Ομάδα: netmg034
Μιχάλης Παπαδόπουλλος 03114702
Χαράλαμπος Κάρδαρης 03114074

Διαχείριση Δικτύων – Ευφυή Δίκτυα

3η Ομάδα Ασκήσεων

Διαχείριση Δικτύων με το πρωτόκολλο SNMP

http://www.netmode.ntua.gr/courses/undergraduate/netman/documents/RFC1213-MIB.txt

http://www.netmode.ntua.gr/courses/undergraduate/netman/documents/MIB-2.pdf

Άσκηση 1

1. Πραγματοποιήστε τις ακόλουθες μετρήσεις:

- ❑ Με τη βοήθεια του πρωτοκόλλου SNMP (snmpget) υπολογίστε το ρυθμό απόδοσης (**throughput)** σε **bytes/sec,** σε επίπεδο interface, προς και από το interface με IP διεύθυνση 147.102.13.19 του κόμβου maria.netmode.ece.ntua.gr, καθώς και τη χρησιμοποίηση (**utilization, σε ποσοστό %**) στη σύνδεση αυτή. Επίσης, υπολογίστε το ρυθμό απόδοσης (**throughput)** σε **packets/sec,** σε επίπεδο interface, προς και από το ίδιο interface.

  netmg034@maria:~$ snmpwalk -v 2c -c public maria interfaces.ifTable.ifEntry.ifPhysAddress
  IF-MIB::ifPhysAddress.1 = STRING:
  IF-MIB::ifPhysAddress.2 = STRING: 0:c:29:78:c4:14

  Αρχικά ψάχνουμε για τα MAC Addresses των interfaces που υπαρχουν στο server. Παρατηρούμε ότι το interface/2 είναι το μοναδικό με MAC Address την 0:c:29:78:c4:14.

  netmg034@maria:~$ ip a
  1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
  valid_lft forever preferred_lft forever
  2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
  link/ether **00:0c:29:78:c4:14** brd ff:ff:ff:ff:ff:ff
  inet 147.102.13.19/24 brd 147.102.13.255 scope global eth0
  valid_lft forever preferred_lft forever

Τρέχοντας την πιο πάνω εντολή βλέπουμε ότι όντως το node με IP 147.102.13.19 έχει και την αντίστοιχη Ethernet Address του interface/2.

netmg034@maria:~$ snmpget -v 2c -c public maria ipAdEntIfIndex.147.102.13.19
RFC1213-MIB::ipAdEntIfIndex.147.102.13.19 = INTEGER: 2

Βρίσκουμε τις μεταβλητές που μας ενδιαφέρουν:

netmg034@maria:~$ snmpget -v 2c -c public maria interfaces.ifTable.ifEntry.{ifSpeed,if{In,Out}Octets}.2

IF-MIB::ifSpeed.2 = Gauge32: 1000000000
IF-MIB::ifInOctets.2 = Counter32: 418272308
IF-MIB::ifOutOctets.2 = Counter32: 149607298

Συλλεγουμε δεδομένα για δύο χρονικές στιγμές που διαφέρουν κατα 1 λεπτο (60 second).

Το script για ευκολια συλλογης δεδομένων:
netmg034@maria:~/lab3$ cat **minute_command.sh**
#!/usr/bin/bash
SNMP_GET="snmpget -v 2c -c public"
HOSTN="maria"
VAR1=ifInOctets.2

$SNMP_GET $HOSTN $VAR1
sleep 60
$SNMP_GET $HOSTN $VAR1

netmg034@maria:~/lab3$ sh **minute_command.sh**
IF-MIB::**ifInOctets**.2 = Counter32: **478897806**
IF-MIB::**ifInOctets**.2 = Counter32: **478947480**

netmg034@maria:~/lab3$ sh minute_command.sh
IF-MIB::**ifOutOctets**.2 = Counter32: **166712156**
IF-MIB::**ifOutOctets**.2 = Counter32: **166718022**

Για τον υπολογισμό του throughput σε bytes/sec:
IN: (**478947480 - 478897806) / 60 = 827.9**
OUT: (**166718022 – 166712156) / 60 = 97.76666666667**

