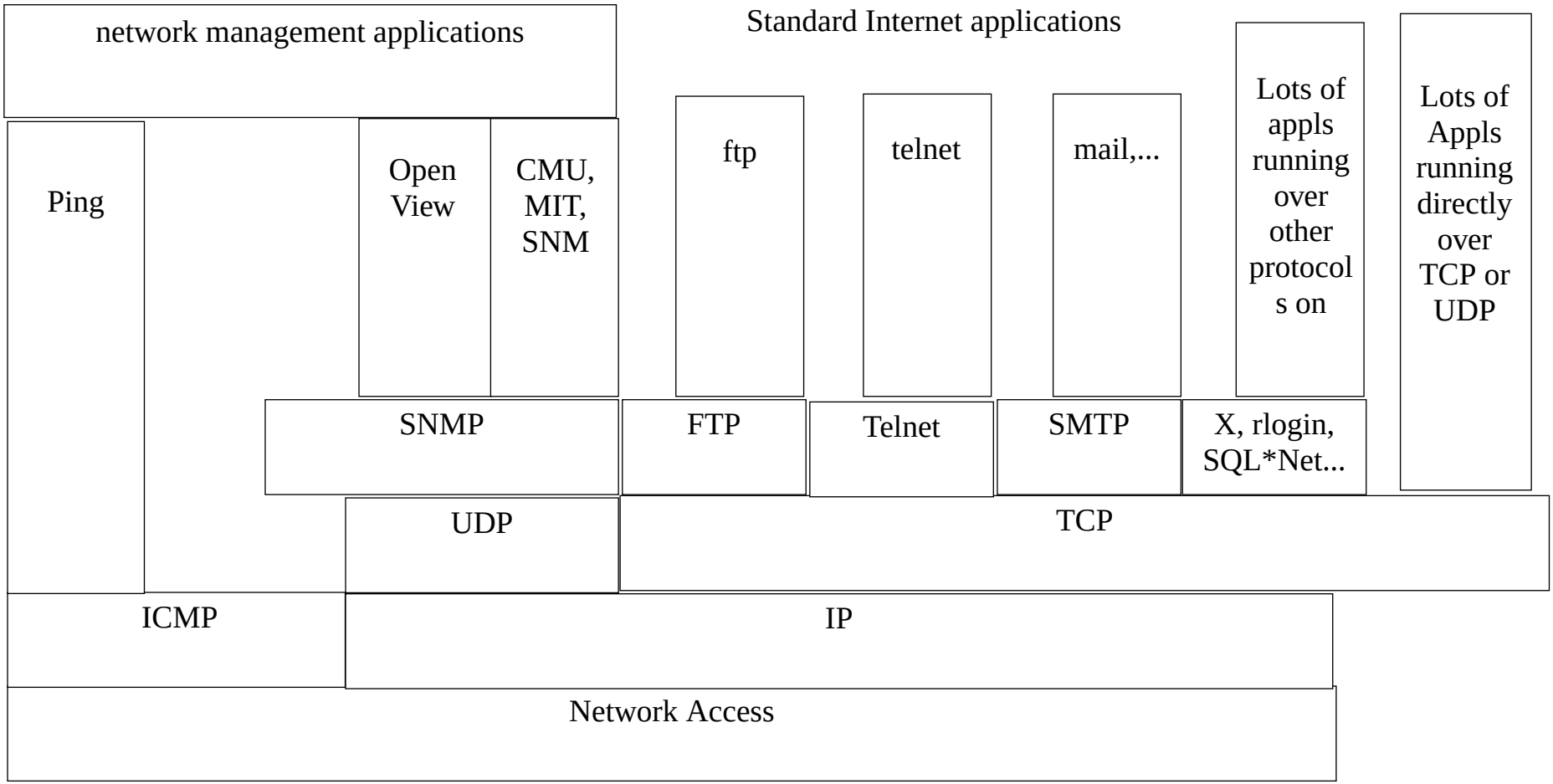
SNMP, NMS, OV

Simple Network Management Protocol

Network Management Systems

HP - OpenView

Internet protocol interfaces



(notes on) Internet protocol interfaces

IP can run over a lot of media, it was designed for lossy communications lines that were fairly slow, and with dedicated communications processors at the ends. It can be tweaked to work very well on LANS. These days it has run out of address space, so there is a next generation IP coming.

