

The Multi-protocol Management of the Ariadne Network

To Multiprotocol Management του Δικτύου ΑΡΙΑΔΝΗ

A.S. DRIGAS
Applied Technologies Department
NCSR "DEMOKRITOS"
Ag. Paraskevi
GREECE
dr@imm.demokritos.gr
<http://imm.demokritos.gr>

1. Εισαγωγή

Η ευρεία εξάπλωση των δικτύων υπολογιστών, σε γεωγραφικό επίπεδο, επίπεδο υπηρεσιών αλλά και πρωτοκόλλων, κατά την τελευταία δεκαετία, δημιούργησε την ανάγκη για έναν από κεντρικό σημείο έλεγχο των διαφόρων δικτυακών συστημάτων και υπηρεσιών, και γενικότερα της δικτυακής υποδομής. Στο Internet το οποίο είναι το περισσότερο ευρέως διαδεδομένο και εξαπλωμένο δίκτυο υπολογιστών παγκοσμίως η ανάγκη για εποπτεία και διαχείριση των δικτυακών συστημάτων από κεντρικό σημείο, δημιούργησε την απαίτηση για την ανάπτυξη ενός νέου πρωτοκόλλου, το οποίο θα περιγράφει τις αρχές που πρέπει να διέπουν τις λειτουργίες αυτές. Παρόμοια προσπάθεια για την ανάπτυξη αρχών διαχείρισης δικτύου γίνεται και από την πλευρά του OSI δικτύου. Επειδή το OSI δίκτυο δεν είναι τόσο εκτεταμένο όσο το Internet, οι κατασκευαστές έδωσαν αρχικά μεγαλύτερη έμφαση στην ανάπτυξη προϊόντων δικτυακών υπηρεσιών οι οποίες να εμπεριέχουν και λειτουργίες διαχείρισης για απομακρυσμένο κεντρικό έλεγχο, όπως αυτές ορίζονται στις σχετικές συστάσεις των RFCs του Internet ενώ έχουν καθυστερήσει στην ανάπτυξη αντίστοιχων λειτουργιών για τα OSI προϊόντα τους. Έτσι ενώ στον κόσμο του Internet έχει γίνει από καιρό δυνατή η διαχείριση όλων των υπηρεσιών του σε όλα τα επίπεδα, για τον OSI κόσμο έχουν αναπτυχθεί μόνο προϊόντα λίγων κατασκευαστών τα οποία διαχειρίζονται ένα περιορισμένο φάσμα λειτουργιών.

□ Το πρωτόκολλο που αναπτύχθηκε και περιγράφει τον τρόπο επικοινωνίας και τις διαχειριστικές λειτουργίες ενός server ο οποίος ασκεί την εποπτεία στα διάφορα δικτυακά στοιχεία agents στον κόσμο του Internet ονομάζεται SNMP (Simple Network Management Protocol) [45]. Το SNMP συμπληρώνεται με δύο άλλες προτάσεις οι οποίες περιγράφουν, η μια τη βάση της πληροφορίας ελέγχου και η άλλη τη δομή της πληροφορίας. Οι δύο προτάσεις περιγράφονται στις αντίστοιχες συστάσεις με τα ονόματα MIB (Management Information Base) [46], [47] και SMI (Structure Management Information) [48].

□ Το δίκτυο Αριάδνη αναγνωρίζοντας την ανάγκη για διαχείριση της κατανεμημένης σε ολόκληρη τη χώρα δικτυακής του υποδομής εγκατέστησε

και λειτουργεί στο κέντρο διαχείρισης ένα ολοκληρωμένο πακέτο διαχείρισης του Internet και του OSI. Το πακέτο διαχείρισης είναι το Open View [49] της Hewlett Packard. Το πακέτο αυτό έχει προς το παρόν τη δυνατότητα να διαχειρίζεται από τα συστήματα στον OSI κόσμο μόνο το X.400 μέσω ενός proprietary πρωτοκόλλου. Η επιλογή του προγράμματος αυτού έγινε εξ' αιτίας της μεγάλης λειτουργικότητας την οποία διαθέτει και λόγω των εργαλείων που το συνοδεύουν και τα οποία δίνουν τη δυνατότητα ανάπτυξης ειδικών εφαρμογών. Η αδυναμία του πακέτου να υποστηρίξει διαχειριστικές λειτουργίες στον κόσμο του OSI μέσω των πρωτοκόλλων διαχείρισης του OSI μοντέλου δεν οφείλεται στην δομή του αλλά στην έλλειψη σχετικών προϊόντων δικτυακών υπηρεσιών (X.400, X.500, FTAM) οι οποίες να ενσωματώνουν τους αντίστοιχους agents. Το ίδιο το Open View διαθέτει ενσωματωμένους CMIP [53] και CMOT [52] servers σύμφωνα με τις αντίστοιχες συστάσεις του OSI μοντέλου για το management. Αναμένεται ότι σύντομα θα υπάρξουν προϊόντα διαφόρων κατασκευαστών που να υποστηρίζουν τα προαναφερόμενα OSI πρωτόκολλα.

Στην συνέχεια θα περιγραφούν τα πρωτόκολλα SNMP, MIB και SMI και θα δοθούν ορισμένα στοιχεία για τον τρόπο λειτουργίας του Openview στο δίκτυο Αριάδνη.

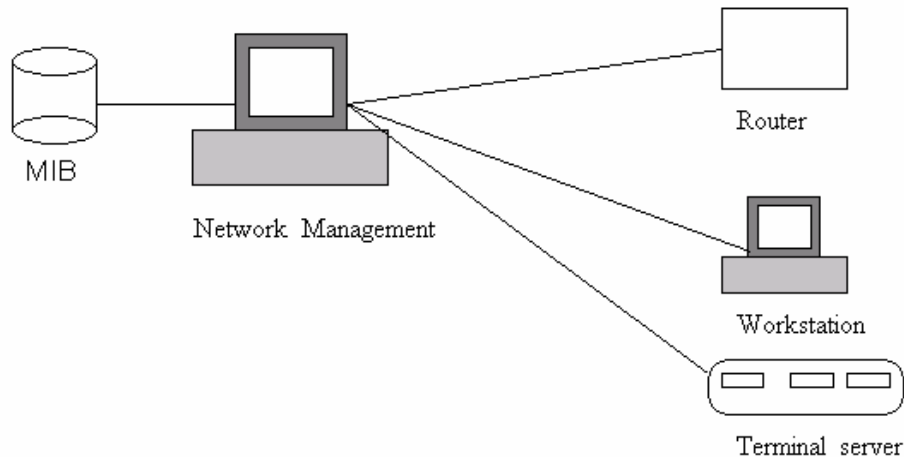
2. Το πρωτόκολλο SNMP

2.1. Γενικά

Το SNMP (Simple Network Management Protocol) αναπτύχθηκε από την εκφρασμένη ανάγκη των διαχειριστών των διαφόρων Internets οι οποίοι χρειαζόνταν μία απλή και εύκολη στην χρήση και ανάπτυξη διαδικασία για την επικοινωνία με τα διάφορα διασκορπισμένα συστήματα ώστε να μπορούν από το κέντρο διαχείρισης να ασκούν εποπτεία. Το SNMP αναπτύχθηκε και λειτουργεί στο Internet για να δημιουργήσει το υπόβαθρο (standard) πάνω στο οποίο θα στηρίζονταν η ανάπτυξη δικτυακών υπηρεσιών αλλά και εργαλείων διαχείρισης που θα έκαναν τη λειτουργία του Internet περισσότερο σταθερή και συνεκτική.

