Chapter 1

Data Communications and NM Overview

Outline

- Analogy of telephone network
- Data and telecommunication network
- Distributed computing environment
- Internet
- Protocols and standards
- IT management
- Network and system management
- Current status and future of network management

Notes

Telephone Network

- Characteristics:
  - Reliable - does what is expected of it
  - Dependable - always there when you need it (remember 911?)
  - Good quality (connection) - hearing each other well
- Reasons:
  - Good planning, design, and implementation
  - Good operation and management of network

Notes

Telephone Network Model

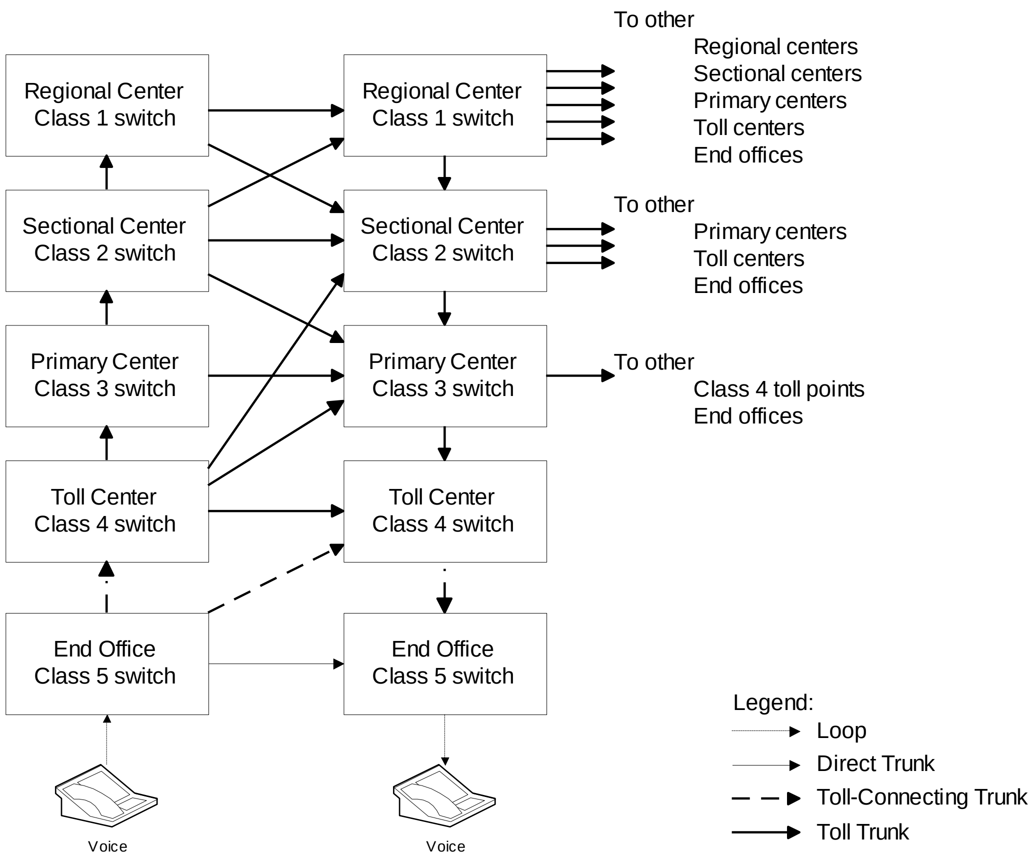


Figure 1.1 Telephone Network Model

Notes

- Notice the hierarchy of switches
- Primary and secondary routes programmed
- Automatic routing
- Where is the most likely failure?
- Use of Operations Systems to ensure QoS

Operations Systems / NOC

- Monitor telephone network parameters
    - S/N ratio, transmission loss, call blockage, etc.
- Real-time management of network
- Trunk (logical entity between switches) maintenance system measures loss and S/N. Trunks not meeting QoS are removed before customer notices poor quality
- Traffic measurement systems measure call blockage. Additional switch planned to keep the call blockage below acceptable level
- Operations systems are distributed at central offices
- Network management done centrally from Network Operations Center (NOC)

Notes

Data and Telecommunication Network

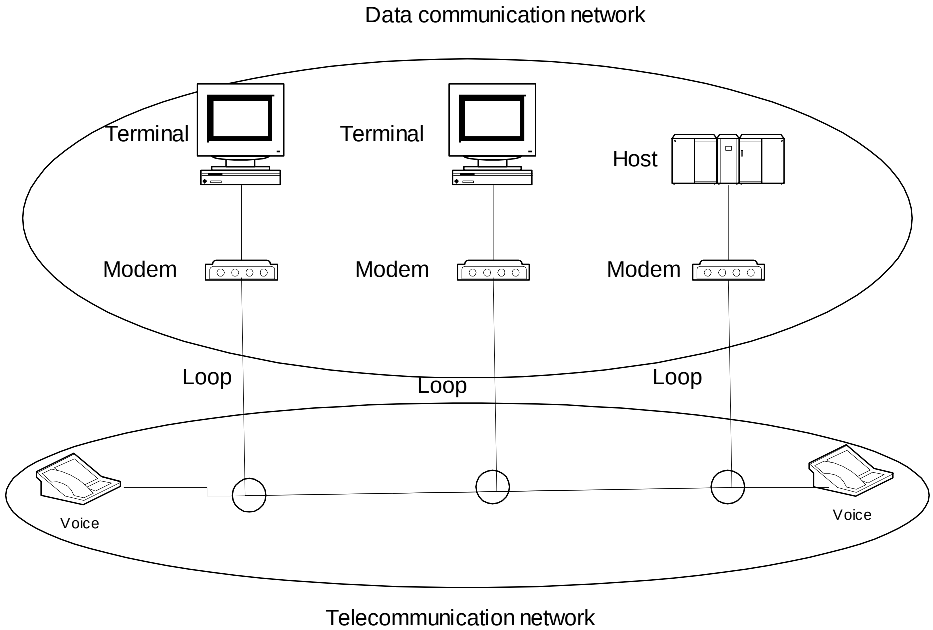


Figure 1.3 Data and Telecommunication Networks

Notes

- Computer data is carried over long distance by telephone (telecommunication network)
- Output of telephone is analog and output of computers is digital
- Modem is used to “modulate” and “demodulate” computer data to analog format and back
- Clear distinction between the two networks is getting fuzzier with modern multimedia networks

IBM SNA Architecture

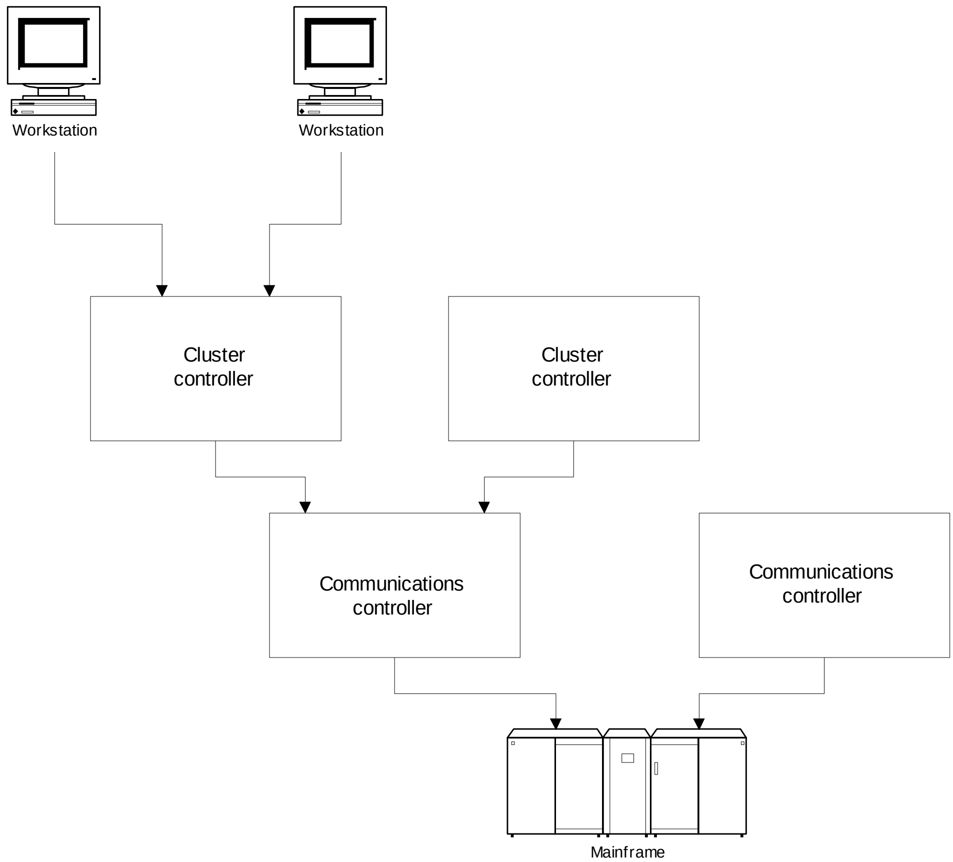


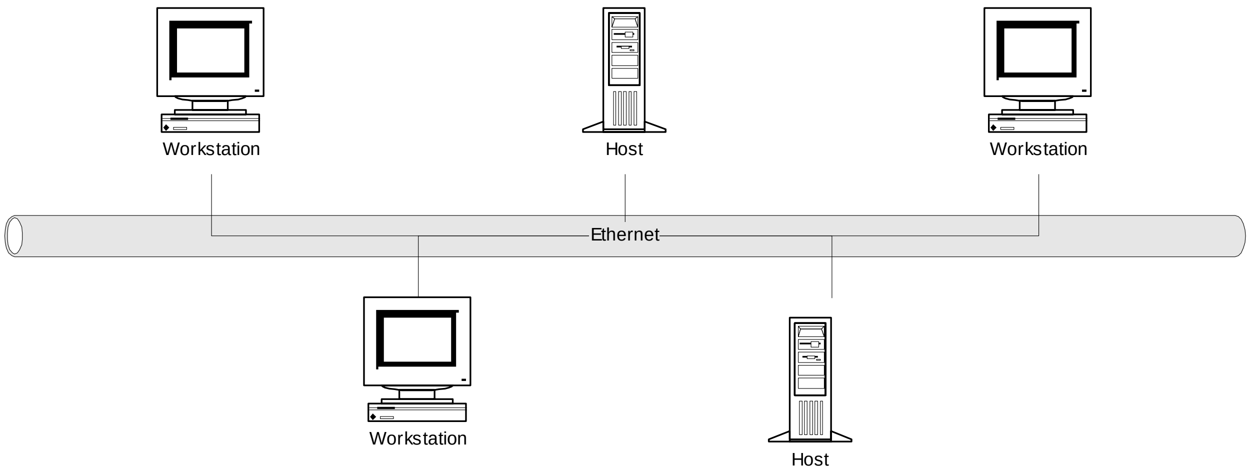
Figure 1.5 IBM Systems Network Architecture Model

Notes

- IBM System Network Architecture (SNA) is a major step in network architecture
- SNA is based on multitude of (dumb) terminals accessing a mainframe host at a remote location

DCE with LAN

DCE.. Distributed Computing Environment

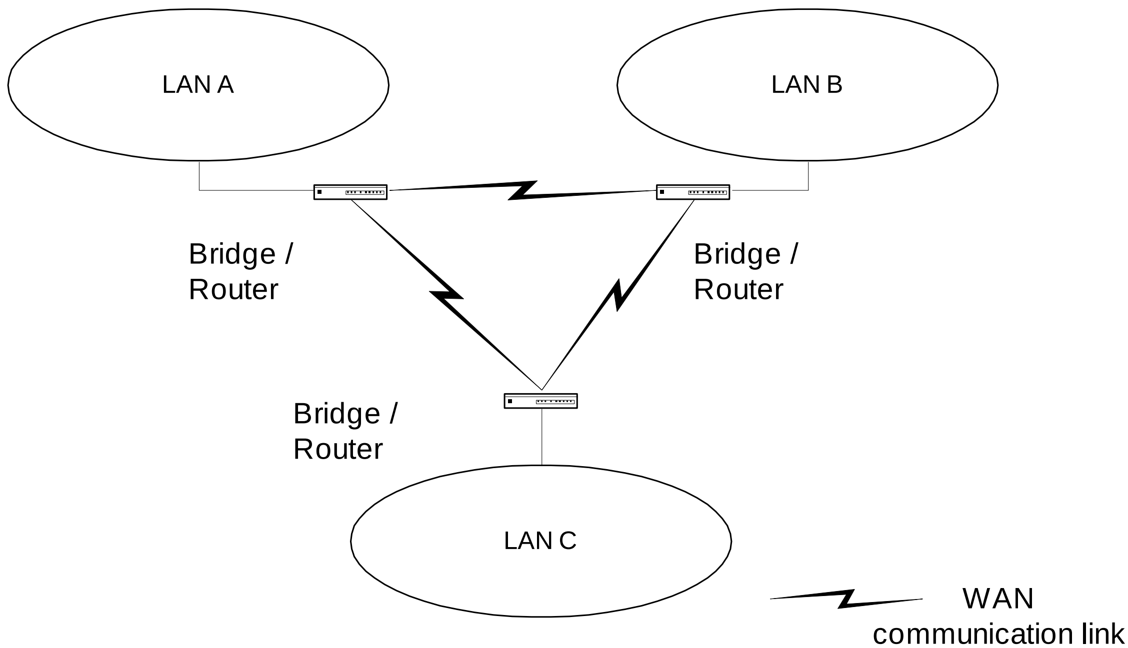


(a) Hosts and Workstations on Local LAN

Notes

- Driving technologies for DCE:
  - Desktop processor
  - LAN
  - LAN - WAN network

LAN-WAN Network



Notes

- Major impacts of DCE:
  - No more monopolistic service provider
  - No centralized IT controller
  - Hosts doing specialized function
  - Client/Server architecture formed the core of DCE network

Client/Server Model

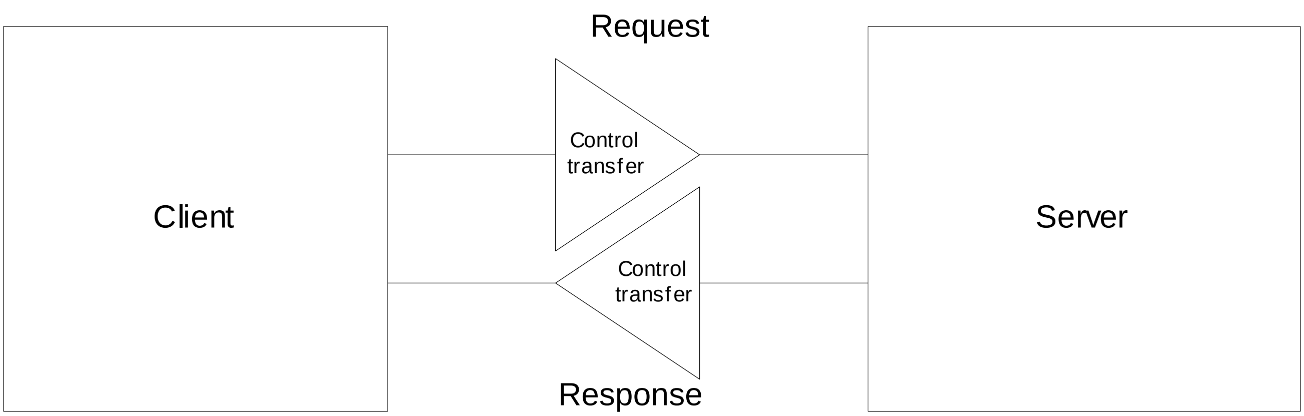


Figure 1.7 Simple Client-Server Model

Notes

- Post office analogy; clerk the server, and the customer the client
- Client always initiates requests
- Server always responds
- Notice that control is handed over to the receiving entity.

Client/Server Examples

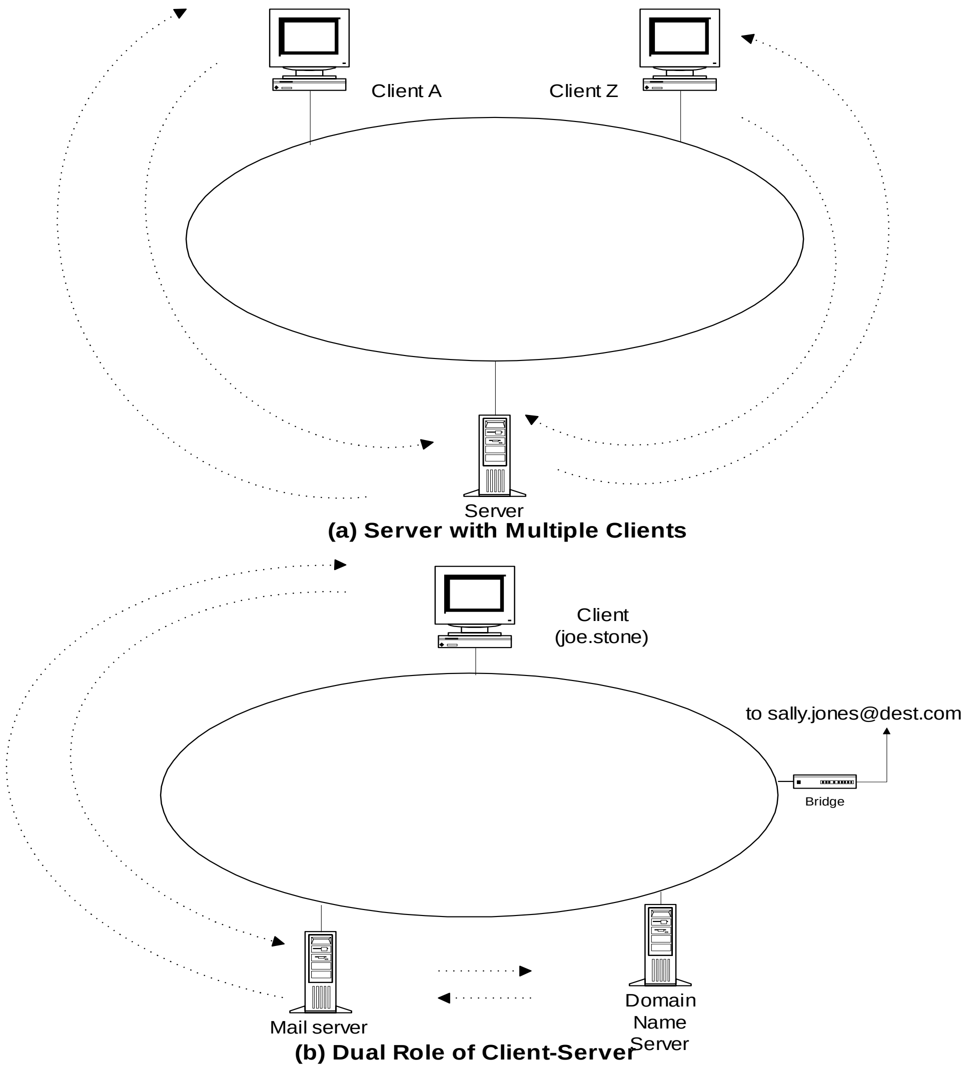


Figure 1.8 Client-Server in Distributed Computing Environment

Notes

TCP/IP Based Networks

- TCP/IP is a suite of protocols
- Internet is based on TCP/IP
- IP is Internet protocol at the network layer level
- TCP is connection-oriented transport protocol and ensures end-to-end connection
- UDP is connectionless transport protocol and provides datagram service
- Internet e-mail and much of the network mgmt. messages are based on UDP/IP
- ICMP part of TCP/IP suite

Notes

Internet Configuration

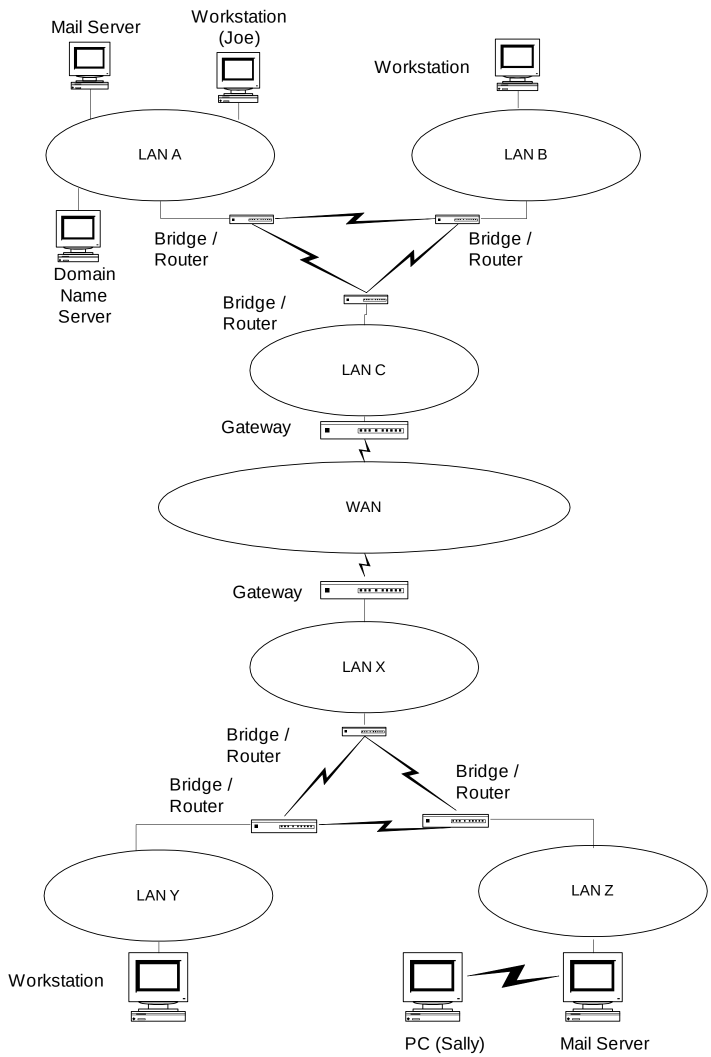


Figure 1.9 Internet Configuration

Notes

- Walk through the scenario of e-mail from Joe to Sally

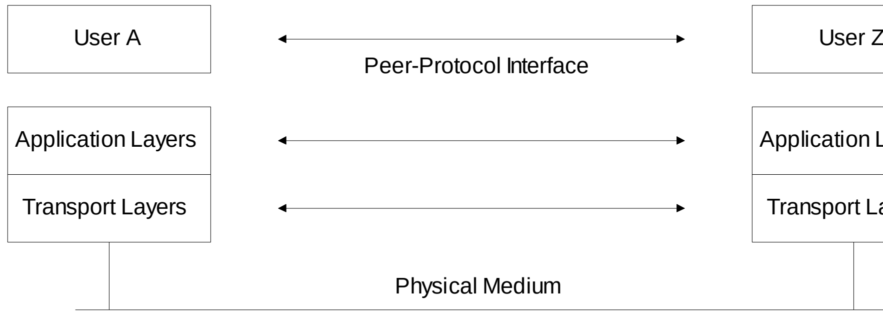
Architecture, Protocols and Standards

- Communication architecture
    - Modeling of communication systems, comprising
        - functional components and
        - operations interfaces between them
- Communication protocols
    - Operational procedures
        - intra- and inter-modules
- Communication standards
    - Agreement between manufacturers on protocols of communication equipment on
        - physical characteristics and
        - operational procedures

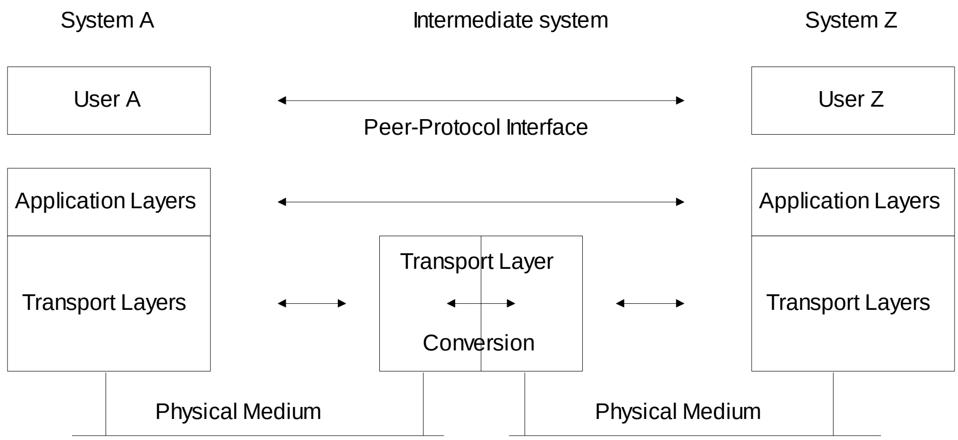
Notes

- Examples: (Students to call out)

Communication Architecture



(a) Direct Communication between End Systems



(b) Communication between End Systems via an Intermediate System

Figure 1.11 Basic Communication Architecture

Notes

- Inter-layer interface: user and service provider
- Peer-layer protocol interface
- Analogy of hearing-impaired student
- Role of intermediate systems
- Gateway: Router with protocol conversion as gateway to an autonomous network or subnet

OSI Reference Model

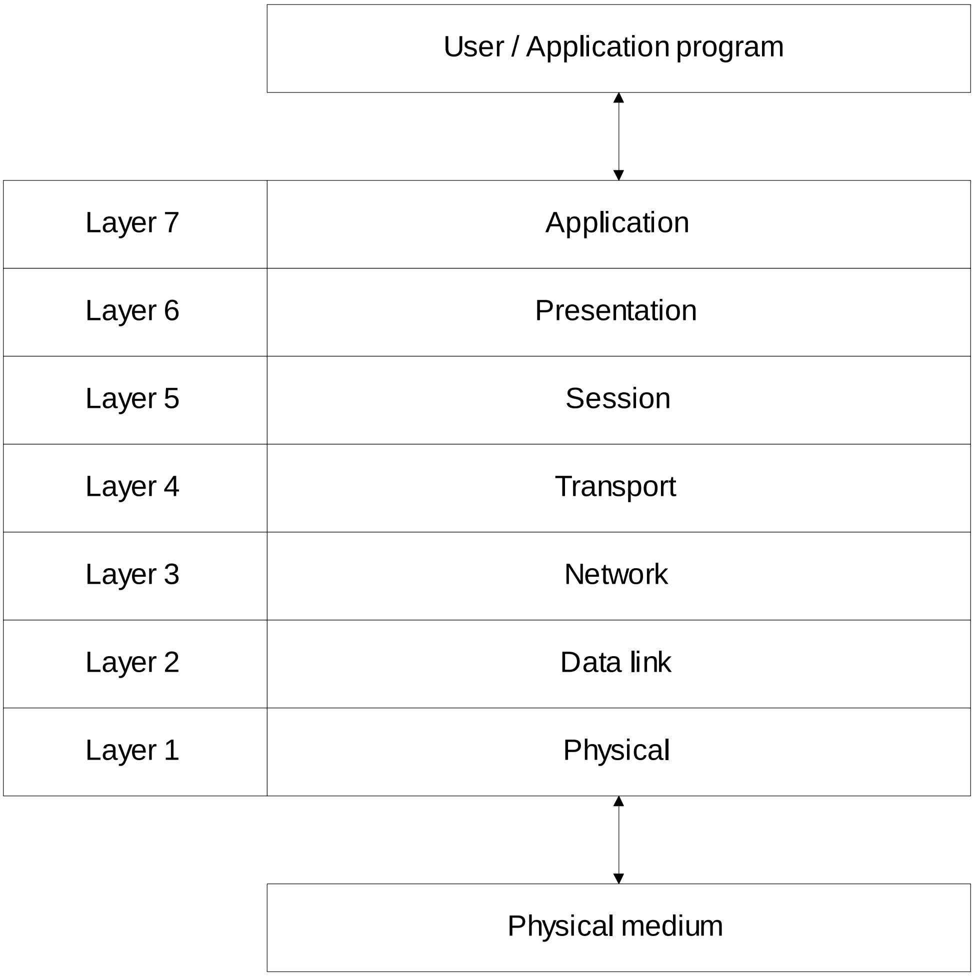


Figure 1.12 OSI Protocol Layers

Notes

- Importance of the knowledge of layer structure in NM

OSI Layers and Services

Layer No.	Layer Name	Salient services provided by the layer
1	Physical	-Transfers to and gathers from the physical medium raw bit data -Handles physical and electrical interfaces to the transmission medium
2	Data link	-Consists of two sublayers: Logical link control (LLC) and Media access control (MAC) -LLC: Formats the data to go on the medium; performs error control and flow control -MAC: Controls data transfer to and from LAN; resolves conflicts with other data on LAN
3	Network	Forms the switching / routing layer of the network
4	Transport	-Multiplexing and de-multiplexing of messages from applications -Acts as a transparent layer to applications and thus isolates them from the transport system layers -Makes and breaks connections for connection-oriented communications -Flow control of data in both directions
5	Session	-Establishes and clears sessions for applications, and thus minimizes loss of data during large data exchange
6	Presentation	-Provides a set of standard protocols so that the display would be transparent to syntax of the application -Data encryption and decryption
7	Application	-Provides application specific protocols for each specific application and each specific transport protocol system

Notes

- Importance of services offered by different layers and the protocol conversion at different layers in NM

PDU Communication Model

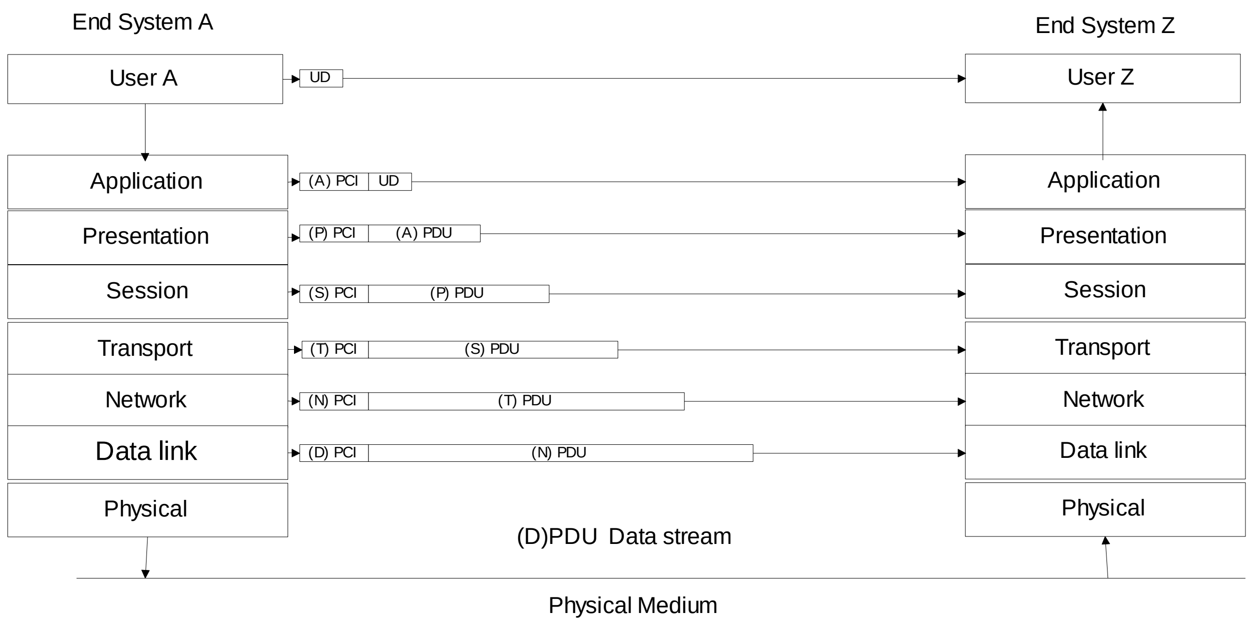
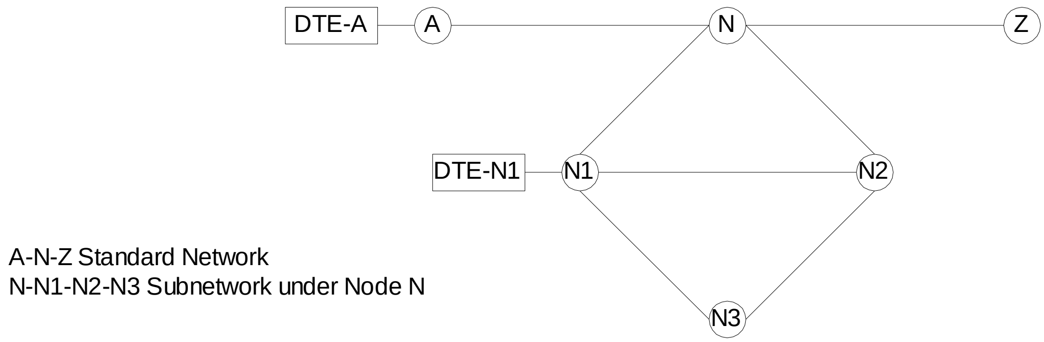


Figure 1.14 PDU Communication Model between End Systems

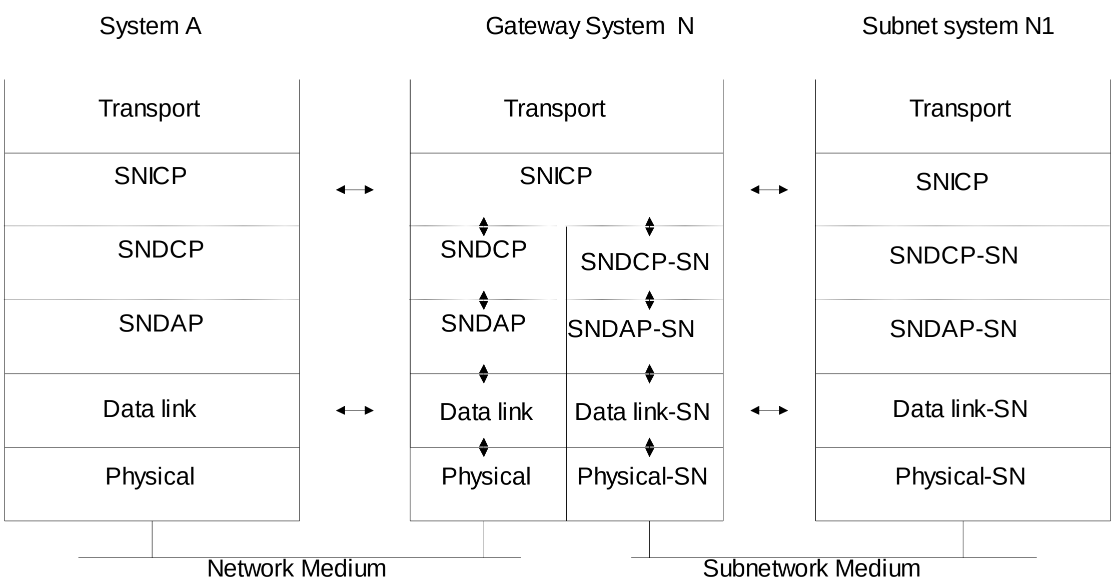
Notes

- What is the relevance of PDU model in NM?

Gateway



(a) Network configuration



(b) Protocol Communication

Figure 1.17 Gateway Communication to Proprietary Subnetwork

Notes

- cc:mail from a station in Novel IPX network to an Internet station with SMTP e-mail

SNA, OSI, and Internet

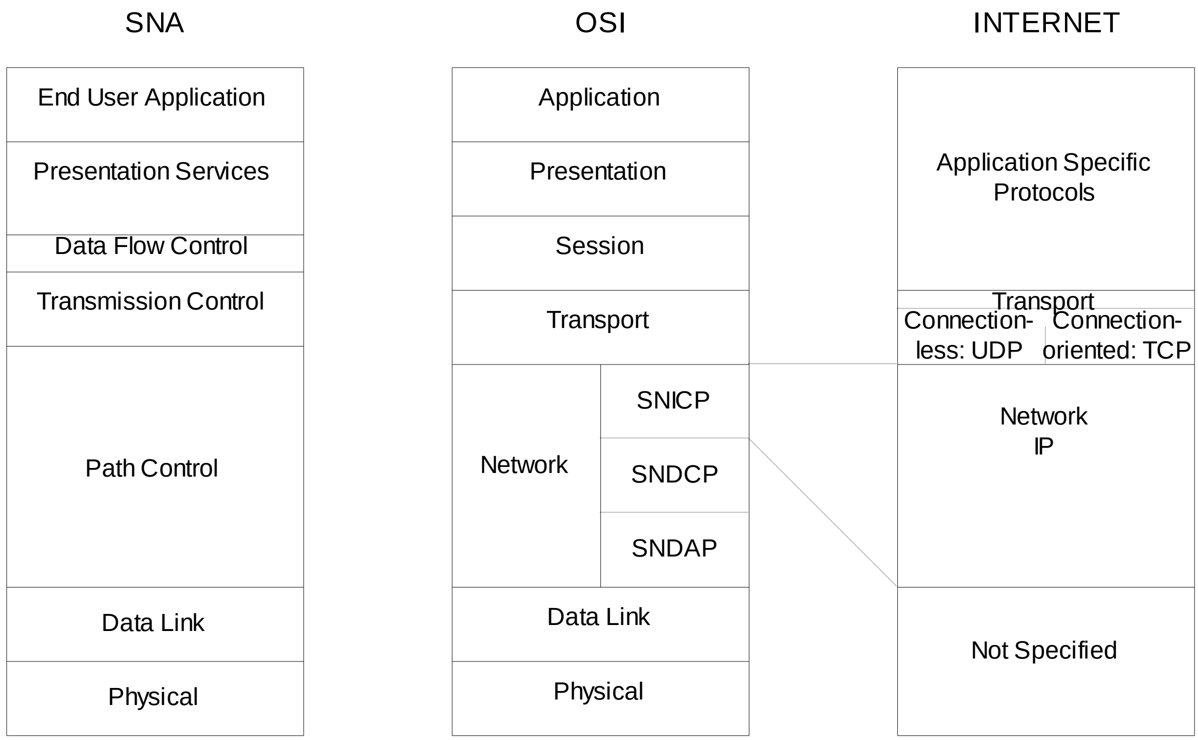


Figure 1.18 Comparison of OSI, Internet, and SNA Protocol Layer Models

Notes

- Similarity between SNA and OSI
- Simplicity of Internet; specifies only layers 3 and 4
- Integrated application layers over Internet
- Commonality of layers 1 and 2 - IEEE standard

Application Protocols

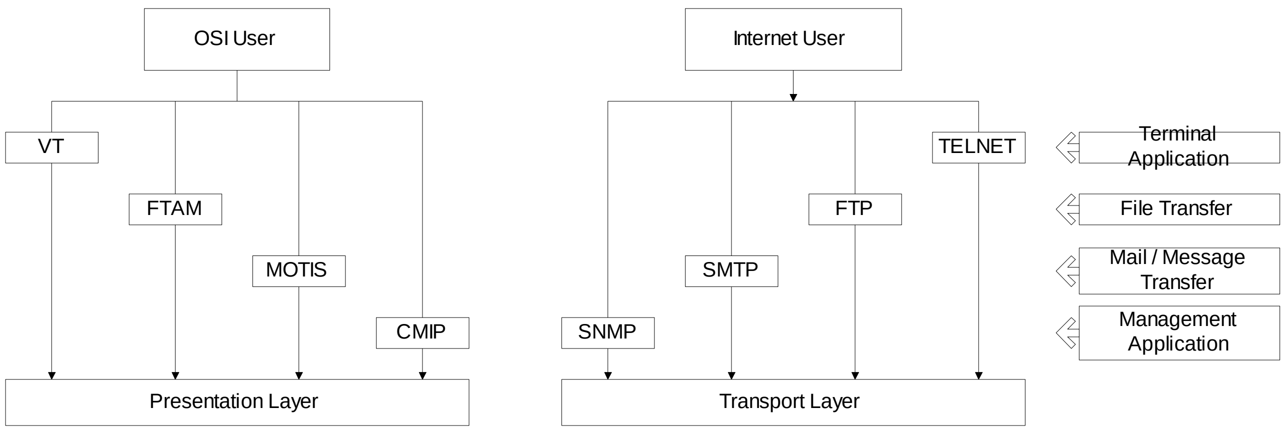


Figure 1.19 Application Specific Protocols in ISO and Internet Models

Notes

Internet user	OSI user
Telnet	Virtual Terminal
File Transfer Protocol	File Transfer Access & Mgmt
Simple Mail Transfer Protocol	Message-oriented Text Interchange Standard
Simple Network Management Protocol	Common Management Information Protocol

NM Case Histories

- The case of the Footprint
- Case of the crashing bridge

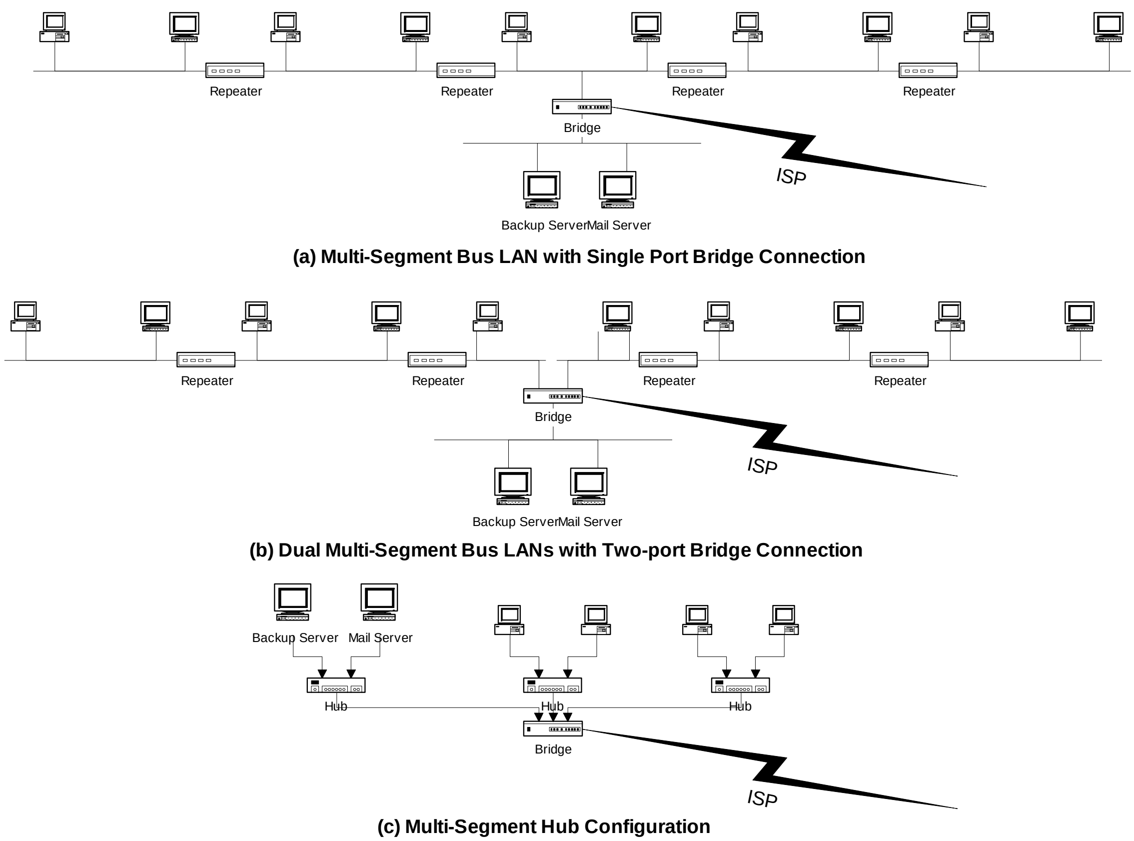


Figure 1.20 Case History 2: Network Configuration Evolution

Notes

Common Network Problems

- Loss of connectivity
- Duplicate IP address
- Intermittent problems
- Network configuration issues
- Non-problems
- Performance problems

Notes

Challenges of IT Managers

- Reliability
- Non-real time problems
- Rapid technological advance
- Managing client/server environment
- Scalability
- Troubleshooting tools and systems
- Trouble prediction
- Standardization of operations - NMS helps
- Centralized management vs “sneaker-net”

Notes

Network Management

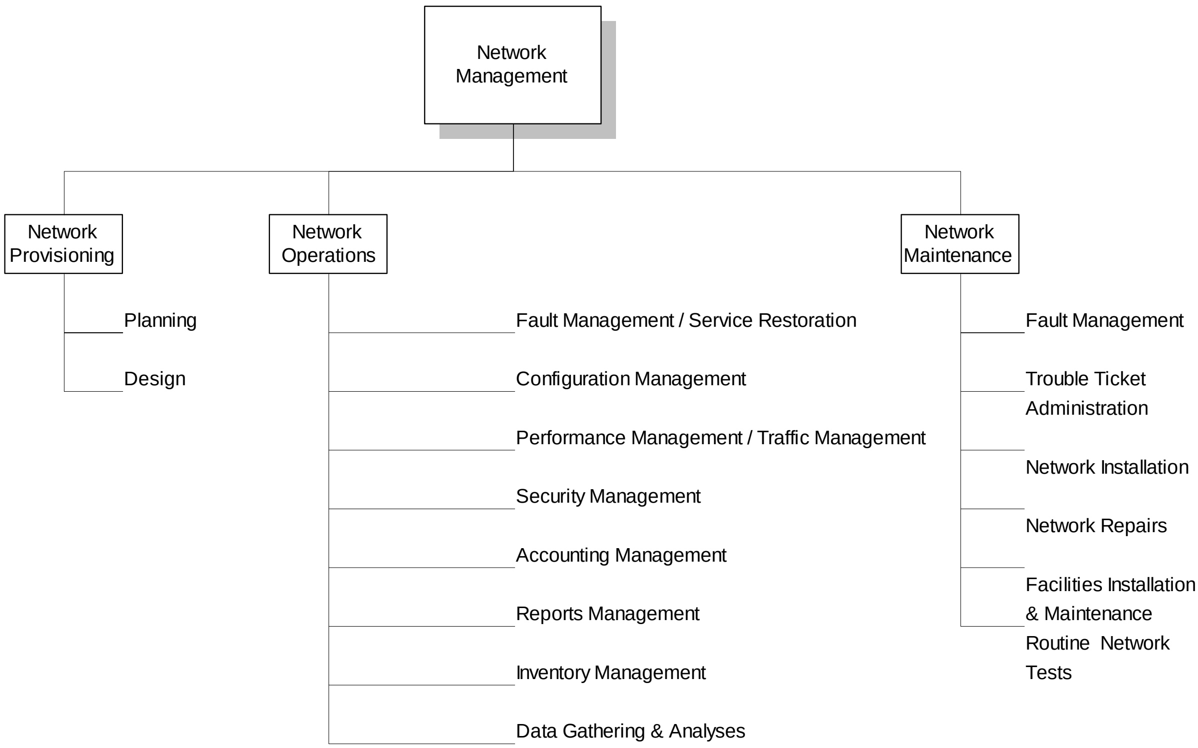


Figure 1.21 Network Management Functional Groupings

Notes

- OAM&P
    - Operations
    - Administration
    - Maintenance
    - Provisioning