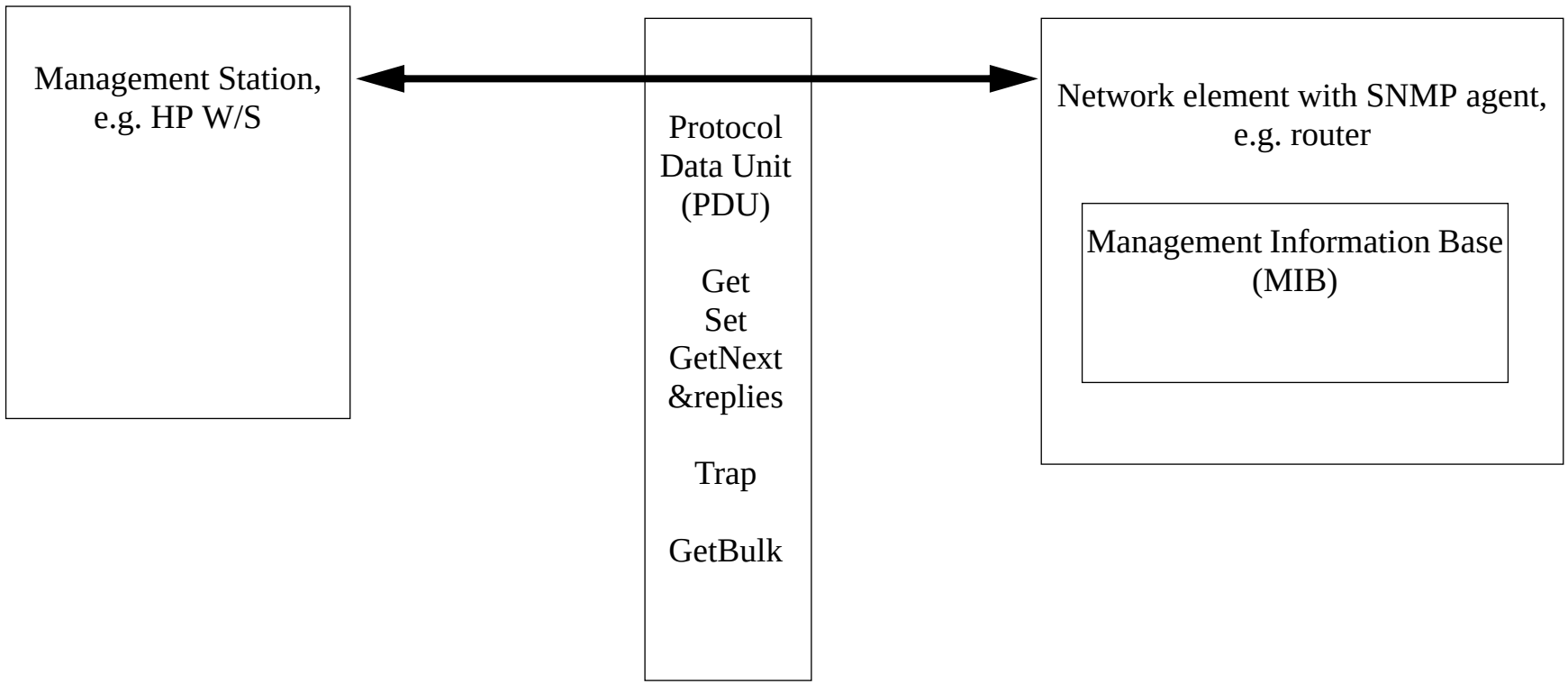
TCP is the most common transport layer that runs over it. TCP also runs out of steam on very fast networks as there is too much inherent latency in the protocol, however it will go on for some time yet. TCP is the equivalent of the OSI connection oriented protocols (CO). The byte stream style interface, and handshaking provided by connection oriented protocols make it easier for a variety of applications, and that is why there are so many. FTAM == FTP, VT == telnet, X.400 == SMTP, minus functions and options.

