

3 Wireless Infrastructure

Ref: <http://www.asecuritysite.com/wireless/wireless03>

3.1 Introduction

This unit provides a foundation in some of the key issues related to wireless networks, especially related to the infrastructure of the network. The basic elements of any type of network infrastructure are:

- **Throughput.** This typically involves the creation of a hierarchical structure, where the
- **Robustness.** This is a key factor in any type of network infrastructure, especially when connecting into the core of the network.
- **Fallback.** This is another important factor, especially with business critical parts of the infrastructure, where alternative routes can be provided, or backup for key network devices.
- **Scalability.** Few networks ever stay the same, and the demand for services and bandwidth increases each year, this a good network infrastructure typically supports scalability, where the network can grow without affecting the current provision.
- **Ease of setup.** This typically makes the connection of devices simple.
- **Ease of connection.** This typically makes its easy for devices to connection, especially in connecting to the network without complex network settings.
- **Support for heterogeneous systems.** This typically allows for different types of applications and system to interconnect over the network. A good example of this is to support an IP infrastructure, along with other address systems such as NetBIOS and AppleTalk.

3.2 Three-layer model

Figure 3.1 outlines the three-layer model. If possible, the model should mirror the requirements of the network at different levels, such as connectivity (gaining access to the network), creating workgroups, security, policy and distribution. With this model, routers are used at each layer to limit the broadcasts to within that layer. The layers can be defined as:

- **Core.** Provides optimal transport between sites, which provides fast wide-area connections between geographically remote sites within an organization. Normally these are point-to-point links between routers. Typically connections are T1/T3, ATM, Frame Relay and SMDS), and are often provided by telecommunications provider. The core layer provides **low latency** connections between remote sites, and does not generally implement any filtering of the traffic (such as with firewalls and ACLs). If possible, there should be **redundant paths** which can be

switched-in when a route becomes unavailable, or slows down. The redundant paths can also be used to share **traffic loads**. There should also be **rapid convergence** of the network.

- **Distribution.** Provides policy-based connectivity, which connects multiple LANs into a larger network infrastructure, such as an organizational backbone. It will typically connect between buildings. Typically, connections to the LANs are with Fast Ethernet, or even Gigabit Ethernet, and to the core layer with ATM, FDDI and SMDS. This layer also provides the demarcation point between the access and core layers and thus helps to define the operation of the core, and isolate it from the access layer. At this layer packets can be filtered using a policy-based system (typically using a firewall). At this level campus-wide networks would be implemented, with the possibility of campus-wide servers. It is unlikely that nodes would connect directly onto the distribution layer. In a non-campus-based network, this would be the layer at which remote sites would connect to each other. Typical functions include: concentration of LANs, access to core layer, VLAN routing, and media translations (such as between Frame Relay and Ethernet) and security.
- **Access.** Provides workgroup and user access to the network, which creates LANs, and workgroups. At this level, most of the hosts connect to a network. There can be some policy-based filtering of network traffic at this layer, which will refine the access control implemented at the distribution level. At this layer, the filter will typically be based on user access (such as whether certain individuals are allowed access to certain services). The main functions at this layer are: shared bandwidth (using hubs), switched bandwidth (using switches); MAC-layer filtering (routing based on MAC address, such as using in a switch or a bridge), isolating broadcast traffic, creating workgroups, and microsegmentation.

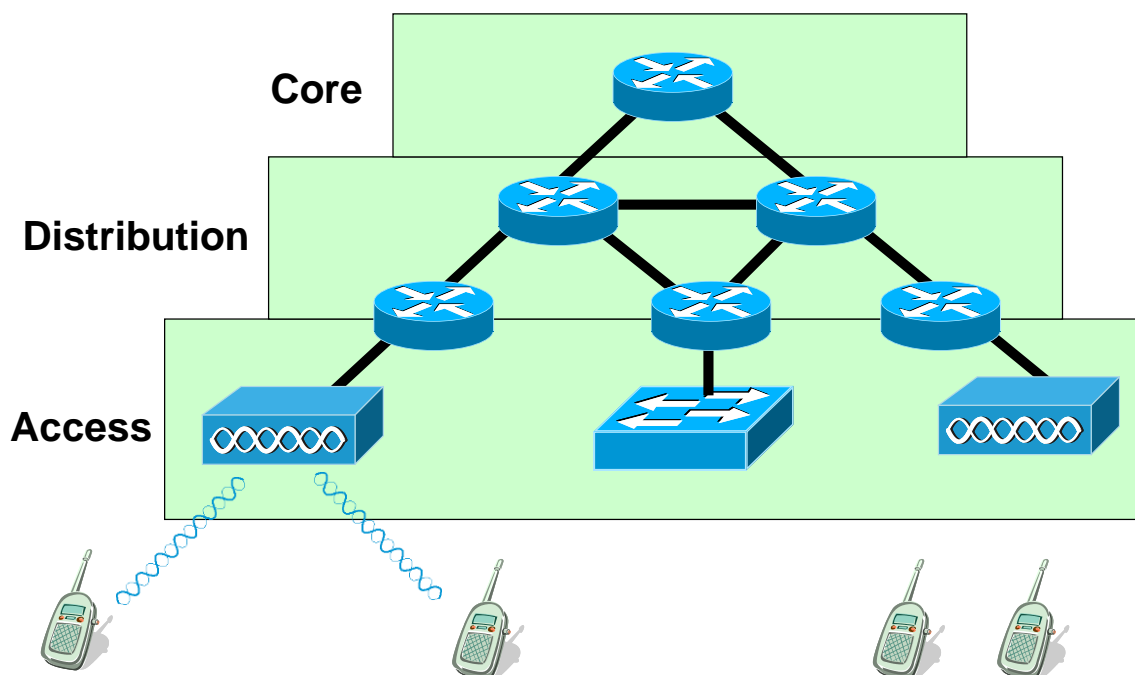


Figure 3.1 Three-layered component model

At present wireless devices are used mostly for access to the network infrastructure, but in the future, once key issues on network throughput, robustness, security and authentication have been solved, they may move more towards the core of the network infrastructure.

