε κίνδυνο, μπορεί να γίνουν γνωστά μόνο τα κλειδιά που αυτός γνωρίζει, και όχι τα κλειδιά των χρηστών που χρησιμοποιούν οι άλλοι πράκτορες.

1.3 Η αρχιτεκτονική στο OSI/TMN [2], [3]

Για ευκολία στο σχεδιασμό και στην ανάπτυξη ενός δικτύου διαχείρισης, η ITU-T στις συστάσεις της περιγράφει τρεις αρχιτεκτονικές: τη λειτουργική αρχιτεκτονική, τη φυσική αρχιτεκτονική και την αρχιτεκτονική πληροφορίας. Καθεμία από αυτές τις αρχιτεκτονικές αποτελεί μία όψη του δικτύου. Στη συνέχεια θα αναλύσουμε τις τρεις αρχιτεκτονικές του TMN.

➢ **Λειτουργική αρχιτεκτονική**

Η λειτουργική αρχιτεκτονική ορίζει τις λειτουργίες διαχείρισης και τις διεπαφές τους, οι οποίες καλούνται σημεία αναφοράς (reference points). Συνδυάζοντας τις λειτουργίες διαχείρισης, μπορούμε να παράγουμε την επιθυμητή λειτουργικότητα για το δίκτυο σαν σύνολο. Οι διαφορετικές λειτουργίες διαχείρισης που ορίζονται περιγράφονται στη συνέχεια.

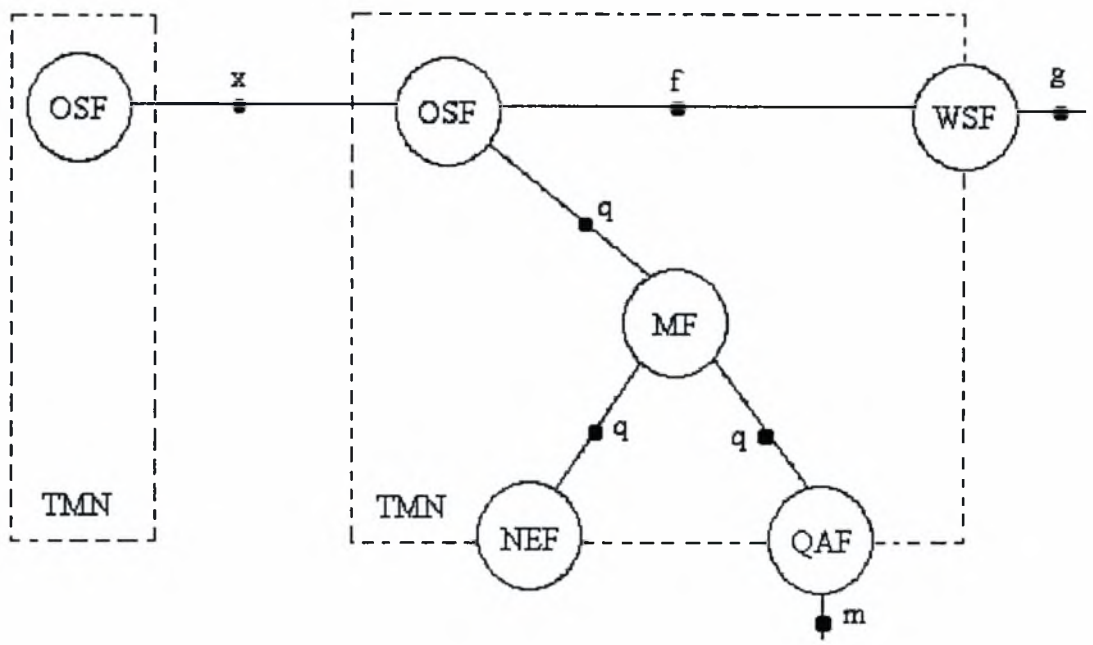
- **Network Element Function (NEF):** είναι υπεύθυνη για τη διαχείριση των στοιχείων του δικτύου.

- **Operations Systems Function (OSF):** έχει τη γενική ευθύνη για τη διαχείριση του δικτύου διαχείρισης. Υπάρχουν τέσσερα είδη OSF: element OSF, network OSF, service OSF και business OSF, για τη διαχείριση σε επίπεδο στοιχείων δικτύου, σε επίπεδο δικτύου, σε επίπεδο υπηρεσιών και σε επιχειρηματικό επίπεδο. Το σημείο αναφοράς που χωρίζει ένα OSF από ένα άλλο OSF ίδιου επιπέδου, ή από ένα OSF σε άλλο δίκτυο διαχείρισης, είναι το x. Αντίθετα, OSF διαφορετικών επιπέδων στο ίδιο δίκτυο χωρίζονται από το σημείο αναφοράς q_3. Η διεπαφή που αντιστοιχεί στο σημείο αναφοράς q_3 είναι η Q_3, η οποία υποστηρίζει και τα επτά στρώμα τα του OSI, σε αντίθεση με την Q_x.

- **Workstation Function (WSF):** δίνει τη δυνατότητα στους χρήστες να δουν την πληροφορία διαχείρισης. Το σημείο αναφοράς μεταξύ χρήστη και WSF είναι το g. Σημειώνεται ότι το σημείο αναφοράς g δεν ανήκει στο TMN, όπως και το σημείο αναφοράς m. Η WSF συνδέεται με τη λειτουργία OSF με το σημείο αναφοράς f.

- **Q-Adaptor Function (QAF):** χρησιμοποιείται για τη μετατροπή πληροφορίας διαχείρισης, όταν το ένα σημείο αναφοράς ανήκει στο TMN και το άλλο δεν ανήκει. Για το λόγο αυτό στο σχήμα 1.25 μέρος της είναι εκτός του TMN. Το σημείο αναφοράς m δεν ανήκει στο TMN.

- **Mediation Function (MF):** είναι ένα είδος πύλης για τη μεταφορά πληροφορίας διαχείρισης, όταν οι λειτουργίες διαχείρισης έχουν διαφορετικά σημεία αναφοράς.



Σχήμα 1.25: *Λειτουργίες διαχείρισης και σημεία αναφοράς*

- **Φυσική αρχιτεκτονική**

Η φυσική αρχιτεκτονική προκύπτει από τη λειτουργική αρχιτεκτονική και ορίζει τη μεταφορά της σε φυσικά συστήματα . Κάθε σύστημα περιλαμβάνει μία λειτουργία ή τμήματά της. Για την επικοινωνία των συστημάτων το κατάλληλο σημείο αναφοράς υλοποιείται σε φυσικό επίπεδο από μία διεπαφή που συμβολίζεται με το ίδιο όνομα αλλά με κεφαλαία γράμματα. Τα συστήματα που ορίζονται από το TMN περιγράφονται στη συνέχεια.

- **Network Element (NE):** είναι ένα κομμάτι εξοπλισμού του δικτύου που εκτελεί λειτουργίες NEF. Τα στοιχεία δικτύου έχουν κάποια νοημοσύνη. Στην πραγματικότητα ένα NE μπορεί να παρομοιαστεί με ένα πράκτορα. Υποστηρίζει απαραίτητα τη διεπαφή Q και προαιρετικά τις διεπαφές F και X.

- **Operations Systems (OS):** υποστηρίζει την επεξεργασία πληροφορίας που σχετίζεται με τις λειτουργίες και τη διαχείριση ενός δικτύου. Τα OS είναι το ανάλογο των διαχειριστών.

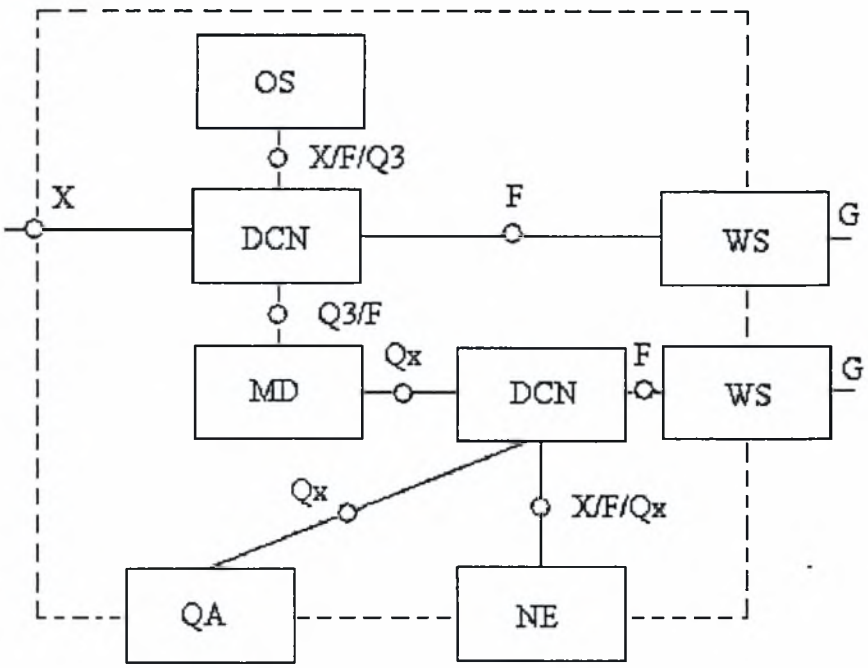
- **Workstation (WS):** είναι το σημείο από το οποίο ένας χρήστης μπορεί να έχει πρόσβαση στην πληροφορία διαχείρισης. Οι σταθμοί εργασίας μπορούν να είναι mainframes ή απλά τερματικά.

- **Q-Adaptor (QA):** μετατρέπει δεδομένα που δεν είναι σε συμβατή μορφή με το TMN σε δεδομένα που είναι και αντίστροφα.

- **Mediation Devices (MD):** δρουν σαν πύλες. Μία MD μπορεί να έχει πολλές λειτουργίες όπως μετάφραση διευθύνσεων, δρομολόγηση, μετατροπές

πρωτοκόλλων κ.τ.λ. Επίσης, μπορεί να χρησιμοποιηθεί για την αποθήκευση και το φιλτράρισμα δεδομένων.

- **Data Communication Network (DCN):** έχει δυνατότητες δρομολόγησης και μεταφοράς, που χρησιμοποιούνται για την ανταλλαγή πληροφορία διαχείρισης ανάμεσα σε OS και OS, OS και NE, WS και OS και WS και NE. Το DCN υποστηρίζει λειτουργίες μόνο για τα τρία πρώτα επίπεδα του OSI.



Σχήμα 1.26: Φυσική αρχιτεκτονική

➢ **Αρχιτεκτονική πληροφορίας**

Η αρχιτεκτονική πληροφορίας ορίζει τις αρχές για την ανταλλαγή πληροφοριών μεταξύ των τμημάτων λειτουργιών. Ορίζονται επίσης η δομή και η σημασία των πληροφοριών που ανταλλάσσονται. Ο ορισμός των διαχειριζόμενων αντικειμένων γίνεται με βάση την ASN.1 και τη GDMO. Η αρχιτεκτονική πληροφορίας χρησιμοποιεί την αντικειμενοστραφή προσέγγιση, στην οποία αναφερθήκαμε όταν αναλύσαμε το πληροφοριακό μοντέλο του μοντέλου διαχείρισης OSI.

1.4 Η αρχιτεκτονική του SNMPv3 [19], [1]

Το 1988 παρουσιάστηκε το SNMPv1. Το συγκεκριμένο πρωτόκολλο χρησιμοποιήθηκε ευρύτατα και λίγα χρόνια αργότερα (1993) παρουσιάστηκε η δεύτερη έκδοσή του, το SNMPv2, η οποία το τροποποιήθηκε 1996.

Η δεύτερη έκδοση του SNMP όμως είχε δύο πολύ σοβαρά μειονεκτήματα. Το πρώτο ήταν ότι δεν ήταν συμβατή με την πρώτη έκδοση του πρωτοκόλλου. Το πρόβλημα αυτό μπορούσε να επιλυθεί με δύο τρόπους. Ο πρώτος ήταν η δημιουργία

διαχειριστών που υποστήριζαν και τις δύο εκδόσεις του SNMP. Ο δεύτερος ήταν η χρησιμοποίηση ενός proxy server. Στο θέμα αυτό έχουμε αναφερθεί όταν αναλύσαμε το οργανωτικό μοντέλο.

Το δεύτερο πρόβλημα που είχε το SNMP2 ήταν ότι δεν κατάφερε να προσφέρει επαρκείς μηχανισμούς ασφάλειας, όπως άλλωστε και το SNMPv1. Στους μηχανισμούς αυτούς έχουμε αναφερθεί σε προηγούμενη ενότητα.

Για να ξεπεραστούν τα προβλήματα που είχε η δεύτερη έκδοση του SNMP το 1988 παρουσιάστηκε το SNMPv3. Η τρίτη έκδοση του SNMP είναι συμβατή με τις δύο προηγούμενες και παρέχει επιπλέον μηχανισμούς ασφάλειας, στους οποίους έχουμε ήδη αναφερθεί. Συμπερασματικά, όπως αναφέρεται και στο RFC 2570, το SNMPv3 μπορεί απλώς να θεωρηθεί ως SNMPv1 ή SNMPv2 με επιπρόσθετες δυνατότητες ασφάλειας και διαχείρισης.

Στη συνέχεια θα παρουσιάσουμε την αρχιτεκτονική του SNMP, όπως αυτή καθορίζεται στο RFC 2271. Σύμφωνα με την αρχιτεκτονική αυτή, υπάρχουν πολλές κατανεμημένες και αλληλεπιδρώσες οντότητες SNMP, καθεμία από τις οποίες περιέχει μία και μόνο μηχανή SNMP. Η μηχανή αυτή υλοποιεί διάφορες λειτουργίες, όπως αποστολή και λήψη μηνυμάτων, αυθεντικοποίηση, κρυπτογράφηση και αποκρυπτογράφηση μηνυμάτων, καθώς και έλεγχο προσπέλασης στα διαχειριζόμενα αντικείμενα. Όλες αυτές οι λειτουργίες παρέχονται ως υπηρεσίες από τις εφαρμογές. Οι εφαρμογές μαζί με τη μηχανή SNMP αποτελούν την οντότητα SNMP. Στο σχήμα 1.27 φαίνονται τα μέρη από τα οποία αποτελείται μία οντότητα SNMP. Ακολουθεί μία σύντομη ανάλυση της μηχανής SNMP και των εφαρμογών μίας οντότητας SNMP.

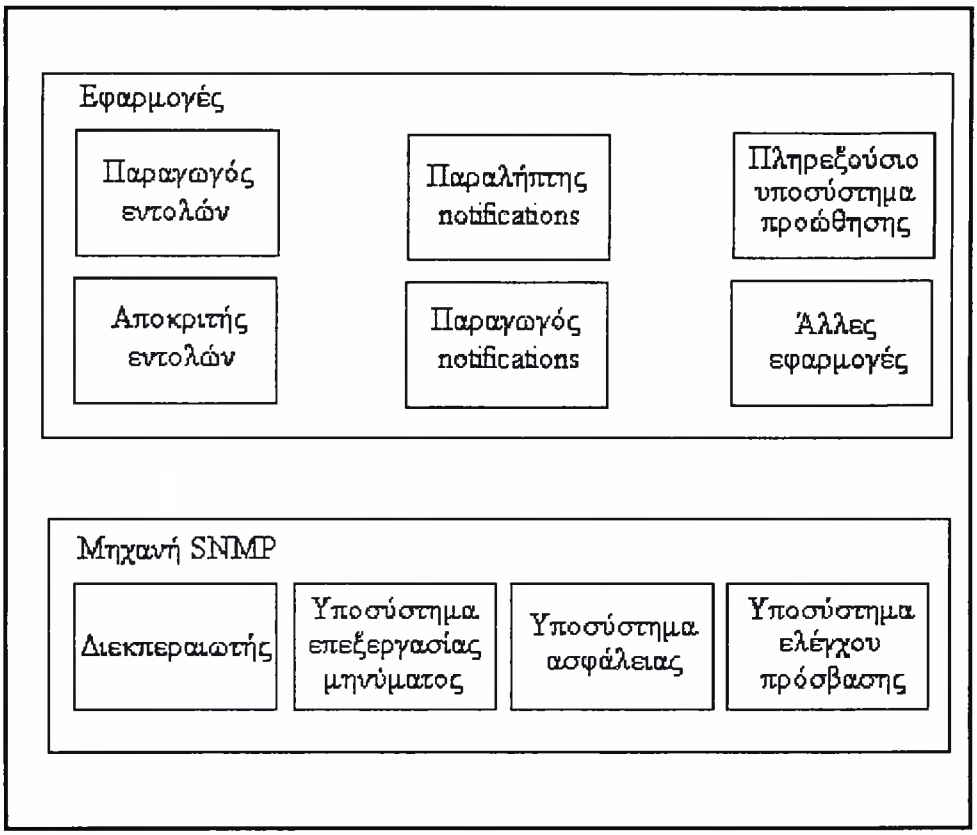
- **Τα μέρη της μηχανής SNMP**

Διεκπεραιωτής (dispatcher): Υπάρχει μόνο ένας διεκπεραιωτής σε μία μηχανή SNMP, αλλά μπορεί να χειρίζεται τα μηνύματα όλων των εκδόσεων του SNMP. Εκτελεί τρεις λειτουργίες. Η πρώτη είναι ότι στέλνει μηνύματα προς και λαμβάνει μηνύματα από το σύστημα. Η δεύτερη είναι ότι καθορίζει την έκδοση του SNMP στην οποία ανήκουν τα μηνύματα, ώστε να μπορεί να τα χειριστεί. Η τρίτη είναι ότι παρέχει μία διεπαφή για να παραδοθούν μηνύματα σε μία τοπική εφαρμογή, ή αντίστοιχα για να αποσταλούν μηνύματα από μία τοπική εφαρμογή σε μία απομακρυσμένη οντότητα.

Υποσύστημα επεξεργασίας μηνύματος (message processing subsystem): Το υποσύστημα επεξεργασίας μηνυμάτων αλληλεπιδρά με τον διεκπεραιωτή για να ενημερωθεί για την έκδοση του SNMP στην οποία ανήκει το μήνυμα. Στη συνέχεια επεξεργάζεται το μήνυμα χρησιμοποιώντας το κατάλληλο μοντέλο επεξεργασίας.

Υποσύστημα ασφαλείας (security subsystem): Το υποσύστημα ασφαλείας παρέχει υπηρεσίες αυθεντικοποίησης και ιδιωτικότητας.

Υποσύστημα ελέγχου πρόσβασης (access control): Το υποσύστημα ελέγχου πρόσβασης παρέχει ασφάλεια, διασφαλίζοντας μόνο εξουσιοδοτημένη πρόσβαση στα δεδομένα.



Σχήμα 1.27: Μία οντότητα SNMP

- **Οι εφαρμογές σε μία οντότητα SNMP**

Παραγωγός εντολών (command generator): Τυπικά η συγκεκριμένη εφαρμογή συνδέεται με τη διεργασία του διαχειριστή. Ο λόγος είναι ότι χρησιμοποιείται για την παραγωγή των μηνυμάτων get-request, get-next-request, get-bulk και set-request. Επίσης, επεξεργάζεται την απάντηση στην εντολή που έστειλε.

Αποκριτής εντολών (command responder): Ο αποκριτής εντολών επεξεργάζεται τα μηνύματα get-request και set-request που έχουν αποσταλεί από μία απομακρυσμένη οντότητα. Επιπλέον, εφαρμόζει ότι του υπαγορεύουν οι παραπάνω εντολές στο κατάλληλο διαχειριζόμενο αντικείμενο, ετοιμάζει το μήνυμα απόκρισης για την εντολή get και το στέλνει.

Παραγωγός notification (notification originator): Η συγκεκριμένη εφαρμογή παράγει μηνύματα τύπου trap ή inform. Η λειτουργίας της μοιάζει με αυτή του αποκριτή εντολών, με τη διαφορά ότι πρέπει μόνη της να βρει πού πρέπει να σταλεί το μήνυμα, καθώς επίσης και ποια έκδοση του SNMP και ποιες παράμετροι ασφαλείας πρέπει να χρησιμοποιηθούν.

Παραλήπτης notifications (notification receiver): Η εφαρμογή παραλήπτης notifications χρησιμοποιείται για να παραλαμβάνει μηνύματα trap και inform.

Πληρεξούσιο υποσύστημα προώθησης (proxy forwarded subsystem): Η εφαρμογή αυτή χρησιμοποιείται για την προώθηση των μηνυμάτων του SNMP, χωρίς να κάνει κανένα απολύτως έλεγχο για το αντικείμενο στο οποίο αναφέρονται τα μηνύματα αυτά.

1.5 Σύγκριση ανάμεσα στα μοντέλα διαχείρισης OSI/TMN και Internet συναρτήσει των πρωτοκόλλων που χρησιμοποιούν

Έχουμε πει ότι τα δύο κυρίαρχα μοντέλα διαχείρισης δικτύων είναι το OSI/TMN και το internet, που χρησιμοποιούν τα πρωτόκολλα CMIP και SNMP αντίστοιχα. Από όσα έχουμε πει στις προηγούμενες ενότητες πρέπει να έχουν γίνει εμφανείς αρκετές διαφορές ανάμεσα στα δύο «στρατόπεδα». Στην παρούσα ενότητα θα αναφέρουμε τις πιο σημαντικές από αυτές.

Η πρώτη σημαντική διαφορά έχει να κάνει με την ίδια τη φύση των χρησιμοποιούμενων πρωτοκόλλων. Στην επικοινωνία με CMIP δημιουργείται μία σύνδεση μεταξύ των δύο πλευρών που επικοινωνούν, ενώ το SNMP λειτουργεί χωρίς σύνδεση. Αυτό έχει ως αποτέλεσμα το SNMP να αντιμετωπίζει όλα τα προβλήματα που αντιμετωπίζουν τα πρωτόκολλα που δε χρησιμοποιούν σύνδεση. Τα πιο σημαντικά έχουν σχέση με θέματα ασφάλειας και με χάσιμο πακέτων, κάτι που επηρεάζει την απόδοση.

Επίσης, μία ακόμα σημαντική διαφορά των δύο πρωτοκόλλων είναι ότι το CMIP ακολουθεί αντικειμενοστραφή προσέγγιση, κάτι που δε συμβαίνει με το SNMP. Αυτό έχει ως αποτέλεσμα στο CMIP να εμφανίζονται στα διαχειριζόμενα αντικείμενα χαρακτηριστικά όπως η ενθυλάκωση και η κληρονομικότητα. Όπως είναι φυσικό τα αντικειμενοστραφή χαρακτηριστικά, στα οποία έχουμε αναφερθεί σε άλλη ενότητα, δεν παρουσιάζονται στα διαχειριζόμενα αντικείμενα του SNMP.

Τα διαχειριζόμενα αντικείμενα στο CMIP έχουν κάποιας μορφής νοημοσύνη, η οποία βέβαια σε καμία περίπτωση δεν πρέπει να συγκρίνεται με αυτή του διαχειριστή, με την έννοια ότι μπορούν να εκτελούν και τα ίδια κάποιες λειτουργίες. Για παράδειγμα, με το μήνυμα action μπορεί να ζητηθεί από ένα διαχειριζόμενο αντικείμενο να εκτελέσει ελέγχους. Τέτοιες δυνατότητες δεν υπάρχουν στο SNMP.

Οι πράκτορες στο CMIP στέλνουν event reports ενώ στο SNMP traps. Τα event reports μπορούν να περιέχουν πληροφορία που σχετίζεται με ελέγχους που έχουν κάνει τα ίδια τα διαχειριζόμενα αντικείμενα, κάτι που δε συμβαίνει με τα traps, οπότε ο διαχειριστής δεν είναι υποχρεωμένος να αποστέλλει συνεχώς μηνύματα get. Αντίθετα, στο SNMP ο διαχειριστής είναι υποχρεωμένος να ενημερώνεται στέλνοντας συνεχώς μηνύματα get-request στους πράκτορες. Κάθε τέτοιο μήνυμα προκαλεί την αποστολή ενός μηνύματος get-response από τον πράκτορα. Τα παραπάνω, έχουν ως αποτέλεσμα ο φόρτος που προκαλείται στο δίκτυο από τη χρήση του SNMP να είναι μεγαλύτερος από αυτόν που προκαλείται από το CMIP.

Επιπλέον, στο CMIP έχουμε καλύτερες δυνατότητες διαχείρισης δεδομένων μεγάλου όγκου. Βέβαια στο SNMPv2 υπάρχει το μήνυμα get-bulk-request που δεν υπήρχε στο SNMPv1, ακόμα και έτσι όμως το CMIP υπερέχει στον τομέα αυτό. Ο λόγος είναι η ύπαρξη των παραμέτρων scope, filtering και synchronization, τη λειτουργία των οποίων αναλύσαμε όταν αναφερθήκαμε στο πληροφοριακό μοντέλο του OSI.

Μία ακόμα διαφορά ανάμεσα στα δύο πρωτόκολλα, είναι η δυνατότητα που δίνει το CMIP για τη δημιουργία και τη διαγραφή στιγμιοτύπων διαχειριζόμενων

αντικειμένων, μέσω των μηνυμάτων create και delete. Το SNMP δεν υποστηρίζει κάτι τέτοιο, όπως επίσης δεν υποστηρίζει και κάποιο μήνυμα ανάλογο του action.

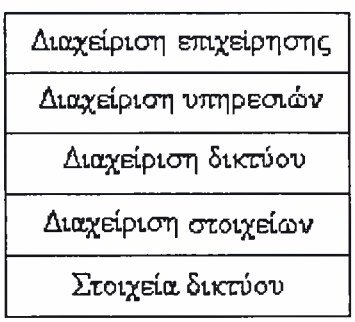
Οι πρώτες δύο εκδόσεις του SNMP είχαν πολύ σημαντικά προβλήματα στον τομέα της ασφάλειας. Το SNMPv3 κατάφερε να δώσει λύση σε αρκετά από αυτά. Βέβαια, πρέπει να γίνει ξεκάθαρο ότι ακόμα και στο SNMPv3 η ασφάλεια περιορίζεται στα δεδομένα διαχείρισης. Αντίθετα, όπως είδαμε, το μοντέλο διαχείρισης OSI καλύπτει συνολικά το θέμα της ασφάλειας ενός δικτύου και δεν ασχολείται μόνο με την πληροφορία διαχείρισης.

Η πληρότητα με την οποία αντιμετωπίζεται το θέμα της ασφάλειας είναι κάτι που χαρακτηρίζει το μοντέλο διαχείρισης OSI και σε όλες τις εκφάνσεις της διαχείρισης δικτύων. Άλλωστε, το συγκεκριμένο μοντέλο διαχείρισης ασχολείται και με τα επτά επίπεδα του OSI. Αντίθετα στο μοντέλο διαχείρισης internet δεν ορίζονται και τα επτά επίπεδα και φυσικά οι αντίστοιχες λειτουργίες διαχείρισης.

Ένα άλλο σημείο στο οποίο υπερτερεί σαφώς το μοντέλο διαχείρισης OSI/TMN είναι ότι προτείνει ένα πλαίσιο (framework) το οποίο περνά από τη διαχείριση των στοιχείων του δικτύου, στη διαχείριση του δικτύου σα σύνολο, στη διαχείριση των υπηρεσιών και από εκεί στη διαχείριση του ίδιου του οργανισμού. Στο μοντέλο διαχείρισης internet δεν υπάρχει καμία πρόβλεψη για τη διαχείριση σε επίπεδο επιχείρησης. Τα επίπεδα διαχείρισης στο OSI/TMN φαίνονται στο σχήμα 1.28.

Επίσης, όπως είδαμε στο μοντέλο διαχείρισης OSI/TMN υπάρχει και η λειτουργία της διαχείρισης κόστους, η οποία δεν υπάρχει στο μοντέλο internet. Η λειτουργία αυτή είναι πάρα πολύ σημαντική για τους τηλεπικοινωνιακούς οργανισμούς καθώς αποτελεί τη βάση για τη χρέωση των πελατών τους. Επίσης, στα σύγχρονα επιχειρηματικά περιβάλλοντα η διαχείριση κόστους είναι απαραίτητη και σε εταιρίες που διαθέτουν ιδιόκτητο δίκτυο, ώστε να επιτευχθεί η καταμέριση του κόστους από τη χρήση του δικτύου ενός οργανισμού στην ίδια την εταιρία και στις θυγατρικές της.

Από όλα τα παραπάνω φαίνεται ότι το μοντέλο διαχείρισης OSI υπερτερεί σημαντικά έναντι του μοντέλου διαχείρισης internet, οπότε το λογικό θα ήταν να περιμένει κανείς ότι το CMIP χρησιμοποιείται κατά κόρον και το SNMP πολύ λιγότερο. Τα πράγματα όμως δεν έχουν ακριβώς έτσι. Οι απαιτήσεις σε μνήμη του CMIP ήταν πολύ μεγάλες, σχεδόν απαγορευτικές, πριν από κάποια χρόνια. Αυτό έδωσε ένα προβάδισμα στο SNMP που άρχισε να χρησιμοποιείται ευρέως αμέσως μόλις παρουσιάστηκε, καθώς οι απαιτήσεις του σε μνήμη είναι πολύ μικρότερες. Βέβαια, αρχικά η επιστημονική κοινότητα πίστευε ότι το SNMP δεν είναι παρά ένα μεταβατικό στάδιο προς την πορεία της διαχείρισης δικτύων με CMIP. Αυτό φαίνεται και από τον κόμβο CMOT που είχε δεσμευτεί στις MIB του SNMP για τη μετάβαση αυτή, η οποία όμως τελικά δεν έγινε.



Σχήμα 1.28: *Επίπεδα διαχείρισης στο OSI/TMN*

Τα τελευταία χρόνια όμως, η τεχνολογία οδήγησε στην κατασκευή μνημών με πολλή μεγαλύτερη χωρητικότητα, οπότε το CMIP μπορεί να χρησιμοποιηθεί πλέον χωρίς πρόβλημα. Αυτός είναι και ο λόγος που το ενδιαφέρον για τη διαχείριση δικτύων με τη χρήση του CMIP έχει αναζωπυρωθεί. Αυτοί που πρωτοστατούν είναι κυρίως μεγάλοι οργανισμοί, οι οποίοι θέλουν να ωφεληθούν από τα προτερήματα του μοντέλου OSI. Αντίθετα, οι μικρότεροι οργανισμοί συνεχίζουν να χρησιμοποιούν κυρίως το πρωτόκολλο SNMP για τη διαχείριση των δικτύων τους.

Αυτή τη στιγμή, τα δίκτυα στα οποία χρησιμοποιείται SNMP πλειοψηφούν σημαντικά. Η εξήγηση μπορεί άνετα να συνδεθεί με το προβάδισμα που πήρε το SNMP έναντι του CMIP όταν παρουσιάστηκε. Τώρα πλέον οι εταιρίες έχουν ήδη επενδύσει αρκετά χρήματα σε συστήματα διαχείρισης και εξοπλισμό που υποστηρίζει το SNMP και το κόστος μετάβασης σε μία άλλη τεχνολογία είναι σημαντικό.

Κεφάλαιο 2

Η ΔΙΑΧΕΙΡΙΣΗ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Στο παρόν κεφάλαιο θα μελετήσουμε την εφαρμογή των λειτουργιών διαχείρισης σε πραγματικά δίκτυα. Το πρώτο βήμα που πρέπει να γίνει, ώστε να μελετηθεί ένα δίκτυο, είναι η πλήρης και συστηματική καταγραφή του. Στην ενότητα 2.1 παρουσιάζεται με λεπτομέρεια ποια είναι η πληροφορία που πρέπει να καταγραφεί, ώστε να μπορέσουμε στη συνέχεια να μελετήσουμε και να διαχειριστούμε ένα δίκτυο.

Μετά την καταγραφή ενός δικτύου, πρέπει να αναλύσουμε ξεχωριστά τον τρόπο με τον οποίο υλοποιείται κάθε λειτουργία διαχείρισης (διαχείριση διαμόρφωσης, διαχείριση ασφάλειας, διαχείριση λαθών, διαχείριση απόδοσης, διαχείριση κόστους) για το συγκεκριμένο δίκτυο. Κατά την ανάλυση αυτή πρέπει να επισημανθούν κενά και παραλείψεις που υπάρχουν σε κάθε λειτουργία ξεχωριστά. Η ενότητα 2.2 περιέχει τη μελέτη ενός πραγματικού δικτύου με βάση την προαναφερθείσα λογική.

Τέλος, στην ενότητα 2.3 θα παρουσιάσουμε πώς προσπελαύνονται τα διαχειριζόμενα αντικείμενα σε δρομολογητές ενός δικτύου. Για την πραγματοποίηση της ενέργειας αυτής είναι απαραίτητη η ύπαρξη ενός κωδικού, που καθιστά δυνατή την πρόσβαση στα διαχειριζόμενα αντικείμενα, και η χρήση κατάλληλου λογισμικού. Στη συγκεκριμένη ενότητα θα αναφέρουμε όλες εκείνες τις ενέργειες στις οποίες προβήκαμε για να εξασφαλίσουμε των κωδικό, να βρούμε το κατάλληλο λογισμικό, να το εγκαταστήσουμε και τελικά να το χρησιμοποιήσουμε.

2.1 Καταγραφή του υπάρχοντος δικτύου [4]

Το πρώτο βήμα για τη μελέτη ενός δικτύου είναι η πλήρης και συστηματική καταγραφή του. Δυστυχώς, σε πολλές περιπτώσεις, ακόμα και εταιρίες που διαθέτουν μεγάλα ιδιόκτητα δίκτυα δεν έχουν καταγεγραμμένη τη ζωτική πληροφορία του δικτύου τους. Επίσης, πολλές φορές γίνεται το λάθος οι εταιρίες να αφήνουν όλα τα στοιχεία για το δίκτυό τους στα χέρια των εταιριών που έχουν αναλάβει τη διαχείρισή του. Είναι σαφές ότι σε τέτοιες περιπτώσεις δημιουργούνται σχέσεις εξάρτησης, στις οποίες οι εταιρίες διαχείρισης έχουν το πάνω χέρι.

Για να γίνει κατανοητό αυτό, μπορούμε να αναλογιστούμε το παρακάτω παράδειγμα. Έστω μία εταιρία Α η οποία θέλει να κατασκευάσει ένα δίκτυο για να εξυπηρετήσει τις ανάγκες της. Η κατασκευή του δικτύου ανατίθεται στην εταιρία Β, η οποία στη συνέχεια αναλαμβάνει και τη διαχείρισή του. Η Α δεν κρατά πουθενά στοιχεία για το δίκτυό της, αφού με αυτό ασχολείται η Β.

Έστω τώρα, ότι μερικά χρόνια αργότερα η Α αποφασίζει να αναθέσει σε άλλη εταιρία τη διαχείριση του δικτύου της, επειδή δε μένει ευχαριστημένη με τις υπηρεσίες που της προσφέρει η Β. Το ερώτημα είναι πώς θα γίνει αυτό. Η Β είναι ο μόνος κάτοχος της πληροφορίας που σχετίζεται με το δίκτυο, οπότε έχει καταφέρει να γίνει κατά κάποιο τρόπο «αναντικατάστατη».

Ένα άλλο σημείο που πρέπει να προσέξουμε, σε σχέση με την καταγραφή των δεδομένων που αφορούν σε ένα δίκτυο, είναι η συνεχής ενημέρωσή τους. Αν καταγράψουμε την πληροφορία που σχετίζεται με το δίκτυο και στη συνέχεια δεν

κάνουμε καμία ενημέρωση για της αλλαγές που γίνονται σε αυτό, είναι βέβαιο ότι μετά από ένα χρονικό διάστημα, η πληροφορία που διαθέτουμε δε θα ανταποκρίνεται στην πραγματικότητα και θα είναι ουσιαστικά άχρηστη.

Από τα παραπάνω, πρέπει να έχει γίνει σαφές ότι είναι απαραίτητη η ύπαρξη μίας πολιτικής ανανέωσης της πληροφορίας. Η υλοποίηση μίας τέτοιας πολιτικής βέβαια προϋποθέτει ότι η εν λόγω πληροφορία είναι καταγεγραμμένη σε αρχεία εύκολα προσβάσιμα από το αρμόδιο προσωπικό που μπορούν να τροποποιηθούν εύκολα.

Μία ακόμα περίπτωση που μπορεί να οδηγήσει σε σοβαρά προβλήματα είναι η ύπαρξη διπλοτύπων της πληροφορίας. Αυτό σημαίνει ότι κρατάμε την ίδια πληροφορία δύο φορές, για παράδειγμα σε δύο διαφορετικές διευθύνσεις της επιχείρησης, χωρίς να ελέγχουμε αν οι δύο εκδόσεις της πληροφορίας είναι συνεπείς μεταξύ τους. Η λύση σε αυτό το πρόβλημα είναι να υπάρχει ένα συγκεντρωτικό σύστημα για την καταγραφή και την ανανέωση της ζωτικής πληροφορίας του δικτύου. Η ύπαρξη ενός τέτοιου συγκεντρωτικού συστήματος αποτρέπει και τον κατακερματισμό της πληροφορίας, οπότε όταν θέλουμε να βρούμε κάτι, ξέρουμε και πού θα το ψάξουμε.

Ανακεφαλαιώνοντας όλα όσα είπαμε παραπάνω, αναφέρουμε τα σημεία που πρέπει να προσέχουμε στη λειτουργία της καταγραφής ενός δικτύου:

- Η ίδια η εταιρία πρέπει να κρατά την πληροφορία για το δίκτυό της.
- Πρέπει να υπάρχει συνεχής ενημέρωση των δεδομένων.
- Τα αρχεία που περιέχουν την πληροφορία πρέπει να είναι εύχρηστα.
- Πρέπει να αποφεύγεται ο κατακερματισμός της πληροφορίας.

Ποια είναι η πληροφορία που καταγράφουμε

Μέχρι τώρα αναφερθήκαμε στη μεγάλη σημασία που έχει η καταγραφή της πληροφορίας που σχετίζεται με το δίκτυο, την οποία αναφέραμε και ως ζωτική πληροφορία, χωρίς όμως να αναφέρουμε ποια είναι ακριβώς η πληροφορία αυτή. Στη συνέχεια θα απαντήσουμε σε αυτό το ερώτημα.

Η πληροφορία που καταγράφουμε για ένα δίκτυο μπορεί να αναλυθεί στις εξής συνιστώσες:

- **Τοπολογία δικτύου**

Αρχικά καταγράφουμε την τοπολογία του δικτύου. Πρέπει να γνωρίζουμε από πόσους κόμβους αποτελείται το δίκτυό μας και που είναι ακριβώς τοποθετημένοι αυτοί οι κόμβοι. Όταν αναφερόμαστε σε δίκτυα ευρείας περιοχής, ένας κόμβος θα είναι ένας δρομολογητής, ενώ όταν αναφερόμαστε σε τοπικά δίκτυα ένας κόμβος μπορεί να αντιστοιχεί, για παράδειγμα, σε ένα hub ή σε ένα απλό τερματικό.

Στην καταγραφή της τοπολογίας είναι σημαντικό να φαίνεται η ιεραρχία που προκύπτει στο δίκτυο, δηλαδή ποιοι κόμβοι του δικτύου ανήκουν στο δίκτυο WAN και ποιο τοπικό δίκτυο συνδέεται σε καθέναν από αυτούς.

Πέραν από τους κόμβους του δικτύου, όπως είναι φυσικό, καταγράφουμε και τις ζεύξεις του.

Η καταγραφή της τοπολογίας ενός δικτύου γίνεται συνήθως σε κατάλληλα διαμορφωμένους πίνακες. Βέβαια, πρέπει να τονίσουμε ότι η παρουσίαση της τοπολογίας ενός δικτύου με σχήμα είναι ιδιαίτερα χρήσιμη, καθώς δίνει μία πλήρη εποπτεία του και μπορεί να οδηγήσει σε χρήσιμα συμπεράσματα για τις αδυναμίες του.

Το ιδανικό θα ήταν το σύστημα που χρησιμοποιείται για την αποθήκευση της τοπολογίας του δικτύου να παράγει αυτόματα και το σχήμα το οποίο αντιστοιχεί σε αυτό.

- **Υλικός εξοπλισμός**

Όταν αναφερόμαστε στον υλικό εξοπλισμό του δικτύου εννοούμε οτιδήποτε έχει σχέση με το υλικό (hardware) που αποτελεί το δίκτυό μας:

- Για κάθε συσκευή που βρίσκεται στο δίκτυο καταγράφουμε τον κατασκευαστή, το μοντέλο και πού είναι τοποθετημένη.

- Για κάθε τερματικό καταγράφουμε το μέγεθος του σκληρού δίσκου και της μνήμης RAM, την ταχύτητα του επεξεργαστή, την κάρτα δικτύου, το μοντέλο της οθόνης, την κάρτα γραφικών κ.τ.λ.

- Για κάθε δρομολογητή καταγράφουμε το μέγεθος του σκληρού δίσκου και της μνήμης RAM, την ταχύτητα του επεξεργαστή, τις κάρτες δικτύου που διαθέτει, τα πρωτόκολλα που υποστηρίζει, την ταχύτητα με την οποία μπορεί να εκπέμψει δεδομένα κ.τ.λ. Όμοια στοιχεία καταγράφουμε και για τις άλλες συσκευές διασύνδεσης που βρίσκονται στο δίκτυο.

- Για κάθε συσκευή του δικτύου καταγράφουμε το όνομά της, καθώς επίσης και τη διεύθυνση που της αντιστοιχεί. Επίσης, αν για την πρόσβαση σε κάποια συσκευή χρειάζεται κάποιος κωδικός, τότε πρέπει να καταγραφεί και αυτός ο κωδικός. Εδώ βέβαια πρέπει να είμαστε ιδιαίτερα προσεκτικοί, καθώς πρέπει να εξασφαλίσουμε ότι οι διάφοροι κωδικοί δε θα γίνουν γνωστοί σε μη εξουσιοδοτημένα άτομα.

- Για κάθε συσκευή που βρίσκεται στο δίκτυο καταγράφουμε τις κενές θέσεις που έχει για την προσθήκη νέων καρτών. Η πληροφορία αυτή είναι ιδιαίτερα χρήσιμη όταν επεκτείνουμε το δίκτυο. Για παράδειγμα, όταν θέλουμε να συνδέσουμε ένα καινούριο τοπικό δίκτυο σε έναν υπάρχοντα δρομολογητή του δικτύου μας, αν υπάρχει κενή θέση στο δρομολογητή η όλη διαδικασία μπορεί να είναι αρκετά εύκολη, ενώ αν δεν υπάρχει, θα πρέπει να αγοράσουμε νέο δρομολογητή.

- Καταγράφουμε τους τύπους καλωδίων που χρησιμοποιούνται σε κάθε περίπτωση (π.χ. UTP, οπτική ίνα κ.τ.λ.).

- Καταγράφουμε όλες τις πρίζες που υπάρχουν στο χώρο της επιχείρησης, καθώς επίσης και τη θέση τους στο χώρο. Οι πρίζες μπορούν να είναι πρίζες για ηλεκτρικό ρεύμα ή για φωνή/δεδομένα.

- Καταγράφουμε όλο εκείνο τον εξοπλισμό τον οποίο διαθέτουμε και δε χρησιμοποιούμε. Με την πληροφορία αυτή μπορούμε να βρούμε αμέσως αν διαθέτουμε κάποιο ανταλλακτικό που χρειαζόμαστε, σε περίπτωση βλάβης. Επίσης, αν ο εφεδρικός εξοπλισμός είναι τοποθετημένος σε περισσότερα του ενός σημεία, πρέπει να αναφέρουμε και πού βρίσκεται.

➢ **Λογισμικό**

Καταγράφουμε κάθε είδους λογισμικό που χρησιμοποιείται στην επιχείρηση. Πιο συγκεκριμένα καταγράφουμε τον κατασκευαστή κάθε προγράμματος, την έκδοσή του, πληροφορίες σχετικές με τα δικαιώματα χρήσης του κ.τ.λ.