UDP is the equivalent of the the connectionless protocols (CL). UDP is the user datagram protocol, also known as the unreliable datagram protocol. With UDP each packet has a life of its own, and may or may not reach the destination. In a situation where the network is cracking up under stress, a heavy-weight protocol may specd its time handling error recovery. Successive UDP packets may be routed differently. With the application end doing retries, and some other things in the protocols, UDP is useful as a low layer transport for network management applications.

ICMP is the Internet Control Management Protocol, and was meant to be used for low level management by the stacks themselves. It generally isn't fully used, maybe not even fully implemented. The most common application seen in day to day life is PING, Packet Internet Groper (ICMP echo request / reply).

Network management is about finding out the state of things on machines other than the one you are connected to. Hopefully NMS applications are proactive in that they point out to the operator what is wrong either before it goes wrong, error counters on an interface, a lost connection between two machines making part of the network unreachable, so system will highlight which particular port or link has failed. We want this to make detection quicker, repair initiation faster, and to not put the operator on the defensive when a user tells them of the fault.

SNMP


Management Station,
e.g. HP W/S
Protocol
Data Unit
(PDU)
Get
Set
GetNext
&replies
Trap
GetBulk
Network element with SNMP agent,
e.g. router
Management Information Base
(MIB)

(notes on) SNMP

SNMP is a simple message based system, an application sends a message called a protocol data unit to an agent process on the remote system, which may be a workstation, or a router or a bridge. There have been demonstrations where they set up a hardware controller for a 1950's Sunbeam toaster and performed controls on it.

In the agent there is a conceptual database called a management information base, or MIB. The operations are simple Get or Sets on attributes in this MIB. The funny feature of these attributes is that they change with time if they are read-only things, like counters of the number of bytes sent or whether there is a carrier signal on the wire or if a back-hoe has cut it. If they are read / write, then they almost certainly have side-effects. These may reboot the box, increase the size of an internal table, or change where messages on some other data channel are routed to. In actual practice there is not likely to be a simple relational database sitting behind a MIB. This does occur, we did it in Fastpac, and there are some simple attributes that are just set for informational or inventory purposes.

There are also TRAPS which originate from the agent to say something is wrong. The management application is then expected to read the MIB to see what.

The GetNext (and SNMPv2's) GetBulk are for getting information out of the MIB when you don't precisely know what is in there. The way the attributes are arranged in the MIB is that there are typically no tables of contents or an index. The application has to precisely know what is there already, such as sysUpTime, or use GetNext to scan sequentially through the MIB attributes. -> Attribute naming

Attribute naming

Attribute name

.iso.org.dod.internet.private.enterprises.hp.nm.system.general.computerSystem.computerSystemUsers

Numerical form (Object ID or Registration ID)

.1.3.6.1.4.1.11.2.3.1.1.2

Particular instance of this simple attribute

.1.3.6.1.4.1.11.2.3.1.1.2.0

For simple attributes of which there is one per IP address, ".0" is appended to the identifier.

Data type for attributes:

Integer, Octet Strings[1], Object Identifiers, IP and Network Addresses

1. Fixed and variable length.

(notes on) Attribute naming

Attribute naming (in tables)

Attribute name

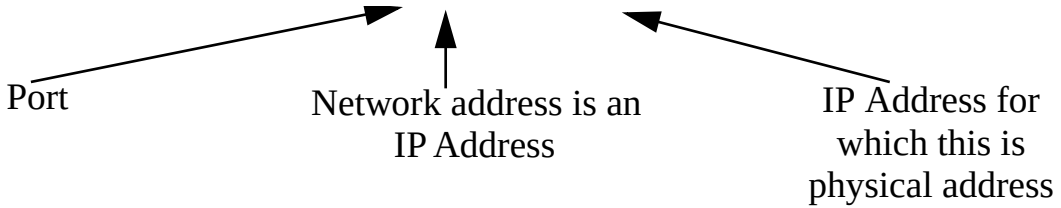
.iso.org.dod.internet.mgmt.mib-2.at.atTable.atEntry.atPhysAddress

Numerical form

.1.3.6.1.2.1.3.1.1.2

Particular instance

.1.3.6.1.2.1.3.1.1.2.1.1.130.102.76.163



For attributes for which there are many at a particular node, i.e. there is a table of these things, the attribute identifier is supplemented with the numerical values of the distinguishing attributes.