3.3 Repeaters or root devices

A root access point is used to connect a wireless client to a fix network, whereas a repeater access point does not connect to a wired LAN, and basically forwards the data packets to another repeater or to a wireless access point which is connected to a wired network (Figure 3.2). With a repeater, of course, the Ethernet port will not operate. The repeater access point typically associates with an access point which has the best connectivity, however they can be setup to connect to a specific access point. In the following case, the access point will associate with the parent with the specified MAC address (1111.2222.3333):

```
# config t
(config)# interface d0
(config-if)# ssid napier
(config-ssid)# infrastructure-ssid
(config-ssid)# exit
(config-if)# station-role repeater
(config-if)# dot11 extensions aironet
(config-if)# parent 1 1111.2222.3333
(config-if)# parent 2 2222.aaaa.bbbb
```

It is possible to define up to four parents, so that if one fails to association, it can use others. In most cases the Cisco Aironet extensions must be enabled, as it aids the association process, but this can cause incompatibility problems with non-Cisco devices.

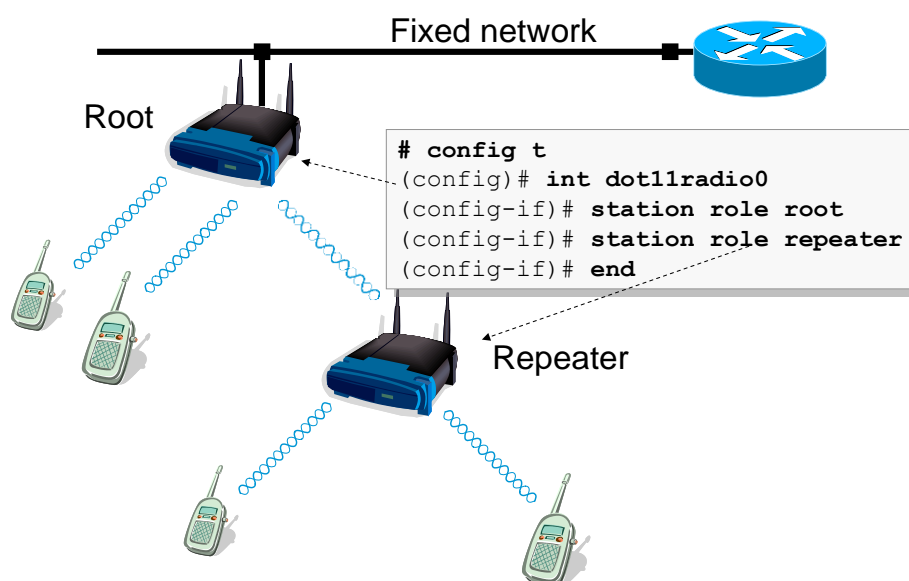


Figure 3.2 Repeater or root

The repeater will start with the first parent, and, if it cannot connect, it will then try the next parent, and so on. Overall, repeaters are fairly good at extending the range of a wireless network, but they reduce the throughput, as bandwidth is wasted in relaying the data from repeaters. As an approximation the actual throughput will be reduced by at least half.

3.4 Device fallback

The hot standby function is used to provide a backup to another access point, and is configured in the same way, so that if it fails, the hot standby device can become active, and associates the active clients, automatically. The only setting that will differ is the IP address of the device. In the following configuration, the MAC address of the device to be monitored is **1111.abcd.ef10**. The timeout period in which the device will determine if the monitored device has stopped working is five seconds, and the poll time is two seconds:

```
# config t
(config)# iapp standby mac 1111.abcd.ef10
(config)# iapp standby timeout 5
(config)# iapp standby polltime 2
```

The hot standby device has a different IP address (as it may cause a conflict when the two devices are operating at the same time, but, for the sake of seamless operation, the hot standby device must be setup with the following settings by identical:

- SSID.
- IP Subnet Mask.
- Default gateway.
- Data rates.
- Encryption and authentication settings.

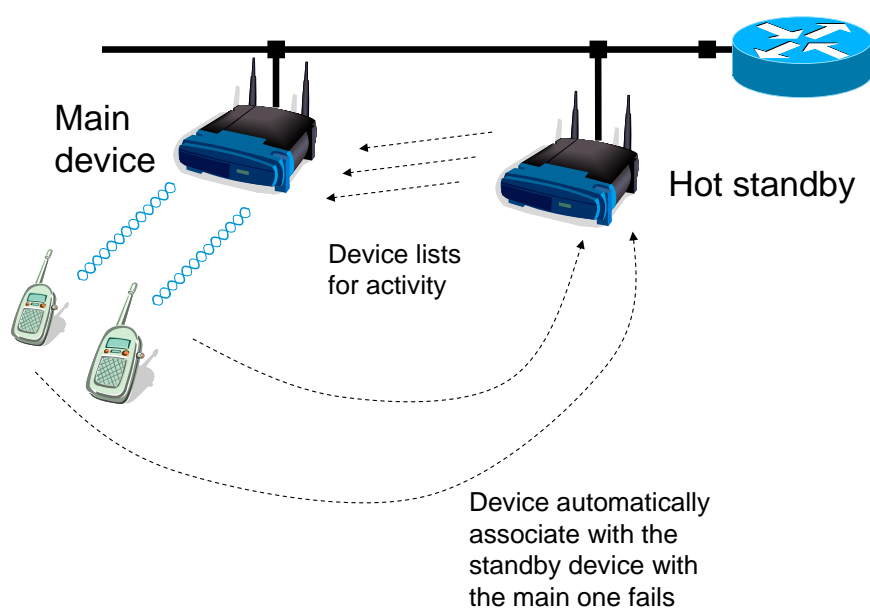


Figure 3.3 Repeater or root

3.5 Bridging

In the same way that an Ethernet bridge works, a wireless bridge can be used to interconnect two or more networks. They are typically used in hard-to-wire places, or where cable runs would spoil the look of the environment. The basic modes include:

- **Point-to-point** (master/slave). This is used to connect two LANs using two bridging units, and thus provide an extended broadcast domain.
- **Point-to-multipoint**. This allows the connection of multiple LANs using a wireless bridge.

A good example of a wireless bridge is the Cisco Aironet 350 workgroup bridge (WGB) which connects an Ethernet network to a wireless access point. Figure 3.4 shows an example of a remote workgroup which connect to a fixed network using a wireless bridge. The bridge has the advantage of the repeater is that the bridge can learn the structure of the network, and the devices which connect, and can thus learn which data frames to send across the bridge, and which not to forward. A repeater blindly forwards data frames without checking their destination. It can be seen in Figure 3.4 that a broadcast is sent over the bridge and onto every device within the broadcast domain. This domain is bounded by routers, which do not forward broadcasts. Figure 3.5 shows an example of a point-to-multipoint bridge, where three bridges are used to bridge three LANs.