NM Functional Flow Chart

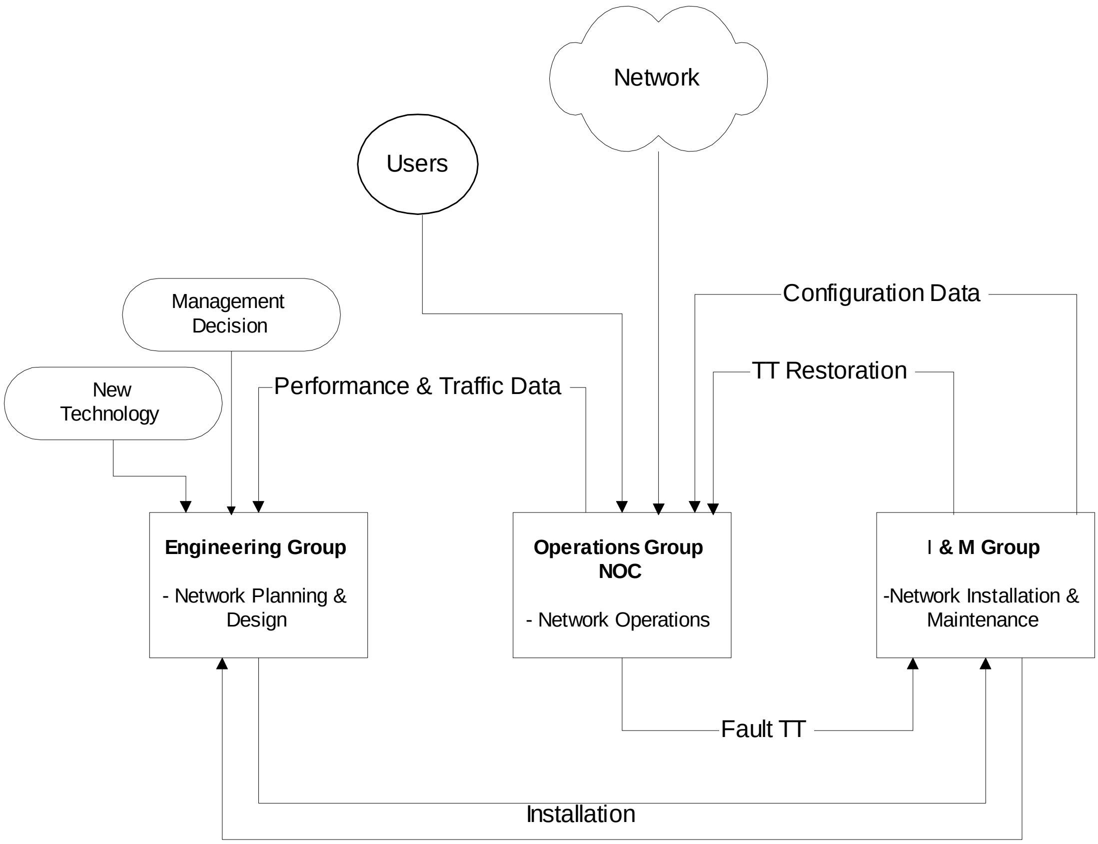


Figure 1.22. Network Management Functional Flow Chart

Notes

NM Components

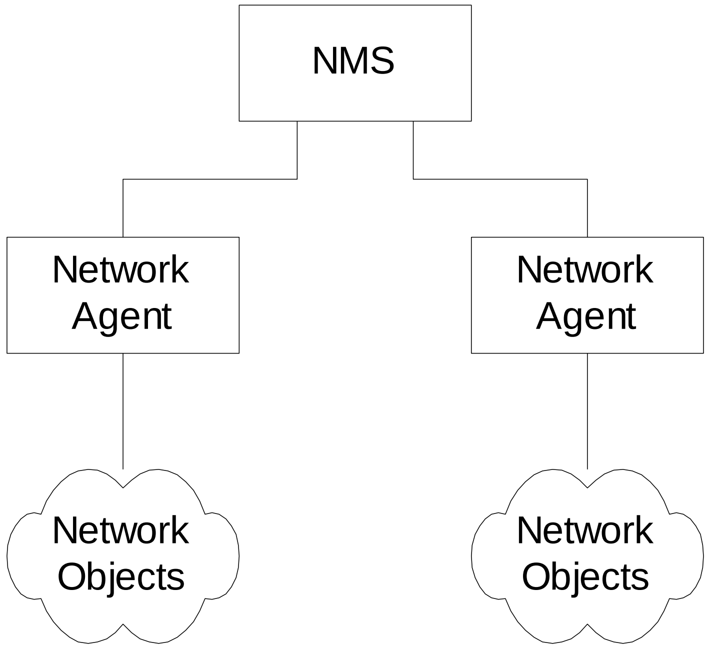
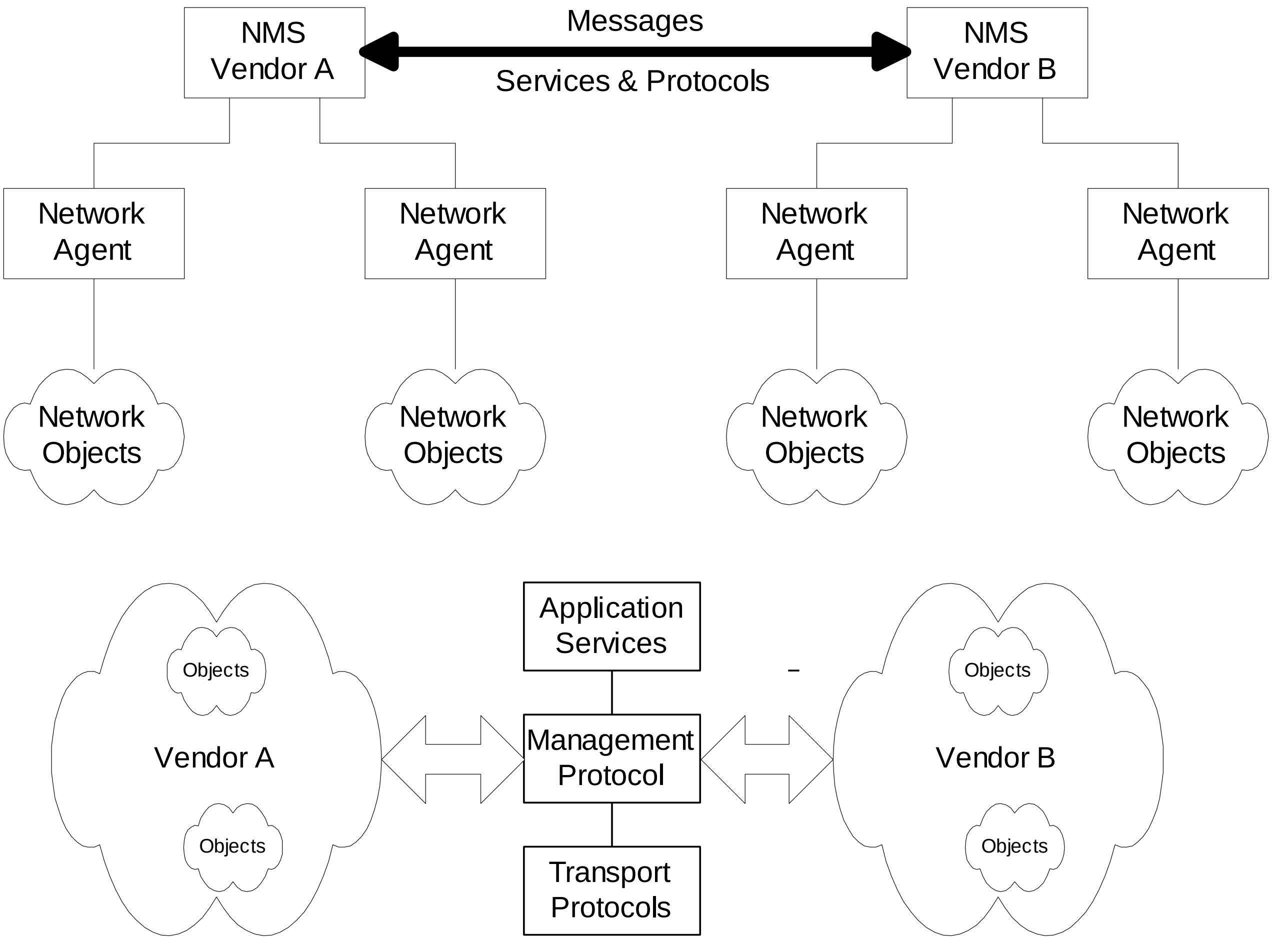


Figure 1.24 Network Management Components

Notes

Interoperability



(b) Services and Protocols

Figure 1.23 Network Management Dumbbell Architecture

Notes

- Message exchange between NMSs managing different domains

Status and Future Trends

- Status:
  - SNMP management
  - Limited CMIP management
  - Operations systems
  - Polled systems
- Future trends:
  - Object-oriented approach
  - Service and policy management
  - Business management
  - Web-based management

Notes

Chapter 2

Computer Network Technology

Technology and Management

- What are the technologies that need to managed?
- Challenges of technological progress on network management

Notes

Computer Network Technology

- Network comprises
  - Nodes
  - Links
- Topology: How they're configured
  - LAN
  - WAN

Notes

- Distinction between LAN and WAN

LANs

- Type of LANs
  - Ethernet
  - Fast Ethernet
  - Gigabit Ethernet
  - Half-duplex Vs Full-duplex
  - Switched Ethernet
  - VLAN
  - Token ring
  - FDDI
  - ATM / LANE

Notes

Nodes

- Hubs
- Bridges
- Remote bridges
- Routers
- Gateways
- Half bridge / half router
- Switches

Notes

WANs

- Facilities / Media
  - Wired
    - Copper
    - Coaxial
    - Fiber
  - Wireless
    - Terrestrial
    - Satellite
- Mode
  - Digital
  - Analog
- Services
  - POTS
  - ISDN
  - Broadband

Notes

Basic LAN Topologies

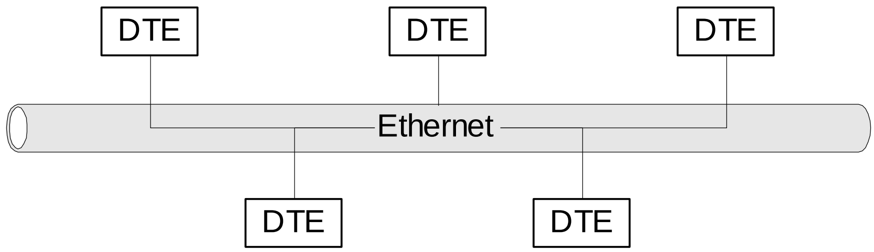


Figure 2.1(a) Bus Topology

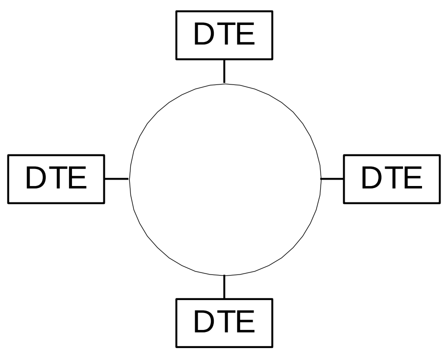


Figure 2.1(b) Ring Topology

Notes

- Bus topology
    - Used in Ethernet LAN family
    - Common shared medium
    - Randomized access (CSMA/CD)
    - Easy to implement
    - Lower utilization under heavy traffic 30%-40%
    - Single culprit could effect the entire LAN

Basic LAN Topologies

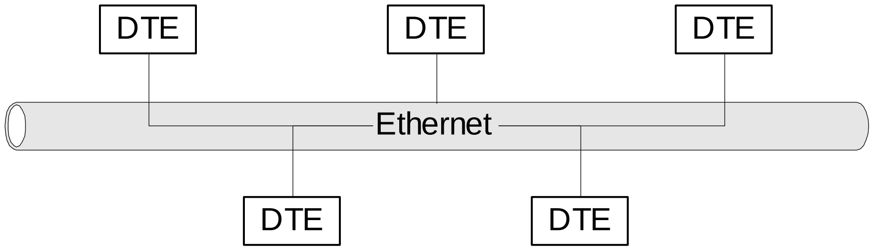


Figure 2.1(a) Bus Topology

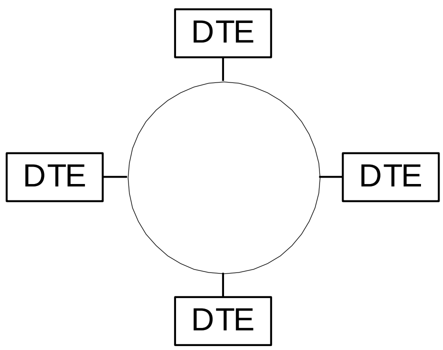


Figure 2.1(b) Ring Topology

Notes

- Ring Topology
    - Used in token ring and FDDI
    - Shared medium
    - Deterministic access
    - Master DTE has control
    - High utilization >90%

Star & Hybrid LAN Topologies

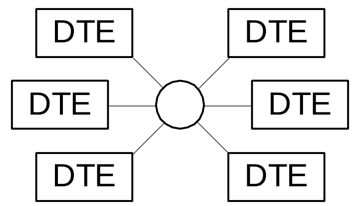


Figure 2.1(c) Star Topology

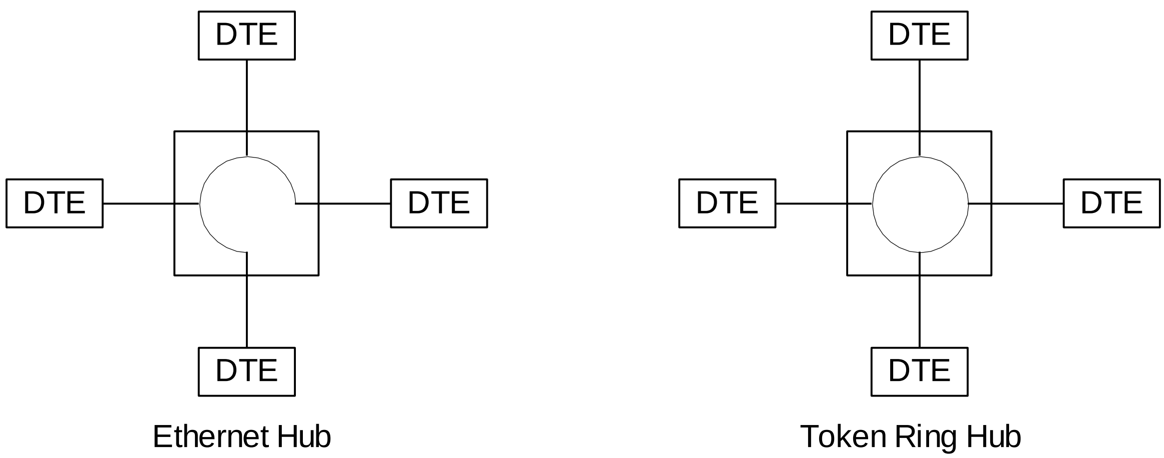


Figure 2.1(d) Hub Configurations

Notes

- Star topology used with bus and ring topology
- Hub is “LAN in a box”
- What does the electronic LAN inside the box look like?
- Why has hub become so popular?

A Campus Network

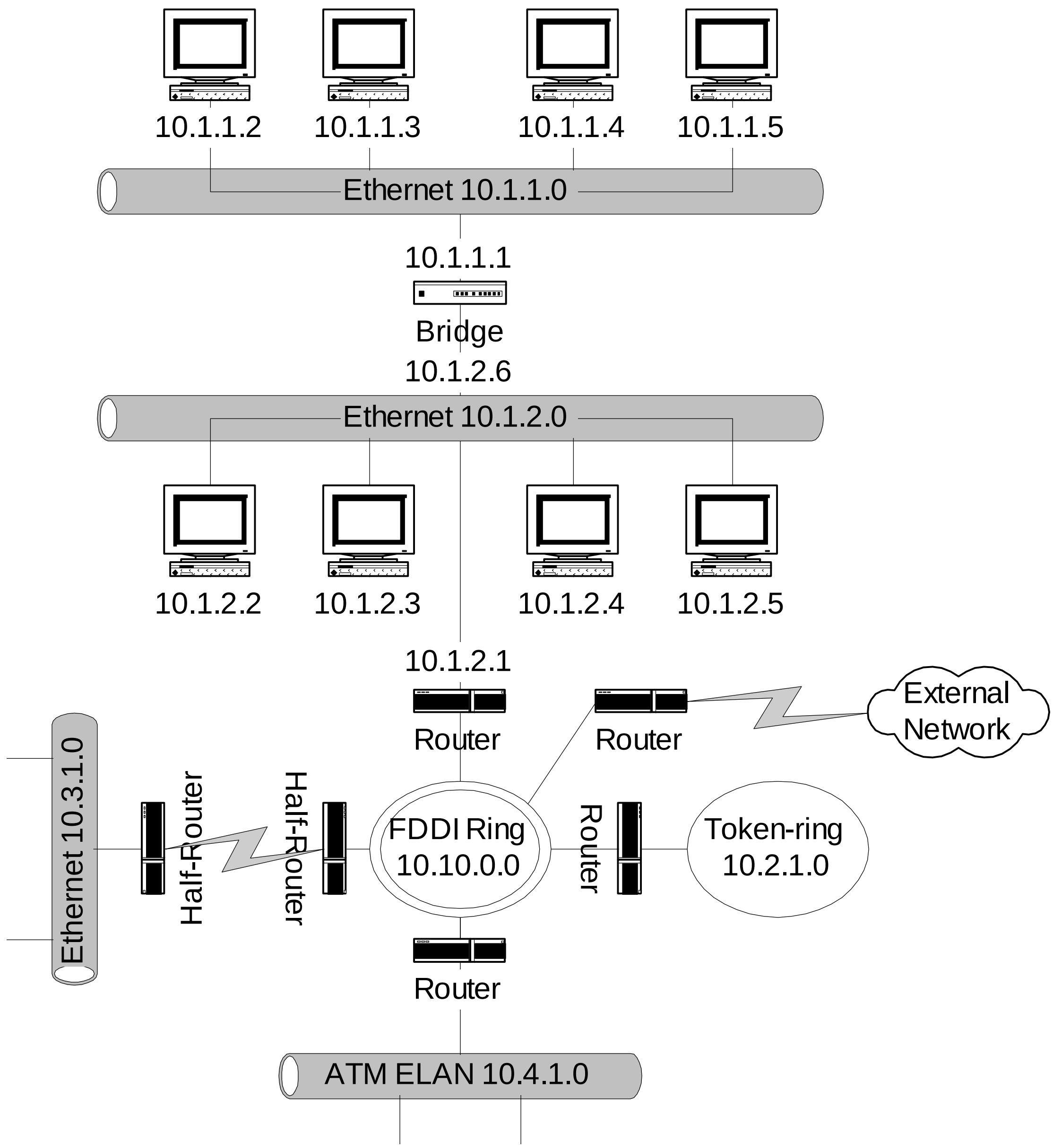


Figure 2.3 Campus Network of LANs

Notes

- ATM VLAN an alternative to FDDI backbone

WAN Topologies

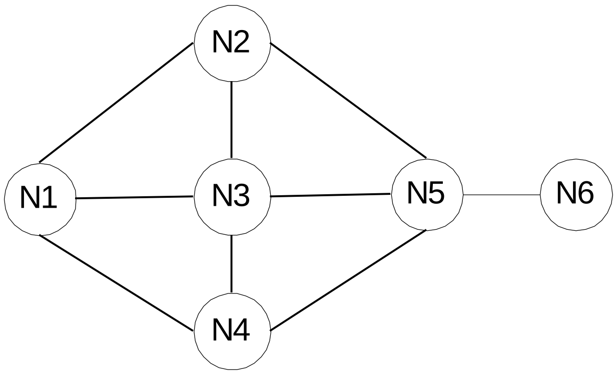


Figure 2.2(a) Mesh Topology

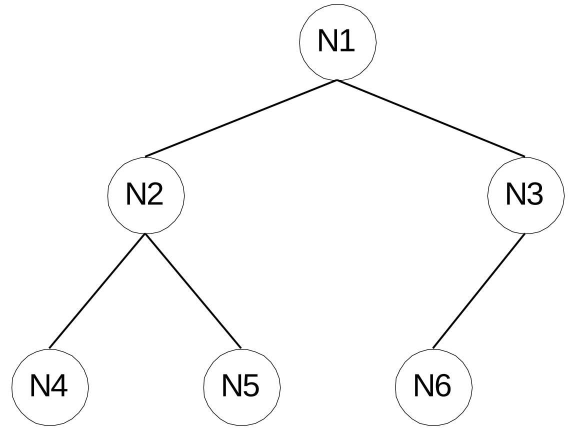


Figure 2.2(b) Tree Topology

Notes

- Mesh topology
    - Implemented in network layer level
    - Multiple paths between nodes
    - Flat topology
    - Redundancy
    - Load balancing
    - Shortest path
- Tree topology
    - Used with Ethernet bridges
    - Hierarchical
    - Efficient for small networks and special purpose networks

Ethernet

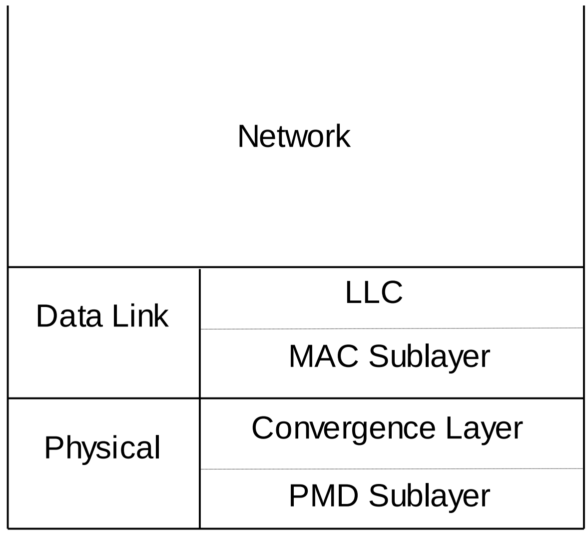
Table 2.1 Ethernet LAN Topology Limits

TYPE	DESCRIPTION	SEGMENT LENGTH	DROP CABLE
10Base2	Thin coax (0.25”)	200 meters	Not allowed
10Base5	Thick Coax (0.5”)	500 meters	Twisted pair: 50 meters
10Base-T	Hub topology	N/A	Twisted pair: 100 meters
10Base-F	Hub topology	N/A	2 km

Notes

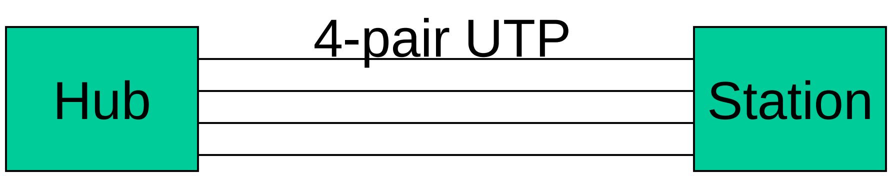
- IEEE 802.3 standard
- 10 Mbps data rate
- Collision - analogy of hollow pipe
- Principle of operation; CSMA/CD
- Segment length and drop cable length
- Minimum size of packet 64 bytes
- Maximum size of packet 1500 bytes
- Hub configuration

Fast Ethernet



LLC Logical Link control
MAC Medium Access Control
PMD Physical Medium Dependent

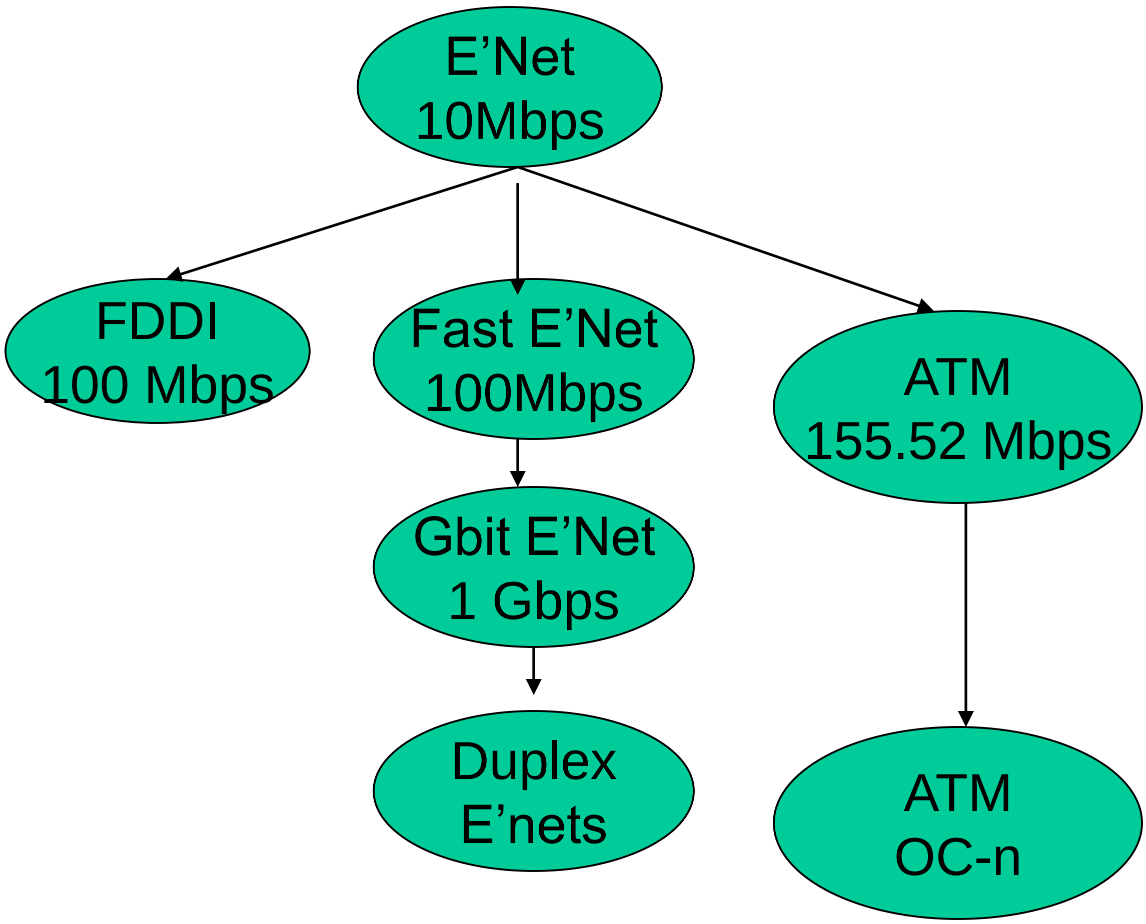
Figure 2.4 100Base-T Fast Ethernet Protocol Architecture



Notes

- Rationale
  - Max drop length 100m => Max round-trip time 1/10 of Ethernet; hence 10 times data rate
- Standard 100Base-T4
- Compatibility with 10BaseT
- UTP limitation; Use 4-pair UTP @ 25 Mbps/pair
- Alternatives: 2-pair 100BaseTX Cat 5(Max 100 m) and 100Base FX optical fiber (Max 2 km)

LAN Data Rate Race



Notes

Gigabit Ethernet

Table 2.2 Gigabit Ethernet Topology Limits

	9 micron Single-Mode	50 micron Single Mode	50 micron Multimode	62.5 micron Multimode	Balance Shielded Cable	UTP
1000BASE-LX	10 km	3 km	550 m	440 m	-	-
1000BASE-SX	-		550 m	260 m	-	-
1000BASE-CX	-		-	-	25 m	-
1000BASE-T	-		-	-	-	100 m

Notes

- Packet size 512 bytes, slot size 4.096 microseconds
- Minimum frame size 64 bytes for backward compatibility; Slot filled with carrier extension
- Packet bursts with no idle time between frames increases efficiency

Switched Ethernet

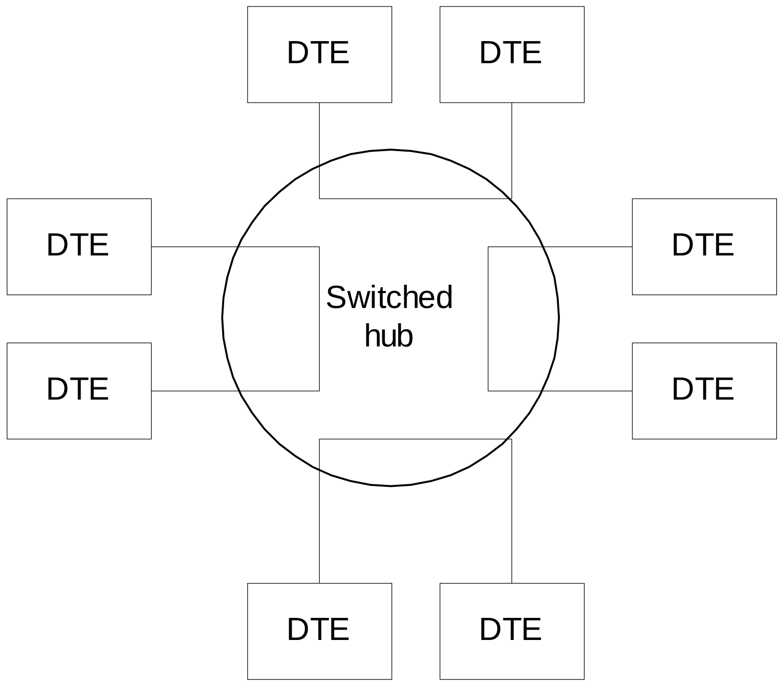


Figure 2.8 Switched Ethernet Hub

Notes

- Maximum throughput increased ~N/2 in N-port hub
- Snooping capability lost for management

Client/Server Configuration using Switched Hub

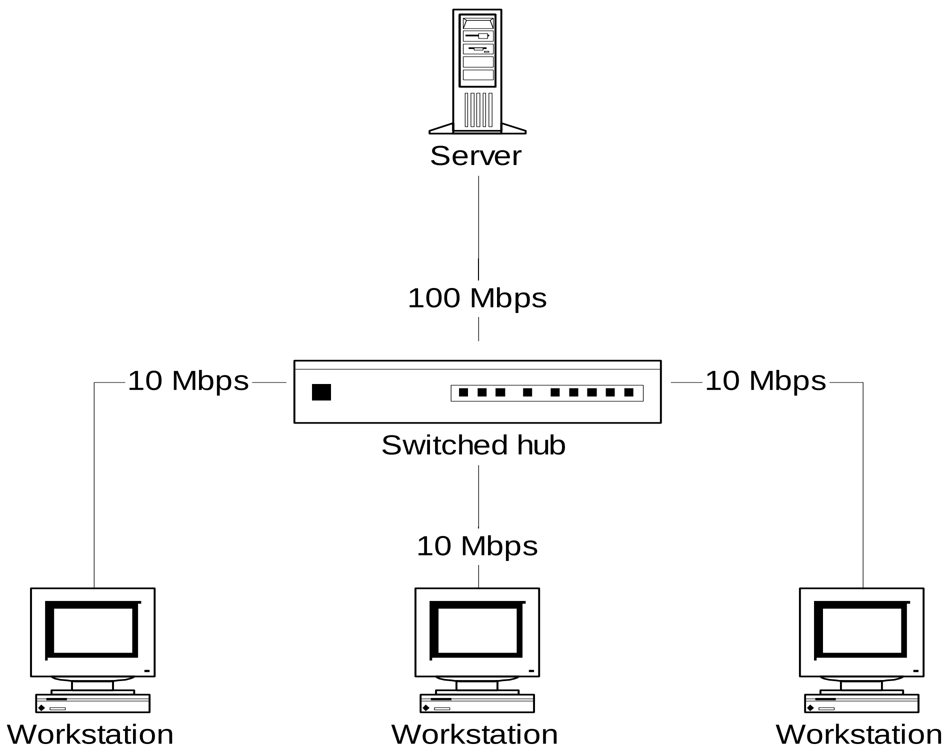


Figure 2.9 Switched Hub in Client-Server Configuration

Notes

Virtual LAN

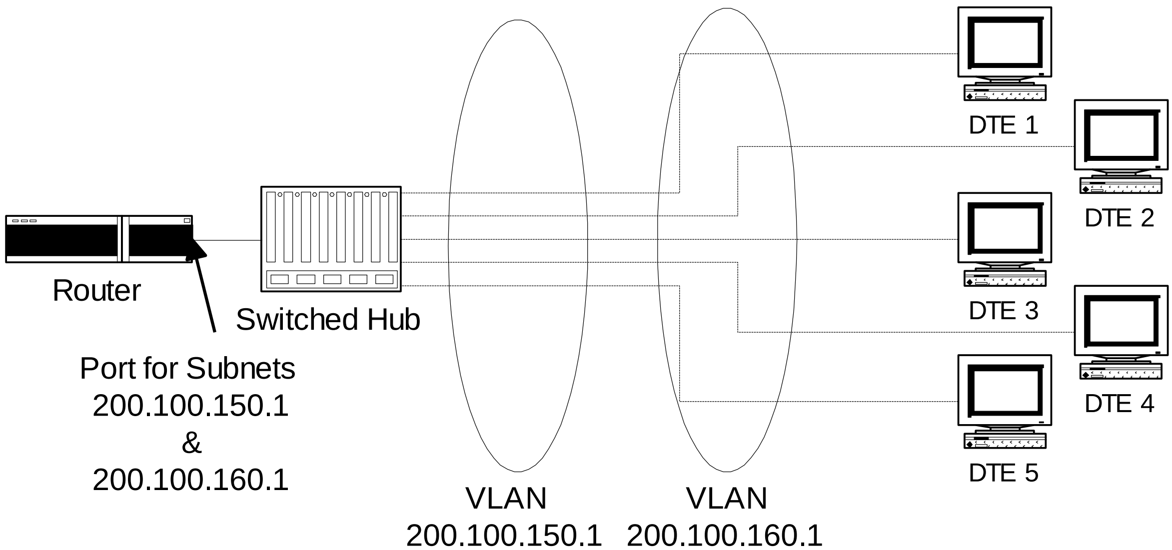


Figure 2.10 Virtual LANs

Notes

- Switched hub enables establishing virtual LANs
- Permits switching stations between LANs without physical moving of equipment
- Walk through scenario

Token Ring

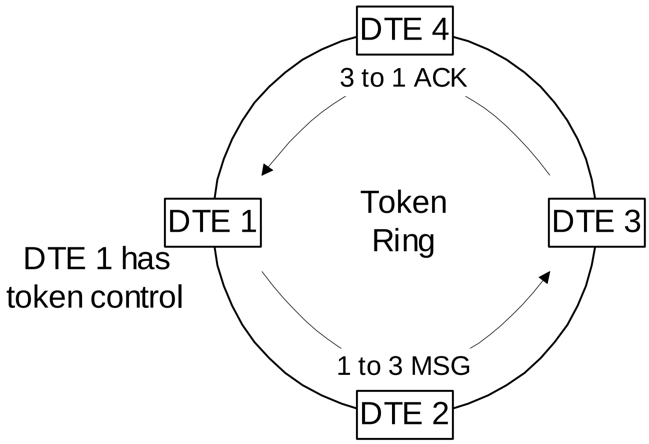


Figure 2.11 Token Ring LAN

Notes

- Adopted by IBM
- IEEE 802.5 standard
- Data rates of 4Mbps and 16 Mbps
- Single and dual ring LANs

Dual Ring TR LAN

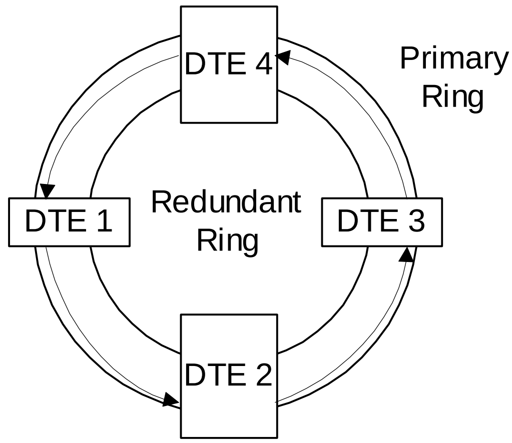


Figure 2.12(a) Token Ring Dual Ring Management

Notes

Failure Recovery in TR LAN

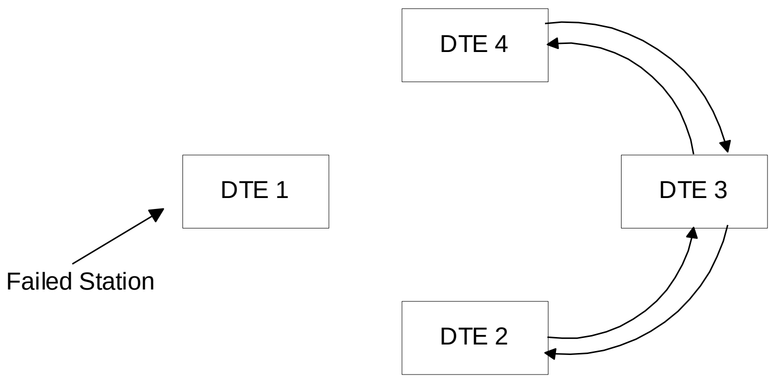


Figure 2.12(b) Token Ring DTE Isolation

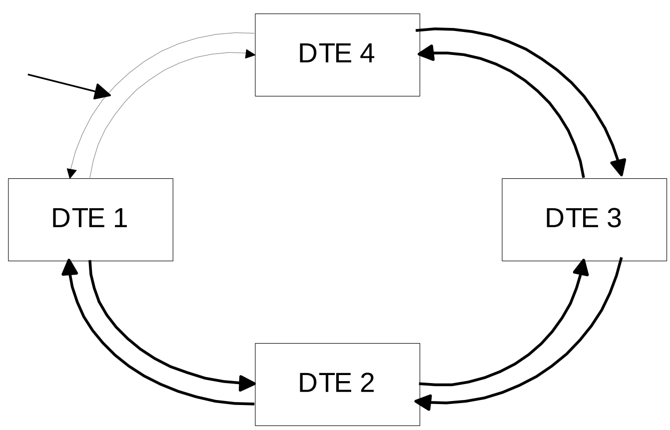
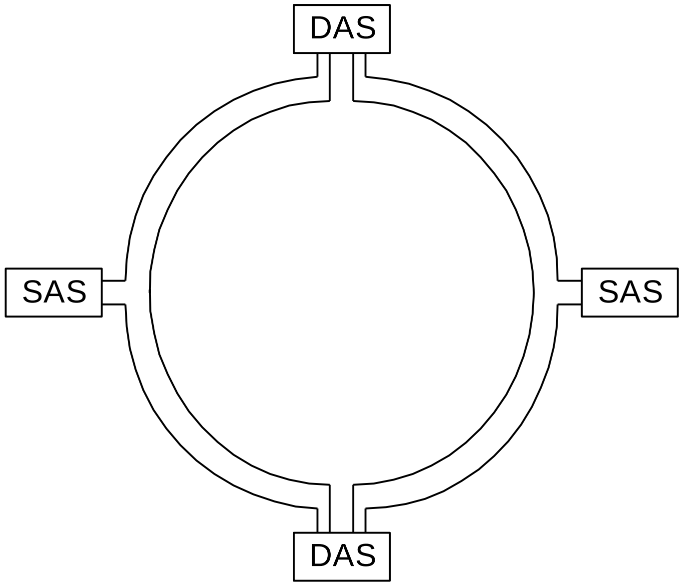


Figure 2.12(c) Token Ring Segment Isolation

Notes

- Station failure recovery
- Link failure recovery

FDDI



SAS Single Attached Station
DAS Dual Attached Station

Figure 2.13(a) Dual Ring FDDI Network Configuration

Notes

- Uses fiber optics medium
- Modified token ring protocol
- Data rate 100 Mbps
- Segment length 100 km
- 500 stations in the ring with max separation of 2 km
- Single and dual attached stations
- Dual attached stations load share the two rings

Basic Network Nodes

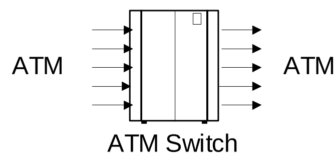


Figure 2.14(a) Switch

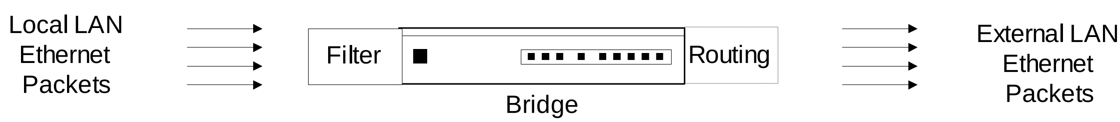


Figure 2.14(b) Bridge

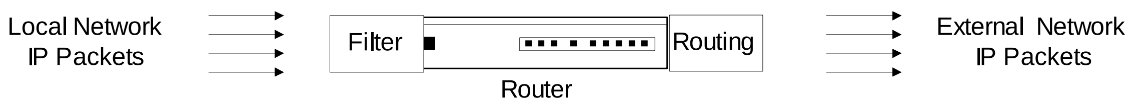


Figure 2.14(c) Router

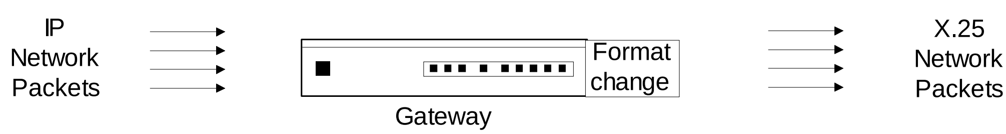


Figure 2.14(d) Router

Figure 2.14 Basic Network Node Components

Notes

Network Node Components

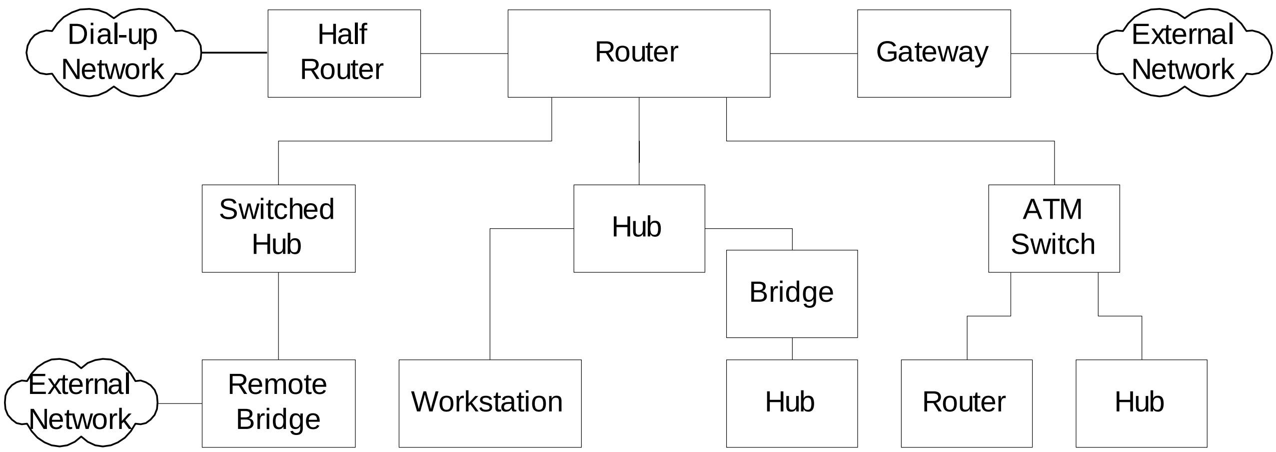


Figure 2.15 Networked Components

Notes

- Hubs
- Bridges
- Remote bridges
- Routers
- Gateways
- Half bridge / half router
- Switches

Hubs

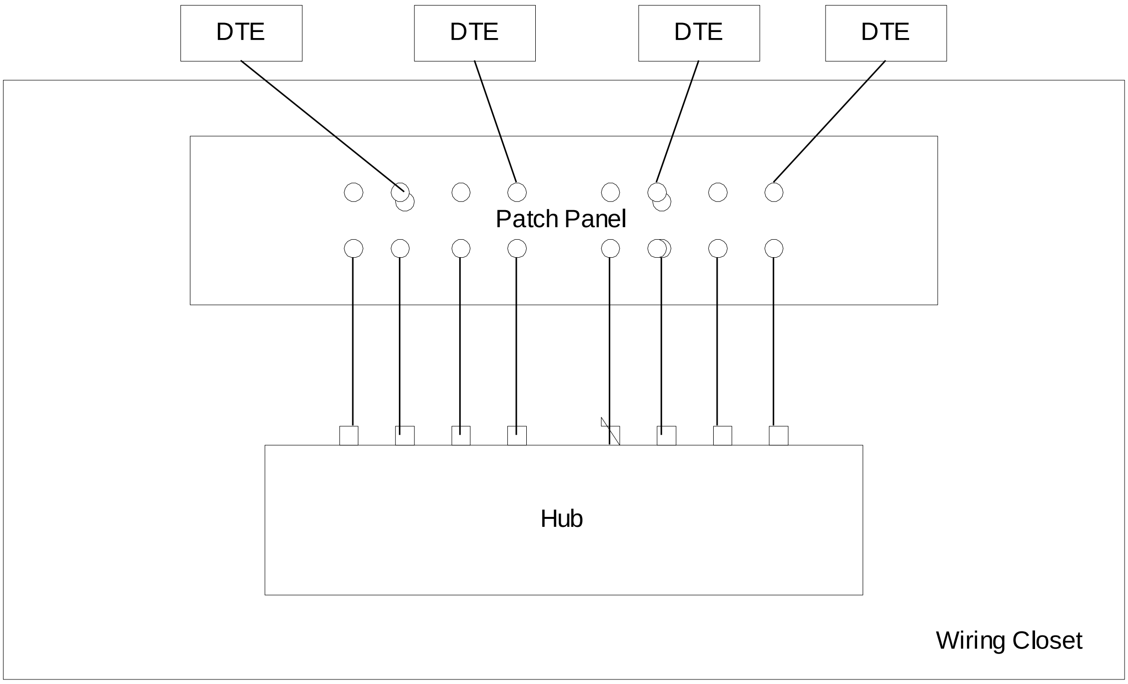


Figure 2.16(a) Hub Configuration

Notes

- Hub is a platform
- Function dependent on what is housed
    - LAN
    - Switched LAN
    - Bridge