(notes on) Attribute naming (in tables)

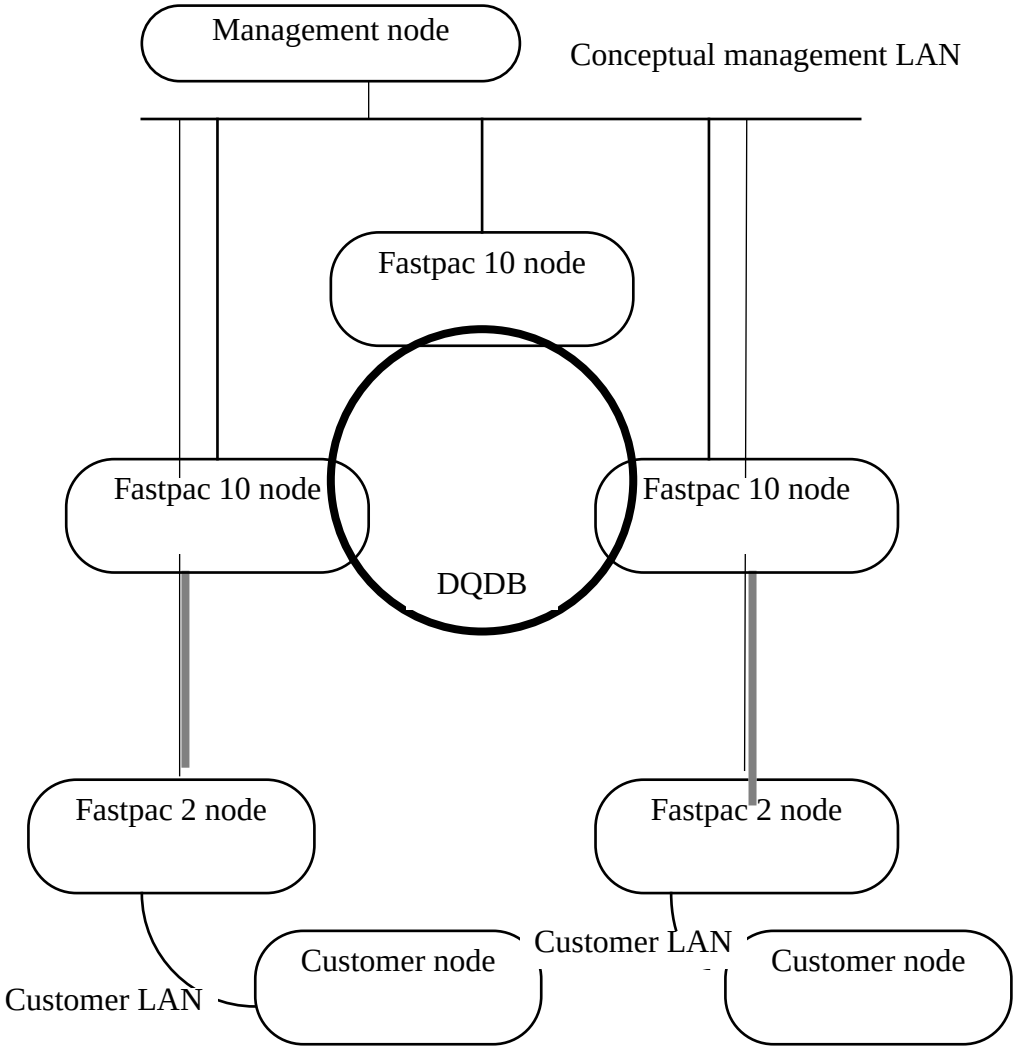
This is from the address resolution tables (or ARP tables) supported in the most commonly supported MIB (MIB-II). What we are holding in this variable is the ethernet (or other physical) address of a node accessible via this particular port.