Ο σχεδιασμός του SNMP στηρίχτηκε στο πρωτόκολλο το οποίο είχε αναπτυχθεί προηγουμένως για να βοηθήσει στην διαχείριση των routers και το οποίο είναι γνωστό με τα αρχικά SGMP (Simple Gateway Management Protocol) [50]. Το SNMP σχεδιάστηκε για να διαχειρίζεται πληροφορία ελέγχου η οποία ακολουθεί τις περιγραφές SMI και MIB. Οι περιγραφές SMI και MIB έχουν γίνει καθολικά αποδεκτές και χρησιμοποιούνται ως ο κορμός για την ανάπτυξη και του αντίστοιχου πρωτοκόλλου για το OSI δίκτυο.

□ Το μοντέλο ανάπτυξης και λειτουργίας του SNMP παρουσιάζεται στο ακόλουθο σχήμα.



Σύμφωνα με το μοντέλο αυτό ένας ή περισσότεροι SNMP servers ελέγχουν έναν ή περισσότερους SNMP agents. Μέσω του πρωτοκόλλου SNMP το πακέτο διαχείρισης το οποίο είναι εγκατεστημένο στους SNMP servers επικοινωνεί με τους SNMP agents οι οποίοι είναι εγκατεστημένοι στα διάφορα δικτυακά συστήματα (hosts, routers, modems, terminal servers κ.τ.λ). Οι agents παρακολουθούν και συλλέγουν πληροφορίες όπως connections, packets transmitted, packets received και γενικότερα οποιαδήποτε πληροφορία σχετική με το σύστημα στο οποίο λειτουργούν και έχουν προγραμματιστεί να λαμβάνουν. Οι agents παρέχουν την πληροφορία που συλλέγουν με τη χρησιμοποίηση του πρωτοκόλλου SNMP στους servers η οποία καταλήγει τελικά στα NMS (Network Management Station) τα οποία και επεξεργάζονται την πληροφορία. Τα NMS είναι τα πακέτα που συλλέγουν διαχειρίζονται από κεντρικό σημείο την πληροφορία που συγκεντρώνουν μέσω του πρωτοκόλλου SNMP από τους SNMP agents. Τα NMS παρέχουν την πληροφορία που συλλέγουν στους network administrators μέσω του user interface που διαθέτουν. Η πληροφορία παρουσιάζεται στους network administrators σε διάφορες μορφές όπως γραφικές απεικονίσεις, με παρουσίαση της κατάστασης των δικτυακών συστημάτων με εναλλαγές των χρωμάτων σε γραφικό περιβάλλον κ.τ.λ.

□ Το SNMP σχεδιάστηκε για να είναι απλό στην χρήση του και εύκολο στην ανάπτυξη του από τους κατασκευαστές προϊόντων δικτυακών εφαρμογών. Για το λόγο αυτό μειώνονται στο ελάχιστο οι λειτουργίες διαχείρισης που ασκεί ο SNMP server, όπως θα φανεί και στην συνέχεια, στην ανάλυση της λειτουργίας του server. Ο κώδικας προγράμματος ο οποίος θα πρέπει να προστεθεί στα δικτυακά συστήματα για την υποστήριξη του πρωτοκόλλου είναι περιορισμένος. Επιπλέον αυτού υπάρχει μεγάλη δυνατότητα αύξησης των λειτουργιών στις οποίες ασκείται εποπτεία μέσω του πρωτοκόλλου χωρίς την αλλαγή του κώδικα, απλά με κατάλληλες προσθέσεις στην management information base (MIB) αρκεί να ακολουθούνται τα πρότυπα SMI και MIB. Το SNMP είναι πρωτόκολλο στο επίπεδο εφαρμογής και έτσι είναι κατά το δυνατόν ανεξάρτητο από την αρχιτεκτονική των μηχανών στις οποίες έχει εγκατασταθεί. Τα πλεονεκτήματα αυτά του SNMP το έκαναν ιδιαίτερα δημοφιλή για τη

διαχείριση δικτυακών συσκευών στο Internet και οι εταιρίες κατασκευών δικτυακών προϊόντων πολύ γρήγορα ανέπτυξαν προϊόντα που υποστηρίζουν το πρωτόκολλο αυτό.

□ Στο μοντέλο του SNMP όλες οι λειτουργίες του agent είναι είτε ελέγχου είτε αλλαγής λειτουργικών χαρακτηριστικών. Έτσι ένας agent σε ένα απομακρυσμένο δικτυακό σύστημα συνομιλεί με το σύστημα προκειμένου να λάβει (get) ή να αλλάξει (set) τις τιμές διαφόρων μεταβλητών του συστήματος. Η στρατηγική αυτή έχει το πλεονέκτημα ότι περιορίζει τις διαχειριστικές λειτουργίες σε δύο, μία για την απόκτηση της τιμής μίας παραμέτρου και η δεύτερη για την τοποθέτηση μίας παραμέτρου. Παρ' όλα αυτά μέσω του SNMP μπορεί να γίνουν διαχειριστικές λειτουργίες σε οποιοδήποτε βάθος και αν απαιτηθεί.

□

2.2. Προστασία στην επικοινωνία μέσω του SNMP

□

Κάθε υπηρεσία δικτύου θα πρέπει να προστατεύεται από παρεμβολή ανεπιθύμητης πληροφορίας ή από "συνακρόαση" των πακέτων, γεγονός το οποίο δεν είναι επιθυμητό. Η φύση των λειτουργιών του SNMP είναι πολύ σημαντική στην λειτουργική υποστήριξη και διαχείριση του δικτύου. Ο λόγος αυτός επιβάλλει την προστασία της επικοινωνίας του διαχειριστή με τα απομακρυσμένα δικτυακά συστήματα, πολύ περισσότερο όταν η επικοινωνία γίνεται προκειμένου να συντελεστεί αλλαγή στις παραμέτρους λειτουργίας των δικτυακών ή υπολογιστικών συστημάτων. Η προστασία που παρέχει το SNMP όπως ορίζεται στην περιγραφή του έχει δύο επίπεδα. Το πρώτο επίπεδο προστασίας είναι το community name. Ένας κωδικός, κάτι σαν password το οποίο πρέπει να διαθέτει ο SNMP server ώστε να επικοινωνήσει με τους agents. Το δεύτερο επίπεδο προστασίας σχετίζεται με τα permissions τα οποία θέτει ο agent στην πληροφορία των MIB objects. Έτσι υπάρχει η δυνατότητα από την πλευρά του agent να οριστούν οι καταστάσεις READ -ONLY και READ -WRITE. Κάθε μία από αυτές τις καταστάσεις μπορεί να οριστεί σε όλα ή σε μία ομάδα από MIB objects του agent. Η READ ONLY κατάσταση είναι η περισσότερο ακίνδυνη αφού οποιοσδήποτε δικτυακός SNMP server μπορεί μόνο να διαβάσει την πληροφορία των agents. Στην κατάσταση αυτή το community name είναι σημαντικό να υπάρχει εφ' όσον δεν είναι επιθυμητό η πληροφορία να διατίθεται ελεύθερα σε κάθε SNMP server. Η κατάσταση READ WRITE είναι περισσότερο σημαντική γιατί η χρήση επηρεάζει ολόκληρη τη λειτουργία των δικτυακών ή υπολογιστικών συστημάτων. Στην κατάσταση αυτή το community name είναι επιβεβλημένο να υπάρχει. Η περιγραφή του SNMP δεν απαιτεί την κωδικοποίηση της μεταφερόμενης από αυτό πληροφορίας και για το λόγο αυτόν είναι σχετικά ανασφαλές. Η αδυναμία αυτή του πρωτοκόλλου επιβάλλει σε ορισμένες εταιρίες να παρέχουν πρόσθετη ασφάλεια στο SNMP (server or agent) το οποίο τοποθετούν στα προϊόντα τους, τα οποία προορίζονται κυρίως για ειδικές εφαρμογές π.χ στρατιωτικές, βιομηχανικές κ.λ.π.