➢ **Ζεύξεις**

Πέραν των καλωδίων που χρησιμοποιούμε στα LAN, καταγράφουμε και τις ζεύξεις των WAN. Οι ζεύξεις αυτές σπάνια είναι ιδιόκτητες. Συνήθως μισθώνονται από κάποιο πάροχο. Καταγράφουμε τον πάροχο, το κόστος και τον τύπο κάθε ζεύξης.

➢ **Πρωτόκολλα**

Καταγράφουμε όλα τα πρωτόκολλα τα οποία χρησιμοποιούνται στο δίκτυο της επιχείρησης.

➢ **Στατιστικά**

Μία πολύ σημαντική πληροφορία η οποία σχετίζεται με το δίκτυο είναι τα στατιστικά στοιχεία. Τα στατιστικά στοιχεία μπορούν να αναφέρονται:

- Στο φόρτο των ζεύξεων. Η πληροφορία αυτή είναι πολύ σημαντική, αφού μπορεί να αποτελέσει τη βάση για την απόφαση αναβάθμισης ή υποβάθμισης μίας ζεύξης.

- Στην κίνηση που δημιουργεί κάθε εφαρμογή.

- Στη χρησιμοποίηση του δικτύου ανά χρήστη/πελάτη. Η πληροφορία αυτή αποτελεί τη βάση για την ανάλυση κόστους.

- Στη διαθεσιμότητα σε επίπεδο στοιχείων δικτύου, δικτύου και υπηρεσιών.

- Στις βλάβες που παρατηρούνται.
- Στην καθυστέρηση μετάδοσης που παρατηρείται.
- Στις ώρες αιχμής για το δίκτυο.

2.2 Ιδιωτικά δίκτυα WAN μεγάλων εταιριών

Κατά τη διάρκεια εκπόνησης της παρούσας εργασίας επισκεφτήκαμε το κέντρο διαχείρισης δικτύου μίας μεγάλης εταιρίας. Η συγκεκριμένη εταιρία διαθέτει ένα από τα μεγαλύτερα ιδιωτικά δίκτυα στην Ελλάδα. Οι κύριοι στόχοι μας ήταν:

1) Η μελέτη του δικτύου της εταιρίας.
2) Η μελέτη του τρόπου με τον οποίο διαχειρίζεται η συγκεκριμένη εταιρία το δίκτυό της.
3) Η επισήμανση προβλημάτων που προκύπτουν κατά τη διαδικασία διαχείρισης ενός πραγματικού δικτύου.
4) Η επισήμανση κενών που υπάρχουν στη διαδικασία διαχείρισης.
5) Η συγκέντρωση στατιστικών που σχετίζονται με τη διαθεσιμότητα του δικτύου της εταιρίας.

Στον τελευταίο στόχο της επίσκεψής μας θα αναφερθούμε αναλυτικά σε άλλο σημείο της εργασίας, ενώ με τους υπόλοιπους τέσσερις θα ασχοληθούμε στη συνέχεια.

2.2.1 Καταγραφή δικτύου επιχείρησης

Στην παράγραφο αυτή θα περιγράψουμε το δίκτυο της επιχείρησης την οποία επισκεφτήκαμε. Μεγαλύτερη έμφαση θα δοθεί στην περιγραφή του δικτύου ευρείας περιοχής.

Τοπολογία δικτύου επιχείρησης

Το δίκτυο ευρείας περιοχής της συγκεκριμένης επιχείρησης μπορεί να χωριστεί σε τρία μέρη:

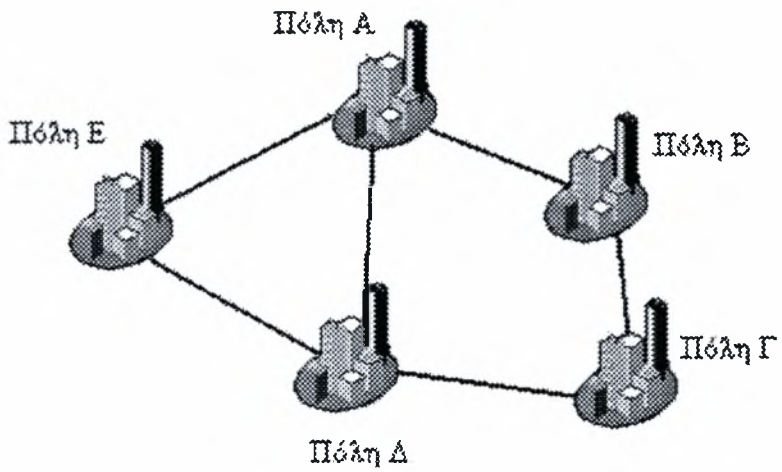
- Το δίκτυο κορμού
- Το δίκτυο περιοχής

- Το δίκτυο καταστημάτων

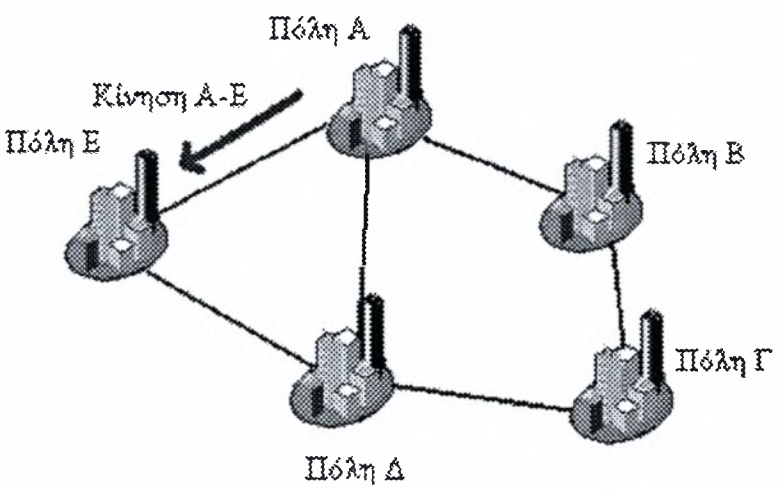
Στη συνέχεια θα αναφερθούμε ξεχωριστά στην τοπολογία του κάθε μέρους.

➢ **Δίκτυο κορμού**

Το δίκτυο κορμού (Back Bone Network) της επιχείρησης έχει μορφή πλέγματος, όμοια με αυτή που παρουσιάζεται στο σχήμα 2.1.α. Οι κόμβοι του δικτύου κορμού βρίσκονται σε μεγάλες πόλεις της Ελλάδας. Ο λόγος που επιλέχθηκε αυτή η τοπολογία είναι προφανής: αν υπάρξει πρόβλημα σε κάποιο κόμβο ή ζεύξη, η κίνηση μπορεί να διοχετευθεί από εναλλακτικές διαδρομές.



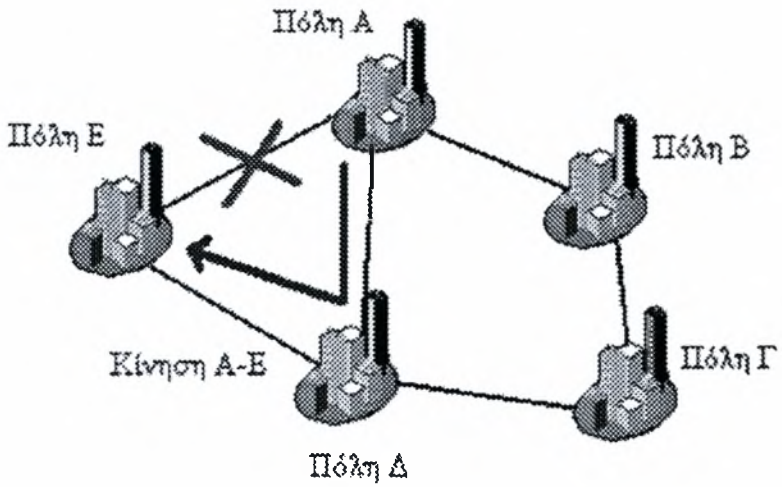
Σχήμα 2.1.α: *Τοπολογία δικτύου κορμού*



Σχήμα 2.1.β: *Φυσιολογική διοχέτευση κίνησης Α–Ε*

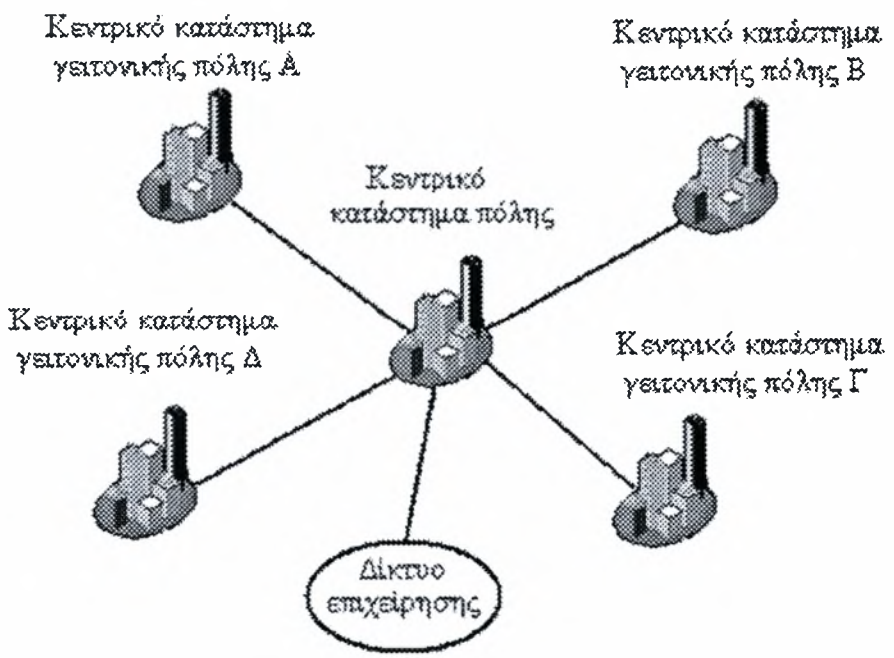
Η περίπτωση αυτή παρουσιάζεται στα σχήματα 2.1.β και 2.1.γ. Το σχήμα 2.1.β δείχνει ότι όταν δεν υπάρχει κανένα πρόβλημα στο δίκτυο, η κίνηση από την πόλη Α στην πόλη Ε διοχετεύεται μέσω της ζεύξης Α–Ε.

Το σχήμα 2.1.γ δείχνει την περίπτωση που η ζεύξη Α–Ε δεν είναι διαθέσιμη λόγω κάποιας βλάβης. Τότε η κίνηση από την πόλη Α στην πόλη Ε διοχετεύεται στον προορισμό της μέσω της εναλλακτικής διαδρομής Α→ Δ→ Ε.



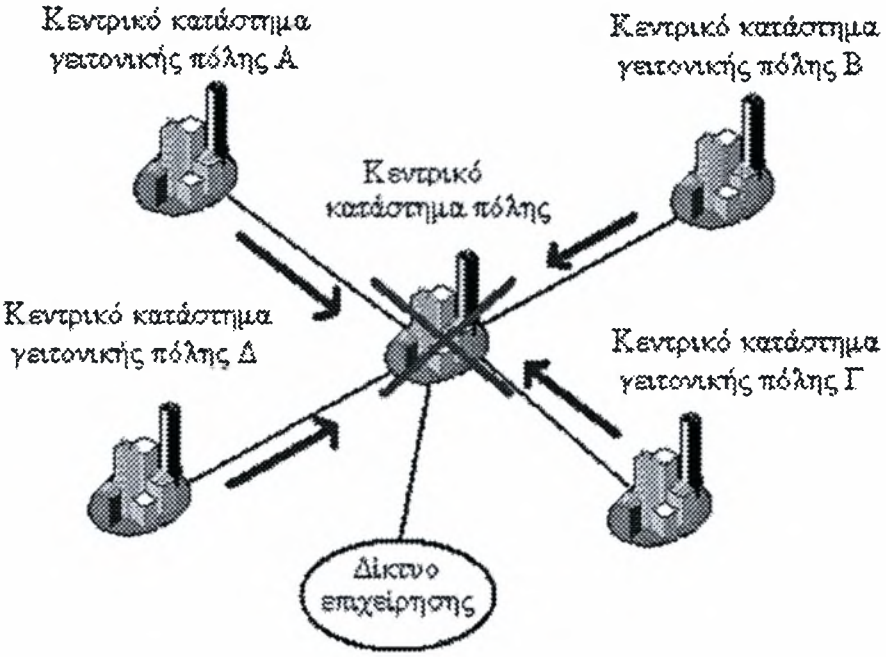
Σχήμα 2.1.γ: *Εναλλακτική διοχέτευση κίνησης Α–Ε σε περίπτωση βλάβης*

- **Δίκτυο περιοχής**



Σχήμα 2.2.α: *Τοπολογία δικτύου περιοχής*

Το δίκτυο περιοχής της επιχείρησης (Regional Office Network) συνδέει κεντρικά καταστήματα τα οποία βρίσκονται σε γειτονικές πρωτεύουσες νομών της Ελλάδας ή σε αρκετά μεγάλες γειτονικές πόλεις. Υπάρχουν περίπου 110 τέτοια καταστήματα. Το δίκτυο περιοχής έχει μορφή αστέρα, όπως αυτή που φαίνεται στο σχήμα 2.2.α.

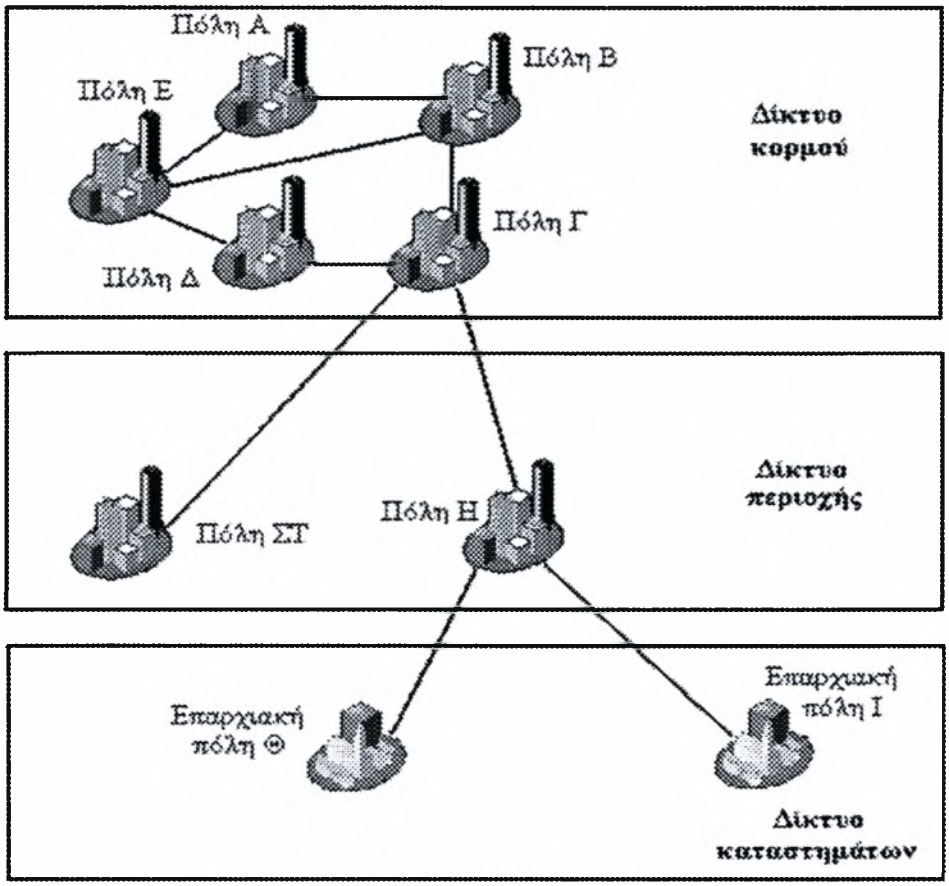


Σχήμα 2.2.β: *Κατάρρευση κεντρικού κόμβου σε τοπολογία αστέρα*

Η τοπολογία αστέρα δημιουργεί μεγάλα προβλήματα διαθεσιμότητας στην περίπτωση που καταρρεύσει ο κεντρικός κόμβος του αστέρα. Αυτό παρουσιάζεται στο σχήμα 2.2.β, όπου δε λειτουργεί ο κόμβος που αντιστοιχεί στο κεντρικό κατάστημα της πρωτεύουσας του νομού. Είναι φανερό ότι τα καταστήματα των γειτονικών πόλεων τα οποία είναι συνδεδεμένα στον κόμβο αυτό, δε μπορούν να έχουν καμία επικοινωνία, ούτε μεταξύ τους, ούτε με το υπόλοιπο δίκτυο της επιχείρησης. Η λύση για να αυξήσουμε τη διαθεσιμότητα σε τέτοιες περιπτώσεις είναι η δημιουργία τοπολογίας πλέγματος. Βέβαια, το κόστος μίας τέτοιας λύσης είναι αυξημένο και για αυτό η συγκεκριμένη επιχείρηση δεν τη χρησιμοποιεί. Πρέπει πάντως να σημειωθεί ότι σε αρκετές περιπτώσεις υπάρχουν εφεδρικές ISDN γραμμές. Στο θέμα αυτό θα αναφερθούμε όταν καταγράψουμε τις ζεύξεις του δικτύου.

- **Δίκτυο καταστημάτων**

Το δίκτυο καταστημάτων (Branch office Network) συνδέει τα μικρά καταστήματα, τα οποία βρίσκονται συνήθως σε μικρές επαρχιακές πόλεις, με τα κεντρικά καταστήματα του δικτύου περιοχής. Υπάρχουν περίπου 480 μικρά καταστήματα και η σύνδεσή τους με τα κεντρικά της περιοχής τους γίνεται με τοπολογία αστέρα. Φυσικά, και σε αυτή την περίπτωση, για την τοπολογία αστέρα ισχύει ότι αναφέραμε και προηγουμένως. Η συνολική μορφή που έχει το δίκτυο ευρείας περιοχής της εταιρίας φαίνεται στο σχήμα 2.3.



Σχήμα 2.3: Συνολική μορφή δικτύου WAN

Τέλος, σε κάθε κατάστημα υπάρχει ένα τοπικό δίκτυο. Ο αριθμός των τερματικών σε κάθε κατάστημα ποικίλει ανάλογα με το μέγεθός του. Για παράδειγμα, υπάρχουν καταστήματα που έχουν μόνο 5 τερματικά και καταστήματα που έχουν 120 τερματικά. Τα τοπικά δίκτυα έχουν τοπολογία αστέρα.

Υλικός εξοπλισμός επιχείρησης

- Στο δίκτυο κορμού της επιχείρησης υπάρχουν 25 ATM IGX switches της Cisco.
- Η δρομολόγηση στο δίκτυο περιοχής γίνεται με τη χρήση δρομολογητών 6560 ή 6455 της Motorola.
- Η δρομολόγηση στο δίκτυο καταστημάτων γίνεται με τη χρήση δρομολογητών 6520 ή 6435 της Motorola.
- Τα hubs που υπάρχουν στα τοπικά δίκτυα είναι της 3Com.
- Στα τοπικά δίκτυα για τη σύνδεση των συσκευών χρησιμοποιείται στην πλειοψηφία των περιπτώσεων καλώδιο UTP και σε κάποιες περιπτώσεις οπτική ίνα.

- Τέλος σημειώνεται η ύπαρξη δύο SNA mainframes, καθώς επίσης και 113 εξυπηρετητών αλληλογραφίας, οι οποίοι βρίσκονται διάσπαρτοι στο δίκτυο της επιχείρησης.

Το λογισμικό της επιχείρησης

- Το λειτουργικό σύστημα που χρησιμοποιείται στα τερματικά της επιχείρησης είναι τα ***Windows NT*** και ***XP***. Επίσης υπάρχουν και τερματικά που λειτουργούν σε περιβάλλον ***Unix***.

- Οι ***Transactions Services*** είναι οι εφαρμογές που χρησιμοποιούνται για να πραγματοποιηθούν οι συναλλαγές της επιχείρησης με τους πελάτες της.

- Για το χειρισμό των λογιστικών θεμάτων γίνεται χρήση της εφαρμογής ***SAP***.

- Τα καταστήματα της επιχείρησης είναι δυνατό να επικοινωνήσουν μεταξύ τους με τη χρήση της εφαρμογής ***SwiftNet***.

- Για την υπηρεσία ηλεκτρονικού ταχυδρομείου χρησιμοποιείται η πλατφόρμα ***Exchange Server***.

- Η περιήγηση στο διαδίκτυο γίνεται με τη χρήση φυλλομετρητών (***Web Browsers***).

- Επειδή το ενδιαφέρον της επιχείρησης για οικονομικές ειδήσεις είναι μεγάλο, χρησιμοποιεί την πλατφόρμα ***Bloomberg*** για να έχει on-line οικονομικές ειδήσεις.

- Το ***What's up*** χρησιμοποιείται για να παρέχει εποπτεία – μέσω ενός γραφικού περιβάλλοντος – του δικτύου της επιχείρησης και άμεσο εντοπισμό των προβλημάτων που παρουσιάζονται.

- Τα ***RDD tools*** είναι εργαλεία τα οποία χρησιμοποιούνται για την παραγωγή στατιστικών που αφορούν στο δίκτυο.

Οι ζεύξεις του δικτύου ευρείας περιοχής της επιχείρησης

- Στο δίκτυο κορμού της επιχείρησης χρησιμοποιούνται, τόσο ενσύρματες όσο και ασύρματες ζεύξεις. Οι ταχύτητες των ζεύξεων κυμαίνονται από 2 έως 155 Mbps, ανάλογα με τις ανάγκες που πρέπει να καλυφθούν. Οι γραμμές αυτές είναι γραμμές ATM που μισθώνονται από τον Ο.Τ.Ε.

- Στο δίκτυο περιοχής της επιχείρησης χρησιμοποιούνται μισθωμένες γραμμές του Ο.Τ.Ε., τύπου Hellascom. Οι γραμμές αυτές έχουν ταχύτητα από 256 έως

512 Kbps. Επιπλέον, υπάρχουν και εναλλακτικές γραμμές ISDN, σε περίπτωση που παρουσιαστεί πρόβλημα στις κανονικές ζεύξεις.

- Στο δίκτυο καταστημάτων της επιχείρησης χρησιμοποιούνται, όπως και στο δίκτυο περιοχής, μισθωμένες γραμμές του Ο.Τ.Ε., τύπου Hellascom. Οι γραμμές αυτές έχουν ταχύτητα από 64 έως 512 Kbps. Επιπλέον, μπορεί να υπάρχουν και εναλλακτικές γραμμές ISDN, σε περίπτωση που παρουσιαστεί πρόβλημα στις κανονικές ζεύξεις.

Χρησιμοποιούμενα πρωτόκολλα στο δίκτυο της επιχείρησης

- Το πρωτόκολλο που χρησιμοποιείται για να γίνει η επικοινωνία στο δίκτυο κορμού της εταιρίας είναι το ***ATM***.

- Στα δίκτυα περιοχής και καταστημάτων χρησιμοποιείται το πρωτόκολλο ***Frame Relay***.

- Όπως είπαμε οι συναλλαγές γίνονται μέσω της εφαρμογής SAP. Η εφαρμογή αυτή χρησιμοποιεί το πρωτόκολλο ***SNA*** της IBM.

- Οι εφαρμογές που σχετίζονται με τη διαχείριση του δικτύου της εταιρίας χρησιμοποιούν το πρωτόκολλο ***SNMP***.

- Το πρωτόκολλο ***IP*** χρησιμοποιείται σε υπηρεσίες, όπως αυτή του ηλεκτρονικού ταχυδρομείου και της περιήγησης στο διαδίκτυο.

- Το πρωτόκολλο που χρησιμοποιείται στα τοπικά δίκτυα της εταιρίας είναι το ***Ethernet***. Αξίζει να σημειωθεί ότι μερικά χρόνια πριν, σε κάποια τοπικά δίκτυα, χρησιμοποιούνταν και το πρωτόκολλο Token Ring. Σήμερα, σε όλα τα τοπικά δίκτυα της εταιρίας χρησιμοποιείται, είτε το 10BaseX Ethernet, είτε το 100BaseX Ethernet.

Στατιστικά για το δίκτυο της επιχείρησης

Η εταιρία κρατά στατιστικά για την απόδοση και τη διαθεσιμότητα του δικτύου της. Για παράδειγμα, τα RDD tools μπορούν να χρησιμοποιηθούν για να δώσουν τη γραφική απεικόνιση της χρήσης του επεξεργαστή ενός δρομολογητή. Ομοίως, μπορούν να δώσουν γραφικές παραστάσεις για το φόρτο των ζεύξεων.

Επιπλέον, μπορεί να γίνει παρουσίαση των στατιστικών σε διαφορετικές κλίμακες χρόνου, δηλαδή ανά μέρα, ανά εβδομάδα κ.τ.λ., καθώς επίσης και να συγκεντρωθούν οι μέγιστες τιμές για κάποια δεδομένα. Για παράδειγμα, μπορούμε να δούμε το μέγιστο ρυθμό εισερχόμενων δεδομένων σε κάποιο δρομολογητή.

Τα δεδομένα αυτά αποτελούν τη βάση για αποφάσεις που έχουν σχέση με αλλαγές που πρέπει να γίνουν στο δίκτυο, με στόχο τη βελτίωση της λειτουργίας του.

2.2.2 Λειτουργίες διαχείρισης στην επιχείρηση

Σκοπός μας στην παρούσα υπο-ενότητα είναι να παρουσιάσουμε τον τρόπο με τον οποίο υλοποιούνται κάποιες από τις λειτουργίες διαχείρισης, στην επιχείρηση που επισκεφτήκαμε.

Επίσης, θα κάνουμε κάποιες παρατηρήσεις σε σχέση με τη διαχείριση του δικτύου της συγκεκριμένης εταιρίας και με παραλείψεις που υπάρχουν. Τέλος, θα επισημάνουμε γενικότερα ζητήματα που προκύπτουν κατά τη διαχείριση μεγάλων δικτύων.

Ασφάλεια

Η ασφάλεια θεωρείται ύψιστης σημασίας στη εν λόγω εταιρία, καθώς μέσω του δικτύου της διακινούνται ευαίσθητα δεδομένα των πελατών της. Η ασφάλεια σε φυσικό επίπεδο υλοποιείται στο κέντρο δικτύου της με δύο κυρίως τρόπους:

- Υπάρχει φύλαξη του κτηρίου, με αποτέλεσμα μόνο εξουσιοδοτημένα άτομα να μπορούν να έχουν πρόσβαση σε αυτό.

- Ακόμα και αν κάποιος μπει στο κτήριο, δε σημαίνει ότι έχει πρόσβαση σε όλους τους χώρους του. Και εδώ υπάρχουν συστήματα ασφαλείας, τα οποία επιτρέπουν την είσοδο σε χώρους όπου φυλάσσονται ευαίσθητα δεδομένα μόνο σε συγκεκριμένα άτομα.

Επιπλέον, κάθε εργαζόμενος έχει ένα δικό του password με το οποίο διασφαλίζεται ότι αυτός και μόνο αυτός έχει πρόσβαση στα δεδομένα που βρίσκονται στο profile του.

Η ασφάλεια για τα δεδομένα που κινούνται στις ζεύξεις του δικτύου εξασφαλίζεται από τις μισθωμένες γραμμές του Ο.Τ.Ε, αφού ο πάροχος έχει δεσμευτεί για αυτό.

Σε αυτό το σημείο θα κάνουμε κάποιες γενικότερες επισημάνσεις για την ασφάλεια σε μεγάλα δίκτυα. Το πρώτο πράγμα που πρέπει να προσέξουμε είναι ότι η ασφάλεια δεν είναι κάτι που μπορούμε να εξασφαλίσουμε με αποσπασματικά μέτρα. Σε κάθε περίπτωση πρέπει να υπάρχει μία γενικότερη πολιτική ασφάλειας, η οποία θα προλαμβάνει κάθε πιθανό κίνδυνο. Δυστυχώς πολλές εταιρίες ακόμα και σήμερα δε διαθέτουν μία σαφή πολιτική ασφαλείας για το δίκτυό τους

Για παράδειγμα, αφού μία πολιτική ασφαλείας ορίζει ρητά ποιος έχει πρόσβαση σε τι, θα πρέπει να υπάρχει μέριμνα για την προστασία της ευαίσθητης πληροφορίας για όσο χρονικό διάστημα αυτή υφίσταται, από τη στιγμή της δημιουργίας της μέχρι τη στιγμή της καταστροφής της. Αυτό θα μπορούσε να μεταφραστεί σε ένα κανόνα ασφαλείας ο οποίος προβλέπει την καταστροφή των αναφορών που αφορούν σε στοιχεία του δικτύου, μετά την πάροδο ορισμένου χρονικού διαστήματος, εντός του κτηρίου της επιχείρησης. Με τον τρόπο αυτό διασφαλίζεται ότι άτομα που δεν μπορούν να εισέλθουν στο κτήριο δεν θα έχουν ποτέ πρόσβαση στις αναφορές.

Το παραπάνω παράδειγμα κάνει σαφές ότι ακόμα και αν υπήρχε μέριμνα για τη σωστή φύλαξη των αναφορών μέχρι τη στιγμή της καταστροφής τους, αλλά τελικά η διαδικασία καταστροφής γίνονταν εκτός του κτηρίου της επιχείρησης, δίνοντας την

ευκαιρία σε μη εξουσιοδοτημένα άτομα να αποκτήσουν πρόσβαση σε αυτές, θα μπορούσε να παρουσιαστεί παραβίαση των κανόνων ασφαλείας. Το γεγονός αυτό αναδεικνύει και μία ακόμα σημαντική πτυχή που σχετίζεται με την ασφάλεια. Η πολιτική ασφάλειας είναι τόσο ισχυρή όσο το πιο αδύναμο σημείο της. Είναι απαραίτητο σε μία πολιτική ασφαλείας να εντοπίζονται οι αδυναμίες και να γίνονται οι απαραίτητες βελτιώσεις.

Διαχείριση σφαλμάτων

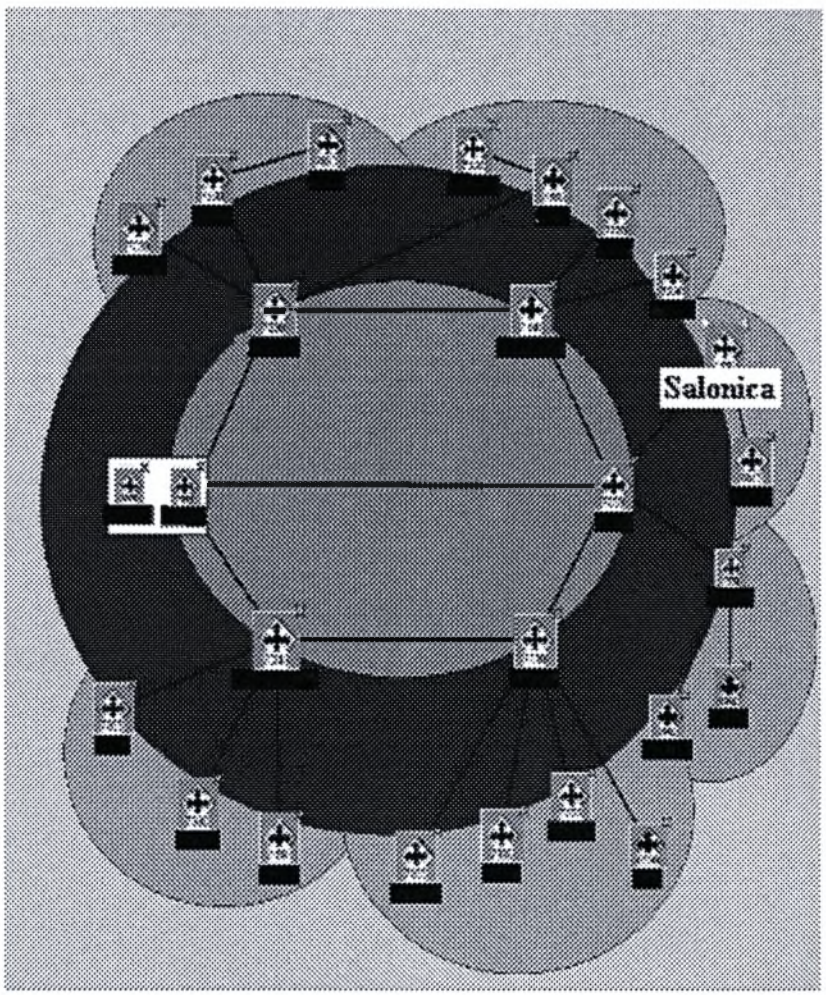
Το σύστημα που χρησιμοποιείται στην εταιρία που επισκεφτήκαμε για τον εντοπισμό προβλημάτων σε πραγματικό χρόνο είναι το What's up, το οποίο λειτουργεί με χρήση του πρωτοκόλλου SNMP. Η τοπολογία του δικτύου της εταιρίας βρίσκεται αποθηκευμένη σε πίνακες excel. Το What's up παρέχει ένα γραφικό περιβάλλον με το οποίο μπορεί κάποιος να σχεδιάσει ένα συγκεκριμένο δίκτυο. Επισημαίνεται ότι η ενημέρωση του What's up για αλλαγές στο δίκτυο δε γίνεται αυτόματα μέσω των πινάκων excel – όπως θα ήταν το ιδανικό – αλλά με το χέρι.

Η λογική με την οποία σχεδιάζεται το δίκτυο για πρώτη φορά στο What's up περιγράφεται στη συνέχεια: Υπάρχουν εικονίδια τα οποία αντιστοιχούν σε δρομολογητές. Επιλέγεις για κάθε δρομολογητή του πραγματικού δικτύου ένα εικονίδιο και παρέχεις σε αυτό τα στοιχεία που αντιστοιχούν στον πραγματικό δρομολογητή (IP διεύθυνση, θύρες που λειτουργούν κ.τ.λ.). Στη συνέχεια, ενώνεις τα εικονίδια με γραμμές που αντιστοιχούν στις πραγματικές ζεύξεις. Το δίκτυο που προκύπτει στο γραφικό περιβάλλον της εφαρμογής βρίσκεται πλέον σε επικοινωνία με το πραγματικό δίκτυο. Αν γίνει εισαγωγή δεδομένων που δεν ανταποκρίνονται στην πραγματικότητα (για παράδειγμα ότι υπάρχει μία ζεύξη στο πραγματικό δίκτυο που στην πραγματικότητα δεν υφίσταται) τότε το αντίστοιχο μέρος του γραφικού περιβάλλοντος χρωματίζεται κόκκινο. Αυτός είναι στην ουσία και ο τρόπος με τον οποίο το σύστημα ειδοποιεί ότι σε κάποιο μέρος του δικτύου υπάρχει πρόβλημα: το αντίστοιχο τμήμα του δικτύου κοκκινίζει. Επιπλέον, υπάρχει η δυνατότητα κάθε φορά που εντοπίζεται ένα πρόβλημα να σημαίνει και ηχητικός συναγερμός.

Το What's up δίνει τη δυνατότητα μίας ιεραρχικής παρουσίασης του δικτύου. Αυτό σημαίνει ότι αρχικά βλέπουμε το δίκτυο κορμού. Αν παρουσιαστεί κάποιο πρόβλημα, είτε σε έναν από του δρομολογητές που βλέπουμε, είτε σε κάποιον από τους δρομολογητές ή τις ζεύξεις που δε φαίνονται και ανήκουν σε υποδέντρο που ξεκινά από το συγκεκριμένο δρομολογητή, τότε αυτός χρωματίζεται κόκκινος. Αν το πρόβλημα βρίσκεται στο υποδίκτυο «κάτω» από το δρομολογητή που βλέπουμε, τότε αυτό εμφανίζεται «κλικάροντας» πάνω του. Το μέρος του υποδικτύου που έχει το πρόβλημα είναι χρωματισμένο κόκκινο. Στο σχήμα 2.4 φαίνεται η μορφή που έχει το απεικονιζόμενο δίκτυο στο περιβάλλον What's up. Στη συγκεκριμένη περίπτωση φαίνεται το δίκτυο κορμού της επιχείρησης. Σημειώνεται ότι έχουν σβηστεί πολλά από τα χαρακτηριστικά του δικτύου για λόγους ασφαλείας. Παρόλα αυτά, έχουμε κρατήσει το όνομα του κόμβου που αντιστοιχεί στη Θεσσαλονίκη. Ο λόγος που έγινε αυτό είναι για να επισημάνουμε ότι ο συγκεκριμένος κόμβος δεν ανήκει στο πλέγμα που σχηματίζει το δίκτυο κορμού, όπως θα περίμενε κανείς, προφανώς λόγω κόστους. Πρόκειται για μία σχεδιαστική απόφαση που μας έκανε εντύπωση και η οποία ίσως θα έπρεπε να επανεξεταστεί.

Σημειώνεται, ότι αυτό που δείχνει το What's up είναι *το **μέρος του δικτύου που έχει το πρόβλημα*** και όχι ποια είναι ***η αιτία του προβλήματος***. Αυτό σημαίνει ότι,

είτε ένας δρομολογητής έχει πρόβλημα, είτε η ζεύξη που οδηγεί σε αυτόν, στο γραφικό περιβάλλον της εφαρμογής και τα δύο χρωματίζονται κόκκινα. Το γεγονός αυτό επιβάλει να εντοπιστεί με ακρίβεια η αιτία του προβλήματος και στη συνέχεια να αποκατασταθεί η βλάβη. Η διαδικασία που ακολουθείται σε γενικές γραμμές έχει ως εξής:



Σχήμα 2.4: *Μορφή δικτύου κορμού επιχείρησης*

- Αρχικά λαμβάνει χώρα μία τηλεφωνική επικοινωνία ανάμεσα στο υποκατάστημα που αντιστοιχεί στον κόμβο που χρωματίζεται κόκκινος και στο κέντρο δικτύου της επιχείρησης. Κύριος σκοπός αυτής της τηλεφωνικής επικοινωνίας είναι να ελεγχθεί αν υπάρχει πρόβλημα με την τάση ρεύματος στο συγκεκριμένο υποκατάστημα. Αν αναφερθεί από την πλευρά του υποκαταστήματος ότι δεν παρατηρείται κανένα πρόβλημα που να δικαιολογεί τη διακοπή της επικοινωνίας, περνάμε στο δεύτερο βήμα.

- Το δεύτερο βήμα για τον εντοπισμό της αιτίας της βλάβης είναι να ελεγχθεί αν οι μισθωμένες ζεύξεις του παρόχου λειτουργούν. Αυτό γίνεται με τηλεφωνική επικοινωνία ανάμεσα στο κέντρο δικτύου της επιχείρησης και στον πάροχο.

- Αν ο πάροχος των μισθωμένων γραμμών δηλώσει ότι το πρόβλημα δεν οφείλεται σε αυτόν, το συμπέρασμα είναι ότι το πρόβλημα βρίσκεται στον ιδιόκτητο δρομολογητή της εταιρίας, οπότε καλείται η εταιρία που είναι υπεύθυνη για την τεχνική υποστήριξη του δικτύου για να τον επισκευάσει.

Από την παραπάνω διαδικασία πρέπει να δοθεί ιδιαίτερη προσοχή σε μία λεπτομέρεια. Η εταιρία που έχει αναλάβει την τεχνική υποστήριξη του δικτύου καλείται τελευταία. Ο λόγος είναι προφανής: οι υπηρεσίες της χρεώνονται.

Τέλος, αναφέρουμε ότι η συγκεκριμένη εταιρία στην οποία αναφερόμαστε, κρατά ξεχωριστά αρχεία όπου καταχωρούνται οι βλάβες, οι χρόνοι και οι αιτίες τους. Το συγκεκριμένο σύστημα ticketing δε συνεργάζεται με το What's up.

Αυτά που αναφέραμε παραπάνω δίνουν αφορμή να μιλήσουμε για δύο μεγάλα θέματα που προκύπτουν κατά τη διαχείριση σφαλμάτων σε μεγάλα δίκτυα:

Το πρώτο είναι η δυσκολία στον εντοπισμό της αιτίας του σφάλματος. Πολλές φορές το δίκτυο μίας εταιρίας μπορεί να χωρίζεται σε πολλούς τομείς, είτε γεωγραφικούς, είτε λειτουργικούς και για κάθε τομέα να είναι υπεύθυνη μία διαφορετική εταιρία τεχνικής υποστήριξης. Αν δεν είναι γνωστή η αιτία του προβλήματος, τότε πρέπει να κληθούν όλες τις εταιρίες, ώστε να βρεθεί σε ποιο τομέα βρίσκεται το πρόβλημα και να αποκατασταθεί. Η τακτική που μπορεί να ακολουθείται σε αυτή την περίπτωση μπορεί να πάρει δύο μορφές:

- Καλούνται όλες οι εταιρίες τεχνικής υποστήριξης μαζί. Η τακτική αυτή έχει ως αποτέλεσμα το πρόβλημα να εντοπίζεται και να επισκευάζεται στο λιγότερο δυνατό χρόνο, αλλά προφανώς είναι ιδιαίτερα δαπανηρή, αφού θα πρέπει να πληρωθούν όλες οι εταιρίες.

- Η τακτική που χρησιμοποιείται συνήθως είναι να καλείται μία εταιρία τεχνικής υποστήριξης τη φορά και να διαπιστώνει αν ο τομέας της λειτουργεί κανονικά. Η διαδικασία επαναλαμβάνεται μέχρι να εντοπιστεί η βλάβη. Το πρόβλημα σε αυτή τη διαδικασία – πέραν του γεγονότος ότι δεν υπάρχει διαβεβαίωση ότι δε θα κληθούν και πάλι εταιρίες που δεν είναι υπεύθυνες για τη βλάβη – είναι ότι ο χρόνος αποκατάσταση της βλάβης αυξάνεται πολύ.

Το δεύτερο θέμα στο οποίο θα αναφερθούμε είναι αυτό του ticketing των βλαβών. Κανονικά ένα ticket (καταχώριση) μίας βλάβης πρέπει να περιέχει το χρόνο που παρουσιάστηκε και τα συμπτώματά της. Επίσης, πρέπει να συμπληρωθούν από τον τεχνικό που έκανε την επισκευή το όνομά του, η ακριβής αιτία της βλάβης και ο ακριβής χρόνος επιδιόρθωσης. Η όλη διαδικασία «χωλαίνει» όμως συνήθως στη μη παροχή έγκυρων στοιχείων από τους τεχνικούς. Οι βασικοί λόγοι είναι δύο:

- Η όλη διαδικασία θεωρείται απλώς τυπική, οπότε κανείς δε μπαίνει στον κόπο να καταγράψει με ακρίβεια τα πραγματικά αίτια της βλάβης και το χρόνο που πήρε η αποκατάστασή της.

- Οι εταιρίες τεχνικής υποστήριξης έχουν υπογράψει συμβόλαια στα οποία προβλέπεται ο μέγιστος χρόνος που πρέπει να κάνουν για να επισκευάσουν μία βλάβη. Υπέρβαση του χρόνου μπορεί να έχει ως αποτέλεσμα την επιβολή

χρηματικής αποζημίωσης, οπότε αν μπορούν, «τροποποιούν» κάπως τον πραγματικό χρόνο επιδιόρθωσης προς το συμφέρον τους.

Απόδοση

Όπως είπαμε στη συγκεκριμένη εταιρία κρατούνται στατιστικά για το δίκτυο, τα οποία σχετίζονται με την απόδοσή του. Τα στατιστικά αυτά αναφέρονται στο φόρτο στις ζεύξεις, στην καθυστέρηση μετάδοσης, στη διαθεσιμότητα κ.τ.λ. Μία παρατήρηση που έχουμε να κάνουμε είναι ότι ενώ υπάρχουν στατιστικά για τη διαθεσιμότητα του δικτύου, δεν υπάρχουν στατιστικά που να μας λένε ποιος προκάλεσε τη βλάβη που οδήγησε στην έλλειψη διαθεσιμότητας. Δηλαδή, δεν είναι γνωστό πόσες βλάβες οφείλονται στον πάροχο των μισθωμένων γραμμών και πόσες στον ιδιόκτητο εξοπλισμό της εταιρίας.