Για τον υπολογισμό του utilization σε %:
IN: **827.9 * 8 / 1000000000 = 0.0000066232**
OUT: **97.76666666667 * 8 / 1000000000 = 7.8213e-7**

netmg034@maria:~/lab3$ sh minute_command.sh
IF-MIB::**ifInUcastPkts**.2 = Counter32: **6318487**
IF-MIB::**ifInUcastPkts**.2 = Counter32: **6318924**

netmg034@maria:~/lab3$ sh minute_command.sh
IF-MIB::**ifOutUcastPkts**.2 = Counter32: **772839**
IF-MIB::**ifOutUcastPkts**.2 = Counter32: **772902**
Για τον υπολογισμο σε packets/sec:
IN: (**6318924 – 6318487) / 60 = 7.28333333333**
OUT: (**772902 – 772839) / 60 = 1.05**

- Με τη βοήθεια του πρωτοκόλλου SNMP (snmpget) να υπολογιστεί η συνολική **πιθανότητα** απόρριψης πακέτου στο επίπεδο interface προς και από το παραπάνω interface. Να υπολογιστεί επίσης ο **ρυθμός** των παραπάνω απορρίψεων (σε πακέτα που απορρίπτονται ανά δευτερόλεπτο). Συγκρίνατε ποιοτικά τα δύο μεγέθη (δηλ. πιθανότητα και ρυθμό) και αναφέρατε που θα μπορούσε να χρησιμοποιηθεί καλύτερα το καθένα.

```
ifInDiscards OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
      "The number of inbound packets which were chosen
      to be discarded even though no errors had been
      detected to prevent their being deliverable to a
      higher-layer protocol.  One possible reason for
      discarding such a packet could be to free up
      buffer space."
::= { ifEntry 13 }

ifOutDiscards OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
      "The number of outbound packets which were chosen
      to be discarded even though no errors had been
      detected to prevent their being transmitted.  One
      possible reason for discarding such a packet could
      be to free up buffer space."
::= { ifEntry 19 }
```

netmg034@maria:~/lab3$ snmpget -v 2c -c public maria ifInDiscards.2
IF-MIB::**ifInDiscards**.2 = Counter32: **203**

netmg034@maria:~/lab3$ snmpget -v 2c -c public maria ifOutDiscards.2
IF-MIB::**ifOutDiscards**.2 = Counter32: 0

Παρατηρούμε ότι συνολικά εχουν απορριφθεί 203 πακέτα προς το ανώτερο επίπεδο, σε διάστημα Uptime=7705850 sec. Συνεπώς ο ρυθμός καθώς και η πιθανότητα απόρριψης πακέτων είναι σχεδόν “μηδενική” από και προς αλλα επίπεδα.

Οι δύο (2) αυτές μετρικές μπορούν να χρησιμοποίηθουν για αποστολή TRAP μυνηματων απο τον AGENT ώστε ο διαχειριστής να ενημερώνεται για πιθανή δυσλειτουργία ή επιθέσεις στο δίκτυο.

- ❑ Με τη βοήθεια του πρωτοκόλλου SNMP (snmpget) υπολογίστε το ποσοστό των συνολικών λαθών στα IP datagrams που λαμβάνονται από τον κόμβο maria.netmode.ece.ntua.gr.

Οι μεταβλητές που μας ενδιαφέρουν είναι:

```
ipInReceives OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of input datagrams received from
            interfaces, including those received in error."
    ::= { ip 3 }

ipInHdrErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of input datagrams discarded due to
            errors in their IP headers, including bad
            checksums, version number mismatch, other format
            errors, time-to-live exceeded, errors discovered
            in processing their IP options, etc."
    ::= { ip 4 }

ipInAddrErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of input datagrams discarded because
            the IP address in their IP header's destination
            field was not a valid address to be received at
            this entity.  This count includes invalid
            addresses (e.g., 0.0.0.0) and addresses of
            unsupported Classes (e.g., Class E).  For entities
            which are not IP Gateways and therefore do not
            forward datagrams, this counter includes datagrams
            discarded because the destination address was not
            a local address."
    ::= { ip 5 }

ipInUnknownProtos OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of locally-addressed datagrams
            received successfully but discarded because of an
            unknown or unsupported protocol."
    ::= { ip 7 }
```

```
RFC1213-MIB::ipInReceives.0 = Counter32: 1722931
RFC1213-MIB::ipInHdrErrors.0 = Counter32: 0
RFC1213-MIB::ipInAddrErrors.0 = Counter32: 279
RFC1213-MIB::ipInUnknownProtos.0 = Counter32: 0
ErrorRate = (0 + 0 + 279) / 1722931 = 0.00016193336
```