2.3. Απαιτήσεις του SNMP

Η επικοινωνία μέσω του SNMP είναι connectionless και γίνεται μέσω του datagram service του Internet, με τη χρησιμοποίηση του πρωτοκόλλου UDP (User Datagram Protocol). Για τη λειτουργία του SNMP έχουν δεσμευτεί δύο internet πόρτες η 161 και η 162. Η internet πόρτα 161 χρησιμοποιείται για την επικοινωνία του SNMP server με τους SNMP agents όταν ο SNMP server ξεκινάει την επικοινωνία. Ο SNMP agent έχει τη δυνατότητα να στείλει πληροφορία στον server χωρίς να έχει προσκληθεί από αυτόν. Η πληροφορία αυτή στέλνεται στις περιπτώσεις που έχει γίνει κάποια σημαντική αλλαγή στην κατάστασή του. Τέτοιες αλλαγές είναι μία εντολή shutdown, ένα υπερφόρτωμα των δίσκων, ή οτιδήποτε έχει ορίσει ο χρήστης. Η φύση του connectionless transport πρωτοκόλλου δίνει μεγαλύτερη ανεξαρτησία στην επικοινωνία του server με τους agents. Κάθε εντολή του server και κάθε απάντηση του agent μεταφέρεται με ένα μόνο πακέτο UDP. Έτσι κερδίζεται χρόνος και αμεσότητα αφού η ανταλλαγή μηνυμάτων δεν πρέπει να συγχρονίζεται με το άνοιγμα μίας connection. Τα SNMP πακέτα είναι κωδικοποιημένα με τους κανόνες του ASN.1. [51] Το πακέτο SNMP αποτελείται από ένα πεδίο με προσδιοριστεί της έκδοσης του agent ή του server όπου χρησιμοποιείται ένα πεδίο που περιέχει το community name και ένα PDU (Protocol Data Unit). Η πληροφορία η οποία μεταφέρεται από το PDU δεν πρέπει να υπερβαίνει τα 484 octets. Το πρωτόκολλο περιγράφει τους ακόλουθους πέντε τύπους PDU, GetRequest-PDU, GetNextRequest-PDU, GetResponse-PDU, SetRequest-PDU και το Trap_PDU.

2.4. Λειτουργίες του πρωτοκόλλου κατά την ανταλλαγή πληροφοριών

Η διαδικασία που ακολουθείται για την αποστολή πακέτου πληροφορίας SNMP μεταξύ server - agent περιγράφεται στην συνέχεια.

□1. Καταρχήν ο server συντάσσει ένα κατάλληλο πακέτο PDU το οποίο προσδιορίζει την απαίτησή του από τον agent (π.χ GetRequest-PDU) χρησιμοποιώντας την κωδικοποίηση ASN.1.

2. Ο server προσθέτει στο PDU το community name και τις διευθύνσεις του server και του agent (αποστολής και προορισμού). Έπειτα περνάει το πακέτο στο επίπεδο μεταφοράς.

3. Το επίπεδο μεταφοράς τοποθετεί το πακέτο στη ουρά για έξοδο και το στέλνει στον απέναντι σταθμό (server ή agent) με τη χρησιμοποίηση του πρωτοκόλλου UDP.

□ Η διαδικασία που ακολουθείται κατά τη λήψη του πακέτου πληροφορίας SNMP μεταξύ server - agent είναι όπως περιγράφεται παρακάτω.

□1. Κατά τη λήψη του εισερχόμενου datagram πακέτου γίνεται έλεγχος αν το

πακέτο είναι έγκυρο. Αν δηλαδή ακολουθεί τους κανόνες του SNMP και ASN.1. Αν το πακέτο δεν περάσει αυτό τον πρώτο έλεγχο απορρίπτεται αμέσως.

□**2.** Μετά γίνεται έλεγχος αν η έκδοση του SNMP στο αντίστοιχο πεδίο του πακέτου είναι έγκυρη. Αν ο έλεγχος δεν περάσει το πακέτο απορρίπτεται.

□**3.** Στο στάδιο αυτό γίνεται έλεγχος του πεδίου community name. Αν το community name είναι έγκυρο η διαδικασία μεταβαίνει στο επόμενο στάδιο.

□**4.** Αν το πακέτο φτάσει στο επίπεδο αυτό τότε είναι έτοιμο για περαιτέρω επεξεργασία. Το πακέτο ανοίγεται διαβάζεται και εκτελούνται με τη σειρά οι εντολές που περιέχει. Το αποτέλεσμα των ενεργειών που γίνονται κωδικοποιούνται σε άλλο PDU και επιστρέφουν ως απάντηση όπως ήδη έχει περιγραφεί προηγούμενα.

2.5. SNMP PDUs

Το SNMP υποστηρίζει πέντε τύπους πακέτων PDU (Packet Data Unit), GetRequest-PDU, GetNextRequest-PDU, SetRequest-PDU, GetResponse-PDU, και Trap-PDU. Ο περιορισμένος αριθμός των απαιτούμενων για τη λειτουργία του πρωτοκόλλου πακέτων δίνει απλότητα και αξιοπιστία και κάνει το SNMP ιδιαίτερα δημοφιλές στους χρήστες.

□

GetRequest-PDU

□ GetRequest-PDU ::=

□ [0]

□ IMPLICIT SEQUENCE {

□ request-id RequestID,

□ error-status ErrorStatus,

□ error-index ErrorIndex,

□ variable-bindings VarBindList

□ }

□ Στο πακέτο GetRequest-PDU το οποίο στέλνει ο SNMP server, ο agent απαντά με ένα πακέτο GetResponse-PDU ως εξής:

□**1.** Αν κανένα από τα ονομαζόμενα MIB object στο πεδίο variable-bindings δε διατίθενται από τον agent ή δεν επιτρέπεται η χρήση του στον server τότε ο agent απαντά με ένα πακέτο GetResponse-PDU του οποίου το πεδίο error-status έχει την τιμή noSuchName και στο πεδίο error-index καταγράφονται τα MIB objects οι τιμές των οποίων ζητήθηκαν από τον server.

□**2.** Αν το μέγεθος του πακέτου GetRequest-PDU είναι πολύ μεγάλο τότε ο agent απαντά με ένα πακέτο GetResponse-PDU του οποίου το πεδίο error-status έχει την τιμή tooBig και στο πεδίο error-index καταγράφονται τα MIB objects.

□ **3.** Αν ο agent μπορεί να ανταποκριθεί στην πληροφορία που ζητά ο server τότε απαντάει με ένα GetResponse-PDU όπου στο πεδίο variable-bindings δίνονται οι τιμές των MIB objects το πεδίο error-status έχει την τιμή noError, το πεδίο error-index την τιμή 0 και το πεδίο request-id έχει την τιμή του request-id του πακέτου GetRequest-PDU.

□

□ **GetNext-PDU**

□ Το πακέτο GetNextRequest-PDU είναι παρόμοιο και έχει την ίδια συμπεριφορά με το πακέτο GetRequest-PDU. Η διαφορά βρίσκεται στον τύπο του PDU.

□

GetNextRequest-PDU ::=

□ [1]
□ IMPLICIT SEQUENCE {
□ request-id RequestID,
□ error-status ErrorStatus,
□ error-index ErrorIndex,
□ variable-bindings VarBindList
□ }

□ **GetResponse-PDU**

□ GetResponse-PDU ::=

□ [2]
□ IMPLICIT SEQUENCE {
□ request-id RequestID,
□ error-status ErrorStatus,
□ error-index ErrorIndex,
□ variable-bindings VarBindList
□ }

□ Το πακέτο αυτό δημιουργείται μόνο σαν απάντηση στα πακέτα GetRequest-PDU, GetNextRequest-PDU και SetRequest-PDU. Η συμπεριφορά και οι τιμές των πεδίων που παίρνει αναφέρονται στην περιγραφή των υπόλοιπων πακέτων.