Stacked Hubs

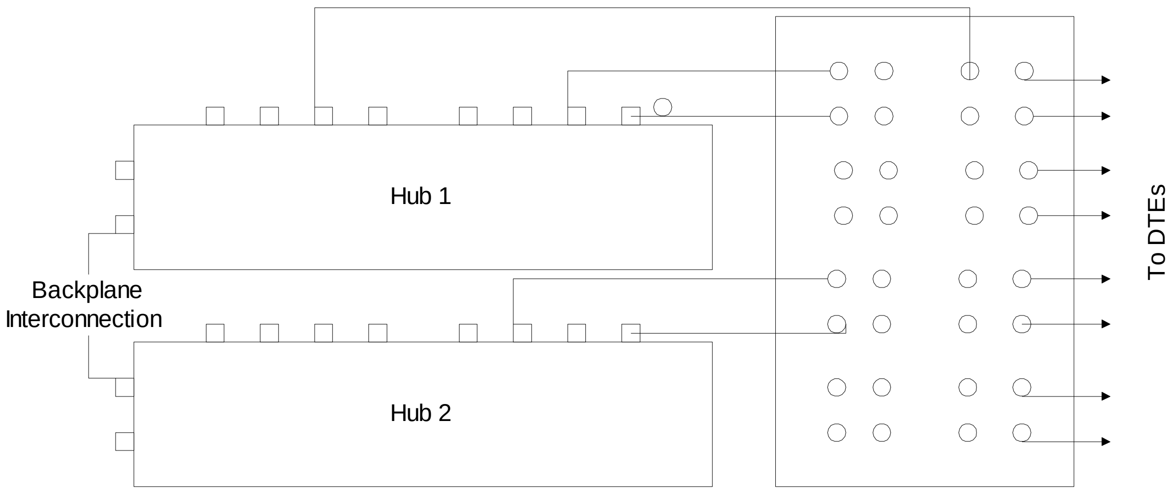
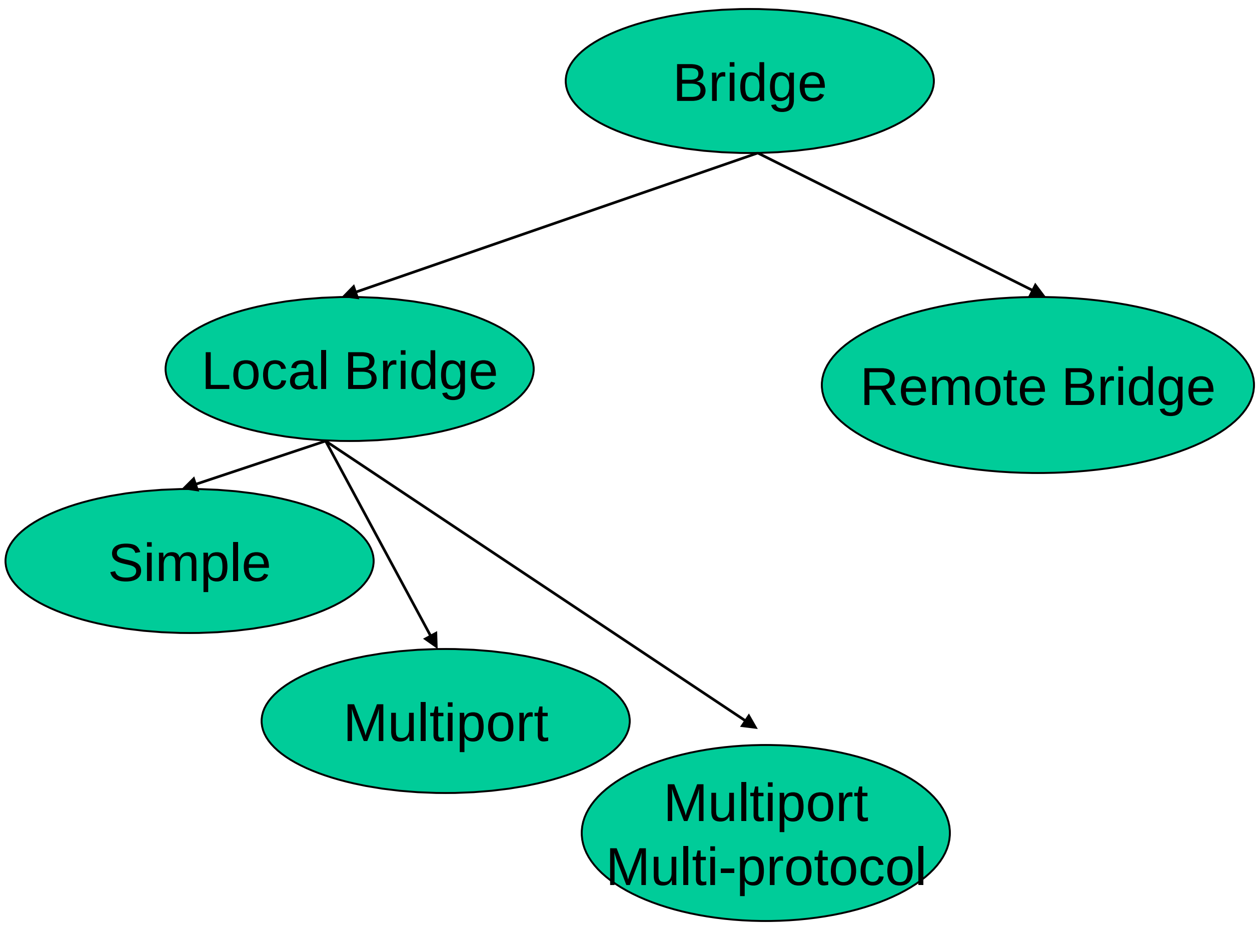


Figure 2.16(b) Stacked Hub

Notes

- Hub ports can be scaled up using stacked hubs
- Stacked hub
  - extend back plane
  - connected as daisy chain

Bridges



Notes

- Bridges two nodes at data link control layer
    - Ethernet: tree topology, transparent bridge
    - Token ring:mesh topology, source routing bridge
- Remote bridge uses WAN interface cards;
  same protocol used at both ends
- Ethernet bridge is a learning bridge

Routers

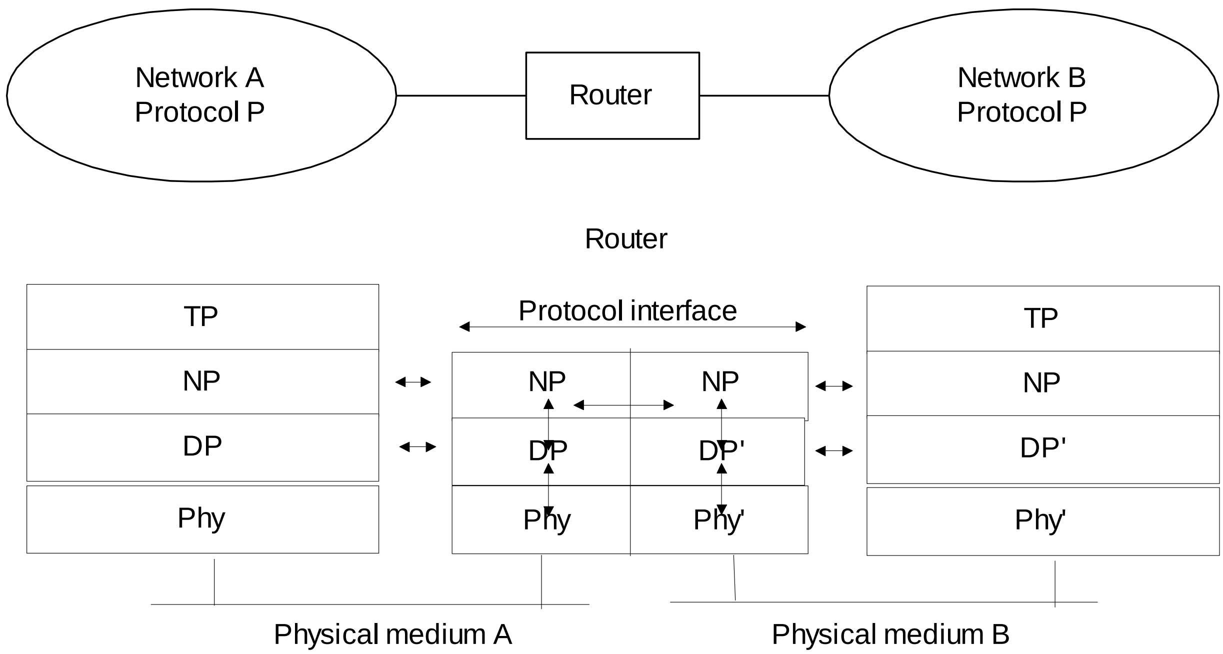


Figure 2.21 Router Configuration

Notes

- Routers operate at network layer
- Routes packets between nodes of similar network protocols
- Routing table used to route packets
- DLC and Physical layers could be different under the same common network layer protocol

Gateway

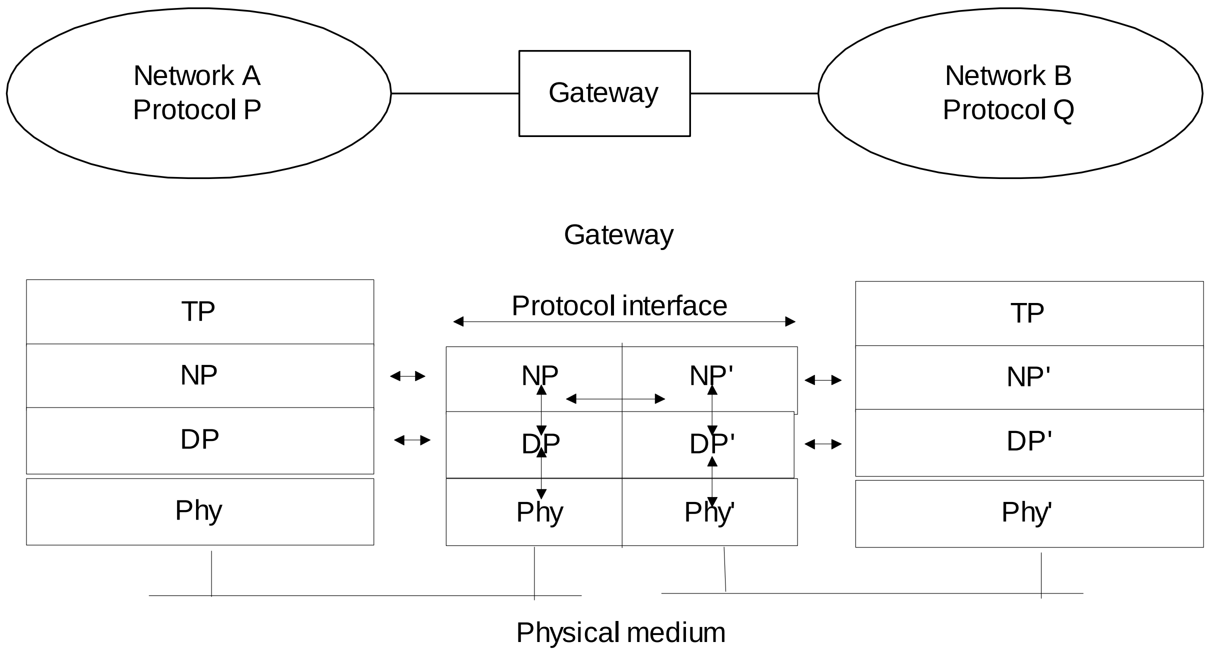


Figure 2.22 Gateway Configuration

Notes

- Gateway is router connecting two networks with dissimilar network protocols
- Gateway does the protocol conversion at the network layer
- Protocol converter does the conversion at the application layer

Tunneling

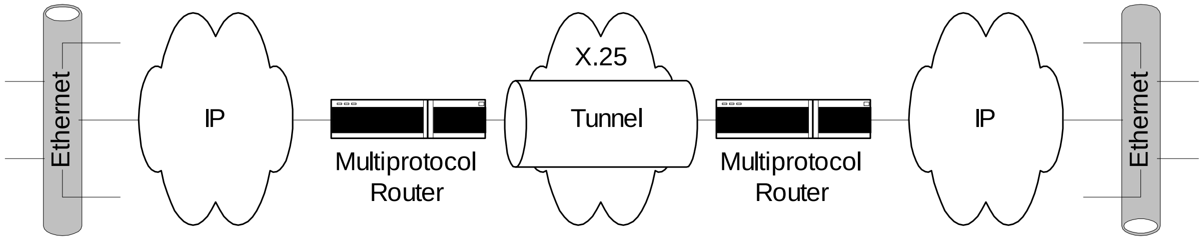
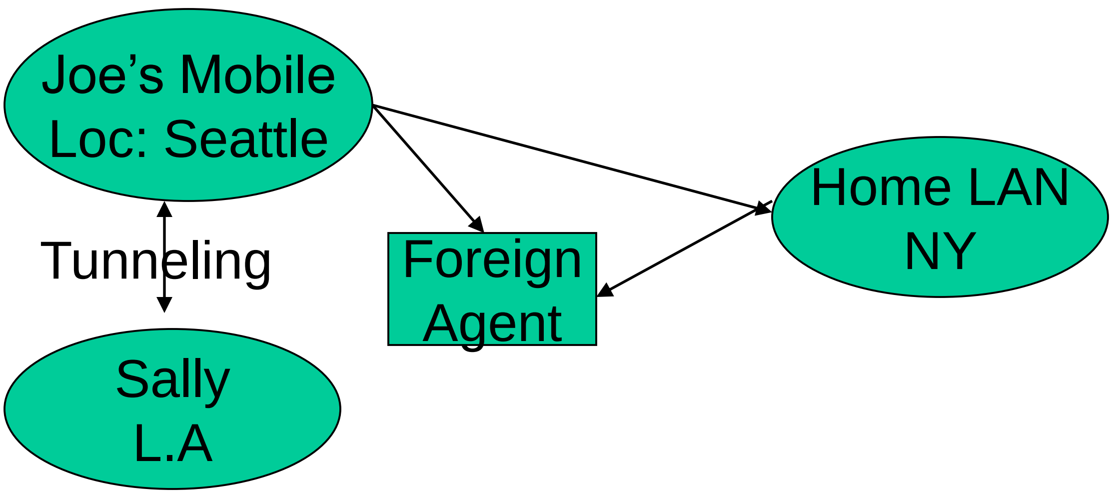


Figure 2.24 Tunneling Using Multiprotocol Routers



Notes

- Tunneling is transmission of packets (via multiprotocol routers) by encapsulation
- In Figure 2.24, packets are encapsulated and transmitted through X.25 network in a serial mode
- In the mobile environment, Joe and his home agent in NY communicate Joe's Seattle location to the foreign agent. His communication with Sally in LA is tunneled

Half-Bridge

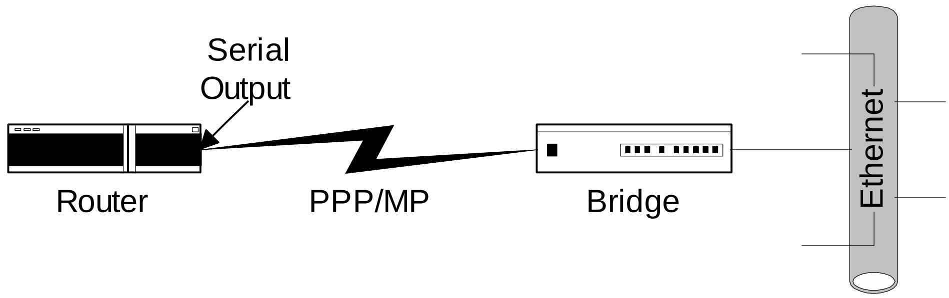


Figure 2.25 Half-Bridge Configuration

Notes

- Half-bridge (also referred to as half-router) is point-to-point communication
- Uses PPP protocol
- Helps low-end users to communicate with ISP on dial-up link saving the expense of dedicated link
- Router encapsulates packets in PPP frames and puts serial outputs to the bridge, and vice-versa

Switched Networks

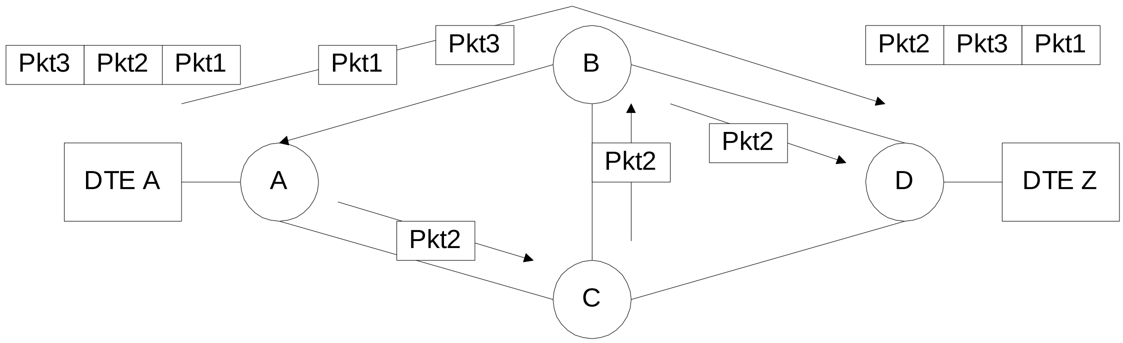


Figure 2.26(a) Datagram Configuration

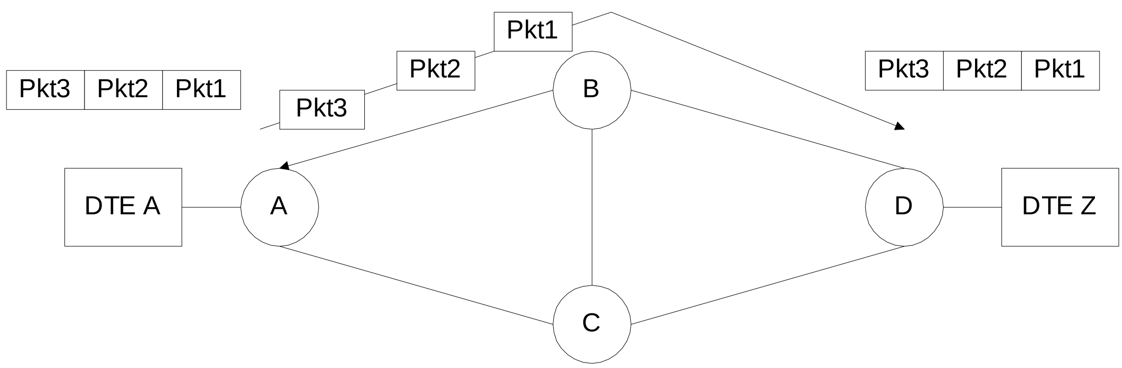
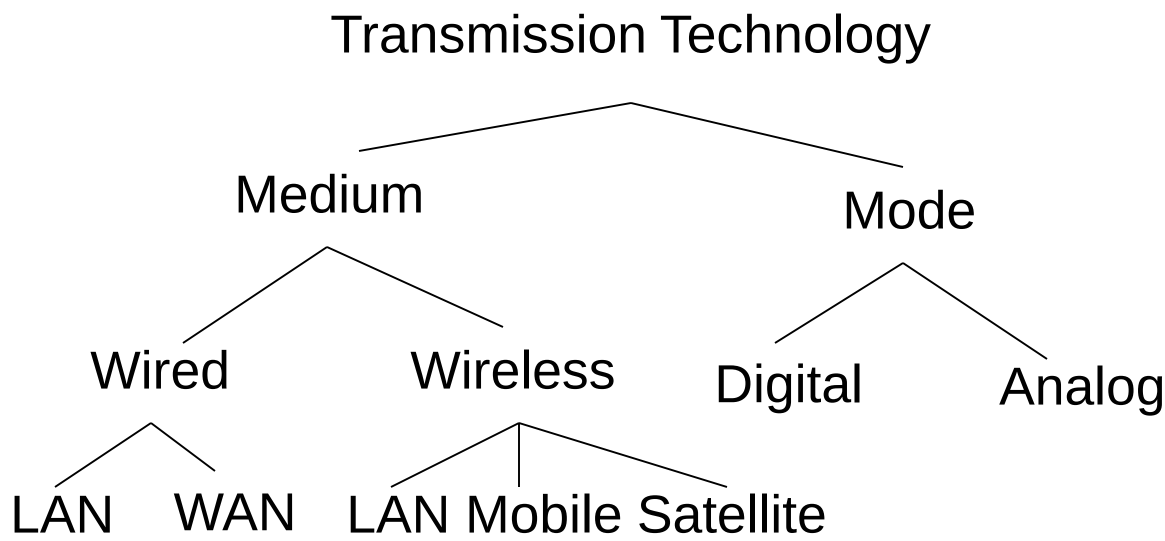


Figure 2.26(b) Virtual Circuit Configuration

Notes

- Switches are embedded in bridges and routers
- Switched network used in WAN
- Two types of switched networks
  - Circuit-switched
  - Packet-switched
    - Datagram service
    - Virtual circuit

Transmission Technology



Notes

- Physical transport media
  - UTP
  - Coax
  - Fiber
  - Terrestrial wireless
  - Satellite transmission

Transmission Modes

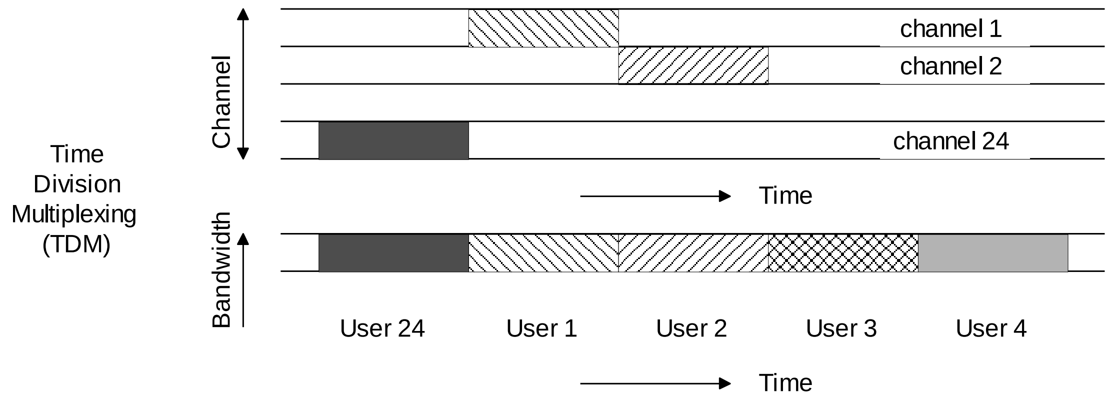


Figure 2.27(a) T1 Time Division Multiplexing (TDM) Transmission

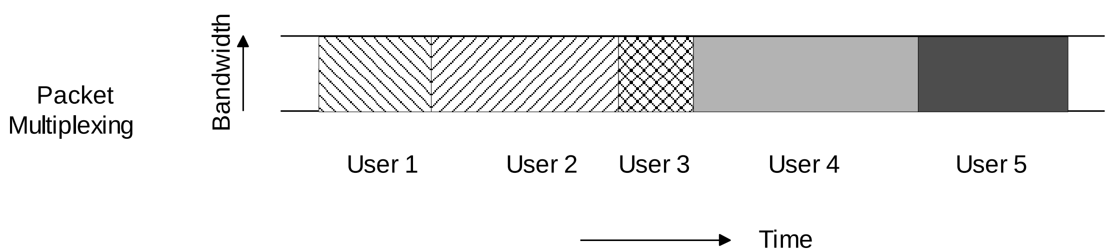


Figure 2.27(b) Packet Transmission (X.25)

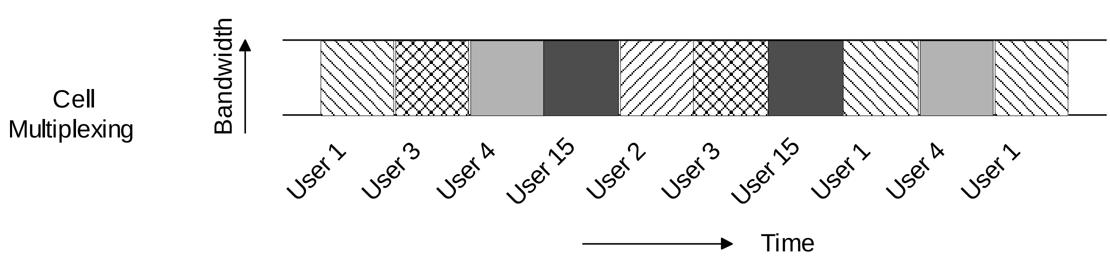


Figure 2.27(c) Cell Transmission (ATM)

Notes

Broadband Services

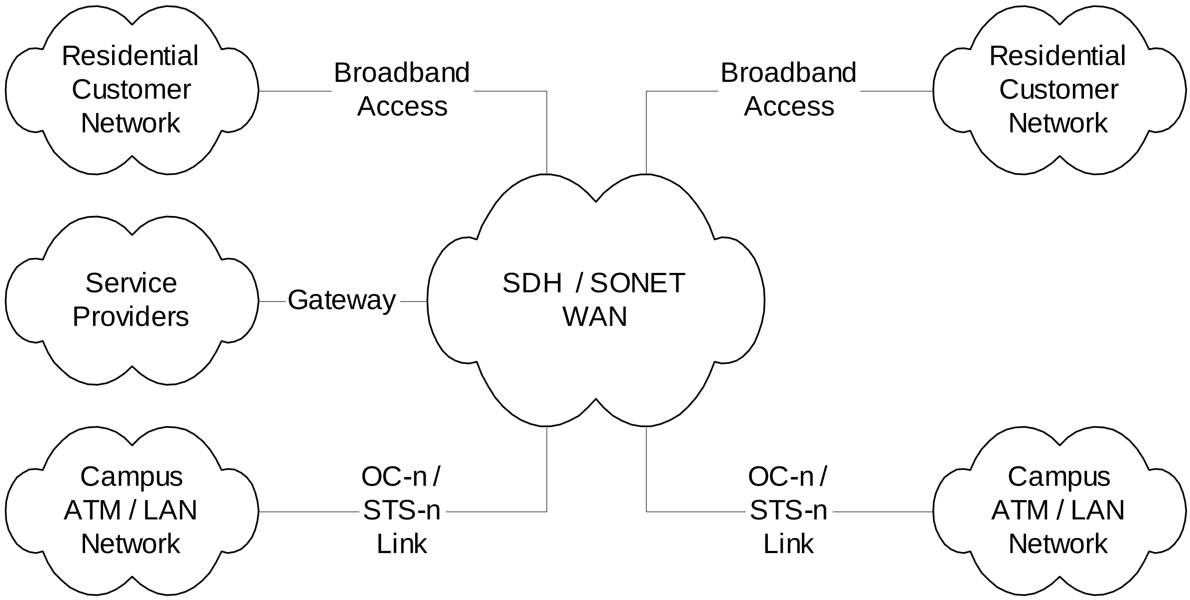


Figure 2.28 Broadband Services Network

Notes

- Integrated services: Voice, video, and data
- Narrow band ISDN (Integrated Services Digital Net.)
    - Basic rate:2B + D (B channel 64 kbps and D channel 16 kbps
    - Primary rate: 23B + D channels
- Broadband (ISDN) Services uses ATM technology
    - SONET (Synchronous Optical Network) or SDH (Synchronous Digital Hierarchy)
    - Data rate OC-n
        - OC-1 51.84 Mbps
        - OC-3 155.52 Mbps
    - Access technologies:
        - HFC (Hybrid Fiber Coaxial) / Cable modem
        - ADSL (Asymmetric Digital Subscriber Line)

Chapter 3

Basic Foundations:

Standards, Models, and Language

Introduction

- Standards
  - Standards organizations
  - Protocol standards of transport layers
  - Protocol standards of management (application) layer
- Management Models
- Language

Notes

Table 3.1 Network Management Standards

Standard	Salient Points
OSI / CMIP	■ International standard (ISO / OSI) ■ Management of data communications network - LAN and WAN ■ Deals with all 7 layers ■ Most complete ■ Object oriented ■ Well structured and layered ■ Consumes large resource in implementation
SNMP / Internet	■ Industry standard (IETF) ■ Originally intended for management of Internet components, currently adopted for WAN and telecommunication systems ■ Easy to implement ■ Most widely implemented
TMN	■ International standard (ITU-T) ■ Management of telecommunications network ■ Based on OSI network management framework ■ Addresses both network and administrative aspects of management
IEEE	■ IEEE standards adopted internationally ■ Addresses LAN and MAN management ■ Adopts OSI standards significantly ■ Deals with first two layers of OSI RM
Web-based Management	■ Web-Based Enterprise Management (WBEM) ■ Java Management Application Program Interface (JMAPI)

OSI Architecture and Model

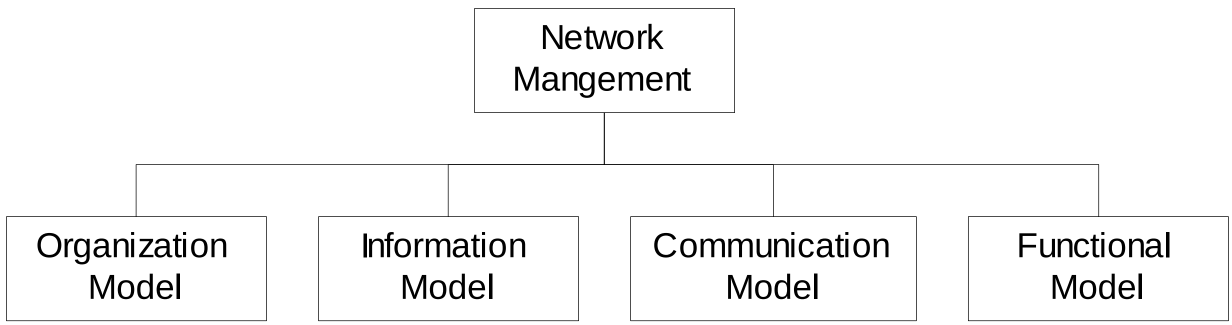


Figure 3.1 OSI Network Management Model

Notes

- Organization
    - Network management components
    - Functions of components
    - Relationships
- Information
    - Structure of management information (SMI)
        - Syntax and semantics
    - Management information base (MIB)
        - Organization of management information
    - Object-oriented

OSI Architecture and Model

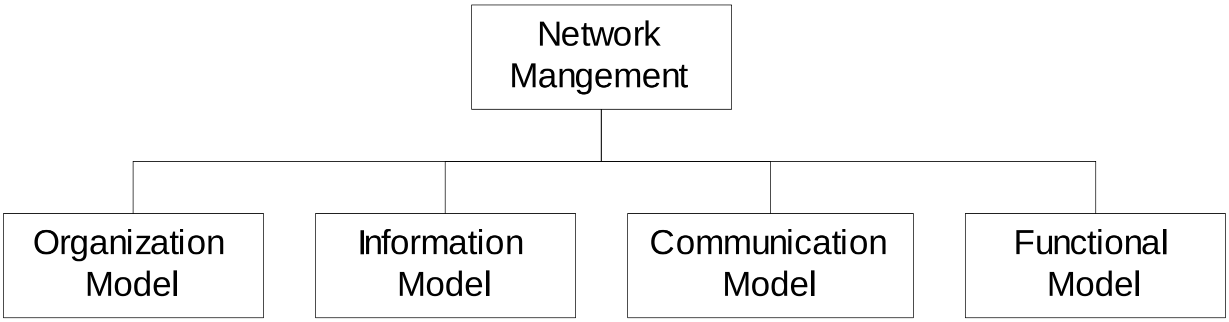


Figure 3.1 OSI Network Management Model

Notes

- Communication
    - Transfer syntax with bi-directional messages
    - Transfer structure (PDU)
- Functions
    - Application functions
        - Configure components
        - Monitor components
        - Measure performance
        - Secure information
        - Usage accounting

SNMP Architecture and Model

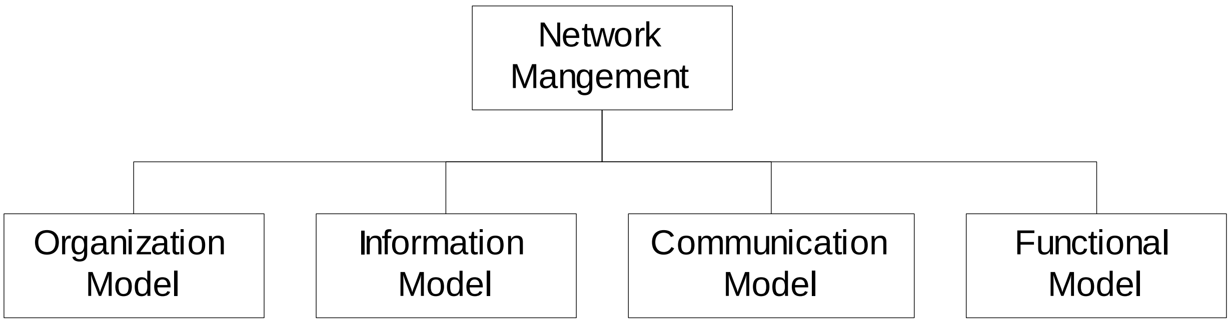


Figure 3.1 OSI Network Management Model

Notes

- Organization
    - Same as OSI model
- Information
    - Same as OSI, but scalar
- Communication
    - Messages less complex than OSI and unidirectional
    - Transfer structure (PDU)
- Functions
    - Application functions
        - Operations
        - Administration
        - Security

TMN Architecture

- Addresses management of telecommunication networks
- Based on OSI model
- Superstructure on OSI network
- Addresses network, service, and business management

Notes

Organizational Model

- Manager
  - Sends requests to agents
  - Monitors alarms
  - Houses applications
  - Provides user interface
- Agent
  - Gathers information from objects
  - Configures parameters of objects
  - Responds to managers' requests
  - Generates alarms and sends them to mangers
- Managed object
  - Network element that is managed
  - Houses management agent
  - All objects are not managed / manageable

Notes

Two-Tier Model

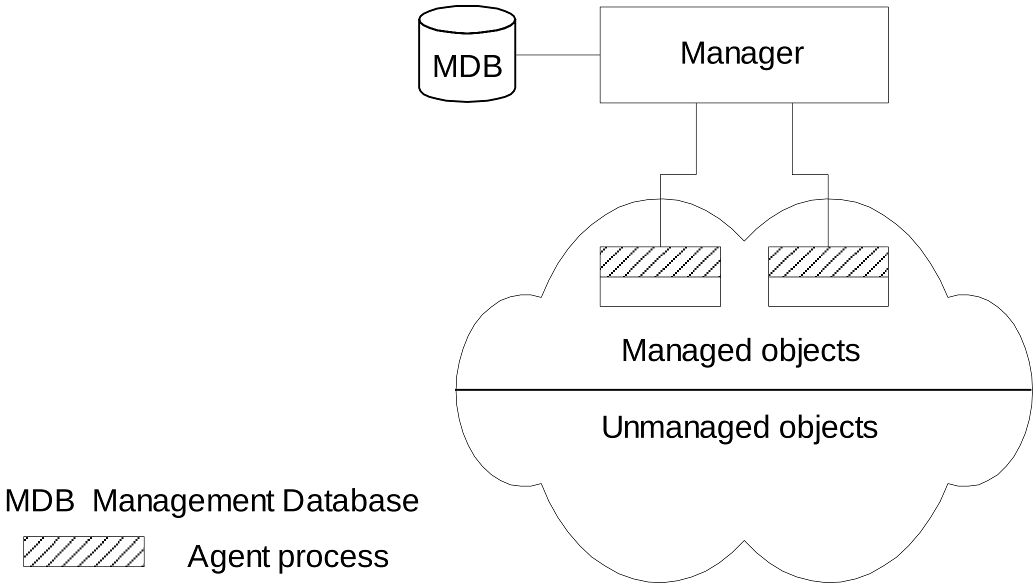


Figure 3.2 Two-Tier Network Mangement Organization Model

Notes

- Agent built into network element
  Example: Managed hub, managed router
- An agent can manage multiple elements
  Example: Switched hub, ATM switch
- MDB is a physical database
- Unmanaged objects are network elements that are not managed - both physical (unmanaged hub) and logical (passive elements)

Three-Tier Model

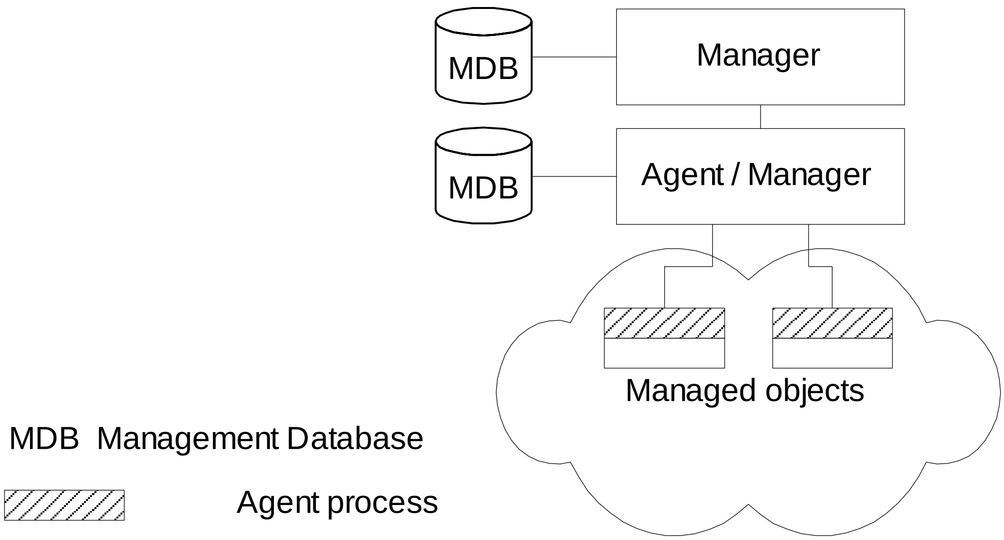


Figure 3.3 Three-Tier Network Mangement Organization Model

Notes

- Middle layer plays the dual role
    - Agent to the top-level manager
    - Manager to the managed objects
- Example of middle level: Remote monitoring agent (RMON)

Manager of Managers

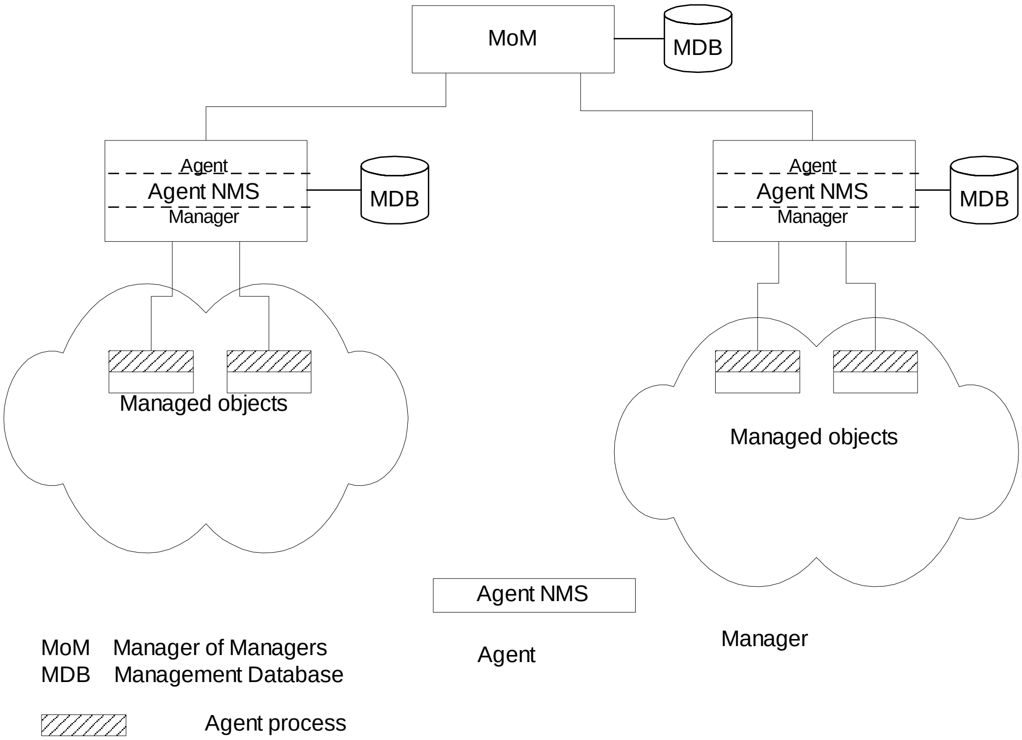


Figure 3.4 Network Mangement Organization Model with MoM

Notes

- Agent NMS manages the domain
- MoM presents integrated view of domains
- Domain may be geographical, administrative, vendor-specific products, etc.