2. (α). Με τη βοήθεια του πρωτοκόλλου SNMP (εντολές snmpget/snmpwalk) περιγράψτε τον πίνακα δρομολόγησης του κόμβου **netmg.netmode.ece.ntua.gr** και του κόμβου **lexmark.netmode.ece.ntua.gr**. Κάθε γραμμή του πίνακα δρομολόγησης πρέπει να είναι στη μορφή **[Destination, Netmask, Gateway]**.

   Μας ενδιαφέρουν τα πεδία:

```
ipRouteTable OBJECT-TYPE
 SYNTAX  SEQUENCE OF IpRouteEntry
 ACCESS  not-accessible
 STATUS  mandatory
 DESCRIPTION
         "This entity's IP Routing table."
 ::= { ip 21 }

ipRouteDest OBJECT-TYPESYNTAX  IpAddress
 ACCESS  read-write
 STATUS  mandatory
 DESCRIPTION
         "The destination IP address of this route.  An
         entry with a value of 0.0.0.0 is considered a
         default route.  Multiple routes to a single
         destination can appear in the table, but access to
         such multiple entries is dependent on the table-
         access mechanisms defined by the network
         management protocol in use."
 ::= { ipRouteEntry 1 }

ipRouteIfIndex OBJECT-TYPE
 SYNTAX  INTEGER
 ACCESS  read-write
 STATUS  mandatory
 DESCRIPTION
         "The index value which uniquely identifies the
         local interface through which the next hop of this
         route should be reached.  The interface identified
         by a particular value of this index is the same
         interface as identified by the same value of
         ifIndex."
 ::= { ipRouteEntry 2 }
```

```
ipRouteMask OBJECT-TYPE
 SYNTAX  IpAddress
 ACCESS  read-write
 STATUS  mandatory
 DESCRIPTION
         "Indicate the mask to be logical-ANDed with the
         destination address before being compared to the
         value in the ipRouteDest field.  For those systems
         that do not support arbitrary subnet masks, an
         agent constructs the value of the ipRouteMask by
         determining whether the value of the correspondent
         ipRouteDest field belong to a class-A, B, or C
         network, and then using one of:

              mask           network
              255.0.0.0      class-A
              255.255.0.0    class-B
              255.255.255.0  class-C

         If the value of the ipRouteDest is 0.0.0.0 (a
         default route), then the mask value is also
         0.0.0.0.  It should be noted that all IP routing
         subsystems implicitly use this mechanism."
 ::= { ipRouteEntry 11 }

$ snmpwalk -v 2c -c public lexmark.netmode.ece.ntua.gr ip.ipRouteTable
RFC1213-MIB::ipRouteDest.0.0.0.0 = IpAddress: 0.0.0.0
RFC1213-MIB::ipRouteDest.147.102.13.0 = IpAddress: 147.102.13.0
RFC1213-MIB::ipRouteIfIndex.0.0.0.0 = INTEGER: 2
RFC1213-MIB::ipRouteIfIndex.147.102.13.0 = INTEGER: 2
RFC1213-MIB::ipRouteMetric1.0.0.0.0 = INTEGER: 1
RFC1213-MIB::ipRouteMetric1.147.102.13.0 = INTEGER: 0
RFC1213-MIB::ipRouteNextHop.0.0.0.0 = IpAddress: 147.102.13.200
RFC1213-MIB::ipRouteNextHop.147.102.13.0 = IpAddress: 0.0.0.0
RFC1213-MIB::ipRouteType.0.0.0.0 = INTEGER: indirect(4)
RFC1213-MIB::ipRouteType.147.102.13.0 = INTEGER: direct(3)
RFC1213-MIB::ipRouteProto.0.0.0.0 = INTEGER: local(2)
RFC1213-MIB::ipRouteProto.147.102.13.0 = INTEGER: local(2)
RFC1213-MIB::ipRouteMask.0.0.0.0 = IpAddress: 0.0.0.0
RFC1213-MIB::ipRouteMask.147.102.13.0 = IpAddress: 255.255.255.0
RFC1213-MIB::ipRouteInfo.0.0.0.0 = OID: SNMPv2-SMI::zeroDotZero
RFC1213-MIB::ipRouteInfo.147.102.13.0 = OID: SNMPv2-SMI::zeroDotZero

$ snmpget -v 2c -c public netmg.netmode.ece.ntua.gr
  interfaces.ifTable.ifEntry.ifPhysAddress.2
IF-MIB::ifPhysAddress.2 = STRING: 0:c:29:1b:e1:a
```