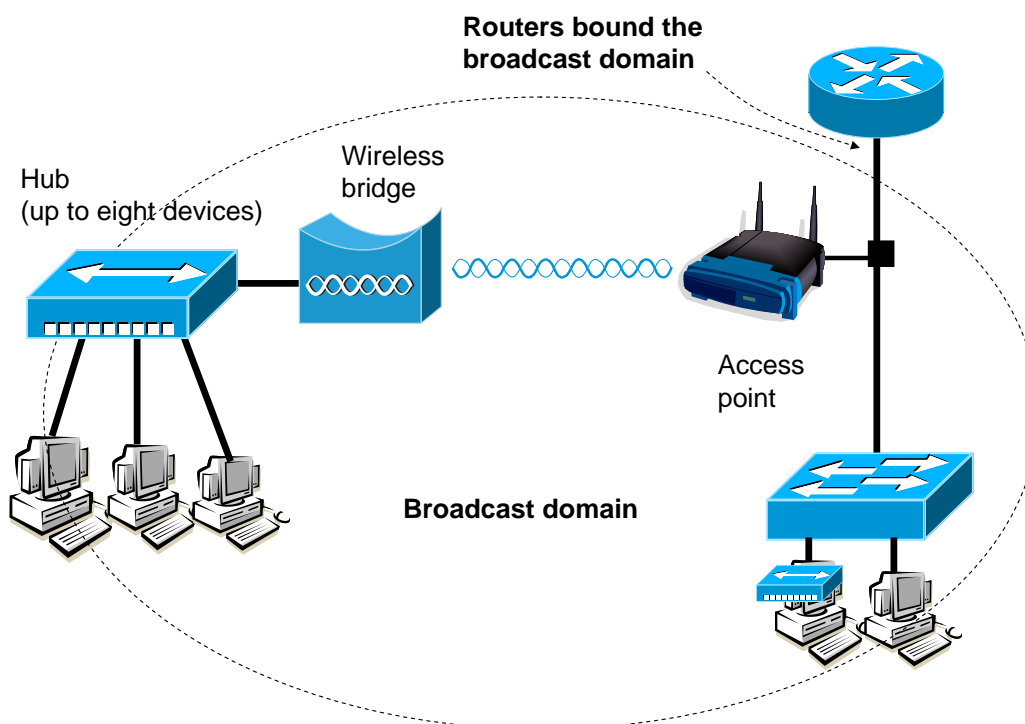


Figure 3.4 Point-to-point Wireless bridge

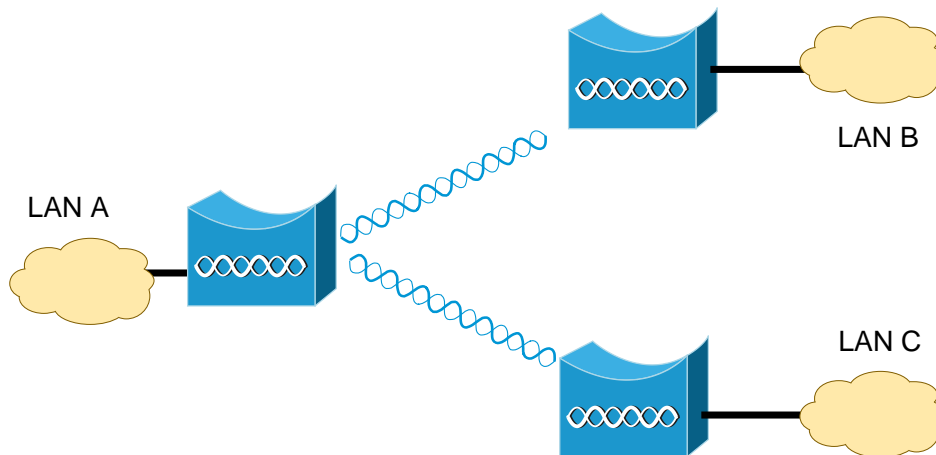


Figure 3.5 Point-to-multipoint Wireless bridge

3.6 Remote access

A wireless access point is typically accessible through the TELNET and/or HTTP proposal. The HTTP service is important as it allows remote access through a Web browser, and can be authenticated locally with:

```
# config t
(config) # username fred password bert
(config) # ip http server
(config) # ip http authentication local
(config) # exit
```

This type of authentication is not the most secure but it offers a simple way to block access to the access point. Thus, when the user tries to access to the wireless access point they will not be allowed to connect, unless the have the correct username and password, such as shown in Figure 3.6. If the user has the correct username and password, the Web page will show the device settings (left-hand side of Figure 3.7), otherwise there will be an authentication failure (right-hand side of Figure 3.7).