Γενικά, είναι πλέον κανόνας ότι στις επιχειρήσεις που διαθέτουν δικό τους δίκτυο κρατούνται κάποια στατιστικά για την απόδοσή του. Σε αυτή την κατεύθυνση βοηθά ότι υπάρχουν πια αρκετά εργαλεία, τα οποία είναι εύχρηστα και μπορούν να χρησιμοποιηθούν για το σκοπό αυτό.

Βέβαια, πρέπει να σημειωθεί ότι η ύπαρξη των εργαλείων από μόνη της δεν είναι αρκετή για την παραγωγή αποτελεσμάτων που θα μπορούσαν να αξιοποιηθούν. Οι επιχειρήσεις που πραγματικά θέλουν να έχουν μία σαφή εικόνα για τις αποδόσεις του δικτύου τους θα πρέπει να επενδύσουν και στην απόκτηση εξειδικευμένου προσωπικού. Για παράδειγμα, μπορεί να υπάρχει ένα εργαλείο το οποίο έχει τη δυνατότητα να κάνει μετρήσεις για πολλές παραμέτρους που σχετίζονται με την απόδοση. Το πρόβλημα είναι ότι όλη αυτή η πληροφορία, για να έχει κάποια πρακτική αξία, θα πρέπει να επεξεργαστεί πρώτα. Η απόφαση για το ποιο μέρος της πληροφορίας θα πρέπει να χρησιμοποιηθεί και ποιες παράμετροι απόδοσης είναι αυτές στις οποίες πρέπει να δοθεί έμφαση, είναι απαραίτητο να ληφθεί από κάποιον ο οποίος είναι εξειδικευμένος στον τομέα αυτό.

Δυστυχώς, σε πολλές εταιρίες δεν έχει γίνει ακόμα κατανοητή η σημασία ύπαρξης εξειδικευμένου προσωπικού που θα μπορούσε, όχι μόνο να συλλέξει δεδομένα για την απόδοση αλλά και να δώσει ερμηνεία σε αυτά. Η αντίληψη αυτή έχει να κάνει σε μεγάλο βαθμό με την έλλειψη στρατηγικού σχεδιασμού για την εξέλιξη του δικτύου. Αν είχε γίνει συνείδηση ότι η προσφορά μίας νέας υπηρεσίας μέσω του δικτύου ή η επέκταση του δικτύου πρέπει να γίνονται με εκ των προτέρων σχεδιασμό, τότε θα ήταν σαφές ότι η μέτρηση και ερμηνεία των αποδόσεων του συστήματος είναι απαραίτητες, αφού αποτελούν τη βάση για να πραγματοποιηθεί κάτι τέτοιο.

Επίσης, μία ακόμα σημαντική παρατήρηση είναι ότι όταν δεν υπάρχει εξειδικευμένο προσωπικό που να ασχολείται με την ερμηνεία των στατιστικών απόδοσης είναι πολύ πιθανό να χαθούν στοιχεία που σχετίζονται με άλλες λειτουργίες διαχείρισης. Η πιο συνηθισμένη περίπτωση τέτοιας λειτουργίας είναι η ασφάλεια.

Για να αναδείξουμε τη σχέση απόδοσης–ασφάλειας θα αναφερθούμε σε δύο παραδείγματα.

- Έστω ότι στο δίκτυο μίας επιχείρησης παρατηρείται σημαντική αύξηση των πακέτων που χάνονται κατά τη διάρκεια μίας επικοινωνίας. Το γεγονός αυτό καταγράφεται από το σύστημα μέτρησης αποδόσεων, αλλά δε δίνεται ιδιαίτερη σημασία, καθώς τα πακέτα αφορούν σε μη διαλογικές εφαρμογές και έτσι δεν προκαλείται ιδιαίτερο πρόβλημα κατά τη χρήση. Φαινομενικά, το

σοβαρότερο πρόβλημα στην περίπτωση αυτή είναι ότι θα πρέπει να επανεκπέμπονται τα πακέτα, οπότε προκαλείται ένας επιπλέον φόρτος στο δίκτυο, ο οποίος όμως δεν είναι μεγάλος. Μία πιο προσεκτική ματιά όμως στα δεδομένα απόδοσης, θα μπορούσε να προκαλέσει υποψίες για παραβίαση της ασφάλειας στο δίκτυο. Ποιος εξασφαλίζει ότι ο λόγος που χάνονται τα πακέτα δεν είναι ότι κάποιος έχει καταφέρει να παρεμβληθεί στο δίκτυο και τα υποκλέπτει; Προφανώς για να δοθεί σίγουρη απάντηση, το θέμα πρέπει να διερευνηθεί. Όμως για να δοθεί εντολή διερεύνησης, κάποιος πρέπει πρώτα να επισημάνει ότι έχουμε πιθανή παραβίαση της ασφάλειας.

- Το δεύτερο παράδειγμα που θα δώσουμε είναι ανάλογο με το πρώτο με τη διαφορά ότι δεν παρατηρείται πλέον αύξηση των χαμένων πακέτων, αλλά αύξηση του μέσου χρόνου που κάνει ένα πακέτο να φτάσει στον προορισμό του. Και σε αυτή την περίπτωση μία πιθανή αιτία είναι ότι κάποιος έχει καταφέρει να παρεμβληθεί μεταξύ των εφαρμογών που επικοινωνούν. Μάλιστα, η αύξηση του χρόνου για να φτάσει το μήνυμα στον προορισμό του μπορεί να δηλώνει ότι ο επιτιθέμενος δεν παρακολουθεί παθητικά τα πακέτα, αλλά πραγματοποιεί και αλλαγές στα δεδομένα που περιέχουν.

Ανάλυση κόστους

Έχουμε ήδη αναφέρει τη σημασία που έχει η ανάλυση κόστους για μία σύγχρονη επιχείρηση. Η ανάλυση κόστους στη συγκεκριμένη εταιρία θα μπορούσε, σε πρώτη φάση τουλάχιστον, να αναχθεί στο ερώτημα «πόσο φόρτο προσθέτει στο δίκτυο η κάθε εφαρμογή;».

Αυτό που μας προξένησε ιδιαίτερη εντύπωση ήταν ότι, ενώ κρατούνταν στατιστικά για την SNA και την IP κίνηση, δεν υπήρχε καταγραφή του ποσοστού του φόρτου που προκαλούνταν ανά ζεύξη, αλλά και συνολικά στο δίκτυο, από τη χρήση των εφαρμογών διαχείρισης του δικτύου. Όπως έχουμε πει οι εφαρμογές διαχείρισης στο συγκεκριμένο δίκτυο χρησιμοποιούν το πρωτόκολλο SNMP.

Επισημαίνεται ότι το πρωτόκολλο διαχείρισης SNMP προκαλεί μεγάλο φόρτο στα δίκτυα. Μάλιστα, έχουν γίνει μελέτες που δείχνουν ότι περίπου το 30% της κίνησης των δικτύων στα οποία χρησιμοποιείται SNMP, προκαλείται από το συγκεκριμένο πρωτόκολλο διαχείρισης.

Η παραπάνω περίπτωση φανερώνει κάτι που συμβαίνει σε δίκτυα πολλών εταιριών: δεν υπάρχει γνώση του ποσοστού της κίνησης που προκαλείται από κάθε εφαρμογή. Η γνώση αυτού του μεγέθους μπορεί να χρησιμοποιηθεί και σαν εργαλείο για τον εντοπισμό «παράξενης» συμπεριφοράς στο δίκτυο, όταν κίνηση ενός είδους παρουσιάσει ανεξήγητη αύξηση ή μείωση [5].

2.3 Ανάλυση της διαχείρισης δικτύων σε πραγματικό περιβάλλον

Για μια πρώτη προσέγγιση του περιβάλλοντος διαχείρισης και της MIB χρησιμοποιήσαμε το δίκτυο του Πανεπιστημίου Θεσσαλίας και ένα freeware. Ένας από τους στόχους της παρούσας εργασίας ήταν να αποκτήσουμε πρόσβαση στις MIB

των δρομολογητών ενός πραγματικού δικτύου, ώστε να μελετήσουμε τα διαχειριζόμενα αντικείμενα. Όπως προαναφέραμε, για το σκοπό αυτό αποφασίστηκε να χρησιμοποιηθούν οι δρομολογητές του πανεπιστημίου Θεσσαλίας. Τα βήματα που ακολουθήθηκαν ήταν:

1) Απόκτηση community string για πρόσβαση στις MIB των δρομολογητών.

2) Εύρεση λογισμικού που θα μπορούσε να χρησιμοποιηθεί για να έχουμε πρόσβαση στους δρομολογητές.

3) Εγκατάσταση λογισμικού.

4) Χρησιμοποίηση λογισμικού.

Στη συνέχεια θα αναφερθούμε ξεχωριστά σε καθένα από τα παραπάνω βήματα.

Απόκτηση community string για πρόσβαση στις MIB των δρομολογητών

Η διαχείριση του δικτύου του πανεπιστημίου Θεσσαλίας στηρίζεται στο πρωτόκολλο SNMP. Όπως αναλύσαμε στο πρώτο κεφάλαιο, για να μπορέσει ο διαχειριστής να έχει πρόσβαση στους πράκτορες του SNMP, πρέπει να ανήκει στην ίδια κοινότητα με αυτούς. Αυτό δηλώνεται με ένα κωδικό, το community string.

Για να αποκτήσουμε το community string που θα μας έδινε τη δυνατότητα να επικοινωνούμε με τους πράκτορες, απευθυνθήκαμε στο κέντρο δικτύου του πανεπιστημίου Θεσσαλίας. Αφού εξηγήσαμε τι ακριβώς θέλουμε, μας δόθηκε τελικά το community string για πρόσβαση σε δύο δρομολογητές. Τα δικαιώματα της κοινότητας στην οποία ανήκαμε ήταν «*μόνο για ανάγνωση*».

Εύρεση λογισμικού για πρόσβαση στους δρομολογητές

Η απόκτηση ενός community sting δε σε καθιστά αυτόματα ικανό να επικοινωνήσεις με τους πράκτορες ενός δικτύου που ανήκουν στη συγκεκριμένη κοινότητα. Πρέπει να υπάρχει και κάποιο λογισμικό, το οποίο να παίζει το ρόλο του διαχειριστή. Το λογισμικό αυτό αποφασίστηκε να εγκατασταθεί στα εργαστήρια του Τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Δικτύων.

Αρχικά, ρωτήσαμε στο κέντρο διαχείρισης δικτύου του Πανεπιστημίου Θεσσαλίας, αν μπορούσαν να μας προτείνουν κάποιο δωρεάν λογισμικό που θα μπορούσε να χρησιμοποιηθεί. Δυστυχώς, η απάντηση ήταν ότι δεν είχαν κάτι υπόψη τους, αφού το λογισμικό που χρησιμοποιούσαν οι ίδιοι δε διαθέτονταν δωρεάν.

Μετά από αυτό αρχίσαμε μόνοι μας την αναζήτηση για το κατάλληλο λογισμικό. Ο τρόπος αναζήτησης ήταν, ποιος άλλος, η χρήση μηχανών αναζήτησης του διαδικτύου. Οι λέξεις κλειδιά που χρησιμοποιήσαμε: SNMP Network Management Tools. Ο όγκος της πληροφορίας που επιστρέφονταν σε αυτές τις αναζητήσεις ήταν τεράστιος και η αξιολόγησή της δύσκολη και ιδιαίτερα χρονοβόρα. Ανάμεσα στα επιστρεφόμενα αποτελέσματα υπήρχε μία ιστοσελίδα [Δ], η οποία προτάθηκε ως καλό σημείο αναφοράς και από τον επιβλέποντα της παρούσας διπλωματικής εργασίας, που περιείχε λεπτομέρειες σε σχέση με τη διαχείριση δικτύων με βάση το

πρωτόκολλο SNMP. Επιπλέον, περιείχε ένα μακροσκελή κατάλογο από εργαλεία που μπορούσαν να χρησιμοποιηθούν για το σκοπό αυτό. Ο κατάλογος αυτός απετέλεσε τη βάση για την περαιτέρω έρευνά μας. Το λογισμικό που ψάχναμε έπρεπε να πληρεί τις παρακάτω προϋποθέσεις:

- Έπρεπε να καλύπτει τη λειτουργία get του SNMP. Δεν πρέπει να ξεχνάμε ότι είχαμε δικαιώματα μόνο για ανάγνωση στις MIB των δρομολογητών, οπότε ούτως ή άλλως δε θα μπορούσαμε να χρησιμοποιήσουμε λειτουργία που απαιτούσε και δικαιώματα εγγραφής.

- Έπρεπε να διατίθεται δωρεάν.

- Οι απαιτήσεις του σε υπολογιστικούς πόρους (μνήμη και ταχύτητα επεξεργαστή) δεν έπρεπε να είναι μεγάλες, αφού θα έτρεχε πάνω σε ένα απλό τερματικό που βρίσκονταν στα εργαστήρια του ΤΜΗΥΔ.

- Η εγκατάσταση και η χρήση του έπρεπε να είναι όσο πιο απλά γίνεται, καθώς δεν είχαμε μεγάλα χρονικά περιθώρια.

Από τα λογισμικά που ελέγξαμε βρήκαμε ότι αυτό που ήταν πιο κοντά στις προϋποθέσεις που θέσαμε ήταν το getif. Πρόκειται για ένα λογισμικό που μπορεί να πραγματοποιήσει τη λειτουργία get, δεδομένου ότι ο χρήστης διαθέτει το απαραίτητο community string, διατίθεται δωρεάν, δεν έχει ιδιαίτερες απαιτήσεις σε υπολογιστικούς πόρους και οι οδηγίες εγκατάστασης και χρήσης του δε μας φάνηκαν πολύπλοκες.

Εγκατάσταση λογισμικού

Για να εγκαταστήσουμε το λογισμικό σε ένα τερματικό του ΤΜΗΥΔ, έπρεπε να διαθέτουμε δικαιώματα administrator στο τερματικό αυτό. Έτσι, απευθυνθήκαμε στην τεχνική υποστήριξη του τμήματος. Μετά από λίγες μέρες μας δόθηκαν δικαιώματα administrator σε ένα υπολογιστή.

Τα βήματα που ακολουθήσαμε για να εγκαταστήσουμε το λογισμικό περιγράφονται στη συνέχεια:

- Αρχικά κατεβάσαμε την έκδοση 2.2 του λογισμικού.

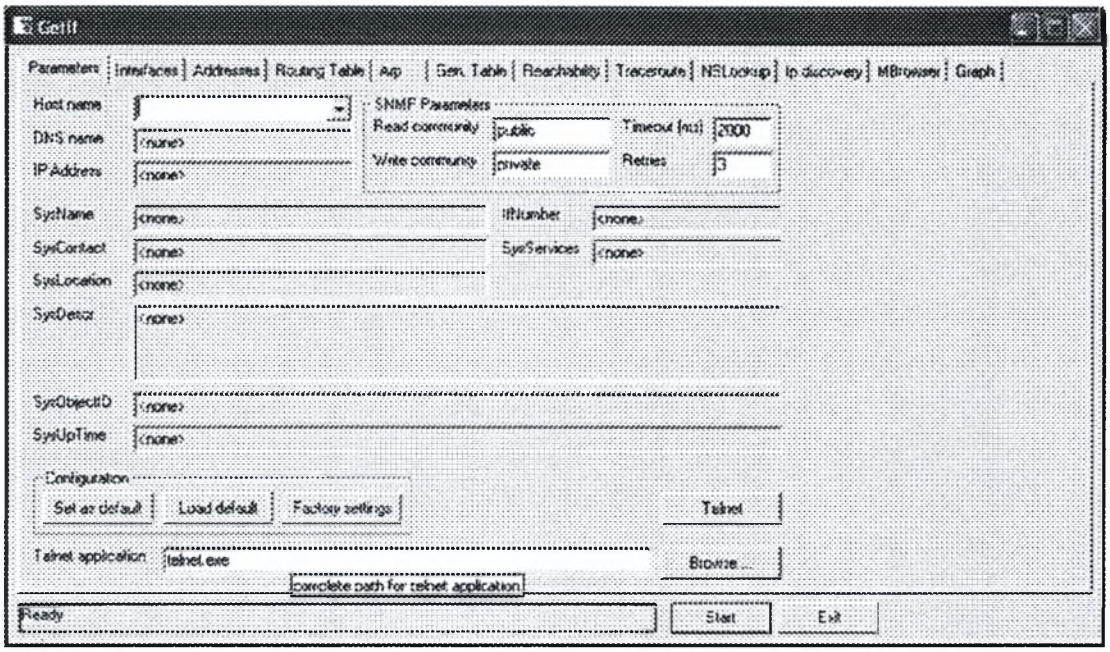
- Στη συνέχεια, αποσυμπιέσαμε το αρχείο που κατεβάσαμε στο σκληρό δίσκο του υπολογιστή και διπλοκλικάραμε στο αρχείο set up. Η πρώτη φάση της εγκατάστασης είχε πλέον ολοκληρωθεί.

- Για να λειτουργήσει το συγκεκριμένο λογισμικό, έπρεπε στον ίδιο κατάλογο που ήταν εγκατεστημένο να υπάρχουν και αρχεία με τις MIB που θα χρησιμοποιούσαμε. Εμείς κατεβάσαμε μία συλλογή από MIB που προτείνεται στην ιστοσελίδα όπου βρήκαμε πληροφορίες για το getif. Εντός της συλλογής βρίσκονταν και η MIB 2 η οποία μας ενδιέφερε.

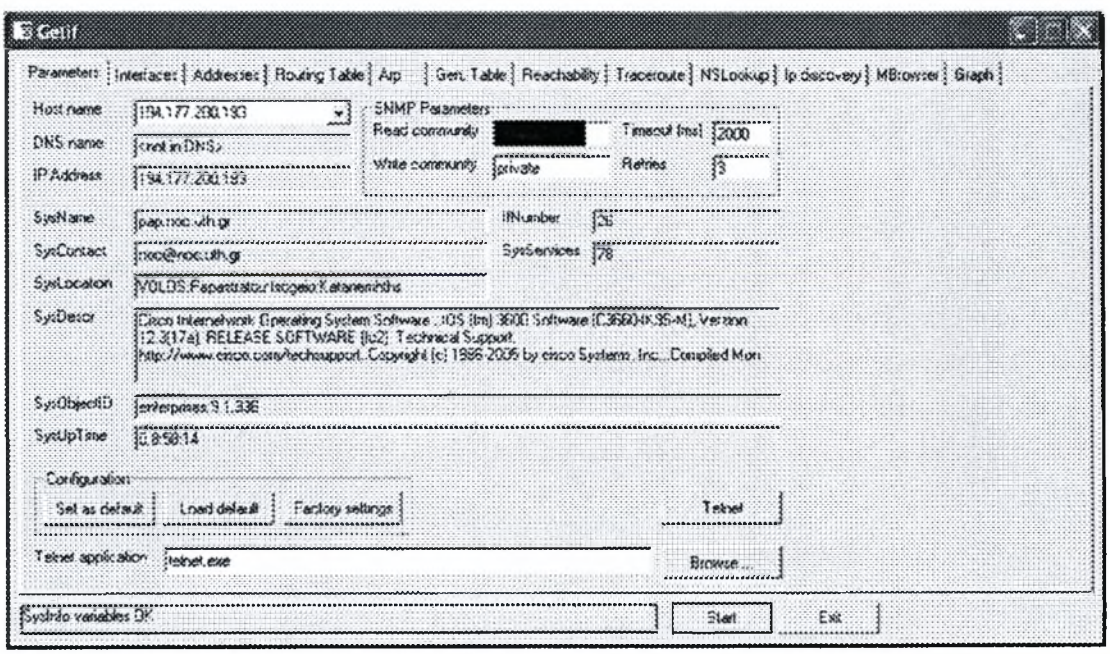
Μετά από αυτή τη διαδικασία, το λογισμικό λειτουργούσε κανονικά και εμείς ήμασταν σε θέση να δούμε το περιεχόμενο των MIB που μας ενδιέφεραν. Στη συνέχεια, παράλληλα με μία μικρή επίδειξη του λογισμικού, θα παρουσιάσουμε μερικά δεδομένα από τις MIB των δρομολογητών του Πανεπιστημίου Θεσσαλίας.

Χρησιμοποίηση λογισμικού

Επιλέγουμε start → all programs → get if 2.2 και εμφανίζεται το παράθυρο διαλόγου του σχήματος 2.5.

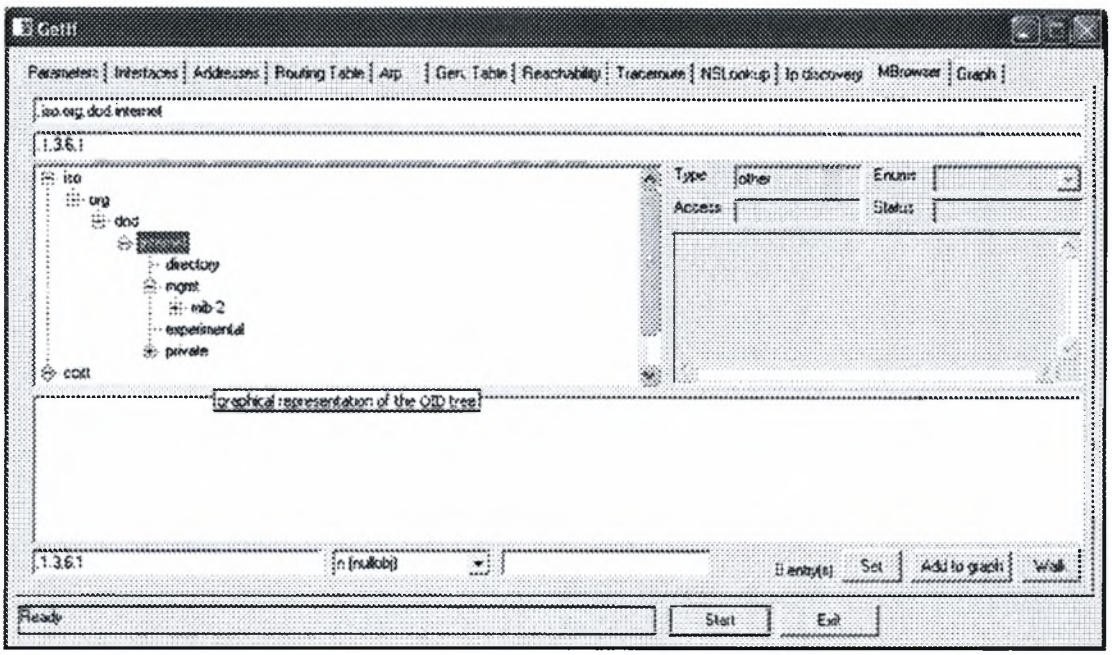


Σχήμα 2.5: Αρχικό παράθυρο διαλόγου

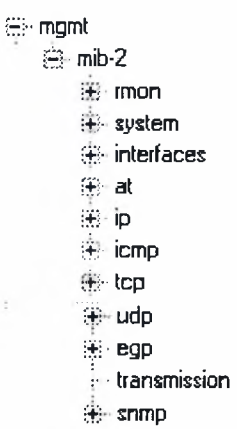


Σχήμα 2.6: Παράθυρο διαλόγου μετά την εισαγωγή του community string

Συμπληρώνουμε στο πεδίο Host name την IP διεύθυνση του δρομολογητή που μας ενδιαφέρει, στο πεδίο Read community το community string και επιλέγουμε start. Το αποτέλεσμα θα είναι όμοιο με αυτό που παρουσιάζεται στο σχήμα 2.6. Παρατηρούμε ότι πάνω στο παράθυρο έχουν συμπληρωθεί πεδία τα οποία αντιστοιχούν σε αντικείμενα της MIB 2. Έτσι μπορούμε να δούμε, για παράδειγμα, ότι ο συγκεκριμένος δρομολογητής έχει 26 interfaces (IfNumber) και ότι βρίσκεται τοποθετημένος στο ισόγειο του κτηρίου Παπαστράτου στο Βόλο (SysLocation). Επισημαίνουμε ότι έχουμε αλλοιώσει το πεδίο που περιέχει τον κωδικό πρόσβασης.



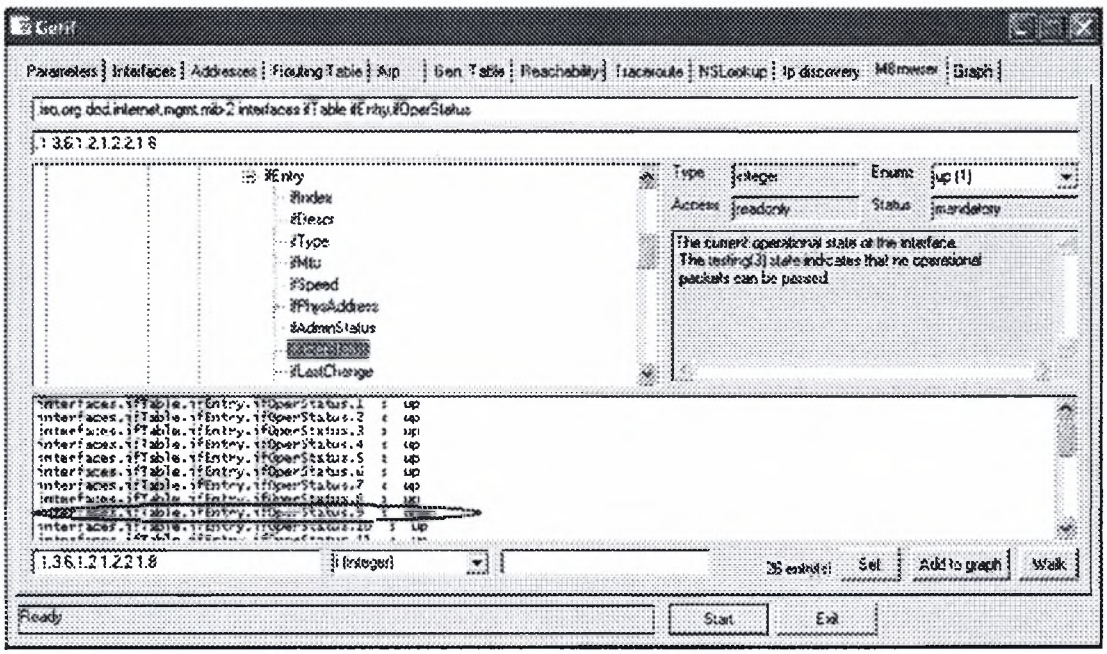
Σχήμα 2.7: *Εμφάνιση Management Information Tree*



Σχήμα 2.8: *Οι ομάδες της MIB 2*

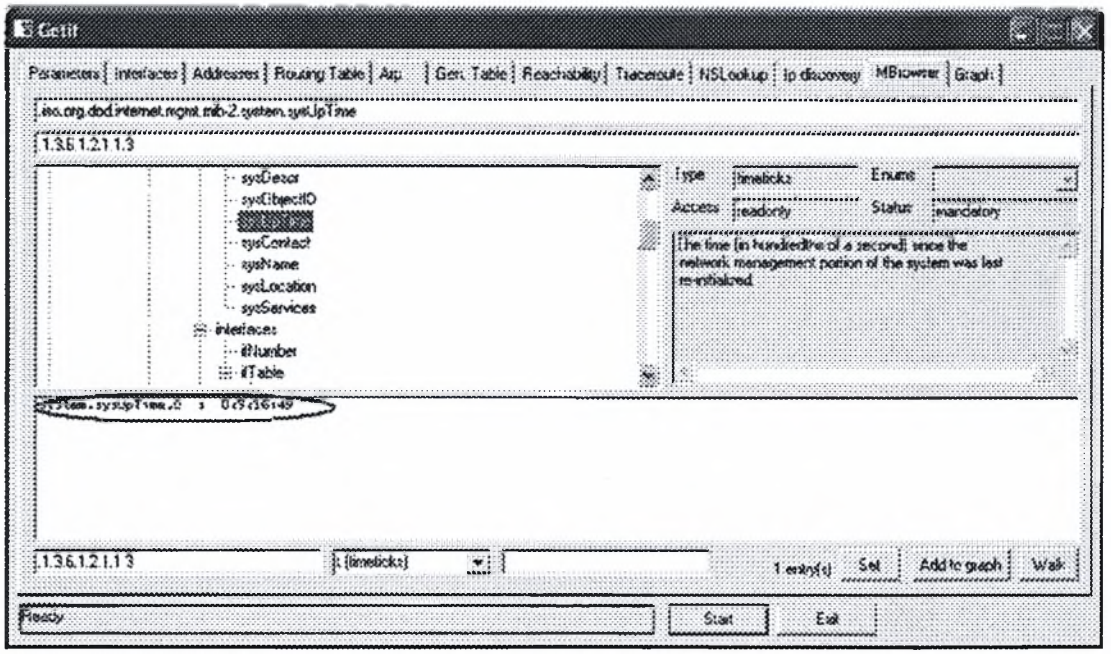
Αν κοιτάξουμε στο πάνω μέρος του παραθύρου, θα δούμε ότι το συγκεκριμένο λογισμικό μπορεί να χρησιμοποιηθεί για αρκετές λειτουργίες. Εμείς ενδιαφερόμαστε για τη MIB του πράκτορα του δρομολογητή, οπότε επιλέγουμε MBrowser. Μετά από αυτή την επιλογή στο παράθυρο εμφανίζεται το Management Information Tree ,όπως φαίνεται στο σχήμα 2.7.

Αν επιλέξουμε MIB 2, θα δούμε όλες της ομάδες αντικειμένων της MIB 2, όπως φαίνεται στο σχήμα 2.8.



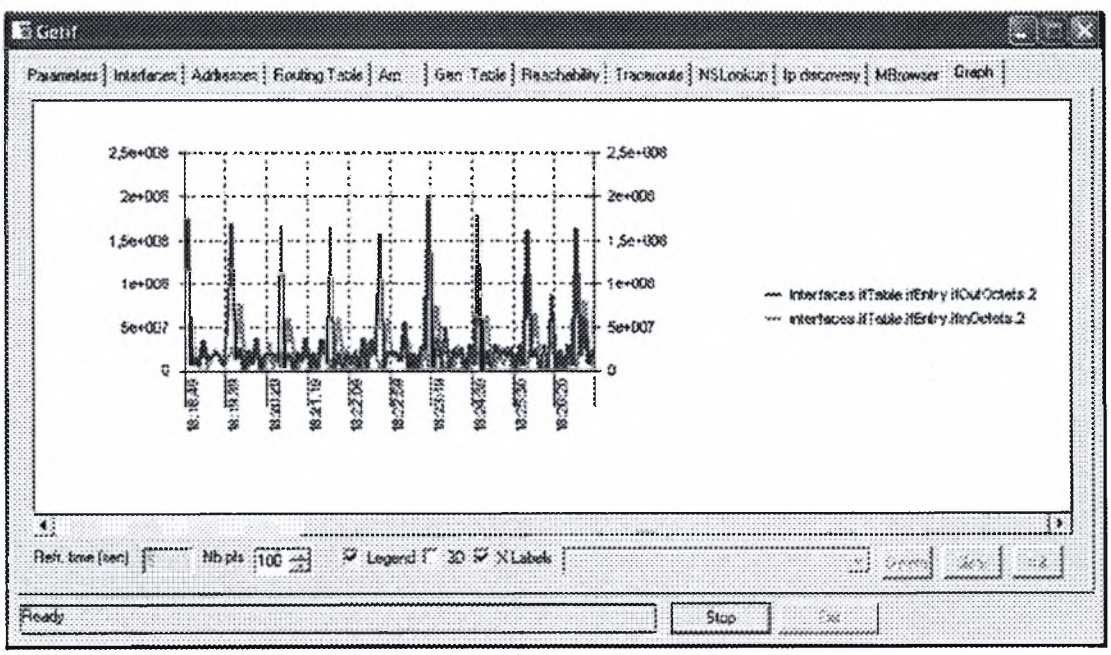
Σχήμα 2.9: Status των interface

Στη συνέχεια, αν κινηθούμε προς το κάτω μέρος του δέντρου και επιλέξουμε ifOperStatus θα δούμε το status όλων των interface του δρομολογητή. Στο σχήμα 2.9 βλέπουμε ότι ένα interface είναι down. Αυτό δε δηλώνει αναγκαστικά βλάβη. Μπορεί απλώς το συγκεκριμένο interface να μη χρησιμοποιείται.



Σχήμα 2.10: *χρόνος τελευταίας επανεκκίνησης*

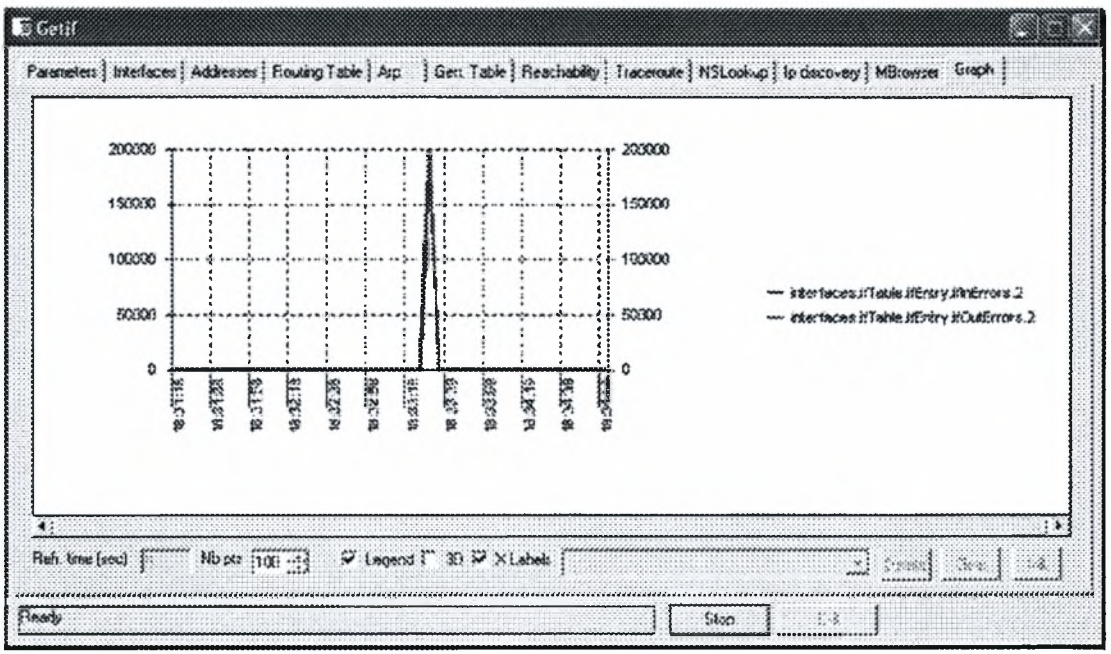
Όμοια μπορούμε να προσπελάσουμε το αντικείμενο sysUpTime, το οποίο δηλώνει πότε έγινε η τελευταία επανεκκίνηση του δρομολογητή. Στο σχήμα 2.10 βλέπουμε ότι ο δρομολογητής που προσπελάσαμε λειτουργεί συνεχώς για 9 ώρες, 16 λεπτά και 49 δευτερόλεπτα.



Σχήμα 2.11: *Εισερχόμενος και εξερχόμενος αριθμός Octets*

Το getif παρέχει και τη δυνατότητα της γραφικής απεικόνισης των τιμών των αντικειμένων της MIB, συναρτήσει του χρόνου. Αυτό γίνεται με την επιλογή graph.

Στο σχήμα 2.11 ο αριθμός των εισερχόμενων (ifInOctets) και των εξερχόμενων (ifOutOctets) Octets για ένα interface σημειώνεται με πράσινο και με κόκκινο χρώμα αντίστοιχα. Τα εξερχόμενα Octets είναι σαφώς περισσότερα.

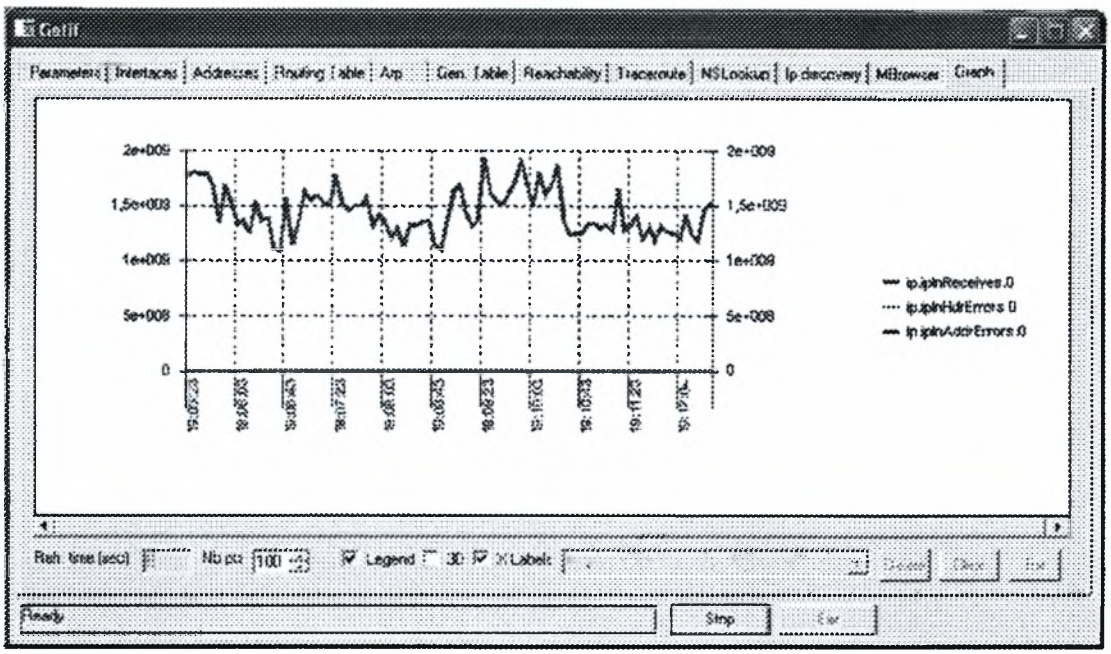


Σχήμα 2.12: *Αριθμός λάθος εισερχόμενων και εξερχόμενων πακέτων*

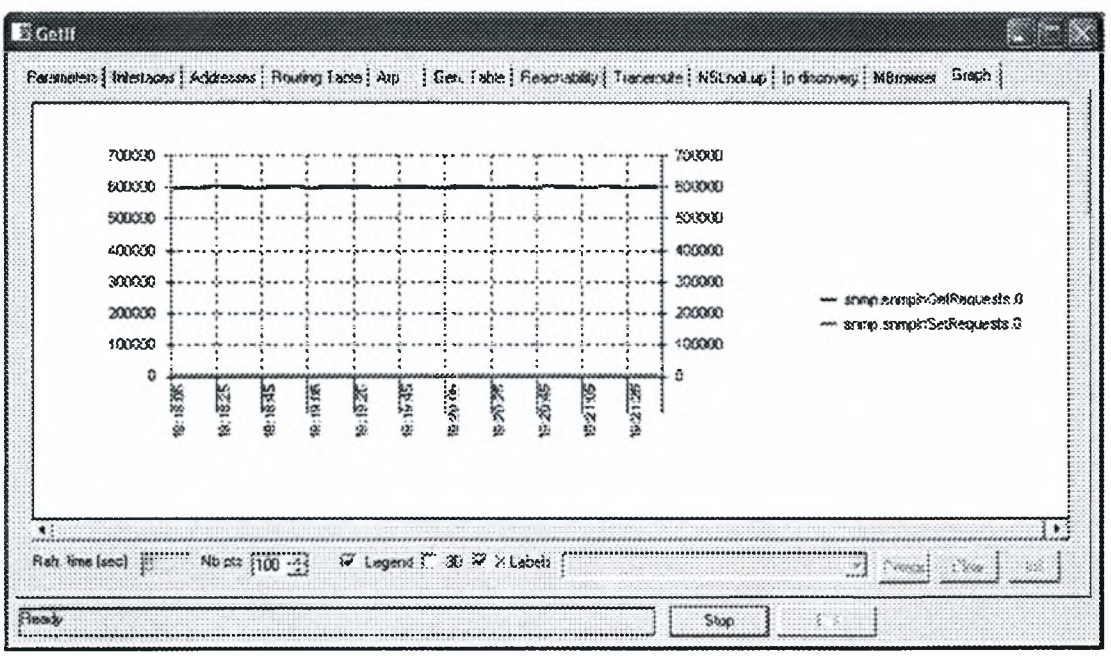
Στο σχήμα 2.12 η γραφική παράσταση αναφέρεται στον αριθμό των λάθος εισερχόμενων (ifInErrors) και εξερχόμενων πακέτων (ifOutErrors) σε ένα interface. Παρατηρούμε πως σε κάποια στιγμή έρχονται πολλά λάθος πακέτα στο interface (η αντίστοιχη καμπύλη είναι η κόκκινη). Κατά τα άλλα η λειτουργία είναι απολύτως ομαλή.

Στη γραφική παράσταση του σχήματος 2.13 έχουμε τον αριθμό των συνολικών εισερχόμενων πακέτων IP (ipInReceives), τον αριθμό των εισερχόμενων πακέτων IP που είχαν λάθος στην IP διεύθυνση (ipInAddrErrors) και τον αριθμό των εισερχόμενων πακέτων IP που είχαν λάθος σε άλλες κεφαλίδες του πακέτου (ipInHdrErrors). Η καμπύλη για κάθε μέγεθος είναι αντίστοιχα η κόκκινη, η πράσινη και η μπλε, ενώ όλα τα στοιχεία αφορούν στο ίδιο interface. Παρατηρούμε ότι δεν υπάρχουν λάθη στα εισερχόμενα πακέτα IP.

Τέλος στο σχήμα 2.14 παρουσιάζεται ο αριθμός των get (inGetRequests) και set PDUs (inSetRequests) που λαμβάνει ο πράκτορας του δρομολογητή. Όπως παρατηρούμε, δε λήφθηκαν set PDUs κατά το χρονικό διάστημα στο οποίο αναφέρεται η συγκεκριμένη γραφική παράσταση (πράσινη καμπύλη).



***Σχήμα 2.13**: Εισερχόμενη IP κίνηση και λάθη*



***Σχήμα 2.14:** Αριθμός ληφθέντων get και set PDUs*

Κλείνοντας την ενότητα αυτή, θεωρούμε σκόπιμο να κάνουμε μία σύντομη αναφορά σε κάποια πράγματα που μάθαμε από τη διαδικασία εύρεσης, εγκατάστασης και χρήσης του λογισμικού.

Το πρώτο σημείο στο οποίο θα αναφερθούμε είναι η πληθώρα των δωρεάν διατιθέμενων λογισμικών για την προσπέλαση των MIB. Ο αριθμός αυτός είναι τόσο

μεγάλος που, στην αρχή τουλάχιστον, μπορεί να δημιουργήσει σύγχυση. Η διαδικασία της εύρεσης μας ανάγκασε να δούμε αρκετά λογισμικά και να μελετήσουμε τις δυνατότητες που παρέχουν.

Κάποια από τα λογισμικά αυτά, παρέχουν εκτός από τη δυνατότητα απλής παρακολούθησης της MIB και δυνατότητα υλοποίησης της λειτουργίας SET, οπότε μπορούν να χρησιμοποιηθούν και ως συστήματα διαχείρισης δικτύων. Φυσικά, πρέπει να επισημάνουμε ότι η χρησιμοποίηση freeware δεν αποτελεί την ενδεδειγμένη λύση για τη σωστή διαχείριση ενός δικτύου.