Η οποία MAC αντιστοιχεί στον ίδιο τον κόμβο. Όσον αφορά το default τώρα, θα έχει το default gateway, το οποίο για όλο το υποδίκτυο έχει ip 147.102.13.200 (το βρίσκουμε τρέχωντας ip r στη maria)

	Destination	Netmask	Gateway
1	0.0.0.0	0.0.0.0	**147.102.13.200**
2	147.102.13.0	255.255.255.0	**147.102.13.91**

```
$ snmpwalk -v 2c -c public lexmark.netmode.ece.ntua.gr ip.ipRouteTable
RFC1213-MIB::ipRouteDest.0.0.0.0 = IpAddress: 0.0.0.0
RFC1213-MIB::ipRouteDest.127.0.0.0 = IpAddress: 127.0.0.0
RFC1213-MIB::ipRouteDest.147.102.13.0 = IpAddress: 147.102.13.0
RFC1213-MIB::ipRouteIfIndex.0.0.0.0 = INTEGER: 2
RFC1213-MIB::ipRouteIfIndex.127.0.0.0 = INTEGER: 2
RFC1213-MIB::ipRouteIfIndex.147.102.13.0 = INTEGER: 2
RFC1213-MIB::ipRouteMetric1.0.0.0.0 = INTEGER: 0
RFC1213-MIB::ipRouteMetric1.127.0.0.0 = INTEGER: 0
RFC1213-MIB::ipRouteMetric1.147.102.13.0 = INTEGER: 0
RFC1213-MIB::ipRouteMetric2.0.0.0.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric2.127.0.0.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric2.147.102.13.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric3.0.0.0.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric3.127.0.0.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric3.147.102.13.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric4.0.0.0.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric4.127.0.0.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric4.147.102.13.0 = INTEGER: -1
RFC1213-MIB::ipRouteNextHop.0.0.0.0 = IpAddress: 0.0.0.0
RFC1213-MIB::ipRouteNextHop.127.0.0.0 = IpAddress: 0.0.0.0
RFC1213-MIB::ipRouteNextHop.147.102.13.0 = IpAddress: 0.0.0.0
RFC1213-MIB::ipRouteType.0.0.0.0 = INTEGER: direct(3)
RFC1213-MIB::ipRouteType.127.0.0.0 = INTEGER: direct(3)
RFC1213-MIB::ipRouteType.147.102.13.0 = INTEGER: direct(3)
RFC1213-MIB::ipRouteProto.0.0.0.0 = INTEGER: local(2)
RFC1213-MIB::ipRouteProto.127.0.0.0 = INTEGER: local(2)
RFC1213-MIB::ipRouteProto.147.102.13.0 = INTEGER: local(2)
RFC1213-MIB::ipRouteAge.0.0.0.0 = INTEGER: 0
RFC1213-MIB::ipRouteAge.127.0.0.0 = INTEGER: 0
RFC1213-MIB::ipRouteAge.147.102.13.0 = INTEGER: 0
RFC1213-MIB::ipRouteMask.0.0.0.0 = IpAddress: 0.0.0.0
RFC1213-MIB::ipRouteMask.127.0.0.0 = IpAddress: 255.0.0.0
RFC1213-MIB::ipRouteMask.147.102.13.0 = IpAddress: 255.255.255.0
RFC1213-MIB::ipRouteMetric5.0.0.0.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric5.127.0.0.0 = INTEGER: -1
RFC1213-MIB::ipRouteMetric5.147.102.13.0 = INTEGER: -1
RFC1213-MIB::ipRouteInfo.0.0.0.0 = OID: SNMPv2-SMI::zeroDotZero
RFC1213-MIB::ipRouteInfo.127.0.0.0 = OID: SNMPv2-SMI::zeroDotZero
RFC1213-MIB::ipRouteInfo.147.102.13.0 = OID: SNMPv2-SMI::zeroDotZero

$ snmpget -v 2c -c public lexmark.netmode.ece.ntua.gr
interfaces.ifTable.ifEntry.ifPhysAddress.2

IF-MIB::ifPhysAddress.2 = STRING: 0:4:0:76:48:5f
```