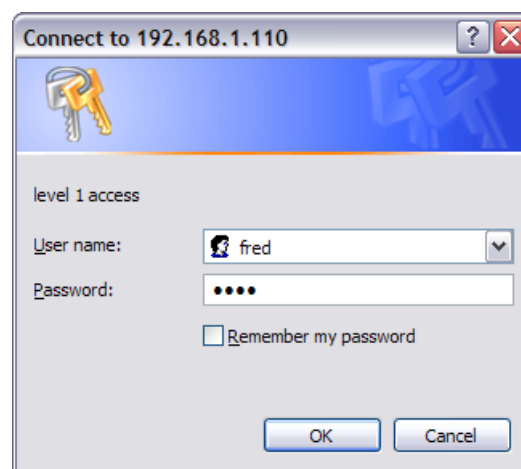


Figure 3.6 Local authentication

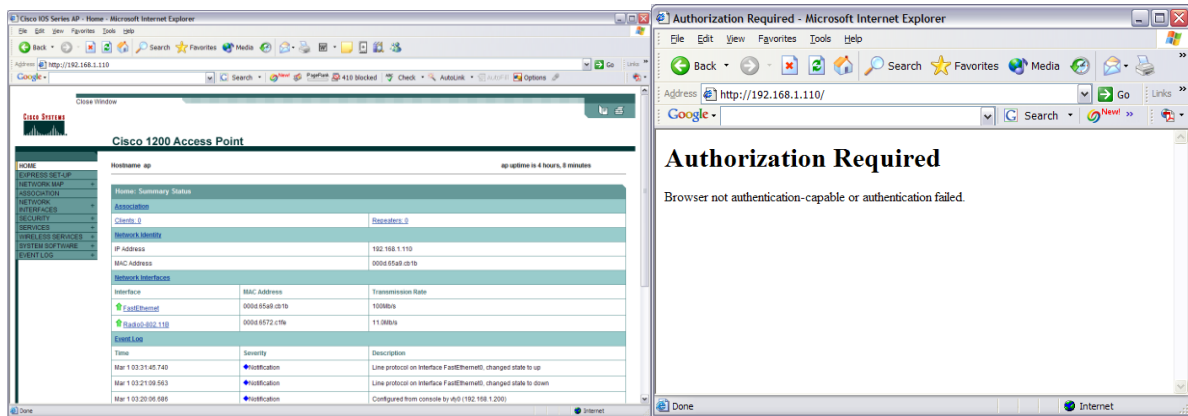


Figure 3.7 Web access success and failure

Often a new HTTP port is required (to stop users from trying to access the Web page). Thus to change the port:

```
# config t
(config) # ip http port 8080
```

Now we cannot access the Web page with the standard port (80), and we must change the address with a colon to define the port, such as shown in Figure 3.8.

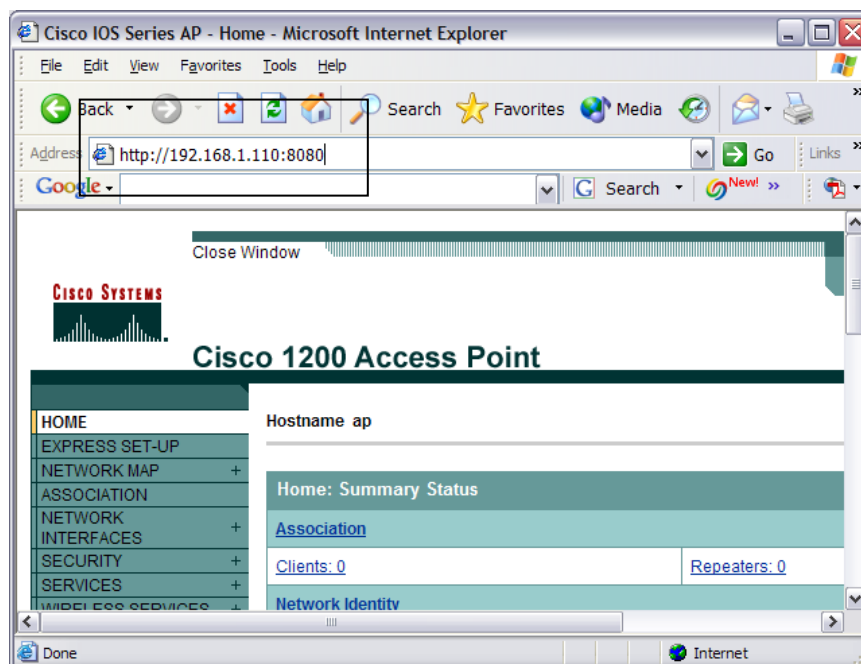


Figure 3.8 Change of HTTP port

TELNET access is another important method, and the methods used to access can be seen from Tutorial 2.

3.7 CDP

CDP (Cisco Discovery Protocol) is used to discover Cisco devices which connect to a given port. It is set globally on the device with **cdp run**, and then the timers are set as:

```
# config t
(config)# cdp holdtime 120
(config)# cdp timer 50
(config)# end
```

To enable CDP on the wireless access point:

```
# config t
(config)# cdp run
(config)# end
```

To enable CDP on an interface:

```
# config t
(config)# int fa0
(config-if)# cdp enable
(config-if)# end
```

To show CDP information:

```
# show cdp neighbors
# show cdp neighbors detail
# show cdp neighbors traffic
```

3.8 SNMP

SNMP (Simple Network Management Protocol) is a well-supported standard which can be used to monitor and control devices. It typically runs on hubs, switches and bridges. Many SNMP devices provide both general network management and device management through a serial cable, modem, or over the network from a remote computer. It involves a primary management station communicating with different management processes. Figure 3.9 shows an outline of an SNMP-based system. An SNMP agent runs SNMP management software. An SNMP server sends commands to the agent which responds back with the results. In this figure the server asks the agent for its routing information and the agent responds with its routing table. These responses can either be polled (the server sends a request for information) or interrupt-driven (where the agent sends its information at given events). A polled system tends to increase network traffic as the agent may not have any updated information (and the server must re-poll for the information).

The SNMP (Simple Network Management Protocol) protocol is initially based in the RFC1157 document. It defines a simple protocol which gives network element management information base (MIB). There are two types of MIB: MIB-1 and MIB-2. MIB-1 was defined in 1988 and has 114 table entries, divided into two groups. MIB-2

is a 1990 enhancement which has 171 entries organized into 10 groups (RFC 1213). Most devices are MIB-1 compliant and newer one with both MIB-1 and MIB-2.

The database contains entries with four fields:

- Object type. Defines the name of the entry.
- Syntax. Gives the actual value (as string or an integer).
- Access field. Defines whether the value is read-only, read/write, write-only and not accessible.
- Status field. Contains an indication on whether the entry in the MIB is mandatory (the managed device must implement the entry), optional (the managed device may implement the entry) or obsolete (the entry is not used).

SNMP is a very simple protocol but suffers from the fact that it is based on connectionless, unreliable, UDP. The IAB have recommended that the Common Management Information Services (CMIS) and Common Management Information Protocol (CMIP) be accepted as standard for future TCP/IP systems. The two main version of SNMP are SNMP Ver1 and SNMP Ver2. SNMP has added security to stop intruders determining network loading or the state of the network.

The SNMP architecture is based on a collection of:

- Network management stations. These execute management applications which monitor and control network elements.
- Network elements. These are devices such as hosts, gateways, terminal servers, and so on and have management agents which perform network management functions replying to requests from network management stations.