Επίσης, κάποια από τα λογισμικά αυτά έχουν αυξημένες απαιτήσεις σε υπολογιστικούς πόρους (μνήμη και επεξεργαστική ισχύ), ενώ η ευχρηστία παρουσιάζει τεράστια διακύμανση από λογισμικό σε λογισμικό. Συμπερασματικά μπορούμε να πούμε ότι πριν ακόμα αρχίσουμε να ψάχνουμε για ένα λογισμικό πρέπει να καθορίσουμε ακριβώς ποιες είναι οι ανάγκες μας και τι επεξεργαστικούς πόρους διαθέτουμε.

Η μελέτη της διαδικασίας εγκατάστασης των διάφορων λογισμικών ανέδειξε την ελλιπή τεκμηρίωση των freeware, κάτι που άλλωστε ήταν αναμενόμενο. Όπως αναφέραμε και παραπάνω, εμείς τελικά επιλέξαμε ένα λογισμικό το οποίο είχε αρκετά σαφείς οδηγίες για τη χρήση του και που η εγκατάστασή του ήταν εύκολη.

Η χρήση του λογισμικού μας έδωσε τη δυνατότητα να δούμε στην πράξη κάποια από τα πράγματα τα οποία μελετήσαμε θεωρητικά στο πρώτο κεφάλαιο. Για παράδειγμα είδαμε πώς υλοποιείται η έννοια της κοινότητας στο SNMP. Όπως είπαμε, εμείς ανήκαμε σε μία κοινότητα που είχε δυνατότητα μόνο για ανάγνωση στη MIB.

Επιπλέον, το λογισμικό έγινε το μέσον για να μελετήσουμε τα διαχειριζόμενα αντικείμενα σε πραγματικούς δρομολογητές. Η όλη διαδικασία αποτέλεσε ουσιαστικά μία εισαγωγή στη διαχείριση σε πραγματικό περιβάλλον, καθώς μας έδωσε τη δυνατότητα να δούμε τις τιμές που παίρνουν τα αντικείμενα αυτά.

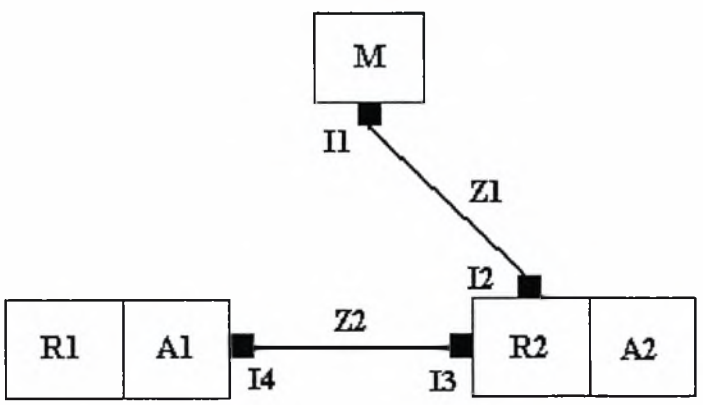
Η μελέτη των διάφορων διαχειριζόμενων αντικειμένων, συσχετίζοντας το ένα με το άλλο, μπορεί να οδηγήσει σε σημαντικά συμπεράσματα. Ως παράδειγμα αναφέρουμε το σχήμα 2.11, όπου η ταυτόχρονη μελέτη εισερχόμενης και εξερχόμενης κίνησης αναδεικνύει ότι το συγκεκριμένο interface λειτουργεί κυρίως ως «έξοδος» δεδομένων. Το σχήμα 2.11, δείχνει επίσης πόσο χρήσιμο εργαλείο μπορούν να γίνουν οι γραφικές παραστάσεις κατά τη μελέτη ενός δικτύου. Το γεγονός αυτό γίνεται αφορμή να επισημάνουμε ότι τα εργαλεία διαχείρισης είναι καλό να υποστηρίζουν την αυτόματη δημιουργία τέτοιων γραφικών παραστάσεων.

Κεφάλαιο 3

ΑΝΑΛΥΣΗ ΔΙΚΤΥΟΥ ΒΑΣΗ ΤΩΝ ΕΡΓΑΛΕΙΩΝ ΚΑΙ ΛΕΙΤΟΥΡΓΙΩΝ ΔΙΑΧΕΙΡΙΣΗΣ

Στο προηγούμενο κεφάλαιο αναφερθήκαμε στο δίκτυο μίας μεγάλης επιχείρησης την οποία επισκεφτήκαμε και στον τρόπο με τον οποίο πραγματοποιούνταν η διαχείριση στο δίκτυο αυτό. Μία από της αδυναμίες που είχαμε εντοπίσει ήταν ότι ενώ κρατούνταν στατιστικά για τη διαθεσιμότητα του δικτύου, δεν κρατούνταν στατιστικά για το ποιος ήταν ο υπεύθυνος κάθε φορά που σημειώνονταν βλάβη: ο πάροχος των μισθωμένων ζεύξεων ή ο ιδιόκτητος εξοπλισμός της εταιρίας.

Στο παρόν κεφάλαιο θα επιχειρήσουμε να «προβάλουμε» όλες τις λειτουργίες διαχείρισης από το πραγματικό δίκτυο της επιχείρησης σε ένα μικρό δίκτυο που αποτελείται από δύο δρομολογητές και ένα σύστημα διαχείρισης. Ο λόγος είναι ότι μερικές φορές το μέγεθος ενός πραγματικού δικτύου μας αποπροσανατολίζει και μας κάνει να χάνουμε την ουσία. Σκοπός μας είναι η διευκόλυνση της συστηματικής μελέτης των λειτουργιών διαχείρισης, με έμφαση στον εντοπισμό προβλημάτων που σχετίζονται με τη διαθεσιμότητα. Στο σχήμα 3.1.α παρουσιάζεται η τοπολογία του δικτύου το οποίο θα χρησιμοποιήσουμε για την ανάλυση που θα κάνουμε.



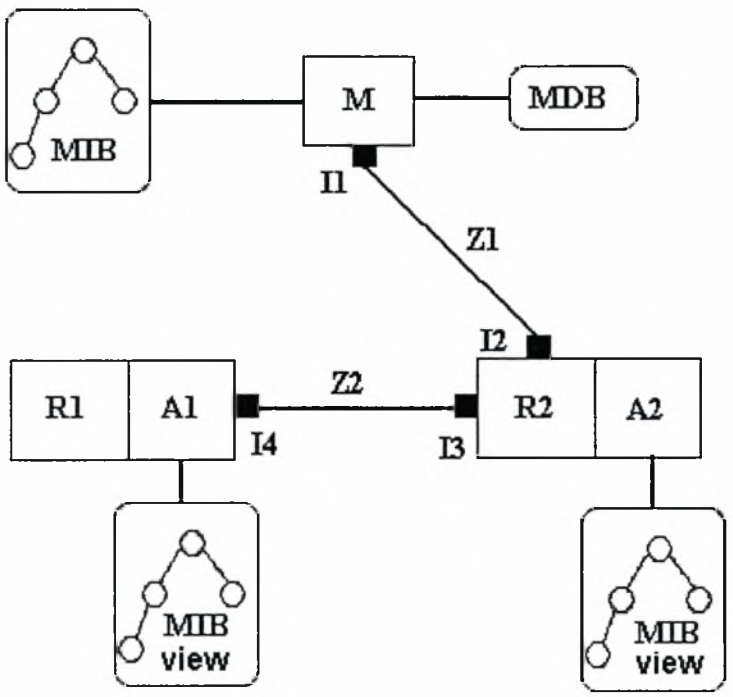
3.1.α: *Τοπολογία δικτύου*

Βλέπουμε ότι το δίκτυο αποτελείται από τους δρομολογητές R1 και R2, στους οποίους βρίσκονται οι πράκτορες A1 και A2 αντίστοιχα, από τις ζεύξεις Z1 και Z2 και από ένα κόμβο που αντιστοιχεί στο σύστημα διαχείρισης στο οποίο βρίσκεται ο διαχειριστής M. Επιπλέον, οι κάρτες δικτύου αντιστοιχούν στα μαύρα τετράγωνα και συμβολίζονται με το γράμμα I. Στο σχήμα έχουμε τέσσερις κάρτες δικτύου.

Αυτό που αξίζει να προσέξουμε είναι ότι δεν υπάρχει ξεχωριστό δίκτυο για την επικοινωνία του διαχειριστή με τους πράκτορες. Σε κάθε περίπτωση χρησιμοποιούνται οι βασικές ζεύξεις του δικτύου. Επίσης, δεν υπάρχουν

εναλλακτικές διαδρομές για την αποστολή μηνυμάτων, σε περίπτωση που μία ζεύξη, δρομολογητής ή κάρτα δικτύου παρουσιάσει κάποιο πρόβλημα. Τα παραπάνω βρίσκονται σε πλήρη αντιστοιχία με το δίκτυο περιοχής και το δίκτυο καταστημάτων της πραγματικής επιχείρησης (όπως είπαμε το δίκτυο κορμού σχημάτιζε πλέγμα).

Επίσης, στην ανάλυση που θα κάνουμε θα υποθέσουμε ότι το χρησιμοποιούμενο πρωτόκολλο διαχείρισης είναι το SNMP, όπως συνέβαινε και στην επιχείρηση. Στο σχήμα 3.1.β παρουσιάζονται οι απόψεις της MIB που έχουν οι πράκτορες και η MIB του διαχειριστή. Η MDB αντιστοιχεί στη βάση όπου ο διαχειριστής αποθηκεύει τα αποτελέσματα των αναλύσεων που κάνει.



Σχήμα 3.1.β: *MIB και απόψεις MIB*

3.1 Ανάλυση διαμόρφωσης

Στη διαχείριση διαμόρφωσης μπορούμε να περιλάβουμε τις εξής ενέργειες:

- Συμπλήρωση των τιμών για κάποια από τα αντικείμενα του System Group. Στο System Group υπάρχουν κάποια αντικείμενα οι τιμές των οποίων πρέπει να συμπληρωθούν από το διαχειριστή του δικτύου. Στην περίπτωση που εξετάζουμε πρέπει για κάθε δρομολογητή να συμπληρωθεί: i) η τιμή του sysName, η οποία είναι ένα κατανοητό και ευκολομνημόνευτο όνομα. ii) η τιμή του sysLocation. Το αντικείμενο αυτό δηλώνει πού βρίσκεται ο δρομολογητής (κτήριο, όροφο, αίθουσα). iii) η τιμή του sysContact, η οποία δηλώνει ποιος είναι ο υπεύθυνος για το συγκεκριμένο δρομολογητή. Το αντικείμενο sysContact μπορεί να περιέχει το όνομα ενός τεχνικού, το όνομα

μίας εταιρίας που έχει αναλάβει την τεχνική υποστήριξη, μία διεύθυνση ηλεκτρονικού ταχυδρομείου, ένα τηλέφωνο ή οποιοδήποτε συνδυασμό των παραπάνω. Πρέπει να σημειωθεί ότι τα αντικείμενα του System Group αναφέρονται στους δρομολογητές και ότι δεν υπάρχουν ανάλογα αντικείμενα για κάθε κάρτα δικτύου (interface) ξεχωριστά. Η διαδικασία συμπλήρωσης των τιμών των αντικειμένων του System Group είναι μία ενέργεια που αναφέρεται στο έβδομο επίπεδο του OSI, καθώς όλες οι τιμές που προαναφέραμε είναι πλήρως κατανοητές από τον άνθρωπο και υπάρχουν απλώς για διευκόλυνσή του.

- Συμπλήρωση των πινάκων που περιέχουν πληροφορίες σχετικές με τις διευθύνσεις και τη δρομολόγηση. Υπάρχουν τρεις πίνακες που περιέχουν αυτές τις πληροφορίες. Ο ipAddrTable περιέχει πληροφορίες σχετικές με τις ip διευθύνσεις που αντιστοιχούν στο συγκεκριμένο δρομολογητή, καθώς είναι δυνατό στον ίδιο δρομολογητή να αντιστοιχούν περισσότερες από μία ip διευθύνσεις. Στο συγκεκριμένο πίνακα περιέχεται και η αντιστοίχιση κάθε ip διεύθυνσης με ένα interface του δρομολογητή. Στον πίνακα ipRouteTable περιέχεται η διαδρομή για κάθε προορισμό, ενώ μπορεί να υπάρξουν περισσότερες από μία διαδρομές για τον ίδιο προορισμό. Ο συγκεκριμένος πίνακας ενημερώνεται δυναμικά κατά τη διάρκεια λειτουργίας του συστήματος με η χρήση κατάλληλων αλγορίθμων. Τέλος, ο πίνακας ipNetToMediaTable περιέχει την αντιστοιχία διευθύνσεων ip σε φυσικές διευθύνσεις. Η συμπλήρωση των παραπάνω πινάκων μπορούμε να πούμε ότι αναφέρεται στο δεύτερο και τρίτο επίπεδο του επιπέδου OSI, καθώς αφορά τόσο σε MAC όσο και σε IP διευθύνσεις. Σε αυτό το σημείο πρέπει να αναφέρουμε ότι και το address translation group περιέχει ένα πίνακα με την αντιστοιχία φυσικών διευθύνσεων σε ip, ο οποίος πρέπει να συμπληρωθεί. Βέβαια, όπως έχουμε πει το συγκεκριμένο group είναι deprecated στη MIB-II και το πιο πιθανό είναι να μην υπάρχει καν σε επόμενη έκδοση της MIB.

- Θέτουμε τα κατώφλια των traps. Ο διαχειριστής θα πρέπει να ενημερώνεται όταν συμβαίνει κάποιο γεγονός στο δίκτυο. Για παράδειγμα, μπορεί να επιθυμούμε όταν σε ένα interface του σχήματος 3.1 εισέρχεται ένα ποσοστό λάθος πακέτων να ενημερώνεται ο διαχειριστής Μ. Ο καθορισμός των κατωφλίων είναι μία αρκετά σύνθετη υπόθεση και μπορεί να γίνει, τόσο με αναλυτικές μεθόδους, όσο και με παρατήρηση του δικτύου ή προσομοιώσεις. Τα κατώφλια για την αποστολή των traps αναφέρονται σε περισσότερα από ένα στρώματα του OSI, ανάλογα με τη φύση του γεγονότος που προκαλεί το trap. Για παράδειγμα, αν στέλνεται trap όταν η ουρά των εξερχόμενων πακέτων σε ένα interface ξεπερνά ένα ορισμένο μέγεθος, τότε το trap αυτό αναφέρεται στο πρώτο επίπεδο του OSI. Αντίστοιχα, αν ο πράκτορας στέλνει trap όταν τα εισερχόμενα ip πακέτα σε ένα interface έχουν σε υψηλό ποσοστό λάθη στο πεδίο της ip διεύθυνσης, τότε λέμε ότι το trap αυτό αναφέρεται στο τρίτο επίπεδο του OSI.

- Σε κάποιες περιπτώσεις μπορεί να δίνεται η δυνατότητα στο διαχειριστή να θέτει διαφορετικές προτεραιότητες στα πακέτα που κινούνται στο δίκτυο, ανάλογα με την εφαρμογή στην οποία ανήκουν. Για παράδειγμα, μπορεί να αποδίδεται στα πακέτα που ανταλλάσσονται μεταξύ του R1 και του R2 – και

αφορούν σε μία εφαρμογή τηλεδιάσκεψης – μεγαλύτερη προτεραιότητα σε σχέση με αυτά που αφορούν σε μία εφαρμογή ηλεκτρονικού ταχυδρομείου. Η λειτουργία αυτή έχει σχέση με τη δρομολόγηση των πακέτων, οπότε αναφέρεται στο τρίτο επίπεδο του OSI.

Όλα τα παραπάνω συνοψίζονται στον πίνακα 3.1. Στη στήλη *επίπεδο διαχείρισης*, σημειώνεται το επίπεδο διαχείρισης στο οποίο μπορεί να καταταγεί κάθε ενέργεια. Υπάρχουν τρία επίπεδα διαχείρισης: διαχείριση σε επίπεδο στοιχείου δικτύου (EML - Element Management Level), διαχείριση σε επίπεδο δικτύου (NML – Network Management level) και διαχείριση σε επίπεδο υπηρεσίας (SML – Service Management Level).

Διαχείριση διαμόρφωσης	
Ενέργεια	**Επίπεδο διαχείρισης**
Θέτουμε προτεραιότητα στα πακέτα ανάλογα με την εφαρμογή στην οποία ανήκουν	SML
Θέτουμε τα κατώφλια των traps	SML, EML
Συμπλήρωση πινάκων διευθύνσεων/δρομολόγησης	NML,EML
Συμπλήρωση πεδίων System Group	EML

***Πίνακας 3.1**: Σύνοψη ενεργειών στη διαχείριση διαμόρφωσης*

3.2 Ανάλυση ασφάλειας

Στη διαχείριση ασφάλειας μπορούμε να περιλάβουμε τις παρακάτω ενέργειες:

- Τοποθέτηση των δρομολογητών σε φυλασσόμενο χώρο. Η φύλαξη μπορεί να υλοποιηθεί είτε με την τοποθέτηση κλειδαριών, είτε με την τοποθέτηση φύλακα, είτε με την εγκατάσταση συστημάτων που θα επιτρέπουν την είσοδο μόνο σε συγκεκριμένα άτομα. Τα συστήματα αυτά μπορούν να χωριστούν σε τρεις μεγάλες κατηγορίες. Στην πρώτη κατηγορία ανήκουν αυτά που πραγματοποιούν την αναγνώριση με βάση ένα στοιχείο που ο χρήστης κατέχει, για παράδειγμα μία έξυπνη κάρτα. Στη δεύτερη κατηγορία ανήκουν τα συστήματα στα οποία η αναγνώριση πραγματοποιείται με κάτι που ο χρήστης γνωρίζει, για παράδειγμα έναν κωδικό. Τέλος, η τρίτη κατηγορία αποτελείται από τα συστήματα που πραγματοποιούν την αναγνώριση με βάση κάτι που χαρακτηρίζει το χρήστη ως φυσική οντότητα. Τα συστήματα αυτά δηλαδή κάνουν χρήση βιομετρικών τεχνικών αναγνώρισης, όπως αναγνώριση δακτυλικού αποτυπώματος, αναγνώριση φωνής και αναγνώριση της μορφολογίας του αμφιβληστροειδούς χιτώνα. Φυσικά, υπάρχουν και συστήματα που μπορούν να καταταγούν ταυτόχρονο σε περισσότερες από μία κατηγορίες, όπως ένα σύστημα που πραγματοποιεί την αναγνώριση όταν ο χρήστης εισάγει σε αυτό μία κάρτα (κάτι που κατέχει) και δώσει και ένα

κωδικό (κάτι που γνωρίζει). Τα παραπάνω μέτρα αφορούν στη φύλαξη των δρομολογητών σε φυσικό επίπεδο (πρώτο επίπεδο του OSI).

- Στην περίπτωση που εξετάζουμε, οι ζεύξεις Z1 και Z2 είναι μισθωμένες γραμμές, οπότε την ευθύνη για τη φυσική τους ακεραιότητα έχει ο πάροχος.

- Η ανάλυση της επικινδυνότητας του προσωπικού είναι επίσης μία ενέργεια που πρέπει να γίνεται για τη διασφάλιση του δικτύου σε φυσικό επίπεδο, αλλά και για την αποφυγή υποκλοπών, διαρροής πληροφοριών που θα έθιγαν τα συμφέροντα της επιχείρησης κ.τ.λ. Μέρος της ανάλυσης επικινδυνότητας προσωπικού μπορεί να είναι ο έλεγχος του ποινικού μητρώου των εργαζομένων, η διερεύνηση ύποπτων συνδιαλλαγών κάποιων εργαζομένων με ανταγωνίστριες εταιρίες κ.τ.λ.

- Η πρόσβαση στα δεδομένα των δρομολογητών πρέπει να γίνεται επίσης μόνο από συγκεκριμένα άτομα. Για το λόγο αυτό είναι η απαραίτητη η ύπαρξη κωδικών, ώστε να επιτυγχάνεται αυθεντικοποίηση του χρήστη. Η ύπαρξη κωδικών προσφέρει ασφάλεια σε επίπεδο εφαρμογής.

- Μέχρι τώρα αναφερθήκαμε σε μέτρα ασφαλείας που προστατεύουν το δίκτυο από κακόβουλες ενέργειες που προέρχονται κυρίως από το ίδιο το προσωπικό της εταιρίας. Δεν πρέπει να ξεχνάμε όμως, ότι η πιθανότητα να εκδηλωθεί κάποια επίθεση στο δίκτυό μας από τον «έξω» κόσμο, κυρίως μέσω διαδικτύου, είναι πολύ μεγάλη. Για το λόγο αυτό φροντίζουμε να τοποθετηθεί στο δίκτυό μας λογισμικό που θα μπορεί να το προστατέψει από τέτοιες επιθέσεις. Στο λογισμικό αυτό περιλαμβάνονται αναχώματα ασφαλείας (firewalls) και προγράμματα που προστατεύουν από ιός, στη σημασία των οποίων έχουμε αναφερθεί σε προηγούμενο κεφάλαιο.

- Η διασφάλιση της εμπιστευτικότητας των δεδομένων που διακινούνται στο δίκτυο είναι μία ακόμα σημαντική πτυχή της ασφάλειας. Στην περίπτωση που χρησιμοποιούμε μισθωμένες γραμμές η εμπιστευτικότητα μπορεί να εξασφαλίζεται από τον πάροχο, μέσω της υπηρεσίας που χρησιμοποιούμε. Σε διαφορετική περίπτωση, αν κριθεί αναγκαίο, μπορούμε να χρησιμοποιήσουμε κρυπτογράφηση των διακινούμενων δεδομένων. Η κρυπτογράφηση αναφέρεται στο έκτο επίπεδο του OSI.

- Η εξασφάλιση της ακεραιότητας των δεδομένων που διακινούνται στο δίκτυο είναι μία ακόμα μέριμνα που πρέπει να ληφθεί και έχει σχέση με την ασφάλεια. Αναφέρεται στο έκτο επίπεδο του OSI και εξασφαλίζεται κυρίως με τη χρήση συναρτήσεων συνόψεων [5]. Όταν ένα μήνυμα περάσει από μία τέτοια συνάρτηση, παράγεται ένας μεγάλος ακέραιος αριθμός. Το μήνυμα χαρακτηρίζεται μονοσήμαντα από τον αριθμό αυτό. Ο παραλήπτης του μηνύματος λαμβάνει εκτός από το ίδιο το μήνυμα και το αποτέλεσμα της συνάρτησης σύνοψης. Αν περάσει το μήνυμα από την ίδια συνάρτηση σύνοψης και βρει το ίδιο αποτέλεσμα με αυτό που του στάλθηκε, τότε γνωρίζει ότι το μήνυμα δεν αλλοιώθηκε στην πορεία του. Ο μηχανισμός των συνόψεων χρησιμοποιείται συνήθως μαζί με αυτόν της κρυπτογράφησης, ώστε να εξασφαλίζονται, τόσο η εμπιστευτικότητα όσο και η ακεραιότητα

των δεδομένων. Βέβαια, πρέπει να σημειωθεί ότι οι μηχανισμοί κρυπτογράφησης και παραγωγής συνόψεων έχουν αρκετά μεγάλες απαιτήσεις σε υπολογιστικούς πόρους και επιβραδύνουν την απόδοση του συστήματος.

- Οι επιθέσεις τύπου επανεκπομπής μηνυμάτων και η προσπάθεια για συνακροάσεις/υποκλοπές είναι δύο ακόμα προβλήματα που πρέπει να αντιμετωπιστούν. Τα κύριο εργαλείο για τον εντοπισμό τέτοιων επιθέσεων, όπως έχουμε αναλύσει σε προηγούμενο κεφάλαιο, είναι η ανάλυση των στατιστικών που αναφέρονται στις καθυστερήσεις και στον αριθμό των χαμένων πακέτων.

Όλα τα παραπάνω συνοψίζονται στον πίνακα 3.2.

Διαχείριση ασφάλειας	
Ενέργεια	**Επίπεδο διαχείρισης**
Ανάλυση επικινδυνότητας προσωπικού	Διαχείριση έμψυχου δυναμικού
Ύπαρξη κωδικών για πρόσβαση στα δεδομένα (π.χ. community string)	SML
Διασφάλιση εμπιστευτικότητας (μηχανισμός κρυπτογράφησης)	SML
Διασφάλιση ακεραιότητας (μηχανισμός συνόψεων)	SML
Προστασία από επανεκπομπή μηνυμάτων/συνακροάσεις (έλεγχος στατιστικών καθυστέρησης και χαμένων πακέτων)	SML
Φυσική φύλαξη χώρου δρομολογητών (κλειδαριές, φύλακες, συστήματα ασφαλείας)	SML
Εγκατάσταση αναχωμάτων ασφαλείας και προγραμμάτων για προστασία από ιούς	NMS, SML (programs)

***Πίνακας 3.2**: Σύνοψη ενεργειών στη διαχείριση ασφάλειας*

3.3 Ανάλυση απόδοσης

Παρακάτω περιγράφονται όλες εκείνες οι ενέργειες στις οποίες πρέπει να προβούμε, προκειμένου να υλοποιήσουμε τη διαχείριση απόδοσης.

- Καταγράφουμε την ταχύτητα με την οποία διακινούνται δεδομένα από κάθε interface των δρομολογητών. Το αντίστοιχο μέγεθος αναφέρεται στο πρώτο επίπεδο του OSI και εκφράζεται από το αντικείμενο ifSpeed της MIB-II.

- Καταγράφουμε το φόρτο που υπάρχει σε κάθε ζεύξη, ανά ώρα, μέρα, εβδομάδα κ.τ.λ. Δίνουμε ιδιαίτερη έμφαση στις ώρες αιχμής για το δίκτυο. Τα παραπάνω δεδομένα αφορούν στο πρώτο επίπεδο του OSI.

- Καταγράφουμε, από το σύνολο της κίνησης που διακινείται μέσω ενός interface, ποια είναι τα ποσοστά που αντιστοιχούν σε εισερχόμενη και σε εξερχόμενη κίνηση. Τα αντικείμενα που δείχνουν τον αριθμό των εισερχόμενων και τον αριθμό των εξερχόμενων octets σε ένα interface είναι το ifInOctets και το ifOutOctets.

- Καταγράφουμε το ποσοστό των εισερχόμενων πακέτων που έχει λάθη σε κάθε interface. Το αντικείμενο ifInErrors δίνει τον αριθμό των λάθος εισερχόμενων πακέτων σε ένα interface.

- Καταγράφουμε το ποσοστό των εξερχόμενων πακέτων που έχει λάθη σε κάθε interface. Το αντικείμενο ifOutErrors δίνει τον αριθμό των λάθος εξερχόμενων πακέτων από ένα interface.

- Καταγράφουμε το ποσοστό πακέτων που απορρίπτεται στους buffers εισόδου και εξόδου των δρομολογητών, λόγω έλλειψης χώρου, ενώ δεν έχουν λάθη. Τα αντικείμενα της MIB-II που δίνουν τον αριθμό των πακέτων που απορρίπτονται στους buffers εισόδου και εξόδου χωρίς να έχουν σφάλματα είναι το ifInDiscards και το ifOutDiscards αντίστοιχα. Σημειώνουμε ότι όλα τα παραπάνω μεγέθη τα οποία εκφράζονται σε ποσοστά αναφέρονται στο έβδομο επίπεδο του OSI.

- Καταγράφουμε την καθυστέρηση των πακέτων για να φτάσουν από την πηγή στον προορισμό τους. Επειδή η καθυστέρηση αυτή σχετίζεται και με τη δρομολόγηση (πέρα από τις καθυστερήσεις στις ουρές εξυπηρέτησης), λέμε ότι το μέγεθος αυτό αναφέρεται στο τρίτο επίπεδο του OSI.

- Καταγράφουμε το μέγεθος των ουρών εξυπηρέτησης στους δρομολογητές. Για παράδειγμα, το αντικείμενο ifOutQLen μας δίνει το μέγεθος της ουράς που σχηματίζεται από τα πακέτα που στέλνονται από το δρομολογητή. Το μέγεθος αυτό αναφέρεται στο πρώτο επίπεδο του OSI.

- Καταγράφουμε τη διαθεσιμότητα κάθε στοιχείου το δικτύου ξεχωριστά, ολόκληρου του δικτύου σα σύνολο και κάθε εφαρμογής-υπηρεσίας που προσφέρεται από το δίκτυο.

Υπενθυμίζουμε ότι τα στατιστικά που κρατάμε για την απόδοση του δικτύου πρέπει να μπορούν να δώσουν την εικόνα του για διαφορετικές χρονικές κλίμακες, δηλαδή πρέπει για κάθε μέγεθος να έχουμε στατιστικά ανά ώρα, ανά μέρα ανά βδομάδα κ.τ.λ. Επιπλέον, πρέπει να δίνεται έμφαση στην μέγιστη τιμή που παίρνουν κάποια μεγέθη, όπως για παράδειγμα ο φόρτος των ζεύξεων κατά τις ώρες αιχμής.

Επίσης, τονίζουμε πως η αναπαράσταση των στατιστικών απόδοσης με γραφικές παραστάσεις βοηθά ιδιαίτερα στην γρήγορη και εύκολη κατανόησή τους. Μερικές

φορές είναι δυνατόν ρίχνοντας μόνο μία ματιά σε μία γραφική παράσταση να βγάλουμε χρήσιμα συμπεράσματα. Για παράδειγμα μία γραμμή που ο φόρτος της ξεπερνά συνεχώς το 80%, προφανώς χρήζει αναβάθμισης.

Στατιστικά που δείχνουν τον αριθμό των λάθος εισερχόμενων πακέτων μπορούν να κρατηθούν ξεχωριστά και για κάθε είδος κίνησης, όπως έχουμε δείξει και σε προηγούμενο κεφάλαιο.

Τέλος, αναφέρουμε για μία ακόμη φορά ότι τα στατιστικά απόδοσης είναι ιδιαίτερα χρήσιμα και για τις άλλες λειτουργίες διαχείρισης. Σε προηγούμενα κεφάλαια έχουμε δώσει έμφαση στον τρόπο με τον οποίο η απόδοση σχετίζεται με την ασφάλεια.

Οι ενέργειες που πρέπει να κάνουμε και σχετίζονται με την απόδοση ενός δικτύου συνοψίζονται στον πίνακα 3.3.

Διαχείριση απόδοσης	
Ενέργεια	**Επίπεδο διαχείρισης**
Καταγραφή διαθεσιμότητας σε επίπεδο στοιχείων δικτύου, δικτύου και υπηρεσιών	EML, NML, SML
Καταγραφή χρόνου για να φτάσουν τα πακέτα από την πηγή στον προορισμό τους	NML
Καταγραφή πακέτων που απορρίπτονται στους buffers εισόδου και εξόδου χωρίς να έχουν λάθη	EML
Καταγραφή, ανά interface,εισερχόμενων και εξερχόμενων πακέτων που έχουν λάθη	EML
Καταγραφή εισερχόμενης και εξερχόμενης κίνησης ανά interface	EML
Καταγραφή φόρτου ανά ζεύξη	EML
Καταγραφή ταχύτητας διακινούμενων δεδομένων ανά interface	EML
Καταγραφή μεγέθους ουρών	EML

***Πίνακας 3.3**: Σύνοψη ενεργειών στη διαχείριση απόδοσης*

3.4 Ανάλυση κόστους

Έχουμε ήδη αναφέρει τη μεγάλη σημασία της ανάλυσης κόστους στα σύγχρονα επιχειρηματικά περιβάλλοντα. Στην περίπτωση που εξετάζουμε θα ασχοληθούμε μόνο με το κόστος διαχείρισης με τη χρήση SNMP. Ουσιαστικά αυτό μεταφράζεται στην εύρεση της κίνησης που προκαλείται από τη χρήση του συγκεκριμένου πρωτοκόλλου, όπως δηλώνεται και στον πίνακα 3.4

Διαχείριση κόστους	
Ενέργεια	**Επίπεδο διαχείρισης**
Εύρεση ποσοστό φόρτου που προκαλείται από τις λειτουργίες διαχείρισης (κίνηση SNMP)	SML

***Πίνακας 3.4**: Διαχείριση κόστους λειτουργιών διαχείρισης*

Αυτό για το απλό δίκτυο με τους δύο δρομολογητές το οποίο εξετάζουμε σημαίνει υπολογισμό των αθροισμάτων των PDU του SNMP που στέλνονται και λαμβάνονται από τον A1 και τον M. Το άθροισμα για τον A1 (έστω ΣA1) θα δώσει το συνολικό αριθμό SNMP PDUs που κινήθηκαν στη ζεύξη Z2, ενώ το άθροισμα για τον M (έστω ΣM) θα δώσει το συνολικό αριθμό των SNMP PDUs που κινήθηκαν στη ζεύξη Z1. Αν πολλαπλασιάσουμε αυτά τα αθροίσματα με το μέσο μέγεθος ενός SNMP PDU σε octets (έστω MO) και διαιρέσουμε το ΣA1*MO με το συνολικό αριθμό των octets που διακινήθηκαν μέσω του interface I4 (έστω O4) και το ΣM*MO με το συνολικό αριθμό των octets που διακινήθηκαν μέσω του interface I2 (έστω O2) θα πάρουμε το ποσοστό της κίνησης που προκαλείται από το SNMP στις ζεύξεις Z2 και Z1 αντίστοιχα.

Σημειώνουμε, ότι το μέσο μέγεθος του πακέτου SNMP μπορεί να βρεθεί με προσομοιώσεις. Επίσης, μπορούμε να θέσουμε το MO ίσο με το μέγιστο μέγεθος πακέτου SNMP που μπορεί να υποστηρίξει το σύστημα, απλοποιώντας τα πράγματα. Βέβαια, αν ακολουθηθεί αυτή η σύμβαση, το αποτέλεσμα που θα πάρουμε στο τέλος θα δηλώνει ότι το ποσοστό της κίνησης SNMP στο δίκτυο είναι μεγαλύτερο από το πραγματικό.

Τα αντικείμενα snmpInPks και snmpOutPkts δίνουν τον αριθμό SNMP PDUs που λαμβάνονται και αποστέλλονται από μία οντότητα SNMP, ενώ τα αντικείμενα ifInOctets και ifOutOctets δίνουν τον αριθμό των octets που εισήλθαν και εξήλθαν από ένα Interface.

Είπαμε ότι εμείς θα ασχοληθούμε μόνο με τον υπολογισμό του κόστους της κίνησης SNMP, όμως η λογική για την εύρεση του κόστους και των άλλων κινήσεων ακολουθεί τη λογική που περιγράφηκε παραπάνω. Επίσης, αν θέλουμε να κάνουμε μία πλήρη ανάλυση κόστους θα πρέπει να λάβουμε υπόψη και άλλες παραμέτρους που δε σχετίζονται αναγκαστικά με την τεχνολογία που χρησιμοποιείται για να λειτουργήσει το δίκτυο. Για παράδειγμα, οι δρομολογητές πρέπει να βρίσκονται σε κάποιο χώρο τον οποίο ενοικιάζουμε (ή κατέχουμε), οπότε πρέπει να υπολογιστεί πόσα πληρώνουμε για ενοίκιο (ή πόσα χάνουμε που δεν έχουμε τη δυνατότητα να αξιοποιήσουμε το χώρο με άλλο τρόπο).

Άλλες παράμετροι που έχουν σχέση με την ανάλυση κόστους και οι οποίες πρέπει να εξεταστούν είναι τα χρήματα τα οποία ξοδεύτηκαν για την απόκτηση των δρομολογητών, τα χρήματα που ξοδεύονται για τη φύλαξή τους, τα χρήματα που ξοδεύονται για αναβαθμίσεις και συντήρηση, τα χρήματα που πληρώνονται στον πάροχο για τις μισθωμένες γραμμές, τα χρήματα που ξοδεύονται για την ενέργεια που καταναλώνουν οι δρομολογητές κ.τ.λ.

Τελικά, το σύνολο των χρημάτων που έχουμε υπολογίσει ότι ξοδεύεται μπορεί να επιμεριστεί σε κάθε είδος κίνησης του δικτύου μας, ανάλογα με το ποσοστό φόρτου

που αυτή προκαλεί, ώστε να έχουμε μία εκτίμηση του κόστους κάθε τύπου κίνησης και σε χρήματα.

3.5 Ανάλυση σφαλμάτων [11], [14], [17], [18]

Σε προηγούμενα κεφάλαια έχουμε αναφέρει ότι η διαχείριση σφαλμάτων περιλαμβάνει την ανίχνευση κάποιου σφάλματος μέσω των συμπτωμάτων που προκαλεί στο δίκτυο, την αναφορά του, το συσχετισμό των συμπτωμάτων με κάποια μέθοδο, τον εντοπισμό του σφάλματος και τελικά την επιδιόρθωσή του. Επίσης, έχουμε αναφέρει την αναγκαιότητα να κρατούνται αρχεία log με τα σφάλματα που προκλήθηκαν, αλλά και τα προβλήματα που παρατηρούνται, όταν η διαδικασία αυτή εφαρμόζεται σε πραγματικό περιβάλλον.

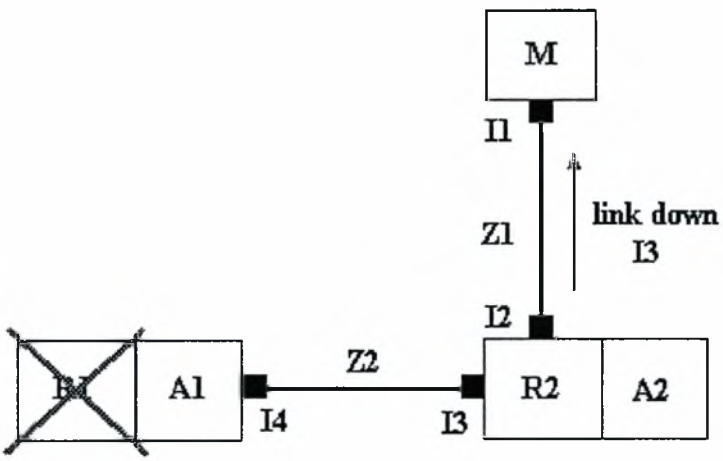
Η διαχείριση σφαλμάτων έχει άμεση σχέση με τη διαχείριση απόδοσης, αφού, για παράδειγμα, η απόρριψη μεγάλου αριθμού πακέτων που δεν έχουν λάθη ή η λήψη μεγάλου αριθμού πακέτων που έχουν λάθη μπορεί να υποδηλώνει κάποιο πρόβλημα που αντιμετωπίζει το δίκτυο. Με τις παραμέτρους της απόδοσης που έχουν σχέση με την αξιοπιστία του συστήματος δε θα ασχοληθούμε εδώ, καθώς τις έχουμε ήδη αναλύσει στην ενότητα 3.3. Στην παρούσα ενότητα θα ασχοληθούμε με το θέμα της διαθεσιμότητας και με την προβολή του στο σχήμα 3.1, το οποίο όπως έχουμε πει είναι μία κλιμάκωση προς τα κάτω ενός πραγματικού δικτύου εταιρίας. Στον πίνακα 3.5 παραθέτουμε τις ενέργειες που έχουν να κάνουν με τη λειτουργία της διαχείρισης σφαλμάτων, ενώ στο ακόλουθο κείμενο αναλύουμε τι γίνεται, όταν παρουσιαστεί μία βλάβη στο δίκτυο που έχει ως αποτέλεσμα την απώλεια της διαθεσιμότητας.

Διαχείριση σφαλμάτων	
Ενέργεια	**Επίπεδο διαχείρισης**
Μελέτη και ανάλυση στατιστικών που σχετίζονται με τη διαθεσιμότητα	EML, NML, SML
Μελέτη και ανάλυση στατιστικών απόδοσης που σχετίζονται με την αξιοπιστία	EML, NML, SML
Ανίχνευση σφάλματος	EML, NML, SML
Αναφορά σφάλματος	EML, NML, SML
Συσχετισμός συμπτωμάτων	NML, SML
Εντοπισμός σφάλματος	EML, NML, SML
Επιδιόρθωση σφάλματος	EML, NML, SML
Καταγραφή σφάλματος σε αρχείο log	EML, NML, SML

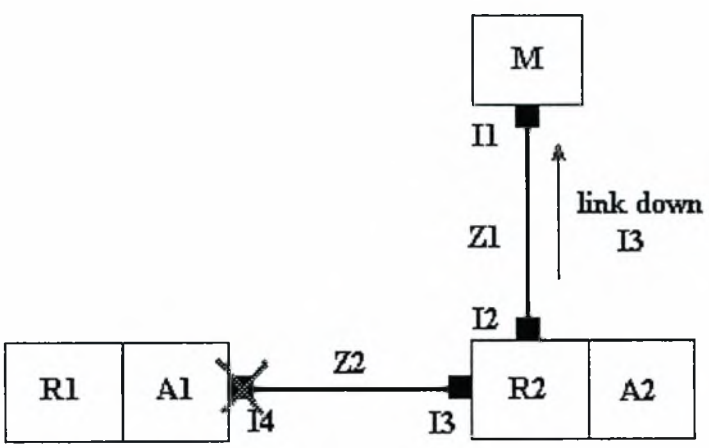
Πίνακας 3.5: *Σύνοψη ενεργειών στη διαχείριση σφαλμάτων*

Ας υποθέσουμε ότι ο A2 στέλνει στο διαχειριστή ένα trap τύπου link down για το interface I3. Αυτό πρακτικά σημαίνει ότι ο διαχειριστής δεν μπορεί να έχει καμία επαφή με τον A1. Το ερώτημα είναι ποιο είναι το πραγματικό σφάλμα που είχε ως αποτέλεσμα την αποστολή του συγκεκριμένου trap. Οι πιθανές εξηγήσεις είναι τέσσερις:

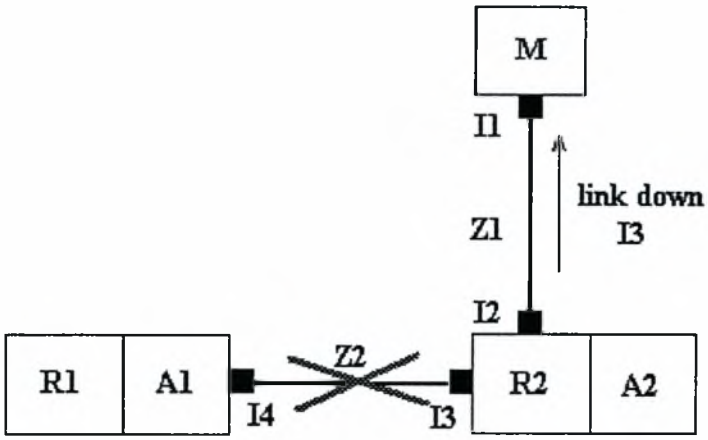
1) Ο δρομολογητής R1 δε λειτουργεί
2) Το interface I4 δε λειτουργεί
3) Η ζεύξη Z2 δε λειτουργεί
4) Το interface I3 δε λειτουργεί



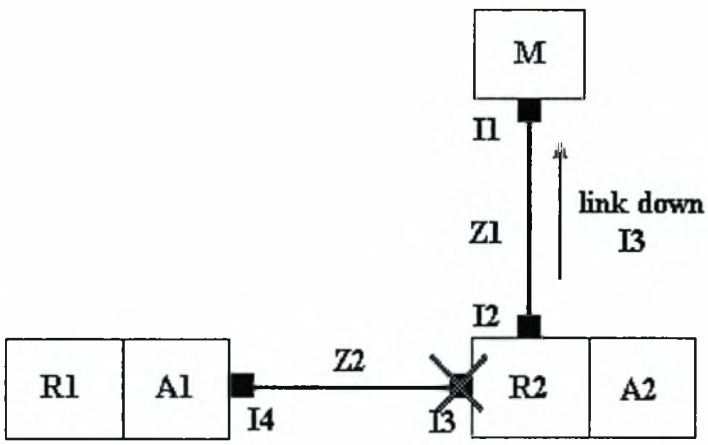
Σχήμα 3.3.1: Βλάβη στον R1



Σχήμα 3.3.2: Βλάβη στο I4



Σχήμα 3.3.3: *Βλάβη στη Z2*



Σχήμα 3.3.4: *Βλάβη στο I3*

Και οι τέσσερις παραπάνω βλάβες, όπως παρουσιάζεται και στα σχήματα 3.3.1 ως 3.3.4, θα είχαν ως αποτέλεσμα τη δημιουργία του ίδιου trap και το ίδιο ακριβώς σύμπτωμα: η επικοινωνία μεταξύ A1 και διαχειριστή θα χάνονταν. Επίσης, κανείς δε μπορεί να αποκλείσει την πιθανότητα να έχουν συμβεί ταυτόχρονα περισσότερες από μία βλάβες.