Peer NMSs

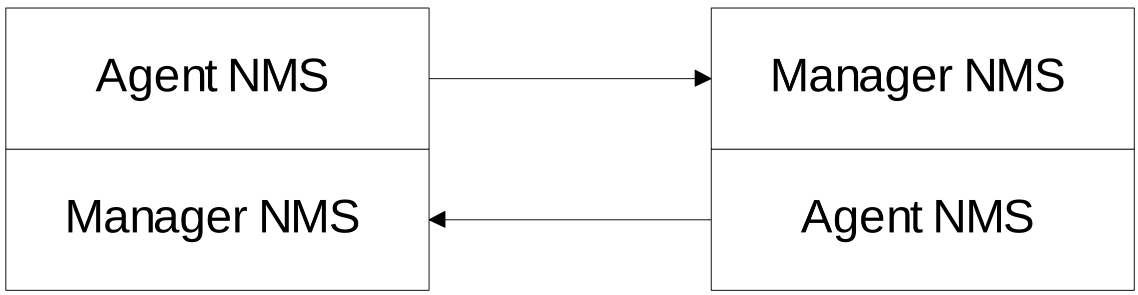


Figure 3.5 Dual Role of Management Process

Notes

- Dual role of both NMSs
- Network management system acts as peers
- Dumbbell architecture discussed in Chapter 1
- Notice that the manager and agent functions are processes and not systems

Information Model: Analogy

- Figure in a book uniquely identified by
    - ISBN, Chapter, and Figure number in that hierarchical order
- ID: {ISBN, chapter, figure}
- The three elements above define the syntax
- Semantics is the meaning of the three entities according to Webster's dictionary
- The information comprises syntax and semantics about an object

Notes

Structure of Management Information (SMI)

- SMI defines for a managed object
  - Syntax
  - Semantics
  - plus additional information such as status
- Example

sysDescr: { system 1 }
    Syntax: OCTET STRING
    Definition: "A textual description of the entity. "
    Access: read-only
    Status: mandatory

Notes

Management Information Base (MIB)

- Information base contains information about objects
- Organized by grouping of related objects
- Defines relationship between objects
- It is NOT a physical database. It is a *virtual* database that is compiled into management module

Notes

Information Base View: An Analogy

- Fulton County library system has many branches
- Each branch has a set of books
- The books in each branch is a different set
- The information base of the county has the view (catalog) of all books
- The information base of each branch has the catalog of books that belong to that branch. That is, each branch has its view (catalog) of the information base
- Let us apply this to MIB view

Notes

MIB View and Access of an Object

- A managed object has many attributes - its information base
- There are several operations that can be performed on the objects
- A user (manager) can view and perform only certain operations on the object by invoking the management agent
- The view of the object attributes that the agent perceives is the MIB view
- The operation that a user can perform is the MIB access

Notes

Management Data Base / Information Base

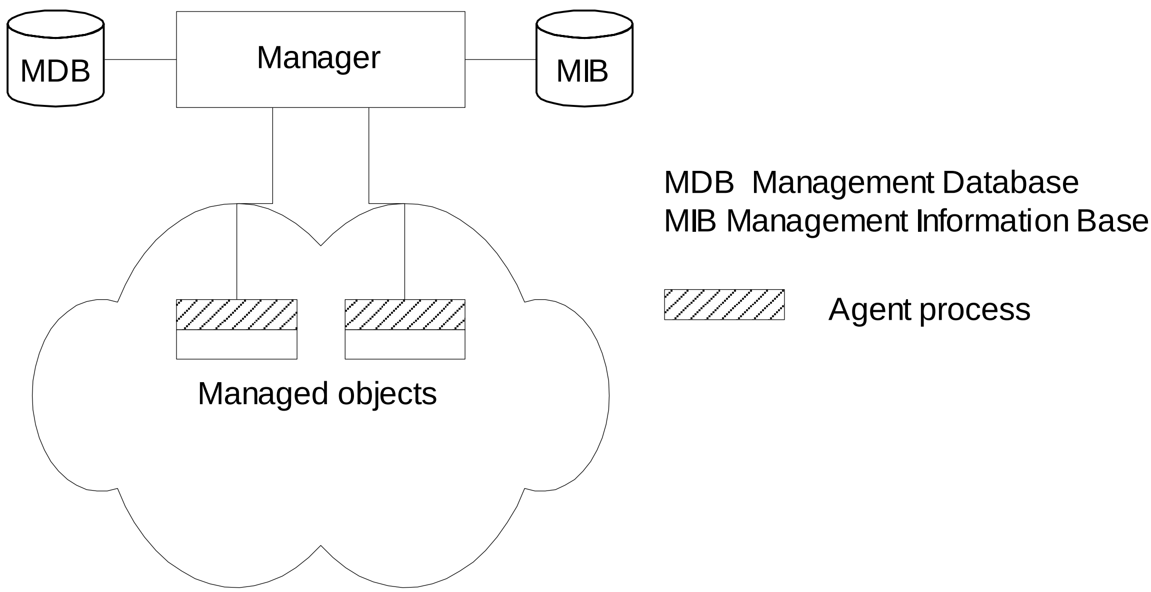


Figure 3.6 Network Configuration with Data and Information Base

Notes

- Distinction between MDB and MIB
    - MDB physical database; e.g.. Oracle, Sybase
    - MIB virtual database; schema compiled into management software
- An NMS can automatically discover a managed object, such as a hub, when added to the network
- The NMS can identify the new object as hub only after the MIB schema of the hub is compiled into NMS software

Managed Object

- Managed objects can be
  - Network elements (hardware, system)
    - hubs, bridges, routers, transmission facilities
  - Software (non-physical)
    - programs, algorithms
  - Administrative information
    - contact person, name of group of objects (IP group)

Notes

Management Information Tree

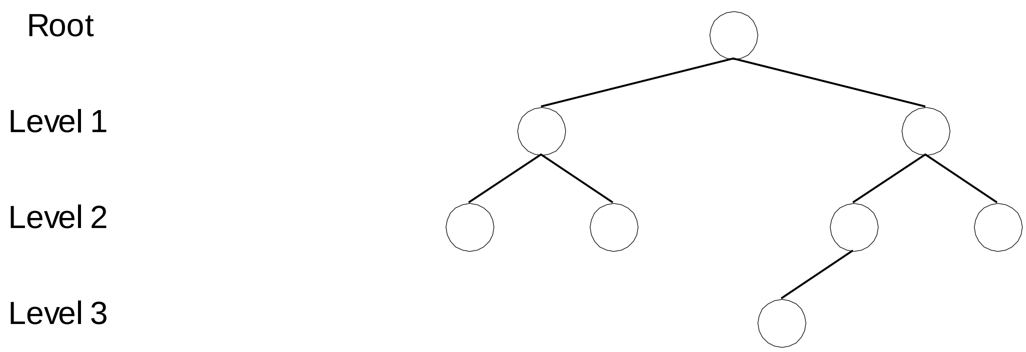


Figure 3.7 Generic Representation of Management Information Tree

Notes

OSI Management Information Tree

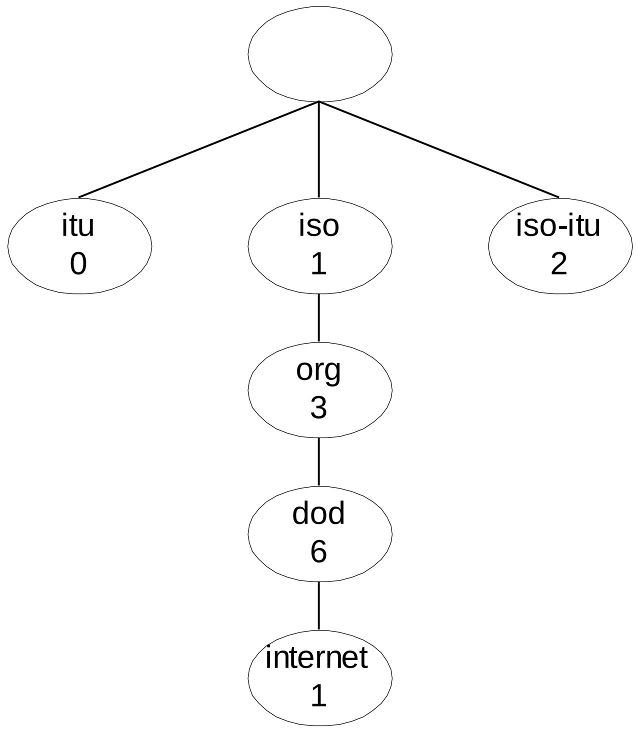


Figure 3.8 **OSI Management Information Tree**

Notes

- iso International Standards Organization
  itu International Telecommunications Union
  dod Department of Defense
- Designation:
  - iso 1
  - org 1.3
  - dod 1.3.6
  - internet 1.3.6.1

Object Type and Instance

- Type
  - Name
  - Syntax
  - Definition
  - Status
  - Access
- Instance

Notes

- Example of a circle
  - “circle” is syntax
  - Semantics is definition from dictionary”
    “A plane figure bounded by a single curved line, every point of which is of equal distance from the center of the figure.”
- Analogy of nursery school

Managed Object: Internet Perspective

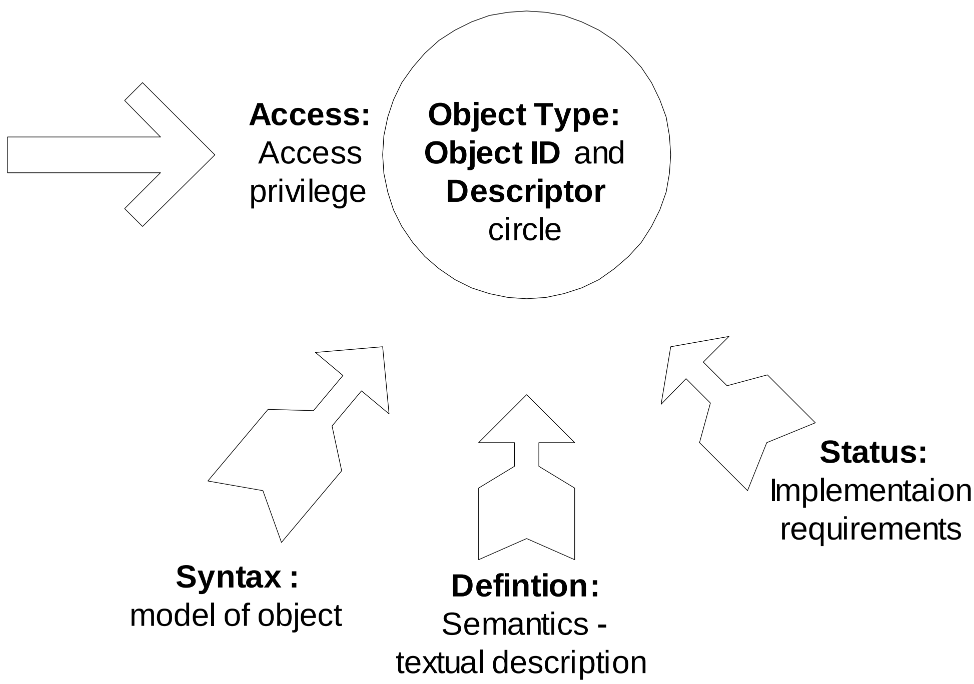


Figure 3.9(a) Internet Perspective

Notes

•*object ID*	unique ID
•and *descriptor*	and name for the object
•*syntax*	used to model the object
•*access*	access privilege to a managed object
•*status*	implementation requirements
•*definition*	textual description of the semantics of object type

Managed Object: OSI Perspective

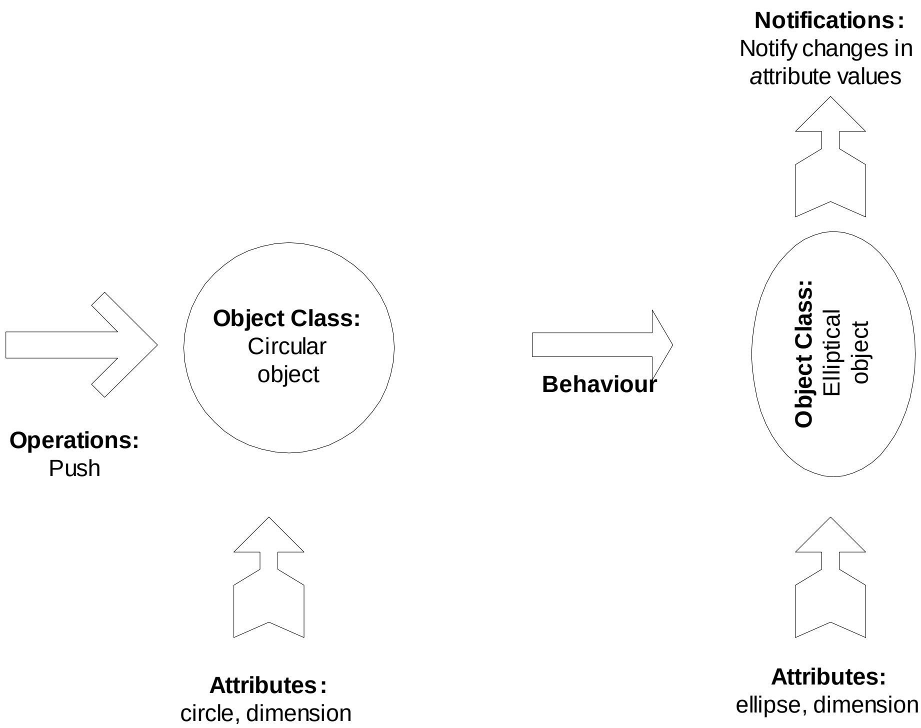


Figure 3.9(b) OSI Perspective

Notes

- *object class* managed object
- *attributes* attributes visible at its boundary
- *operations* operations which may be applied to it
- *behaviour* behaviour exhibited by it in response to operation
- *notifications* notifications emitted by the object

Packet Counter Example

Characteristics	Example
Object type	PktCounter
Syntax	Counter
Access	Read-only
Status	Mandatory
Description	Counts number of packets

Figure 3.10(a) Internet Perspective

Characteristics	Example
Object class	Packet Counter
Attributes	Single-valued
Operations	get, set
Behavior	Retrieves or resets values
Notifications	Generates notifications on new value

Figure 3.10 (b) OSI Perspective

Figure 3.10 Packet Counter As Example of Managed Object

Notes

Internet Vs OSI Managed Object

- Scalar object in Internet Vs Object-oriented approach in OSI
- OSI characteristics of operations, behaviour, and notification are part of communication model in Internet: get/set and response/alarm
- Internet syntax is absorbed as part of OSI attributes
- Internet access is part of OSI security model
- Internet status is part of OSI conformance application
- OSI permits creation and deletion of objects; Internet does not: Enhancement in SNMPv2

Notes

Mgmt. Communication Model

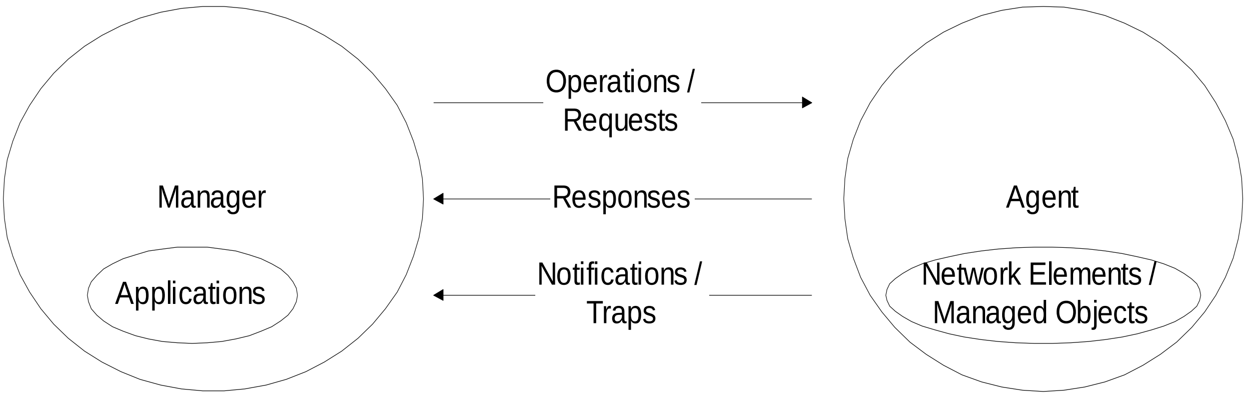


Figure 3.11 Management Message Communication Model

Notes

- In Internet requests/responses, in OSI operations
- In Internet traps and notifications (SNMPv2), in OSI notifications

Transfer Protocols

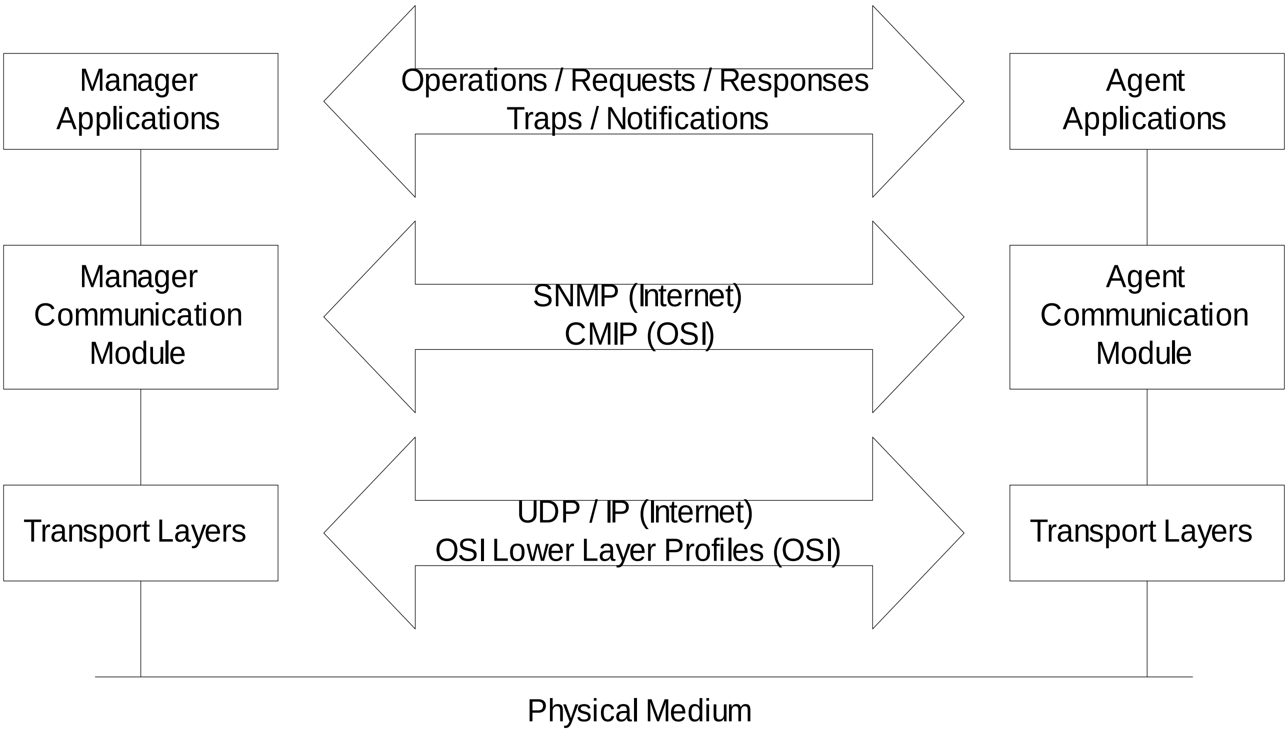


Figure 3.12 Management Communication Transfer Protocols

Notes

- Internet is based on SNMP; OSI is based on CMIP
- OSI uses CMISE (Common Management Information Service Element) application with CMIP
- OSI specifies both c-o and connectionless transport protocol; SNMPv2 extended to c-o, but rarely used

Abstract Syntax Notation One

- ASN.1 is more than a syntax; it's a language
- Addresses both syntax and semantics
- Two type of syntax
    - Abstract syntax: set of rules that specify data type and structure for information storage
    - Transfer syntax: set of rules for communicating information between systems
- Makes application layer protocols independent of lower layer protocols
- Can generate machine-readable code: Basic Encoding Rules (BER) is used in management modules

Notes

Backus-Nauer Form (BNF)

Definition:

<name> ::= <definition>

Rules:

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<number> ::= <number> | <digit> <number>

<op> ::= +|-|x|/

<SAE> ::= <number>|<SAE>|<SAE><op><SAE>

Example:

- 9 is *primitive* 9
- 19 is *construct* of 1 and 9
- 619 is *construct* of 6 and 19

Notes

- BNF is used for ASN.1 constructs
- Constructs developed from primitives
- The above example illustrates how numbers are constructed from the primitive <digit>
- Simple Arithmetic Expression entity (<SAE>) is constructed from the primitives <digit> and <op>

Simple Arithmetic Expression

<SAE> ::= <number> | <SAE><op><number>

Example: 26 = 13 x 2

Constructs and primitives

Notes

Type and Value

- Assignments
  - <BooleanType> ::= BOOLEAN
  - <BooleanValue> ::= TRUE | FALSE

- ASN.1 module is a group of assignments

```
person-name    Person-Name::=
          {
          first      "John",
          middle   "I",
          last       "Smith"
          }
```

Notes

Data Type: Example 1

```
PersonnelRecord ::= SET
            {          Name,
             title      GraphicString,
             division    CHOICE
                marketing       [0]   SEQUENCE
                  {Sector,
                   Country},
            research       [1]   CHOICE
                  {product-based    [0]   NULL,
                   basic            [1]   NULL},
            production          [2]   SEQUENCE
                  {Product-line,
                   Country }               }
etc.
```

Figure 3.13 ASN.1 Data Type Definition Example 1

Notes

- Module name starts with capital letters
- Data types:
    - Primitives: NULL, GraphicString
    - Constructs
        - Alternatives : CHOICE
        - List maker: SET, SEQUENCE
        - Repetition: SET OF, SEQUENCE OF:
- Difference between SET and SEQUENCE

Data Type: Example 2

```
Trade-message ::= SEQUENCE
      {invoice-no          INTEGER
       name                GraphicString,
       details        SEQUENCE OF
                           SEQUENCE
                  {part-no        INTEGER
                   quantity       INTEGER},
       charge          REAL,
       authenticator       Security-Type}

Security-Type ::= SET
      {     …
            …
            …   }
```

Figure 3.14 ASN.1 Data Type Definition Example 2

Notes

- SEQUENCE OF SEQUENCE makes tables of rows

ASN.1 Symbols

Symbol	Meaning
::=	Defined as
|	or, alternative, options of a list
-	Signed number
--	Following the symbol are comments
{}	Start and end of a list
[]	Start and end of a tag
()	Start and end of subtype
..	Range

Notes

Keyword Examples

- CHOICE
- SET
- SEQUENCE
- OF
- NULL

Notes

- Keywords are in all UPPERCASE letters

ASN.1 Data Type Conventions

Data Types	Convention	Example
Object name	Initial lowercase letter	sysDescr, etherStatsPkts
Application data type	Initial uppercase letter	Counter, IpAddress
Module	Initial uppercase letter	PersonnelRecord
Macro, MIB module	All uppercase letters	RMON-MIB
Keywords	All uppercase letters	INTEGER, BEGIN

Notes

Data Type: Structure & Tag

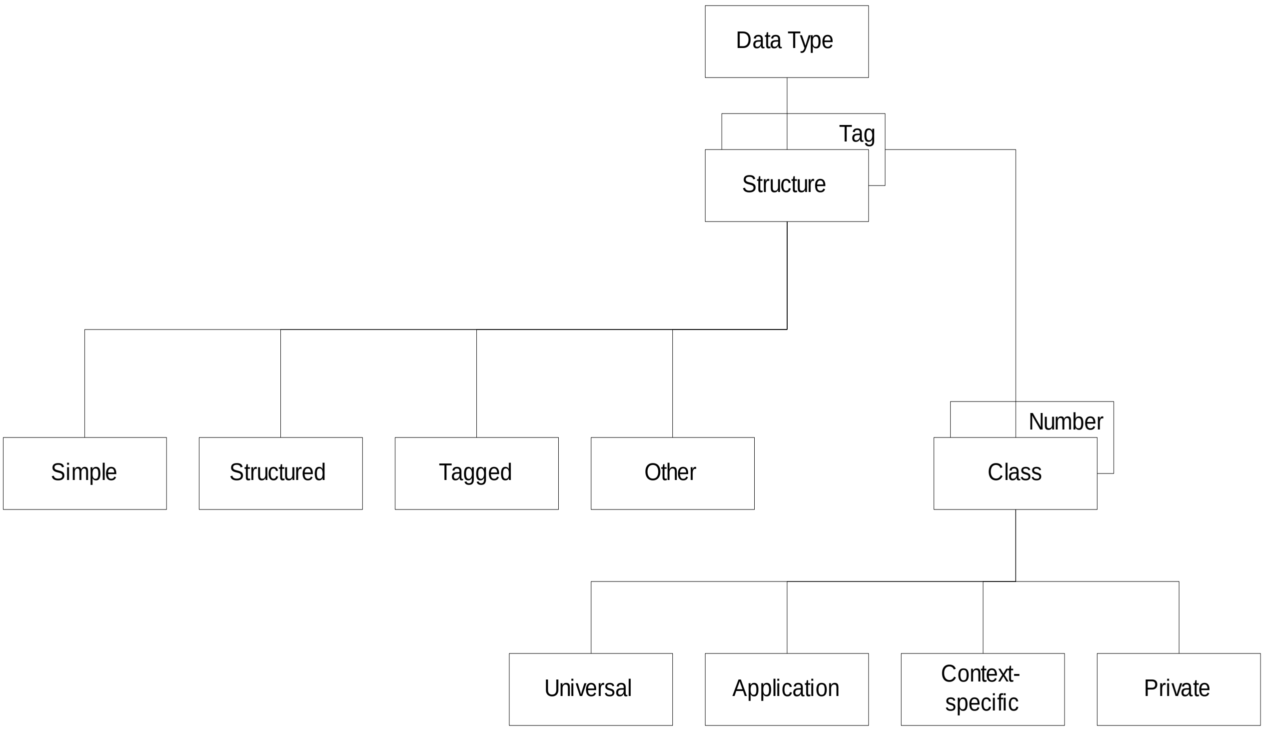


Figure 3.15 ASN.1 Data Type Structure and Tag

Notes

- Structure defines how data type is built
- Tag uniquely identifies the data type

Structure

- Simple
  - PageNumber ::= INTEGER
  - ChapterNumber ::= INTEGER
- Structure / Construct
  - BookPageNumber ::=
    SEQUENCE
    {ChapterNumber, Separator, PageNumber

  Example: {1-1, 2-3, 3-39}
- Tagged
  - Derived from another type; given a new ID
  - In Fig. 3-14, INTEGER is either universal or application specific
- Other types:
  - CHOICE, ANY

Notes

- BookPages ::= SEQUENCE OF { BookPageNumber}

or

BookPages ::=
SEQUENCE OF
{
SEQUENCE
{ChapterNumber, Separator, PageNumber}
}

Tag

- Tag uniquely identifies a data type
- Comprises *class* and *tag number*
- Class:
  - Universal - always true
  - Application - only in the application used
  - Context-specific - specific context in application
  - Private - used extensively by commercial vendors

Notes

Example:

BOOLEAN	Universal 1
INTEGER	Universal 2
research	Application [1] (Figure 3.13)
product-based	Context-specific under *research* [0]

Enumerated Integer

```
RainbowColors ::= ENUMERATED
    {
        violet      (0)
        indigo      (1)
        blue        (2)
        green       (3)
        yellow      (4)
        orange      (5)
        red         (6)
    }
```

Notes

- ENUMERATED is a special case of INTEGER
- Example: RainbowColors(5) is orange

ASN.1 Module Example

```
IpNetMediaEntry ::=SEQUENCE{
        ipNetToMediaIfIndex          INTEGER
        ipNetToMediaPhysAddress      PhysAddress
        ipNetToMediaNetAddress       IpAddress
        ipNetToMediaType             INTEGER}
```

Notes

```
Name:                John P Smith
Title:               Director
Employee Number      51
Date of Hire:        17 September 1971
Name of Spouse;      Mary T Smith
Number of Children   2
Child Information
        Name             Ralph T Smith
        Date of Birth    11 November 1957
Child Information
        Name             Susan B Jones
        Date of Birth    17 July 1959
```

(a) Informal description of personnel record

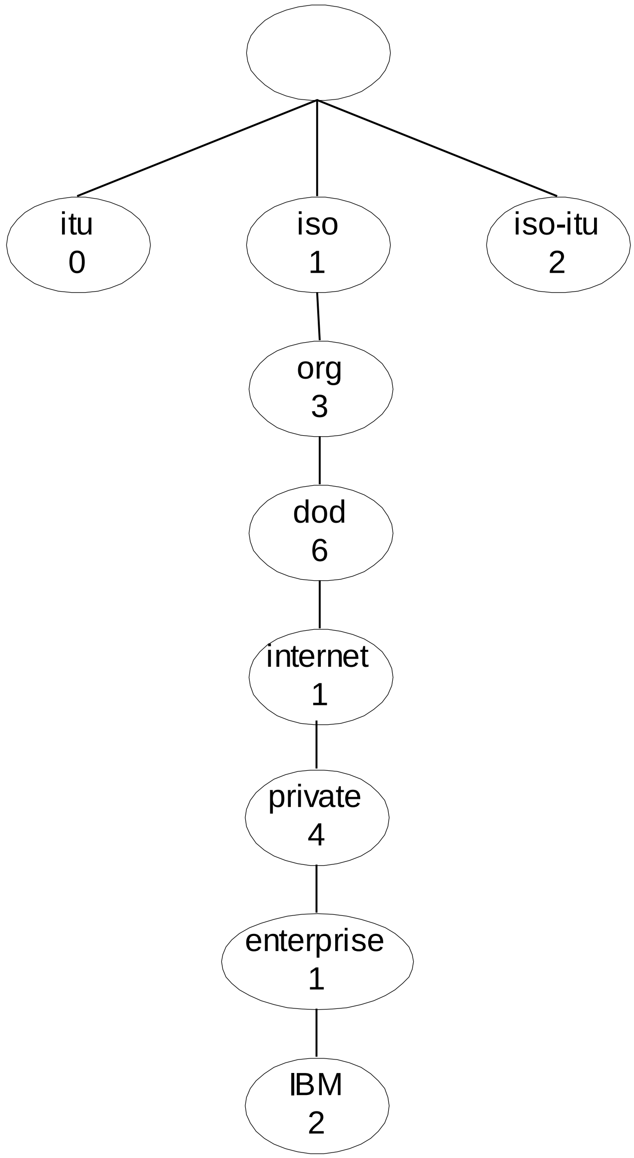
```
PersonnelRecord ::= [APPLICATION 0] IMPLICIT SET {
        Name,
        title [0] VisibleString,
        number EmployeeNumber,
        dateOfHire [1] Date,
        nameOfSpouse [2] Name,
        children [3] IMPLICIT SEQUENCE OF ChildInformation DEFAULT { } }
ChildInformation ::= SET {
        Name,
        dateOfBirth [0] Date }
Name ::= [APPLICATION 1] IMPLICIT SEQUENCE {
        givenName VisibleString,
        initial VisibleString,
        familyName VisibleString }

EmployeeNumber ::= [APPLICATION 2] IMPLICIT INTEGER

Date ::=  [APPLICATION 3] IMPLICIT VisibleString -- YYYYMMDD
```

(b) ASN.1 description of the record structure

```
{                          {givenName “John”, initial “T”, familyName “Smith”},
        title              “Director”
        number             “51”
        dateOfHire         “19710917”
        nameOfSpouse       {givenName “Mary”, initial “T”, familyName “Smith”},
        children
        { {                {givenName “Ralph”, initial “T”, familyName “Smith”},
           dateOfBirth             “19571111”},
          {                {givenName “Susan”, initial “B”, familyName “Jones”}
           dateOfBirth             “19590717”}}}
```

(c) ASN.1 description of a record value

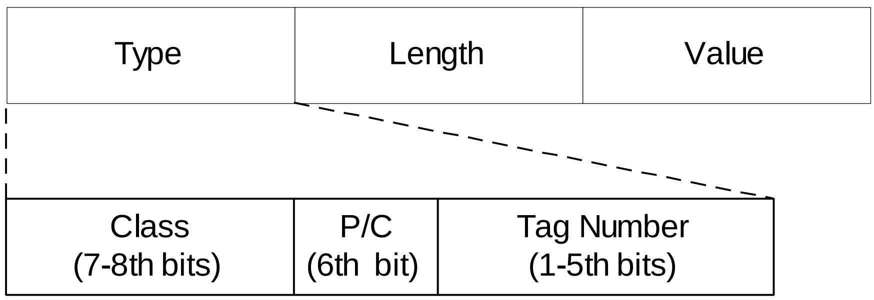
Object Name



Notes

- internet OBJECT IDENTIFIER ::=
  {ISO(1) ORG(3) DOD(6) INTERNET(1)}

TLV Encoding



Class	8th bit	7th bit
Universal	0	0
Application	0	1
Context-specific	1	0
Private	1	1

Notes

- TLV Type, length, and value are components of the structure

Macro

```
<macroname> MACRO ::=
BEGIN
    TYPE NOTATION ::=  <syntaxOfNewType>
    VALUE NOTATION ::= <syntaxOfNewValue>
    <auxiliaryAssignments>
END
```

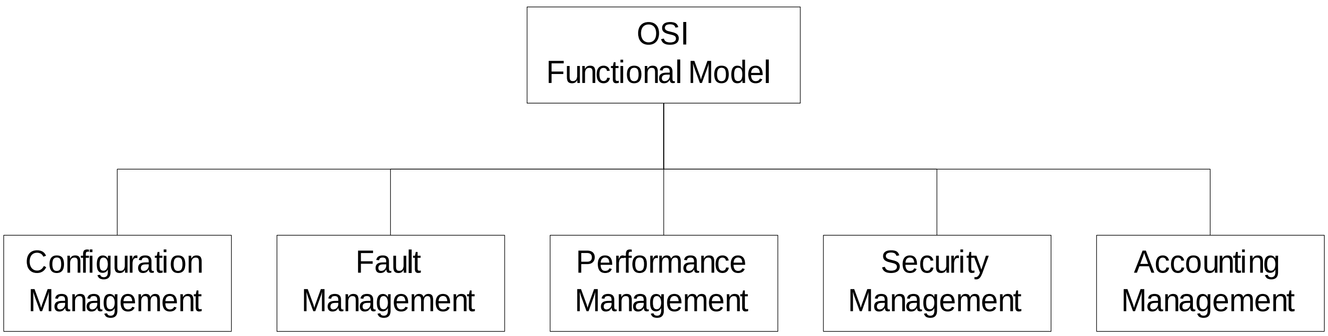
Example:

```
CS8803  OBJECT-IDENTITY
    STATUS       current
    DESCRIPTION    "A graduate-level network
    management course offered every fall by
    College of Computing in Georgia Institute of
    Technology."
        ::= {csclasses 50}
```

Notes

- Macro is used to create new data types

Functional Model



Notes

- Configuration management
  - set and change network configuration and component parameters
  - Set up alarm thresholds
- Fault management
  - Detection and isolation of failures in network
  - Trouble ticket administration
- Performance management
  - Monitor performance of network
- Security management
  - Authentication
  - Authorization
  - Encryption
- Accounting management
  - Functional accounting of network usage

Chapter 4

SNMPv1:

Organization and Information Models

Case Histories

- AT&T Network Management Centers
  - Network Control Centers
  - Network Operations Center
- CNN World Headquarters
- Centralized troubleshooting of NIC
- Performance degradation due to NMS
- Bell Operating company procedure

Notes

Managed LAN

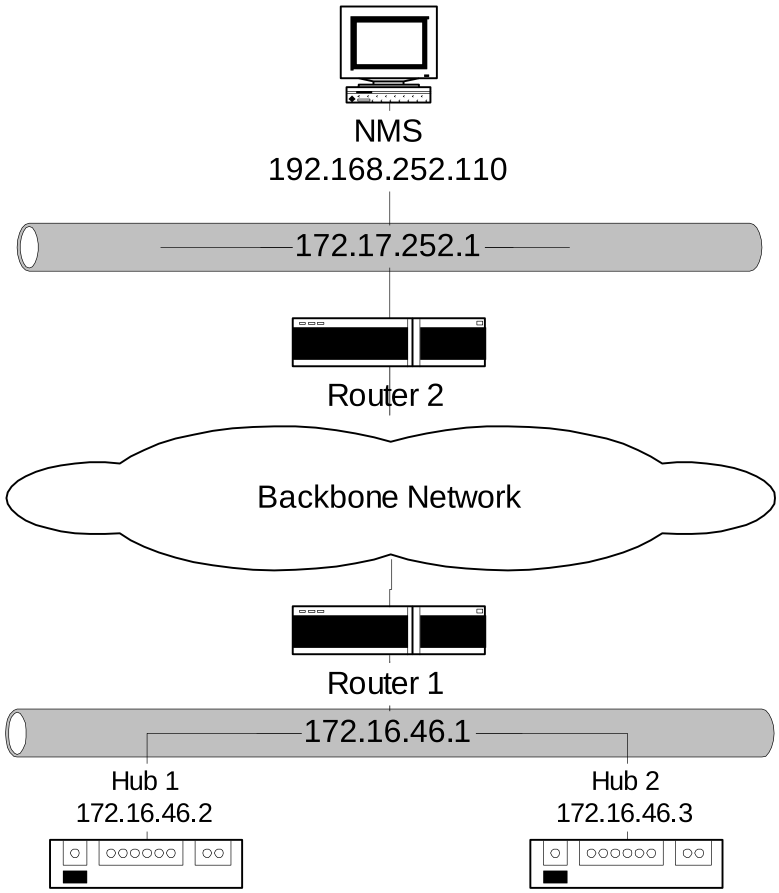


Figure 4.1 A Managed LAN Network

Notes

- NMS on subnet 192.168.252.1 manages the router and the hubs on subnet 172.16.46.1 across the backbone network

Managed Hub: System Information

Title: System Information: 172.16.46.2
Name or IP Address: 172.16.46.2

System Name :
System Description : 3Com LinkBuilder FMS, SW version:3.02
System Contact :
System Location :
System Object ID :
.iso.org.dod.internet.private.enterprises.43.1.8.5
System Up Time : (2475380437) 286 days, 12:03:24.37

Figure 4.2(a) System Information on 172.16.46.2 Hub

Notes

- Information obtained querying the hub
- Data truly reflects what is stored in the hub

Managed Router: System Information

Title: System Information: router1.gatech.edu
Name or IP Address: 172.16.252.1

System Name	:	router1.gatech.edu
System Description	:	Cisco Internetwork Operating System Software
	:	IOS (tm) 7000 Software (C7000-JS-M), Version
	:	11.2(6),RELEASE SOFTWARE (ge1)
	:	Copyright (c) 1986-1997 by Cisco Systems, Inc.
	:	Compiled Tue 06-May-97 19:11 by kuong
System Contact		
System Location	:	
System Object ID	:	iso.org.dod.internet.private.enterprises.cisco.ciscoProducts. cisco 7000
System Up Time	:	(315131795) 36 days, 11:21:57.95

Figure 4.2(c) System Information on Router

Notes

Managed Hub: Port Addresses

Index	Interface	IP address	Network Mask	Network Address	Link Address
1	3Com	172.16.46.2	255.255.255.0	172.16 46.0	0x08004E07C25C
2	3Com	192.168.101.1	255.255.255.0	192.168.101.0	<none>

Notes

- Information acquired by the NMS on hub interfaces
- Index refers to the interface on the hub
- Link address is the MAC address
- The second row data is a serial link

Managed Router: Port Addresses

Index	Interface	IP address	Network Mask	Network Address	Link Address
23	LEC.1.0	192.168.3.1	255.255.255.0	192.168.3.0	0x00000C3920B4
25	LEC.3.9	192.168.252.15	255.255.255.0	192.168.252.0	0x00000C3920B4
13	Ethernet2/0	172.16..46.1	255.255.255.0	172.16..46.0	0x00000C3920AC
16	Ethernet2/3	172.16.49.1	255.255.255.0	172.16.49.0	0x00000C3920AF
17	Ethernet2/4	172.16.52.1	255.255.255.0	172.16.52.0	0x00000C3920B0
9	Ethernet1/2	172.16.55.1	255.255.255.0	172.16.55.0	0x00000C3920A6
2	Ethernet 0/1	172.16.56.1	255.255.255.0	172.16.56.0	0x00000C39209D
15	Ethernet2/2	172.16.57.1	255.255.255.0	172.16.57.0	0x00000C3920AE
8	Ethernet1/1	172.16.58.1	255.255.255.0	172.16.58.0	0x00000C3920A5
14	Ethernet2/1	172.16.60.1	255.255.255.0	172.16.60.0	0x00000C3920AD

Notes

- Information acquired by NMS on the router interfaces
- Index refers to the interface on the router
- LEC is the LAN emulation card
- Ethernet 2/0 interface refers to the interface card 2 and port 0 in that card

Internet SNMP Management

- 1970 Advanced Research Project Agency Network (ARPANET)
  Internet control Message Protocol (ICMP)
- Internet Engineering Task Force (IETF)
    - 1990 SNMPv1
    - 1995 SNMPv2
    - 1998 SNMPv3
- Internet documents:
    - Request for Comments (RFC)
    - IETF STD Internet Standard
    - FYI For your information
- Source for RFCs
    - ftp://nic.mil/rfc
    - ftp://ftp.internic.net/rfc
    - http://nic/internet.net/

Notes

SNMPv1 & SNMPv2 Documents

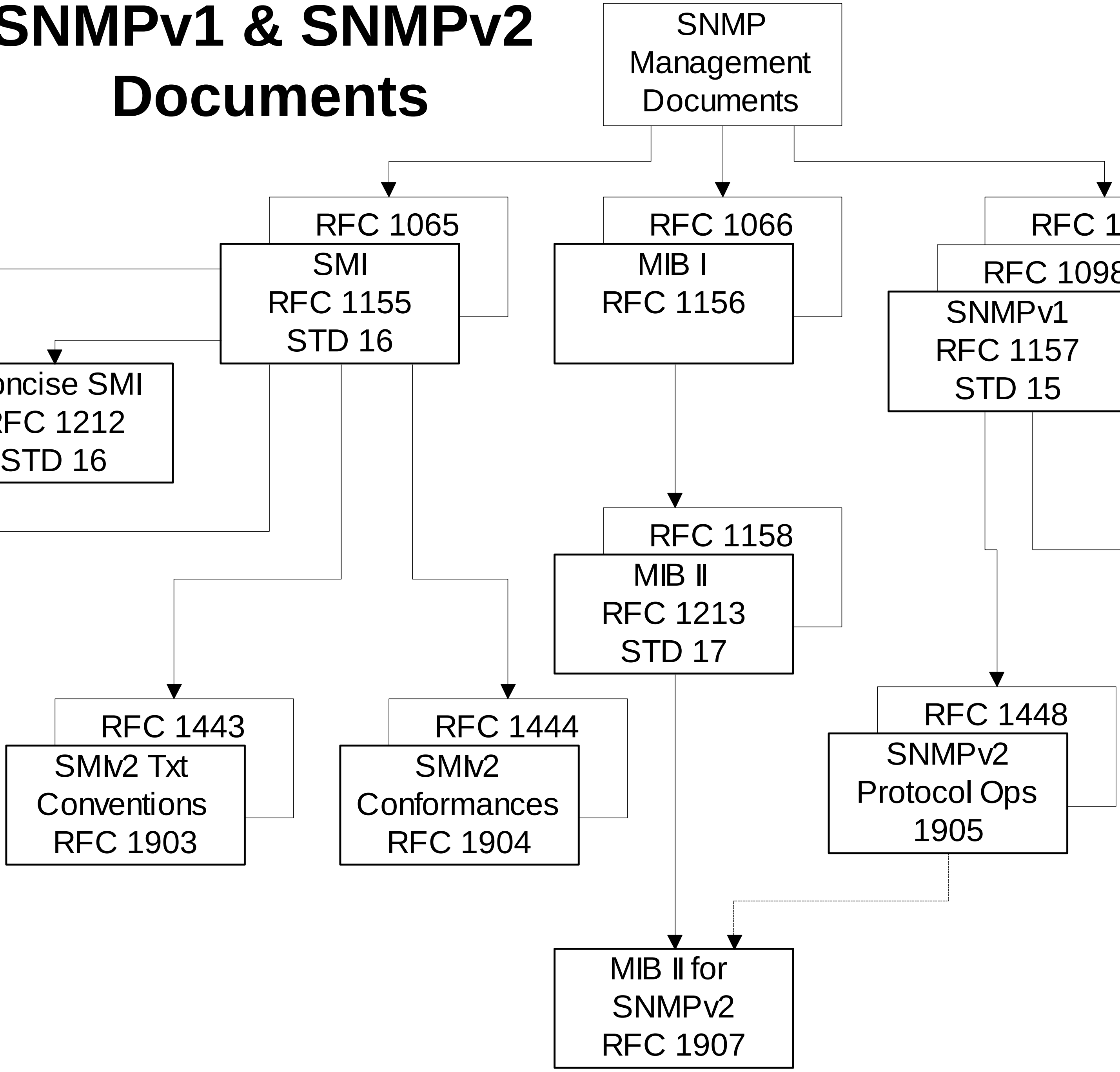


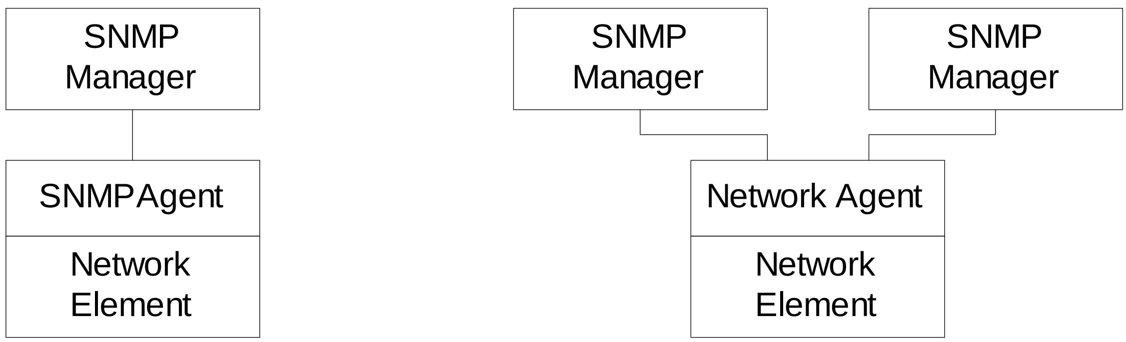
Figure 4.4 SNMP Document Evolution

SNMP Model

- Organization Model
  - Relationship between network element, agent, and manager
  - Hierarchical architecture
- Information Model
  - Uses ASN.1 syntax
  - SMI (Structure of Management Information
  - MIB (Management Information Base)
- Communication Model
  - Transfer syntax
  - SNMP over TCP/IP
  - Communication services addressed by messages
  - Security framework community-based model

Notes

Two-Tier Organization Model

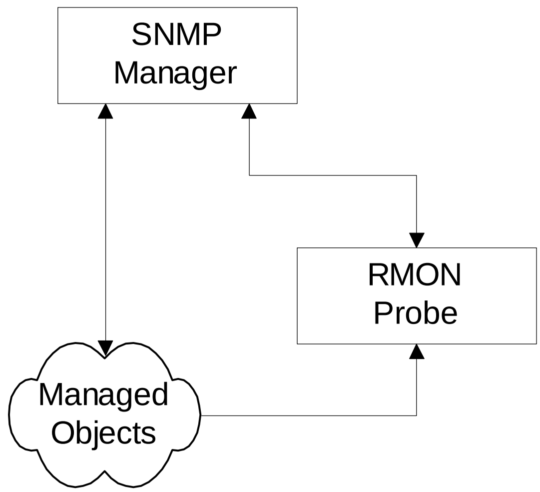


(a) One Manager - One Agent Model **(b) Multiple Managers - One Agent Model**

Notes

- Any host that could query an agent is a manager

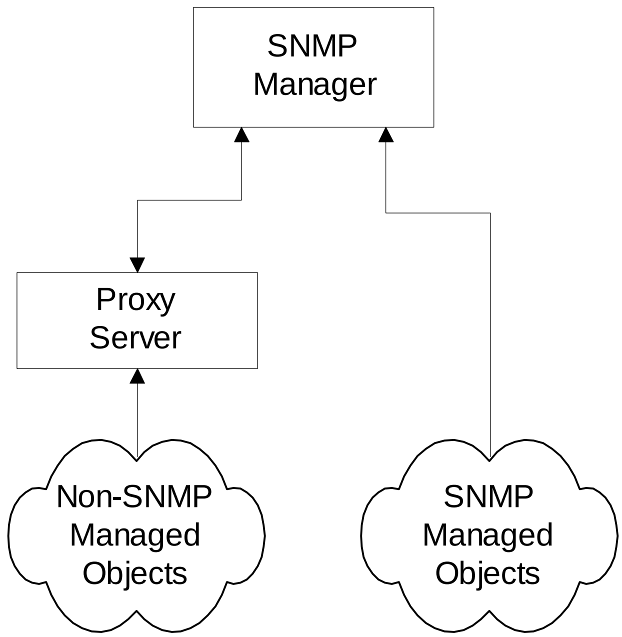
Three-Tier Organization Model: RMON



Notes

- Managed object comprises network element and management agent
- RMON acts as an agent and a manager
- RMON (Remote Monitoring) gathers data from MO, analyses the data, and stores the data
- Communicates the statistics to the manager

Three-Tier Organization Model: Proxy Server



Notes

- Proxy server converts non-SNMP data from non-SNMP objects to SNMP compatible objects and messages

System Architecture

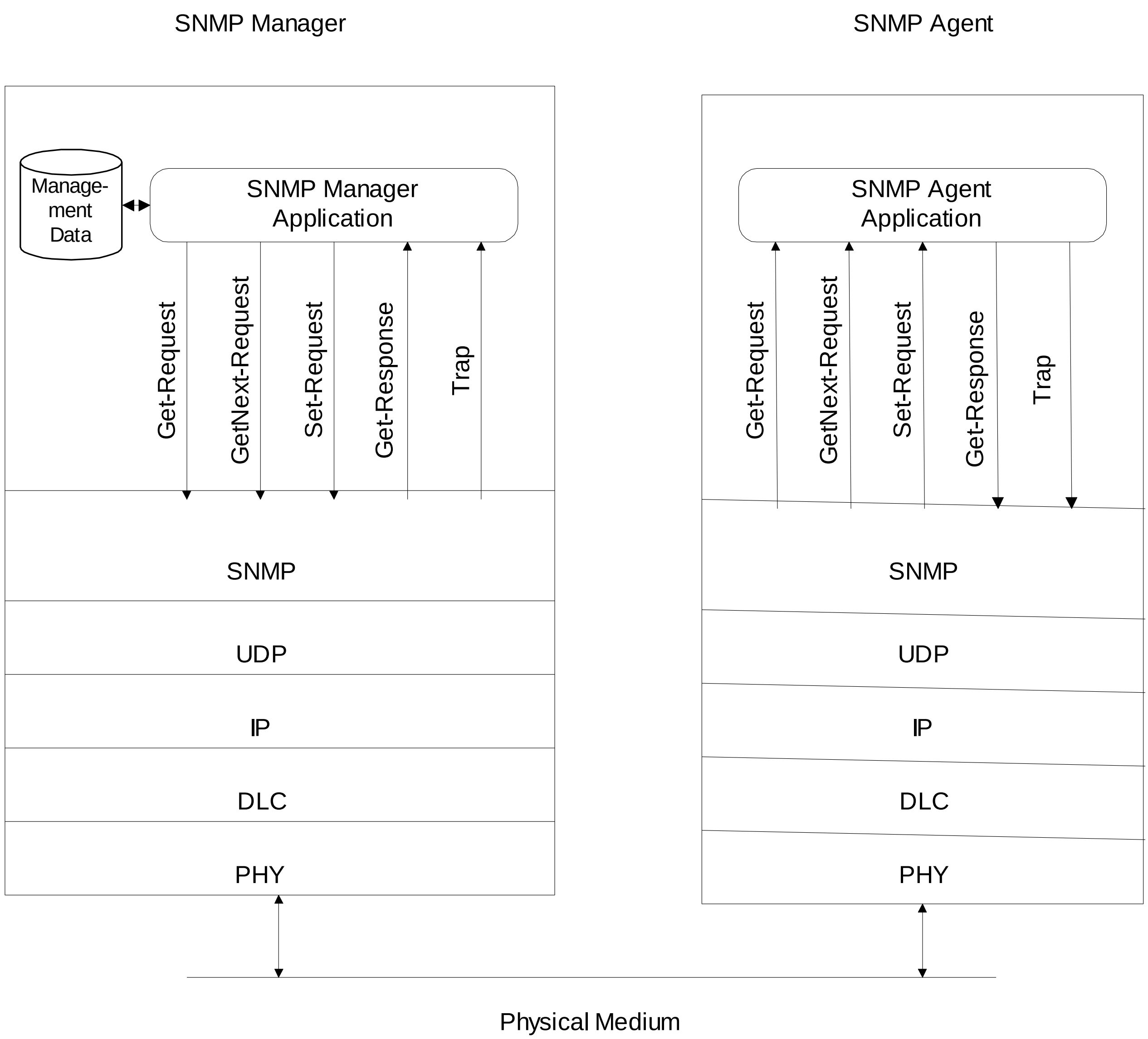


Figure 4.9 SNMP Network Management Architecture

Notes

- Messages between manager and agent
- Direction of messages - 3 from manager and 2 from agent