	Destination	Netmask	Gateway
1	0.0.0.0	0.0.0.0	147.102.13.40
2	127.0.0.0	255.0.0.0	147.102.13.40
3	147.102.13.0	255.255.255.0	147.102.13.40

(β). Υποθέστε ότι εκτελείτε την εντολή **“ping –s 2500 –c 1 147.102.222.210”** από το κόμβο **netmg.netmode.ece.ntua.gr**. Λαμβάνοντας υπόψη τον πίνακα δρομολόγησης του συγκεκριμένου μηχανήματος που περιγράψατε στο ερώτημα (α) και βρίσκοντας με τη βοήθεια των εντολών **snmpget/snmpwalk** ότι περαιτέρω πληροφορίες είναι απαραίτητες, εξηγείστε αναλυτικά την ακολουθία των πακέτων που ανταλλάχτηκαν λόγω της εκτέλεσης του ping ερωτήματος.

Το entry του πίνακα που θα κάνει match είναι μόνο το default 0.0.0.0
Συνεπώς η δρομολόγηση θα γίνει μέσω του default gateway με ip **147.102.13.200**.
Δεν μπορώ να κάνω snmpwalk πάνω στο default gateway. Αν ο **147.102.222.210** είναι στο ίδιο υποδίκτυο με το default gateway (πράγματι είναι αφού πρόκειται για τον κόμβο με host name achilles.noc.ntua.gr) τότε το πακέτο icmp request στέλνεται κατευθείαν. Τα αντίστοιχα icmp replies ακολοθούν την αντίστροφη πορεία.

3. Πρόσφατα ο διαχειριστής του δικτύου εισήγαγε ένα καινούργιο «μηχάνημα» στο τοπικό μας δίκτυο και του απέδωσε την IP διεύθυνση **147.102.13.254**. Με τη βοήθεια του πρωτοκόλλου SNMP προσπαθήστε να ανακαλύψετε λεπτομέρειες για τη συσκευή αυτή. Συγκεκριμένα, χρησιμοποιώντας μόνο τις πληροφορίες που μπορείτε να αντλήσετε μέσω SNMP, απαντήστε στα παρακάτω ερωτήματα:

   - Ποιο το είδος της συσκευής; (H/Y, router, switch, workstation, printer,άλλο;) Αιτιολογείστε επαρκώς την απάντησή σας.*

     SNMPv2-MIB::**sysDescr**.0 = STRING: **24-Port 10/100/1000 Gigabit Switch** w/WebView **(Cisco-Linksys, LLC)**
     SNMPv2-MIB::**sysName**.0 = STRING: **switch**

   - Αναφέρατε το πλήθος, τον τύπο, και την ταχύτητα των δικτυακών interfaces της συσκευής. Ποιο είναι το μέγιστο μέγεθος δεδομένων που μπορεί να μεταδοθεί από κάθε interface; (Δώστε επεξήγηση όπου χρειάζεται στις απαντήσεις σας)*

     Total: 32 physical + 3 virtual = 35
     IF-MIB::ifType.{1..32} = INTEGER: ethernetCsmacd(6)
     IF-MIB::ifSpeed.{1..32} = Gauge32: 1000000000

- Αναφέρατε την υπάρχουσα κατάσταση λειτουργίας των δικτυακών interfaces της συσκευής. Μπορείτε να προσδιορίσετε την επιθυμητή κατά τον διαχειριστή κατάσταση λειτουργίας των interfaces; Είναι όλα συνδεδεμένα στο δίκτυο;

  ifAdminStatus OBJECT-TYPE
  SYNTAX INTEGER {
  up(1), -- ready to pass packets
  down(2),
  testing(3) -- in some test mode
  }
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
  "The desired state of the interface. The
  testing(3) state indicates that no operational
  packets can be passed."
  ::= { ifEntry 7 }

  IF-MIB::ifAdminStatus.{1..32} = INTEGER: up(1)
  Η επιθυμητή απο τον διαχειριστή κατάσταση είναι UP για όλα τα INTERFACES.