Στο σημεία αυτό θα κάνουμε μία αναφορά στα μηνύματα που θα στέλνονταν αν χρησιμοποιούνταν το μοντέλο διαχείρισης OSI, δηλαδή το πρωτόκολλο CMIP, στην περίπτωση των παραπάνω βλαβών. Με τον τρόπο αυτό θα αναδειχθούν οι δυνατότητες που δίνει η διαχείριση με βάση το μοντέλο OSI.

Αρχικά, πρέπει να πούμε ότι κάθε επεξεργαστής ή κάρτα (οπότε και κάθε interface) που υπάρχει στο δίκτυο είναι ένα ξεχωριστό στιγμιότυπο της κλάσης circuitPack. Η κλάση αυτή περιέχει το γνώρισμα (attribute) operationalState, το οποίο δηλώνει αν το συγκεκριμένο διαχειριζόμενο αντικείμενο λειτουργεί ή όχι. Αλλαγή της τιμής του operationalState, προκαλεί την αποστολή notification (notification change attribute) που δηλώνει ποια είναι η νέα κατάσταση του αντικειμένου.

Αντίστοιχα, κάθε ζεύξη είναι στιγμιότυπο της κλάσης trailR1, σε αντίθεση με το SNMP, όπου δεν υπάρχει διαχειριζόμενο αντικείμενο για τις ζεύξεις. Η κλάση trailR1 περιέχει, όπως και η circuitPack, το γνώρισμα operationalState, η αλλαγή της τιμής του οποίου προκαλεί την αποστολή του αντίστοιχου notification. Επιπλέον, υπάρχει η κλάση terminationPoint που χρησιμοποιείται για τη δημιουργία διαχειριζόμενων αντικειμένων που αντιστοιχούν στα άκρα μίας ζεύξης. Προφανώς, κάθε ζεύξη έχει δύο άκρα. Η κλάση terminationPoint διαθέτει, όμοια με τις κλάσεις που προαναφέραμε, το γνώρισμα operationalState.

Στη συνέχεια θα υποθέτουμε ότι οι πράκτορες που είναι υπεύθυνοι για τα interfaces και τους επεξεργαστές «κατοικούν» στους αντίστοιχους δρομολογητές όπου βρίσκονται τα στοιχεία αυτά. Ο πράκτορας που είναι υπεύθυνος για τη Z1 και τα άκρα της βρίσκεται στον κόμβο M και ο πράκτορας που είναι υπεύθυνος για τη Z2 και τα άκρα της βρίσκεται στον κόμβο R2.

Με βάση τα παραπάνω, αν παρουσιαστεί βλάβη στο I3 (σχήμα 3.3.4), αντί του trap linkDown I3, θα αποσταλεί ένα notification που δηλώνει τη βλάβη του I3 και ένα notification που δηλώνει πρόβλημα στο αντίστοιχο άκρο της Z2. Είναι προφανές ότι με τον τρόπο αυτό ο διαχειριστής γνωρίζει ποια είναι η πραγματική βλάβη.

Αν παρουσιαστεί βλάβη στη Z2 (σχήμα 3.3.3), αντί του trap linkDown I3, θα αποσταλεί ένα notification που δηλώνει τη βλάβη της Z2. Άρα και σε αυτή την περίπτωση εντοπίζεται αμέσως η πραγματική αιτία της βλάβης.

Αν παρουσιαστεί βλάβη στο I4 (σχήμα 3.3.2), αντί του trap linkDown I3, θα αποσταλεί ένα notification που δηλώνει πρόβλημα στο αντίστοιχο άκρο της Z2. Αφού ο πράκτορας που είναι υπεύθυνος για το I4 βρίσκεται στον R1, το I4 έχει πρόβλημα και δεν υπάρχει εναλλακτική ζεύξη επικοινωνίας με τον διαχειριστή, το notification που δηλώνει τη βλάβη του I4 δε θα καταφέρει να φτάσει στο διαχειριστή. Σημειώνεται ότι το notification για μη λειτουργία του άκρου της Z2 θα ήταν ίδιο και στην περίπτωση που δε λειτουργούσε ο R1. Οπότε ο διαχειριστής δε μαθαίνει αμέσως ποια ήταν η πραγματική αιτία του προβλήματος.

Παρόλα αυτά, μόλις αποκατασταθεί η βλάβη, θα αποσταλεί notification ότι το συγκεκριμένο interface ξαναλειτουργεί (και ότι το αντίστοιχο άκρο της Z2 δεν έχει πρόβλημα), οπότε ο διαχειριστής θα γνωρίζει ποια ήταν η αιτία του προβλήματος. Για την ακρίβεια, επειδή όταν παρουσιάζεται βλάβη σε ένα interface αυτό αλλάζεται, τα notifications που θα αποσταλούν στο διαχειριστή θα είναι για τη δημιουργία (creation) ενός νέου διαχειριζόμενου αντικειμένου (του συγκεκριμένου interface) και τη διαγραφή (deletion) του παλιού, οπότε η αιτία της βλάβης γίνεται ξεκάθαρη.

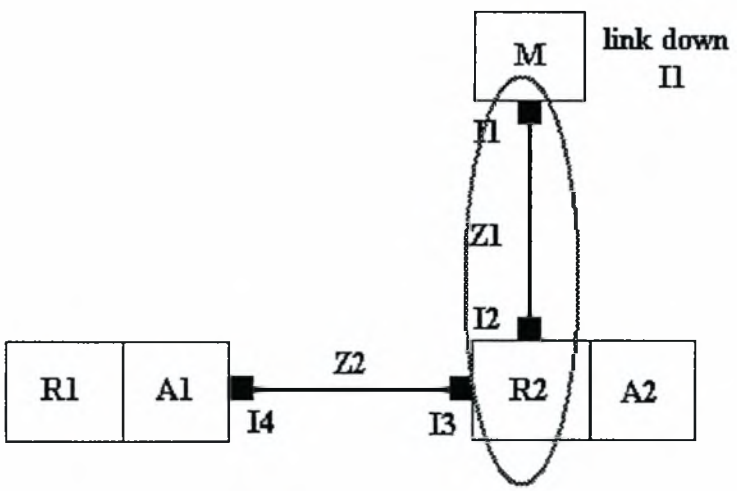
Αν παρουσιαστεί βλάβη στο δρομολογητή R1 (σχήμα 3.3.1), δηλαδή πάψει να λειτουργεί ο επεξεργαστής του, τότε, αντί του trap linkDown I3, θα σταλεί notification για πρόβλημα στο αντίστοιχο άκρο της Z2. Επιπλέον, θα πρέπει να αποσταλεί notification για μη λειτουργία του επεξεργαστή. Βέβαια, για να αποσταλεί το notification αυτό θα πρέπει να υποθέσουμε ότι ο πράκτορας που βρίσκεται στον R1 λειτουργεί ανεξάρτητα από τον επεξεργαστή του δρομολογητή, έχει δικιά του πηγή ενέργειας κ.τ.λ. Εμείς θα υποθέσουμε ότι ο πράκτορας δεν έχει τέτοιες δυνατότητες, οπότε ο διαχειριστής δε μαθαίνει αμέσως πιο είναι το πραγματικό πρόβλημα.

Όμοια με πριν, μόλις αποκατασταθεί η βλάβη, θα σταλεί notification ότι ξανάρχισε να λειτουργεί ο επεξεργαστής (και φυσικά ότι το αντίστοιχο άκρο της Z2 ξαναλειτουργεί), οπότε και πάλι εντοπίζεται με ακρίβεια η αιτία του προβλήματος.

Ξαναπερνώντας στην περίπτωση που η διαχείριση του δικτύου γίνεται με SNMP, αν ο διαχειριστής λάμβανε trap που δήλωνε ότι το interface I1 είναι down οι πιθανοί λόγοι θα ήταν:

1) Ο δρομολογητής R2 δε λειτουργεί
2) Το interface I2 δε λειτουργεί
3) Η ζεύξη Z1 δε λειτουργεί
4) Το interface I1 δε λειτουργεί

Όλοι οι παραπάνω λόγοι είναι πιθανοί και η ανάλυσή τους – τόσο με βάση το SNMP όσο και με βάση το CMIP – είναι όμοια με πριν. Επιπλέον, κανείς δε μπορεί να αποκλείσει το γεγονός να έχουν συμβεί ταυτόχρονα περισσότερες από μία βλάβες. Μάλιστα κάποια βλάβη μπορεί να έχει συμβεί και στο R1, στα I4 και I3 ή στη ζεύξη Z2, απλώς επειδή ο διαχειριστής δε μπορεί να επικοινωνήσει με τον A2 δε θα λάβει και το ανάλογο trap.



Σχήμα 3.4: *Πιθανά σημεία βλάβης όταν χάνεται επαφή με τον R2*

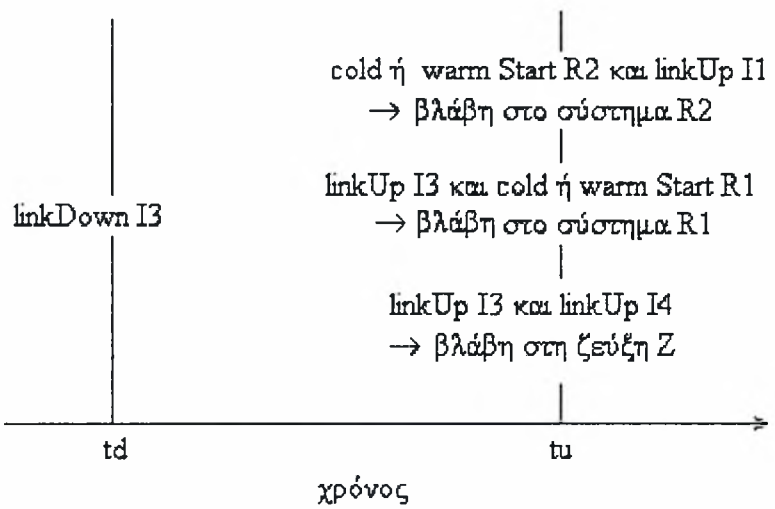
Ακόμα και όταν οι βλάβες αποκατασταθούν ο διαχειριστής θα λάβει τα αντίστοιχα trap που δηλώνουν την επισκευή της βλάβης (linkUp, warmStart, coldStart), αλλά και πάλι δε θα ξέρει ποια ήταν η αιτία του προβλήματος. Αυτό ακριβώς είναι το πρόβλημα που επισημάναμε ότι αντιμετωπίζει και η πραγματική επιχείρηση: κατέχει στατιστικά για τη διαθεσιμότητα αλλά δεν ξέρει την αιτία που προκάλεσε την απώλεια της διαθεσιμότητας. Παρακάτω θα επιχειρήσουμε να περιγράψουμε τη λογική με την οποία θα πρέπει να «διερευνήσει» ο διαχειριστής τις πιθανές αιτίες του προβλήματος, όταν πλέον αυτό λυθεί. Σκοπός είναι η δημιουργία μίας αυτοματοποιημένης μεθόδου, η οποία θα μας λέει κάθε φορά που έχουμε απώλεια της διαθεσιμότητας στο δίκτυο, ποιος ήταν ο υπεύθυνος: ο πάροχος (δηλαδή οι μισθωμένες γραμμές) ή ο ιδιόκτητος εξοπλισμός της επιχείρησης (δηλαδή κάρτες δικτύου και δρομολογητές).

Ας υποθέσουμε ότι ο διαχειριστής λαμβάνει trap link down για το I3 από τον A2 τη χρονική στιγμή td. Τη στιγμή αυτή ο διαχειριστής δε θα κάνει καμία ενέργεια, αλλά θα περιμένει μέχρι να επιδιορθωθεί η βλάβη (η οποία όπως είπαμε μπορεί να

οφείλεται στο I3, στο I4 στη Z2 ή τον R1). Όταν επιδιορθωθεί η βλάβη ο A2 θα στείλει στο διαχειριστή ένα trap. Αν το trap αυτό είναι τύπου coldStart ή warmStart, τότε η αιτία του προβλήματος ήταν το σύστημα R2-I3 (για την ακρίβεια το I3).

Όπως έχουμε ήδη πει σε προηγούμενο κεφάλαιο, τα trap coldStart και warmStart δηλώνουν ότι επανεκκινήθηκε ένας δρομολογητής και μόνη τους διαφορά είναι ότι το coldStart δηλώνει επιπλέον ότι έχουν αλλάξει και κάποιες ρυθμίσεις του δρομολογητή. Επίσης, όπως θα εξηγήσουμε και παρακάτω, για να επισκευαστεί ένα interface πρέπει να κλείσει ο δρομολογητής στον οποίο ανήκει. Έτσι, στην περίπτωση που είχε πρόβλημα το I3, θα έκλεινε ο R2 (οπότε θα στέλνονταν linkDown trap για το I1), θα αλλάζονταν η καμένη κάρτα δικτύου και θα επανεκκινούνταν ο R2 (οπότε θα στέλνονταν μαζί με το trap που δηλώνει την επανεκκίνηση και ένα trap linkUp για το I1).

Στην περίπτωση που ο A2 στείλει trap link up για το I3 και ο A1 στείλει trap τύπου coldStart ή warmStart, τότε η αιτία του προβλήματος ήταν το σύστημα R1-I4. Αν απλώς σταλεί ένα trap link up για το I3 από τον A2 και ένα trap link up για το I4 από τον A1, τότε το πρόβλημα ήταν στη ζεύξη Z2. Τα παραπάνω συνοψίζονται στο σχήμα 3.5, όπου tu ονομάζουμε τη χρονική στιγμή που στάλθηκε το trap που δηλώνει την επιδιόρθωση.



Σχήμα 3.5: *Traps που φτάνουν στο διαχειριστή*

Επειδή, όπως αναφέραμε, μπορεί στο ίδιο χρονικό διάστημα να συμβούν περισσότερες από μία βλάβες, θα πρέπει ο διαχειριστής, όταν λάβει το trap που δηλώνει την επισκευή της βλάβης, να ρωτήσει όλους τους δρομολογητές που βρίσκονται «κάτω» από το interface για το οποίο στάλθηκε το link up trap (ή «κάτω» από το δρομολογητή που έστειλε το coldStart ή warmStart trap) ποιο είναι το sysUpTime τους. Με τον τρόπο αυτό ο διαχειριστής μαθαίνει αν οι δρομολογητές αυτοί λειτουργούσαν από το td, μέχρι να επισκευαστεί η βλάβη. Δηλαδή, στο σχήμα 3.1.α, αν στέλνονταν cold ή warm start trap για τον R2, θα έπρεπε ο διαχειριστής να ρωτήσει το sysUpTime του R1, ή αντίστοιχα αν στέλνονταν linkUp trap για το I3 θα έπρεπε πάλι να ρωτηθεί ο A1 για το sysUpTime του R1.

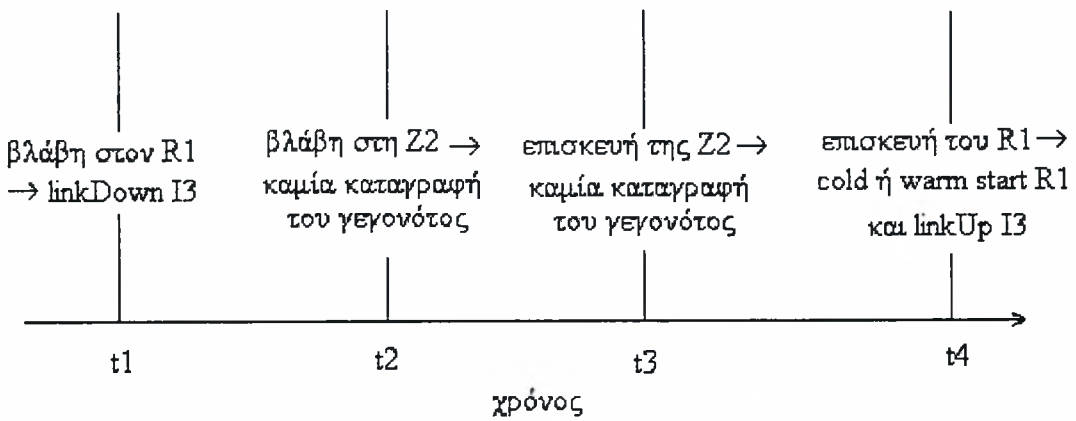
Η λογική αντιμετώπισης μίας βλάβης που θα είχε ως σύμπτωμα την αποστολή ενός link down trap για το I1 είναι ακριβώς η ίδια με αυτή που περιγράφηκε παραπάνω.

Σε αυτό το σημείο πρέπει να κάνουμε κάποιες παρατηρήσεις. Η πρώτη έχει σχέση με την αντιμετώπιση των δρομολογητών σαν να είναι ένα και το αυτό με τα interface τους. Ο λόγος είναι ότι ακόμα και αν έχει βλάβη κάποιο interface και πρέπει να αλλαχτεί, για να γίνει αυτό πρέπει να κλείσει για λίγο ο δρομολογητής στον οποίο ανήκει και να επανεκινηθεί, οπότε θα επηρεαστεί το sysUpTime του και θα σταλεί ένα coldStart ή warmStart trap. Επίσης, στην πραγματικότητα οι περιπτώσεις στις οποίες παρατηρείται βλάβη σε μία κάρτα δικτύου, είναι πολύ λιγότερες από τις περιπτώσεις στις οποίες παρατηρείται πρόβλημα, που έχει ως αποτέλεσμα τη μη διαθεσιμότητα του δικτύου, σε μία ζεύξη ή ένα δρομολογητή, οπότε ξεχωριστή αντιμετώπιση τέτοιων βλαβών μάλλον έχει ως αποτέλεσμα μόνο επιπλέον φόρτο για τους υπολογισμούς και αύξηση της πολυπλοκότητας των χρησιμοποιούμενων αλγορίθμων. Τέλος, το πιο σημαντικό πρόβλημα είναι ότι οι δυνατότητες που δίνει το SNMP δεν επιτρέπουν την αποτελεσματική ξεχωριστή αντιμετώπιση των βλαβών interfaces και δρομολογητών.

Η δεύτερη παρατήρηση έχει να κάνει με περιπτώσεις βλαβών που η λύση που προτείναμε δεν ανιχνεύει. Οι βλάβες αυτές είναι βλάβες των ζεύξεων που συμβαίνουν την ίδια ώρα που έχει συμβεί και κάποια άλλη βλάβη σε ένα δρομολογητή. Για να γίνει κατανοητό αυτό αναφέρουμε τα παρακάτω δύο παραδείγματα:

Παράδειγμα 1

Έστω ότι συμβαίνει μία βλάβη στον R1, οπότε ο A2 στέλνει το trap linkDown για το I3 στο M. Αν μέχρι να επιδιορθωθεί η βλάβη, «πέσει» και η ζεύξη Z2, ο M δε θα ενημερωθεί ποτέ για αυτό. Στην περίπτωση μάλιστα που η ζεύξη Z2 επισκευαστεί πριν επισκευαστεί ο R1, δεν υπάρχει καμία απολύτως δυνατότητα να εντοπιστεί η βλάβη αυτή, καθώς ο διαχειριστής δε λαμβάνει ποτέ trap που να τη δηλώνει και δεν καταγράφεται πουθενά η χρονική στιγμή από την οποία η ζεύξη είναι και πάλι διαθέσιμη. Μία λύση στο συγκεκριμένο πρόβλημα θα ήταν ο ορισμός μίας ιδιωτικής MIB στην οποία η ζεύξη να είναι ένα ξεχωριστό διαχειριζόμενο αντικείμενο με δικό του sysUpTime. Η αλληλουχία των γεγονότων του παραπάνω παραδείγματος φαίνεται στο σχήμα 3.6.

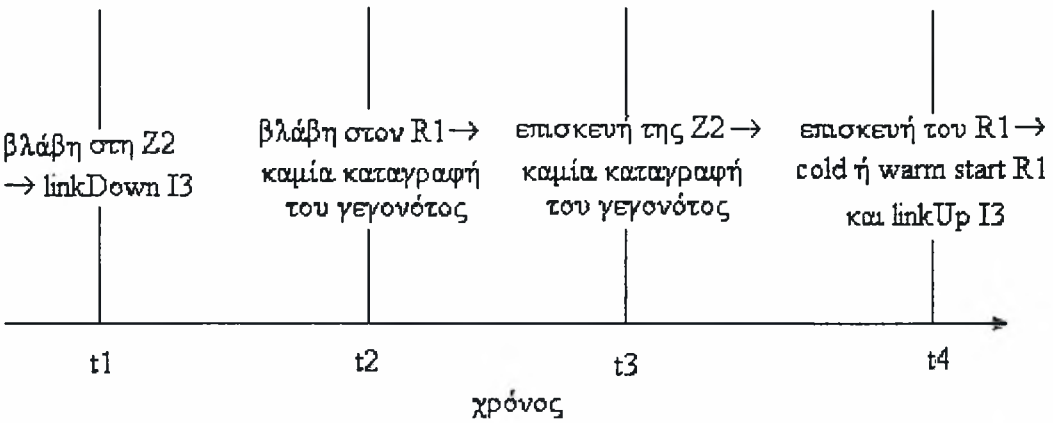


Σχήμα 3.6: *Ακολουθία γεγονότων παραδείγματος 1*

Παράδειγμα 2

Μία άλλη περίπτωση βλάβης της ζεύξης Z2, η οποία δεν καταγράφεται, είναι η εξής:

Έστω ότι παθαίνει βλάβη η ζεύξη Z2. Αυτό έχει ως αποτέλεσμα να σταλεί ένα trap linkDown για το I3 από τον A2 τη χρονική στιγμή t1. Στη συνέχεια παθαίνει βλάβη και ο δρομολογητής R1 τη χρονική στιγμή t2, όμως στο διαχειριστή δε θα σταλεί κανένα μήνυμα. Τη χρονική στιγμή t3 επισκευάζεται η Z2, όμως αφού ο R1 εξακολουθεί να έχει βλάβη, ο A2 δε βλέπει καμία διαφορά στην κατάσταση του I3, οπότε δε στέλνει κανένα μήνυμα στο διαχειριστή. Τη χρονική στιγμή t4 επισκευάζεται ο R1, οπότε ο A1 στέλνει ένα μήνυμα cold ή warm start για τον R1 και ο A2 ένα linkUp trap για το I3. Είναι φανερό πως σε αυτή την περίπτωση εντοπίζεται μόνο η βλάβη του R1. Η παραπάνω ακολουθία γεγονότων συνοψίζεται στο σχήμα 3.7.



Σχήμα 3.7: Ακολουθία γεγονότων παραδείγματος 2

Στην πραγματικότητα, αν υπήρχε διαθέσιμη μία εναλλακτική γραμμή για την επικοινωνία του M με τον A1, όταν έπεφτε η Z2 θα έφτανε στο διαχειριστή και ένα linkDown trap για το I4 – που θα δήλωνε έμμεσα ότι τη συγκεκριμένη χρονική στιγμή ο R1 λειτουργούσε – οπότε ο διαχειριστής θα είχε τη δυνατότητα να κάνει τους κατάλληλους συσχετισμούς και να εντοπίσει όλες τις βλάβες. Δυστυχώς όμως, όπως είπαμε, η τοπολογία των πραγματικών δικτύων είναι συνήθως όμοια με αυτή του παραδείγματός μας.

Με βάση όσα είπαμε όταν αναφερθήκαμε στα μηνύματα που αποστέλλονται, όταν χρησιμοποιείται το OSI/CMIP, πρέπει να έχει γίνει ήδη σαφές, ότι το πρόβλημα του εντοπισμού της αιτίας μίας βλάβης είναι ευκολότερο στην περίπτωση που χρησιμοποιείται το συγκεκριμένο μοντέλο διαχείρισης.

Έτσι στο παράδειγμα 1 (σε αντιστοιχία με το σχήμα 3.6), την t1 αποστέλλεται notification, που δηλώνει ότι υπάρχει πρόβλημα στο άκρο της Z2 προς τον R1. Έτσι, διαχειριστής μαθαίνει ότι δε λειτουργεί ή το I4 ή ο R1.

Την t2 αποστέλλεται notification που δηλώνει τη βλάβη της Z2, έτσι ο διαχειριστής ενημερώνεται για τη βλάβη αυτή!

Την t3 αποστέλλεται notification που δηλώνει την επισκευή της Z2.

Την t4 επισκευάζεται ο R1, οπότε αποστέλλονται notification για να δηλώσουν ότι επεξεργαστής λειτουργεί ξανά και ότι το αντίστοιχο άκρο της Z2 δεν έχει πρόβλημα, οπότε γίνεται σαφές ότι το πρόβλημα το είχε ο επεξεργαστής του R1 και όχι το I4. Αν είχε πρόβλημα το I4 θα έπρεπε να αντικατασταθεί και θα στέλνονταν το αντίστοιχο notification, όπως εξηγήσαμε και παραπάνω.

Στο παράδειγμα 2, αν η διαχείριση του δικτύου γινόταν με OSI/CMIP, η ακολουθία των γεγονότων θα είχε ως εξής:

Τη χρονική στιγμή t1 αποστέλλεται notification που δηλώνει τη βλάβη της Z2, οπότε η βλάβη αυτή εντοπίζεται εξαρχής.

Τη χρονική στιγμή t2 χαλάει ο R1, οπότε ο διαχειριστής δεν έχει αντίστοιχη ενημέρωση.

Τη χρονική στιγμή t3 επισκευάζεται η Z2 και στέλνεται το αντίστοιχο notification στο διαχειριστή.

Τη χρονική στιγμή t4 επισκευάζεται ο R1 και στέλνονται τα αντίστοιχα notification (ίδια με αυτά που στάλθηκαν την t4 του παραδείγματος 1), οπότε εντοπίζεται και η βλάβη του R1.

Σε αυτό το σημείο πρέπει να αναφέρουμε ότι υπάρχει η κλάση stateChangeRecord στην οποία καταγράφονται σε μορφή log file αλλαγές στις τιμές των attributes και πότε αυτές συνέβηκαν. Δημιουργώντας ένα στιγμιότυπο της κλάσης για κάθε διαχειριζόμενο αντικείμενο, έχουμε τη δυνατότητα να αποθηκεύσουμε τη στιγμή από την οποία ξανάρχισε να λειτουργεί κανονικά το αντικείμενο αυτό. Έτσι, έχουμε το αντίστοιχο του sysUpTime, οριζόμενο όμως και για του δρομολογητές και για τα interfaces και για τις ζεύξεις. Αντίθετα, στο SNMP το sysUpTime ορίζεται μόνο για τους δρομολογητές.

Στο επόμενο κεφάλαιο θα αναπτύξουμε μία προσομοίωση για τον εντοπισμό βλαβών σε ένα δίκτυο με δεντροειδή μορφή. Όπως θα δούμε αναλυτικά, τα διαχειριζόμενα αντικείμενα που θα ορίσουμε για τη μοντελοποίηση του δικτύου της προσομοίωσης, δε βρίσκονται σε πλήρη αντιστοιχία με αυτά που ορίζονται στο SNMP, καθώς θα ορίσουμε τις ζεύξεις σα διαχειριζόμενα αντικείμενα, όμως ο αλγόριθμος που θα προτείνουμε αναλύει τα μηνύματα SNMP που φτάνουν στο διαχειριστή και ακολουθεί τη λογική που σκιαγραφήσαμε στην παρούσα ενότητα.

Κεφάλαιο 4

ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΓΙΑ ΤΗΝ ΕΥΡΕΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ

Στο προηγούμενο κεφάλαιο σκιαγραφήσαμε έναν αλγόριθμο για την αυτοματοποιημένη καταγραφή προβλημάτων που έχουν ως αποτέλεσμα την απώλεια της διαθεσιμότητας σε ένα δίκτυο. Βασικός στόχος του αλγορίθμου είναι να γίνει διάκριση ανάμεσα στα σφάλματα που οφείλονται στον ιδιόκτητο εξοπλισμό της εταιρίας και σε αυτά που οφείλονται στον πάροχο των μισθωμένων γραμμών, καθώς επίσης και να υπολογιστεί η διάρκεια των βλαβών. Στο παρόν κεφάλαιο θα περιγράψουμε αναλυτικότερα και θα υλοποιήσουμε το συγκεκριμένο αλγόριθμο.

Για να καταστεί δυνατή η αξιολόγηση ενός αλγορίθμου που εντοπίζει βλάβες και χρόνους βλαβών, πρέπει να είναι γνωστές οι πραγματικές βλάβες και η διάρκειά τους, αφού μόνο έτσι μπορούν να εξαχθούν τα ποσοστά επιτυχίας του αλγορίθμου. Για να είμαστε σε θέση να ξέρουμε ακριβώς ποιες βλάβες συνέβησαν και πόσο διήρκεσαν, υλοποιήσαμε ένα εικονικό δίκτυο, δηλαδή ένα δίκτυο που δεν αποτελείται από φυσικά μέρη, αλλά στην ουσία είναι ένα λογισμικό που προσομοιώνει τις λειτουργίες που θέλαμε να μελετήσουμε.

Το εικονικό δίκτυο – σε αναλογία με ένα πραγματικό δίκτυο – αποτελείται από διάφορα στοιχεία: δρομολογητές, ζεύξεις, interfaces. Σε επόμενη ενότητα θα εξηγήσουμε πώς μοντελοποιήσαμε τα στοιχεία αυτά. Όπως θα δούμε, αν και ο αλγόριθμος που θα προτείνουμε εφαρμόζεται σε δεδομένα διαχείρισης του SNMP, η μοντελοποίηση που επιλέξαμε για τα διαχειριζόμενα αντικείμενα δε βρίσκεται σε πλήρη αντιστοιχία με τον τρόπο που αυτά ορίζονται στο SNMP. Φυσικά, αυτό δε θέτει κανένα περιορισμό στην εφαρμογή του αλγορίθμου σε ένα δίκτυο στο οποίο τα διαχειριζόμενα αντικείμενα ακολουθούν τον ορισμό της MIB-II.

Σε επόμενη ενότητα θα εξηγήσουμε ότι στο εικονικό δίκτυο παράγονται σφάλματα που προκαλούν απώλεια της διαθεσιμότητας και αποστολή των ανάλογων μηνυμάτων στο διαχειριστή. Ο αλγόριθμος εφαρμόστηκε στα δεδομένα που τελικά φτάνουν στο διαχειριστή. Το εικονικό δίκτυο μας επέτρεπε να γνωρίζουμε τα σφάλματα που προκλήθηκαν πραγματικά, καθώς και τη διάρκειά τους, οπότε μπορούσαμε να υπολογίσουμε την απόδοση του αλγορίθμου. Όπως θα δούμε παρακάτω, η απόδοση του αλγορίθμου εξαρτάται από τη διάρκεια των σφαλμάτων και από τη συχνότητα με την οποία παρουσιάζονται.

4.1 Πληροφοριακό μοντέλο

Στην παρούσα ενότητα θα παρουσιάσουμε τον τρόπο με τον οποίο έγινε η μοντελοποίηση του δικτύου, εστιάζοντας στα διαχειριζόμενα αντικείμενα τα οποία ορίσαμε. Η λογική που ακολουθήσαμε ήταν να ορίσουμε ένα διαφορετικό τύπο διαχειριζόμενου αντικειμένου για κάθε «φυσική» συνιστώσα του δικτύου (δρομολογητές, ζεύξεις, interfaces). Έτσι, ορίστηκαν τρεις κλάσεις διαχειριζόμενων

αντικειμένων. Κάθε μία από αυτές της κλάσεις είναι υποκλάση της κλάσης Network Element.

Κλάση Network Element: Η κλάση αυτή είναι υπερκλάση όλων των κλάσεων που χρησιμοποιούμε για να μοντελοποιήσουμε τα διαχειριζόμενα αντικείμενα. Ουσιαστικά κάθε διαχειριζόμενο αντικείμενο είναι ένα στοιχείο δικτύου (Network Element). Το μόνο γνώρισμα (attribute) που περιέχει η Network Element είναι το ID. Το γνώρισμα αυτό, το οποίο είναι ένα μοναδικό αναγνωριστικό, το κληρονομούν όλες οι υποκλάσεις. Για την κλάση Network Element δε δημιουργούνται στιγμιότυπα. Ο ορισμός της εν λόγω κλάσης δίνεται στη συνέχεια. Επειδή η δέσμευση ενός κόμβου στο δέντρο καταχώρησης γίνεται από μία διεθνή αρχή, εμείς συμβολικά δηλώνουμε ότι η συγκεκριμένη κλάση δεσμεύει τον κόμβο Κ.

```
NetworkElement MANAGED OBJECT CLASS
DERIVED FROM top;
DEFINED AS "Η κλάση αυτή χρησιμοποιείται ως υπερκλάση όλων των κλάσεων των
    διαχειριζόμενων αντικειμένων.";;
ATTRIBUTES
    ID GET– SET BY CREATE;;;
REGISTERED AS { K };
```

Κλάση Router: Χρησιμοποιείται για τη μοντελοποίηση των δρομολογητών. Κάθε στιγμιότυπό της χαρακτηρίζεται από ένα μοναδικό αναγνωριστικό (ID) και περιέχει ένα πίνακα με τα interfaces του συγκεκριμένου δρομολογητή (interfaces table). Επιπλέον, τα στιγμιότυπα της συγκεκριμένης κλάσης διαθέτουν το γνώρισμα sysUpTime, το οποίο περιέχει το χρόνο από την τελευταία επανεκκίνηση του δρομολογητή. Τα παραπάνω γνωρίσματα είναι αυτά του χρησιμοποιούμε για την υλοποίηση του αλγορίθμου, όμως η συγκεκριμένη κλάση περιέχει και άλλα γνωρίσματα, τα οποία φαίνονται στον ορισμό που παραθέτουμε παρακάτω. Η σημασία αυτών τον γνωρισμάτων εξηγείται στο παράρτημα, αφού ορίζονται και στο SNMP. Είναι προφανές ότι ο τρόπος που ορίσαμε το διαχειριζόμενο αντικείμενο που αντιστοιχεί στους δρομολογητές είναι συνεπής προς τον ορισμό που δίνεται στη MIB-II.

```
Router MANAGED OBJECT CLASS
DERIVED FROM NetworkElement;

DEFINED AS "Τα στιγμιότυπα της κλάσης αυτής χρησιμοποιούνται για να αντιπροσωπεύουν
    δρομολογητές ";;
ATTRIBUTES
    interfacesTable GET-REPLACE;
    ifNumber GET– SET BY CREATE;
    ifIndex GET– SET BY CREATE;
    sysDescr GET– SET BY CREATE;
    sysUpTime GET;
```

sysContact GET-REPLACE;
sysName GET– SET BY CREATE;
sysLocation GET-REPLACE;
sysServices GET– SET BY CREATE;;;

REGISTERED AS { Λ };

Κλάση Link: Χρησιμοποιείται για τη μοντελοποίηση των ζεύξεων του δικτύου. Κάθε στιγμιότυπό της χαρακτηρίζεται από ένα μοναδικό αναγνωριστικό (ID). Επιπλέον, κάθε στιγμιότυπο της συγκεκριμένης κλάσης περιέχει γνωρίσματα στα οποία αποθηκεύονται τα αναγνωριστικά των δρομολογητών και των interfaces που βρίσκονται σε κάθε άκρο της ζεύξης (router1, router2, interface1, interface2). Τα γνωρίσματα αυτά χρησιμοποιούνται στον αλγόριθμο που θα προτείνουμε. Επιπλέον, ορίσαμε ένα ακόμα γνώρισμα – το οποίο όμως δε χρησιμοποιούμε στην υλοποίηση του αλγορίθμου – το οποίο δηλώνει αν η ζεύξη λειτουργεί ή όχι. Πρόκειται για το γνώρισμα linkState. Όπως έχουμε ήδη πει, στη MIB-II δεν ορίζεται διαχειριζόμενο αντικείμενο για τις ζεύξεις. Εμείς επιλέξαμε να ορίσουμε ένα ξεχωριστό αντικείμενο για τις ζεύξεις, ώστε να έχουμε έμμεσα αποθηκευμένη την τοπολογία του δικτύου (μέσω των γνωρισμάτων που αποθηκεύουν πού βρίσκονται τα άκρα της ζεύξης). Ουσιαστικά, ορίζουμε μία ιδιωτική MIB, μία δυνατότητα που ούτως ή άλλως δίνει το SNMP. Αξίζει να σημειώσουμε ότι δεν προσθέσαμε επιπλέον γνωρίσματα στην κλάση Link (όπως το sysUpTime που ορίζεται για τις ζεύξεις, όπως είδαμε, στο μοντέλο διαχείρισης OSI), που θα έδιναν δυνατότητες εξαγωγής καλύτερων αποτελεσμάτων, επειδή σκοπός μας είναι ο αλγόριθμος που θα προτείνουμε να μπορεί να εφαρμοστεί σε ένα δίκτυο για τη διαχείριση του οποίου χρησιμοποιείται το SNMP και η MIB-II.

Link MANAGED OBJECT CLASS
DERIVED FROM NetworkElement;

DEFINED AS "Τα στιγμιότυπα της κλάσης αυτής χρησιμοποιούνται για να αντιπροσωπεύουν ζεύξεις ";;
ATTRIBUTES
router1 GET-REPLACE;
router2 GET-REPLACE;
interface1 GET-REPLACE;
interface2 GET-REPLACE;
linkState GET;;;

REGISTERED AS { M };

Κλάση Interface: Χρησιμοποιείται για τη μοντελοποίηση των interfaces του δικτύου. Κάθε στιγμιότυπο αυτής της κλάσης χαρακτηρίζεται από ένα μοναδικό αναγνωριστικό (ID). Επιπλέον, διαθέτει ένα γνώρισμα στο οποίο δηλώνεται το αναγνωριστικό του δρομολογητή στον οποίο ανήκει (router), ένα γνώρισμα στο οποίο

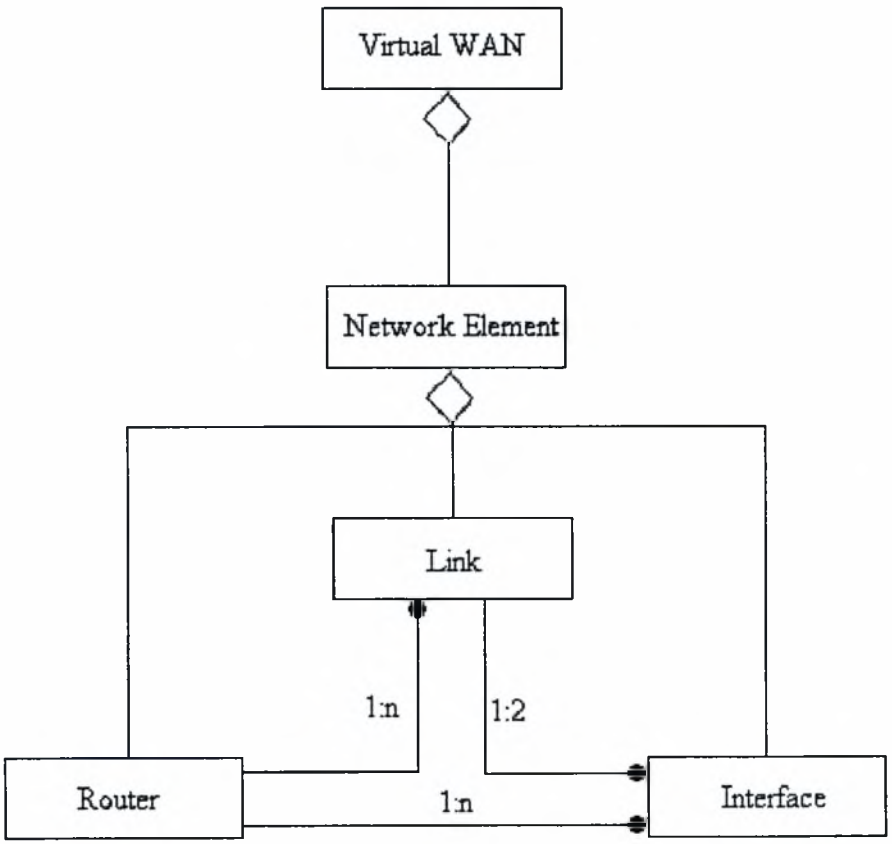
δηλώνεται ποια ζεύξη έχει άκρο σε αυτό το interface (link), καθώς επίσης και ένα γνώρισμα στο οποίο δηλώνεται αν το interface λειτουργεί ή όχι (ifOperStatus). Τα παραπάνω γνωρίσματα χρησιμοποιήθηκαν για να υλοποιήσουμε τον αλγόριθμο. Η κλάση βέβαια περιέχει και άλλα γνωρίσματα τα οποία ορίζονται και στη MIB-II και φαίνονται στον ορισμό που δίνουμε παρακάτω. Η σημασία τους εξηγείται στο παράρτημα. Βλέπουμε ότι ο ορισμός του διαχειριζόμενου αντικειμένου για τα interfaces είναι σε γενικές γραμμές συνεπής με αυτόν που δίνεται στη MIB-II. Οι μόνες διαφορές που υπάρχουν είναι ότι στη MIB-II η προσπέλαση των interfaces γίνεται μόνο μέσω του δρομολογητή στον οποίο ανήκουν, ενώ εμείς μπορούμε να τα προσπελάσουμε σαν τελείως ανεξάρτητα αντικείμενα, και ότι υπάρχει γνώρισμα που δηλώνει ποια ζεύξη έχει το ένα άκρο της στο συγκεκριμένο interface.

```
Interface MANAGED OBJECT CLASS
DERIVED FROM NetworkElement;

DEFINED AS "Τα σπγμιότυπα της κλάσης αυτής χρησιμοποιούνται για να αντιπροσωπεύουν
      interfaces ";;
ATTRIBUTES
      router GET– SET BY CREATE;
      link GET– SET BY CREATE;
      ifDescr GET– SET BY CREATE;
      ifType GET– SET BY CREATE;
      ifMtu GET;
      ifSpeed GET;
      ifPhysAddress GET;
      ifAdminStatus GET-REPLACE;
      ifOperStatus GET;
      ifLastChange GET;
      ifInOctets GET;
      ifInUcastPkts GET;
      ifInNUcastPkts GET;
      ifInDiscards GET;
      ifInErrors GET;
      ifInUnknownProtos GET;
      ifOutOctets GET;
      ifOutUcastPkts GET;
      ifOutNUcastPkts GET;
      ifOutDiscards GET;
      ifOutErrors GET;
      ifOutQLen GET;
      ifSpecific GET– SET BY CREATE;;;

REGISTERED AS { N };
```

Τα μηνύματα που παράγονται όταν παρουσιαστεί ή επισκευαστεί κάποια βλάβη σε ένα στοιχείο του δικτύου βρίσκονται σε απόλυτή συνέπεια με αυτά που παράγονται στο SNMP, όπως περιγράφηκαν σε προηγούμενο κεφάλαιο. Τα μηνύματα αυτά είναι το linkDown, το linkUp και το start (δηλώνουμε με το ίδιο μήνυμα και το cold και το warm start).