SNMP Messages

- Get-Request
  - Sent by manager requesting data from agent
- Get-Next-Request
  - Sent by manager requesting data on the next MO to the one specified
- Set-Request
  - Initializes or changes the value of network element
- Get-Response
  - Agent responds with data for get and set requests from the manager
- Trap
  - Alarm generated by an agent

Notes

Information

- Structure of Management Information (SMI) (RFC 1155)
- Managed Object
  - Scalar
  - Aggregate or tabular object
- Management Information Base (RFC 1213)

Notes

- RFCs can be downloaded from ftp.internic.net/rfc

Managed Object

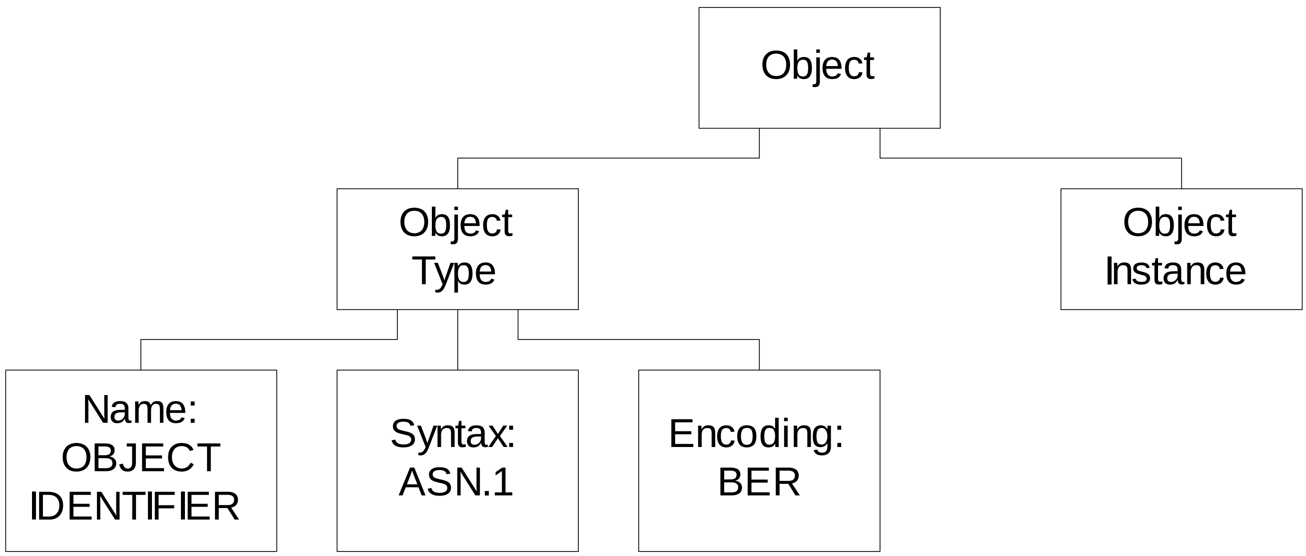


Figure 4.10 Managed Object : Type and Instance

Notes

- Object type and data type are synonymous
- Object identifier is data type, not instance
- Object instance IP address (See Figure 4.2)

Managed Object: Multiple Instances

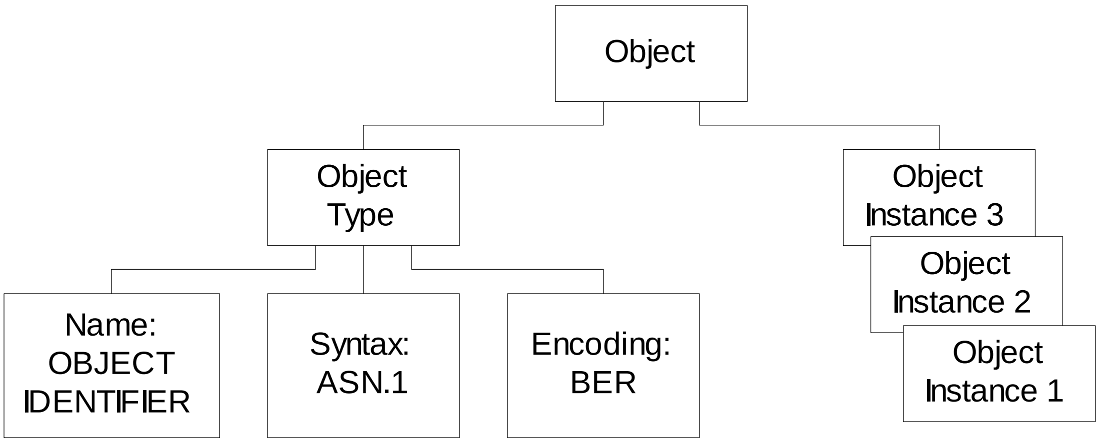


Figure 4.11 Managed Object : Type with Multiple Instances

Notes

- All 3 Com hubs of the same version have identical identifier; they are distinguished by the IP address
- Each IP address is an instance of the object

Name

Uniquely defined by

- DESCRIPTOR AND
- OBJECT IDENTIFIER

internet OBJECT IDENTIFIER ::=
{iso org(3) dod(6) 1 }.

internet OBJECT IDENTIFIER ::= {iso(1) standard(3) dod(6) internet(1)}
internet OBJECT IDENTIFIER ::= {1 3 6 1}
internet OBJECT IDENTIFIER ::= {iso standard dod internet }
internet OBJECT IDENTIFIER ::= { iso standard dod(6) internet(1) }
internet OBJECT IDENTIFIER ::= { iso(1) standard(3) 6 1 }

Notes

Example

ipAddrTable ip 20

Internet Subnodes

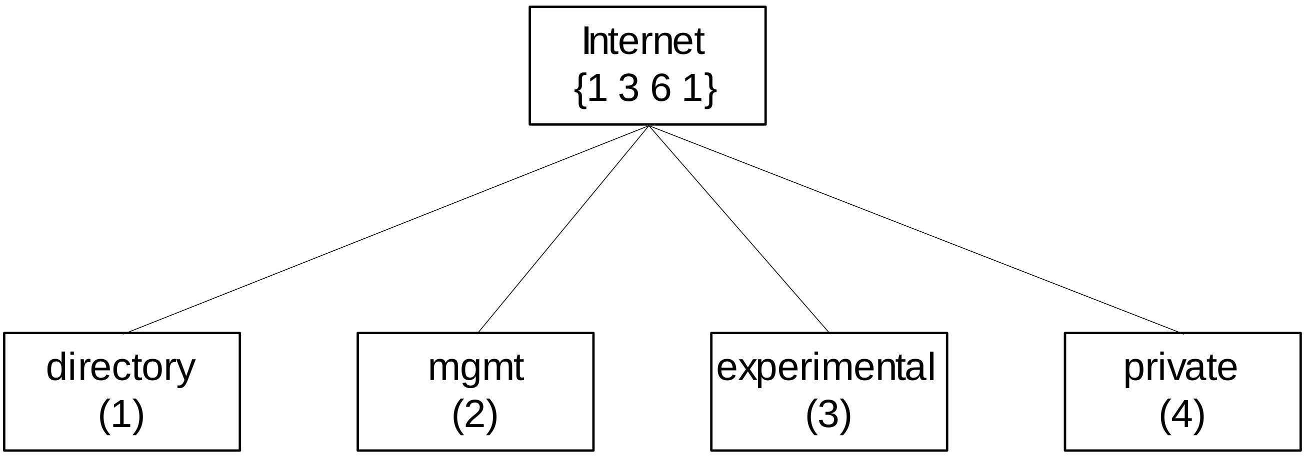


Figure 4.13 Subnodes under Internet Node in SNMPv1

Notes

- directory OBJECT IDENTIFIER ::= {internet 1}
  mgmt OBJECT IDENTIFIER ::= {internet 2}
  experimental OBJECT IDENTIFIER ::= {internet 3}
  private OBJECT IDENTIFIER ::= {internet 4}

Private MIB Example

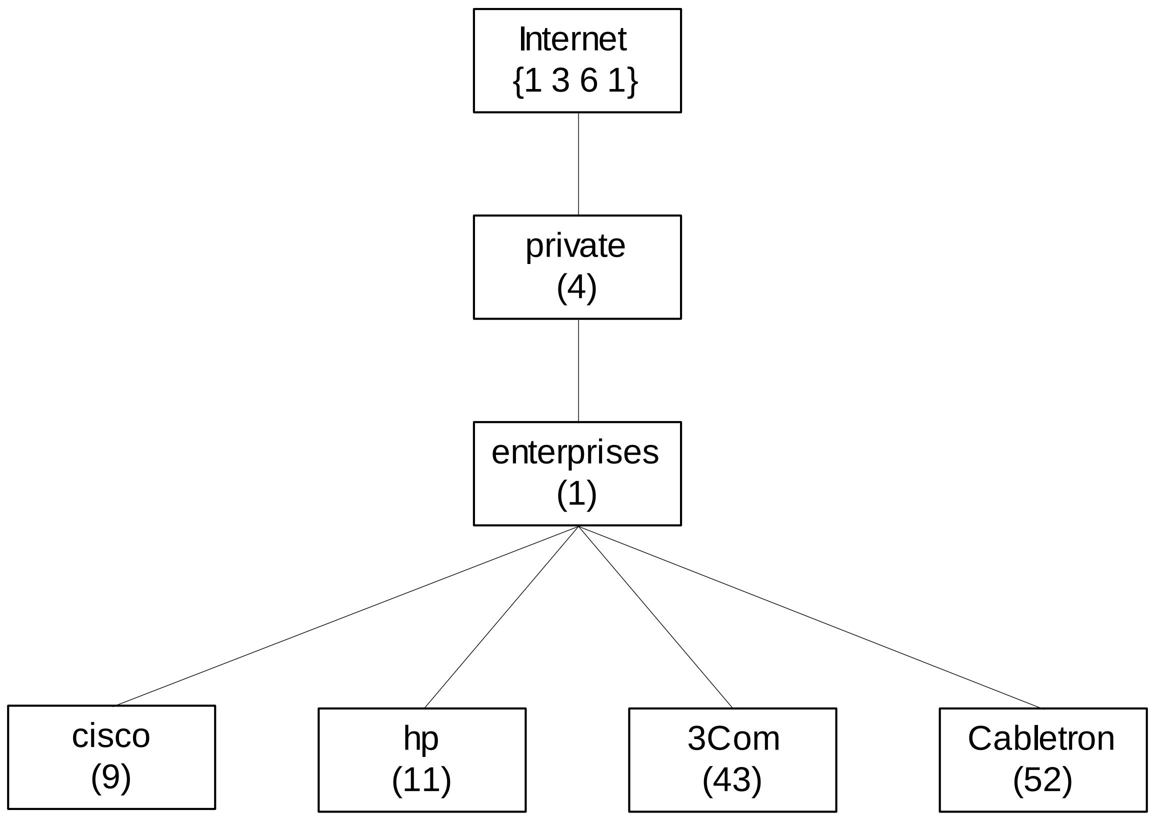


Figure 4.14 Private Subtree for Commercial Vendors

Notes

- *private* MIB intended for vendor equipment
- IANA (Internet Assigned Numbers Authority) assigns identifiers

SNMP ASN.1 Data Type

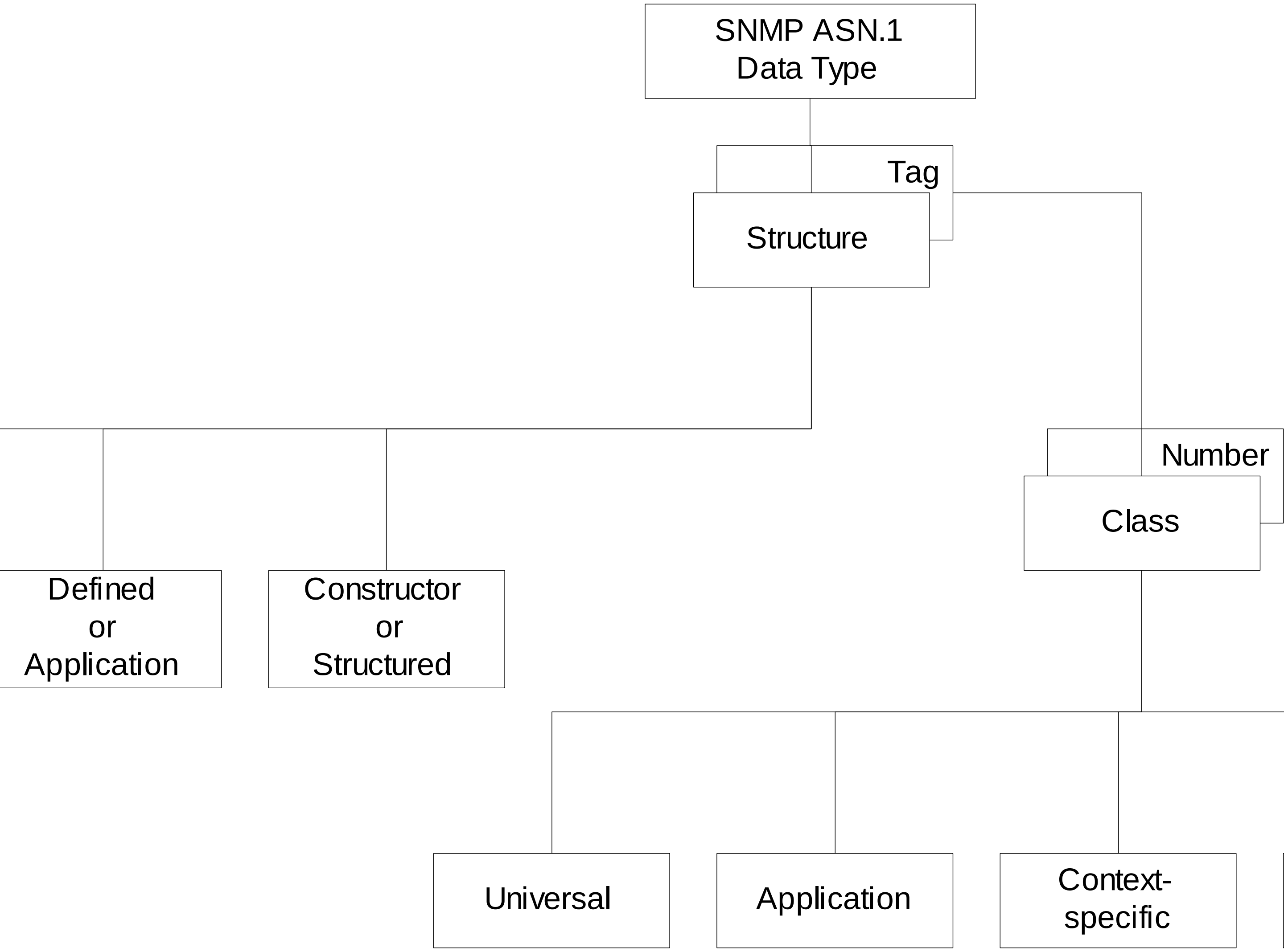


Figure 4.15 SNMP ASN.1 Data Type

Primitive Data Types

Structure	Data Type	Comments
Primitive types	INTEGER	Subtype INTEGER (n1..nN) Special case: Enumerated INTEGER type
	OCTET STRING	8-bit bytes binary and textual data Subtypes can be specified by either range or fixed
	OBJECT IDENTIFIER	Object position in MIB
	NULL	Placeholder

Notes

- *get-request* message has NULL for value fields and *get-response* from agent has the values filled in
- subtype:
  - INTEGER (0..255)
  - OCTET STRING (SIZE 0..255)
  - OCTET STRING (SIZE 8)

Enumerated

- Special case of INTEGER data type

```
error-status INTEGER {
        noError(0)
        tooBig(1)
        genErr(5)
        authorizationError(16)
    }
```

Notes

- noError NULL by convention

Defined or Application Data Type

Defined types	NetworkAddress	Not used
	IpAddress	Dotted decimal IP address
	Counter	Wrap-around, non-negative integer, monotonically increasing, max 2^32 -1
	Gauge	Capped, non-negative integer, increase or decrease
	TimeTicks	Non-negative integer in hundredths of second units
	Opaque	Application-wide arbitrary ASN.1 syntax, double wrapped OCTET STRING

Notes

- Defined data types are simple or base types
- Opaque is used to create data types based on previously defined data types

Constructor or Structured Data Type: SEQUENCE

- List maker

SEQUENCE { <type1>, <type2>,…, <typeN> }

	Object	OBJECT IDENTIFIER	ObjectSyntax
1	ipAdEntAddr	{ipAddrEntry 1}	IpAddress
2	ipAdEntIfIndex	{ipAddrEntry 2}	INTEGER
3	ipAdEntNetMask	{ipAddrEntry 3}	IpAddress
4	ipAdEntBcastAddr	{ipAddrEntry 4}	INTEGER
5	ipAdEntReasmMaxSize	{ipAddrEntry 5}	INTEGER
6	ipAddrEntry	{ipAddrTable 1}	SEQUENCE

```
List:   IpAddrEntry  ::=
            SEQUENCE {
                ipAdEntAddr             IpAddress
                ipAdEntIfIndex          INTEGER
                ipAdEntNetMask          IpAddress
                ipAdEntBcastAddr        INTEGER
                ipAdEntReasmMaxSize     INTEGER (0..65535)
            }
```

Managed Object IpAddrEntry as a list

Notes

Constructor or Structured Data Type: SEQUENCE OF

SEQUENCE OF <entry>
where <entry> is a list constructor

	Object Name	OBJECT IDENTIFIER	Syntax
7	ipAddrTable	{ip 20}	SEQUENCE OF

Table: IpAddrTable ::=
SEQUENCE OF IpAddrEntry

Managed Object ipAddrTable as a table

Notes

SEQUENCE OF Example

Title: System Information : router1.gatech.edu
Name or IP Address: 172.16252.1

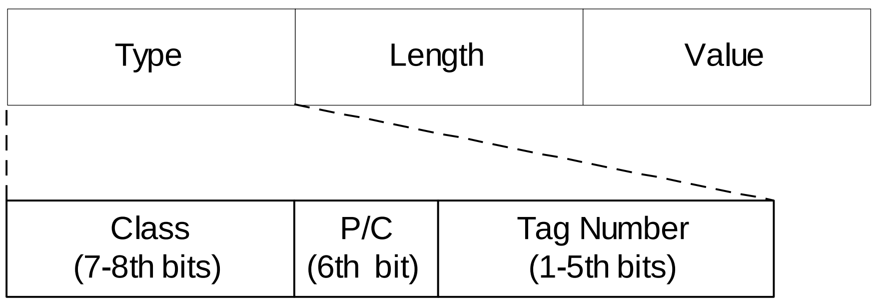
Index	Interface	IP address	Network Mask	Network Address	Link Address
23	LEC.1.0	192.168.3.1	255.255.255.0	192.168.3.0	0x00000C3920B4
25	LEC.3.9	192.168.252.15	255.255.255.0	192.168.252.0	0x00000C3920B4
13	Ethernet2/0	172.16..46.1	255.255.255.0	172.16..46.0	0x00000C3920AC
16	Ethernet2/3	172.16.49.1	255.255.255.0	172.16.49.0	0x00000C3920AF
17	Ethernet2/4	172.16.52.1	255.255.255.0	172.16.52.0	0x00000C3920B0
9	Ethernet1/2	172.16.55.1	255.255.255.0	172.16.55.0	0x00000C3920A6
2	Ethernet 0/1	172.16.56.1	255.255.255.0	172.16.56.0	0x00000C39209D
15	Ethernet2/2	172.16.57.1	255.255.255.0	172.16.57.0	0x00000C3920AE
8	Ethernet1/1	172.16.58.1	255.255.255.0	172.16.58.0	0x00000C3920A5
14	Ethernet2/1	172.16.60.1	255.255.255.0	172.16.60.0	0x00000C3920AD

Notes

- The above example (Figure 4.3) uses part of the IP MIB discussed for SEQUENCE OF construct

Encoding

- Basic Encoding Rules (BER)
    - Tag, Length, and Value (TLV)



- SNMP Data Types and Tags

Type	Tag
OBJECT IDENTIFIER	UNIVERSAL 6
SEQUENCE	UNIVERSAL 16
IpAddress	APPLICATION 0
Counter	APPLICATION 1
Gauge	APPLICATION 2
TimeTicks	APPLICATION 3
Opaque	APPLICATION 4

Notes

Managed Object: Structure

<u>OBJECT:</u>

sysDescr:	{ system 1 }
Syntax:	OCTET STRING
Definition:	"A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters."
Access:	read-only
Status:	mandatory

Figure 4.17 Specifications for System Description

Notes

Managed Object: Macro

```
OBJECT-TYPE MACRO ::=
BEGIN
        TYPE NOTATION ::= "SYNTAX" type(TYPE ObjectSyntax)
                "ACCESS" Access
                "STATUS" Status
        VALUE NOTATION ::= value(VALUE ObjectName)

        Access ::=  "read-only" | "write-only" | "not-accessible"
        Status ::= "mandatory" | "optional" | "obsolete"

END
```

Figure 4.18(a) OBJECT-TYPE Macro [RFC 1155]

Notes

```
sysDescr OBJECT-TYPE
        SYNTAX DisplayString (SIZE (0..255))
        ACCESS read-only
        STATUS   mandatory
        DESCRIPTION
                "A textual description of the entity. This value should
                include the full name and version identification of the
                system's hardware type, software operating-system, and
                networking software. It is mandatory that this only contain
                printable ASCII characters."
::= {system 1 }
```

Figure 4.18(b) Scalar or Single Instance Macro: sysDescr

[RFC 1213]

Aggregate Object

- A group of objects
- Also called tabular objects
- Can be represented by a table with
  - Columns of objects
  - Rows of instances

Table of Objects

↓

List of Objects

↓

Objects

Notes

- Example: IP address table
- Consists of objects:
  - IP address
  - Interface
  - Subnet mask (which subnet this address belongs to)
  - Broadcast address (value of l.s.b. in IP broadcast address)
  - Largest IP datagram that can be assembled
- Multiple instances of these objects associated with the node

Aggregate M.O. Macro: Table Object

```
ipAddrTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IpAddrEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
            "The table of addressing
            information relevant to this entity's IP
           addresses."
    ::= {ip 20}
```

Notes

```
ipAddrTable       OBJECT-TYPE
          ::= {ip 20}
ipAddrEntry       OBJECT-TYPE
          ::= {ipAddrTable 1}
```

Aggregate M.O. Macro: Entry Object

```
ipAddrEntry OBJECT-TYPE
    SYNTAX  IpAddrEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
         "The addressing information for one of this
entity's IP addresses."

    INDEX   { ipAdEntAddr }
    ::= { ipAddrTable 1 }

  IpAddrEntry ::=
    SEQUENCE {
      ipAdEntAddr
        IpAddress,
      ipAdEntIfIndex
        INTEGER,
      ipAdEntNetMask
        IpAddress,
      ipAdEntBcastAddr
        INTEGER,
      ipAdEntReasmMaxSize
        INTEGER (0..65535)
```

Notes

- Index *ipAdEntAddr* uniquely identifies an instance
- May require more than one object in the instance to uniquely identify it

Aggregate M.O. Macro: Columnar Objects

```
ipAdEntAddr OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The IP address to which this entry's
              addressing information pertains."

    ::= { ipAddrEntry 1 }

ipAdEntReasmMaxSize OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The size of the largest IP datagram which this
            entity can re-assemble from incoming IP
            fragmented datagrams received on this interface."
::= { ipAddrEntry 5 }
```

Notes

Tabular Representation of Aggregate Object

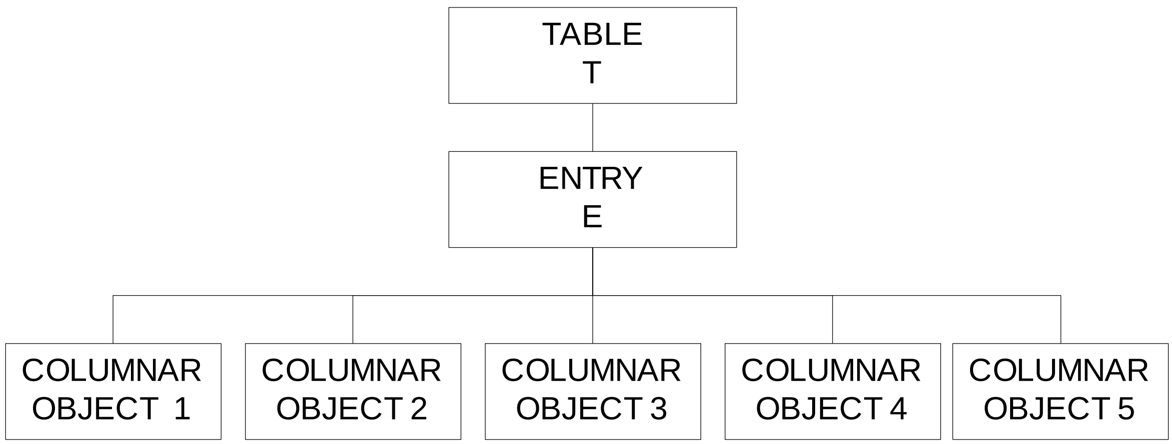


Figure 4.22(a) Multiple Instance Managed Object

Notes

- The objects *TABLE T* and *ENTRY E* are objects that are logical objects. They define the grouping and are not accessible
- Columnar objects are objects that represent the attributes and hence are accessible
- Each instance of *E* is a row of columnar objects 1 through 5
- Multiple instances of *E* are represented by multiple rows

Tabular Representation of Aggregate Object

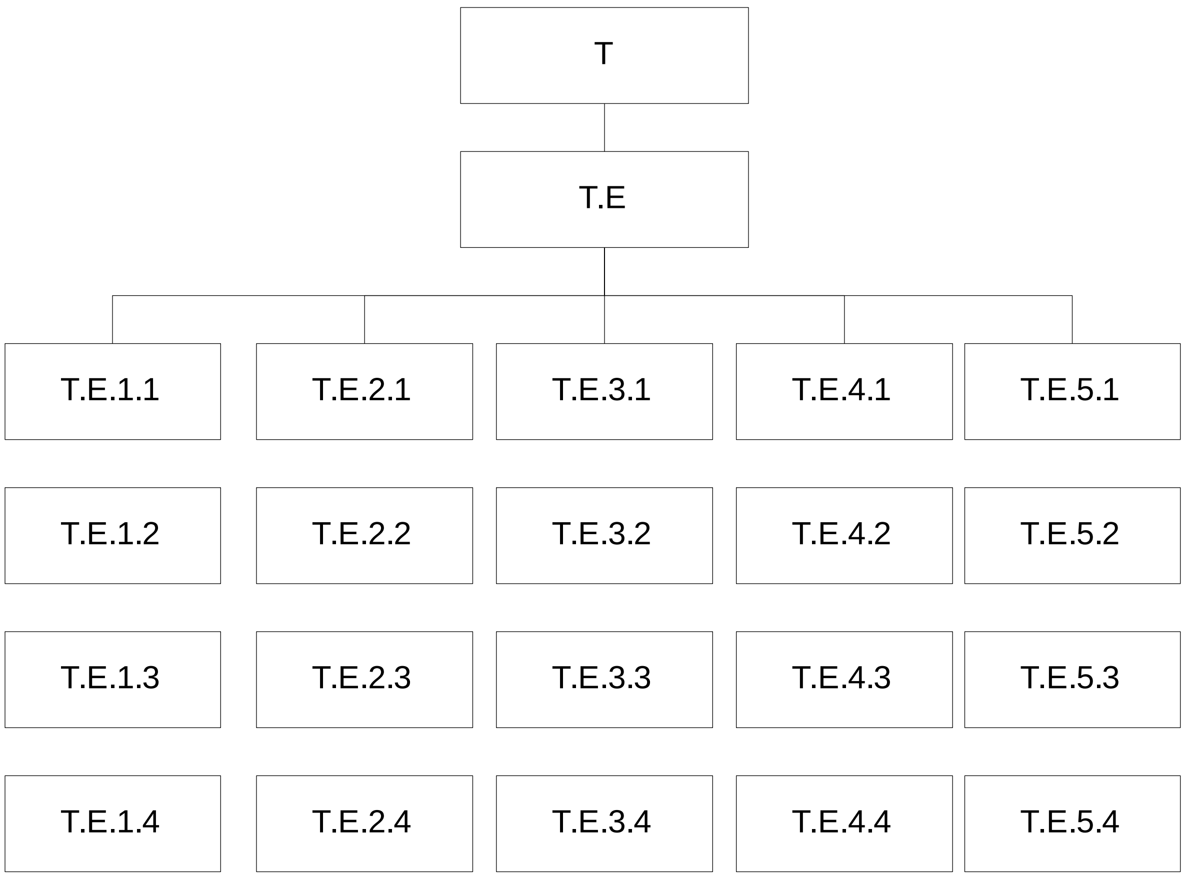


Figure 4.22(b) Example of 5 Columnar Object with 4 Instances (rows)

Notes

- Notice that the column-row numeric designation is reverse of what we are used to as row-column

Multiple Instances of Aggregate Managed Object

ipAddrTable {1.3.6.1.2.1.4.20}
    ipAddrEntry (1)
        ipAdEntAddr (1)
        ipAdEntIfIndex (2)
        ipAdEntNetMask (3)
        ipAdEntBcastAddr (4)
        ipAdEntReasmMaxSize (5)

Columnar object ID of ipAdEntBcastAddr is (1.3.6.1.2.1.4.20.1.4):

iso	org	dod	internet	mgmt	mib	ip	ipAddrTable	ipAddrEntry	ipAdEntBcastAddr
1	3	6	1	2	1	4	20	1	4

Figure 4.23(a) Columnar objects under ipAddrEntry

Row	ipAdEntAddr	ipAdEntIfIndex	IpAdEntNetMask	IpAdEntBcastAddr	IpAdEntReasmMaxSize
1	**123.45.2.1**	1	255.255.255.0	0	12000
2	**123.45.3.4**	3	255.255.0.0	1	12000
3	**165.8.9.25**	2	255.255.255.0	0	10000
4	**9.96.8.138**	4	255.255.255.0	0	15000

Figure 4.23(b) Object instances of ipAddrTable (1.3.6.1.2.1.4.20)

Columnar Object	Row # in (b)	Object Identifier
ipAdEntAddr 1.3.6.1.2.1.4.20.1.1	2	{1.3.6.1.2.1.4.20.1.1.123.45.3.4}
ipAdEntIfIndex 1.3.6.1.2.1.4.20.1.2	3	{1.3.6.1.2.1.4.20.1.2.165.8.9.25}
ipAdEntBcastAddr 1.3.6.1.2.1.4.20.1.4	1	{1.3.6.1.2.1.4.20.1.4.123.45.2.1}
IpAdEntReasmMaxSize 1.3.6.1.2.1.4.20.1.5	4	{1.3.6.1.2.1.4.20.1.5.9.96.8.138}

Figure 4.23(c) Object Id for specific instance

SMI Definition STD 16 / 1155 RFC

```
RFC1155-SMI DEFINITIONS ::= BEGIN

    EXPORTS -- EVERYTHING
        internet, directory, mgmt, experimental, private, enterprises,
        OBJECT-TYPE, ObjectName, ObjectSyntax, SimpleSyntax,
        ApplicationSyntax, NetworkAddress, IpAddress, Counter, Gauge,
        TimeTicks, Opaque;

    -- the path to the root

    internet      OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }

    directory     OBJECT IDENTIFIER ::= { internet 1 }
    mgmt          OBJECT IDENTIFIER ::= { internet 2 }
    experimental  OBJECT IDENTIFIER ::= { internet 3 }
    private       OBJECT IDENTIFIER ::= { internet 4 }

    enterprises   OBJECT IDENTIFIER ::= { private 1 }
```

Notes

- EXPORTS identifies the objects that any other module could import

SMI Definition STD 16 / 1155 RFC

-- definition of object types

```
OBJECT-TYPE MACRO ::=
BEGIN
   TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
                "ACCESS" Access
                "STATUS" Status
   VALUE NOTATION ::= value (VALUE ObjectName)

   Access ::= "read-only" | "read-write" | "write-only"  "not-accessible"
   Status ::= "mandatory" | "optional" | "obsolete"
END
```

Notes

SMI Definition STD 16 / 1155 RFC

```
-- names of objects in the MIB

    ObjectName ::=
        OBJECT IDENTIFIER

    -- syntax of objects in the MIB

    ObjectSyntax ::=
        CHOICE {
            simple
                SimpleSyntax,

            application-wide
                ApplicationSyntax
        }
```

Notes

SMI Definition STD 16 / 1155 RFC

```
SimpleSyntax ::=
        CHOICE {
            number
                INTEGER,
            string
                OCTET STRING,
            object
                OBJECT IDENTIFIER,
            empty
                NULL
        }
```

```
ApplicationSyntax ::=
        CHOICE {
            address
                NetworkAddress,
            counter
                Counter,
            gauge
                Gauge,
            ticks
                TimeTicks,
            arbitrary
                Opaque

    -- other application-wide types, as they are defined,
       will be added here
        }
```

SMI Definition STD 16 / 1155 RFC

```
-- application-wide types

NetworkAddress ::=
    CHOICE {
        internet
            IpAddress
    }
IpAddress ::=
    [APPLICATION 0]          -- in network-byte order
        IMPLICIT OCTET STRING (SIZE (4))
Counter ::=
    [APPLICATION 1]
        IMPLICIT INTEGER (0..4294967295)
Gauge ::=
    [APPLICATION 2]
        IMPLICIT INTEGER (0..4294967295)
TimeTicks ::=
    [APPLICATION 3]
        IMPLICIT INTEGER (0..4294967295)
Opaque ::=
    [APPLICATION 4]          -- arbitrary ASN.1 value,
        IMPLICIT OCTET STRING   --   "double-wrapped"

END
```

Notes

MIB

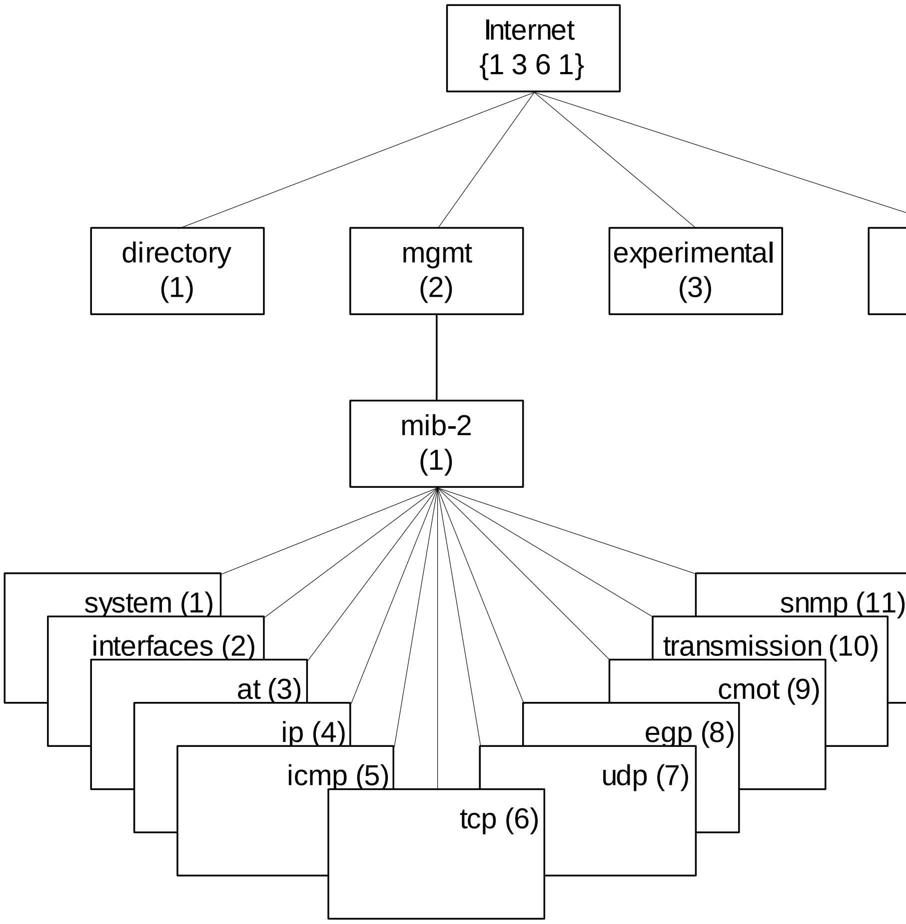


Figure 4.26 Internet MIB-II Group

Notes

- MIB-II (RFC 1213) is superset of MIB-I
- Objects that are related grouped into object groups
- MIB module comprises module name, imports from other modules, and definitions of current module
- RFC 1213 defines eleven groups; expanded later

System Group

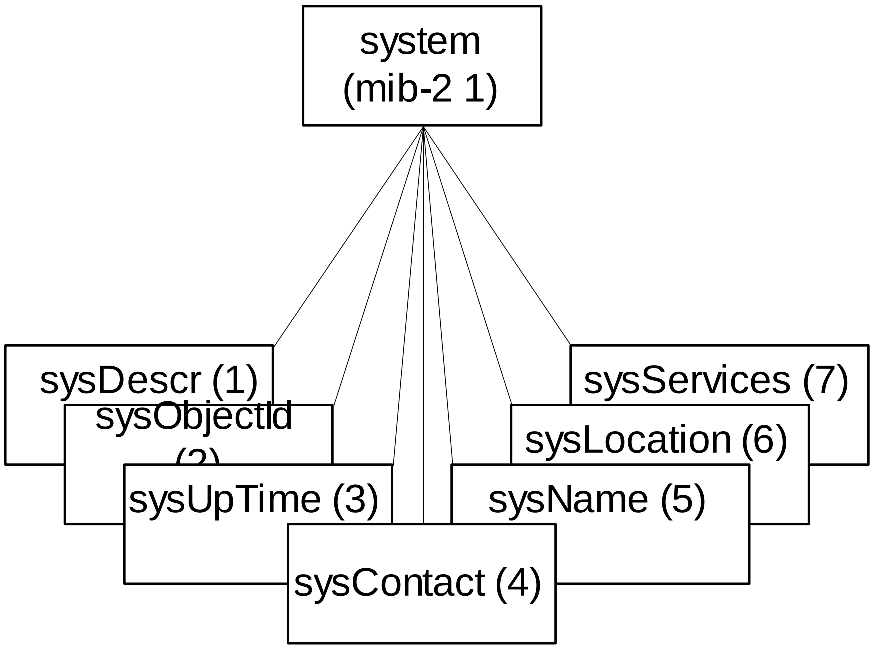


Figure 4.27 System Group

Notes

Entity	OID	Description (brief)
sysDescr	system 1	Textual description
sysObjectID	system 2	OBJECT IDENTIFIER of the entity
sysUpTime	system 3	Time (in hundredths of a second since last reset)
sysContact	system 4	Contact person for the node
sysName	system 5	Administrative name of the system
sysLocation	system 6	Physical location of the node
sysServices	system 7	Value designating the layer services provided by the entity

sysServices

```
sysServices OBJECT-TYPE
        SYNTAX  INTEGER (0..127)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "A value which indicates the set of services that
            this entity primarily offers.

            The value is a sum.  This sum initially takes the
            value zero, Then, for each layer, L, in the range
            1 through 7, that this node performs transactions
            for, 2 raised to (L - 1) is added to the sum.  For
            example, a node which performs primarily routing
            functions would have a value of 4 (2^(3-1)).  In
            contrast, a node which is a host offering
            application services would have a value of 72
            (2^(4-1) + 2^(7-1)).  Note that in the context of
            the Internet suite of protocols, values should be
            calculated accordingly:

               layer functionality
                 1  physical (e.g., repeaters)
                 2  datalink/subnetwork (e.g., bridges)
                 3  internet (e.g., IP gateways)
                 4  end-to-end  (e.g., IP hosts)
                 7  applications (e.g., mail relays)

            For systems including OSI protocols, layers 5 and
            6 may also be counted."
        ::= { system 7 }
```

Notes

Interfaces Group

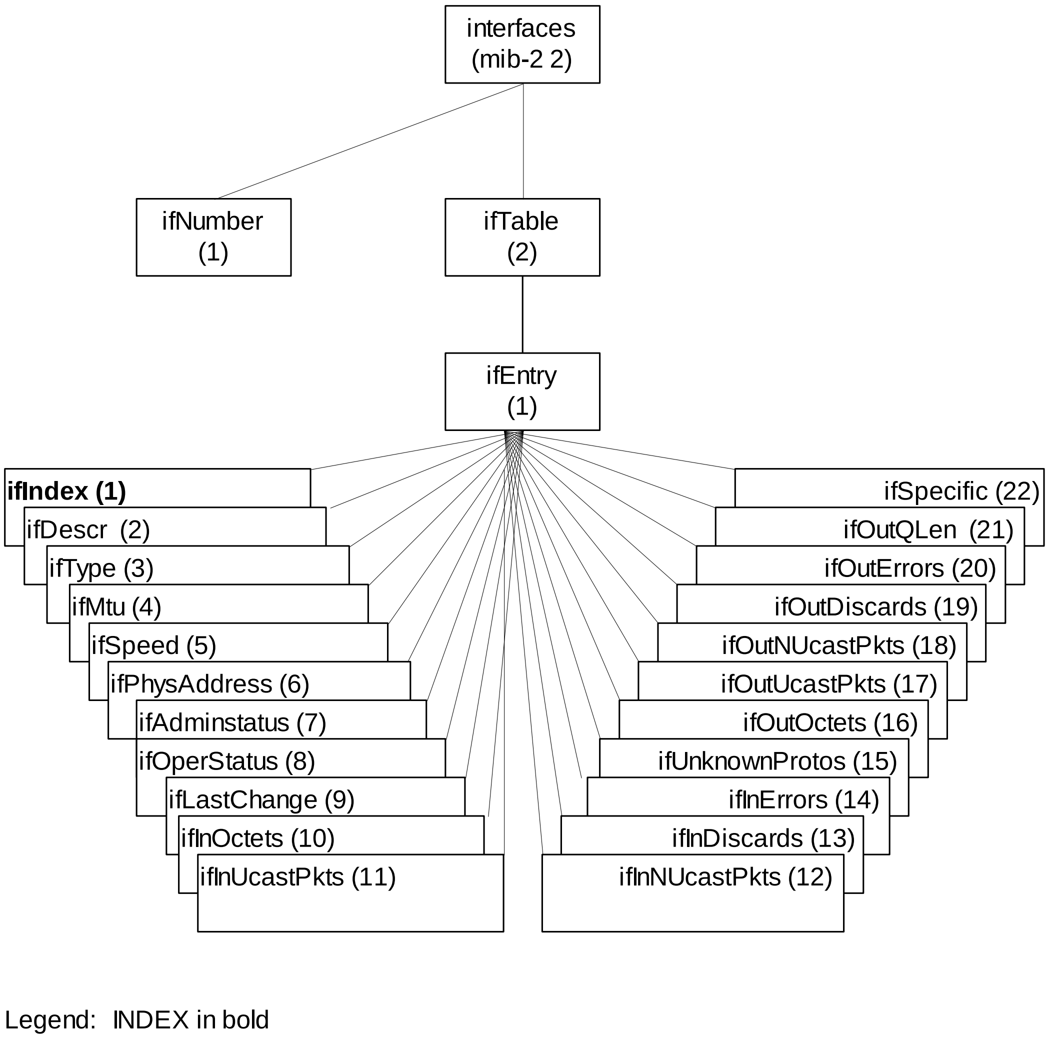


Figure 4.28 Interfaces Group

Notes

ifEntry

```
IfEntry  OBJECT-TYPE
         SYNTAX      IfEntry
         ACCESS      not-accessible
         STATUS      mandatory
         DESCRIPTION
              "An interface entry containing
              objects at the subnetwork layer and
              below for a particular interface."
         INDEX  {ifIndex}
         ::= {ifTable 1}
```

Notes

- ifEntry specifies the objects in a row in the ifTable
- Each interface is defined as a row in the table

ifType

```
ifType OBJECT-TYPE
        SYNTAX  INTEGER {
                other(1),          -- none of the following
                regular1822(2),
                hdh1822(3),
                ddn-x25(4),
                rfc877-x25(5),
                ethernet-csmacd(6),
                iso88023-csmacd(7),
                iso88024-tokenBus(8),
                iso88025-tokenRing(9),
                iso88026-man(10),
                starLan(11),
                proteon-10Mbit(12),
                proteon-80Mbit(13),
                hyperchannel(14),
                fddi(15),
                lapb(16),
                sdlc(17),
                ds1(18),           -- T-1
                e1(19),            -- european equiv. of T-1
                basicISDN(20),
                primaryISDN(21),   -- proprietary serial
                propPointToPointSerial(22),
                ppp(23),
                ……….
```