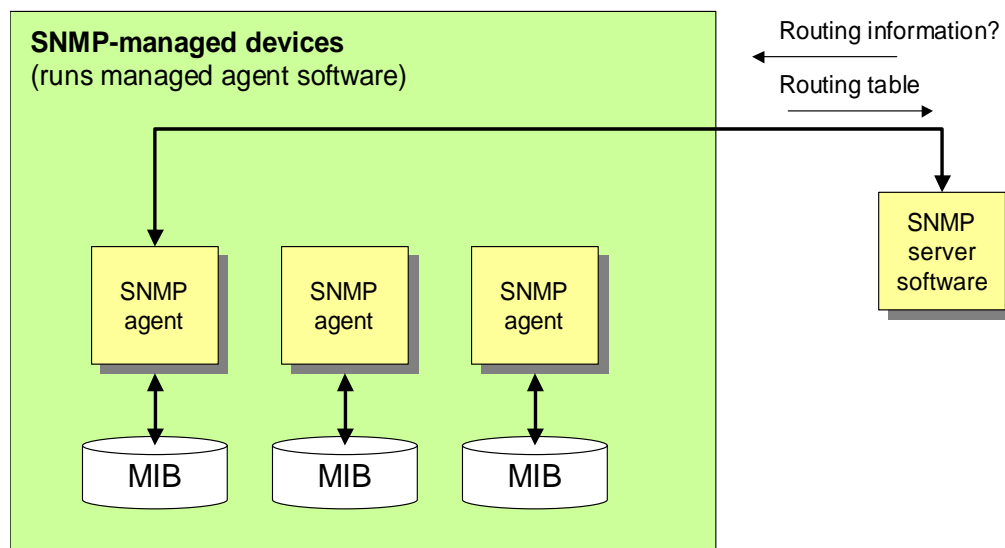


Figure 3.9 SNMP architecture

3.9 Protocol specification

The network management protocol operates by inspecting or altering variables on an agent's MIB (management information base). They communicate by exchanging

messages within UDP datagrams. These messages are defined using ASN.1 and are specified in listing 1. They consist of:

- A Version identifier (*version*). An integer value defining the version number.
- SNMP community name (*community*). An eight character string defining the community name.
- A protocol data unit (*data*). All SNMP implementations five PDUs: GetRequest-PDU, GetNextRequest-PDU, GetResponse-PDU, SetRequest-PDU, and Trap-PDU.

The protocol receives messages from:

- UDP port 161. For all messages apart from report traps (Trap-PDU).
- UDP port 162. Report trap Messages

MIB-2 added a number of groups, including system, interfaces, at, ip, icmp, tcp, udp, egp, and snmp (see Figure 2).

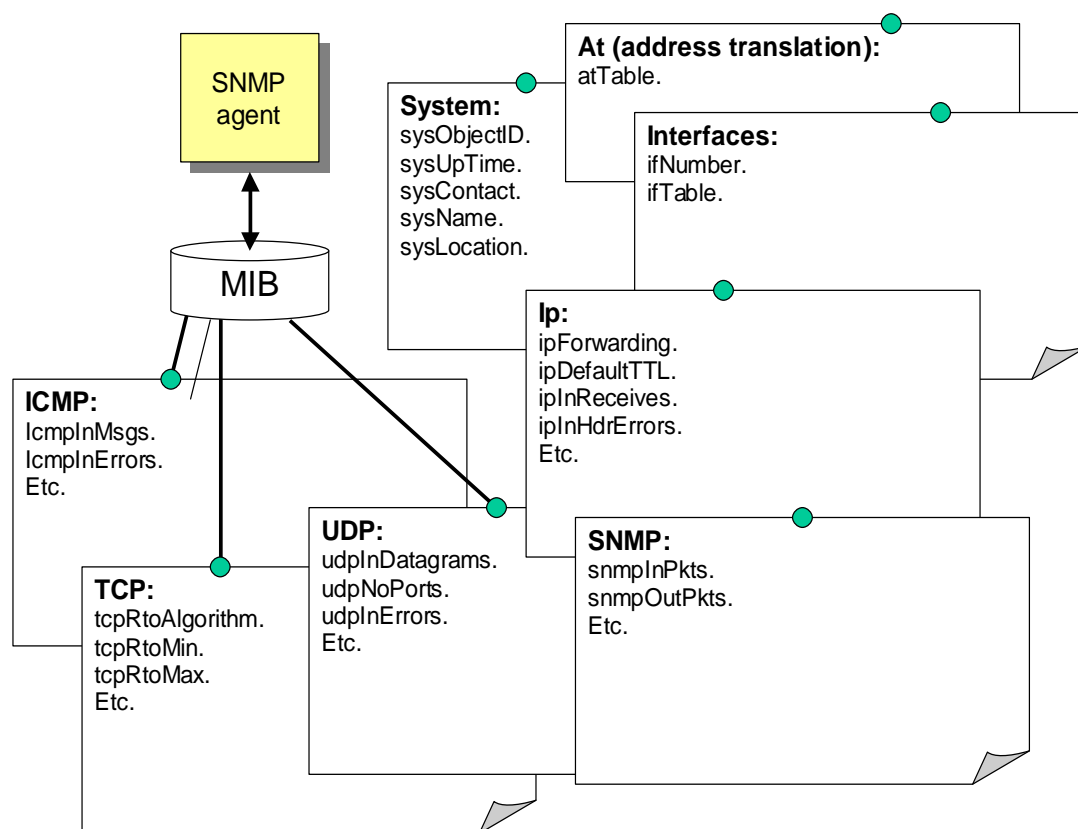


Figure 3.10 MIB-2 tables

Listing 1

```

RFC1157-SNMP DEFINITIONS ::= BEGIN
IMPORTS
    ObjectName, ObjectSyntax, NetworkAddress, IpAddress, TimeTicks
        FROM RFC1155-SMI;

-- top-level message
Message ::=

```

```

SEQUENCE {
    version      -- version-1 for this RFC
        INTEGER {
            version-1(0)
        },
    community    -- community name
        OCTET STRING,
    data         -- e.g., PDUs if trivial
        ANY      -- authentication is being used
}

-- protocol data units
PDUs ::=
    CHOICE {
        get-request          GetRequest-PDU,
        get-next-request     GetNextRequest-PDU,
        get-response         GetResponse-PDU,
        set-request           SetRequest-PDU,
        trap                  Trap-PDU
    }

-- the individual PDUs and commonly used
-- data types will be defined later
END

```

3.10 SNMP on a wireless access point

The SNMP (Simple Network Management Protocol) is a powerful method of gaining information on the operation of the network. The **snmp-server** command is used to enable SNMP monitoring. The **snmp-server community** command is used to initialise SNMP, and set the community string (which is basically used as a type of password for the SNMP access). For example to define the read-only string to public:

```

# config t
(config)# snmp-server community public RO