Στο σχήμα 4.1 παρουσιάζονται οι σχέσεις που υπάρχουν ανάμεσα στις συνιστώσες του συστήματος που φτιάξαμε. Ο ρόμβος στο συγκεκριμένο σχήμα χρησιμοποιείται για να δηλώσει συλλογή (aggregation) και ο μαύρος κύκλος σχέση από ένα προς πολλά. Οι πληθικότητες των σχέσεων αυτών δηλώνονται και πάνω στο σχήμα. Έτσι, το 1:2 δηλώνει μία σχέση από ένα προς δύο, ενώ η σχέση 1:n μία σχέση από ένα προς n, όπου n ένας θετικός ακέραιος.

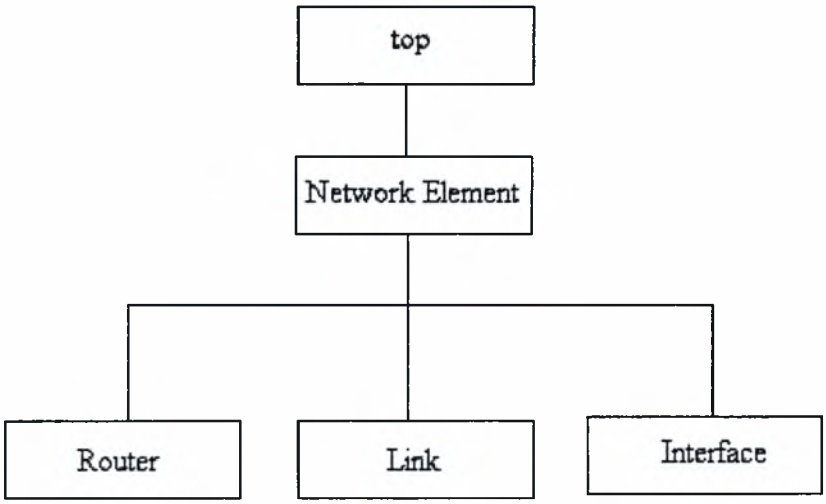
Ερμηνεύοντας το σχήμα, μπορούμε να πούμε ότι το εικονικό μας δίκτυο (ή στη γενική περίπτωση ένα οποιοδήποτε δίκτυο) αποτελείται από μία συλλογή στοιχείων. Τα στοιχεία αυτά είναι δρομολογητές, ζεύξεις και interfaces. Κάθε δρομολογητής έχει n interfaces και n ζεύξεις. Επιπλέον, κάθε ζεύξη έχει τα άκρα της σε δύο interfaces.



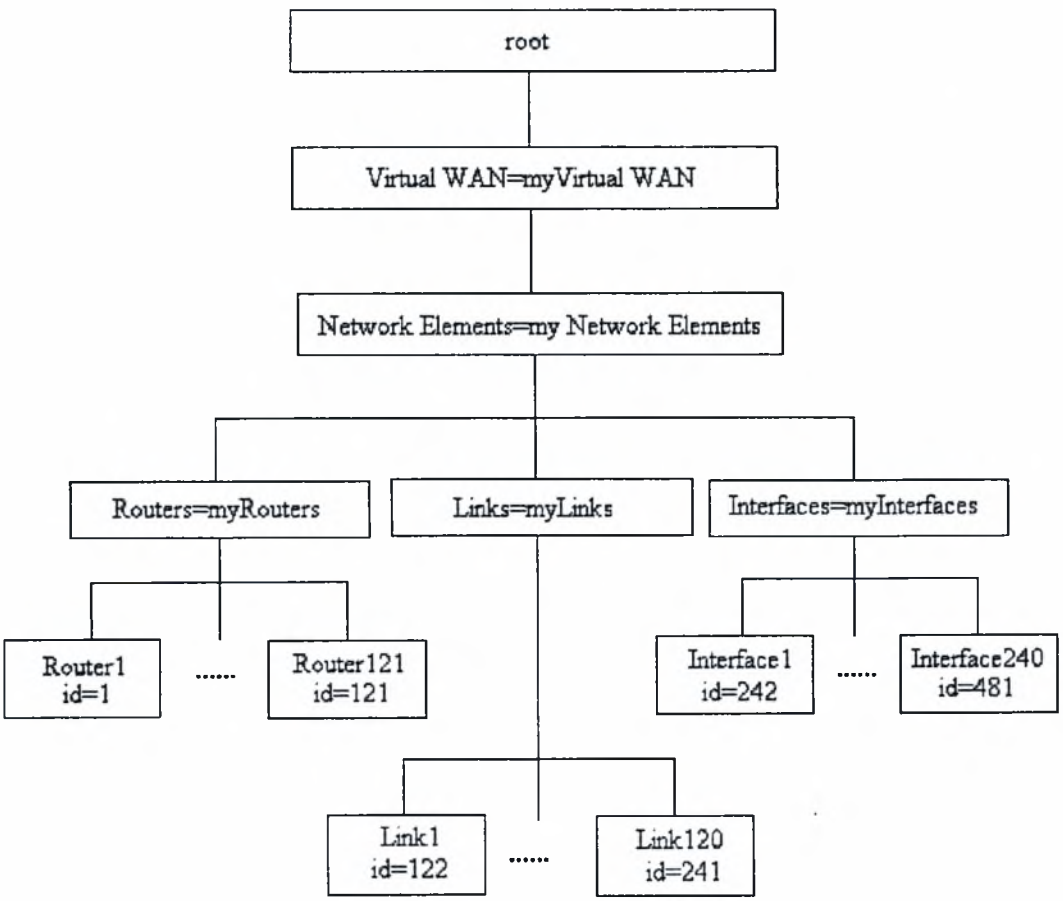
Σχήμα 4.1: *Σχέσεις ανάμεσα στις συνιστώσες του συστήματος*

Όπως είπαμε, για να φτιάξουμε τις κλάσεις των διαχειριζόμενων αντικειμένων ορίσαμε αρχικά την κλάση Network Element. Η κλάση αυτή δε χρησιμοποιείται για να δώσει στιγμιότυπα, ενώ το μόνο γνώρισμα που περιέχει είναι το ID. Στη συνέχεια επεκτείναμε την κλάση Network Element για να πάρουμε τις τρεις υποκλάσεις που περιγράψαμε λεπτομερώς παραπάνω: την υποκλάση Router, την υποκλάση Link και την υποκλάση Interface. Όπως είδαμε, και οι τρεις υποκλάσεις περιέχουν το

γνώρισμα ID, καθώς το κληρονομούν από τη Network Element. Στο σχήμα 4.2 παρουσιάζεται το δέντρο κληρονομικότητας για τις κλάσεις που χρησιμοποιήσαμε.



Σχήμα 4.2: *Δέντρο κληρονομικότητας*



Σχήμα 4.3: *Δέντρο ονοματολογίας*

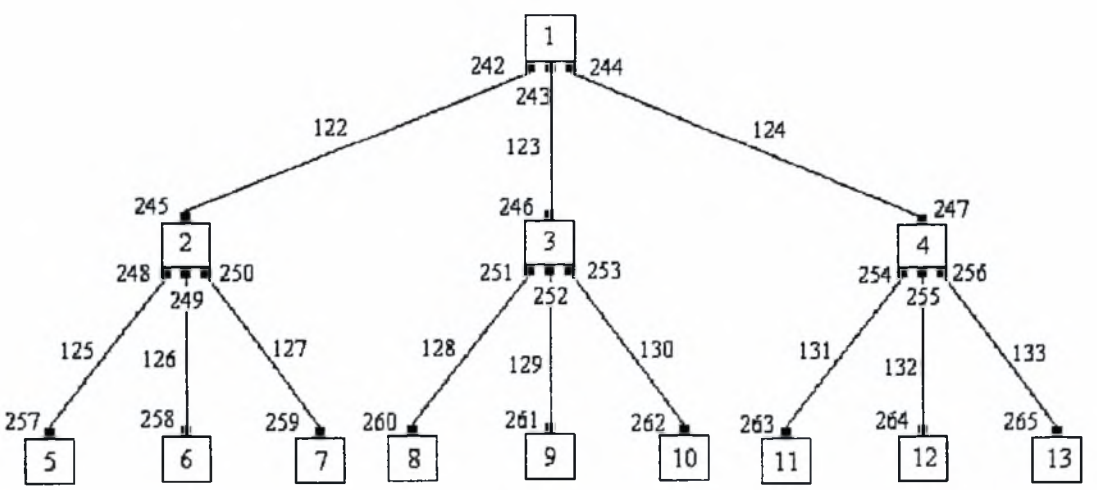
Στο σχήμα 4.3 παρουσιάζεται το δέντρο ονοματολογίας για το δίκτυο που μοντελοποιήσαμε. Αυτό που κατασκευάσαμε είναι ένα εικονικό δίκτυο ευρείας περιοχής που ονομάζουμε myVirtual WAN. Το δίκτυο αυτό αποτελείται από (εικονικά) στοιχεία δικτύου που ονομάζονται myNetwork Elements. Τα στοιχεία αυτά είναι δρομολογητές, ζεύξεις και interfaces, τα οποία ονομάζουμε myRouters, myLinks και myInterfaces αντίστοιχα. Τέλος, έχουμε 121 δρομολογητές, 120 ζεύξεις και 240 interfaces. Καθένα από αυτά τα στοιχεία έχει ένα μοναδικό αναγνωριστικό.

4.2 Τοπολογία εικονικού δικτύου

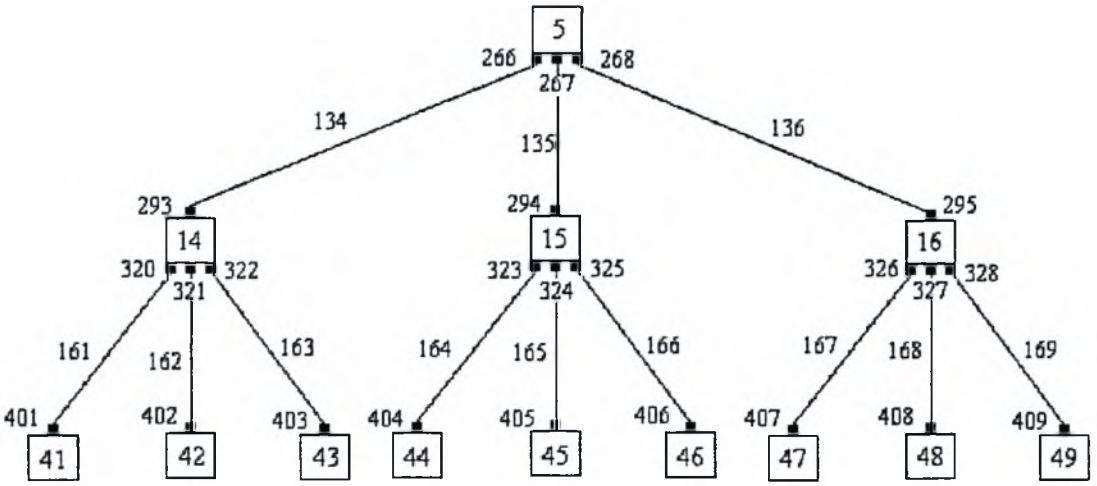
Στην ενότητα αυτή θα παρουσιάσουμε την τοπολογία του εικονικού δικτύου που κατασκευάσαμε. Οι δρομολογητές, οι ζεύξεις και τα interfaces του δικτύου αυτού είναι στιγμιότυπα των κλάσεων που παρουσιάστηκαν στην προηγούμενη ενότητα.

Στο σχήμα 4.4 παρουσιάζεται η πλήρης τοπολογία του δικτύου, το οποίο είναι στην ουσία ένα τριαδικό δέντρο με πέντε επίπεδα. Σε κάθε δρομολογητή, ζεύξη και interface έχει ανατεθεί ένας ακέραιος, ο οποίος αποτελεί ένα μοναδικό αναγνωριστικό. Πρόκειται για το γνώρισμα ID, στο οποίο αναφερθήκαμε όταν αναλύσαμε τις κλάσεις που ορίσαμε για τα διαχειριζόμενα αντικείμενα. Λόγω προβλήματος χώρου, παρουσιάζουμε το δίκτυο σε κομμάτια. Πιο συγκεκριμένα στο σχήμα 4.4.1 παρουσιάζονται τα τρία ανώτερα στρώματα της ιεραρχίας του δικτύου και στα σχήματα 4.4.2 ως 4.4.10 παρουσιάζονται τα υποδέντρα που ξεκινούν από τους κόμβους που βρίσκονται στο χαμηλότερο επίπεδο της ιεραρχίας του σχήματος 4.4.1.

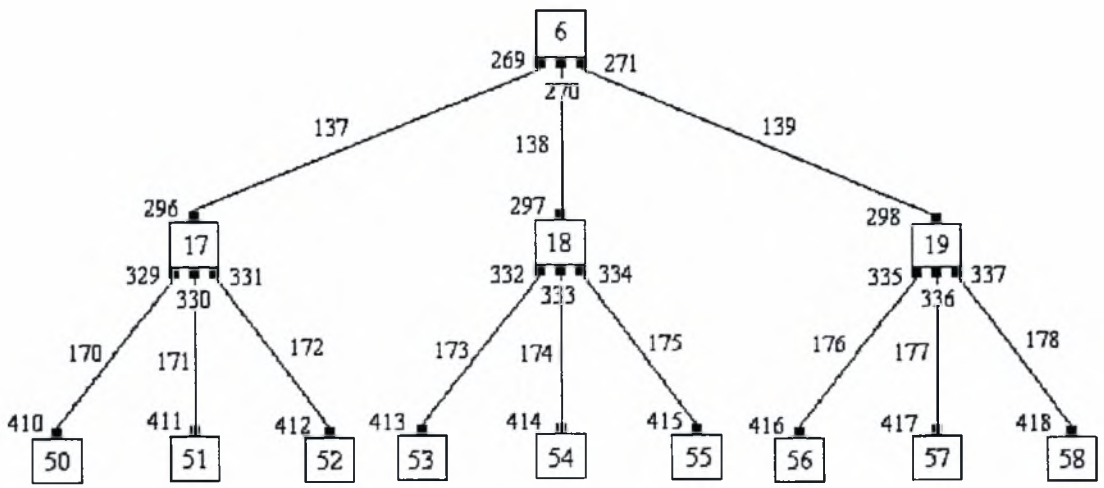
Η τοπολογία του δικτύου δεν επιλέχθηκε τυχαία. Στην πραγματικότητα βρίσκεται σε πλήρη αναλογία με τα υποδέντρα που σχηματίζει το δίκτυο περιοχής και καταστημάτων της επιχείρησης στην οποία έχουμε αναφερθεί. Τα υποδέντρα αυτά έχουν 4-5 επίπεδα και συνδέονται πάνω στο δίκτυο κορμού, το οποίο σχηματίζει πλέγμα. Υποθέτουμε ότι ο διαχειριστής βρίσκεται στο δρομολογητή με ID 1 και ότι ο δρομολογητής αυτός δεν παθαίνει βλάβες.



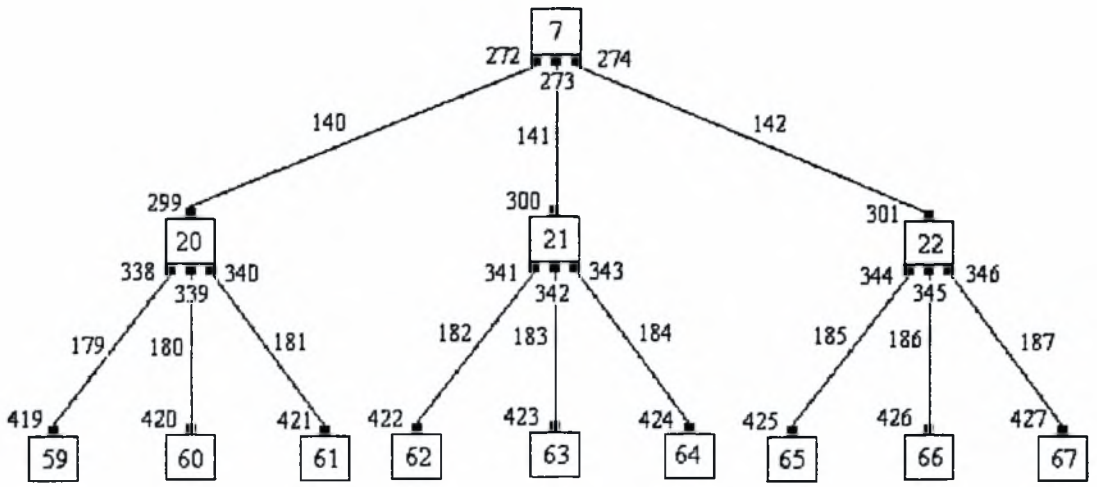
Σχήμα 4.4.1: *Ανώτερα επίπεδα δικτύου*



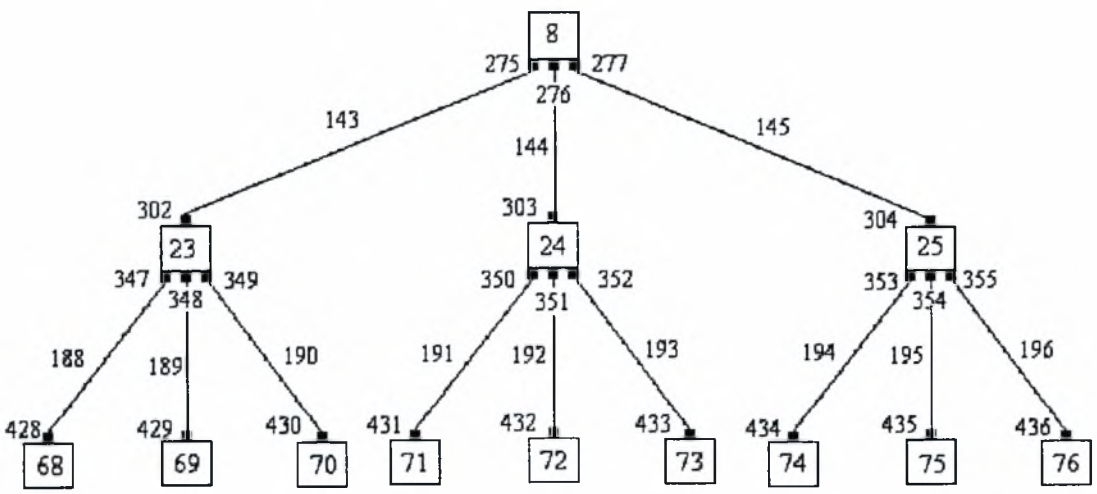
Σχήμα 4.4.2: *Υποδέντρο κόμβου 5*



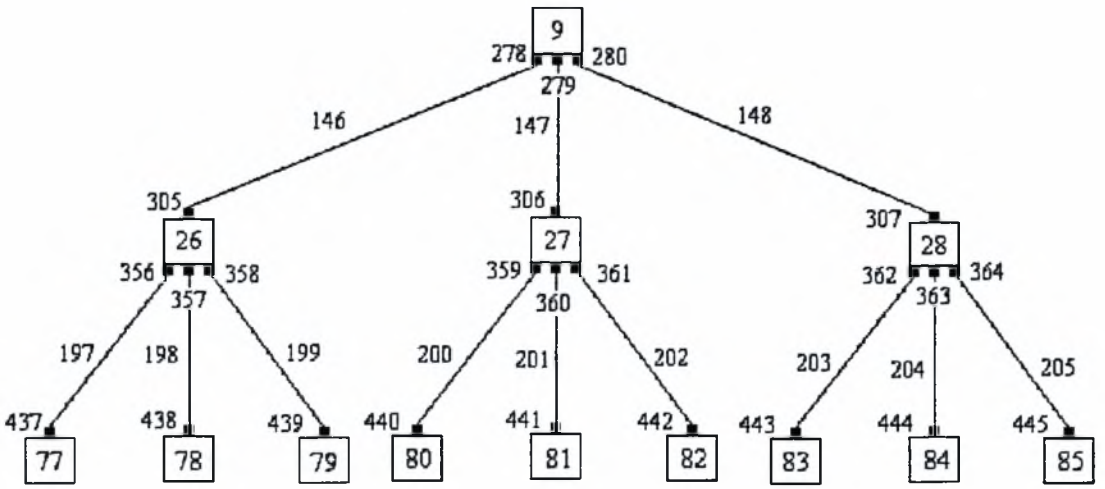
Σχήμα 4.4.3: *Υποδέντρο κόμβου 6*



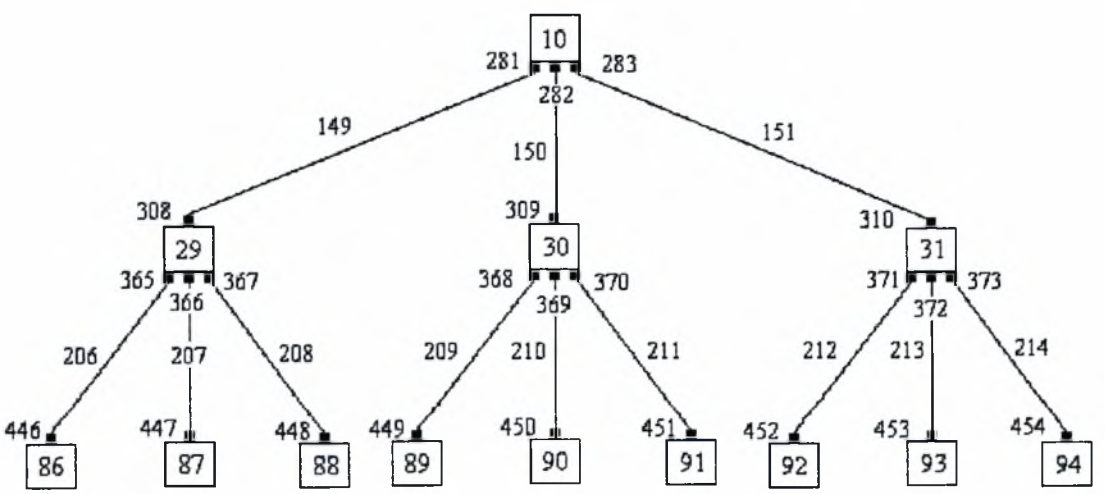
Σχήμα 4.4.4: *Υποδέντρο κόμβου 7*



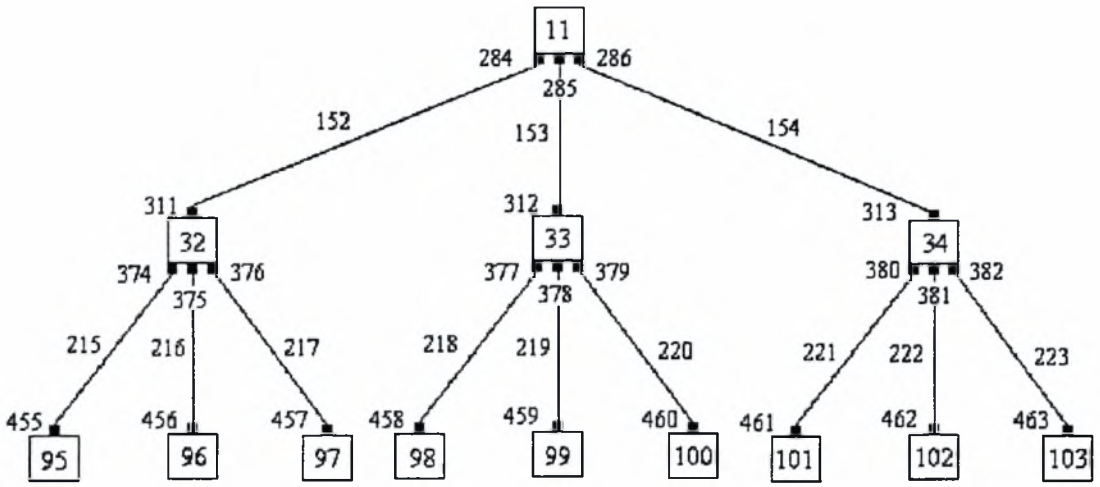
Σχήμα 4.4.5: *Υποδέντρο κόμβου 8*



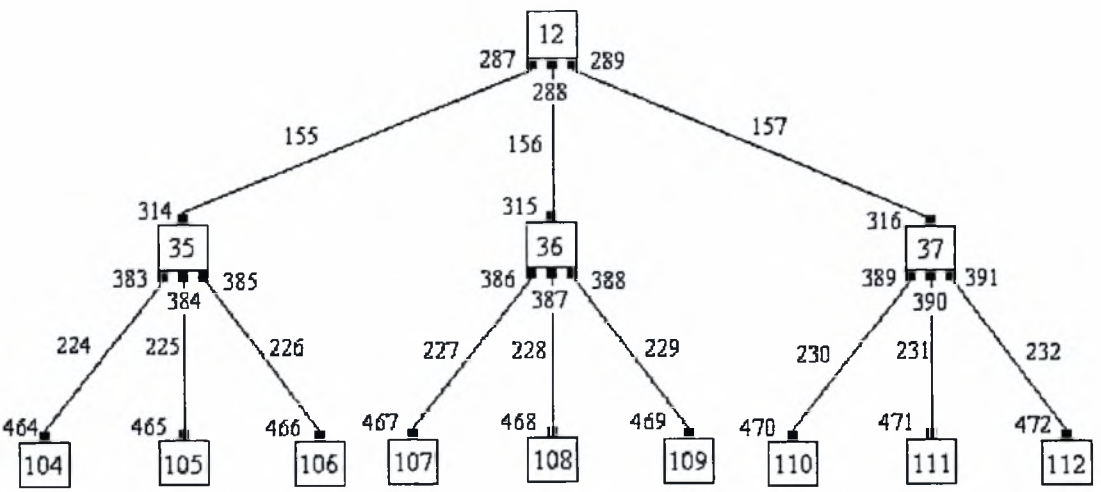
Σχήμα 4.4.6: *Υποδέντρο κόμβου 9*



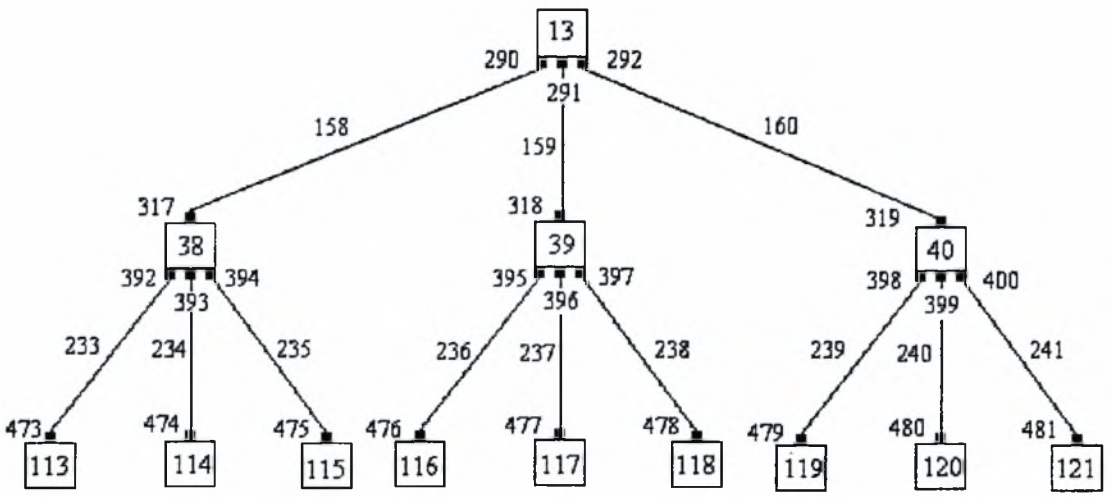
Σχήμα 4.4.7: *Υποδέντρο κόμβου 10*



Σχήμα 4.4.8: *Υποδέντρο κόμβου 11*



Σχήμα 4.4.9: *Υποδέντρο κόμβου 12*



Σχήμα 4.4.10: *Υποδέντρο κόμβου 13*

Στην ενότητα 4.4 θα παρουσιάσουμε τον αλγόριθμο για τον εντοπισμό βλαβών σε δίκτυα με δενδροειδή τοπολογία, σαν αυτό που παρουσιάσαμε παραπάνω. Ο αλγόριθμος, για να συσχετίσει τα διάφορα γεγονότα, χρησιμοποιεί συλλογιστική που βασίζεται σε κανόνες. Στην επόμενη ενότητα (4.3) θα παρουσιάσουμε μερικές βασικές τεχνικές για τη συσχέτιση γεγονότων, συμπεριλαμβανόμενης και της συλλογιστικής που βασίζεται σε κανόνες.

4.3 Τεχνικές συσχέτισης γεγονότων [1], [8]

Όταν ένα κεντρικό σύστημα διαχείρισης λαμβάνει ένα trap ή ένα notification λέμε ότι ενημερώνεται για ένα γεγονός. Το πρόβλημα είναι ότι δεν υπάρχει μία ένα προς ένα σχέση ανάμεσα σε ένα συγκεκριμένο σύμπτωμα, το οποίο καταγράφεται σα γεγονός, και στην αιτία που το προκαλεί. Με άλλα λόγια μπορεί ένα πρόβλημα σε ένα δίκτυο να προκαλεί μία πληθώρα συμπτωμάτων και αντίστοιχα ένα σύμπτωμα να προκαλείται από περισσότερα του ενός προβλήματα. Το ζητούμενο σε κάθε περίπτωση είναι να εντοπίσουμε το πρόβλημα που υπάρχει στο δίκτυο και να το απομονώσουμε. Προφανώς, αν θεωρήσουμε ότι κάθε γεγονός που καταγράφεται στο σύστημα διαχείρισης είναι ασυσχέτιστο με τα άλλα, μπορούμε να οδηγηθούμε σε λανθασμένα συμπεράσματα. Για το λόγο αυτό θέλουμε το ίδιο το σύστημα διαχείρισης να είναι σε θέση να συσχετίζει τα διάφορα γεγονότα που λαμβάνει και να εντοπίζει τη ρίζα του προβλήματος. Οι μέθοδοι που χρησιμοποιούνται για το σκοπό αυτό ονομάζονται *τεχνικές συσχέτισης γεγονότων*. Στην παρούσα ενότητα θα αναφέρουμε μερικές από τις πιο σημαντικές από αυτές.

Συλλογιστική βασισμένη σε κανόνες

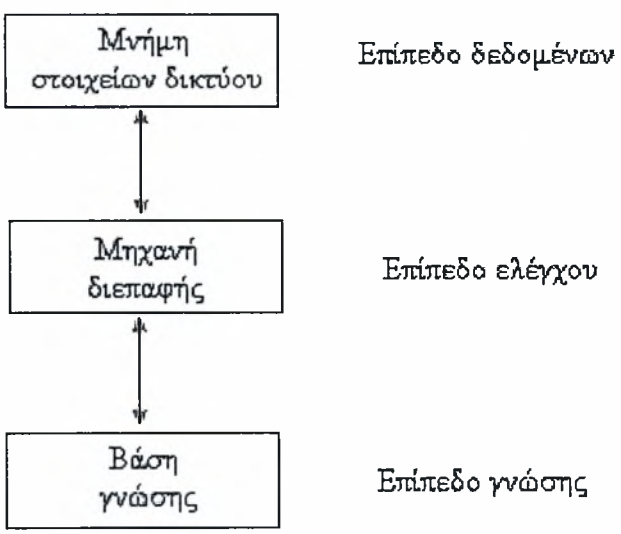
Η συλλογιστική που βασίζεται σε κανόνες (Rule Based Reasoning ή RBR) είναι η πρώτη τεχνική που χρησιμοποιήθηκε για το συσχετισμό γεγονότων. Τα συστήματα που χρησιμοποιούν την τεχνική αυτή είναι γνωστά με αρκετά ονόματα, όπως συστήματα βασισμένα σε κανόνες, έμπειρα συστήματα βασισμένα σε κανόνες ή απλώς έμπειρα συστήματα. Τα συστήματα αυτά αποτελούνται από τρεις βασικές συνιστώσες.

Η πρώτη συνιστώσα είναι η βάση γνώσης. Η γνώση στη βάση αυτή είναι αποθηκευμένη με τη μορφή κανόνων *if συνθήκη then δράση*. Αυτός είναι και ο λόγος που η συγκεκριμένη τεχνική αναφέρεται ως συλλογιστική βασισμένη σε κανόνες.

Η δεύτερη συνιστώσα των συστημάτων που βασίζονται σε κανόνες είναι η *μνήμη με τα στοιχεία του δικτύου*. Τη συνιστώσα αυτή μπορούμε να την παρομοιάσουμε με τη MIB, αφού πέρα από την τοπολογία του δικτύου περιέχει και την κατάσταση των στοιχείων του. Η μνήμη των στοιχείων του δικτύου είναι επίσης σε θέση να αναγνωρίζει πότε το δίκτυο έχει περάσει σε μία κατάσταση που δεν είναι επιθυμητή.

Η τρίτη συνιστώσα είναι η μηχανή διεπαφής, η οποία παίζει το ρόλο του μεσάζοντα ανάμεσα στη βάση γνώσης και τη μνήμη με τα στοιχεία του δικτύου. Πιο συγκεκριμένα η μηχανή διεπαφής ελέγχει την κατάσταση στην οποία βρίσκονται τα στοιχεία του δικτύου και τη συγκρίνει με τους κανόνες της βάσης γνώσης. Όταν εκπληρώνεται η συνθήκη ενός κανόνα, τότε ο κανόνας αυτός εκτελείται και πραγματοποιείται η δράση που προβλέπει πάνω στα στοιχεία του δικτύου. Η δράση αυτή μπορεί να προκαλέσει ένα νέο γεγονός κ.ο.κ. Η όλη διαδικασία συνεχίζεται μέχρι η μνήμη των στοιχείων του δικτύου να διαπιστώσει ότι το δίκτυο έχει επανέλθει σε μία επιθυμητή κατάσταση.

Στο σχήμα 4.5 φαίνονται οι συνιστώσες ενός συστήματος που βασίζεται σε κανόνες. Βλέπουμε ότι η μνήμη με τα στοιχεία του δικτύου βρίσκεται στο επίπεδο δεδομένων, αφού περιέχει πληροφορία για την τρέχουσα κατάσταση του δικτύου. Η μηχανή διεπαφής με τη σειρά της βρίσκεται στο επίπεδο ελέγχου, αφού εκτελεί τον έλεγχο των κανόνων. Τέλος η βάση γνώσης, η οποία περιέχει τους κανόνες, βρίσκεται στο επίπεδο γνώσης.



Σχήμα 4.5: *Οι συνιστώσες ενός συστήματος που βασίζεται σε κανόνες*

Η δημιουργία των κανόνων που περιέχονται στη βάση γνώσης δεν είναι μία απλή υπόθεση. Στην πραγματικότητα, για να φτιαχτούν οι κανόνες αυτοί θα πρέπει να «καταθέσουν» τη γνώση και την εμπειρία τους άνθρωποι που ασχολούνται χρόνια με τη διαχείριση δικτύων. Εναλλακτικοί τρόποι για τη δημιουργία κανόνων είναι οι προσομοιώσεις, οι αναλυτικές-μαθηματικές μέθοδοι και η παρατήρηση-ανάλυση των δεδομένων που έχουμε από το δίκτυο.

Επίσης, ένα άλλο πρόβλημα που υπάρχει με τους κανόνες είναι ότι τα όρια στις συνθήκες τους είναι πολύ σαφή. Αυτό ίσως να φαίνεται παράλογο, για αυτό θα το εξηγήσουμε μέσα από ένα παράδειγμα. Έστω οι κανόνες:

if απώλεια πακέτων < 5% then κατάσταση φυσιολογική
if απώλεια πακέτων >=5% then συναγερμός

Οι κανόνες αυτοί θέτουν ένα σαφές όριο για το πότε πρέπει να γίνει κάτι. Το όριο αυτό είναι το 5%. Αν έχουμε απώλεια πακέτων της τάξης του 4,9%, τότε, σύμφωνα με τους κανόνες, είμαστε σε μία φυσιολογική κατάσταση. Αντίθετα αν έχουμε απώλεια πακέτων της τάξης του 5,1% πρέπει να σημάνει συναγερμός. Βέβαια, το 5,1% είναι πολύ κοντά με το 4,9% και αυτό δηλώνει ότι ίσως θα έπρεπε να δηλωθεί μία μεταβατική κατάσταση μεταξύ των δύο ακραίων. Επίσης, είναι πιθανόν η απώλεια πακέτων να κινείται συνεχώς γύρω από το 5%, άλλοτε πιο πάνω και άλλοτε πιο κάτω, «τρελαίνοντας» το σύστημα. Μία λύση θα ήταν η προσθήκη ενός επιπλέον κανόνα. Για παράδειγμα θα μπορούσαμε να έχουμε:

if απώλεια πακέτων < 3% then κατάσταση φυσιολογική
if απώλεια πακέτων >=3 and < 5% then κίτρινος συναγερμός
if απώλεια πακέτων >=5% then κόκκινος συναγερμός

Βέβαια, πρέπει να σημειώσουμε ότι η συνεχής προσθήκη κανόνων αυξάνει πολύ την πολυπλοκότητα των ελέγχων που κάνει το σύστημα, μειώνοντας την απόδοσή του. Επιπλέον, κάποιος θα μπορούσε να υποστηρίξει ότι το πρόβλημα που αντιμετωπίζαμε πριν, μεταφέρθηκε πλέον στα άκρα της συνθήκης του νέου κανόνα. Για την αντιμετώπιση των προαναφερθέντων προβλημάτων έχουν προταθεί λύσεις που στηρίζονται στην ασαφή λογική.

Συλλογιστική βασισμένη σε περιπτώσεις

Η συλλογιστική που βασίζεται σε περιπτώσεις (case-based reasoning ή CBR) κατάφερε να καλύψει κάποιες από τις αδυναμίες της συλλογιστική που βασίζεται σε κανόνες. Στη δεύτερη τεχνική η μονάδα γνώσης είναι ένας απλός κανόνας, ενώ στην πρώτη είναι μία ολόκληρη περίπτωση. Η λογική πίσω από τη συλλογιστική που βασίζεται σε περιπτώσεις είναι ότι στον πραγματικό κόσμο κάποιες καταστάσεις επαναλαμβάνονται. Ο όρος επανάληψη αναφέρεται και στην περίπτωση που μία κατάσταση εμφανίζεται ξανά με τον ίδιο ακριβώς τρόπο, αλλά και στην περίπτωση που εμφανίζεται κάπως παραλλαγμένη. Στις καταστάσεις που επαναλαμβάνονται συμπεριλαμβάνονται φυσικά και βλάβες που παρουσιάζονται σε δίκτυα.

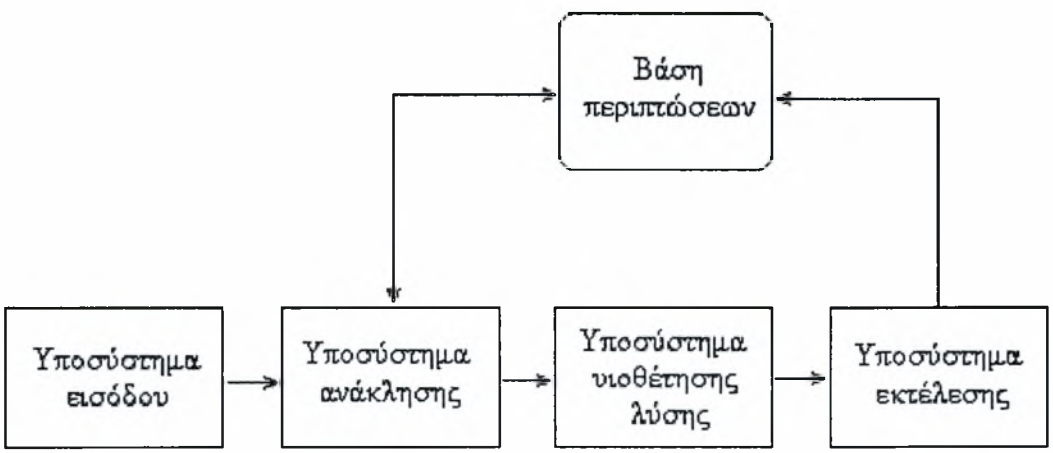
Στο σχήμα 4.6 παρουσιάζεται η αρχιτεκτονική των συστημάτων που χρησιμοποιούν τη λογική που βασίζεται στις περιπτώσεις. Βλέπουμε ότι τα συστήματα αυτά αποτελούνται από 5 βασικά υποσυστήματα: τη βάση περιπτώσεων, το υποσύστημα εισόδου, το υποσύστημα ανάκλησης, το υποσύστημα υιοθέτησης λύσης και το υποσύστημα εκτέλεσης.

Στη *βάση περιπτώσεων* βρίσκεται αποθηκευμένη η γνώση του συστήματος, με τη μορφή περιστατικών βλαβών που συνέβησαν στο παρελθόν και των ενεργειών που έγιναν για την αποκατάστασή τους.

Το *υποσύστημα εισόδου* χρησιμοποιείται για να ληφθεί η παρούσα κατάσταση του δικτύου. Στη συνέχεια το *υποσύστημα ανάκλησης* ανακαλεί από τη βάση περιπτώσεων τις παρελθούσες περιπτώσεις βλαβών που μοιάζουν περισσότερο με την τρέχουσα κατάσταση του δικτύου. Ο ορισμός των κριτηρίων για να ληφθεί η απόφαση για το ποιες περιπτώσεις μοιάζουν με την τρέχουσα, μπορεί να είναι μία αρκετά περίπλοκη υπόθεση.

Η τελική απόφαση για το ποια περίπτωση θα ληφθεί ως πρότυπο για να επιλυθεί το τρέχον πρόβλημα παίρνεται από το *υποσύστημα υιοθέτησης λύσης*. Το υποσύστημα αυτό είναι υπεύθυνο να κάνει και τροποποίηση της λύσης που είχε δοθεί στο παρελθόν, ώστε να ταιριάζει περισσότερο με τα νέα δεδομένα.

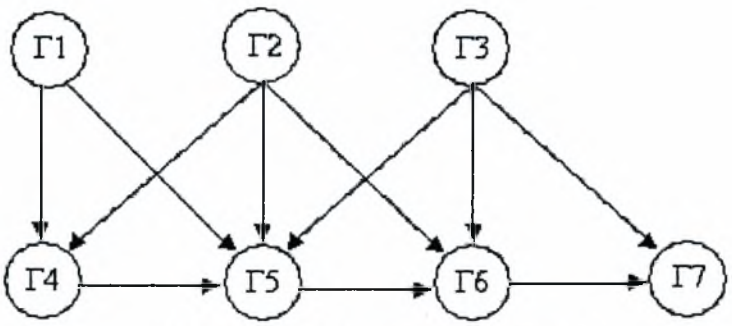
Τελικά, η προτεινόμενη λύση υλοποιείται από το *υποσύστημα εκτέλεσης*. Το υποσύστημα εκτέλεσης είναι επίσης υπεύθυνο για την ενημέρωση της βάσης περιπτώσεων με τη νέα περίπτωση βλάβης.