Notes

- Type of interface below the network layer defined as enumerated integer

IP Group

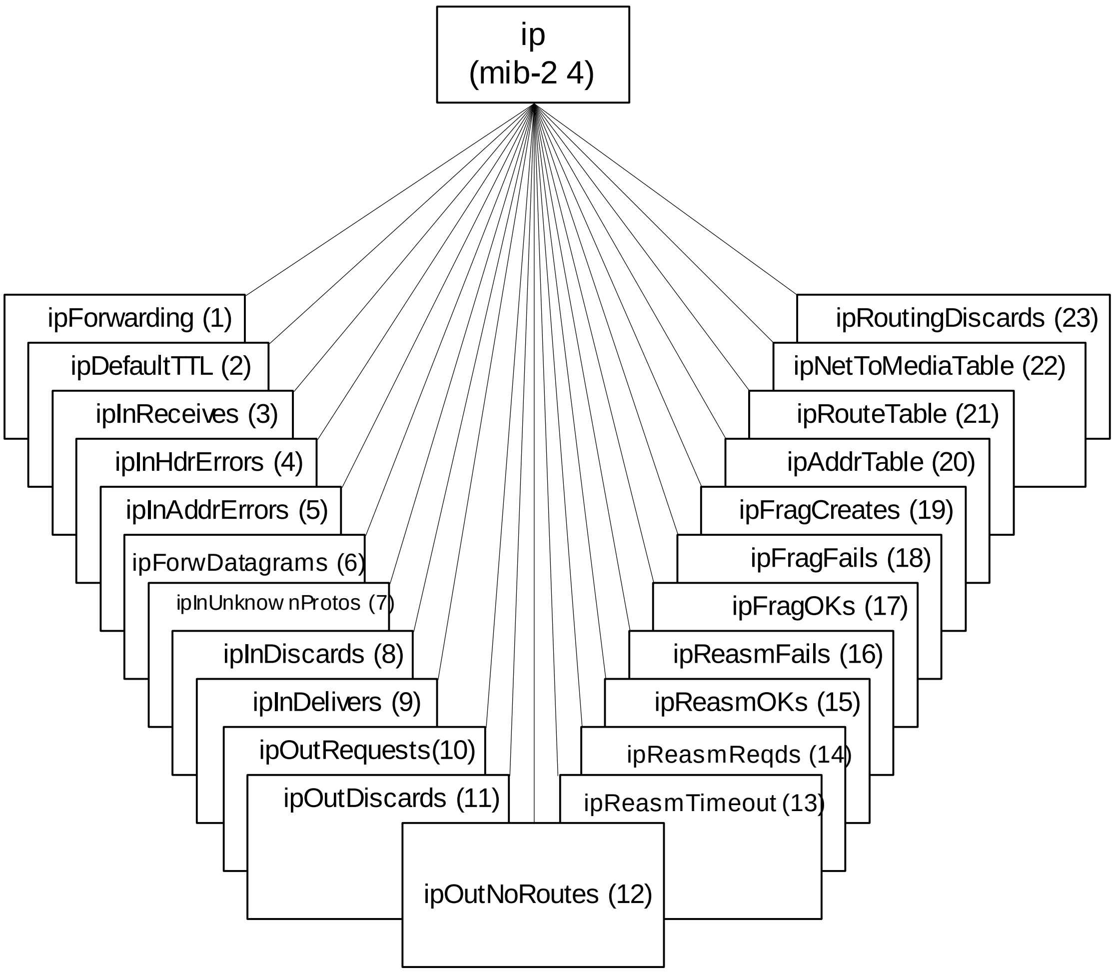


Figure 4.29 IP Group

Notes

- ipForwarding: Gateway(1) and Router(2)
- IP Address Table contains table of IP addresses
- IP Route Table contains an entry for each route
- IP Network-to-Media Table is address translation table mapping IP addresses to physical addresses

IP Address Table

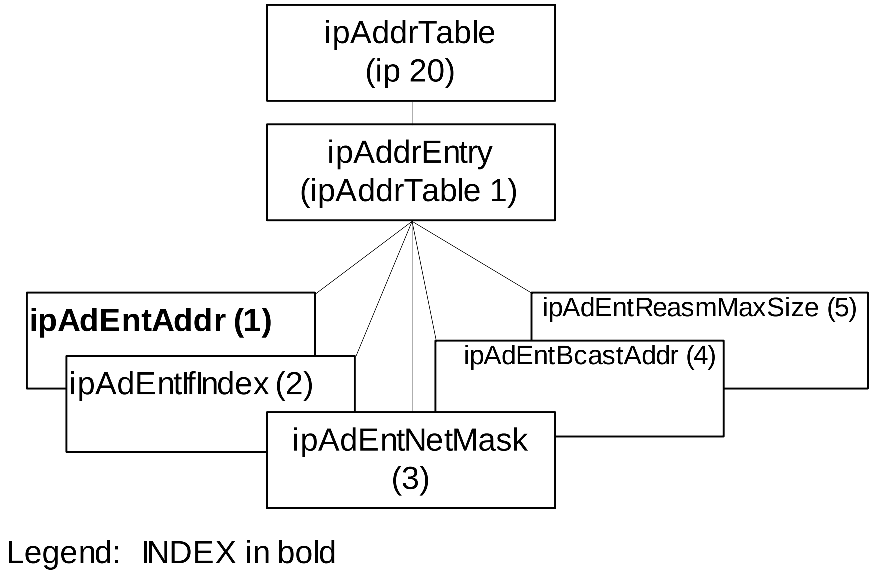


Figure 4.30 IP Address Table

Notes

Entity	OID	Description (brief)
ipAddrTable	ip 20	Table of IP addresses
ipAddrEntry	IpAddrTable 1	One of the entries in the IP address table
ipAdEntAddr	IpAddrEntry 1	The IP address to which this entry's addressing information pertains
ipAdEntIfIndex	IpAddrEntry 2	Index value of the entry, same as ifIndex
ipAdEntNetMask	IpAddrEntry 3	Subnet mask for the IP address of the entry
ipAdEntBcastAddr	IpAddrEntry 4	Broadcast address indicator bit
ipAdEntReasmMaxSize	IpAddrEntry 5	Largest IP datagram that can be reassembled on this interface

IP Routing Table

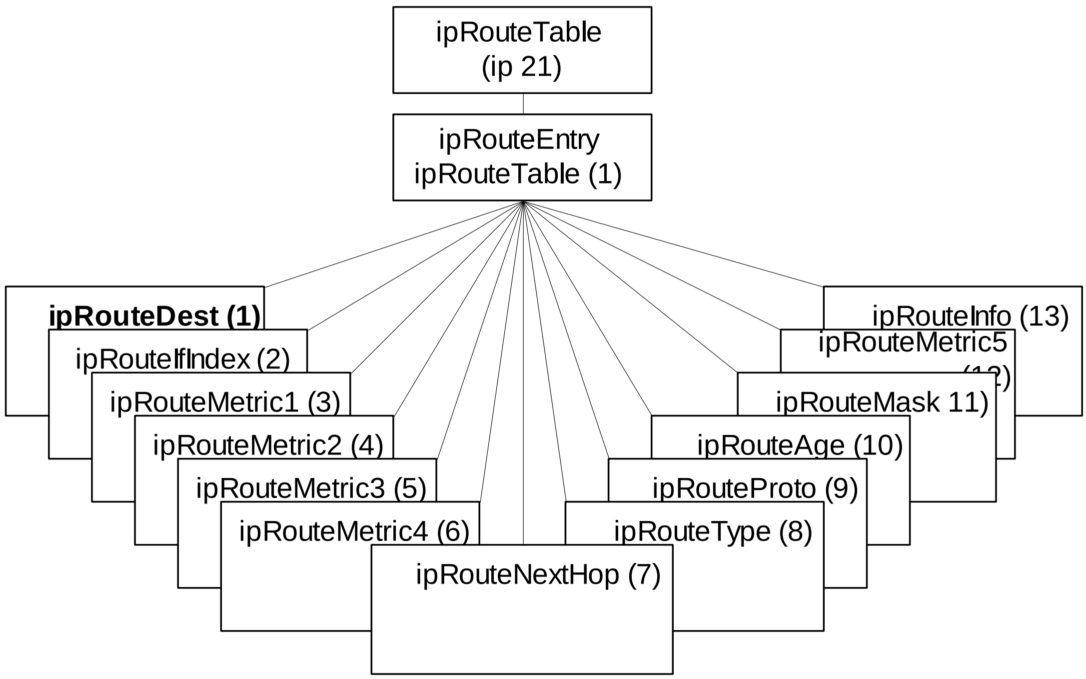


Figure 4.31 IP Routing Table

Entity	OID	Description (brief)
ipRouteTable	ip 21	IP routing table
ipRouteEntry	ipRouteTable 1	Route to a particular destination
ipRouteDest	ipRouteEntry 1	Destination IP address of this route
ipRouteIfIndex	ipRouteEntry 2	Index of interface, same as ifIndex
ipRouteMetric1	ipRouteEntry 3	Primary routing metric for this route
ipRouteMetric2	ipRouteEntry 4	An alternative routing metric for this route
ipRouteMetric3	ipRouteEntry 5	An alternative routing metric for this route
ipRouteMetric4	ipRouteEntry 6	An alternative routing metric for this route
ipRouteNextHop	ipRouteEntry 7	IP address of the next hop
ipRouteType	ipRouteEntry 8	Type of route
ipRouteProto	ipRouteEntry 9	Routing mechanism by which this route was learned
ipRouteAge	ipRouteEntry 10	Number of seconds since routing was last updated
ipRouteMask	ipRouteEntry 11	Mask to be logically ANDed with the destination address before comparing with the ipRouteDest field
ipRouteMetric5	ipRouteEntry 12	An alternative metric for this route
ipRouteInfo	ipRouteEntry 13	Reference to MIB definition specific to the routing protocol

IP Address Translation Table

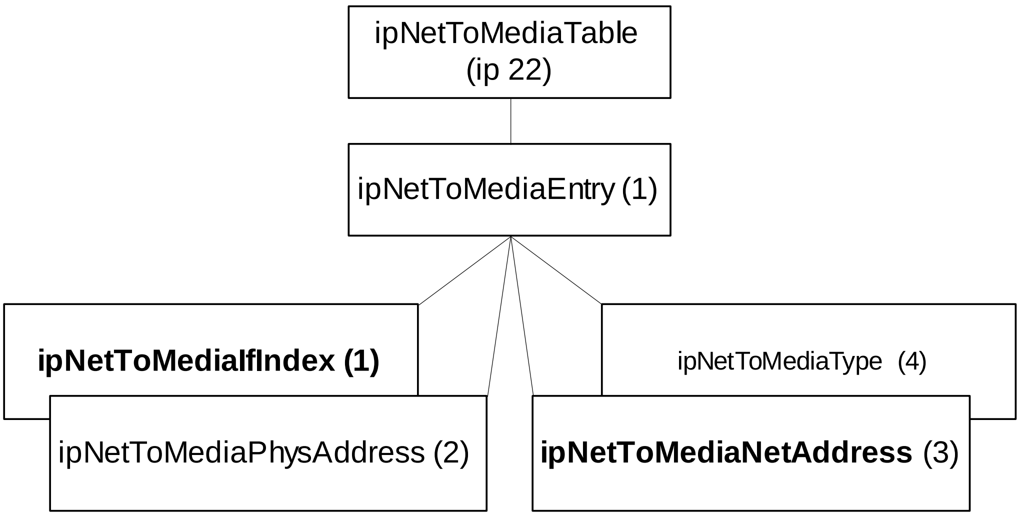


Figure 4.32 IP Address Translation Table

Notes

Entity	OID	Description (brief)
ipNetToMediaTable	ip 22	Table mapping IP addresses to physical addresses
ipNetToMediaEntry	IpNetToMediaTable 1	IP address to physical address for the particular interface
ipNetToMediaIfIndex	IpNetToMediaEntry 1	Interfaces on which this entry's equivalence is effective; same as ifIndex
ipNetToMediaPhysAddress	IpNetToMediaEntry 2	Media dependent physical address
ipNetToMediaNetAddress	IpNetToMediaEntry 3	IP address
ipNetToMediaType	IpNetToMediaEntry 4	Type of mapping

ICMP Group

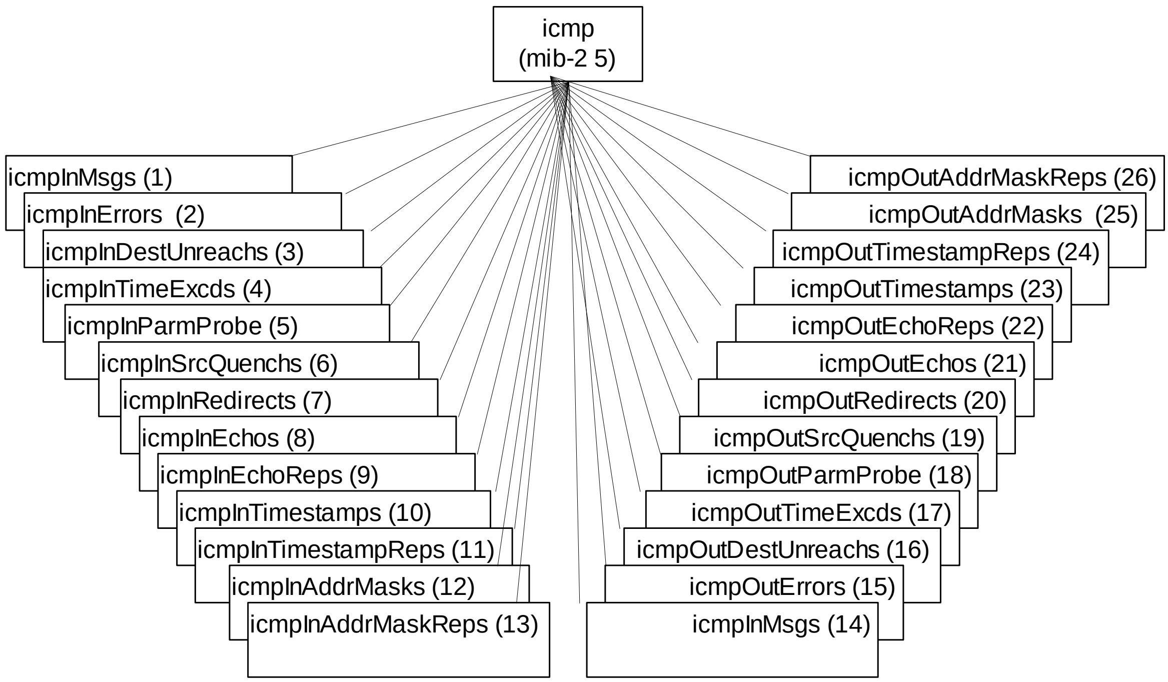


Figure 4.34 ICMP Group

Notes

- Objects associated with *ping*
    - icmpOutEchos # ICMP echo messages sent
    - icmpInEchoReps # ICMP echo reply messages received
- Objects associated with *traceroute/tracert*
    - icmpInTimeExcs # ICMP time exceeded messages received

TCP Group

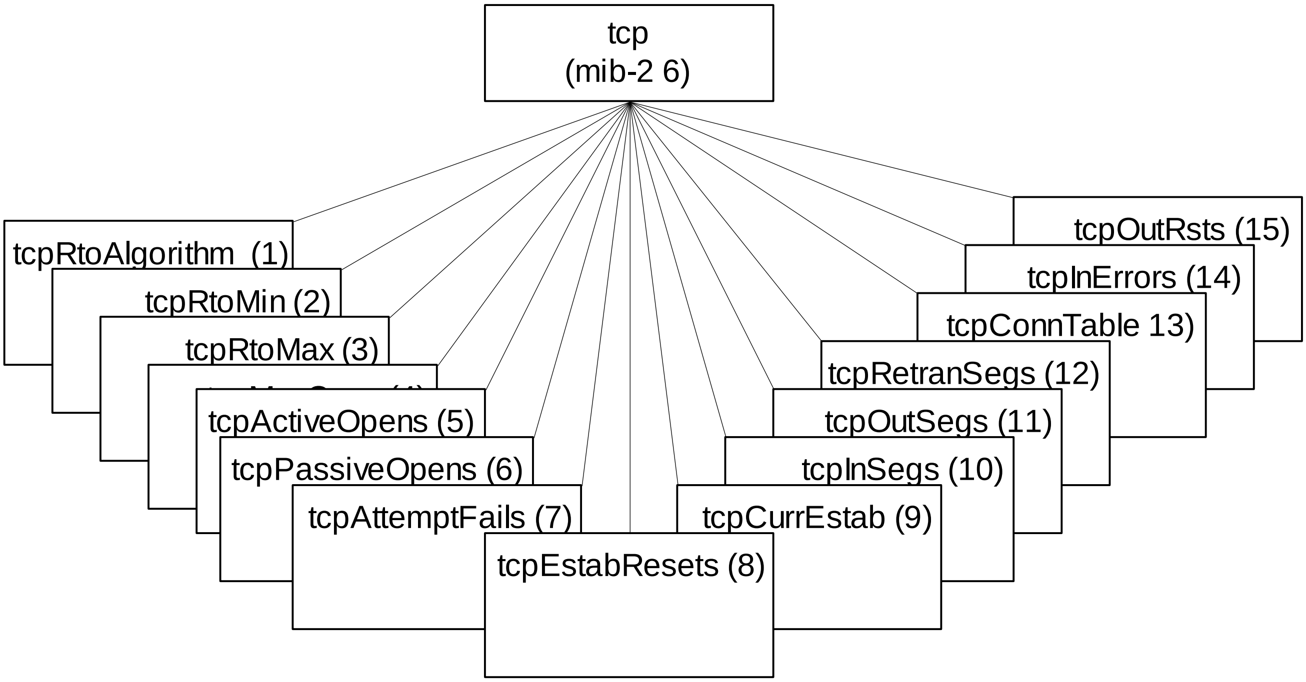


Figure 4.35 TCP Group

Notes

- Connection-oriented transport protocol group
- Has one table

TCP Connection Table

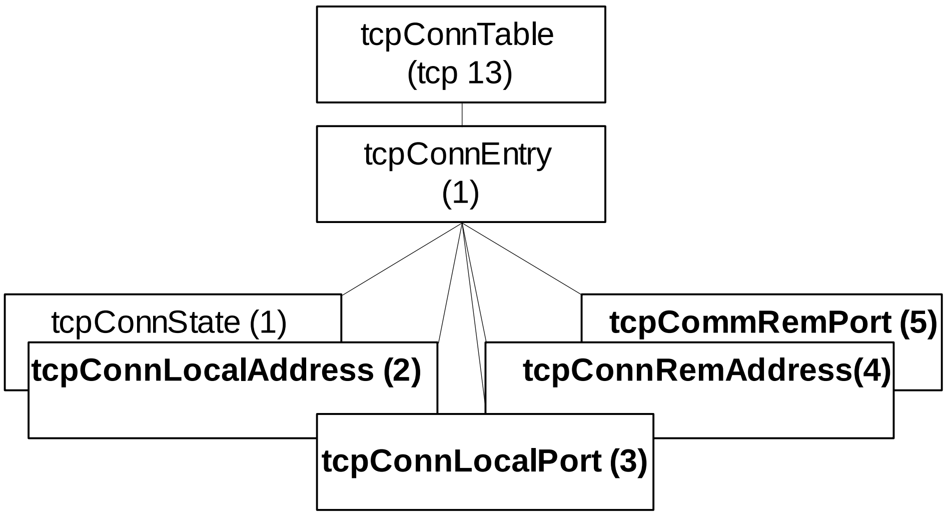


Figure 4.36 TCP Connection Table

Notes

Entity	OID	Description (brief)
tcpConnTable	tcp 13	TCO connection table
tcpconnEntry	TcpConnTable 1	Information about a particular TCP connection
tcpConnState	TcpConnEntry 1	State of the TCP connection
tcpConnLocalAddress	TcpConnEntry 2	Local IP address
tcpConnLocalPort	TcpConnEntry 3	Local port number
tcpConnRemAddress	TcpConnEntry 4	Remote IP address
tcpConnRemPort	TcpConnEntry 5	Remote port number

UDP Group

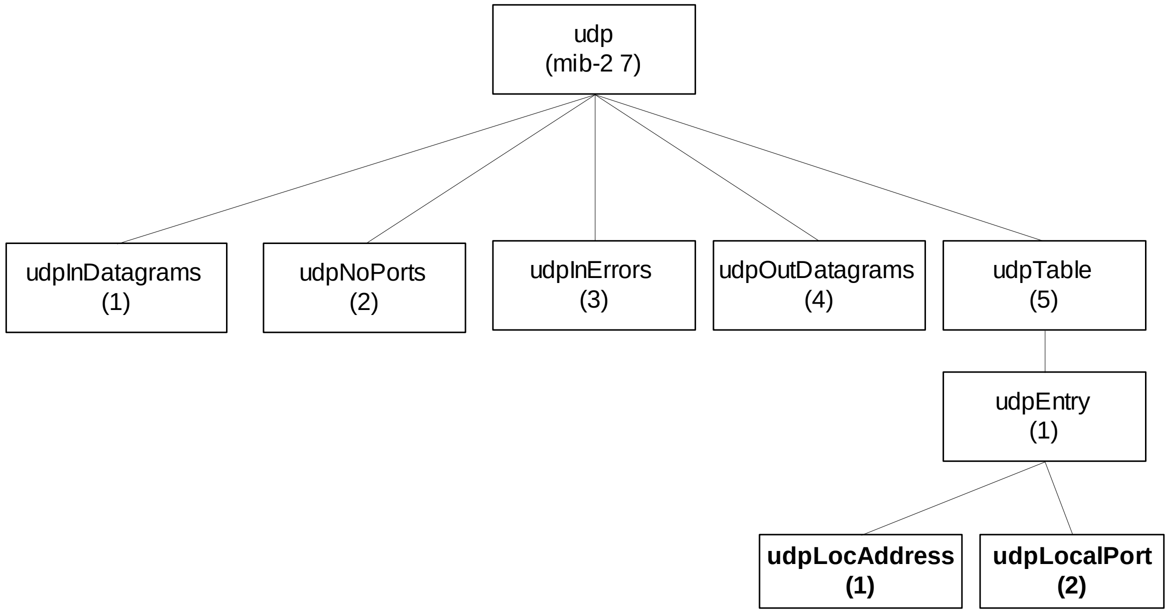


Figure 4.37 UDP Group

Notes

- Connectionless transport protocol group
- Has one table, UDP table

Entity	OID	Description (brief)
udpInDatagrams	udp 1	Total number of datagrams delivered to the users
udpNoPorts	udp 2	Total number of received datagrams for which there is no application
udpInErrors	udp 3	Number of received datagrams with errors
udpOutDatagrams	udp 4	Total number of datagrams sent
udpTable	udp 5	UDP Listener table
udpEntry	udpTable 1	Information about a particular connection or UDP listener
udpLocalAddress	udpEntry 1	Local IP address
udpLocalPort	udpEntry 2	Local UDP port

Chapter 5
SNMPv1:
Communication and Functional Models

SNMP Architecture

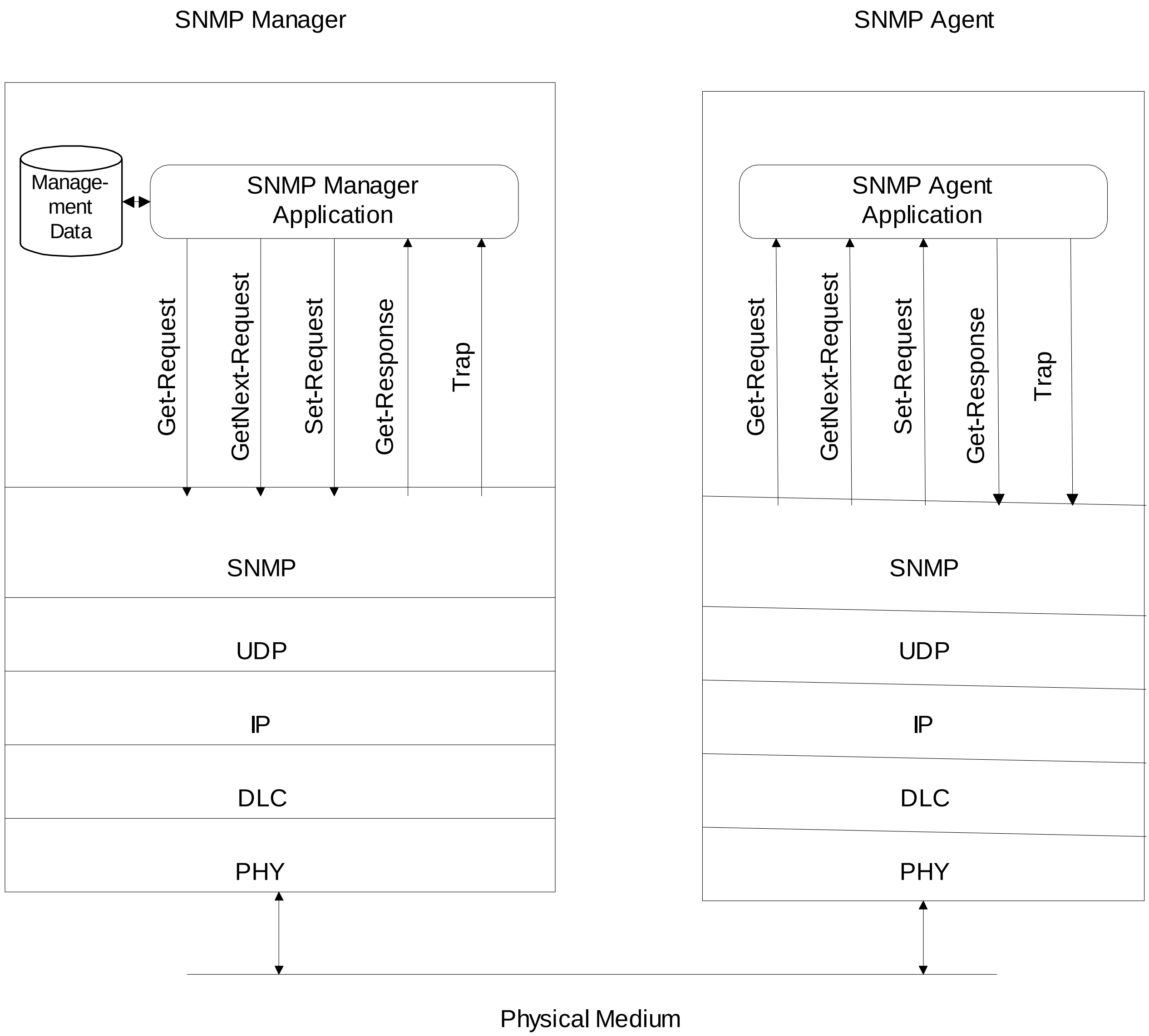


Figure 4.9 SNMP Network Management Architecture

Notes

- Truly *simple* network management protocol
- Five messages, three from manager and two from agent

SNMP Messages

- Get-Request
- Get-Next-Request
- Set-Request
- Get-Response
- Trap
    - Generic trap
    - Specific trap
    - Time stamp

Notes

- Generic trap
    - coldStart
    - warmStart
    - linkDown
    - linkUp
    - authenticationfailure
    - egpNeighborLoss
    - enterpriseSpecific
- Specific trap
    - for special measurements such as statistics
- Time stamp: Time since last initialization

Administrative Model

- Based on community profile and policy
- SNMP Entities:
    - SNMP application entities
        - Reside in management stations and network elements
        - Manager and agent
    - SNMP protocol entities
        - Communication processes (PDU handlers)
        - Peer processes that support application entities

Notes

SNMP Community

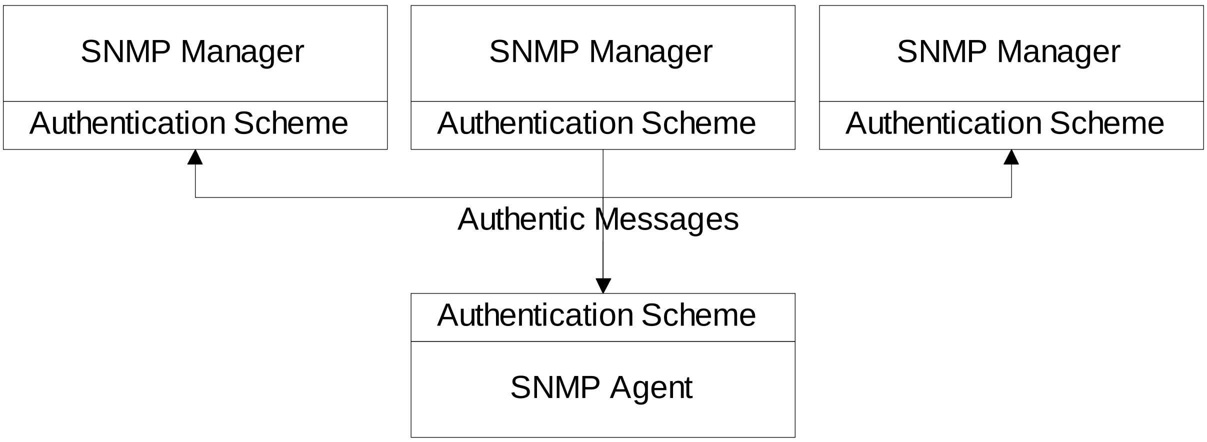


Figure 5.1 SNMP Community

Notes

- Security in SNMPv1 is community-based
- Authentication scheme in manager and agent
- Community: Pairing of two application entities
- Community name: String of octets
- Two applications in the same community communicate with each other
- Application could have multiple community names
- Communication is not secured in SNMPv1 - no encryption

Community Profile

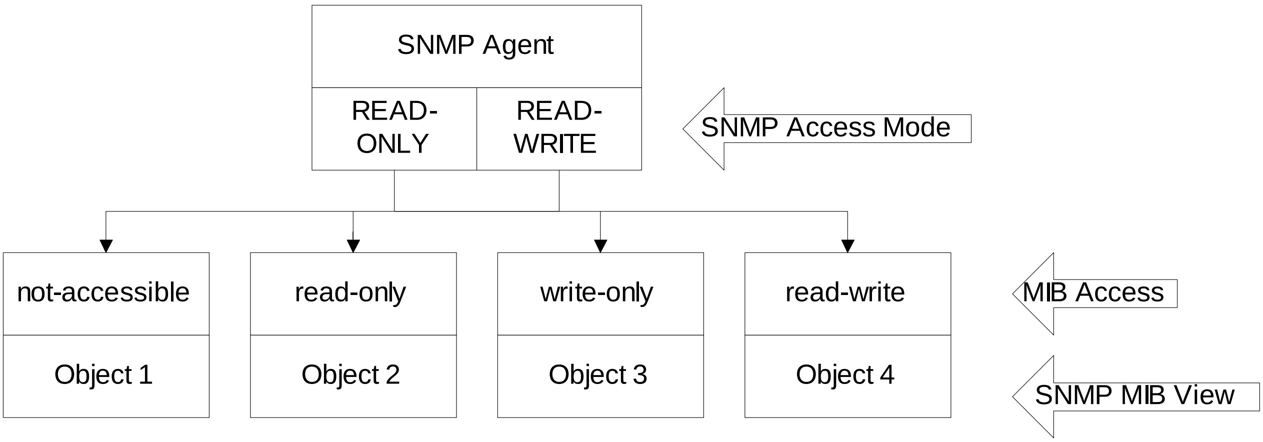


Figure 5.2 SNMP Community Profile

Notes

- MIB view
  - An agent is programmed to view only a subset of managed objects of a network element
- Access mode
  - Each community name is assigned an access mode:: read-only and read-write
- Community profile: MIB view + access mode
- Operations on an object determined by community profile and the access mode of the object
- Total of four access privileges
- Some objects, such as table and table entry are non-accessible

Administration Model

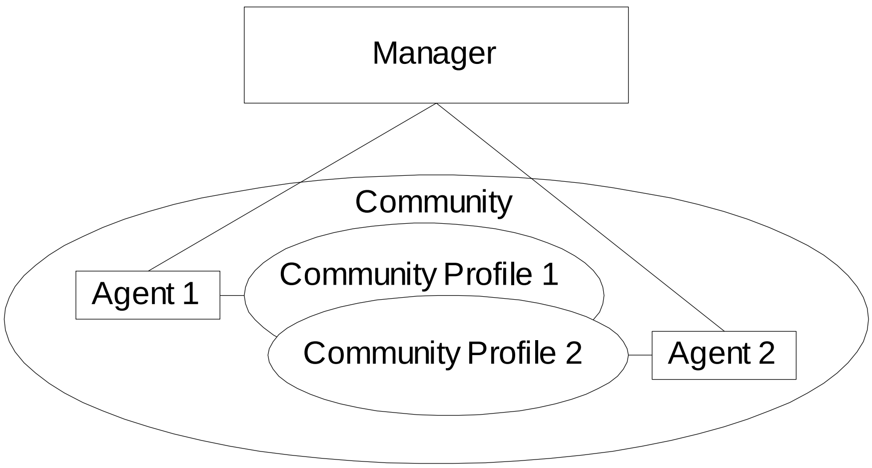
- Administration model is SNMP access policy
- SNMP community paired with SNMP community profile is SNMP access policy

Notes

Parameters:

- Community / communities
- Agent / Agents
- Manager / managers

Access Policy



Notes

- Manager manages Community 1 and 2 network components via Agents 1 and 2
- Agent 1 has only view of Community Profile 1, e.g. Cisco components
- Agent 2 has only view of Community Profile 2, e.g. 3Com components
- Manager has total view of both Cisco and 3Com components

Generalized Administration Model

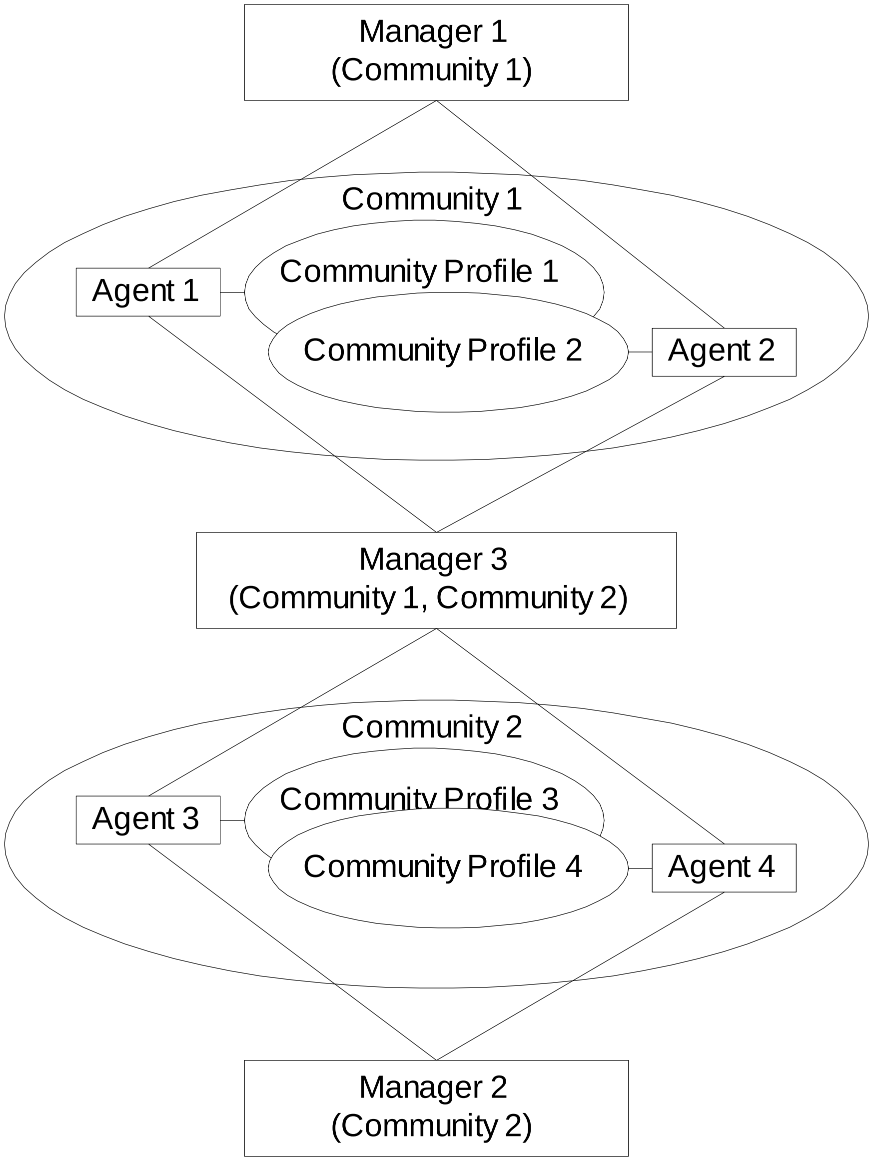


Figure 5.3 SNMP Access Policy

Notes

- Manager 1 manages community 1, manager 2 community 2,and manager 3 (MoM) both communities 1 and 2

Proxy Access Policy

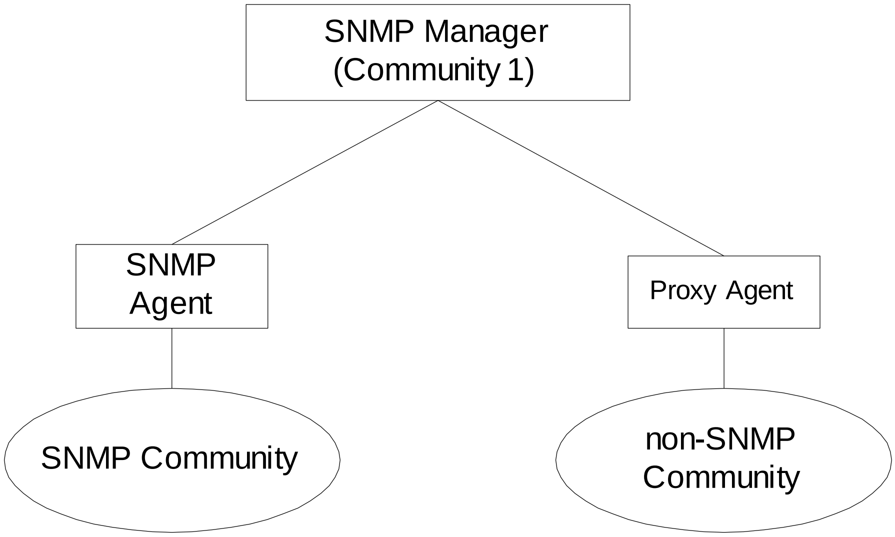


Figure 5.4 SNMP Proxy Access Policy

Notes

- Proxy agent enables non-SNMP community elements to be managed by an SNMP manager.
- An SNMP MIB is created to handle the non-SNMP objects

Protocol Entities

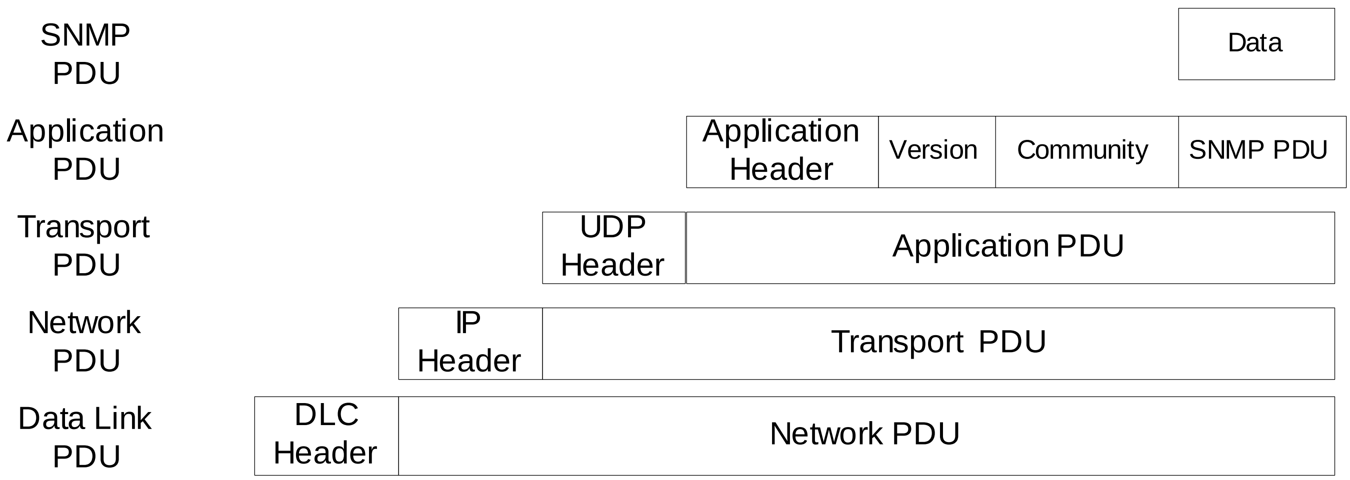


Figure 5.5 Encapsulated SNMP Message

Notes

- Protocol entities support application entities
- Communication between remote peer processes
- Message consists of
    - Version identifier
    - Community name
    - Protocol Data Unit
- Message encapsulated and transmitted

Get and Set PDU

PDU Type	RequestID	Error Status	Error Index	VarBind 1 name	VarBind 1 value	...	VarBind n name	VarBind n value

Figure 5.8 Get and Set Type PDUs

Notes

- VarBindList: multiple instances of VarBind pairs

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

PDU Types: enumerated INTEGER

```
get-request         [0]
get-next-request    [1]
set-request         [2]
get-response        [3]
trap                [4]
```

Error in Response

```
ErrorStatus ::=
    INTEGER {
        noError(0)
        tooBig(1)
        noSuchName(2)
        bad value(3)
        readOnly(4)
        genErr(5)
    }
```

Error Index: No. of VarBind that the first error occurred

Notes

Trap PDU

PDU Type	Enterprise	Agent Address	Generic Trap Type	Specific Trap Type	Timestamp	VarBind 1 name	VarBind 1 value	...	VarBind n name	VarBind n value

Generic Trap Type	Description (brief)
coldStart(0)	Sending protocol entity is reinitializing itself; agent's configuration or protocol entity implementation may be altered
warmStart(1)	Sending protocol entity is reinitializing itself; agent configuration or protocol entity implementation not altered
linkDown(2)	Failure of one of the communication links
linkUp(3)	One of the links has come up
authenticationFailure(4)	Authentication failure
egpNeighborLoss(5)	Loss of EGP neighbor
enterpriseSpecific(6)	Enterprise-specific trap

Notes

- Enterprise and agent address pertain to the system generating the trap
- Seven generic traps specified by enumerated INTEGER
- Specific trap is a trap not covered by enterprise specific trap
- time stamp indicates elapsed time since last re-initialization

SNMP Operations

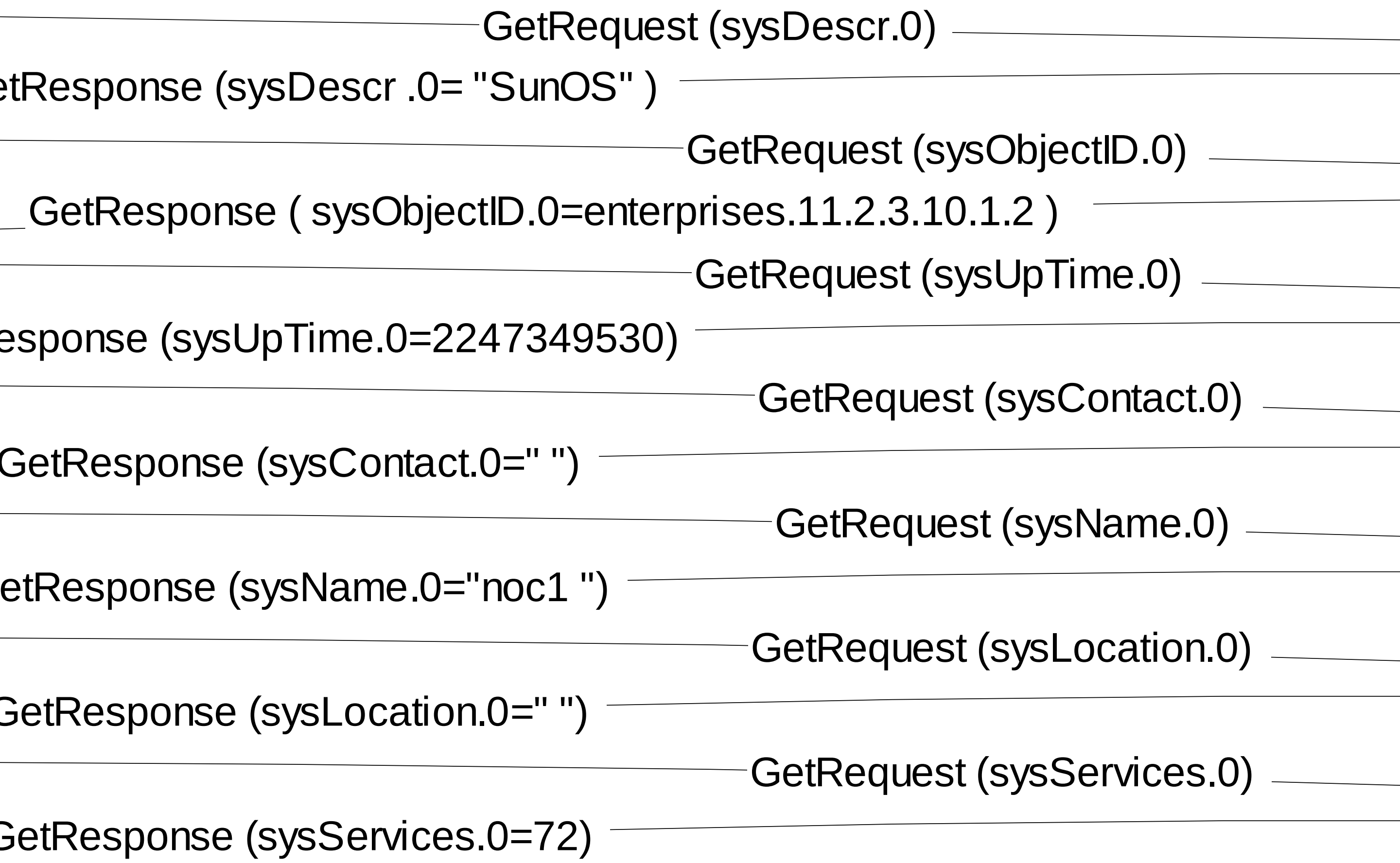


Figure 5.10 Get-Request Operation for System Group

MIB for Get-Next-Request

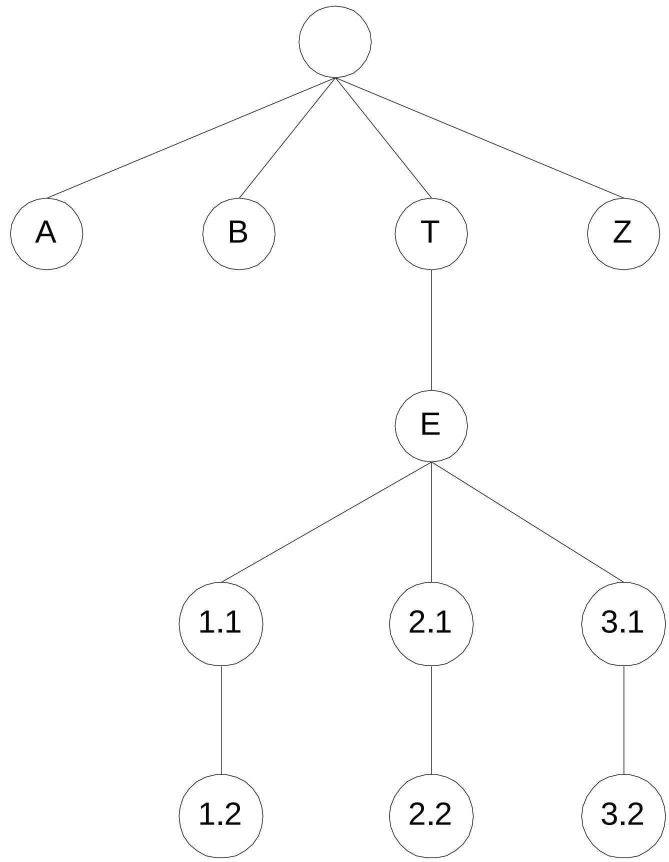


Figure 5.12 MIB for Operation Sequences in Figures 5.13 and 5.15

Notes

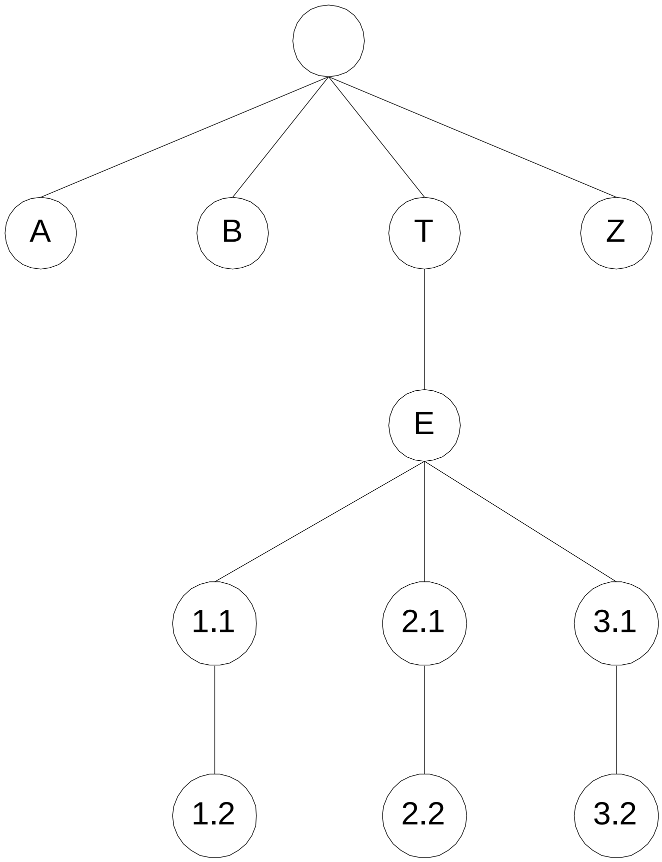
Lexicographic Order

Numerical Order	Lexicographic order
1	1
2	1118
3	115
9	126
15	15
22	2
34	22
115	250
126	2509
250	3
321	321
1118	34
2509	9