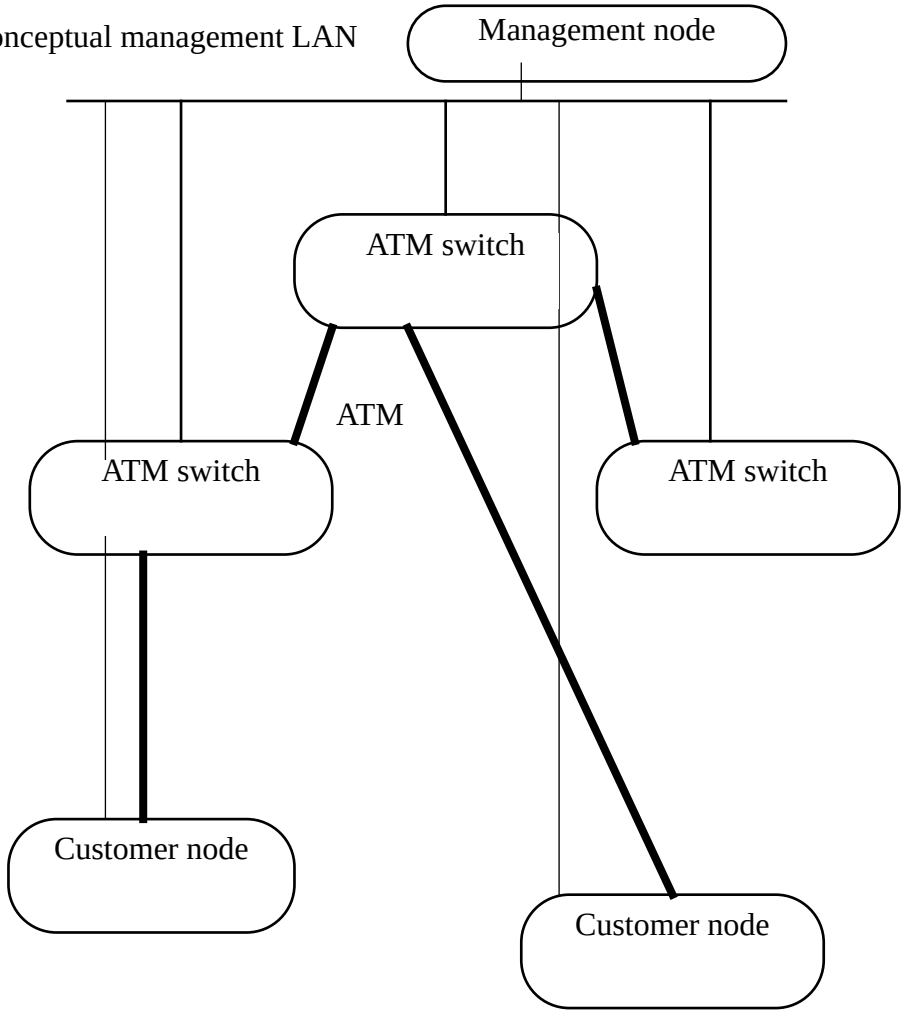
The first bit of supplemental information is the port number. (the first 1)

The second part is the type of address the host knows the remote node by. The second digit 1 indicates that an IP address is following. other values could have been defined to support things like OSI addresses, but they didn't.

The third bit is the actual IP address, which is the last 4 numbers.