```

The RO defines read-only access, while RW defines read-write access. To setup the SNMP contact, the location:

```

(config)# snmp-server contact fred smith
(config)# snmp-server location room c6

```

SNMP contains a database of monitored network conditions, such as the number of errors in data packets, the IP addresses of the interfaces, and so on. It can also be set-up to trigger on certain traps, such as on syslog traps. To enable all of SNMP traps so that all the data is monitored:

```

(config)# snmp-server enable traps

```

Then to send these traps to a remote host (to www.myhost.com):

```

# config t
(config)# snmp-server host www.myhost.com public

```

To determine the status of the SNMP communications:

```
# show snmp
```

and to display the SNMP engine and remote engines:

```
# show snmp engine
```

and to display the SNMP group:

```
# show snmp group
```

SNMP uses an MIB database to store its values. To display its contents:

```
# show snmp mib
```

3.11 SNMP tree structure

The MIB tree structure is defined by a long sequence of numbers separated by dots, such as .1.3.6.1.2.1.1.4.0 (where the .0 represents an end node). This number is called an **Object Identifier (OID)**. The OID is a numerical representation of the MIB tree structure. Each digit represents a node in this tree structure. The trunk of the tree is on the left; the leaves are on the right, as illustrated in Figure 3.13 and Figure 3.14.

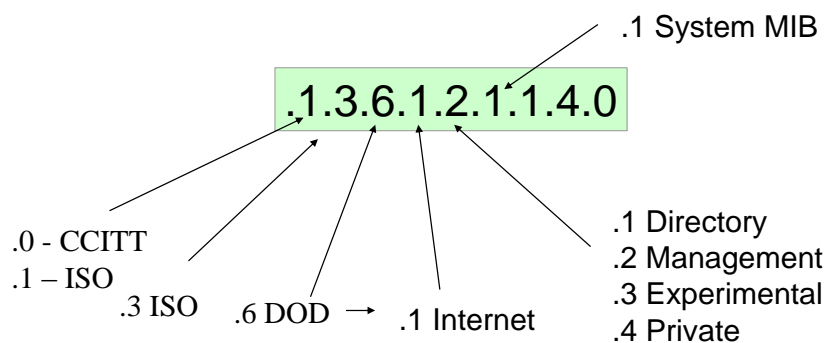


Figure 3.11 SNMP object ID

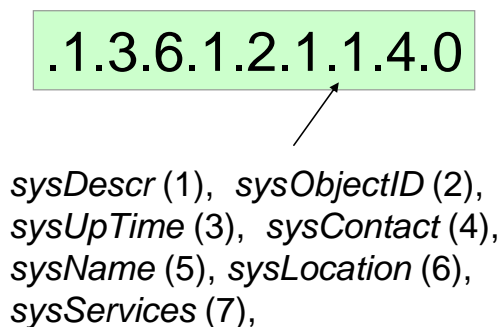


Figure 3.12 SNMP object ID

For example a node with an ID of 1.3.6.1.2.1.5.1.0 has the following structure:

- iso(1).
- org(3).

- dod(6).
- internet(1).
- mgmt(2).
- mib-2(1).
- icmp(5).
- icmpInMsgs(1).

For a router, example objects are:

MIB name	Description	Object ID
sysName	Hostname	.1.3.6.1.2.1.1.5.0
sysUpTime	Uptime	.1.3.6.1.2.1.1.3.0
sysDescr	System Description	.1.3.6.1.2.1.1.1.0
sysContact	System Contact	.1.3.6.1.2.1.1.4.0
sysLocation	System Location	.1.3.6.1.2.1.1.6.0
ciscoImageString	IOS Version	.1.3.6.1.4.1.9.9.25.1.1.1.2.5
avgBusy1	1-Minute CPU Util.	.1.3.6.1.4.1.9.2.1.57.0
avgBusy5	5-Minute CPU Util.	.1.3.6.1.4.1.9.2.1.58.0
freeMem	Free memory	.1.3.6.1.4.1.9.2.1.8.0
ciscoImageString.4	IOS feature set	.1.3.6.1.4.1.9.9.25.1.1.1.2.4

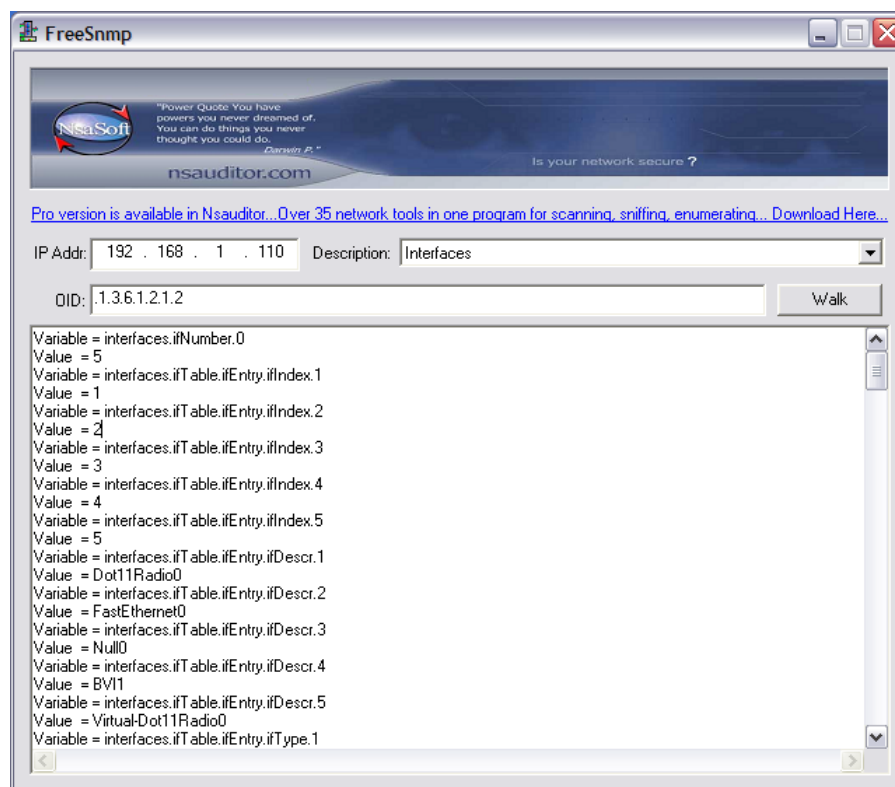


Figure 3.13 SNMP object ID

For example the following are example SNMP values and variables:

Variable = system.sysDescr.0

Value = Cisco Internetwork Operating System Software
IOS (tm) C1200 Software (C1200-K9W7-M), Version 12.2(11)JA, EARLY DEPLOY-
MENT RELEASE SOFTWARE (fc2)
TAC Support: <http://www.cisco.com/tac>
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Fri 23-May-

Variable = interfaces.ifNumber.0, Value = 5
Variable = interfaces.ifTable.ifEntry.ifIndex.1, Value = 1
Variable = interfaces.ifTable.ifEntry.ifIndex.2, Value = 2
Variable = interfaces.ifTable.ifEntry.ifIndex.3, Value = 3
Variable = interfaces.ifTable.ifEntry.ifIndex.4, Value = 4
Variable = interfaces.ifTable.ifEntry.ifIndex.5, Value = 5
Variable = interfaces.ifTable.ifEntry.ifDescr.1, Value = Dot11Radio0
Variable = interfaces.ifTable.ifEntry.ifDescr.2, Value = FastEthernet0
Variable = interfaces.ifTable.ifEntry.ifDescr.3, Value = Null0
Variable = interfaces.ifTable.ifEntry.ifDescr.4, Value = BVI1
Variable = interfaces.ifTable.ifEntry.ifDescr.5, Value = Virtual-Dot11Radio0

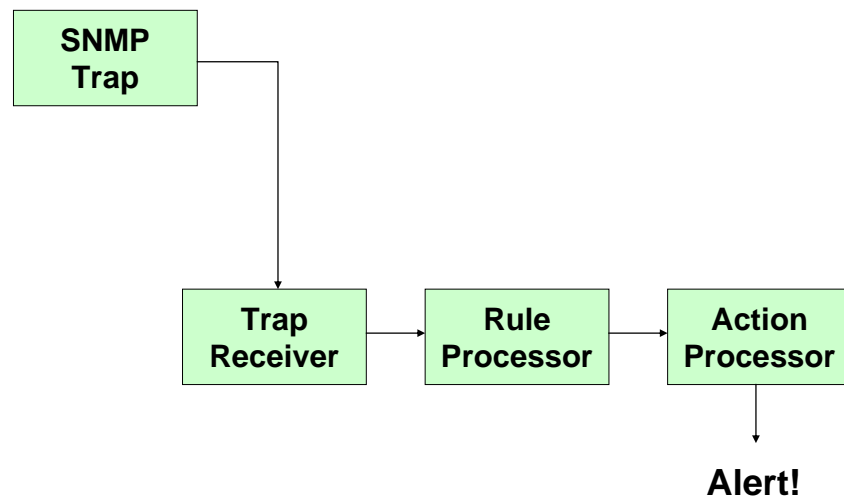


Figure 3.14 SNMP traps

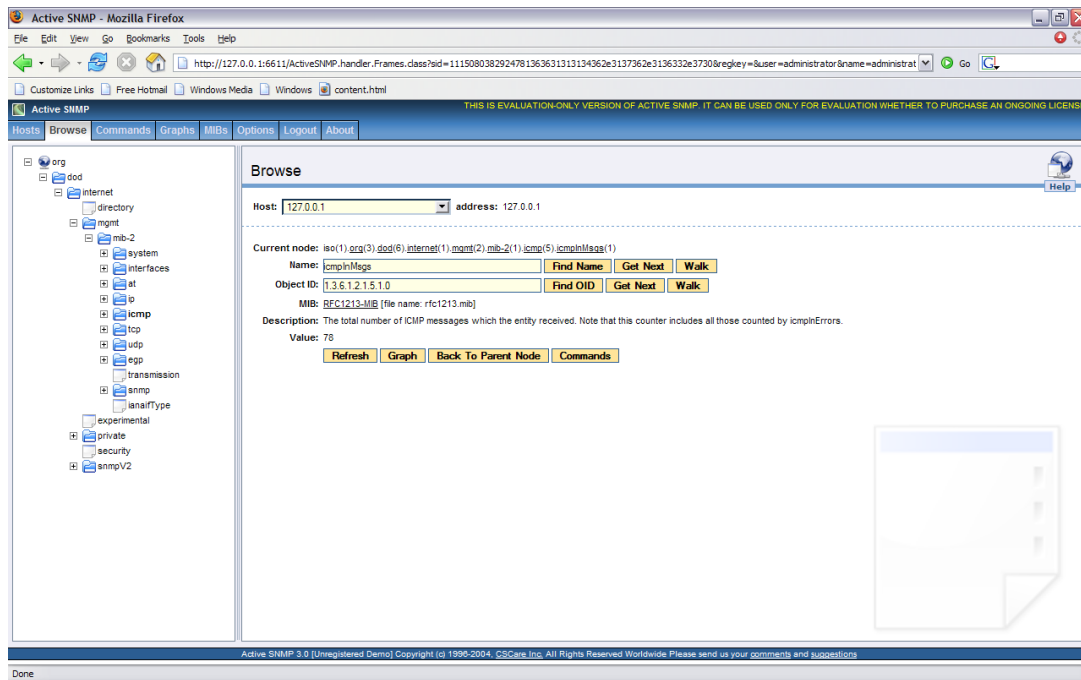


Figure 3.15 Example SNMP structure

3.12 Appendix

3.12.1 MIB-2: system

These include:

- sysObjectID. Identifies object ID.
- sysUpTime. Identifies system up time.
- sysContact. Identifies the system contact.
- sysName. Identifies the system name.
- sysLocation. Identifies the location of the system.
- sysServices. Identifies the system services.

3.12.2 MIB-2: interfaces

The interfaces table includes:

- ifNumber. Number of interfaces.
- ifTable. List of interface entities:
 - ifIndex. Interface index value.
 - ifDescr. Interface description.
 - ifType. Interface type: other(1), regular1822(2), hdlc(3), ddn-x25(4), rfc877-x25(5), ethernet-csmacd(6), iso88023-csmacd(7), iso88024-tokenBus(8), iso88025-tokenRing(9), iso88026-man(10), starLan(11), proteon-10Mbit(12), proteon-80Mbit(13), hyperchannel(14), fddi(15), lapb(16), sdlc(17), ds1(18), e1(19), basicISDN(20), primaryISDN(21), ppp(23), softwareLoopback(24), eon(25), ethernet-3Mbit(26)
 - ifSpeed. Speed of interface, in bits per second.
 - ifPhysAddress.