Notes

- Procedure for ordering:
  - Start with leftmost digit as first position
  - Before increasing the order in the first position, select the lowest digit in the second position
  - Continue the process till the lowest digit in the last position is captured
  - Increase the order in the last position until all the digits in the last position are captured
  - Move back to the last but one position and repeat the process
  - Continue advancing to the first position until all the numbers are ordered
- Tree structure for the above process

MIB Lexicographic Order



Notes

A	3.1
B	3.2
T	Z
E	
1.1	
1.2	
2.1	
2.2	

A More Complex MIB Example

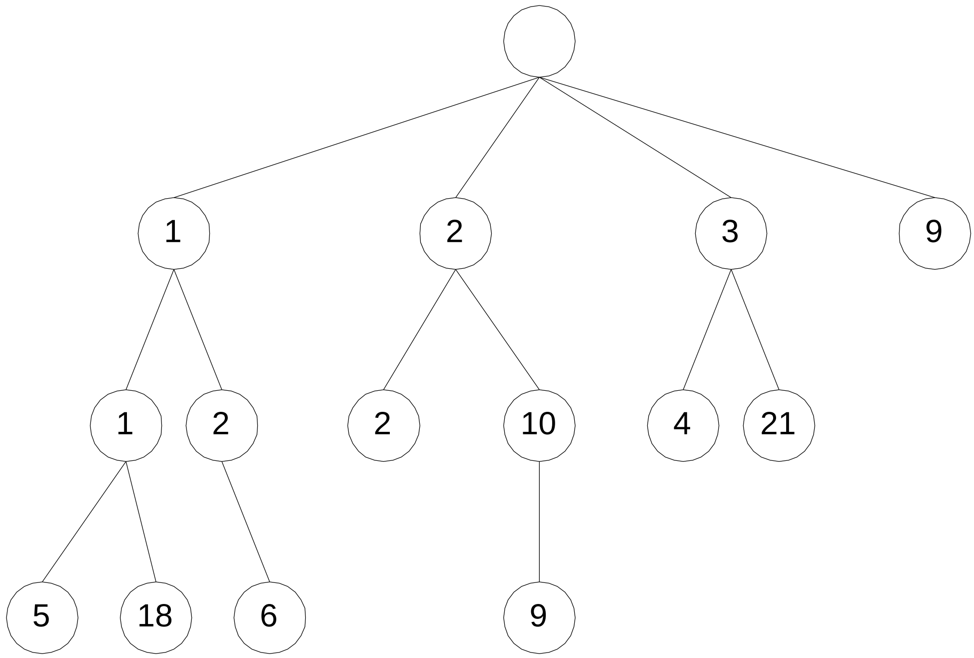


Figure 5.14 MIB Example for Lexicographic Ordering

Notes

1
1.1
1.1.5
1.1.18
1.2
1.2.6
2
2.2
2.10
2.10.9
3
3.4
3.21
9

Get-Next-Request Operation

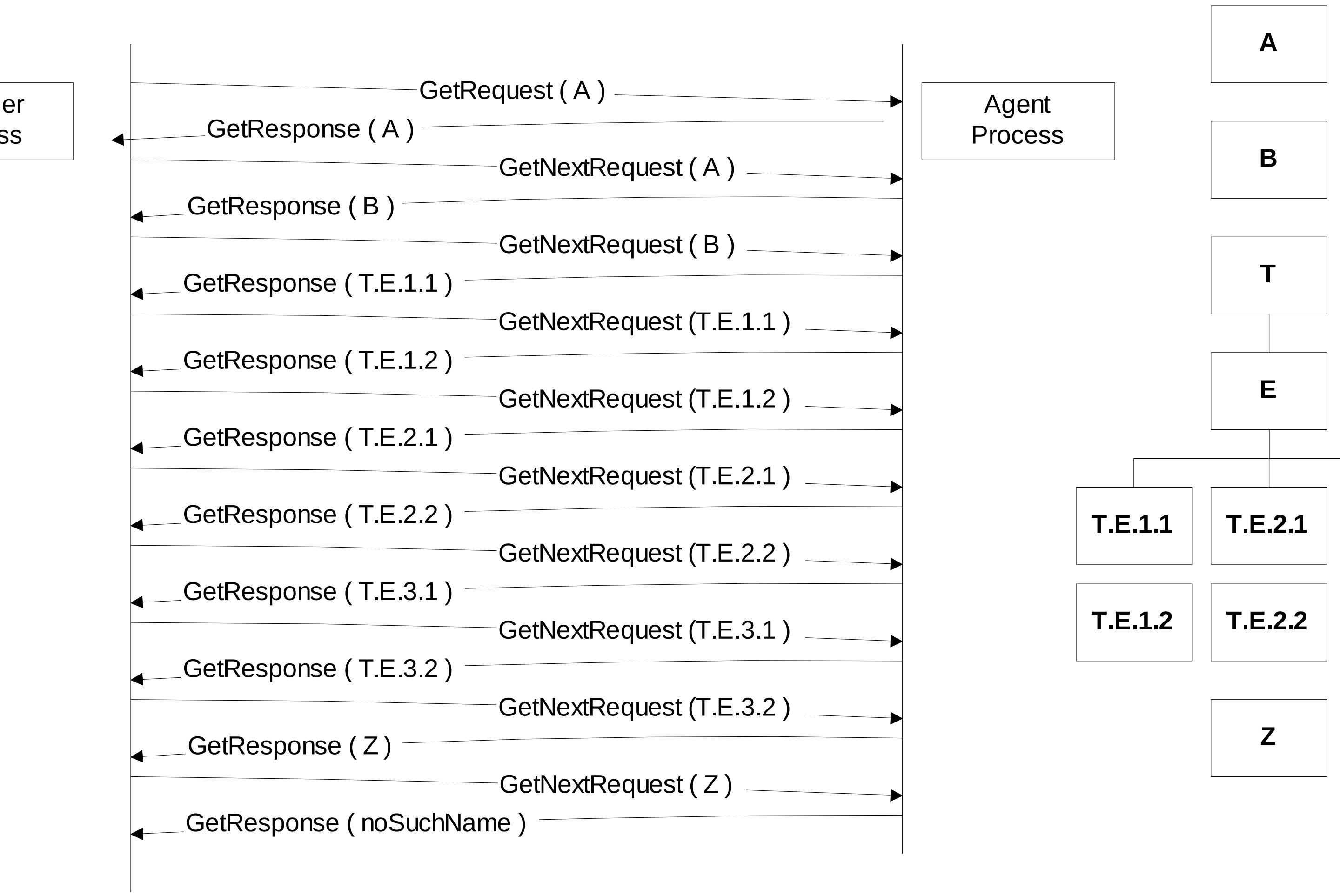
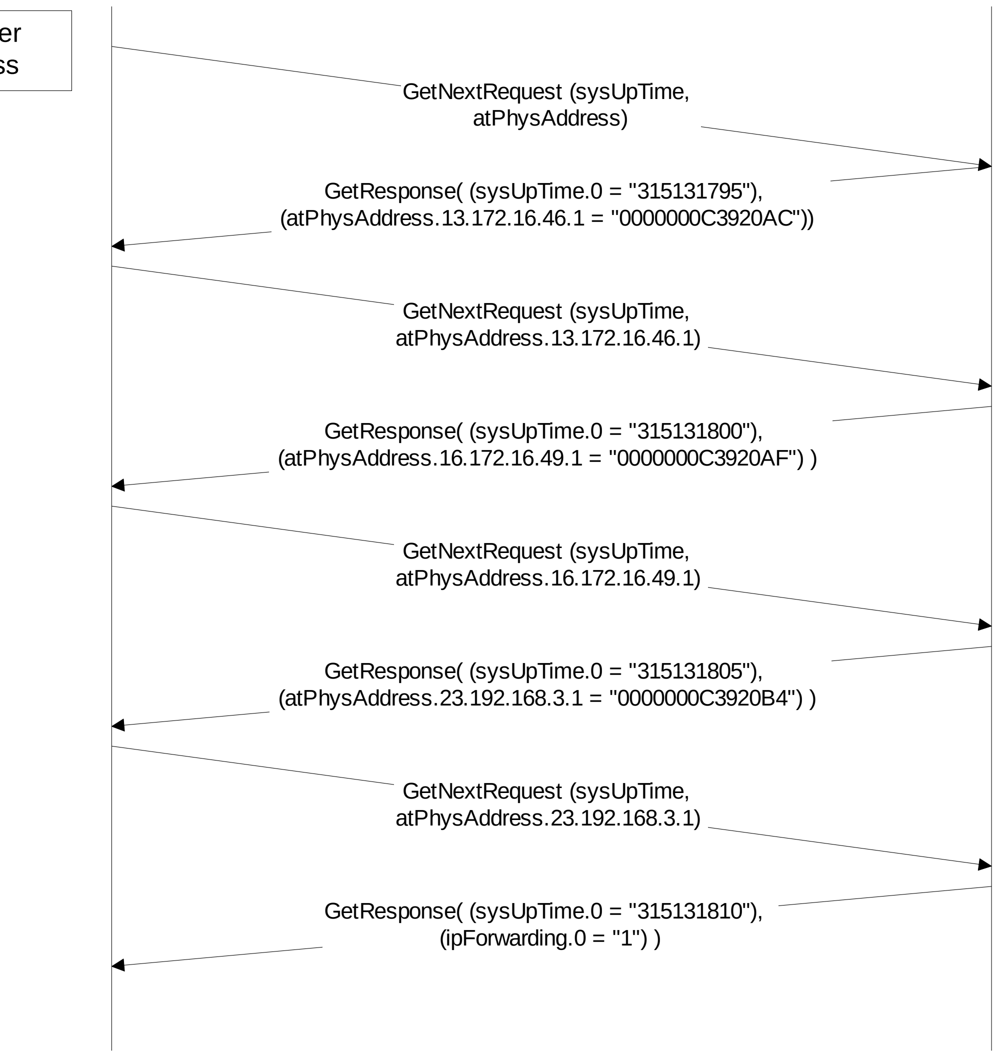


Figure 5.15 Get-Next-Request Operation for MIB in Figure 5.12

Get-Next-Request Operation



atIfIndex	atPhysAddress	**atN**
23	0000000C3920B4	**19**
13	0000000C3920AC	**17**
16	0000000C3920AF	**17**

Figure 5.16 GetNextRequest Example with Indices

Sniffer Data

14:03:36.788270 noc3.btc.gatech.edu.164 >
noc1.btc.gatech.edu.snmp:
Community = public
GetRequest(111)
Request ID = 4
system.sysDescr.0
system.sysObjectID.0
system.sysUpTime.0
system.sysContact.0
system.sysName.0
system.sysLocation.0
system.sysServices.0

Figure 5.17(a) Get-Request Message from Manager-to-Agent

14:03:36.798269 noc1.btc.gatech.edu.snmp >
noc3.btc.gatech.edu.164:
Community = public
GetResponse(196)
Request ID = 4
system.sysDescr.0 = "SunOS noc1 5.5.1 Generic_103640-08 sun4u"
system.sysObjectID.0 = E:hp.2.3.10.1.2
system.sysUpTime.0 = 247396453
system.sysContact.0 = "Brandon Rhodes"
system.sysName.0 = "noc1"
system.sysLocation.0 = "BTC NM Lab"
system.sysServices.0 = 72

Figure 5.17(b) Get-Response Message from Agent-to-Manager (After)

SNMP MIB

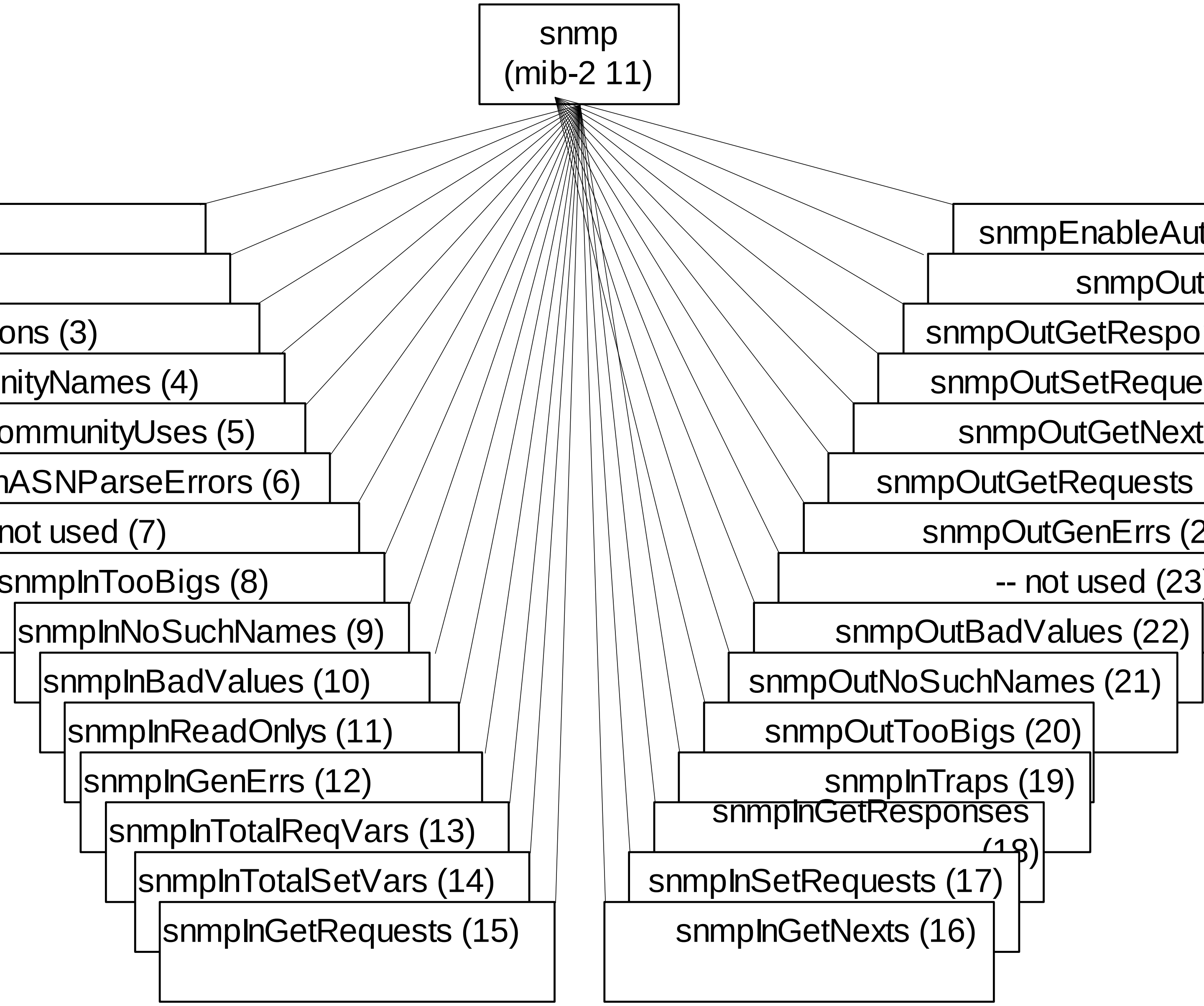


Figure 5.21 SNMP Group

Note: Most of the MIB objects were not used and hence deprecated in SNMPv2

Chapter 7

SNMPv3

Key Features

- Modularization of document
- Modularization of architecture
- SNMP engine
- Security feature
    - Secure information
    - Access control

Notes

Documentation

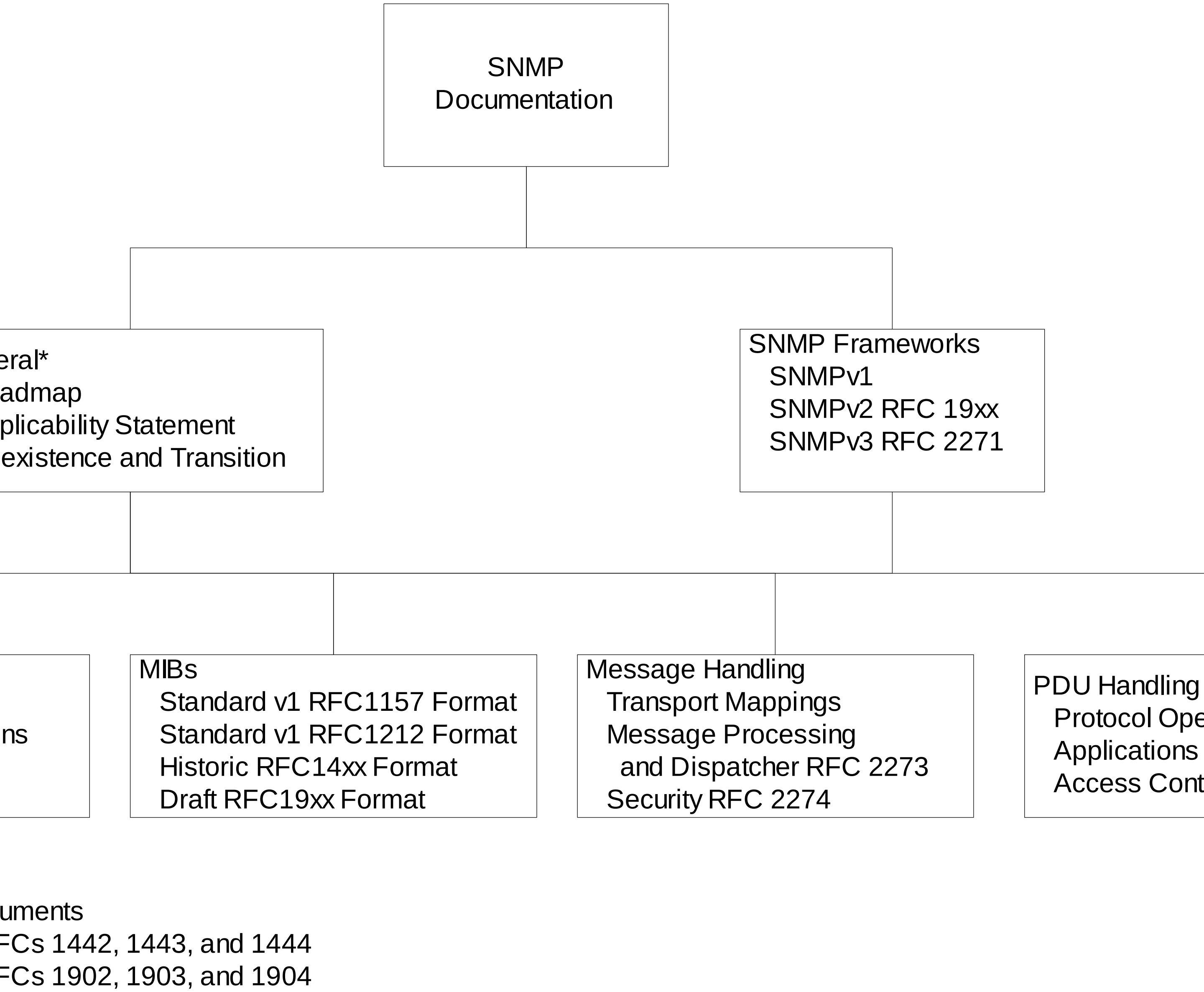


Figure 7.1 SNMP Documentation (recommended in SNMPv3)

- Compare this to the document organization in Chapter 4

Architecture

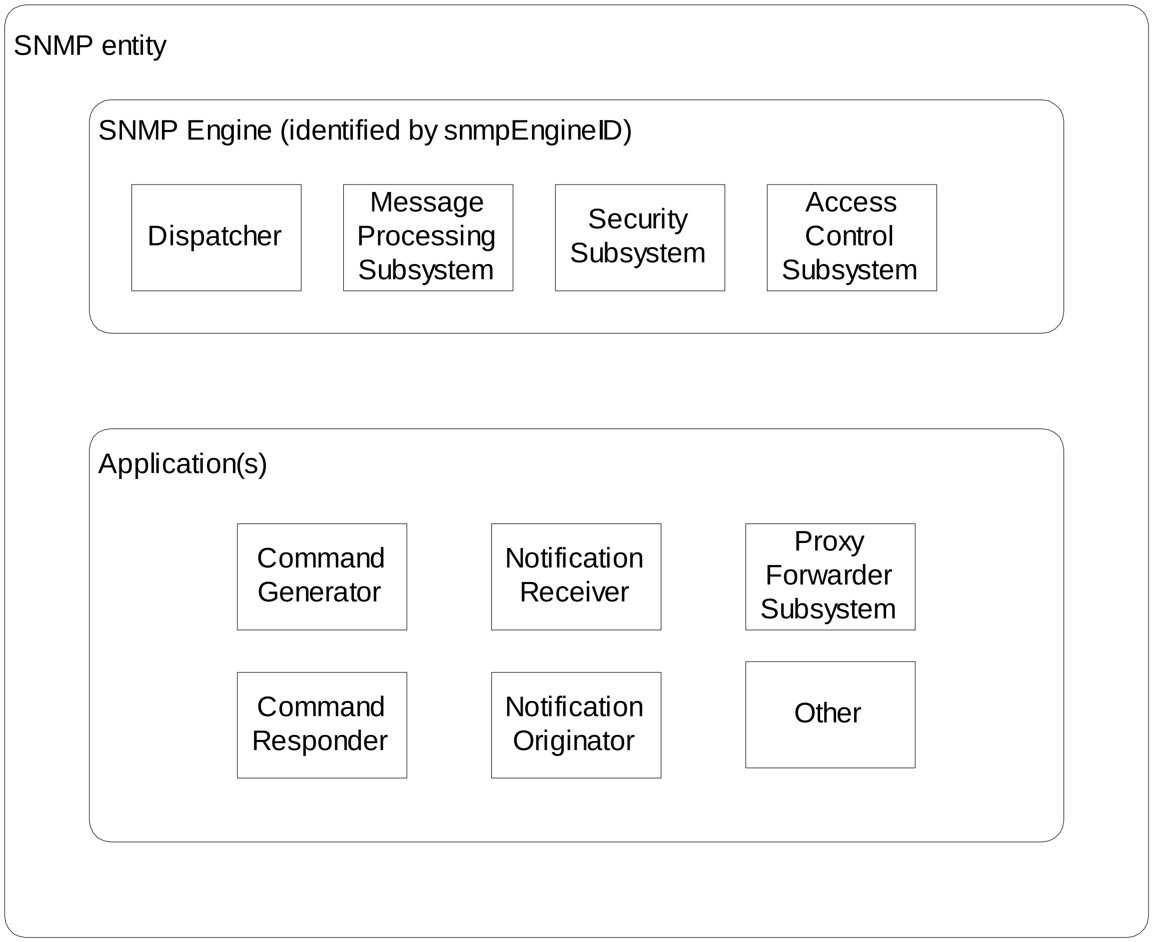


Figure 7.2 SNMPv3 Architecture

Notes

- SNMP entity is a node with an SNMP management element - either an agent or manager or both
- Three names associated with an entity
  - Entities: SNMP engine
  - Identities: Principal and security name
  - Management Information: Context engine

SNMP Engine ID

	1st bit			
SNMPv1 SNMPv2	0	Enterprise ID (1-4 octets)	Enterprise method (5th octet)	Function of the method (6-12 octets)
SNMPv3	1	Enterprise ID (1-4 octets)	Format indicator (5th octet)	Format (variable number of octets)

Figure 7.3 SNMP Engine ID

Notes

- Each SNMP engine has a unique ID: *snmpEngineID*
- Acme Networks {enterprises 696}
- SNMPv1 snmpEngineID ‘000002b8’H
- SNMPv3 snmpEngineID ‘800002b8’H
  (the 1st octet is 1000 0000)

SNMPv3 Engine ID Format 5th Octet

Table 7.2 SNMPv3 Engine ID Format (5th octet)

0	Reserved, unused
1	IPv4 address (4 octets)
2	IPv6 (16 octets) Lowest non-special IP address
3	MAC address (6 octets) Lowest IEEE MAC address, canonical order
4	Text, administratively assigned Maximum remaining length 27
5	Octets, administratively assigned Maximum remaining length 27
6-127	Reserved, unused
128-255	As defined by the enterprises Maximum remaining length 27

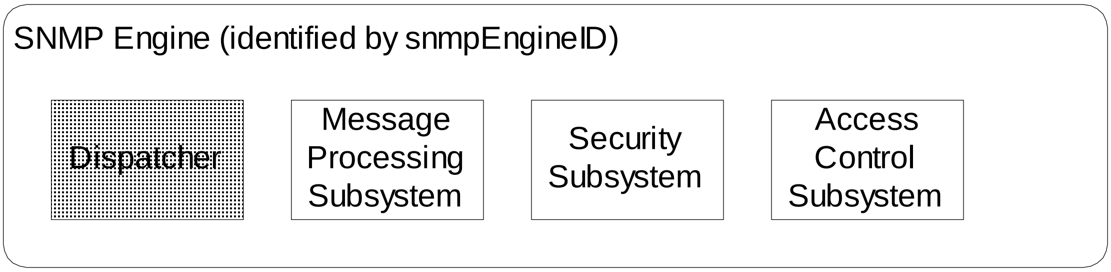
Notes

- For SNMPv1 and SNMPv2:
    - Octet 5 is the method
    - Octet 6-12 is IP address
- Examples: IBM host IP address 10.10.10.10

SNMPv1: 00 00 00 02 01 0A 0A 0A 0A 00 00 00

SNMPv3: 10 00 00 02 02 00 00 00 00 00 00 00 0A 0A 0A 0A

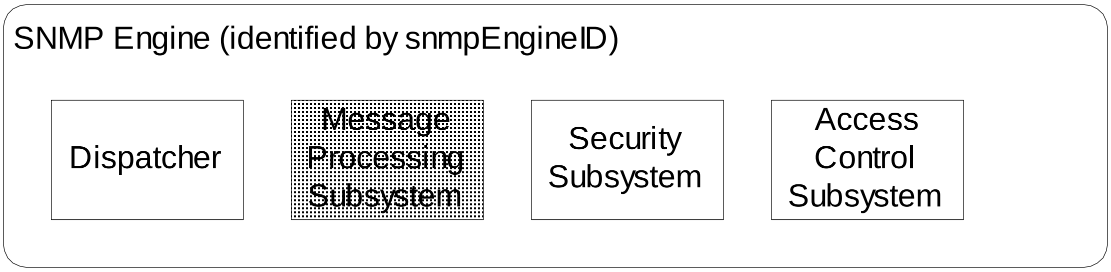
Dispatcher



- One dispatcher in an SNMP engine
- Handles multiple version messages
- Interfaces with application modules, network, and message processing models
- Three components for three functions
    - Transport mapper delivers messages over the transport protocol
    - Message Dispatcher routes messages between network and appropriate module of MPS
    - PDU dispatcher handles messages between application and MSP

Notes

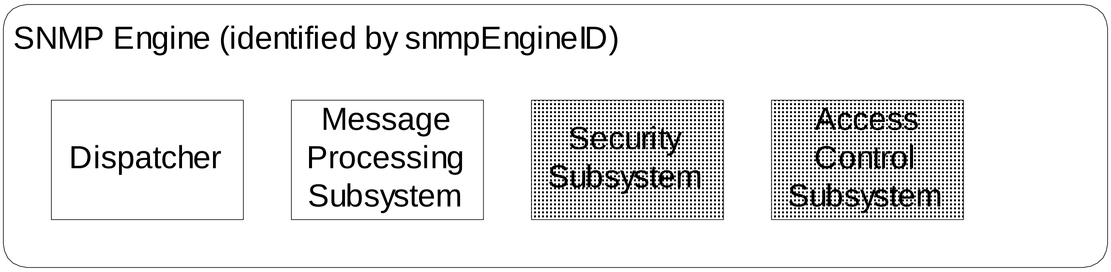
Message Processing Subsystem



- Contains one or more Message Processing Models
- One MPM for each SNMP version
- SNMP version identified in the header

Notes

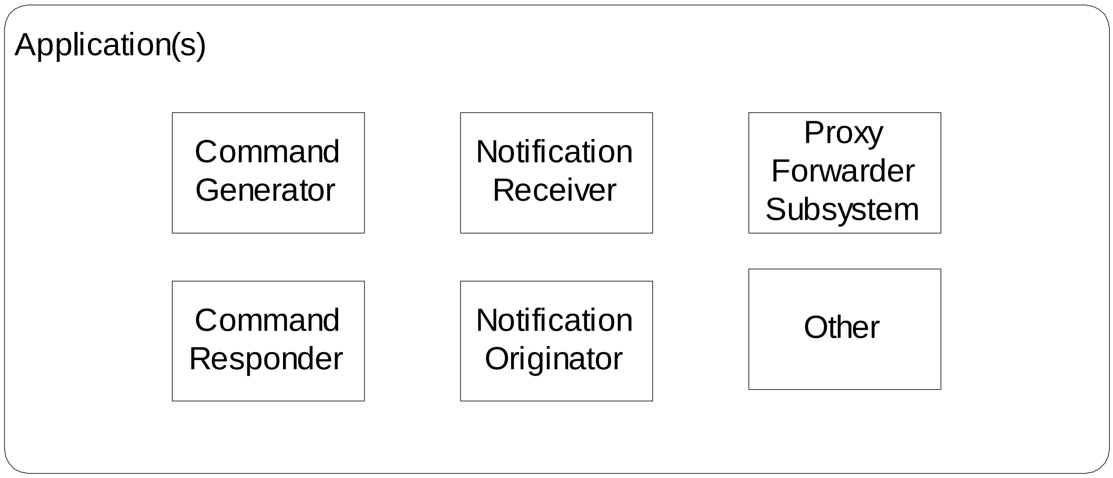
Security and Access Control



- Security at the message level
  - Authentication
  - Privacy of message via secure communication
- Flexible access control
  - Who can access
  - What can be accessed
  - Flexible MIB views

Notes

Applications



Application	Example
• Command generator	get-request
• Command responder	get-response
• Notification receiver	trap generation
• Notification receiver	trap processing
• Proxy Forwarder (SNMP versions only)	get-bulk to get-next
• Other	Special application

Notes

Names

• SNMP Engine ID	snmpEngineID
• Principal	principal
Who: person or group or application	
• Security Name	securityName
human readable name	
• Context Engine ID	contextEngineID
• Context Name	contextName

Notes

- An SNMP agent can monitor more than one network element (context)

Examples:

SNMP Engine ID	IP address
Principal	John Smith
Security Name	Administrator
Principal	Li, David, Kristen, Rashmi,
Security Name	Operator

Abstract Service Interface

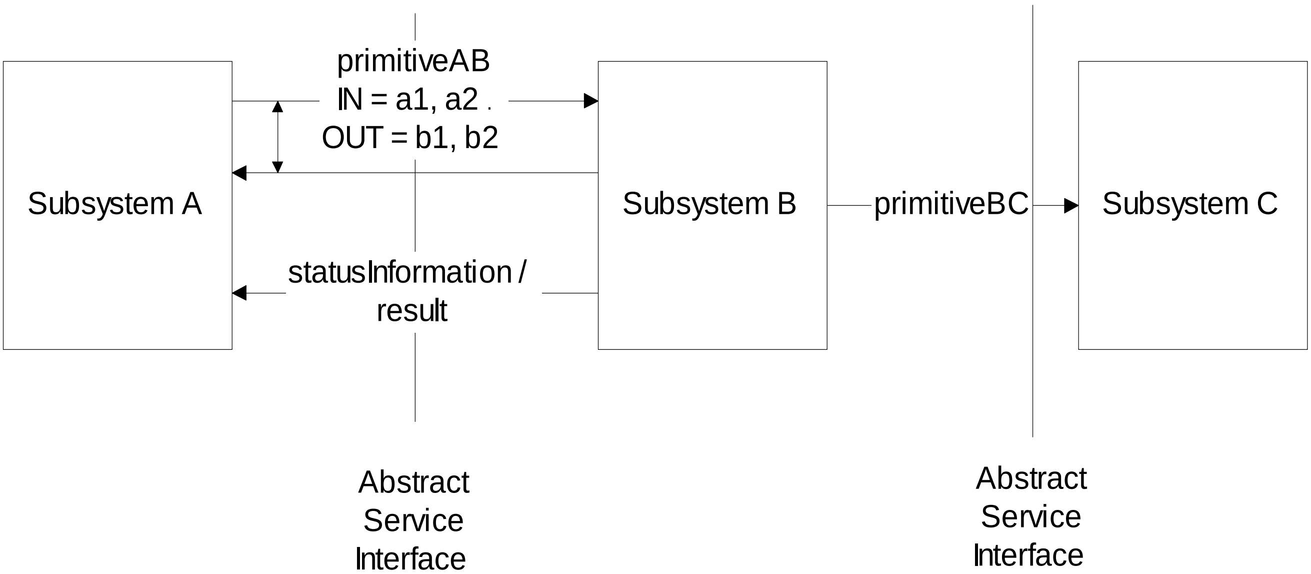


Figure 7.4(a) Abstract Service Interface

Notes

- Abstract service interface is a conceptual interface between modules, independent of implementation
- Defines a set of primitives
- Primitives associated with receiving entities except for Dispatcher
- Dispatcher primitives associated with
  - messages to and from applications
  - registering and un-registering of application modules
  - transmitting to and receiving messages from network
- IN and OUT parameters
- Status information / result

sendPDU Primitive

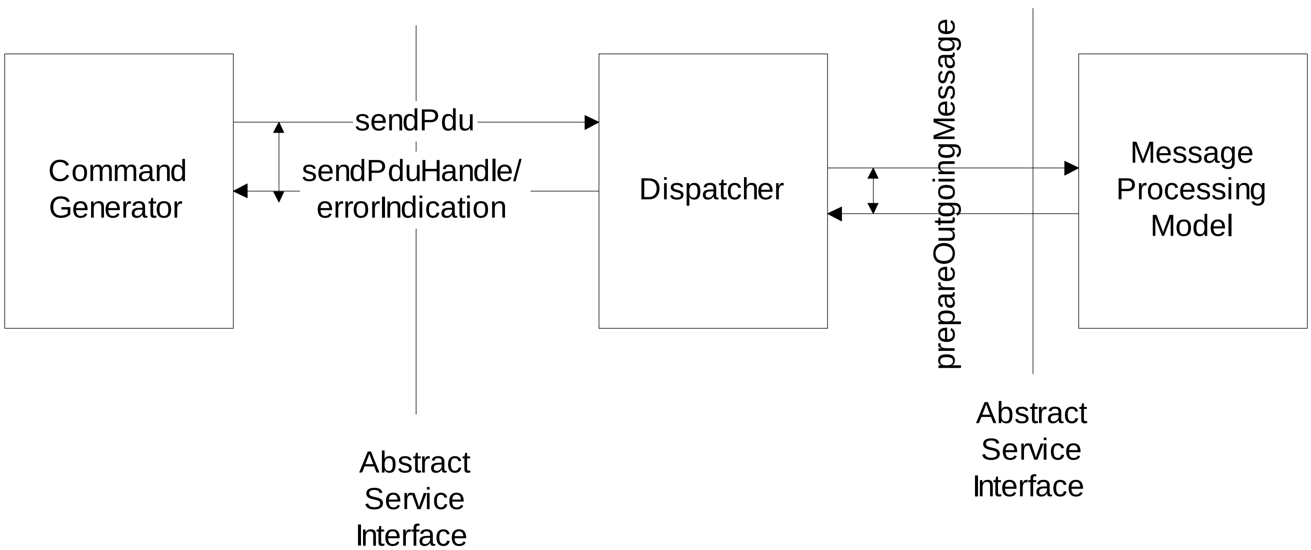


Figure 7.4(b) Abstract Service Interface for sendPdu

Notes

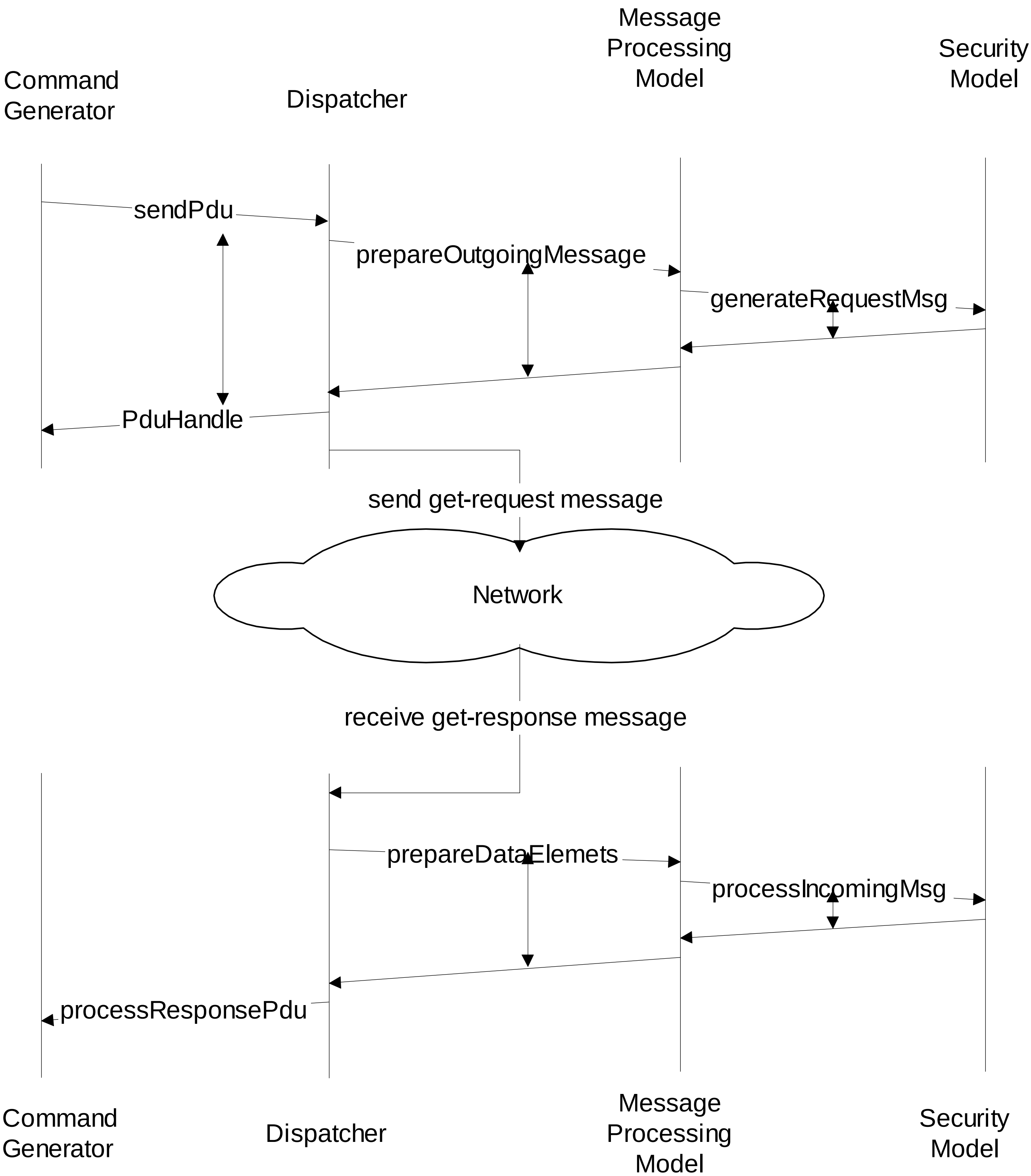
- sendPdu request sent by the application module, command generator, is associated with the receiving module, dispatcher
- After the message is transmitted over the network, dispatcher sends a handle to the command generator for tracking the response
- sendPdu is the IN parameter
- sendPduHandle is the OUT parameter, shown as coupled to the IN parameter

Dispatcher Primitives

Module	Primitive	Service Provided
Dispatcher	sendPdu	Request from application to send a PDU to a remote entity
Dispatcher	processPdu	Processing of incoming message from remote entity
Dispatcher	returnResponsePdu	Request from application to send a response PDU
Dispatcher	processResponsePdu	Processing of incoming response from a remote entity
Dispatcher	registerContextEngineID	Register request from a Context Engine
Dispatcher	unregisterContextEngineID	Unregister request from a Context Engine

Notes

Command Generator



Command Responder

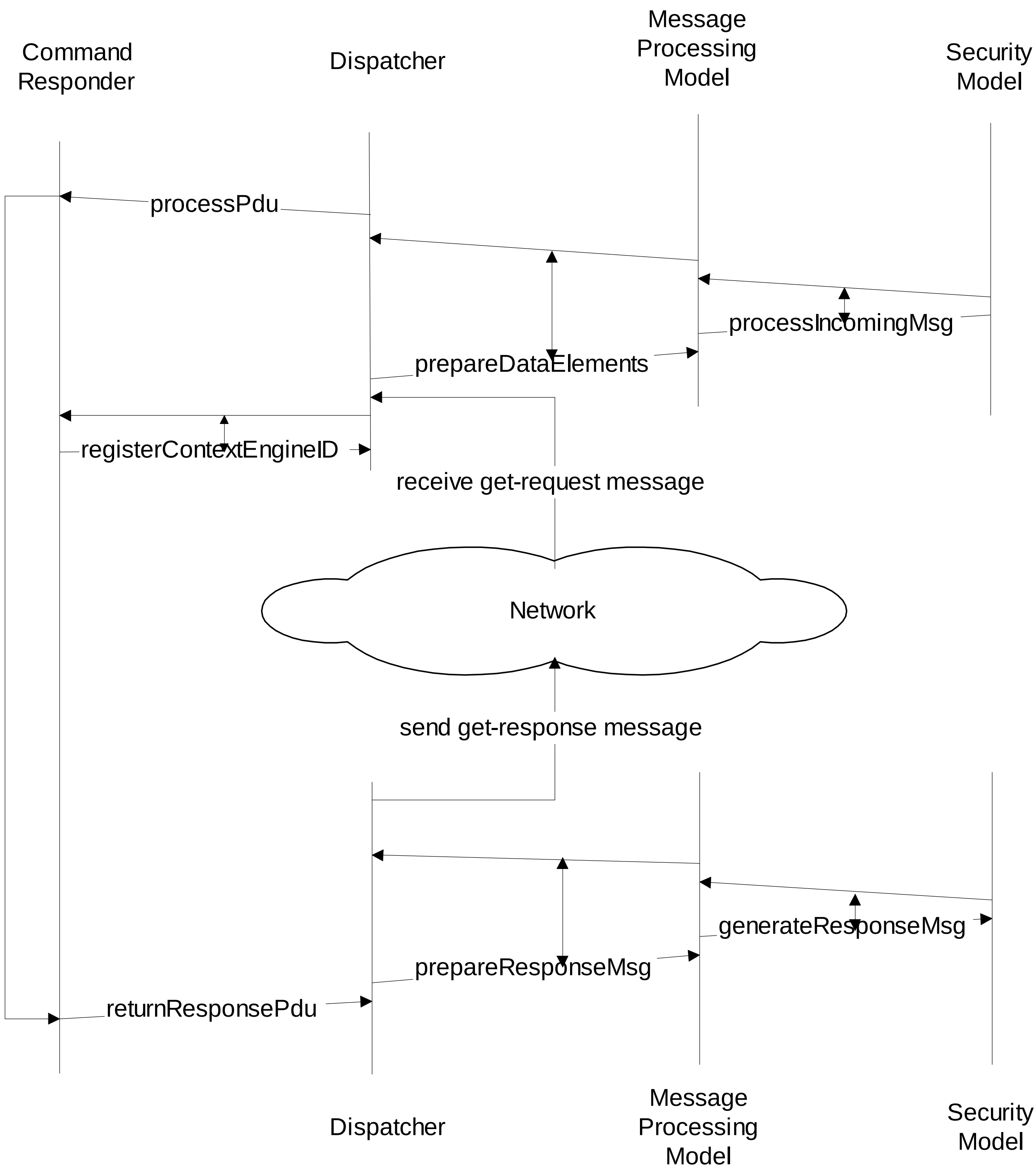


Figure 7.6 Command Responder Application

Notification / Proxy

- Notification originator
  - Generates trap and inform messages
  - Determine target, SNMP version, and security
  - Decides context information
- Notification receiver
  - Registers with SNMP engine
  - Receives notification messages
- Proxy forwarder
  - Proxy server
  - Handles only SNMP messages by
    - Command generator
    - Command responder
    - Notification generator
    - Report indicator
  - Uses the translation table in the proxy group MIB