□

□ **SetRequest-PDU**

□ SetRequest-PDU ::=

□ [3]
□ IMPLICIT SEQUENCE {
□ request-id RequestID,
□ error-status ErrorStatus,
□ error-index ErrorIndex,
□ variable-bindings VarBindList
□ }

□

□ Το πακέτο αυτό εκδίδεται από τον server στην προσπάθεια για αλλαγή των τιμών των MIB objects.

□ **1.** Αν τα MIB objects τα οποία περιέχονται στο πεδίο variable-bindings δεν

είναι διαθέσιμα για τη λειτουργία set του SNMP από τον agent τότε στέλνεται για απάντηση ένα πακέτο GetResponse-PDU του οποίου το πεδίο error-status έχει την τιμή noSuchName και το πεδίο error-index περιέχει τα MIB objects για τα οποία ζητήθηκε η λειτουργία set.

□**2.** Αν στα MIB objects για τα οποία ζητήθηκε η λειτουργία set δίνεται τιμή που δε συμφωνεί με την περιγραφή SMI τότε ο agent επιστρέφει ένα πακέτο GetResponse-PDU με το πεδίο error-status να έχει τιμή badValue.

□**3.** Αν ο agent μπορεί να ανταποκριθεί στην απαίτηση του server τότε πραγματοποιείται η αλλαγή των objects και επιστρέφεται πακέτο GetResponse-PDU με τα πεδία, error-status να έχει την τιμή noError και το πεδίο error-index την τιμή 0.

□

□**Trap_PDU**

□ Το Trap-PDU δημιουργείται από τον agent για να αναγγείλει στον server μία επείγουσα κατάσταση.

□ Η δομή του Trap-PDU είναι.

□

Trap-PDU ::=

□ [4]

```
□ IMPLICIT SEQUENCE {
□ enterprise          OBJECT IDENTIFIER,
□ agent-addr         NetworkAddress,
□ generic-trap       INTEGER {
□                     coldStart(0),
□                     warmStart(1),
□                     linkDown(2),
□                     linkUp(3),
□                     authenticationFailure(4)
□                     egpNeighbourLoss(5),
□                     enterpriseSpecific(6)
□                     },
□ specific-trap      INTEGER,
□ time-stamp         TimeTicks,
□ variable-bindings  VarBindList
□ }
```

□

□ Το Trap-PDU στέλνεται από SNMP agent κάποιας δικτυακής οντότητας κάθε φορά που η κατάσταση στο δικτυακό σύστημα αλλάζει. Η αιτία αποστολής του trap προσδιορίζεται στο πεδίο generic-trap. Αν το trap εκτελεί μία πρόσθετη λειτουργία η οποία δεν περιγράφεται στο SNMP τότε η πληροφορία αυτή δίνεται στο πεδίο specific-trap. Το SNMP ορίζει τις ακόλουθες τιμές στο πεδίο generic-trap.

□

□**coldstart(0):** Το δικτυακό σύστημα μπαίνει ξανά σε δικτυακή λειτουργία, είναι προσπελάσιμο από το SNMP και δηλώνει ότι έχουν γίνει αλλαγές από την προηγούμενη κατάσταση όσον αφορά το configuration του agent.

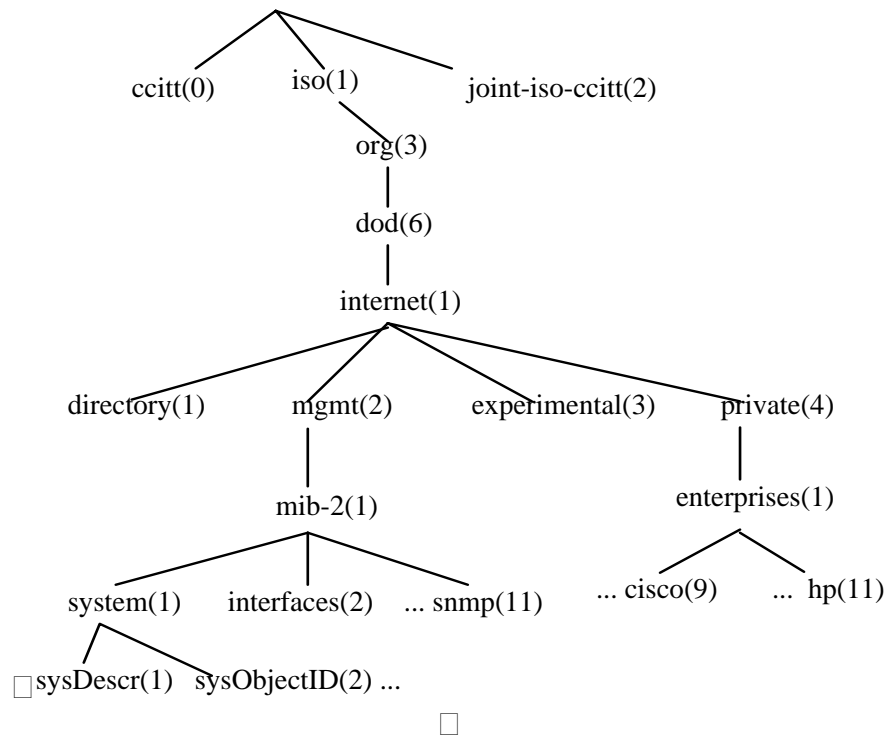
-
- warmstart(1):** Το δικτυακό σύστημα ανακοινώνει την παρουσία του στον SNMP server και δηλώνει ότι δεν έχει γίνει αλλαγή στο configuration του agent.
-
- linkdown(2):** Ενημερώνεται ο server ότι κάποιο από τα communication links έχει πρόβλημα.
-
- linkup(3):** Ενημερώνει ότι τα communication links για τα οποία ο server είχε προηγουμένως ειδοποιηθεί ότι ήταν σε κατάσταση failure έχει αποκατασταθεί η λειτουργία τους.
-

3. Management Information Base και Structure Management Information

3.1. Γενικά

- Σε κάθε δικτυακό σύστημα το οποίο περιέχει SNMP agent ο κατασκευαστής έχει ορίσει λειτουργίες και ελέγχους οι οποίες θα υποστηρίζονται από το SNMP και έχει θέσει το κατάλληλο software για την υποστήριξή τους. Για να υποστηρίζονται οι λειτουργίες αυτές από το SNMP θα πρέπει απαραίτητα να ακολουθούν το πρότυπο που ορίζεται ως SMI (Structure Management Information). Το SMI περιγράφει μια σειρά από κανόνες για τον τρόπο που πρέπει να ορίζονται οι διάφορες λειτουργίες ώστε το πρωτόκολλο διαχείρισης να μπορεί να χειριστεί την πληροφορία που παρέχουν. Οι λειτουργίες αυτές μπορεί να είναι από την παρακολούθηση των πακέτων που περνούν από το δικτυακό σύστημα μέχρι την παρακολούθηση και αλλαγή των routing tables. Ο αριθμός των λειτουργιών αυτών όπως θα φανεί στην συνέχεια δεν μπορεί να υπολογιστεί και είναι διαρκώς επεκτάσιμος, αφού κάθε κατασκευαστής μπορεί να σχεδιάσει και να πραγματοποιήσει λειτουργίες ειδικά για τα προϊόντα του. Κάθε μία από αυτές τις λειτουργίες πρέπει να είναι μονοδιάστατα ορισμένη ώστε να μην υπάρχει περίπτωση να υπάρχουν τέτοιες διαφορετικές λειτουργίες με το ίδιο όνομα ορισμένες, ή να έχει οριστεί η ίδια λειτουργία με περισσότερα από ένα διαφορετικά ονόματα. Όλες αυτές οι λειτουργίες συστήνουν την Management Information Base (MIB). Μία λειτουργία αναφέρεται ως MIB object. Αρχικά στον χώρο του Internet αναπτύχθηκε από τον οργανισμό Internet Activities Board η πρώτη management information base γνωστή ως MIB-I η οποία έγινε πολύ σύντομα ευρύτερα αποδεκτή. Αργότερα αναπτύχθηκε μία νέα βάση η οποία είναι επέκταση της πρώτης και ονομάστηκε MIB-II.
- Τα MIB objects οργανώνονται με ιεραρχική δομή δέντρου. Κάθε κλάδος του οποίου έχει μοναδικό όνομα και μοναδικό αριθμητικό προσδιοριστή. Τα κατώτερα επίπεδα του ιεραρχικού δέντρου καταλαμβάνονται από τα MIB objects. Τα ονόματα των ενδιάμεσων κλάδων της δομής του δέντρου προσδιορίζουν και ομαδοποιούν τα MIB objects τα οποία βρίσκονται κάτω από αυτούς. Η ιεραρχική δομή ενός δέντρου από MIB objects

παρουσιάζεται στο ακόλουθο σχήμα.