  ifOperStatus OBJECT-TYPE
  SYNTAX INTEGER {
  up(1), -- ready to pass packets
  down(2),
  testing(3) -- in some test mode
  }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The current operational state of the interface.
  The testing(3) state indicates that no operational
  packets can be passed."
  ::= { ifEntry 8 }

  IF-MIB::ifOperStatus.1 = INTEGER: **down**(2)
  IF-MIB::ifOperStatus.2 = INTEGER: up(1)
  IF-MIB::ifOperStatus.3 = INTEGER: **down**(2)
  IF-MIB::ifOperStatus.4 = INTEGER: up(1)
  IF-MIB::ifOperStatus.5 = INTEGER: **down**(2)
  IF-MIB::ifOperStatus.6 = INTEGER: **down**(2)
  IF-MIB::ifOperStatus.7 = INTEGER: **down**(2)
  IF-MIB::ifOperStatus.8 = INTEGER: up(1)
  IF-MIB::ifOperStatus.9 = INTEGER: up(1)
  IF-MIB::ifOperStatus.10 = INTEGER: **down**(2)
  IF-MIB::ifOperStatus.11 = INTEGER: **down**(2)
  IF-MIB::ifOperStatus.12 = INTEGER: up(1)
  IF-MIB::ifOperStatus.13 = INTEGER: up(1)

IF-MIB::ifOperStatus.14 = INTEGER: up(1)
IF-MIB::ifOperStatus.15 = INTEGER: up(1)
IF-MIB::ifOperStatus.16 = INTEGER: **down**(2)
IF-MIB::ifOperStatus.17 = INTEGER: up(1)
IF-MIB::ifOperStatus.18 = INTEGER: up(1)
IF-MIB::ifOperStatus.19 = INTEGER: up(1)
IF-MIB::ifOperStatus.20 = INTEGER: up(1)
IF-MIB::ifOperStatus.21 = INTEGER: up(1)
IF-MIB::ifOperStatus.22 = INTEGER: up(1)
IF-MIB::ifOperStatus.23 = INTEGER: up(1)
IF-MIB::ifOperStatus.24 = INTEGER: up(1)
IF-MIB::ifOperStatus.25 = INTEGER: **notPresent**(6)
IF-MIB::ifOperStatus.26 = INTEGER: **notPresent**(6)
IF-MIB::ifOperStatus.27 = INTEGER: **notPresent**(6)
IF-MIB::ifOperStatus.28 = INTEGER: **notPresent**(6)
IF-MIB::ifOperStatus.29 = INTEGER: **notPresent**(6)
IF-MIB::ifOperStatus.30 = INTEGER: **notPresent**(6)
IF-MIB::ifOperStatus.31 = INTEGER: **notPresent**(6)
IF-MIB::ifOperStatus.32 = INTEGER: **notPresent**(6)
IF-MIB::ifOperStatus.100000 = INTEGER: up(1)
IF-MIB::ifOperStatus.100050 = INTEGER: up(1)
IF-MIB::ifOperStatus.100809 = INTEGER: **down**(2)

Όπως βλέπουμε τα INTERFACES με αριθμούς 1,3,5,6,7,10,11,16 είναι σε κατάσταση DOWN.
Επίσης δύο (2) απο τα τρία (3) VLAN είναι σε κατάσταση UP.

- Βρείτε το πλήθος των IP διευθύνσεων που έχουν αποδοθεί στη συσκευή. Ποια είναι η τιμή της κάθε IP διεύθυνσης; Ποια η χρησιμότητά τους για τη συγκεκριμένη συσκευή;

  Τρέχοντας την εντολή
  $ snmpwalk -v2c -c public 147.102.13.254
  ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex

  παίρνουμε
  RFC1213-MIB::ipAdEntIfIndex.147.102.13.254 = INTEGER: 100000

  το οποίο μας πληροφορεί για την ύπραξη μιας μοναδικής IP στον πίνακα που αντιστοιχεί στο vlan με index 100000 και κατ’επέκταση στο switch.