- ifAdminStatus. Administration status is Up (1), down (2) or testing (3).
- ifOperStatus. Operational status is Up (1), down (2) or testing (3).
- ifLastChange. Time since last change.
- ifInUcastPkts.
- ifInNUcastPkts.
- ifInDiscards.
- ifInErrors.
- ifInUnknownProtos.
- ifOutOctets.
- ifOutUcastPkts.
- ifOutNUcastPkts.
- ifOutDiscards.
- ifOutErrors.
- ifOutQLen.
- ifSpecific.

3.12.3 MIB-2: at

The address translations table includes:

- atTable. This defines the addresses translations table, and each interface contains one network address to physical address translation:
 - atIfIndex. Interface interface.
 - atPhysAddress. Physical address of the interface.
 - atNetAddress. Network address of the interface.

3.12.4 MIB-2: ip

This ip table include information on IP traffic, such as:

- ipForwarding. Defines whether the node is a gateway or not. It can be set to: forwarding (for a gateway) or not-forwarding.
- ipDefaultTTL. IP Time-to-live.
- ipInReceives. The total number of IP packets (including ones in error).
- ipInHdrErrors. Discarded IP packets, due to header problems.
- ipInAddrErrors . Discarded IP packets, due to incorrect addresses (such as 0.0.0.0).
- ipForwDatagrams. Number of IP packets which were forwarded.
- ipInUnknownProtos. Number of IP packets with an unknown protocol.
- ipInDiscards. Discarded packets due to processing problems, such as lack of buffer memory.
- ipInDelivers. Number of successfully IP packets.
- ipOutRequests.
- ipOutDiscards.
- ipOutNoRoutes. Discarded IP packets, due to no router for the packets.
- ipFragOKs. Number of completed fragments.
- ipFragFails. Number of unsuccessful fragments.
- ipFragCreates. Number of fragments created.
- ipAddrTable.

- ipAddrEntry:
 - ipAdEntAddr. Network address.
 - ipAdEntIfIndex. Address index.
 - ipAdEntNetMask. Subnet mask.
 - ipAdEntBcastAddr. Broadcast address.
 - ipAdEntReasmMaxSize.
- ipRoutingTable:
 - ipRouteDest. Destination address. A value of 0.0.0.0 is defined as a default route.
 - ipRouteIfIndex. Route index.
 - ipRouteMetric1. Route metric 1. If it is not using the value is set to -1.
 - ipRouteMetric2.
 - ipRouteMetric3.
 - ipRouteMetric4.
 - ipRouteNextHop.
 - ipRouteType. Route types are: other, invalid, direct and indirect.
 - ipRouteProto. Protocol types are: other, local, netmgmt, icmp, egp, ggp, hello, rip, is-is, es-is, ciscoIGRP, bbnSpfIgp, ospf and bgp.
 - ipRouteAge.
 - ipRouteMask.
 - ipRouteMetric5.
- ipRouteInfo:
 - ipNetToMediaIfIndex. Route index.
 - ipNetToMediaPhysAddress. Physical address.
 - ipNetToMediaNetAddress. Network address.
 - ipNetToMediaType. Set to other, invalid, dynamic or static.

3.12.5 MIB-2: icmp

The ICMP table includes:

- icmpInMsgs.
- icmpInErrors.
- icmpInDestUnreachs.
- icmpInTimeExcds
- icmpInParmProbs
- icmpInSrcQuenches.
- icmpInRedirects.
- icmpInEchos.
- icmpInEchoReps.
- icmpInTimestamps.
- icmpInTimestampReps.
- icmpInAddrMasks.
- icmpInAddrMaskReps.
- icmpOutMsgs.
- icmpOutErrors.
- icmpOutDestUnreachs.
- icmpOutTimeExcds.

- icmpOutParmProbs.
- icmpOutSrcQuenches.
- icmpOutEchos.
- icmpOutEchoReps.
- icmpOutTimestamps.
- icmpOutTimestampReps.
- icmpOutAddrMasks.
- icmpOutAddrMaskReps.

3.12.6 MIB-2: Tcp

The TCP table includes:

- tcpRtoAlgorithm. This is used to determine the time-out for unacknowledged segments. This can be: other, constant, rsre or vanj (Van Jacobson's)
- tcpRtoMin. Minimum retransmission time-out (in milliseconds).
- tcpRtoMax. Maximum retransmission time-out (in milliseconds).
- tcpMaxConn. Maximum number of TCP connections.
- tcpActiveOpens. Number of active TCP connections.
- tcpPassiveOpens. Number of passive TCP connection.
- tcpAttemptFails.
- tcpEstabResets.
- tcpCurrEstab.
- tcpInSegs. Number of input segments.
- tcpOutSegs. Number of output segments.
- tcpRetransSegs. Number of retransmitted segments.
- tcpConnTable:
 - tcpConnState. The state can be: closed, listen, synSent, synReceived, established, finWait1, finWait2, closeWait, lastAck, closing, timeWait or deleteTCB.
 - tcpConnLocalAddress. Local address.
 - tcpConnLocalPort. Local port.
 - tcpConnRemAddress. Remote address.
 - tcpConnRemPort. Remote port.

3.12.7 MIB-2: Udp

The Udp table includes:

- udpInDatagrams.
- udpNoPorts.
- udpInErrors
- udpOutDatagrams
- udpTable:
 - udpLocalAddress
 - udpLocalPort

3.12.8 snmp

The snmp table includes information on SNMP:

- snmpInPkts.
- snmpOutPkts.
- snmpInBadVersions.
- snmpInBadCommunityNames.
- snmpInBadCommunityUses.
- snmpInASNParseErrs.
- snmpInTooBigs.
- snmpInNoSuchNames.
- snmpInBadValues.
- snmpInReadOnlys.
- snmpInGenErrs.
- snmpInTotalReqVars.
- snmpInTotalSetVars.
- snmpInGetRequests.
- snmpInGetNexts.
- snmpInSetRequests.
- snmpInGetResponses.
- snmpInTraps.
- snmpOutTooBigs.
- snmpOutNoSuchNames.
- snmpOutBadValues.
- snmpOutGenErrs.
- snmpOutGetRequests.
- snmpOutGetNexts.
- snmpOutSetRequests.
- snmpOutGetResponses.
- snmpOutTraps.
- snmpEnableAuthenTraps.