Όπως φαίνεται από τη δεντρική δομή τρεις είναι οι βασικοί κόμβοι:

□ **ccitt(0)**: Διαχειρίζεται από τον οργανισμό CCITT.

□ **iso(1)**: Διαχειρίζεται από τον οργανισμό ISO.

□ **joint-iso-ccitt(2)**: Διαχειρίζεται από κοινού από τους οργανισμούς ISO και CCITT.

□ Κάτω από τον κόμβο iso(1) έχει οριστεί ένας κόμβος ο org(3) στον οποίο ανήκουν άλλοι διεθνείς οργανισμοί όπως το Internet (internet(1)). Υποκατάλογος της περιοχής που ορίζεται από το org(3) είναι η ομάδα των MIB objects που διαχειρίζονται από το U.S Department of Defense, dod(6). Στην διαχειριστική περιοχή του dod(6) ανήκει ο οργανισμός Internet με το internet(1). Ακολουθώντας τη δομή του δέντρου από πάνω προς τα κάτω και καταγράφοντας τους αριθμούς που ακολουθούν τα ονόματα των κόμβων προκύπτει το μοναδικό όνομα (διάνυσμα) των MIB objects. Με τον τρόπο αυτό γίνεται αντιληπτό ότι όλα τα MIB objects τα οποία αναφέρονται στο Internet αρχίζουν με το πρόθεμα: 1.3.6.1 (iso.org.dod.internet).

□ Στην περιοχή του Internet έχουν οριστεί τέσσερις υποπεριοχές για τον προσδιορισμό όλων των MIB objects που έχουν σχεδιαστεί ή πρόκειται να σχεδιαστούν για τον έλεγχο των συστημάτων στον Internet κόσμο. Οι υποπεριοχές αυτές είναι οι ακόλουθες.

□

directory(1): Ο υποκατάλογος αυτός είναι δεσμευμένος να λειτουργήσει στο μέλλον και θα περιέχει πληροφορία για το OSI Directory service και τη λειτουργία του στο Internet.

□

□ **mgmt(2)**: Όλα τα objects τα οποία παράγονται από τον οργανισμό IAB

Internet Activities Board [9] ο οποίος ασκεί επίσημα το διαχειριστικό έλεγχο βρίσκονται κάτω από αυτό τον υποκατάλογο.

experimental(3): Όλα τα objects τα οποία δημιουργούνται από την ανάγκη που δημιουργείται έπειτα από την εμπειρία στην λειτουργία και το χειρισμό του υπολογιστικού συστήματος.

private(4): Οι κατασκευαστικές εταιρίες συστημάτων δικτυακών υπηρεσιών οι οποίες κατασκευάζουν τα δικά τους objects για να προσφέρουν μεγαλύτερη διαχειριστική ευελιξία στον έλεγχο των συστημάτων τους μέσω του SNMP, έχουν ένα υποκατάλογο κάτω από το δέντρο private(4). Η cisco(9) και η hp(11) είναι δύο από αυτές.

3.2. Δομή του MIB object

Η λειτουργία και το όνομα κάθε MIB object πρέπει να είναι μοναδικά ορισμένα και καταχωρημένα στην MIB βάση. Η δομή όμως κάθε object πρέπει να ακολουθεί την περιγραφή SMI για να υπάρχει συμβατότητα με το SNMP.

Κάθε MIB object έχει τέσσερα πεδία:

OBJECT: Το πεδίο αυτό περιέχει σε μορφή κειμένου (text) το όνομα του object το οποίο ορίζεται και ως OBJECT DESCRIPTOR. Το πεδίο αυτό πρέπει να είναι μοναδικό και να παραπέμπει εννοιολογικά στην λειτουργία του object.

SYNTAX: Στο πεδίο αυτό περιέχεται το πρόγραμμα σε κώδικα ASN.1.

DEFINITION: Το πεδίο αυτό προσδιορίζει τη λειτουργία του object. Το κείμενο αυτό πρέπει να είναι κατανοητό και να περιγράφει περιληπτικά όλες τις δυνατότητες του object.

ACCESS: Το πεδίο αυτό ορίζει τη δυνατότητα πρόσβασης στο object μέσω του SNMP. Οι δυνατότητες που ορίζονται είναι ένα από τα ακόλουθα, read-only, read-write, write-only, ή non-accessible.

STATUS: Ένα από τα mandatory, optional, obsolete.

Παράδειγμα του MIB object iso.org.dod.internet.mgmt.mib-2.system.sysLocation.

Object: sysLocation { system 6 }

Syntax : DisplayString (SIZE (0..255))

Definition: The physical location of this node (e.g., telephone closet, 3rd floor)

Access: read-only

Status: mandatory

4. Περιγραφή της λειτουργίας του προγράμματος διαχείρισης Open View

□4.1. Γενική περιγραφή του Open View

Το Open View παρέχει ένα ολοκληρωμένο περιβάλλον για τη διαχείριση και την εποπτική παρακολούθηση όλων των καταστάσεων σε ένα Internet δίκτυο. Επιπλέον παρέχει ορισμένα proprietary εργαλεία από την κατασκευάστρια εταιρία, για τη διαχείριση ορισμένων υπηρεσιών δικτύων OSI (X.400). Το user interface του προγράμματος παρουσιάζει σε ένα περιβάλλον γραφικών τη σύνθεση του δικτύου με μεγάλη λεπτομέρεια (μέχρι και την οργάνωση του γραφείου στο οποίο είναι τοποθετημένο το δικτυακό ή υπολογιστικό σύστημα) και με εναλλαγές των χρωμάτων των δικτυακών αντικειμένων παρουσιάζει την κατάσταση τους κάθε χρονική στιγμή.

□ Ο σχεδιασμός της συνολικής παρουσίασης του δικτύου μπορεί να γίνει είτε με χρήση του πρωτοκόλλου SNMP για ανίχνευση των δικτυακών συστημάτων τα οποία έχουν SNMP agents, είτε με manual τοποθέτηση των διαφόρων δικτυακών συσκευών αλλά και αντικειμένων τα οποία χρησιμεύουν για τη λεπτομερέστερη αναπαράσταση του δικτύου (χάρτες, κτήρια, αντικείμενα γραφείου κτλ). Το περιβάλλον εργασίας είναι διαλογικό. Υπάρχει όμως η δυνατότητα να γίνει και απ' ευθείας χειρισμός των configuration αρχείων με τη χρήση ενός editor. Το περιβάλλον εργασίας γίνεται έτσι πολύ φιλικό για το διαχειριστή του δικτύου.