Άσκηση 2

Ζητείται η συγγραφή μιας MIB για ένα σύστημα firewall. Το συγκεκριμένο σύστημα είναι ένας υπολογιστής με περισσότερα του ενός δικτυακά interfaces. Όλα τα interfaces υποστηρίζουν το IP πρωτόκολλο και ο υπολογιστής λειτουργεί ως δρομολογητής (προωθεί πακέτα μεταξύ των interfaces).

Η λειτουργία του συστήματος ως firewall έγκειται στην εφαρμογή φίλτρων στα πακέτα που διέρχονται από τα interfaces. Ένα φίλτρο είναι ένα σύνολο από κανόνες που καθορίζουν αν ένα πακέτο επιτρέπεται να διέλθει από το interface ή εάν πρέπει να απορριφθεί. Κάθε interface μπορεί να μην εφαρμόζει κανένα φίλτρο (όλα τα πακέτα διέρχονται ελεύθερα), να εφαρμόζει ένα φίλτρο στα εισερχόμενα από το δίκτυο πακέτα, να εφαρμόζει ένα φίλτρο στα εξερχόμενα προς το δίκτυο πακέτα ή να εφαρμόζει δύο φίλτρα (ένα σε κάθε κατεύθυνση).

Κάθε φίλτρο αποτελείται από ένα σύνολο κανόνων της παρακάτω μορφής:

<RuleNo> <Action> <Protocol> <SrcIP> <SrcMask> <DstIP> <DstMask> <SrcPort> <DstPort>

όπου:
RuleNo: Αύξων αριθμός κανόνα (για το συγκεκριμένο φίλτρο)
Action: Pass ή Drop (καθορίζει αν το διερχόμενο πακέτο θα προωθηθεί ή θα απορριφθεί)
Protocol: IP, ICMP, TCP, UDP
SrcIP: Source IP address του πακέτου
SrcMask: Subnet mask που εφαρμόζεται στο Source IP address του πακέτου
DstIP: Destination IP address του πακέτου
DstMask: Subnet mask που εφαρμόζεται στο Destination IP address του πακέτου
SrcPort: Source port του πακέτου (μπορεί να είναι αριθμός Χ ή εύρος Χ-Υ)
DstPort: Destination port του πακέτου (μπορεί να είναι αριθμός Χ ή εύρος Χ-Υ)

Κάθε πακέτο που διέρχεται από το interface εξετάζεται διαδοχικά από όλους τους κανόνες του φίλτρου κατά αύξουσα σειρά RuleNo. Σε κάθε κανόνα εξετάζονται οι επικεφαλίδες του πακέτου και συγκρίνονται με τα αντίστοιχα πεδία του κανόνα (Protocol, SrcIP, κλπ). Εάν η σύγκριση είναι επιτυχής εφαρμόζεται το action του κανόνα (το πακέτο είτε προωθείται είτε απορρίπτεται οριστικά). Διαφορετικά εξετάζεται ο επόμενος κανόνας.

Η MIB που ζητείται θα πρέπει να περιλαμβάνει αντικείμενα που θα περιγράφουν τα φίλτρα, τους κανόνες τους και τις συσχετίσεις τους με τα interfaces. Φροντίστε να μην περιλαμβάνεται περιττή πληροφορία για τα interfaces, καθώς το σύστημα υποστηρίζει ήδη την MIB-II. Επιπλέον

ζητούνται και τα παρακάτω στοιχεία:

- Για κάθε κανόνα των φίλτρων να καταγράφεται πόσες φορές ενεργοποιήθηκε το action του.
- System Group που να περιέχει τις παρακάτω πληροφορίες: όνομα του συστήματος, email του διαχειριστή και χρόνο λειτουργίας του firewall.

Να παραδοθεί το σχήμα (σε μορφή δένδρου) και ο κώδικας της MIB. Η κωδικοποίηση θα πρέπει να γίνει τουλάχιστον με SNMPv2 SMI (RFCs 1901-1908). Τοποθετήστε τη MIB σε οποιοδήποτε σημείο του δένδρου αντικειμένων SNMP κάτω από το .iso αλλά δώστε συγκεκριμένες αριθμήσεις (Object IDs – OIDs) στα αντικείμενα σας.