Σχήμα 4.6: *Αρχιτεκτονική συστήματος που βασίζεται σε περιπτώσεις*

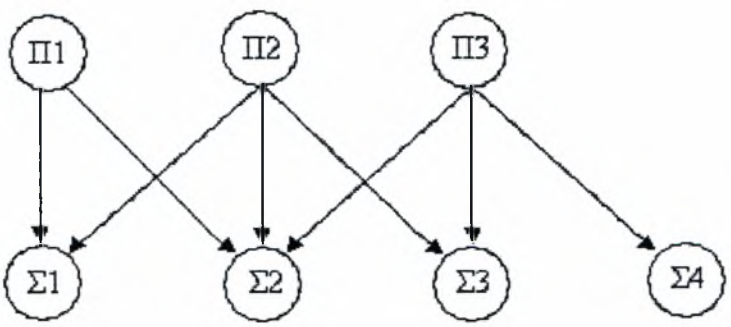
Συσχέτιση γεγονότων με βάση γράφους υπαιτιότητας

Μία από της τεχνικές που υπάρχουν για τη συσχέτιση γεγονότων είναι και αυτή που στηρίζεται στην ανάλυση με βάση γράφους υπαιτιότητας (causality graphs). Ως γράφος υπαιτιότητας ορίζεται ένας κατευθυνόμενος γράφος, κάθε κόμβος του οποίου είναι ένα γεγονός. Ο όρος γεγονός μπορεί να αναφέρεται τόσο στην αιτία ενός προβλήματος που παρουσιάστηκε στο δίκτυο, όσο και σε ένα απλό σύμπτωμα της βλάβης. Η κατεύθυνση των ακμών του γράφου δηλώνει ποιο γεγονός προκαλεί το άλλο. Πρέπει να γίνει ξεκάθαρο ότι το πραγματικό πρόβλημα σε ένα δίκτυο μπορεί να προκαλεί ένα γεγονός, το οποίο είναι πιθανόν να ανιχνεύεται ως σύμπτωμα, αλλά ταυτόχρονα μπορεί να γίνει και η αιτία για να παρουσιαστεί και άλλο ένα σύμπτωμα.



Σχήμα 4.7: *Γράφος υπαιτιότητας*

Στο σχήμα 4.7 παρουσιάζεται ένας γράφος υπαιτιότητας. Στο γράφο αυτό υπάρχουν επτά γεγονότα, τα οποία μπορεί να είναι βλάβες ή απλά συμπτώματα βλαβών. Όταν σχεδιάζουμε ένα τέτοιο γράφο οι βλάβες αντιστοιχούν στους κόμβους στους οποίους δε δείχνει καμία ακμή του γράφου. Οι υπόλοιποι κόμβοι του γράφου αντιστοιχούν σε γεγονότα που είναι αποτελέσματα των βλαβών. Όπως είπαμε ένα αποτέλεσμα μίας βλάβης μπορεί να γίνει η αιτία για τη γένεση ενός καινούριου συμπτώματος. Το πρώτο βήμα που κάνουμε για την ανάλυση με βάση ένα γράφο υπαιτιότητας είναι να σημειώσουμε ποιοι κόμβοι αντιστοιχούν σε προβλήματα και ποιοι κόμβοι αντιστοιχούν σε συμπτώματα. Στη συνέχεια, εξαλείφουμε της ακμές που αρχίζουν από ένα σύμπτωμα και καταλήγουν σε ένα άλλο σύμπτωμα, αφού δεν προσθέτουν καμία σημαντική πληροφορία. Στο σχήμα 4.8 φαίνεται ο γράφος που προκύπτει μετά την εφαρμογή της προαναφερθείσας διαδικασίας στον αρχικό γράφο του παραδείγματός μας.



Σχήμα 4.8: *Απλοποιημένος γράφος υπαιτιότητας*

Παρατηρούμε ότι έχουμε τρία προβλήματα (Π1, Π2, Π3) και τέσσερα συμπτώματα (Σ1, Σ2, Σ3, Σ4). Ο γράφος αυτός μπορεί να κωδικοποιηθεί στον πίνακα 4.1, όπου το προβλήματα αντιστοιχούν στις γραμμές και τα συμπτώματα στις στήλες.

	Σ1	Σ2	Σ3	Σ4
Π1	1	1	0	0
Π2	1	1	1	0
Π3	0	1	1	1

Πίνακας 4.1: *Πίνακας απλοποιημένου γράφου υπαιτιότητας*

	Σ1	Σ2
Π1	1	0
Π2	1	1
Π3	0	1

Πίνακας 4.2: *Πίνακας συσχέτισης*

Το 1 δηλώνει ότι ένα πρόβλημα προκαλεί ένα σύμπτωμα και το 0 το αντίθετο. Αφού στόχος μας είναι να εντοπίζουμε τα προβλήματα με βάση τα παρατηρούμενα συμπτώματα, αρκεί να αντιστοιχίσουμε ένα μοναδικό σύνολο συμπτωμάτων σε κάθε πρόβλημα. Η τακτική αυτή θα μπορούσε να οδηγήσει στον πίνακα 4.2, οποίος προκύπτει από τον πίνακα 4.1 με διαγραφή στηλών. Ο πίνακας 4.2 λέγεται *πίνακας συσχέτισης* και δηλώνει ότι αν παρατηρηθούν τα Σ1 και Σ3 αιτία είναι το Π2, αν παρατηρηθεί το Σ1 και όχι το Σ3 αιτία είναι το Π1 και τέλος αν παρατηρηθεί το Σ3 και όχι το Σ1 αιτία είναι το Π3.

Συσχέτιση γεγονότων με βάση μηχανή καταστάσεων

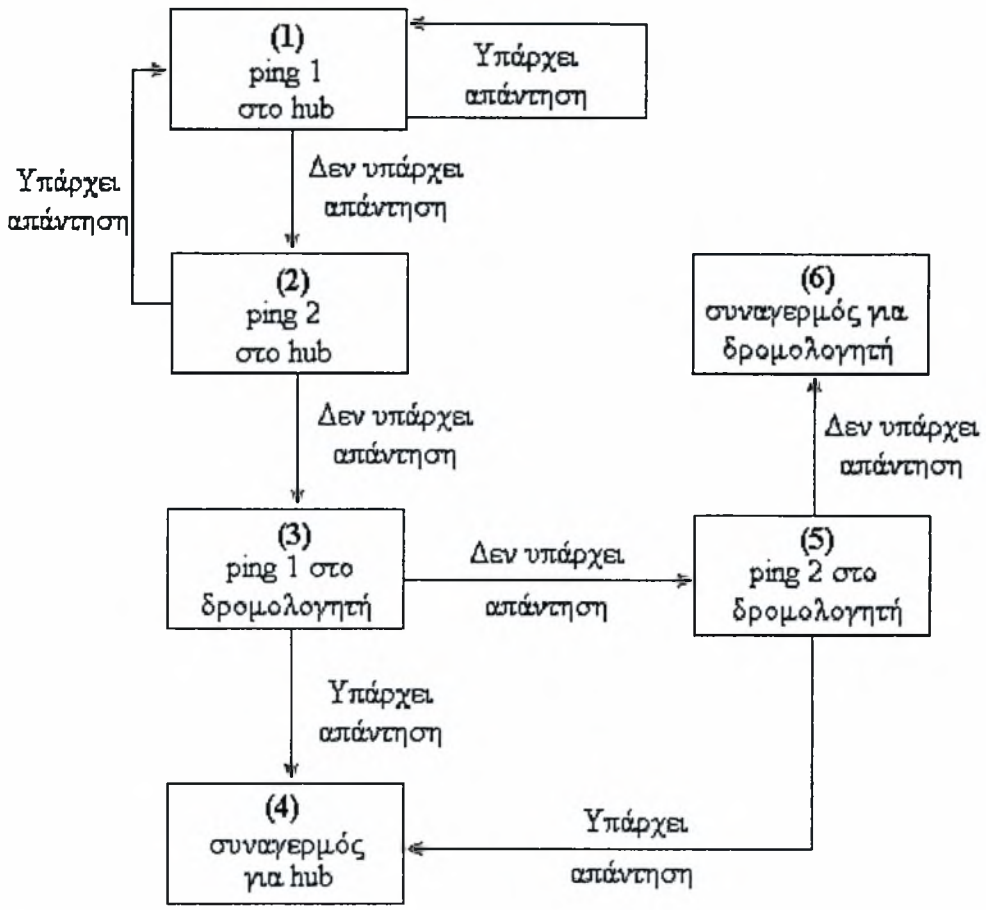
Οι μηχανές καταστάσεων είναι ένα εργαλείο το οποίο μπορεί να χρησιμοποιηθεί για το συσχετισμό γεγονότων. Στη συνέχεια θα δείξουμε τον τρόπο με τον οποίο γίνεται αυτό μέσω ενός παραδείγματος.

Ας υποθέσουμε ότι έχουμε ένα σύστημα διαχείρισης δικτύων Δ το οποίο συνδέεται με ένα hub μέσω ενός δρομολογητή. Το σύστημα διαχείρισης στέλνει μηνύματα ping στο hub κάθε τριάντα δευτερόλεπτα. Αν το hub λειτουργεί κανονικά και δεν υπάρχει πρόβλημα στη σύνδεσή του με το Δ, στέλνει μήνυμα απάντησης στο Δ (το οποίο φτάνει στο Δ το πολύ σε 2 δευτερόλεπτα από την αποστολή του αρχικού ping). Υπάρχει βέβαια και η πιθανότητα κάποιο μήνυμα ping ή ping-response να χαθεί. Στο σχήμα 4.9 παρουσιάζεται ο τρόπος που ενεργεί το Δ κατά τον έλεγχο του hub.

Στην αρχή το σύστημα διαχείρισης βρίσκεται στην κατάσταση 1 και στέλνει ένα μήνυμα ping στο hub. Αν λάβει απάντηση, παραμένει στην κατάσταση 1. Αν δε λάβει απάντηση, περνά στην κατάσταση 2. Από την κατάσταση 2 το Δ στέλνει νέο ping στο hub. Αν λάβει απάντηση, περνά στην κατάσταση 1, ενώ αν δε λάβει απάντηση, περνά στην κατάσταση 3. Από την κατάσταση 3 το Δ στέλνει μήνυμα ping στο δρομολογητή. Ο λόγος είναι ότι μπορεί το hub να λειτουργεί κανονικά, αλλά να υπάρχει πρόβλημα με το δρομολογητή, οπότε τα ping να μην φτάνουν ποτέ σε αυτό.

Αν ο δρομολογητής απαντήσει στο ping τότε περνάμε στην κατάσταση 4, όπου το Δ σημαίνει συναγερμό για μη λειτουργία του hub. Αν ο δρομολογητής δεν απαντήσει, τότε το Δ μεταβαίνει στην κατάσταση 5 και από εκεί στέλνει νέο ping στο

δρομολογητή. Αν ο δρομολογητής απαντήσει, οδηγούμαστε στην κατάσταση 4, όπου σημαίνει συναγερμός για το hub. Αν ο δρομολογητής δεν απαντήσει το Δ μεταβαίνει στην κατάσταση 6 και σημαίνει συναγερμό για μη ορθή λειτουργία του δρομολογητή.



Σχήμα 4.9: *Παράδειγμα μηχανής καταστάσεων*

4.4 Ο αλγόριθμος

Όπως έχουμε εξηγήσει και σε προηγούμενο κεφάλαιο, κάθε βλάβη ή επισκευή βλάβης έχει ως αποτέλεσμα να παράγονται τα αντίστοιχα μηνύματα (linkUp, linkDown, cold ή warm start) τα οποία αποστέλλονται στο διαχειριστή. Φυσικά, τα μηνύματα αυτά μπορεί να μη φτάσουν στο διαχειριστή αν υπάρχουν βλάβες που διακόπτουν την πορεία τους.

Στην ενότητα αυτή θα περιγράψουμε τον αλγόριθμο που εφαρμόζει ο διαχειριστής (όπως τον υλοποιήσαμε) προκειμένου να εντοπίσει τις βλάβες, με βάση τα μηνύματα που τελικά φτάνουν σε αυτόν. Θα παρουσιάσουμε δύο εκδόσεις του αλγορίθμου. Η διαφορά τους είναι ότι στη δεύτερη έκδοση ο διαχειριστής προσπαθεί να ανακαλύψει επιπλέον βλάβες στους δρομολογητές, για τις οποίες δεν έφτασαν τα αντίστοιχα μηνύματα, ενώ αντίθετα στην πρώτη η ανάλυση βασίζεται αποκλειστικά στα μηνύματα που έφτασαν στο διαχειριστή. Ουσιαστικά, όπως θα παρουσιαστεί με λεπτομέρεια στην πορεία, η δεύτερη έκδοση του αλγόριθμου αποτελεί μία επέκταση της πρώτης.

Για να γίνει κατανοητή η παρουσίαση του αλγορίθμου θα πρέπει πρώτα να αναφερθούμε σε κάποιους όρους και σύμβολα τα οποία θα χρησιμοποιήσουμε.

Καταρχάς θεωρούμε ότι ο διαχειριστής διατηρεί ένα πίνακα-διάνυσμα που έχει αριθμό θέσεων ίσο με τον αριθμό των interfaces του δικτύου και κάθε θέση αντιστοιχεί σε ένα interface. Σε κάθε θέση καταγράφεται η ώρα που στάλθηκε στο διαχειριστή το πιο πρόσφατο μήνυμα linkDown για το αντίστοιχο interface. Τον πίνακα αυτό ονομάζουμε ***LDTable***, δηλαδή LDTable[I] σημαίνει ουσιαστικά η ώρα που στάλθηκε το πιο πρόσφατο μήνυμα linkDown για το interface I.

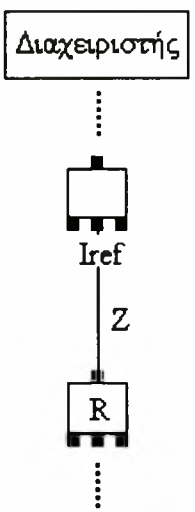
Τόσο το μήνυμα coldStart όσο και το μήνυμα warmStart θα συμβολίζονται απλά με τη λέξη ***start***.

Η ώρα που στέλνεται ένα μήνυμα start (περιέχεται στο μήνυμα) θα συμβολίζεται με ***Tstart***, ενώ η ώρα που στέλνεται ένα μήνυμα linkUp θα συμβολίζεται με ***TLinkUp***.

Οι συμβολισμοί ***TMeanRouter*** και ***TMeanLink*** χρησιμοποιούνται για να δηλώσουν το μέσο χρόνο επισκευής βλάβης για τους δρομολογητές και τις ζεύξεις αντίστοιχα. Τους χρόνους αυτούς τους εισάγουμε πριν την εκτέλεση της προσομοίωσης.

Ορίζουμε για μια ζεύξη και έναν δρομολογητή (μαζί με τα interfaces που βρίσκονται πάνω του) ως ***Iref*** (interface αναφοράς) το interface που βρίσκεται στο άνω άκρο της ζεύξης (προς την πλευρά του διαχειριστή). Για να γίνει κατανοητό αυτό, στο σχήμα 4.10 σημειώνεται το Iref για το δρομολογητή R και τη ζεύξη Z. Ο ***R*** είναι ένας οποιοσδήποτε δρομολογητής του δικτύου και η ***Z*** η αντίστοιχη ζεύξη που φαίνεται στο σχήμα.

Για το ***λογικό και*** χρησιμοποιείται το σύμβολο $\wedge$, ενώ για τη λογική άρνηση χρησιμοποιείται το σύμβολο $\neg$.



Σχήμα 4.10: *Interface αναφοράς*

Με βάση τα παραπάνω μπορούμε να παρουσιάσουμε την πρώτη έκδοση του αλγορίθμου (απλοποιημένος αλγόριθμος) με τη μορφή ψευδοκώδικα:

If έρθει start από R ∧ linkUp από Iref
  then βλάβη στον R
    if Tstart - LDTable[Iref]>3*TmeanRouter
      then διάρκεια βλάβης = 3*TmeanRouter
    else διάρκεια βλάβης = Tstart - LDTable[Iref]
    end if-else
end if

If έρθει linkUp από Iref ∧ ¬(start από R)
  then βλάβη στη Z
    if TLinkUp - LDTable[Iref]>3*TmeanLink
      then διάρκεια βλάβης = 3*TmeanLink
    else διάρκεια βλάβης =TLinkUp - LDTable[Iref]
    end if-else
end if

Στη συνέχεια παραθέτουμε μία περιγραφή του παραπάνω ψευδοκώδικα σε φυσική γλώσσα.

Αν ο διαχειριστής λάβει μήνυμα start από το δρομολογητή R και linkUp από το αντίστοιχο Iref, τότε εντοπίζεται βλάβη στον R. Η διάρκεια της βλάβης υπολογίζεται ως εξής: αν η διαφορά του χρόνου που στάλθηκε το μήνυμα start από το χρόνο που στάλθηκε το τελευταίο μήνυμα linkDown για το Iref, το οποίο παρέλαβε ο διαχειριστής, είναι μεγαλύτερη από τρεις φορές το μέσο χρόνο επιδιόρθωσης για βλάβη σε δρομολογητή, τότε η διάρκεια βλάβης θεωρείται ίση με τρεις φορές το μέσο χρόνο επιδιόρθωσης για βλάβη σε δρομολογητή. Σε διαφορετική περίπτωση ο χρόνος βλάβης θεωρείται ίσος με τη διαφορά του χρόνου που στάλθηκε το μήνυμα start από το χρόνο που στάλθηκε το τελευταίο μήνυμα linkDown για το Iref. Ο λόγος που χρησιμοποιούμε αυτή την ευρεστική μέθοδο για τον υπολογισμό της διάρκειας βλαβών θα εξηγηθεί παρακάτω.

Αν ο διαχειριστής λάβει μήνυμα linkUp από ένα interface Iref, που αποτελεί interface αναφοράς για κάποια ζεύξη Z, και δε λάβει μήνυμα start από το δρομολογητή που έχει το ίδιο Iref, τότε εντοπίζεται βλάβη στη Z. Ο χρόνος της βλάβης υπολογίζεται ως εξής: αν η διαφορά του χρόνου που στάλθηκε το μήνυμα linkUp από το χρόνο που στάλθηκε το τελευταίο μήνυμα linkDown για το Iref, το οποίο παρέλαβε ο διαχειριστής, είναι μεγαλύτερη από τρεις φορές το μέσο χρόνο επιδιόρθωσης για βλάβη σε ζεύξη, τότε η διάρκεια βλάβης θεωρείται ίση με τρεις φορές το μέσο χρόνο επιδιόρθωσης για βλάβη σε ζεύξη. Σε διαφορετική περίπτωση, ο χρόνος βλάβης θεωρείται ίσος με τη διαφορά του χρόνου που στάλθηκε το μήνυμα linkUp από το χρόνο που στάλθηκε το τελευταίο μήνυμα linkDown για το Iref.

Στο σημείο αυτό πρέπει να κάνουμε δύο παρατηρήσεις σε σχέση με τη διάρκεια των υπολογιζόμενων βλαβών.

Η πρώτη δικαιολογεί τις εντολές

if Tstart - LDTable[Iref]>3*TmeanRouter
then διάρκεια βλάβης = 3*TmeanRouter
και
if TLinkUp - LDTable[Iref]>3*TmeanLink

then διάρκεια βλάβης = 3*TmeanLink

Ο λόγος που δεν υπολογίζουμε σε όλες τις περιπτώσεις τη διάρκεια μίας βλάβης ως

διάρκεια βλάβης = Tstart - LDTable[Iref]

ή

διάρκεια βλάβης =TLinkUp - LDTable[Iref]

είναι ότι σε κάποιες περιπτώσεις μερικά μηνύματα linkDown δε φτάνουν στο διαχειριστή, λόγω βλάβης σε ανώτερο επίπεδο. Το γεγονός αυτό έχει ως αποτέλεσμα να μην είναι απόλυτα συνεπής με την πραγματικότητα ο πίνακας LDTable. Όταν χαθεί ένα μήνυμα linkDown για ένα interface αναφοράς, ο υπολογιζόμενος χρόνος διάρκειας μίας βλάβης είναι πολύ μεγάλος, καθώς ο υπολογισμός γίνεται με βάση το χρόνο που είχε σταλεί το προηγούμενο linkDown στο διαχειριστή, για το συγκεκριμένο interface. Υποθέτουμε λοιπόν ότι, όταν ο υπολογιζόμενος χρόνος επισκευής βλάβης είναι τρεις φορές μεγαλύτερος από το μέσο χρόνο επισκευής βλάβης, έχει χαθεί το μήνυμα linkDown από το interface αναφοράς και για αυτό θέτουμε τον υπολογιζόμενο χρόνο επισκευής ίσο με τρεις φορές το μέσο χρόνο επισκευής. Η παράμετρος 3*ΜέσοΧρόνοΕπισκευής αποφασίστηκε να χρησιμοποιηθεί μετά από δοκιμές που έδειξαν ότι οι υπολογιζόμενοι μέσοι χρόνοι βλαβών, με τη χρήση της συγκεκριμένης παραμέτρου, προσεγγίζουν πολύ καλά τους πραγματικούς μέσους χρόνους επισκευής βλαβών, όπως θα δούμε και σε επόμενη ενότητα.

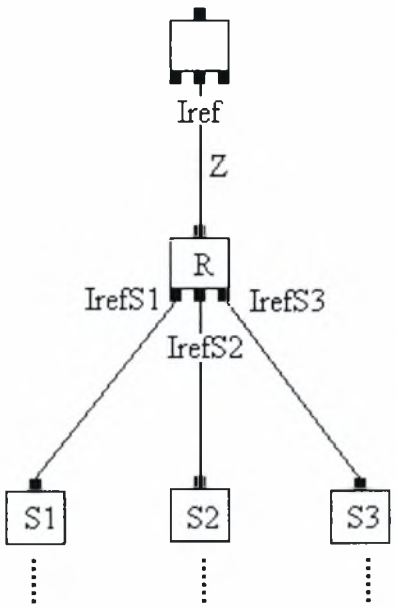
Η δεύτερη παρατήρηση έχει να κάνει με την αντιμετώπιση των δρομολογητών σαν ενιαίο σύστημα με τα interfaces που βρίσκονται πάνω τους. Η αντιμετώπιση αυτή έχει ως αναπόφευκτο αποτέλεσμα όταν υπολογίζεται ο χρόνος επισκευής βλάβης των interfaces που βρίσκονται προς το «κάτω» μέρος του R, όπως φαίνεται στο σχήμα 4.10, να μην υπολογίζεται ο πραγματικός χρόνος επισκευής βλάβης, αφού το interface αναφοράς για όλο το σύστημα του δρομολογητή είναι το Iref για το οποίο δεν στέλνεται μήνυμα linkDown, όταν σημειωθεί βλάβη στα «κάτω» interfaces του R.

Αυτό που συμβαίνει στην πραγματικότητα στη συγκεκριμένη περίπτωση είναι ότι για να αλλαχθεί το χαλασμένο interface, ο δρομολογητής πρέπει να βγει εκτός λειτουργίας (πράγμα που καταγράφεται με μήνυμα linkDown για το Iref), να γίνει η αλλαγή και στη συνέχεια να αρχίσει να λειτουργεί πάλι ο δρομολογητής (οπότε στέλνεται μήνυμα start). Είναι προφανές ότι η διαδικασία αυτή διαρκεί λιγότερο χρόνο από αυτόν που στην πραγματικότητα το interface δε λειτουργεί. Στην προσομοίωση υποθέτουμε ότι η συγκεκριμένη περίπτωση επισκευής βλάβης διαρκεί περίπου 15 λεπτά.

Παρόλα αυτά, αν το μεσοδιάστημα για να καεί δύο φορές ένα interface είναι σημαντικά μεγαλύτερο από το μεσοδιάστημα μεταξύ δύο προβλημάτων διαθεσιμότητας σε ένα δρομολογητή (π.χ. λόγω διακοπής του ηλεκτρικού), πράγμα που ισχύει, τότε, όπως θα δούμε, η προσομοίωση επιστρέφει μέσους χρόνους επισκευής βλαβών πολύ κοντά στους πραγματικούς.

Στο σημείο αυτό μπορούμε να παρουσιάσουμε τη δεύτερη έκδοση του αλγορίθμου. Τα σύμβολα που θα χρησιμοποιήσουμε είναι ίδια με πριν. Επιπλέον,

ορίζουμε τους «γιους» του R ως ***Si*** (i=1, 2, 3), όπως φαίνεται στο σχήμα 4.11, και ονομάζουμε το interface αναφοράς για το γιο i IrefSi. Για να αποφευχθεί η σύγχυση σε κάθε μήνυμα linkUp ή linkDown θα σημειώνεται και το interface αναφοράς για το οποίο στάλθηκε (π.χ. ***linkUpIrefSi***), ενώ το sysUpTime ενός δρομολογητή D θα συμβολίζεται με ***sysUpTimeD***. Επίσης, το σύμβολο ***TCur*** δηλώνει την τρέχουσα ώρα. Τέλος, το σύμβολο ∈ χρησιμοποιείται για να δηλώσει ότι κάτι ***ανήκει*** σε ένα σύνολο.



Σχήμα 4.11: *«Γιοι» ενός δρομολογητή*

If έρθει start από R ∧ linkUp από Iref
  then βλάβη στον R
    flag=1
    if Tstart - LDTable[Iref]>3*TmeanRouter
      then διάρκεια βλάβης = 3*TmeanRouter
    else διάρκεια βλάβης = Tstart - LDTable[Iref]
    end if-else
    descending (flag, Iref)
end if

If έρθει linkUp από Iref ∧ ¬(start από R)
  then βλάβη στη Z
    flag=0
    if TLinkUp - LDTable[Iref]>3*TmeanLink
      then διάρκεια βλάβης = 3*TmeanLink
    else διάρκεια βλάβης =TLinkUp - LDTable[Iref]
    end if-else
    descending (flag, Iref)
end if

Η μεταβλητή flag παίρνει την τιμή 1, όταν η πρώτη βλάβη εντοπίζεται σε δρομολογητή και την τιμή 0, όταν η πρώτη βλάβη εντοπίζεται σε ζεύξη. Στη συνέχεια καλείται η συνάρτηση descending για τον εντοπισμό βλαβών που συνέβησαν σε

δρομολογητές κατώτερων επιπέδων του δικτύου, την ώρα που βρίσκονταν σε εξέλιξη η πρώτη βλάβη που εντοπίστηκε. Κατά τα άλλα, ο αλγόριθμος είναι ίδιος με πριν. Ακολούθως, παραθέτουμε τον ψευδοκώδικα για τη συνάρτηση descending.

```
descending (flag, Iref)

   if flag= =1
      then procedure1(Iref, R)
   end if

  if flag= =0
     then ρώτα το sysUpTimeR
          if (sysUpTimeR < Tcur - LDTable[Iref]) ∧
             (Tcur -sysUpTimeR - LDTable[Iref]<2*TmeanRouter)
              then βλάβη στον R
                   διάρκεια βλάβης = Tcur-sysUpTimeR - LDTable[Iref]
           end if
          procedure1(Iref, R)
  end if
end descending
```

Βλέπουμε ότι η συνάρτηση descending χρησιμοποιεί τη συνάρτηση procedure1, της οποίας τον ψευδοκώδικα παραθέτουμε στη συνέχεια.

```
procedure1 ( Iref, R)

   if R ∈ στο τελευταίο επίπεδο του δικτύου
      then stop
   end if
   for κάθε γιο Si του R
       ρώτα το sysUpTimeSi
       if (sysUpTimeSi < Tcur - LDTable[Iref]) ∧
          (Tcur-sysUpTimeSi - LDTable[Iref]< 1.5*TmeanRouter)
           then βλάβη στον Si
                if Tcur- sysUpTimeSi - LDTable[Iref Si]>2*TmeanRouter
                   then διάρκεια βλάβης = TmeanRouter
                else διάρκεια βλάβης = Tcur-sysUpTimeSi - LDTable[Iref Si]
                end if-else
       end if
    end for
   Κάλεσε την procedure1 (Iref, Si) για κάθε γιο του R
end procedure1
```

Όπως βλέπουμε, η δεύτερη έκδοση το κώδικα είναι ίδια με την πρώτη με τη διαφορά ότι ορίζεται μία ακόμα συνάρτηση. Η συνάρτηση αυτή ονομάζεται descending και καλείται κάθε φορά που εντοπίζεται μία βλάβη από τον απλοποιημένο αλγόριθμο.

Αυτό που κάνει η συγκεκριμένη συνάρτηση είναι ότι αρχικά ελέγχει αν η βλάβη που προκάλεσε την κλίση της, είναι βλάβη δρομολογητή ή ζεύξης (μέσω της μεταβλητής flag). Αν είναι βλάβη δρομολογητή, τότε απλώς καλείται η συνάρτηση procedure1. Αν είναι βλάβη ζεύξης, τότε αρχικά ρωτάται ο δρομολογητής που είναι «κάτω» από τη ζεύξη (δηλαδή ο δρομολογητής που έχει το ίδιο interface αναφοράς με τη ζεύξη) για το sysUpTime του. Αν ισχύει ότι ο δρομολογητής αυτός άρχισε να λειτουργεί μετά τη χρονική στιγμή που στάλθηκε το πιο πρόσφατο μήνυμα linkDown για το Iref της αρχικής βλάβης και η βλάβη του διήρκησε λιγότερο από δύο φορές το μέσο χρόνο που διαρκεί μία βλάβη σε δρομολογητή, τότε η εντοπίζεται βλάβη στο δρομολογητή με διάρκεια Tcur-sysUpTimeR - LDTable[Iref]. Στη συνέχεια καλείται η procedure1.

Στο σημείο αυτό πρέπει να κάνουμε μία παρατήρηση που σχετίζεται με τη συνθήκη που χρησιμοποιείται για να εντοπιστεί μία βλάβη στο δρομολογητή. Ίσως φαινομενικά να αρκεί να ελέγξουμε ότι ο δρομολογητής αυτός άρχισε να λειτουργεί μετά τη χρονική στιγμή που στάλθηκε το πιο πρόσφατο μήνυμα linkDown για το Iref της αρχικής βλάβης, όμως αυτό δεν είναι αλήθεια. Ο λόγος είναι ότι αν εκείνο το μήνυμα linkDown είχε χαθεί λόγω βλάβης σε ανώτερο επίπεδο του δικτύου, είναι πολύ πιθανόν να βρούμε βλάβη που δεν είναι πραγματική (για την ακρίβεια θα έχει εντοπιστεί παλαιότερα, οπότε εμείς απλώς θα την ξαναμετρήσουμε) και η οποία θα έχει μεγάλο χρόνο επισκευής. Για το λόγο αυτό, θέτουμε ως επιπλέον προϋπόθεση για να θεωρήσουμε ότι ο δρομολογητής είχε βλάβη, ότι η βλάβη αυτή δεν πρέπει να έχει διάρκεια μεγαλύτερη από δύο φορές το μέσο χρόνο βλάβης σε ένα δρομολογητή. Φυσικά, ακόμα και αυτός ο έλεγχος δεν μπορεί να αποκλείσει τον λανθασμένο εντοπισμό μίας βλάβης. Όπως, θα δούμε μάλιστα σε επόμενη ενότητα η δεύτερη έκδοση του αλγορίθμου εντοπίζει πραγματικά και βλάβες που δε θα έπρεπε. Ο χρόνος 2*TmeanRouter που χρησιμοποιήθηκε ως κατώφλι για τον έλεγχο επιλέχθηκε μετά από δοκιμές που έδειξαν ότι βοηθά στην εύρεση πολλών αληθινών βλαβών χωρίς να εντοπίζονται πολλές βλάβες λανθασμένα.

Η τελευταία συνάρτηση που περιέχεται στον ψευδοκώδικα είναι η procedure1. Η συνάρτηση αυτή ελέγχει αν οι γιοι ενός δρομολογητή έπαθαν βλάβη, η οποία επισκευάστηκε κατά τη διάρκεια της πρώτης βλάβης που προκάλεσε την κλήση της descending. Η procedure1 καλείται αναδρομικά, ώστε να γίνει έλεγχος όλων των δρομολογητών που βρίσκονται στο υποδέντρο «κάτω» από την αρχική βλάβη (στην προσομοίωση η αναδρομή έχει αντικατασταθεί με βρόχους επανάληψης για να επιτυγχάνεται μεγαλύτερη ταχύτητα στους υπολογισμούς).

Στην procedure1 ρωτάται κάθε γιος του δρομολογητή R για το sysUpTime του. Αν ισχύει ότι ο δρομολογητής αυτός άρχισε να λειτουργεί μετά τη χρονική στιγμή που στάλθηκε το πιο πρόσφατο μήνυμα linkDown, για το Iref της αρχικής βλάβης, και ότι η βλάβη του διήρκησε λιγότερο από μιάμιση φορά το μέσο χρόνο που διαρκεί μία βλάβη σε δρομολογητή, τότε εντοπίζεται βλάβη στο δρομολογητή. Για να υπολογιστεί η χρονική διάρκεια της βλάβης κάνουμε υπολογισμούς με βάση το interface αναφοράς του δρομολογητή-γιου. Αν ισχύει ότι η διάρκεια της υπολογιζόμενης βλάβης (Tcur-sysUpTimeSi - LDTable[Iref Si]) είναι μικρότερη από δύο φορές το μέσο χρόνο επισκευής βλάβης δρομολογητή, τότε τη θέτουμε ίση με Tcur-sysUpTimeSi - LDTable[Iref Si], αλλιώς τη θέτουμε ίση με TmeanRouter. Και σε αυτή την περίπτωση η επιλογή των κατωφλίων έγινε έτσι ώστε οι μέσοι υπολογιζόμενοι χρόνοι να είναι όσο το δυνατόν πιο κοντά στους πραγματικούς. Παρόλα αυτά πρέπει να επισημάνουμε ότι στην πραγματικότητα στη συγκεκριμένη περίπτωση που αναφερόμαστε χάνονται αρκετά μηνύματα linkDown από τα interface

αναφοράς των δρομολογητών γιων. Αυτός είναι και ο λόγος που όταν εντοπίζουμε μια πιθανή τέτοια περίπτωση, θέτουμε το χρόνο επισκευής βλάβης ίσο ακριβώς με TmeanRouter. Αν χάνονταν λιγότερα μηνύματα linkDown, θα μπορούσαμε να θέτουμε το χρόνο επισκευής ίσο με 2*TmeanRouter και να παίρνουμε ικανοποιητικά αποτελέσματα (κάτι που έχει ελεγχθεί ότι δεν ισχύει).

4.5 Παραγωγή σφαλμάτων [9], [10]

Όπως θα δούμε και στην ενότητα όπου θα παρουσιαστεί ένα παράδειγμα χρήσης του προγράμματος προσομοίωσης που κατασκευάσαμε για να ελεγχθούν οι αποδόσεις του αλγορίθμου, ο χρήστης εισάγει αρχικά τους μέσους χρόνους για την επισκευή των δρομολογητών, των ζεύξεων και των interfaces (έστω m_R, m_L και m_I αντίστοιχα), καθώς και το μέσω χρόνο που μεσολαβεί μεταξύ δύο βλαβών σε ένα δρομολογητή σε μία ζεύξη και σε ένα interface (έστω α_R, α_L και α_I αντίστοιχα). Τα μεγέθη αυτά προκύπτουν από τη μακροσκοπική στατιστική παρατήρηση του δικτύου που μας ενδιαφέρει. Στην ενότητα 4.6 θα δούμε ότι ο μέσος χρόνος επισκευής βλάβης στο δίκτυο της επιχείρησης που επισκεφτήκαμε είναι περίπου 2 ώρες. Επιπλέον, θα τρέξουμε την προσομοίωση για δεδομένα που ανταποκρίνονται στην επιχείρηση αυτή και θα δούμε τις αποδόσεις του αλγορίθμου. Στην παρούσα ενότητα θα αναφερθούμε στα μαθηματικά μοντέλα που χρησιμοποιούμε για την παραγωγή των βλαβών και της διάρκειάς τους στο δίκτυο της προσομοίωσης.

Στην προσομοίωση που κάναμε ο χρόνος επισκευής κάθε βλάβης ακολουθεί την *εκθετική κατανομή* με ρυθμό $\frac{1}{m_j}$, όπου το m_j είναι ίσο με m_R, m_L ή m_I ανάλογα με το αν η βλάβη συμβαίνει σε δρομολογητή, ζεύξη ή interface (οπότε ο μέσος χρόνος επισκευής είναι ίσος με m_j).

Ο μέσος χρόνος μεταξύ δύο οποιονδήποτε βλαβών ακολουθεί επίσης την *εκθετική κατανομή* με μέση τιμή:

$$\frac{1}{[z_R\frac{1}{a_R}+z_L\frac{1}{a_L}+z_I\frac{1}{a_I}][z_R(N_R-n_R)+z_L(N_L-n_L)+z_I(N_I-n_I)]} \quad (1)$$

Στον παραπάνω τύπο N_j (με j = R, L ή I) είναι ο αριθμός των δρομολογητών των ζεύξεων και των interfaces που υπάρχουν στο δίκτυο, ενώ n_j (με j = R, L ή I) είναι ο αριθμός των δρομολογητών των ζεύξεων και των interfaces που είναι εκτός λειτουργίας την ώρα που παράγεται μία νέα βλάβη. Τα z_j (με j = R, L ή I) είναι πιθανότητες που αθροίζουν στη μονάδα, δηλαδή $z_R+z_L+z_I=1$.

Πιο συγκεκριμένα η παράμετρος z_j προκύπτει ως εξής:

Έστω α_{max} το μέγιστο των α_j, τότε ορίζουμε $k_j=\frac{a_{max}}{a_j}$. Ο αριθμός αυτός ουσιαστικά είναι τόσο μεγαλύτερος, όσο μικρότερος είναι ο μέσος χρόνος μεταξύ δύο βλαβών για ένα είδος στοιχείων του δικτύου (π.χ. δρομολογητής) σε σχέση με το

είδος των στοιχείων που έχουν το μεγαλύτερο μέσο χρόνο λειτουργίας χωρίς βλάβη. Επίσης, ορίζουμε $\Sigma = \sum_j k_j N_j$ (με j = R, L, I).

Τελικά ορίζουμε $z_j = \frac{k_j N_j}{\Sigma}$. Παρατηρούμε ότι οι πιθανότητες αυτές χρησιμοποιούνται για την παραγωγή σταθμισμένων αθροισμάτων στον παρονομαστή του τύπου (1), καθώς ο μέσος χρόνος μεταξύ δύο βλαβών για κάθε είδος στοιχείου πρέπει να έχει διαφορετική βαρύτητα στην παραγωγή του μέσου χρόνου μεταξύ δύο οποιονδήποτε βλαβών.

Τέλος, αναφέρουμε ότι το κάθε στοιχείο έχει πιθανότητα να πάθει τη νέα βλάβη που παράγεται ίση με $\frac{k_j}{\Sigma}$, δηλαδή όλα τα στοιχεία του ίδιου είδους έχουν την ίδια πιθανότητα να πάθουν βλάβη, αλλά η πιθανότητα να πάθουν βλάβη δύο στοιχεία που ανήκουν σε διαφορετικά είδη (π.χ. δρομολογητής και ζεύξη) είναι διαφορετική. Φυσικά, κάθε φορά που «ανατίθεται» μία βλάβη σε ένα στοιχείο, γίνεται έλεγχος ώστε μην είναι ήδη εκτός λειτουργίας.

4.6 Παράδειγμα εκτέλεσης

Στην παρούσα ενότητα θα παρουσιαστεί η προσομοίωση που φτιάξαμε. Στο σχήμα 4.12 φαίνεται η διαπεφή για την επικοινωνία του χρήστη με το πρόγραμμα. Υπάρχουν έξι πεδία στα οποία ο χρήστης εισάγει δεδομένα που σχετίζονται με τη διαθεσιμότητα των στοιχείων του δικτύου.

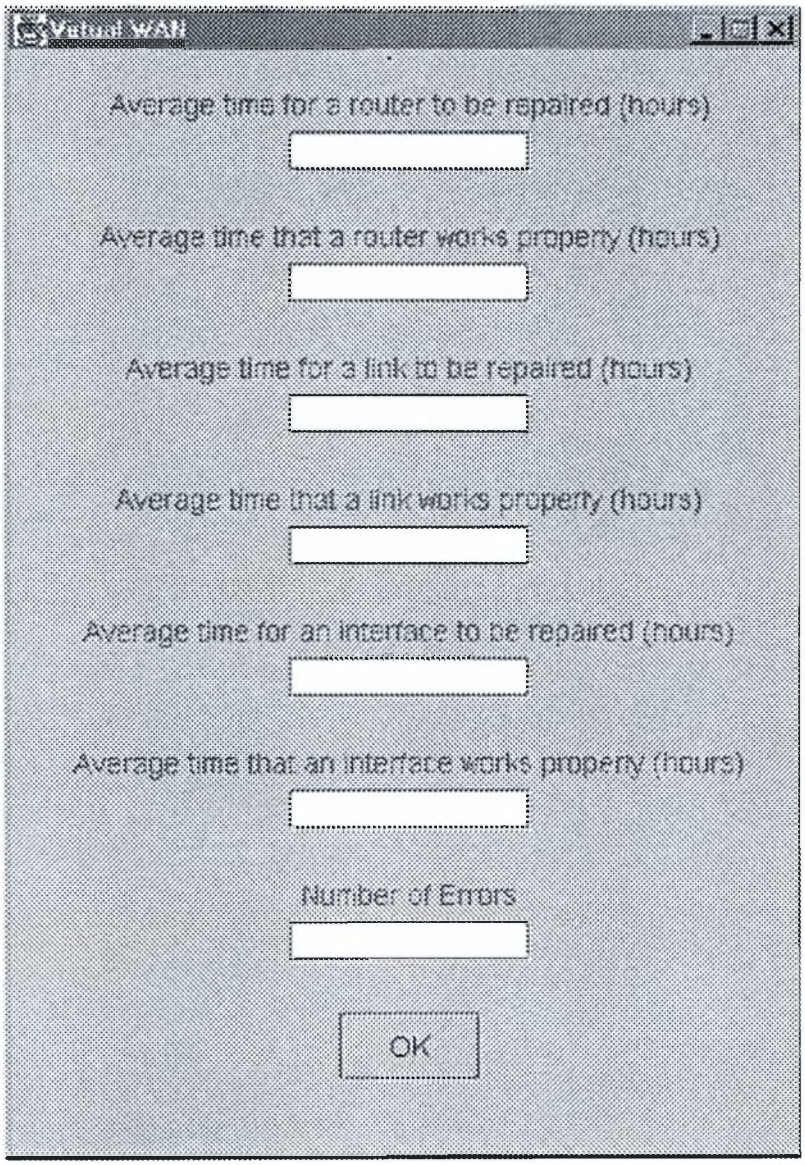
- Στο πρώτο πεδίο συμπληρώνεται ο μέσος χρόνος για την επισκευή ενός δρομολογητή.
- Στο δεύτερο πεδίο συμπληρώνεται ο μέσος χρόνος μεταξύ δύο βλαβών σε ένα δρομολογητή.
- Στο τρίτο πεδίο συμπληρώνεται ο μέσος χρόνος για την επισκευή μίας ζεύξης.
- Στο τέταρτο πεδίο συμπληρώνεται ο μέσος χρόνος μεταξύ δύο βλαβών σε μία ζεύξη.
- Στο πέμπτο πεδίο συμπληρώνεται ο μέσος χρόνος για την επισκευή ενός interface.
- Στο έκτο πεδίο συμπληρώνεται ο μέσος χρόνος μεταξύ δύο βλαβών σε ένα interface.

Στη συνέχεια ο χρήστης «κλικάρει» στο κουμπί OK και εκτελείται η προσομοίωση. Μετά την εκτέλεση τις προσομοίωσης παράγονται τρία αρχεία txt: το αρχείο real_errors, το αρχείο calculated_errors, και το αρχείο calcul2_errors.

Στο αρχείο real_errors περιέχονται όλες οι βλάβες που προκλήθηκαν κατά την προσομοίωση. Κάθε βλάβη παρουσιάζεται με τη μορφή

ID : K Error : Λ Repair : M

όπου Κ είναι το αναγνωριστικό του στοιχείου του δικτύου που έπαθε τη βλάβη, Λ η ώρα που εκδηλώθηκε η βλάβη και Μ η ώρα που επισκευάστηκε η βλάβη. Οι πέντε τελευταίες γραμμές του αρχείου έχουν τη μορφή



Σχήμα 4.12: *Διεπαφή προσομοίωσης*

Total errors : A

Total errors in routers : B

Total errors in links : Γ

Average time for an error in routers : Δ

Average time for an error in links : E

όπου Α είναι ο συνολικός αριθμός των βλαβών που συνέβησαν, Β ο αριθμός των βλαβών που συνέβησαν σε δρομολογητές (και interfaces), Γ ο αριθμός των βλαβών που συνέβησαν σε ζεύξεις, Δ ο μέσος χρόνος που διαρκεί μία βλάβη σε δρομολογητή (και interface) και Ε ο μέσος χρόνος που διαρκεί μία βλάβη σε ζεύξη.