□ Η οργάνωση και η παρουσίαση των δικτύων από το Open View γίνεται με τρόπο ιεραρχικό. Ξεκινώντας από έναν αρχικό χάρτη (Map), ο οποίος περιλαμβάνει μία σειρά από υποχάρτες (Submaps), επεκτείνεται σε μία παρουσίαση του δικτύου σε βάθος. Κάθε ένα από αυτά τα submaps παρουσιάζει μέρος του δικτύου. Συνήθως τα submaps περικλείουν δικτυακά συστήματα τα οποία σχετίζονται με κάποιο τρόπο μεταξύ τους (π.χ ανήκουν στο ίδιο IP δίκτυο, ανήκουν στον ίδιο οργανισμό κτλ). Τα διάφορα δικτυακά αντικείμενα τα οποία χρησιμοποιούνται για την αναπαράσταση του δικτυακού χάρτη ορίζονται ως σύμβολα (symbols) και τα οποία μπορούν να είναι hosts, routers, τοπογραφικοί χάρτες κτλ.). Ένας μεγάλος αριθμός συμβόλων υπάρχουν ήδη έτοιμα στις βιβλιοθήκες του Open View, είναι δυνατόν όμως να αναπτυχθούν σύμβολα από το χρήστη για ειδικές εφαρμογές ή για βελτίωση των δυνατοτήτων γραφικής απεικόνισης του Open View.

Οι έννοιες Maps, Submaps, Objects και Symbols χρησιμοποιούνται εκτεταμένα στην οργάνωση του Open View.

MAP: Ο χάρτης του δικτύου αποτελείται από μία συλλογή από submaps τα οποία παρέχουν μία γραφική και ιεραρχική αναπαράσταση της οργάνωσης του δικτύου. Ο χάρτης είναι το ανώτερο επίπεδο της ιεραρχίας κάτω από το οποίο συγκροτείται το οικοδόμημα της αναπαράστασης του δικτύου. Το Open View έχει τη δυνατότητα να διαχειριστεί ένα χάρτη κάθε φορά, μπορεί όμως να οριστούν περισσότεροι από ένας χάρτες οι οποίοι υποστηρίζουν διαφορετικές εφαρμογές και να καλούνται όταν χρειάζεται

να τρέχουν οι εφαρμογές αυτές.

□

SUBMAP: Κάθε υποχάρτης δίνει μία ιδιαίτερη οπτική και λογική περιγραφή μέρους του δικτύου. Κατά την κλήση του προγράμματος καλείται αυτόματα ο πρώτος υποχάρτης πάνω στον οποίο τοποθετούνται υπό μορφή συμβόλων τα δικτυακά αντικείμενα τα οποία μπορεί να παραπέμπουν σε άλλους υποχάρτες. Η δημιουργία περισσότερων από έναν υποχάρτων είναι απαραίτητη για καλύτερη παρουσίαση του δικτύου.

□

□ **SYMBOL:** Τα σύμβολα είναι η γραφική αναπαράσταση των αντικειμένων όπως παρουσιάζονται στους υποχάρτες. Δύο διαφορετικά σύμβολα μπορούν να αναπαριστούν ένα αντικείμενο. Σε κάθε σύμβολο μπορεί να οριστούν οι χρωματικές αποχρώσεις οι οποίες περιγράφουν την κατάσταση του αντικειμένου. Επιπλέον μπορούν να οριστούν και οι στάθμες των συμβάντων που πρέπει να συμβούν στο αντικείμενο για να αλλάξει χρωματική κατάσταση το σύμβολο. Κάθε σύμβολο τέλος μπορεί να συμπεριφερθεί με δύο διαφορετικούς τρόπους. Κατά τον πρώτο τρόπο είναι δυνατόν με την κλήση του συμβόλου, το σύμβολο αυτό να ανοίγει έναν υποχάρτη ο οποίος δίνει αναλυτικότερη παρουσίαση του αντικειμένου. Κατά το δεύτερο τρόπο είναι δυνατόν με την κλήση του συμβόλου να εκτελείται κάποια λειτουργία. Τέτοιες λειτουργίες είναι, ο έλεγχος κάποιας τιμής ενός μεγέθους σε απομακρυσμένο σύστημα, η εκτέλεση μίας εντολής κτλ.

□

□ **OBJECT:** Αντικείμενο θεωρείται κάθε οντότητα λογική ή φυσική. Οτιδήποτε μπορεί να είναι αξιοποιήσιμο υπό οποιοδήποτε τρόπο (π.χ. ένας τοπογραφικός χάρτης) και χρειάζεται περιγραφή. Ένα αντικείμενο μπορεί να είναι μία πόρτα RS 232, ένα δίκτυο, μία κάρτα ethernet, ένα σύνολο από PCs ή μία λειτουργία mail.

Κάθε object περιέχει μία ομάδα από χαρακτηριστικά τα οποία προσδιορίζουν τον τύπο του αντικειμένου, τις λειτουργίες του και γενικότερα ότι έχει σχέση με τη φύση του αντικειμένου. Ιδιαίτερα χαρακτηριστικά των αντικειμένων είναι το όνομά του, η διεύθυνσή του στο internet, η θέση του στο κτίριο, μία συνοπτική περιγραφή κτλ.

□

□ 4.2.

Έλεγχος του δικτύου

□ Ο έλεγχος του δικτύου γίνεται με την ανταλλαγή συμβάντων (events) από τον SNMP manager ή τον SNMP agent αν τουλάχιστον μία από τις ακόλουθες συνθήκες ισχύει.

□

□ 1. Ένα κατώφλι ορίου της κατάστασης του δικτυακού αντικειμένου έχει ξεπεραστεί.

□

2. Η τοπολογία του δικτύου έχει αλλάξει.

□

3. Ένα μήνυμα πληροφορίας ή ένα μήνυμα λάθους έχει συμβεί.

4. Η κατάσταση ενός αντικειμένου έχει αλλάξει.

5. Έχει γίνει αλλαγή στο μηχανικό ή στο λογισμικό μέρος του αντικειμένου.

Τα συμβάντα δημιουργούνται στα δικτυακά συστήματα τα οποία υπόκεινται σε διαχειριστικό έλεγχο και μέσω των SNMP agents μεταδίδονται στο κεντρικό σύστημα. Όλα τα συμβάντα μαζεύονται στο κεντρικό σύστημα από το δαίμονα trapd και αποθηκεύονται στο αρχείο **/usr/OV/log/trapd.log**. Η πρόσβαση στο αρχείο για την αξιοποίηση της πληροφορίας γίνεται μέσω του διαλογικού interface events το οποίο καλείται αυτόματα κατά την κλήση του προγράμματος Open View.

Άλλες διαδικασίες που πρέπει να τρέχουν στο background είναι:

netmon: Ο δαίμονας netmon περιοδικά διενεργεί κλήσεις στους SNMP agents προκειμένου να ανακαλύψει αλλαγές στην τοπολογία του δικτύου. Ο δαίμονας αυτός κάνοντας χρήση των λειτουργιών του SNMP μπορεί να ανακαλύπτει ολοένα και περισσότερα δίκτυα με την προϋπόθεση ότι όλοι οι routers που συναντά έχουν SNMP agent και επιτρέπουν στον τοπικό SNMP server να διαβάσει την πληροφορία που έχουν. Επειδή όμως δεν απαντούν όλοι οι routers σε κλήσεις SNMP, τουλάχιστον για ελεύθερη χρήση από τους SNMP servers, δεν είναι δυνατόν και φυσικά ούτε επιθυμητό να απεικονιστεί από το τοπικό σύστημα Open View όλο το δίκτυο του Internet. Όσο μεγαλύτερο είναι το δίκτυο που ελέγχεται, τόσο μεγαλύτερη επιβάρυνση υπάρχει στο υπολογιστικό σύστημα που τρέχει την εφαρμογή. Η έκταση χρήσης του netmon περιορίζεται από ένα configuration αρχείο. Το αρχείο αυτό **/usr/OV/conf/nodelist** περιέχει τα αντικείμενα (δικτυακά συστήματα που υποστηρίζουν το SNMP) τα οποία ελέγχονται από το παρόν Network Management, προκειμένου αυτό να βελτιστοποιήσει τη διαδικασία ανακάλυψης νέων σταθμών.