12. netManFirewall
    12.1 ruleTable
        12.1.1 ruleEntry
            12.1.1.1 ruleNo
            12.1.1.2 ruleAction
            12.1.1.3 ruleProtocol
            12.1.1.4 ruleSrcIP
            12.1.1.5 ruleSrcMask
            12.1.1.6 ruleDstIP
            12.1.1.7 ruleDstMask
            12.1.1.8 ruleSrcPort
            12.1.1.9 ruleDstPort
            12.1.1.10 ruleCounter
            12.1.1.11 ruleIfIndex
            12.1.1.12 ruleSrcRange
            12.1.1.13 ruleDstRange
    12.2 sysGroup
        12.2.1 sysName
        12.2.2 sysContact
        12.2.3 sysUptime

FIREWALL SMI MIB-II

```
netManFirewall  OBJECT IDENTIFIER ::= { mib-2 12 }
-- the Firewall group
-- Implementation of Firewall rules

ruleTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF ruleEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
            "A list of firewall rules"
    ::= { netManFirewall 1 }

ruleEntry OBJECT-TYPE
      SYNTAX  ruleEntry
      ACCESS  not-accessible
      STATUS  mandatory
      DESCRIPTION
            "Firewall rule entry"
      INDEX   { ruleNo }
      ::= { ruleTable 1 }

ruleEntry ::=
SEQUENCE {
      ruleNo
            INTEGER,
      ruleAction
            INTEGER,
      ruleProtocol
            INTEGER,
      ruleSrcIP
            IpAddress,
      ruleSrcMask
            IpAddress,
      ruleDstIP
            IpAddress,
      ruleDstMask
            IpAddress,
      ruleSrcPort
            INTEGER (0..65535),
      ruleDstPort
            INTEGER (0..65535),
      ruleCounter
            COUNTER,
      ruleifIndex
            INTEGER,
      ruleSrcRange
            INTEGER (0..65535),
      ruleDstRange
            INTEGER (0..65535)
}


ruleNo OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Firewall rule ID”
      ::= { ruleEntry 1 }
```

```
ruleAction OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Rule Action: PASS / DROP”
      ::= { ruleEntry 2 }

ruleProtocol OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Protocol to match”
      ::= { ruleEntry 3 }

ruleSrcIP OBJECT-TYPE
      SYNTAX  IpAddress
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Firewall rule source IP Address”
      ::= { ruleEntry 4 }

ruleSrcMask OBJECT-TYPE
      SYNTAX  IpAddress
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Firewall rule source IP Address Mask”
      ::= { ruleEntry 5 }

ruleDstIP OBJECT-TYPE
      SYNTAX  IpAddress
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Firewall rule destination IP Address”
      ::= { ruleEntry 6 }

ruleDstMask OBJECT-TYPE
      SYNTAX  IpAddress
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Firewall rule destination IP Address Mask”
      ::= { ruleEntry 7 }

ruleSrcPort OBJECT-TYPE
      SYNTAX  INTEGER (0..65535)
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Firewall rule source port”
      ::= { ruleEntry 8 }
```

```
ruleDstPort OBJECT-TYPE
      SYNTAX  INTEGER (0..65535)
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Firewall rule destination port”
      ::= { ruleEntry 9 }

ruleCounter OBJECT-TYPE
      SYNTAX  COUNTER
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Firewall usage counter”
      ::= { ruleEntry 10 }

ruleIfIndex OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The index value which uniquely identifies the
            local interface to which the rule applies"
    ::= { ipRouteEntry 11 }

ruleSrcRange OBJECT-TYPE
      SYNTAX  INTEGER (0..65535)
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
            “Firewall rule source port range. This number specifies the
            number of ports after ruleSrcPort that the rule applies to.”
      ::= { ruleEntry 12 }

ruleDstRange OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            “Firewall rule destination port range. This number specifies
            the number of ports after ruleDstPort that the rule applies
            to.”
    ::= { ipRouteEntry 13 }

sysGroup OBJECT-TYPE
    SYNTAX  sysGroup
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
            "System group for the firewall"
    ::= { netManFirewall 2 }

sysName OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "An administratively-assigned name for the firewall."
    ::= { sysGroup 1 }

sysContact OBJECT-TYPE
```

```
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "The textual identification of the contact person
            for the firewall, together with information
            on how to contact this person."
    ::= { sysGroup 2 }

sysUpTime OBJECT-TYPE
    SYNTAX  TimeTicks
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The time (in hundredths of a second) since the
            firewall was last
            re-initialized."
    ::= { sysGroup 3 }
```