Στο αρχείο calculated_errors περιέχονται τα αποτελέσματα της εφαρμογής του απλοποιημένου αλγορίθμου που παρουσιάστηκε στην προηγούμενη ενότητα. Κάθε εντοπιζόμενη βλάβη παρουσιάζεται με τη μορφή

ID : Κ Duration : Λ

όπου Κ το αναγνωριστικό του στοιχείου του δικτύου στο οποίο παρουσιάστηκε η βλάβη και Λ η διάρκεια της βλάβης. Επισημαίνεται για μία ακόμα φορά ότι αν η βλάβη συμβεί σε ένα interface, το Κ θα περιέχει το αναγνωριστικό του δρομολογητή στον οποίο βρίσκεται το interface. Οι πέντε τελευταίες γραμμές του αρχείου έχουν ακριβώς την ίδια μορφή με τις πέντε τελευταίες γραμμές του αρχείου real_errors και περιέχουν τα αντίστοιχα αποτελέσματα της πρώτης έκδοσης του αλγορίθμου.

Στο αρχείο calcul2_errors περιέχονται τα αποτελέσματα της εφαρμογής της δεύτερης έκδοσης του αλγορίθμου που παρουσιάστηκε στην προηγούμενη ενότητα. Κάθε εντοπιζόμενη βλάβη παρουσιάζεται με τη μορφή που παρουσιάζεται και στο αρχείο calculated_errors, όπως αυτή περιγράφηκε νωρίτερα. Στις επτά τελευταίες γραμμές του αρχείου περιέχονται τα συγκεντρωτικά αποτελέσματα για την απόδοση της δεύτερης έκδοσης του αλγορίθμου. Οι γραμμές αυτές έχουν τη μορφή

Total errors : A

Total errors in routers : B

Total errors in links : Γ

Non existed errors : Δ

Average time for an error in routers : E

Average time for an error in links : ΣΤ

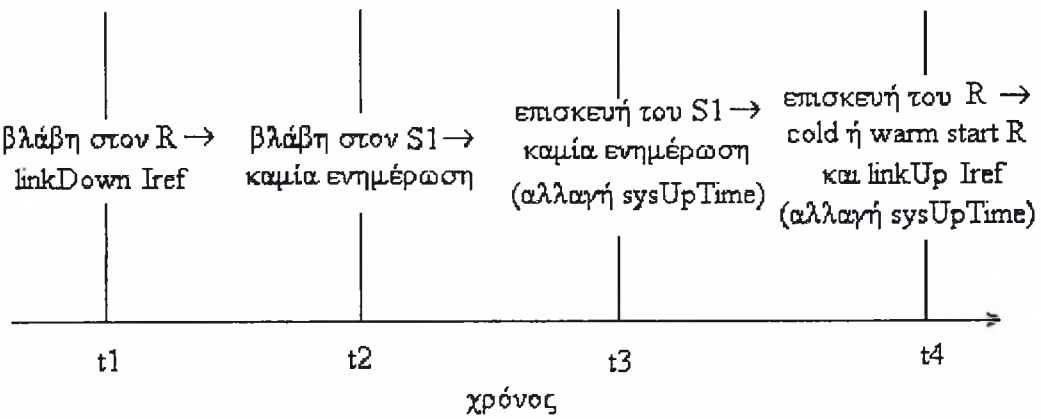
Average time for an error discovered with algorithm 2 : Z

όπου το Α είναι ο συνολικός αριθμός βλαβών που εντόπισε ο αλγόριθμος, Β είναι ο αριθμός των βλαβών που εντοπίστηκαν σε δρομολογητές (και interfaces) και Γ ο αριθμός των βλαβών που εντοπίστηκαν σε ζεύξεις. Το Δ είναι ένας αρνητικός αριθμός, η απόλυτη τιμή του οποίου δείχνει πόσες από τις υπολογιζόμενες βλάβες δεν υφίστανται πραγματικά. Στην προηγούμενη ενότητα έχουμε εξηγήσει ότι μπορεί να εντοπιστούν λανθασμένα κάποιες βλάβες σε δρομολογητές κατά την εκτέλεση της descending. Το Δ δηλώνει τον αριθμό των βλαβών αυτών. Τα Ε και ΣΤ δηλώνουν τους μέσους χρόνους που υπολογίζει η δεύτερη έκδοση του αλγορίθμου για τη διάρκεια μίας βλάβης σε ένα δρομολογητή ή ζεύξη αντίστοιχα. Τέλος το Z περιέχει το

μέσο χρόνο που υπολογίστηκε ότι διαρκούν οι βλάβες οι οποίες εντοπίστηκαν αποκλειστικά με τη συνάρτηση descending.

Όπως είπαμε, η συνάρτηση descending βρίσκει βλάβες δρομολογητών που βρίσκονταν σε εξέλιξη και επισκευάστηκαν, όταν σε ανώτερο επίπεδο του δικτύου συνέβηκε μία άλλη βλάβη. Αυτό έχει σα συνέπεια να είναι μεγάλη η πιθανότητα να έχει χαθεί το linkDown για το αντίστοιχο Iref της βλάβης. Για να παρουσιάσουμε μία τέτοια περίπτωση θα χρησιμοποιήσουμε το σχήμα 4.11. Έστω, ότι τη χρονική στιγμή t1 παθαίνει βλάβη ο δρομολογητής R. Αυτό θα έχει ως συνέπεια την αποστολή ενός μηνύματος linkDown για το interface Iref στο διαχειριστή. Στη συνέχεια, τη χρονική στιγμή t2, παθαίνει βλάβη ο δρομολογητής S1, αλλά ο διαχειριστής δε θα έχει καμία ενημέρωση λόγω της βλάβης του R. Αυτό έχει ως αποτέλεσμα να μην ενημερωθεί ο πίνακας LDTable. Τη χρονική στιγμή t3 επισκευάζεται ο S1, αλλά και πάλι στο διαχειριστή δε θα φτάσει κανένα μήνυμα που να το δηλώνει αυτό. Φυσικά, το γεγονός καταγράφεται στο sysUpTime του S1. Τη χρονική στιγμή t4 επισκευάζεται και ο R, οπότε στέλνονται στο διαχειριστή ένα μήνυμα start για τον R και ένα μήνυμα linkUp για το Iref. Τότε, σύμφωνα με τη δεύτερη έκδοση του αλγορίθμου, θα κλιθεί η συνάρτηση descending. Η descending θα εντοπίσει τη βλάβη στον S1, μέσω του sysUpTime, αλλά όταν υπολογίσει το χρόνο της βλάβης, επειδή χάθηκε το linkDown που αντιστοιχούσε στη βλάβη, θα υπολογίσει ένα χρόνο προφανώς πολύ μεγαλύτερο από τον πραγματικό. Αυτός είναι και ο λόγος που χρησιμοποιήσαμε της αντίστοιχες ευρεστικές παραμέτρους, τη χρήση των οποίων αναλύσαμε όταν παρουσιάσαμε τον αλγόριθμο. Η ακολουθία των γεγονότων του παραδείγματος παρουσιάζεται στο σχήμα 4.13.

Το παραπάνω παράδειγμα δείχνει έμμεσα και το λόγο για τον οποίο παρουσιάζουμε ξεχωριστά στην προσομοίωση το χρόνο των βλαβών που εντοπίζει η descending (πέρα από το E που περιέχει το μέσο χρόνο για όλες τις βλάβες που εντοπίστηκαν σε δρομολογητές). Θέλουμε να έχουμε μία ξεκάθαρη εικόνα για το πόσο καλά αποτελέσματα παίρνουμε με τη χρήση των ευρεστικών παραμέτρων τις οποίες χρησιμοποιήσαμε στη συνάρτηση αυτή.



Σχήμα 4.13: *Ακολουθία γεγονότων παραδείγματος*

Στη συνέχεια θα παρουσιαστούν τα αποτελέσματα της προσομοίωσης για δύο περιπτώσεις. Τα αποτελέσματα θα παρουσιαστούν σε πίνακες ώστε να είναι πιο ευανάγνωστα.

Στον πίνακα 4.3 παρουσιάζονται τα αποτελέσματα που προκύπτουν όταν δώσουμε ως είσοδο στο πρόγραμμα τα παρακάτω δεδομένα.

Μέσος χρόνος επισκευής δρομολογητή: 4
Μέσος χρόνος μεταξύ δύο βλαβών σε δρομολογητή:100
Μέσος χρόνος επισκευής ζεύξης: 3
Μέσος χρόνος μεταξύ δύο βλαβών σε ζεύξη: 100
Μέσος χρόνος επισκευής interface: 4
Μέσος χρόνος μεταξύ δύο βλαβών σε interface:10.000
Αριθμός βλαβών: 10.000

	Πραγματικά δεδομένα	**Έκδοση 1 αλγορίθμου**	**% απόκλιση**	**Έκδοση 2 αλγορίθμου**	**% απόκλιση**
Σύνολο βλαβών	10.000	9.031	9,7	9.420	6
Βλάβες σε δρομολογητές	5.027	4.544	9,7	4.933	2,26
Βλάβες σε ζεύξεις	4.973	4.487	9,8	4.487	9,8
Μέσος χρόνος βλάβης σε δρομολογητές (h)	3,968	4,057	2,2	4,021	1,4
Μέσος χρόνος βλάβης σε ζεύξεις (h)	2,947	3,018	2,4	3,018	2,4
Λανθασμένα υπολογισμένες βλάβες	–	–	–	18	0,2
Μέσος χρόνος βλαβών της descending (h)	–	–	–	3,643	8,9

Πίνακας 4.3: *Αποτελέσματα προσομοίωσης με τυχαίες τιμές*

Τόσο στον πίνακα 4.3 όσο και στον πίνακα 4.4, για κάθε έκδοση του αλγορίθμου, υπάρχει μία στήλη στην οποία αναγράφονται οι % αποκλίσεις από τα πραγματικά δεδομένα στους υπολογισμούς των διαφόρων μεγεθών. Σημειώνουμε ότι στη δεύτερη έκδοση του αλγορίθμου, στην παρουσίαση της απόκλισης του συνόλου βλαβών και των βλαβών σε δρομολογητές, στην απόκλιση περιλαμβάνονται και οι βλάβες που δεν εντοπίστηκαν καθόλου και οι βλάβες που εντοπίστηκαν λανθασμένα από τη descending. Επίσης, η σύγκριση για την % απόκλιση του μέσου χρόνου βλαβών που υπολογίζει η descending, γίνεται σε σχέση με τον πραγματικό μέσο χρόνο διάρκειας βλαβών στους δρομολογητές.

Παρατηρούμε ότι ο απλοποιημένος αλγόριθμος βρήκε περίπου το 90% των βλαβών, ενώ η δεύτερη έκδοση του αλγορίθμου βρήκε περίπου το 94% των συνολικών βλαβών. Το ποσοστά αυτό ξεπερνά το 97% στην περίπτωση των

δρομολογητών. Επίσης, βλέπουμε ότι και η προσέγγιση της διάρκειας των βλαβών είναι καλή. Ο απλοποιημένος αλγόριθμος βρίσκει τους μέσους χρόνους βλαβών με απόκλιση μικρότερη του 2,5% από τα πραγματικά δεδομένα, τόσο για τους δρομολογητές όσο και για τις ζεύξεις. Η δεύτερη έκδοση του αλγορίθμου βρίσκει τους μέσους χρόνους επισκευής βλαβών με απόκλιση περίπου 1,5 % για την περίπτωση των δρομολογητών και 2,5% για την περίπτωση των ζεύξεων.

Παρατηρούμε ότι η απόκλιση στους χρόνους που υπολογίζει η descending είναι της τάξης του 9%. Η descending μειώνει την απόκλιση για τις συνολικές βλάβες που εντοπίζονται με τον απλοποιημένο αλγόριθμο κατά 3,5%, όμως ένα ποσοστό 0,2% επί των συνολικών βλαβών εντοπίστηκαν εσφαλμένα, για αυτό και στη δεύτερη έκδοση του αλγορίθμου έχουμε απόκλιση της τάξης του 6%. Αντίστοιχα, στις βλάβες που εντοπίζονται σε δρομολογητές, η descending μειώνει την απόκλιση κατά 7,8%, όμως εντοπίζει και ένα ποσοστό βλαβών της τάξης του 0,36% λανθασμένα, οπότε το συγκεκριμένο μέγεθος υπολογίζεται από τη δεύτερη έκδοση του αλγορίθμου με απόκλιση 2.26%

Στον πίνακα 4.4 παρουσιάζονται τα αποτελέσματα που προκύπτουν όταν δώσουμε ως είσοδο στο πρόγραμμα τα παρακάτω δεδομένα, τα οποία ανταποκρίνονται στα δεδομένα της επιχείρησης που επισκεφτήκαμε.

Μέσος χρόνος επισκευής δρομολογητή: 2
Μέσος χρόνος μεταξύ δύο βλαβών σε δρομολογητή: 950
Μέσος χρόνος επισκευής ζεύξης: 2
Μέσος χρόνος μεταξύ δύο βλαβών σε ζεύξη: 950
Μέσος χρόνος επισκευής interface: 2
Μέσος χρόνος μεταξύ δύο βλαβών σε interface: 950.000
Αριθμός βλαβών: 10.000

Βλέπουμε ότι για τα συγκεκριμένα δεδομένα ο απλοποιημένος αλγόριθμος βρίσκει περίπου το 99.3% των βλαβών. Αυτό σημαίνει ότι για δεντροειδή δίκτυα, τα οποία έχουν υψηλά ποσοστά διαθεσιμότητας, ο απλοποιημένος αλγόριθμος δίνει πολύ καλά αποτελέσματα. Η δεύτερη έκδοση του αλγορίθμου ανεβάζει το ποσοστό εύρεσης βλαβών στο 99.6%. Καλές είναι επίσης και οι εκτιμήσεις που κάνει ο αλγόριθμος για τους χρόνους που διαρκούν οι βλάβες. Βλέπουμε ότι η απόκλιση και στις δύο εκδόσεις του αλγορίθμου, τόσο στην περίπτωση των δρομολογητών όσο και στην περίπτωση των ζεύξεων, είναι περίπου 5%.

Η descending μειώνει την απόκλιση για τις συνολικές βλάβες που εντοπίζονται με τον απλοποιημένο αλγόριθμο κατά 0,3%, όμως ένα ποσοστό 0,05% επί των συνολικών βλαβών εντοπίστηκαν εσφαλμένα, για αυτό και στη δεύτερη έκδοση του αλγορίθμου έχουμε απόκλιση της τάξης του 0,35%. Αντίστοιχα, στις βλάβες που εντοπίζονται σε δρομολογητές, η descending μειώνει την απόκλιση κατά 0,58%, όμως εντοπίζει και ένα ποσοστό βλαβών της τάξης του 0,1% λανθασμένα, οπότε το συγκεκριμένο μέγεθος υπολογίζεται από τη δεύτερη έκδοση του αλγορίθμου με απόκλιση 0,12%

	Πραγματικά δεδομένα	**Έκδοση 1 αλγορίθμου**	**% απόκλιση**	**Έκδοση 2 αλγορίθμου**	**% απόκλιση**
Σύνολο βλαβών	10.000	9.931	0,6	9.968	0,35
Βλάβες σε δρομολογητές	4.873	4.842	0,6	4.872	0,12
Βλάβες σε ζεύξεις	5.121	5.096	0,5	5.096	0,5
Μέσος χρόνος βλάβης σε δρομολογητές (h)	1,999	1,893	5,3	1,893	5,3
Μέσος χρόνος βλάβης σε ζεύξεις (h)	1,974	1,879	4,8	1,879	4,8
Λανθασμένα υπολογισμένες βλάβες	–	–	–	5	0,05
Μέσος χρόνος βλαβών της descending(h)	–	–	–	1,987	0,6

Πίνακας 4.4: *Αποτελέσματα προσομοίωσης με πραγματικές τιμές*

ΕΠΙΛΟΓΟΣ

Εργασίες μελέτης

Στην παρούσα εργασία ασχοληθήκαμε με το θέμα της διαχείρισης δικτύων, αγγίζοντας όλες της πτυχές του. Σκοπός μας ήταν να αναλύσουμε όλες τις λειτουργίες διαχείρισης σε ένα δίκτυο και να αναδείξουμε τον τρόπο με τον οποίο η σωστή διαχείριση γίνεται το εργαλείο για την παροχή υπηρεσιών υψηλής ποιότητας, μέσω του δικτύου αυτού. Ιδιαίτερη έμφαση δώσαμε στο πρόβλημα της απώλειας της διαθεσιμότητας.

Η εργασία μπορεί να χωριστεί σε τρεις μεγάλες λογικές ενότητες: στην ανάλυση του προβλήματος (παρουσίαση ενός πραγματικού δικτύου και των προβλημάτων που αντιμετωπίζει, καθώς και παρουσίαση των υπαρχόντων μοντέλων και πρωτοκόλλων διαχείρισης δικτύων) στις εργασίες σχεδιασμού και στις εργασίες εφαρμογής.

Με την παρουσίαση των μοντέλων και των πρωτοκόλλων διαχείρισης, επιχειρήσαμε να δώσουμε μία πλήρη εικόνα για τη διαχείριση σε ένα δίκτυο και για τις πρόνοιες που πρέπει να λαμβάνονται, τόσο κατά το σχεδιασμό του, όσο και κατά τη λειτουργία του. Επιπλέον, μέσω της σύγκρισης που γίνεται ανάμεσα στα δύο επικρατέστερα μοντέλα διαχείρισης, αναδεικνύονται τα προτερήματα και τα μειονεκτήματα του καθενός. Τέλος, η παρουσίαση του πραγματικού δικτύου και του τρόπου με τον οποίο υλοποιούνται οι λειτουργίες διαχείρισης σε αυτό, αναδεικνύει προβλήματα που παρουσιάζονται στη διαχείριση σε πραγματικό περιβάλλον.

Η πρώτη από τις εργασίες σχεδιασμού που πραγματοποιήσαμε ήταν η προς τα κάτω κλιμάκωση του πραγματικού δικτύου και η ανάλυσή του, βάση των λειτουργιών διαχείρισης. Προτείναμε ένα απλοποιημένο μοντέλο για τη μοντελοποίηση του συστήματος, το οποίο αποτελούνταν από δύο δρομολογητές, τη μεταξύ τους ζεύξη και ένα διαχειριστή. Στη συνέχεια δημιουργήσαμε ένα πληροφοριακό μοντέλο για τη μοντελοποίηση των συνιστωσών ενός δικτύου: των δρομολογητών, των ζεύξεων και των interfaces. Οι συνιστώσες αυτές μπορούσαν να συνδυαστούν και να δώσουν ένα εικονικό δίκτυο, όμοιο με το πραγματικό δίκτυο που μελετήσαμε αρχικά. Φυσικά, για να μπορέσουμε στη συνέχεια να παράγουμε στο εικονικό δίκτυο βλάβες που θα συμφωνούσαν ως προς τον χρόνο που συμβαίνουν και τη διάρκειά τους με αυτές που συμβαίνουν σε ένα πραγματικό δίκτυο, έπρεπε να βρούμε τα μαθηματικά μοντέλα τα οποία θα έδιναν το επιθυμητό αποτέλεσμα. Το τελικό βήμα στις εργασίες σχεδίασης, ήταν οι δύο εκδοχές του αλγορίθμου για τον εντοπισμό προβλημάτων που σχετίζονται με τη διαθεσιμότητα

Οι εργασίες εφαρμογής μπορούν να χωριστούν σε δύο σκέλη. Στο πρώτο σκέλος περιλαμβάνεται μία πρώτη προσέγγιση της διαχείρισης σε πραγματικό περιβάλλον μέσω της εύρεσης, εγκατάστασης και χρήσης ενός λογισμικού για την παρακολούθηση των διαχειριζόμενων αντικειμένων στους δρομολογητές του Πανεπιστημίου Θεσσαλίας. Στο δεύτερο σκέλος, περιλαμβάνεται η δημιουργία της προσομοίωσης για την υλοποίηση και τον έλεγχο των αποδόσεων του αλγορίθμου. Για τη δημιουργία της προσομοίωσης αυτής, χρησιμοποιήσαμε τα μοντέλα στα οποία είχαμε καταλήξει κατά τις εργασίες σχεδίασης. Το εικονικό δίκτυο το οποίο κατασκευάσαμε, ήταν αντίστοιχο με αυτό της εταιρίας που μελετήσαμε αρχικά και οι συνιστώσες που χρησιμοποιήθηκαν για τη μοντελοποίηση των στοιχείων του ήταν αυτές στις οποίες αναφερθήκαμε παραπάνω. Τα μαθηματικά μοντέλα που

χρησιμοποιήθηκαν για τη μοντελοποίηση των βλαβών ήταν αυτά στα οποία καταλήξαμε όταν μελετήσαμε θεωρητικά το πρόβλημα, ενώ η επεξεργασία των δεδομένων έγινε με βάση τον αλγόριθμο που σχεδιάσαμε.

Στο κείμενο της εργασίας αναφερθήκαμε αναλυτικά στις δυσκολίες που αντιμετωπίσαμε κατά την επιλογή, την εγκατάσταση και τη χρήση του λογισμικού για την προσπέλαση των MIB στους δρομολογητές του Πανεπιστημίου Θεσσαλίας. Στις δυσκολίες αυτές συμπεριλαμβάνονταν η πληθώρα των υποψήφιων λογισμικών και η ελλιπής τεκμηρίωσή τους, κάτι που ήταν βέβαια αναμενόμενο, αφού αναφερόμαστε σε freeware. Όμως, η μεγαλύτερη δυσκολία την οποία αντιμετωπίσαμε κατά τη διάρκεια εκπόνησης της εργασίας σχετίζεται με την απόκτηση πραγματικών δεδομένων από το δίκτυο της εταιρίας, της οποίας το δίκτυο μελετήσαμε, για την εφαρμογή του αλγορίθμου που προτείναμε σε αυτά.

Όπως αναφέρουμε και στο κείμενο της εργασίας, ένας από τους στόχους μας, όταν επισκεφτήκαμε την εταιρία, ήταν η μελέτη του συστήματος διαχείρισής της, καθώς και η διερεύνηση της διασύνδεσης του αλγορίθμου μας με αυτό. Το σύστημα διαχείρισης της εταιρίας αποδείχθηκε ιδιαίτερα αυτοματοποιημένο και περιορισμένων δυνατοτήτων. Χαρακτηριστικό είναι το γεγονός ότι πέρα από τον εντοπισμό βλαβών, δεν έκανε αναγνώριση και ταυτοποίηση βλαβών (θέματα που καλύπτει η παρούσα εργασία) Σε ότι αφορά στη διασύνδεση του αλγορίθμου με το σύστημα διαχείρισης, δεν κατέστη δυνατή, καθώς δεν υπήρχε μία διεπαφή στο σύστημα διαχείρισης δικτύου της εταιρίας που θα μπορούσε να χρησιμοποιηθεί για τι σκοπό αυτό. Μετά από αυτή την εξέλιξη, ψάξαμε για εργαλεία τα οποία θα μπορούσαμε να χρησιμοποιήσουμε για τη δημιουργία μίας τέτοιας διεπαφής. Τα εργαλεία τελικά βρέθηκαν, όμως ο χρόνος που απαιτούνταν για την εξοικείωσή μας με αυτά ήταν απαγορευτικός, καθώς τα χρονικά όρια εκπόνησης της εργασίας ήταν συγκεκριμένα. Το αποτέλεσμα ήταν ότι τελικά δεν καταφέραμε να εφαρμόσουμε τον αλγόριθμο σε πραγματικά δεδομένα.

Εφαρμογή αποτελεσμάτων – Προοπτικές

Κλείνοντας την εργασία αυτή, θα θέλαμε να αναφερθούμε στα επόμενα βήματα που μπορούν να γίνουν με βάση όσα παρουσιάσαμε εδώ. Το πρώτο βήμα που πρέπει να γίνει είναι σίγουρα η εφαρμογή του αλγορίθμου σε πραγματικά δεδομένα. Για το σκοπό αυτό, πρέπει αρχικά να γίνει χρήση των εργαλείων που θα χρησιμοποιηθούν για την υλοποίηση της διεπαφής η οποία θα επιτρέψει την απόκτηση των δεδομένων. Επίσης, είναι πολύ σημαντικό να αποφασιστεί αν η εφαρμογή του αλγορίθμου θα γίνει αρχικά σε ολόκληρο το δίκτυο ή σε ένα μικρό υποδέντρο του. Η εμπειρία που έχουμε αποκομίσει από την εκπόνηση αυτής της εργασίας μας κάνει να προτείνουμε την αρχική-πιλοτική εφαρμογή του αλγορίθμου σε ένα υποδέντρο του δικτύου. Με τον τρόπο αυτό, θα είναι δυνατή η επικέντρωση της υλοποίησης στην ουσία και όχι σε προβλήματα που σχετίζονται με το μέγεθος του μελετούμενου δικτύου. Στο συγκεκριμένο στάδιο θα πρέπει να επιλυθούν προβλήματα που ίσως φαίνονται τετριμμένα, αλλά είναι πιθανό να αποδειχθούν αρκετά χρονοβόρα. Το πιο χαρακτηριστικό παράδειγμα είναι το format των πραγματικών δεδομένων, τα οποία πρέπει να έρθουν σε εύκολα επεξεργάσιμη μορφή.

Μετά την πιλοτική εφαρμογή του αλγορίθμου σε ένα μέρος τους δικτύου θα μπορεί να γίνει εφαρμογή του σε ολόκληρο το δίκτυο. Το μόνο πρόβλημα το οποίο θα παρουσιαστεί σε αυτό το στάδιο θα είναι το πέρασμα της τοπολογίας του

πραγματικού δικτύου, το οποίο στην περίπτωση της εταιρίας που μελετήσαμε ήταν ιδιαίτερα μεγάλο. Το ιδανικό θα ήταν η δημιουργία ενός αυτοματοποιημένου προγράμματος, το οποίο εξάγει την τοπολογία του δικτύου από πίνακες που διατηρεί η εταιρία και συνδέεται με το πρόγραμμα εκτέλεσης του αλγορίθμου. Το πρόγραμμα αυτό προφανώς δεν είναι τετριμμένο, αλλά η δημιουργία του θα παρείχε τεράστια ευελιξία, καθώς οποιαδήποτε αλλαγή συνέβαινε στην τοπολογία του δικτύου, θα μπορούσε να περνιέται στους πίνακες και από εκεί να γνωστοποιείται αυτόματα στο πρόγραμμα που εκτελεί τον αλγόριθμο, χωρίς να χρειάζονται κάποιες επιπλέον αλλαγές.

Το τελικό αποτέλεσμα θα ήταν η εταιρία στην οποία αναφερόμαστε ή οποιαδήποτε άλλη εταιρία με παρόμοιο δίκτυο, χρησιμοποιώντας τα προγράμματα αυτά να έχει μία πολύ καλή εκτίμηση για την αιτία των βλαβών στο δίκτυό της. Η συγκεκριμένη εκτίμηση θα μπορούσε να αποτελέσει τη βάση για διοικητικές αποφάσεις. Για παράδειγμα, ο εντοπισμός πολλών βλαβών που οφείλονται στον τηλεπικοινωνιακό πάροχο θα μπορούσε να οδηγήσει σε διαπραγματεύσεις για δέσμευσή του σε μεγαλύτερη διαθεσιμότητα των μισθωμένων γραμμών ή ακόμα και σε επιλογή άλλου παρόχου.

Τέλος, θα αναφερθούμε στην περίπτωση επέκτασης του αλγορίθμου σε ένα σύστημα το οποίο διαθέτει ιδιωτική MIB. Όπως είδαμε στο τρίτο κεφάλαιο της εργασίας, το μοντέλο OSI/CMIP παρέχει πολύ καλύτερες δυνατότητες εντοπισμού των βλαβών σε σχέση με το μοντέλο internet/SNMP. Βέβαια, όταν μία επιχείρηση διαθέτει ένα δίκτυο που ο εξοπλισμός του διαθέτει μόνο πράκτορες SNMP, η αντικατάστασή του από εξοπλισμό που διαθέτει πράκτορες CMIP είναι ιδιαίτερα δαπανηρή. Παρόλα αυτά, μία κίνηση που θα μπορούσε να γίνει ώστε ο αλγόριθμος που προτείναμε να έχει καλύτερα αποτελέσματα, χωρίς να χρειαστεί η αντικατάσταση ολόκληρου του εξοπλισμού, είναι ο ορισμός μίας ιδιωτικής MIB. Στη MIB αυτή θα πρέπει να προστεθεί ένα αντικείμενο που θα αντιστοιχεί στις ζεύξεις του δικτύου, το οποίο θα διαθέτει το δικό του sysUpTime. Με τον τρόπο αυτό θα είναι δυνατή η επέκταση της συνάρτησης descending της δεύτερης έκδοσης του αλγορίθμου, ώστε να εντοπίζει και βλάβες σε ζεύξεις, οι οποίες δεν εντοπίστηκαν από τον απλοποιημένο αλγόριθμο.

ΠΑΡΑΡΤΗΜΑ

Στο παρόν παράρτημα παρουσιάζονται τα διαχειριζόμενα αντικείμενα του SNMP και του CMIP που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας.

Αντικείμενα SNMP

System Group

sysDescr: Μία περιγραφή σε μορφή κειμένου για την οντότητα στην οποία αναφερόμαστε. Η περιγραφή πρέπει να περιέχει το πλήρες όνομα και την έκδοση του υλικού (hardware) της οντότητας, καθώς και το λειτουργικό σύστημα/λογισμικό που χρησιμοποιείται.

sysObjectID: Ένα αναγνωριστικό που καθορίζει μονοσήμαντα την οντότητα. Το αναγνωριστικό αυτό ορίζεται από τον κατασκευαστή.

sysUpTime: Ο χρόνος που πέρασε από την τελευταία επανεκκίνηση του συστήματος.

sysContact: Κείμενο που δηλώνει ποιος είναι ο υπεύθυνος για το συγκεκριμένο σύστημα και πώς μπορεί κάποιος να επικοινωνήσει μαζί του.

sysName: Το όνομα του συστήματος, το οποίο δίνεται από το διαχειριστή.

sysLocation: Η φυσική τοποθεσία στην οποία βρίσκεται το συγκεκριμένο σύστημα (π.χ κτήριο, όροφος κ.τ.λ.).

sysServices: Ένας ακέραιος ο οποίος δηλώνει το σύνολο των υπηρεσιών που προσφέρεται από αυτή την οντότητα. Στο RFC 1213 καθορίζεται τι σημαίνουν οι τιμές του ακεραίου.

Interfaces Group

ifNumber: Ο αριθμός των interfaces που έχει το συγκεκριμένο σύστημα.

ifTable: Ένας πίνακας με αντικείμενα ifEntry. Ο πίνακας έχει τόσες γραμμές όσα τα interfaces του συστήματος.

ifEntry: Μία γραμμή του πίνακα ifTable. Περιέχει τα αντικείμενα που ορίζονται στη συνέχεια.

ifIndex: Ένας μοναδικός ακέραιος για κάθε interface του συστήματος. Ο ακέραιος αυτός παίρνει τιμές από 1 μέχρι ifNumber.

ifDescr: Κείμενο το οποίο περιέχει πληροφορίες για το interface (κατασκευαστής, όνομα προϊόντος κ.τ.λ.).

ifType: Ο τύπος του interface σε σχέση με τα πρωτόκολλα που χρησιμοποιούνται σε φυσικό επίπεδο και επίπεδο ζεύξης. Στο RFC 1213 καθορίζεται ποιοι είναι οι τύποι αυτοί.

ifMtu: Το μέγεθος του μεγαλύτερου πακέτου (σε octets) που μπορεί να παραληφθεί ή να αποσταλεί από το συγκεκριμένο interface.

ifSpeed: Μία εκτίμηση για την ταχύτητα των δεδομένων που διακινούνται μέσω του συγκεκριμένου interface. Αν δε μπορεί να γίνει η εκτίμηση αυτή, τότε επιστρέφεται η ονομαστική ταχύτητα που υποστηρίζει το interface.

ifPhysAddress: Η φυσική διεύθυνση που αντιστοιχεί στο συγκεκριμένο interface.

ifAdminStatus: Η επιθυμητή κατάσταση του interface, όπως την όρισε ο διαχειριστής. Μπορεί να πάρει τρεις τιμές: up, down και testing.

ifOperStatus: Η πραγματική τρέχουσα κατάσταση του interface. Μπορεί να πάρει τρεις τιμές up, down και testing.

ifLastChange: Η τιμή που είχε το sysUpTime του συστήματος στο οποίο βρίσκεται το interface, όταν το interface πέρασε στην κατάσταση που βρίσκεται τώρα.

ifInOctets: Ο συνολικός αριθμός octets που εισήλθαν στο interface.

ifInUcastPkts: Ο αριθμός των εισερχόμενων unicast πακέτων που παραδόθηκαν σε πρωτόκολλο υψηλότερου επιπέδου.

ifInNUcastPkts: Ο αριθμός των εισερχόμενων μη unicast (δηλαδή broadcast ή multicast) πακέτων που παραδόθηκαν σε πρωτόκολλο υψηλότερου επιπέδου.

ifInDiscards: Ο αριθμός των εισερχόμενων πακέτων που απορρίφθηκαν, μολονότι δεν είχαν κάποιο λάθος. Ο πιο πιθανός λόγος για μία τέτοια απόρριψη είναι η έλλειψη χώρου στους buffers.

ifInErrors: Ο αριθμός των εισερχόμενων πακέτων τα οποία περιείχαν λάθη, που απέτρεψαν την παράδοσή τους σε πρωτόκολλα ανώτερων επιπέδων.

ifInUnknownProtos: Ο αριθμός των εισερχόμενων πακέτων που απορρίφθηκαν, επειδή χρησιμοποιούσαν κάποιο πρωτόκολλο άγνωστο στο σύστημα.

ifOutOctets: Ο συνολικός αριθμός των octets που εξήλθαν από το interface.

ifOutUcastPkts: Ο συνολικός αριθμός των unicast πακέτων που τα πρωτόκολλα ανώτερων επιπέδων ζήτησαν να μεταδοθούν, συμπεριλαμβανομένων και αυτών που τελικά δε μεταδόθηκαν.

ifOutNUcastPkts: Ο συνολικός αριθμός των μη unicast πακέτων που τα πρωτόκολλα ανώτερων επιπέδων ζήτησαν να μεταδοθούν, συμπεριλαμβανομένων και αυτών που τελικά δε στάλθηκαν.

ifOutDiscards: Ο αριθμός των εξερχόμενων πακέτων που απορρίφθηκαν, μολονότι δεν είχαν κάποιο λάθος. Ο πιο πιθανός λόγος για μία τέτοια απόρριψη είναι η έλλειψη χώρου στους buffers

ifOutErrors: Ο αριθμός των πακέτων που δεν μπόρεσαν να μεταδοθούν από το interface, επειδή είχαν λάθη.

ifOutQLen: Το μήκος της ουράς των εξερχόμενων πακέτων (σε πακέτα).

ifSpecific: Περιέχει αναφορά σε συγκεκριμένους ορισμούς για αντικείμενα των MIB που σχετίζονται με το interface. Για παράδειγμα, αν χρησιμοποιείται ethernet, το συγκεκριμένο αντικείμενο δηλώνει ένα RFC που περιέχει τους ορισμούς των αντικειμένων που σχετίζονται με το ethernet.

Αντικείμενα CMIP

Κλάση circuitPack: Στιγμιότυπα της κλάσης αυτής είναι αντικείμενα που μπορούν να αλλαχθούν στο σύστημα. Τα πιο χαρακτηριστικά παραδείγματα είναι κάρτες, επεξεργαστές και μονάδες παροχής ισχύος. Στη συνέχεια παραθέτουμε τον ορισμό της κλάσης αυτής. Το περιεχόμενο του κάθε πακέτου, τόσο αυτής της κλάσης όσο και αυτών που θα παρουσιάσουμε στη συνέχεια, βρίσκεται στο M3100 της ITU-T.

```
circuitPack MANAGED OBJECT CLASS
DERIVED FROM
        equipmentR1;
CHARACTERIZED BY
        circuitPackPackage,
        alarmSeverityAssignmentPointerPackage,
        equipmentAlarmEffectOnServicePackage,
        currentProblemListPackage,
        equipmentsEquipmentAlarmR1Package,
        stateChangeNotificationPackage,
        administrativeOperationalStatesPackage,
        createDeleteNotificationsPackage;
REGISTERED AS {m3100ObjectClass 30};
```

Κλάση trailR1: Τα στιγμιότυπα της κλάσης αυτής χρησιμοποιούνται για την αναπαράσταση των ζεύξεων του δικτύου. Κάθε ζεύξη έχει δύο termination points. Ακολουθεί ο ορισμός της συγκεκριμένης κλάσης.

```
trailR1 MANAGED OBJECT CLASS
DERIVED FROM
        pipe;
CHARACTERIZED BY
        trailR1Package;
CONDITIONAL PACKAGES
        clientConnectionListPackage PRESENT IF
          "an instance supports it",
        serverConnectionListPackage PRESENT IF
          "an instance supports it";
REGISTERED AS {m3100ObjectClass 25};
```

Κλάση terminationPoint: Τα στιγμιότυπα της κλάσης αυτής χρησιμοποιούνται για την αναπαράσταση των άκρων των ζεύξεων. Ακολουθεί ο ορισμός της συγκεκριμένης κλάσης.

```
terminationPoint MANAGED OBJECT CLASS
DERIVED FROM
        "Recommendation X.721 : 1992":top;
CHARACTERIZED BY
        terminationPointPackage;
CONDITIONAL PACKAGES
        alarmSeverityAssignmentPointerPackage PRESENT IF
          "the tmnCommunicationsAlarmInformationPackage package is present AND the
          managed object supports configuration of alarm severities",
        tmnCommunicationsAlarmInformationPackage PRESENT IF
          "the communicationsAlarm notification (as defined in Recommendation X.721) is
          supported by this managed object",
        networkLevelPackage PRESENT IF
          "an instance supports it",
        characteristicInformationPackage PRESENT IF
          "an instance supports it.",
        crossConnectionPointerPackage PRESENT IF
          "the termination point can be flexibly assigned, (i.e. cross connected).",
```

operationalStatePackage PRESENT IF
"the resource represented by this managed object is capable of assessing the ability to generate and/or receive a valid signal.",
stateChangeNotificationPackage PRESENT IF
"the stateChange notification defined in Recommendation X.721 is supported by an instance of this managed object class",
attributeValueChangeNotificationPackage PRESENT IF
"the attributeValueChange notification defined in Recommendation X.721 is supported by an instance of this managed object class",
createDeleteNotificationsPackage PRESENT IF
"the objectCreation and objectDeletion notifications defined in Recommendation X.721 are supported by an instance of this managed object class";
REGISTERED AS {m3100ObjectClass 8};

Κλάση stateChangeRecord: Η κλάση αυτή χρησιμοποιείται για την καταγραφή γεγονότων, όπως η δημιουργία του αντικειμένου και η αλλαγή της τιμής ενός γνωρίσματός του. Ο παρακάτω ορισμός είναι παρμένος από το X.721.

stateChangeRecord MANAGED OBJECT CLASS
DERIVED FROM eventLogRecord;
- - The appropriate object identifier value for the eventType attribute, inherited from eventLogRecord
- - managed object class, is stateChange
CHARACTERIZED BY
stateChangeRecordPackage PACKAGE
BEHAVIOUR
stateChangeRecordBehaviour BEHAVIOUR
DEFINED AS "This managed object is used to represent logged information that resulted from state change notifications or event reports";;
ATTRIBUTES
stateChangeDefinition GET;;;
CONDITIONAL PACKAGES
sourceIndicatorPackage PRESENT IF
"the Source indicator parameter is present in the stateChange notification or event report corresponding to the instance of state change record",

attributeIdentifierListPackage PRESENT IF

"the Attribute identifierList parameter is present in the stateChange notification or event report corresponding to the instance of state change record";

REGISTERED AS {smi2MObjectClass 12};

Γνώρισμα operationalState: Το συγκεκριμένο γνώρισμα περιγράφει αν το αντικείμενο στο οποίο αναφερόμαστε λειτουργεί ή όχι. Ο ορισμός που ακολουθεί είναι παρμένος από το X.721

OperationalState ::= ENUMERATED {disabled(0), enabled(1)}

ΑΝΑΦΟΡΕΣ

Βιβλιογραφικές πηγές

[1] Mani Subramanian, "Network Management, Principles and Practice", 2000.

[2] Divakara K. Udupa, "TMN – Telecommunications Management Network ", 1999.

[3] Lakshmi G. Raman, "Fundamentals of Telecommunications Network Management", 1999.

[4] Σπύρος Δ. Αρσένης, "Σχεδιασμός και υλοποίηση δικτύων – Από μικρά δίκτυα γραφείου μέχρι μεγάλα δίκτυα επιχειρήσεων ", 2005.

[5] Στέφανος Γκρίτζαλης, Σωκράτης Κ. Κάτσικας και Δημήτρης Γκρίτζαλης, "Ασφάλεια Δικτύων Υπολογιστών", 2003.

[6] James F. Kurose and Keith W. Ross, " Computer Networking, A Top-Down Approach Featuring the Internet", 2003 [Απόδοση στην ελληνική γλώσσα: Γιάννης Β. Σαμαράς, "Δικτύωση υπολογιστών, Προσέγγιση από Πάνω προς τα Κάτω με Έμφαση στο Διαδίκτυο"].

[7] William Stallings "SNMP1, SNMPv2, SNMPv3 and RMON 1 and 2", 3rd edition 1999.

[8] Βλαχάβας Ιωάννης, Κεφαλάς Πέτρος, Βασιλειάδης Νικόλαος, Κόκκορας Φώτης και Σακελαρίου Ηλίας, "Τεχνητή Νοημοσύνη", 2η έκδοση 2005.

[9] J. R. Norris, "Markov Chains", 1997.

[10] Στρατής Κουνιάς και Χρόνης Μωυσιάδης, "Θεωρία Πιθανοτήτων Ι, Κλασική Πιθανότητα και Μονοδιάστατες κατανομές", 1995.

[11] M. Rose, K. McCloghrie, "Structure and Identification of Management Information for TCP/IP–based Internets", [RFC 1155], May 1990

[12] J. Case, M. Fedor, M. Scoffstall, J.David, "A Simple Network Management Protocol (SNMP)", [RFC 1157], May 1990.

[13] M. Rose and K. McCloghrie, "Concise MIB Definitions", [RFC 1212], March 1991.

[14] K. McCloghrie, M. Rose, "Management Information Base for Network Management of TCP/IP-baseb internets: MIB-II", [RFC 1213], March 1991.

[15] "Information technology – Open Systems Interconnection – Common management information service definition", [ISO/IEC 9595], 1990.

[16] "Information Processing Systems – Open Systems Interconnection – Systems Management Overview", [Recommendation X.701| ISO/IEC 10040].

[17] "Generic Network Information Model", [ITU-T Recommendation M.3100], 1995.

[18] "Structure of information–Part 2: Definition of Management Information", [ITU-T Recommendation X.721], 1992.

[19] D. Harrington, R. Presuhn and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", [RFC 2271], January 1998.

[20] U. Blumenthal and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", [RFC 2274], January 1998.

Πηγές στο διαδίκτυο

[A] http://www.rfc-editor.org/

[B] http://www.iso.org/iso/en/ISOOnline.frontpage

[Γ] http://www.itu.int/home/index.html

[Δ] http://www.wtcs.org/snmp4tpc/default.htm

[E] http://en.wikipedia.org/wiki/Main_Page