Το αρχείο **/usr/OV/conf/nodelist** έχει τη μορφή:

```
#
# a sample of nodelist file
#
143.233.10.10 #central cisco
143.233.10.3 #dias.servicenet.ariadne-t.gr
143.233.240.2 #cisco clns
```

Το configuration αρχείο για τον netmon δαίμονα είναι το **/usr/OV/lrf/netmon.lrf**. Το αρχείο αυτό διαβάζεται κάθε φορά που ξεκινάει ο δαίμονας και έχει την ακόλουθη μορφή.

```
#
# netmon start-up file
#
netmon:/usr/OV/bin/netmon
```

```
OVs_YES_START:  ovtopmd,trapd,ovwdb:  -P  -s  /usr/OV/conf/nodelist:\nOVs_WELL_BEHAVE:15
```

□

ovtopmd: Η διαδικασία αυτή δέχεται την πληροφορία που συλλέγεται από τον netmon δαίμονα και πραγματοποιεί τις απαραίτητες αλλαγές στην τοπολογία των αντικειμένων κατά την αναπαράσταση του Δικτύου από το Open View. Οι αλλαγές καταγράφονται στο αρχείο **/usr/OV/databases/openview/topo**. Το πρόγραμμα **ovtopmd** διαβάζει το αρχείο αυτό κάθε φορά που τρέχει. Αν το αρχείο αυτό δεν υπάρχει το δημιουργεί.

ovwdb: Ο δαίμονας αυτός ελέγχει τη βάση των δικτυακών αντικειμένων. Η βάση αυτή διατηρεί πληροφορία σχετικά με τα αντικείμενα τα οποία ανακαλύπτει ο netmon δαίμονας. Όταν ο netmon ανακαλύψει κάποια αλλαγή στο δίκτυο καλείται ο δαίμονας ovtopmd για να αξιοποιήσει την πληροφορία αλλάζοντας την τοπολογία του χάρτη. Έπειτα καλείται το πρόγραμμα ovwdb για να πραγματοποιήσει τις αλλαγές στη βάση των δικτυακών αντικειμένων. Το Open View κάνει σημαντική χρήση της φυσικής μνήμης κατά την παρουσίαση των δικτυακών στοιχείων στο user interface. Αυτό δημιουργεί υπερφόρτωση στο υπολογιστικό σύστημα που τρέχει την εφαρμογή. Υπάρχει δυνατότητα να οριστεί ο μέγιστος αριθμός των δικτυακών αντικειμένων τα οποία καταχωρούνται στην πραγματική μνήμη ώστε να περιοριστεί η υπερφόρτωση του υπολογιστικού συστήματος. Κάθε δικτυακό ή υπολογιστικό σύστημα καταχωρείται στην βάση αυτή με κατά μέσο όρο τρία αντικείμενα. Έτσι υπολογίζοντας τον αριθμό των δικτυακών συστημάτων κατά προσέγγιση είναι δυνατό να εξαχθεί ο αριθμός των δικτυακών αντικειμένων στην βάση αυτή. Στο configuration file του ovwdb **/usr/OV/lrf/ovwdb.lrf** μπορεί να οριστεί με την παράμετρο -n ο μέγιστος αριθμός των αντικειμένων στην βάση. Αν το όριο αυτό ξεπεραστεί τότε τα νέα αντικείμενα που ανακαλύπτονται θα καταχωρούνται μόνο στον δίσκο του συστήματος.

□

Το αρχείο /usr/OV/lrf/ovwdb.lrf έχει τη μορφή

```
#\n#  ovwdb start-up file\n#\novwdb:/usr/OV/bin/ovwdb:\nOVs_YES_START::-O -s 5000:OVw_WELL_BEHAVED:15
```

□**snmpcollect:** Ο δαίμονας snmpcollect μαζεύει όλη την πληροφορία που αποστέλλεται από τους καταναμημένους SNMP agents και ελέγχει αν η πληροφορία αυτή είναι αξιοποιήσιμη οπότε την στέλνει στον δαίμονα trapd για επόμενη αξιοποίησή της. Η πρόσβαση στους SNMP agents καθορίζεται όπως περιγράφηκε προηγουμένως κατά κύριο λόγο από το community name. Πολλοί από τους SNMP agents που είναι εγκατεστημένοι στα διάφορα υπολογιστικά συστήματα έχουν community name τη λέξη " public " ή επιτρέπουν (δεν έχουν community name) την άντληση πληροφορίας ελεύθερα από αυτούς. Συνηθίζεται όμως στα δικτυακά συστήματα να τοποθετούνται community names όταν η πληροφορία που διαθέτουν δεν είναι

επιθυμητό να γίνει γνωστή ελεύθερα στο " διαχειριστικό " κοινό. Το αρχείο που περιέχει την πληροφορία για τους SNMP agents και τα community names είναι το **/usr/OV/conf/ovsnmp.conf**.

□

4.3. Προγράμματα που τρέχουν μέσω του user interface OVW

Το user interface του Open View είναι το πρόγραμμα onw. Κάθε φορά που τρέχει το onw διαβάζει το αρχείο **/usr/OV/conf/ovsuf**. Το αρχείο αυτό ενημερώνει το onw για τις απαραίτητες διαδικασίες που πρέπει να τρέχουν στο background. Μετά την εκκίνηση του onw και την εμφάνιση του user interface ο διαχειριστής του δικτύου είναι σε θέση να χειριστεί ένα μεγάλο αριθμό λειτουργιών προκειμένου να δώσει στον τελικό χρήστη ένα σταθερό και συνεκτικό δικτυακό περιβάλλον.

Τα προγράμματα που τρέχουν μέσω του user interface είναι τα ακόλουθα.

/usr/OV/bin/xnmevents: Το πρόγραμμα αυτό καλείται κάθε φορά που τρέχει το onw, και δημιουργεί ένα ξεχωριστό interface για την παρακολούθηση των συμβάντων στο δίκτυο και το οποίο παραμένει ενεργό μέχρι το κλείσιμο του onw. Το πρόγραμμα αυτό αντλεί την πληροφορία από το αρχείο **/usr/OV/log/trap.log** και χρησιμοποιεί το αρχείο **/usr/OV/log/xnmevents** για την αποθήκευση εσωτερικών στοιχείων όπως την ημερομηνία και την ώρα της τελευταίας εξόδου από το πρόγραμμα καθώς και τα ανεπιβεβαίωτα συμβάντα. Η πληροφορία αυτή χρησιμοποιείται κατά την επανεκκίνηση του προγράμματος.

/usr/OV/bin/xnmsnmpconf: Το πρόγραμμα αυτό καλείται από την επιλογή Option: SNMP Configuration του onw. Το xnmsnmpconf βοηθά το χρήστη να προσδιορίσει τα community names των SNMP agents, τους οποίους ελέγχει για συλλογή πληροφοριών, και τους χρονιστές σύμφωνα με τους οποίους επιχειρείται η επαφή με τους agents.

/usr/OV/bin/xnmloadmib: Όταν χρειάζεται να συγκεντρωθεί πληροφορία από MIB objects τα οποία δεν υπάρχουν ήδη στο Open View γιατί αυτά είναι private του κατασκευαστή τότε πρέπει αυτά τα MIB objects να φορτωθούν με τη λειτουργία Option:Load/Unload MIBs -> SNMP Η λειτουργία αυτή πραγματοποιείται μέσω του προγράμματος xnmloadmib.

/usr/OV/bin/xnmbrowser: Το πρόγραμμα αυτό καλείται με την επιλογή Monitor:MIB values -> Browse MIB του onw, και παρουσιάζει την πληροφορία που περιέχεται στα MIB objects. Με τη λειτουργία αυτή ο χρήστης μπορεί να αντλήσει ή να θέσει τιμές στα MIB objects.