Notes

SNMpV2 MIB

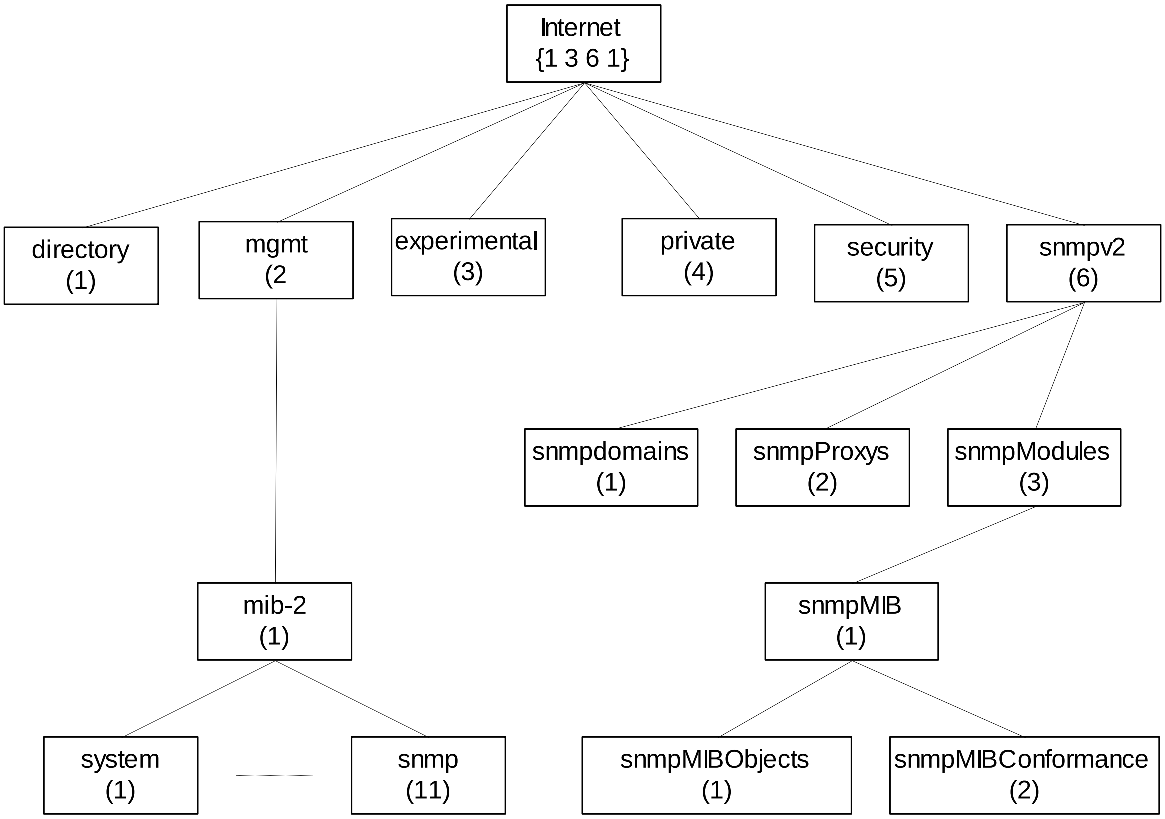


Figure 6.31 SNMPv2 Internet Group

Notes

- SNMPv3 MIB developed under snmpModules
- Security placeholder not used

SNMPv3 MIB

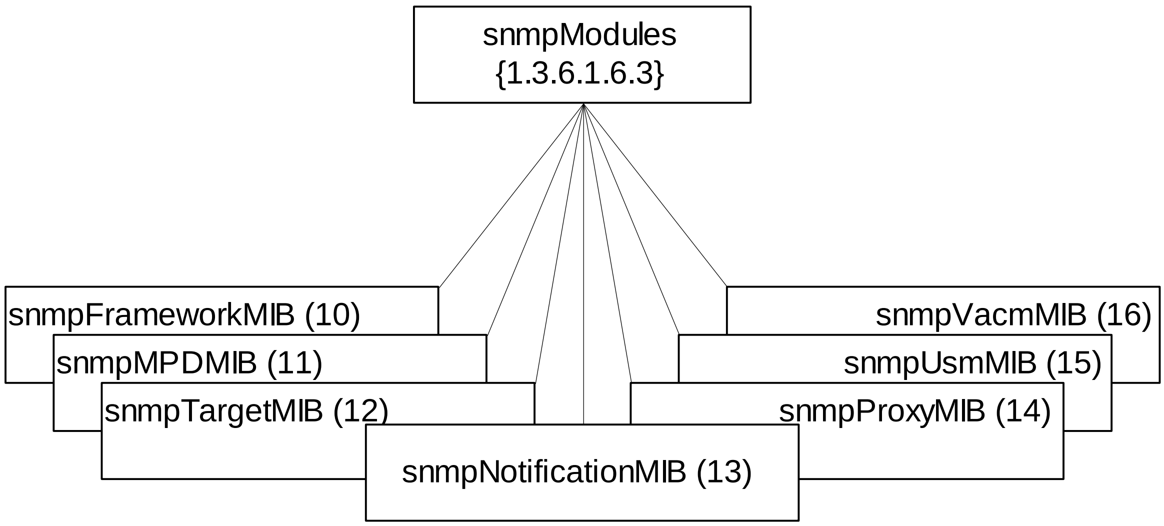


Figure 7.7 SNMPv3 MIB

Notes

- snmpFrameworkMIB describes SNMP management architecture
- snmpMPDMIB identifies objects in the message processing and dispatch subsystems
- snmpTargetMIB and snmpNotificationMIB used for notification generation
- snmpProxyMIB defines translation table for proxy forwarding
- snmpUsMIB defines user-based security model objects
- snmpVacmMIB defines objects for view-based access control

SNMPv3 Target MIB

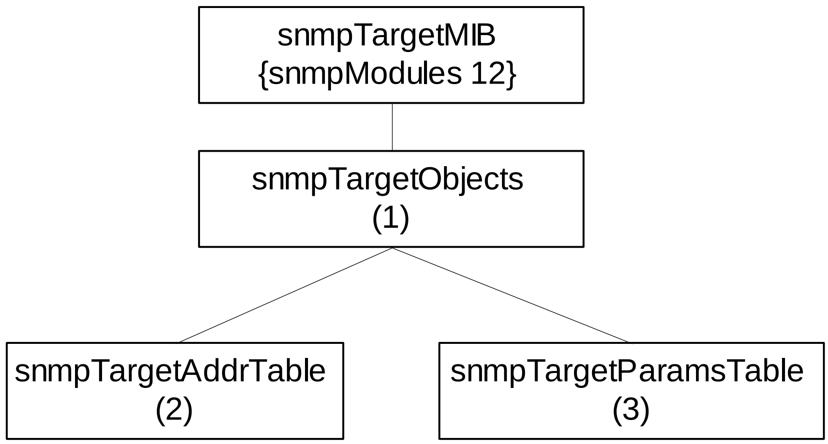


Figure 7.8 Target Address and Target Parameter Tables

Notes

- Target MIB contains two tables
- Target address table contains addresses of the targets for notifications (see notification group)
- Target address table also contains information for establishing the transport parameters
- Target address table contains reference to the second table, target parameter table
- Target parameter table contains security parameters for authentication and privacy

SNMPv3 Notification MIB

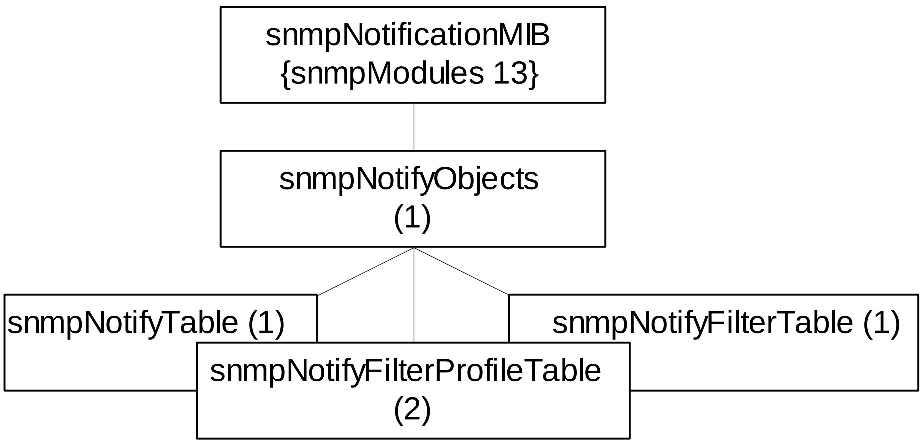


Figure 7.9 SNMP Notification Tables

Notes

- Notification group contains three tables
- Notify table contains groups of management targets to receive notifications and the type of notifications
- The target addresses to receive notifications that are listed in target address table (see target group) are tagged here
- Notification profile table defines filter profiles associated with target parameters
- Notification filter table contains table profiles of the targets

Security Threats

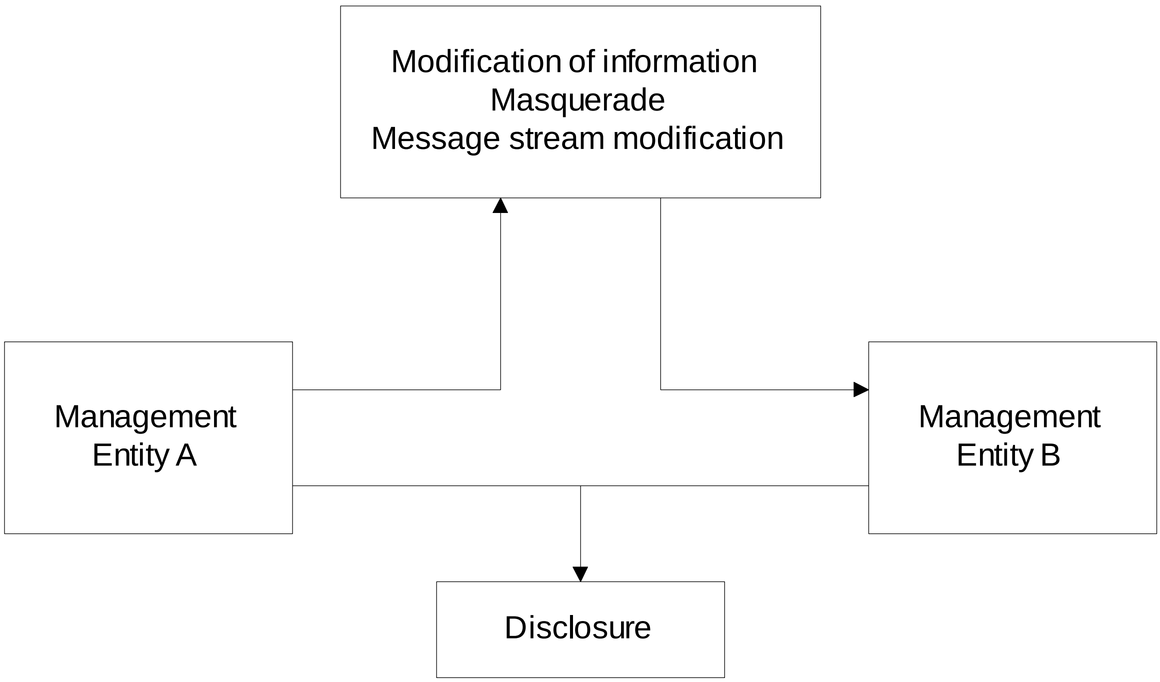


Figure 7.10 Security Threats to Management Information

Notes

- Modification of information: Contents modified by unauthorized user, does not include address change
- Masquerade: change of originating address by unauthorized user
- Fragments of message altered by an unauthorized user to modify the meaning of the message
- Disclosure is eavesdropping
- Disclosure does not require interception of message
- Denial of service and traffic analysis are not considered as threats

Security Services

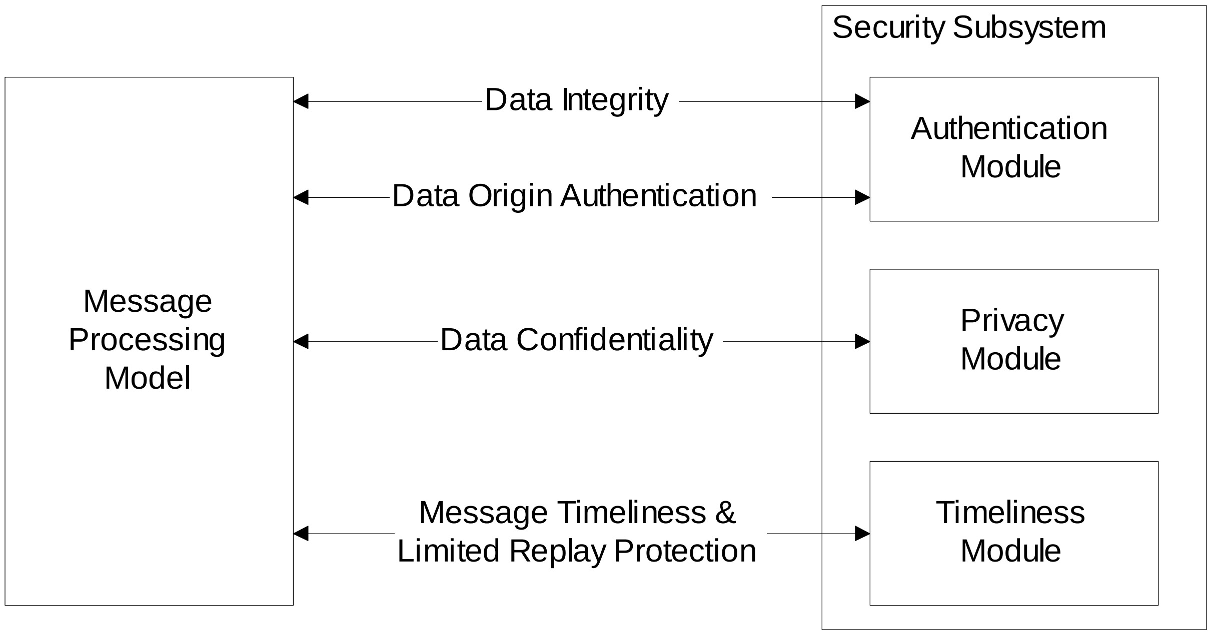
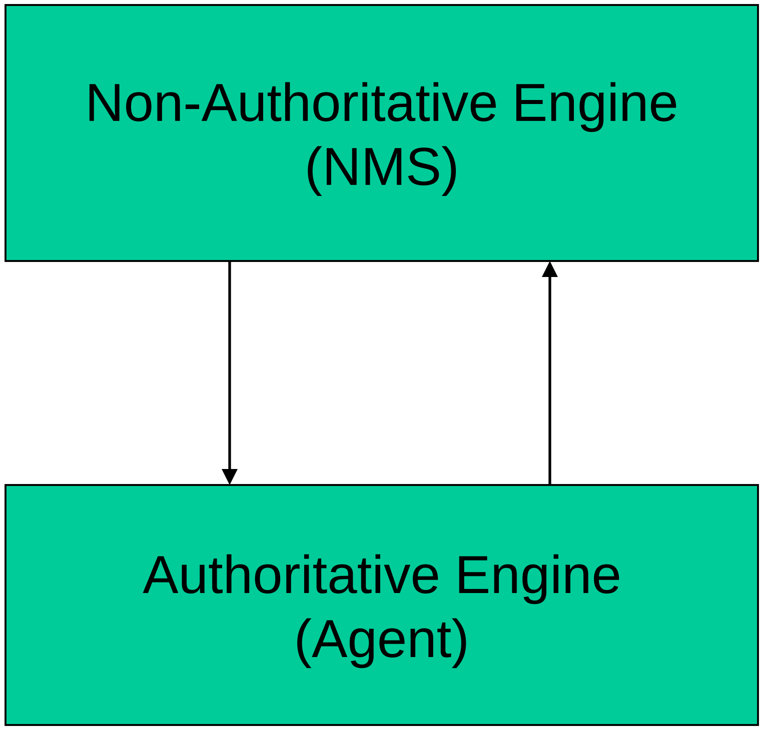


Figure 7.11 Security Services

Notes

- Authentication
  - Data integrity:
    - HMAC-MD5-96 / HMAC-SHA-96
  - Data origin authentication
    - Append to the message a unique Identifier associated with authoritative SNMP engine
- Privacy / confidentiality:
  - Encryption
- Timeliness:
  - Authoritative Engine ID, No. of engine boots and time in seconds

Role of SNMP Engines



Notes

- Responsibility of Authoritative engine:
    - Unique SNMP engine ID
    - Time-stamp
- Non-authoritative engine should keep a table of the time-stamp and authoritative engine ID

SNMPv3 Message Format

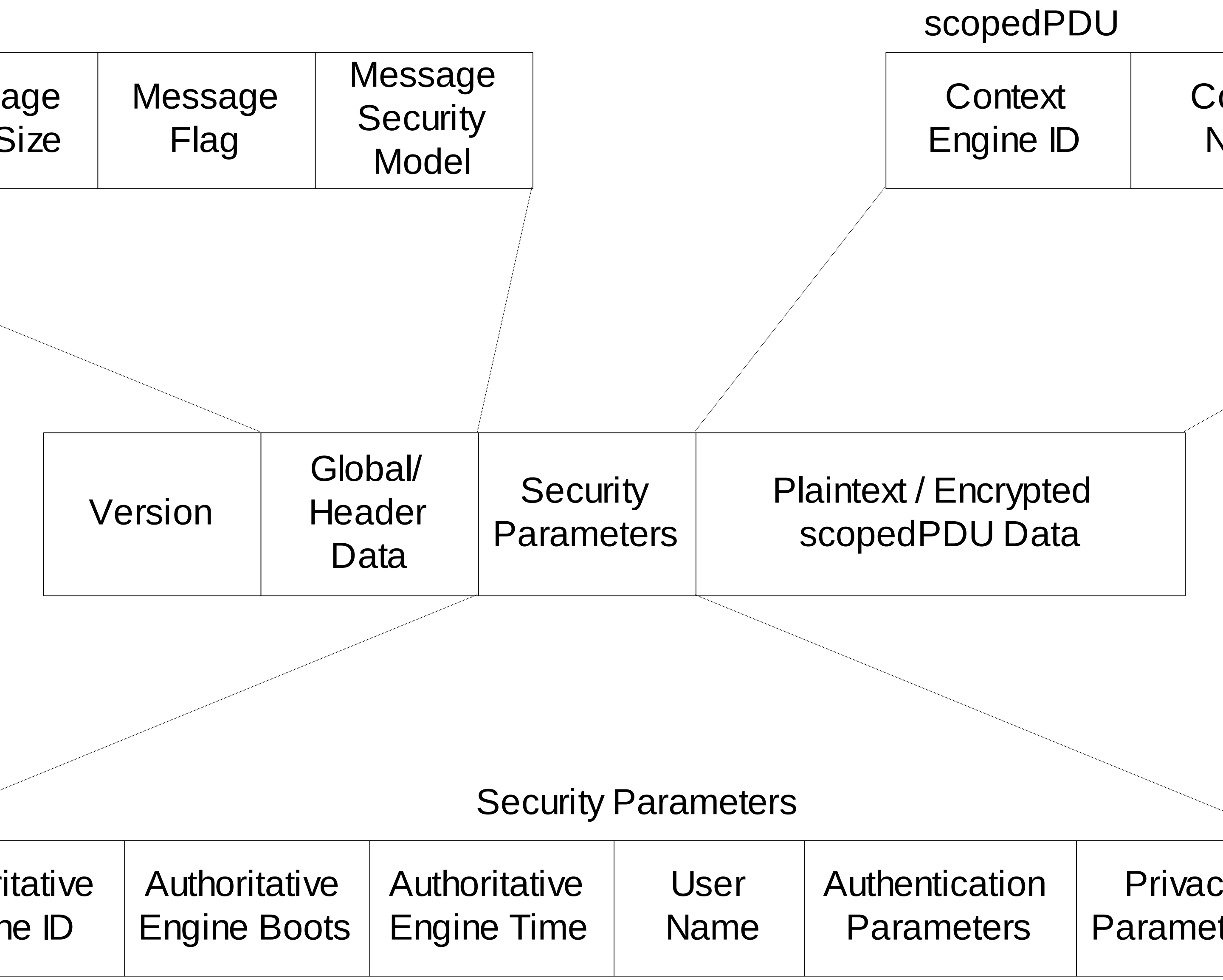


Figure 7.12 SNMPv3 Message Format

SNMPv3 Message Format

Field	Object name	Description
Version	msgVersion	SNMP version number of the message format
Message ID	msgID	Administrative ID associated with the message
Message Max. Size	msgMaxSize	Maximum size supported by the sender
Message flags	msgFlags	Bit fields identifying report, authentication, and privacy of the message
Message Security Model	msgSecurityModel	Security model used for the message; concurrent multiple models allowed
Security Parameters (See Table 7.8)	msgSecurityParameters	Security parameters used for communication between sending and receiving security modules
Plaintext/Encrypted scopedPDU Data	scopedPduData	Choice of plaintext or encrypted scopedPDU; scopedPDU uniquely identifies context and PDU
Context Engine ID	contextEngineID	Unique ID of a context (managed entity) with a context name realized by an SNMP entity
Context Name	contextName	Name of the context (managed entity)
PDU	data	Contains unencrypted PDU

User-Based Security Model

- Based on traditional user name concept
- USM primitives across abstract service interfaces
    - Authentication service primitives
        - authenticateOutgoingMsg
        - authenticateIncomingMsg
    - Privacy Services
        - encryptData
        - decryptData

Notes

Secure Outgoing Message

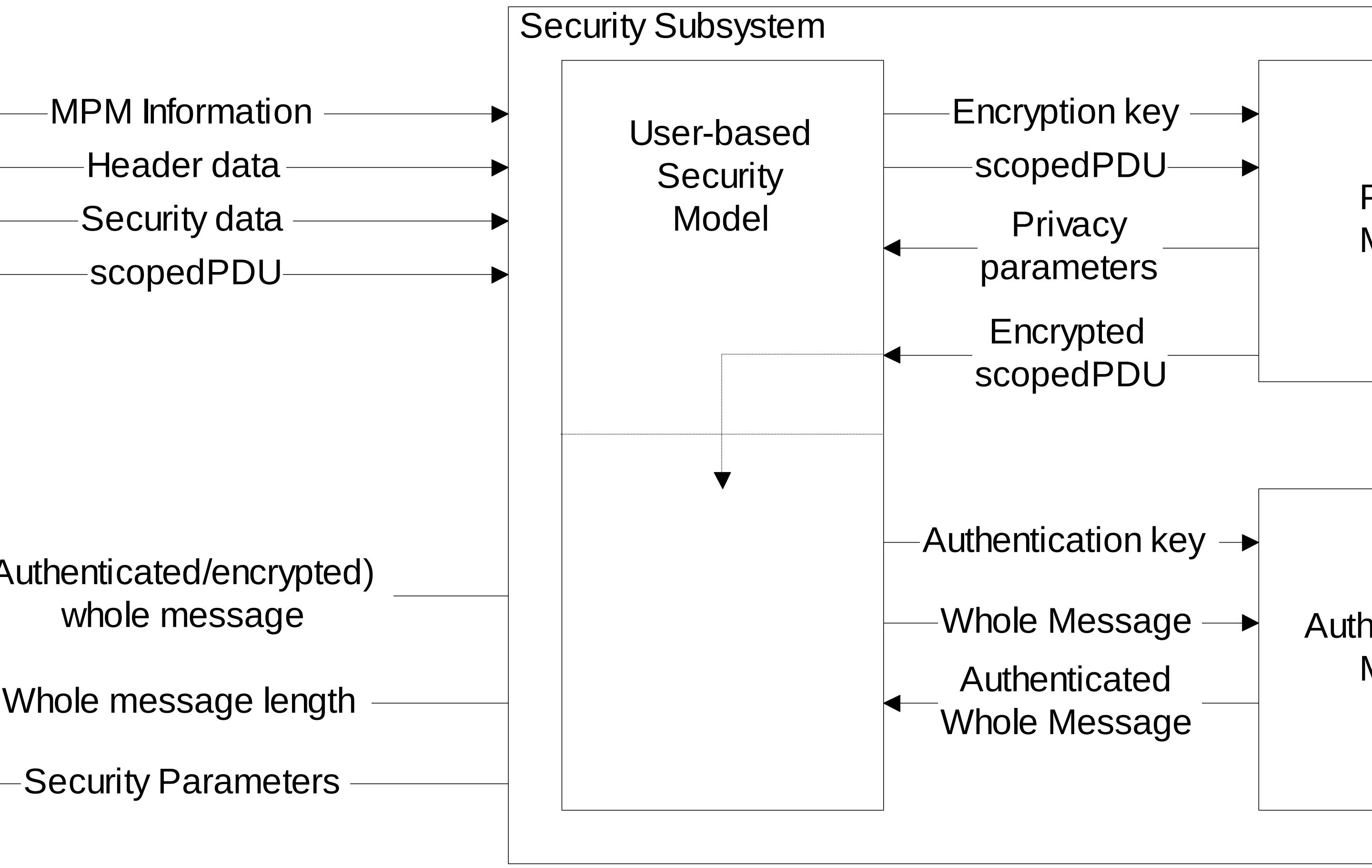


Figure 7.13 Privacy and Authentication Service for Outgoing Message

Notes

- USM invokes privacy module w/ encryption key and scopedPD
- Privacy module returns privacy parameters and encrypted sco
- USM then invokes the authentication module w/authentication whole message and receives authenticated whole message

Secure Incoming Message

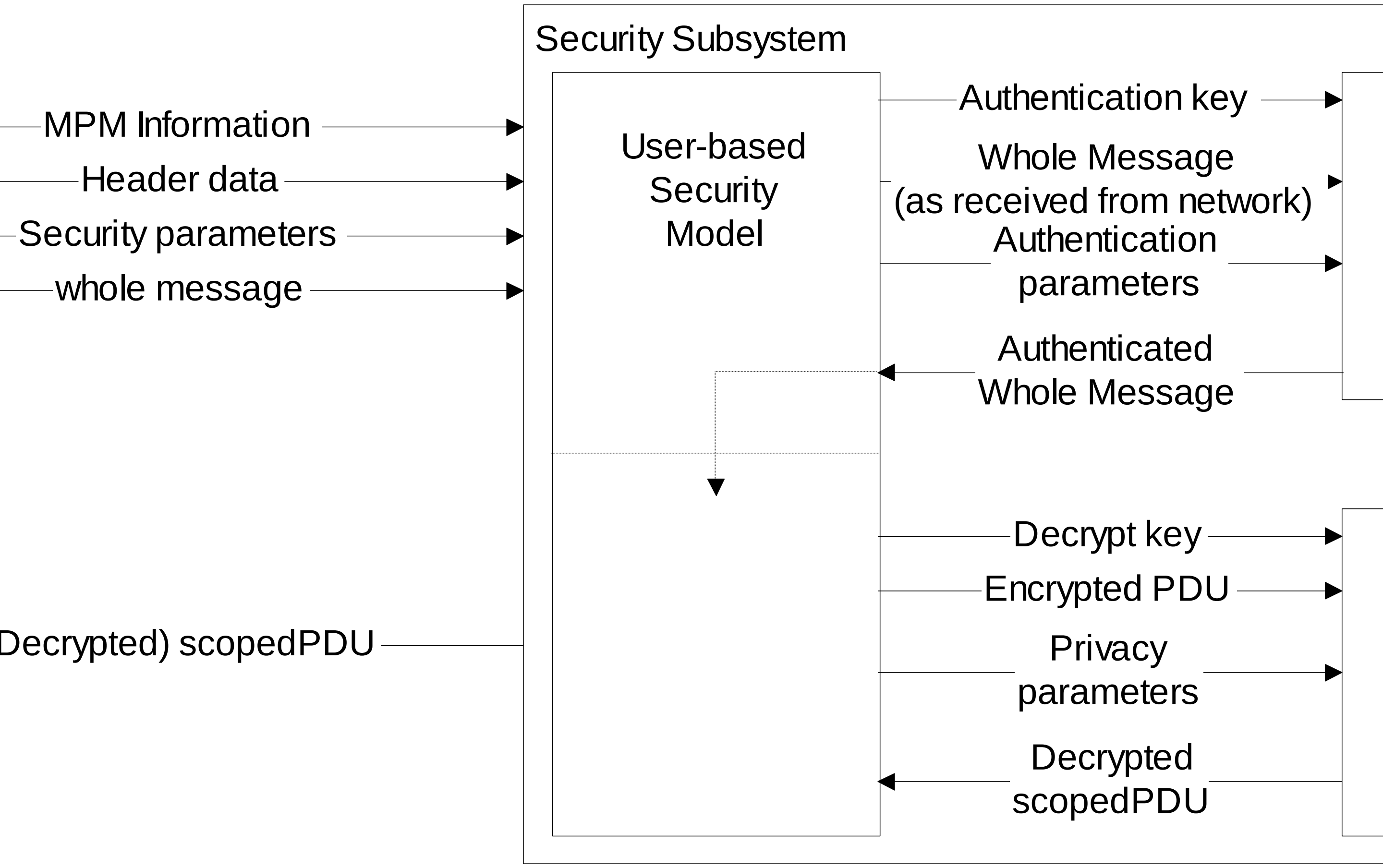


Figure 7.14 Privacy and Authentication Service for Incoming Message

Notes

- Processing secure incoming message reverse of secure
- Authentication validation done first by the authentication
- Decryption of the message done then by the privacy mod

Security Parameters

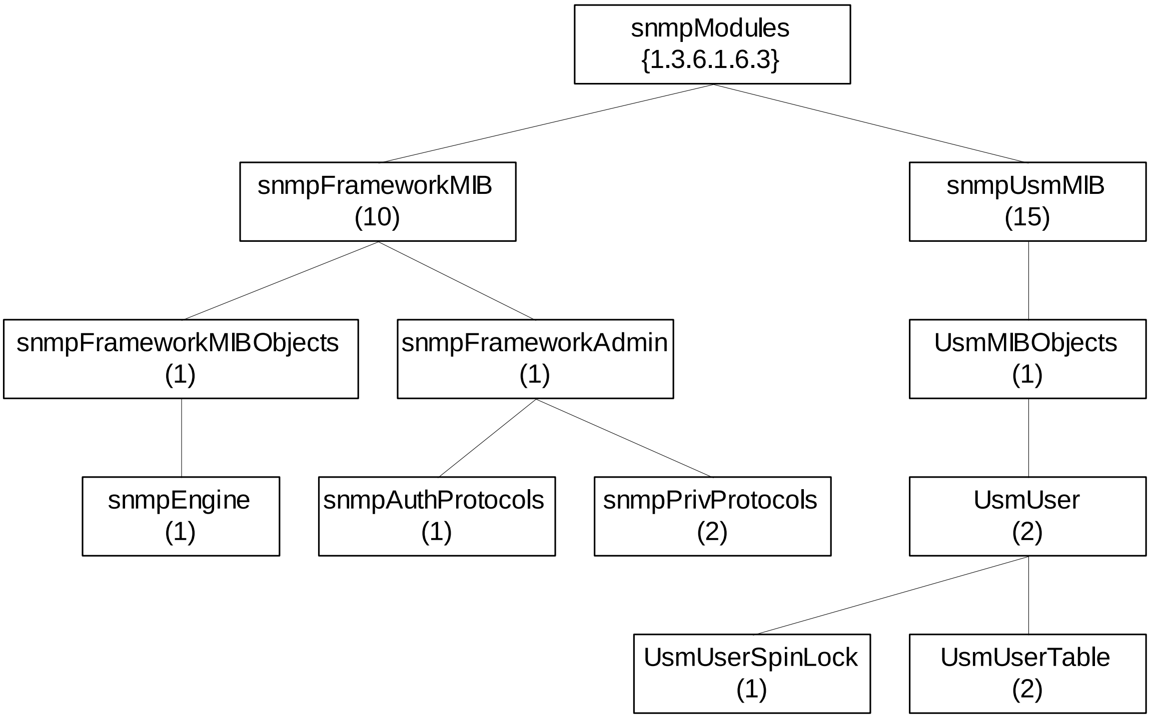


Figure 7.15 SNMPv3 MIB Objects for Security Parameters

Notes

Table 7.8 Security Parameters and Corresponding MIB Objects

Security Parameters	USM User Group Objects
msgAuthoritativeEngineID	snmpEngineID (under snmpEngine Group)
msgAuthoritativeEngineBoots	snmpEngineBoots (under snmpEngine Group)
msgAuthoritativeEngineTime	snmpEngineTime (under snmpEngine Group)
msgUserName	usmUserName (in usmUserTable)
msgAuthenticationParameters	usmUserAuthProtocol (in usmUserTable)
msgPrivacyParameters	usmUserPrivProtocol (in usmUserTable)

Privacy Module

- Encryption and decryption of scoped PDU (context engine ID, context name, and PDU)
- CBC - DES (Cipher Block Chaining - Data Encryption Standard) symmetric protocol
- Encryption key (and initialization vector) made up of secret key (user password), and timeliness value
- Privacy parameter is *salt* value (unique for each packet) in CBC-DES

Notes

Authentication Key

- Secret key for authentication
- Derived from user (NMS) password
- MD5 or SHA-1 algorithm used
- Authentication key is *digest2*

Notes

Procedure:

1. Derive *digest0:*
   Password repeated until it forms 2^{20} octets.
2. Derive *digest1:*
   Hash *digest0* using MD5 or SHA-1.
3. Derive *digest2*:
   Concatenate authoritative SNMP engine ID and *digest1* and hash with the same algorithm

Authentication Parameters

- Authentication parameter is Hashed Message Access Code (HMAC)
- HMAC is 96-bit long (12 octets)
- Derived from authorization key (*authKey*)

Notes

Procedure:

1. Derive *extendedAuthKey*:
   Supplement authKey with 0s to get 64-byte string
2. Define *ipad, opad*, K1, and K2:
   ipad = 0x36 (00110110) repeated 64 times
   opad = 0x5c (01011100) repeated 64 times
   K1 = *extendedAuthKey* XOR *ipad*
   K2 = *extendedAuthKey* XOR *opad*
3. Derive HMAC by hashing algorithm used
   HMAC = H (K2, H (K1, *wholeMsg))*

Encryption Protocol

- Cipher Block Chaining mode of Data Encryption Standard (CBC-DES) protocol
- 16-octet *privKey* is secret key
- First 8-octet of *privKey* used as 56-bit DES key; (Only 7 high-order bits of each octet used)
- Last 8-octet of *privKey* used as pre-initialization vector

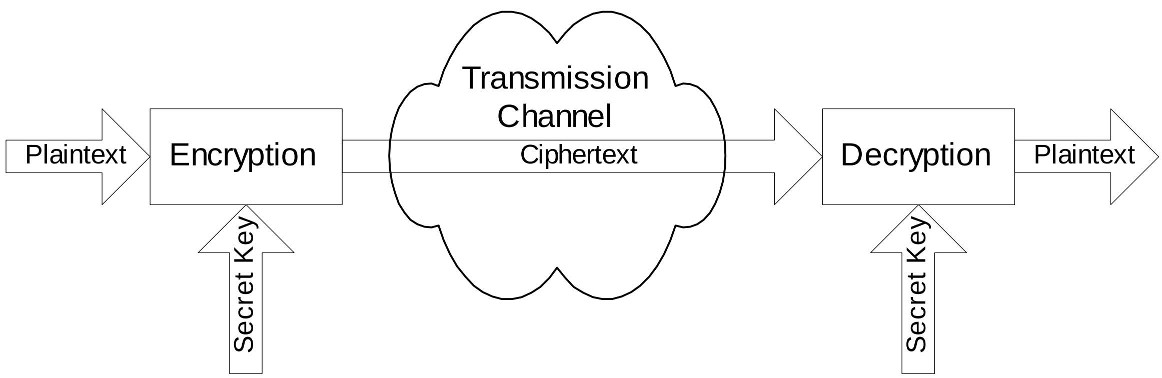


Figure 13.33 Basic Cryptographic Communication

Notes

- CBC Mode
  - Plaintext divided into 64-bit blocks
  - Each block is XOR-d with ciphertext of the previous block and then encrypted
  - Use pre-IV (initialization vector) for prefixing the first message block

Access Control

- View-based Access Control Model
  - Groups: Name of the group comprising security model and security name: In SNMPv1, is community name
  - Security Level
    - no authentication - no privacy
    - authentication - no privacy
    - authentication - privacy
  - Contexts: Names of the context
  - MIB Views and View Families
    - MIB view is a combination of view subtrees
  - Access Policy
    - read-view
    - write-view
    - notify-view
    - not-accessible

Notes

VCAM Process

Answers 6 questions:

1. Who are you (group)?
2. Where do you want to go (context)?
3. How secured are you to access the information (security model and security level)?
4. Why do you want to access the information (read, write, or send notification)?
5. What object (object type) do you want to access?
6. Which object (object instance) do you want to access?

Notes

VCAM Process

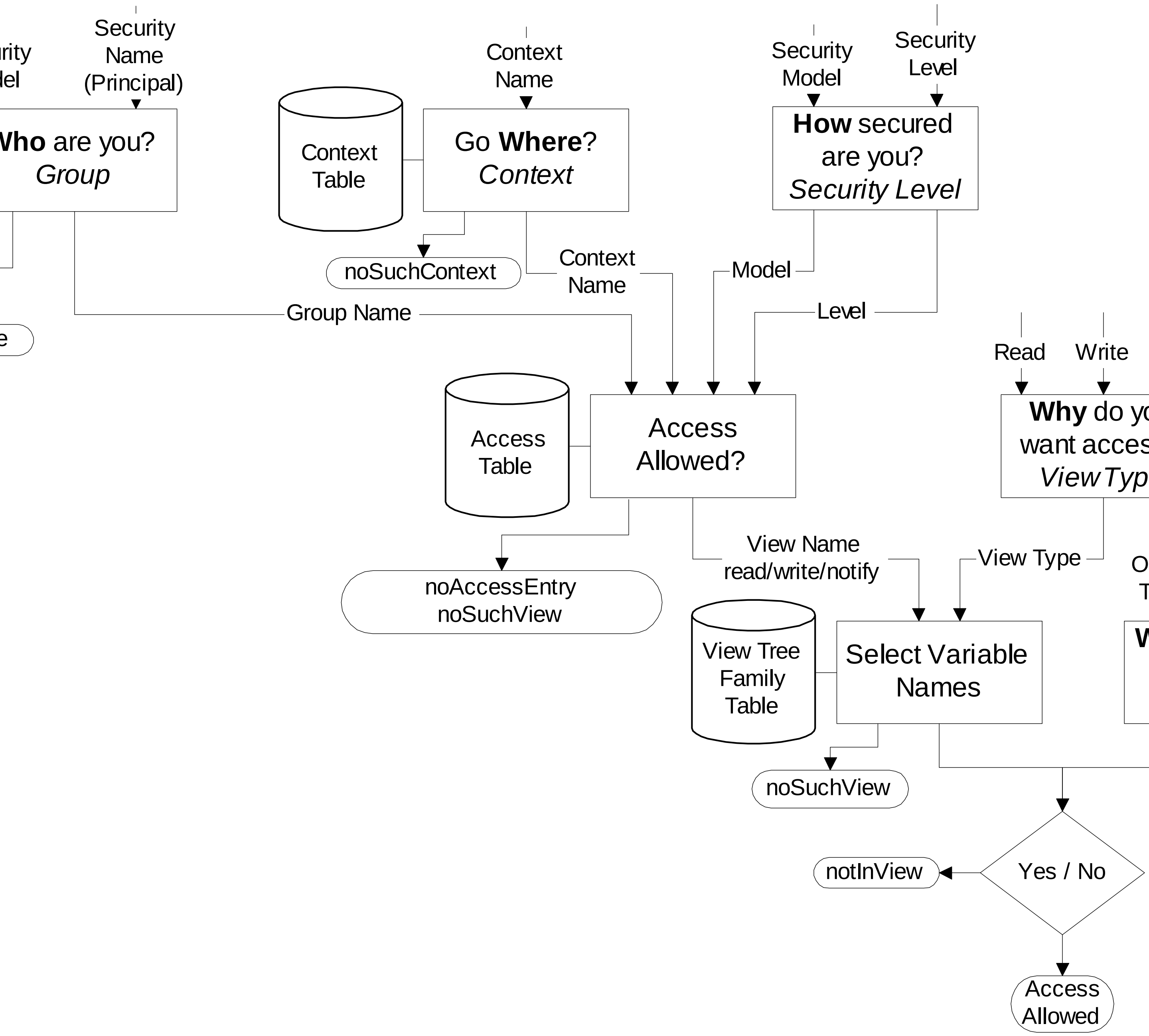


Figure 7.16 VACM Process

VACM MIB

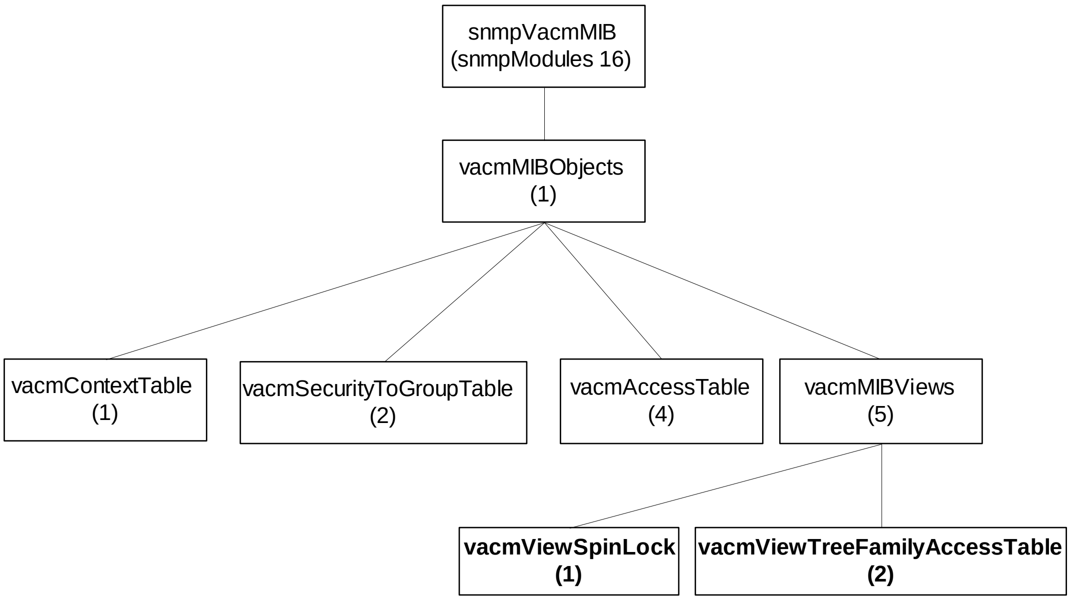


Figure 7.17 VACM MIB

Notes

- Four tables used to achieve access control
    - Group defined by security-to-group table
    - Context defined by context table
    - Access determines access allowed and the view name
    - View tree family table determines the MIB view, which is very flexible

MIB Views

Simple view:

system 1.3.6.1.2.1.1

Complex view:

All information relevant to a particular interface - *system* and *interfaces* groups

Family view subtrees

View with all columnar objects in a row appear as separate subtree.

OBJECT IDENTIFIER (family name)

paired with

bit-string value (family mask)

to select or suppress columnar objects

Notes

VACM MIB View

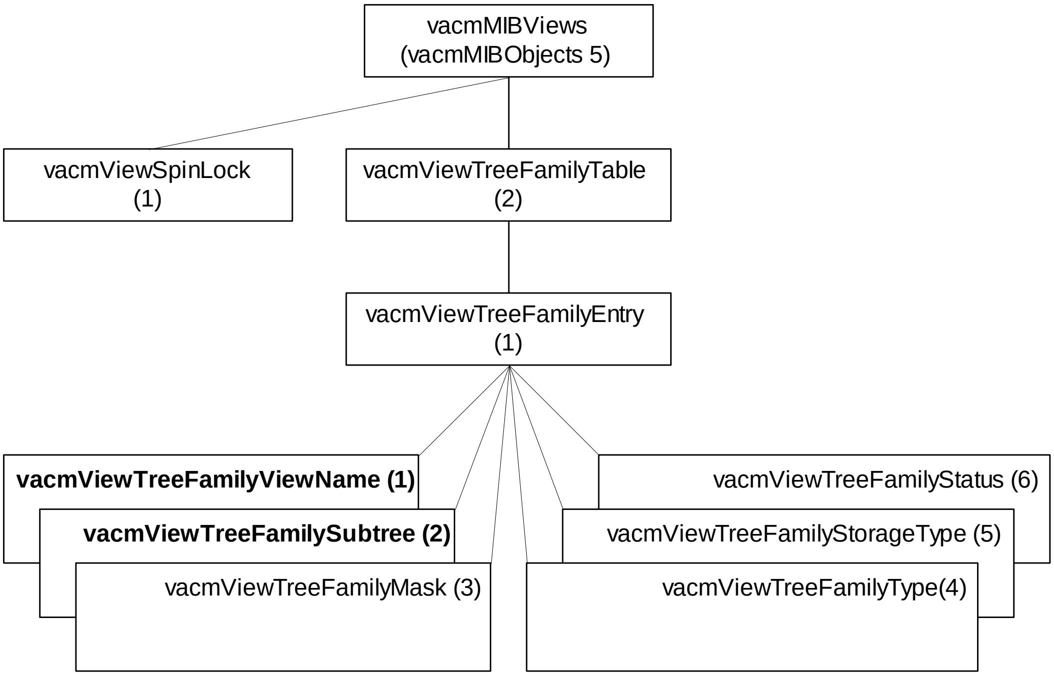


Figure 7.19 VACM MIB Views

Notes

Example:

Family view name = “system”
Family subtree = 1.3.6.1.2.1.1
Family mask = “” (implies all 1s by convention)
Family type = 1 (implies value to be included)