/usr/OV/bin/xnmgraph: Το πρόγραμμα αυτό καλείται για τη γραφική αναπαράσταση τις πληροφορίας από τα MIBs.

4.4. Αντιστοίχιση δικτυακών συστημάτων με Open View symbols

Κάθε φορά που το Open View ανακαλύπτει ένα δικτυακό σύστημα το οποίο υποστηρίζει SNMP και το οποίο περιέχει τα MIB objects κατά τα MIB-I και MIB-II προκειμένου να το τοποθετήσει στον χάρτη ελέγχει το MIB object SysObjectID. Το MIB object SysObjectID περιέχει πληροφορίες για τον τύπο

του δικτυακού συστήματος και για τον κατασκευαστή του. Από την πληροφορία αυτή συμπεραίνεται αν το δικτυακό σύστημα είναι router ή workstation και έχει κατασκευαστεί από την cisco ή την H/P ή από άλλο κατασκευαστή. Η πληροφορία αυτή είναι χρήσιμη για την επιλογή του συμβόλου το οποίο θα αντιπροσωπεύει το δικτυακό σύστημα στο χάρτη. Δύο αρχεία βοηθούν στην αντιστοίχιση του SysObjectID με τα σύμβολα του Open View

oid_to_sym: Το αρχείο αυτό χρησιμοποιείται από το πρόγραμμα ipmap. Όταν το ipmap προσπαθεί να αντιστοιχήσει ένα δικτυακό σύστημα πρέπει να γνωρίζει τον τύπο του συστήματος (workstation, mini, κτλ.). Η αντιστοίχιση αυτή βρίσκεται στο αρχείο /usr/OV/conf/\$LANG/oid_to_sym του οποίου η μορφή είναι.

```
# HP unix-based Agents
1.3.6.1.4.1.11.2.3.2.2.:Computer:Workstation#300
1.3.6.1.4.1.11.2.3.2.3:Computer:Mini #800
1.3.6.1.4.1.11.2.3.2.5:Computer:Workstation #700
```

Το πρώτο πεδίο του αρχείου αυτού είναι το MIB object (system.SysObjectID). Το δεύτερο πεδίο ορίζει την πρώτη κλάση του συστήματος (computer, router, connector). Το τρίτο πεδίο ορίζει την υποκλάση του συστήματος (workstation, mini).

oid_to_type: Το αρχείο αυτό αντιστοιχεί το SysObjectID με τη συμπεριφορά - λειτουργία του δικτυακού συστήματος στο δίκτυο. Ένα σύστημα μπορεί να είναι router, brige ή workstation. Επιπλέον το αρχείο αυτό προσδιορίζει και τον κατασκευαστή του συστήματος. Το αρχείο /usr/OV/conf/oid_to_type έχει τη μορφή.

```
1.3.6.1.4.1.23.1.1.1:Novel:Novel Lantern
1.3.6..1.4.1.9.1.5:cisco Systems:Cisco Gateway Server:G
1.3.6.1.4.1.26.2.1:Hughes:Hughes LAN Systems Bridge:B
```

□

Το πρώτο πεδίο κάθε εγγραφής στο αρχείο προσδιορίζει το MIB SysObjectID. Το δεύτερο πεδίο προσδιορίζει την κατασκευάστρια εταιρία. Το τρίτο πεδίο περιέχει το όνομα του SNMP agent ο οποίος τρέχει στο δικτυακό σύστημα. Το τέταρτο πεδίο προσδιορίζει τη λειτουργία του συστήματος στο δίκτυο (router, bridge, hub κτλ).

Κατά την πρόσθεση με χειρισμό του χρήστη νέων συμβόλων στις βάσεις του Open View πρέπει να γίνουν αλλαγές στα αρχεία αυτά ώστε τα νέα σύμβολα να εμφανιστούν στους χάρτες. Οι αλλαγές στα αρχεία αυτά γίνονται με τη χρησιμοποίηση editor.

4.5. Βοηθητικά προγράμματα

Αρκετά προγράμματα των οποίων η λειτουργία τους δεν απαιτεί γραφικό περιβάλλον

μπορούν να κληθούν είτε από το user interface είτε κατευθείαν από το shell. Ένα από αυτά τα προγράμματα είναι:

onwperms Το πρόγραμμα αυτό χρησιμοποιείται για να ελεγχθούν τα permissions των αρχείων που παράγονται από το onw.

4.6. Διαχειριστικές ικανότητες από το Open View

Οι δυνατότητες διαχείρισης που έχει το Open View πάνω στα διάφορα δικτυακά αντικείμενα μπορούν να ταξινομηθούν σε τέσσερα επίπεδα.

□

1. Αντικείμενα για τα οποία δεν υπάρχει επικοινωνία μέσω IP. Τα αντικείμενα αυτά δεν μπορούν να διαχειριστούν από το Open View. Τα αντικείμενα αυτά εισέρχονται μόνο με χειρισμό του χρήστη στις λίστες (Submaps) του Open View. Στην κατηγορία αυτή ανήκουν οι X.25 κόμβοι, Novel PCs, τοπογραφικοί χάρτες κτλ.

2. Δικτυακά αντικείμενα τα οποία υποστηρίζουν IP όπως τα υπολογιστικά συστήματα που τρέχουν TCP/IP. Το Open View μπορεί να τα ανακαλύψει στο δικτύου και σε ορισμένες περιπτώσεις να ανακαλύψει και την Internet διεύθυνση τους και να ελέγξει την κατάστασή τους με τη λειτουργία ping.

3. Δικτυακά στοιχεία τα οποία υποστηρίζουν το SNMP. Το Open View έχει τη δυνατότητα εκτός από το να ανακαλύπτει την κατάστασή τους να ενεργεί και ορισμένες λειτουργίες διαχείρισης από τα MIB-I και II που διαθέτουν.

4. Δικτυακά στοιχεία που διαθέτουν SNMP και υποστηρίζουν εκτός από τα στάνταρ MIB-I και II και άλλα private objects από την κατασκευάστρια εταιρία. Το Open View μπορεί να συγκεντρώσει περισσότερες πληροφορίες για αυτά τα δικτυακά στοιχεία αρκεί να διαθέτει στην βάση των MIB τα επιπλέον objects που έχει σχεδιάσει ο κατασκευαστής. Το Open View επιτρέπει στον operator να προσθέσει νέα objects ώστε να ελέγξει τα MIB objects τα οποία έχει θέσει ένας κατασκευαστής στα συστήματά του.

4.7. Το περιβάλλον εργασίας του Open View

Το Open View τρέχει σε περιβάλλον X-windows ή σε περιβάλλον H/P Vue.

Το H/P Vue στηρίζεται στο περιβάλλον X-windows.

Το περιβάλλον εργασίας του Open View έχει την παραθυρική δομή των X-windows.

4.8. Ευκολίες του Open View στην διαχείριση του δικτύου

Η λειτουργία MIB Browse επιτρέπει στον χρήστη να ελέγξει τις τιμές των MIB objects σε ένα ή περισσότερα δικτυακά στοιχεία. Ο τρόπος με τον οποίο παρουσιάζεται η πληροφορία έχει σχέση με την πληροφορία που παρέχει το MIB object. Έτσι όταν παρακολουθείται το traffic το αποτέλεσμα

δίνεται σε μορφή γραφήματος, ενώ όταν ανασύρονται τα routing tables η πληροφορία δίνεται υπό μορφή κειμένου (text).

Στην συνέχεια παρατίθενται ορισμένα στιγμιότυπα της λειτουργίας του Open View. Οι χάρτες που παρουσιάζονται είναι ενδεικτικοί και βοηθούν στην αντίληψη του περιβάλλοντος εργασίας του operator και ορισμένων διαχειριστικών ευκολιών του προγράμματος.

□