Network management systems





CiTR

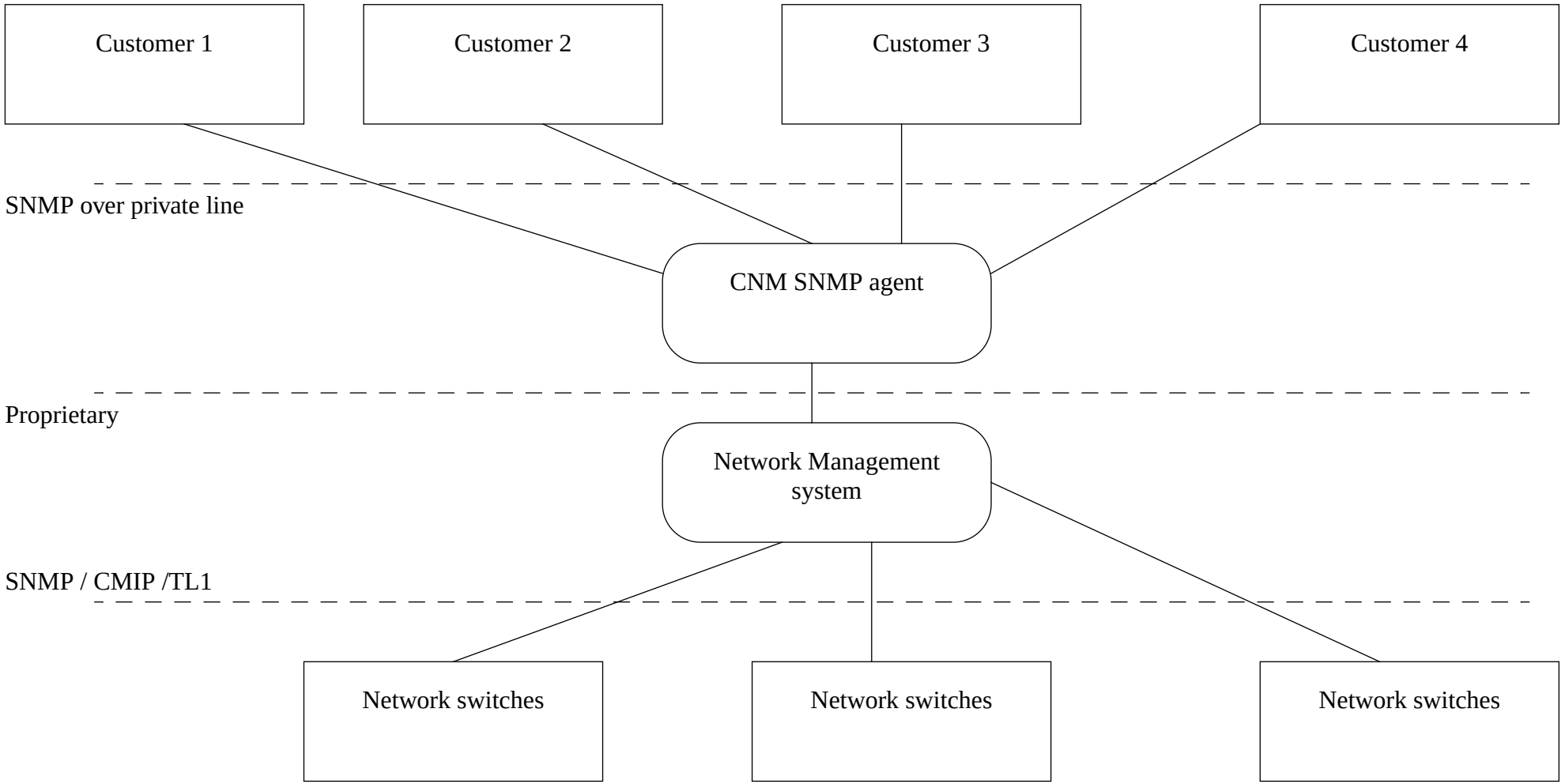
SNMP, NMS, OV

(notes on) Network management systems

The Telecom Fastpac-2 and NEC ATM LAN management systems are designed to manage networks that are faster and bigger than the network that is performing the management. So there are these fast base technologies, DQDB and ATM, pushing packets around at high speeds. Where and if they push the packets is controlled by the fastpac nodes, or by the ATM switches. For the end-customer applications, these devices are invisible, same as most repeaters and bridges.

What the management systems are doing is controlling these special nodes. Some of the management traffic is multiplexed over the primary media. So there aren't really ethernet links out to all the controlled nodes. This out-of-band management is much more expensive due to duplication of wiring, but is used for some particularly high reliability services.

Customer network management systems



(notes on) Customer network management systems

CNM systems allow the end customers to have some access to the network service that is supplied to them, say by a carrier. For example in an ATM service, the CNM service may allow them to view the performance related counters on the ports of the carrier network. It may allow them to request provisioning of new PVCs.

MIB definition

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

(notes on) MIB definition

SNMP implementations

Switched

OpenView Postmaster (DME) (XOM/XMP)

DEC Polycenter (-> TMIP?)

Agent X

SNMP Research (Emanate)

Direct

OpenView (direct)

CMU SNMPv1/v2 & UCD variants

SNMP++

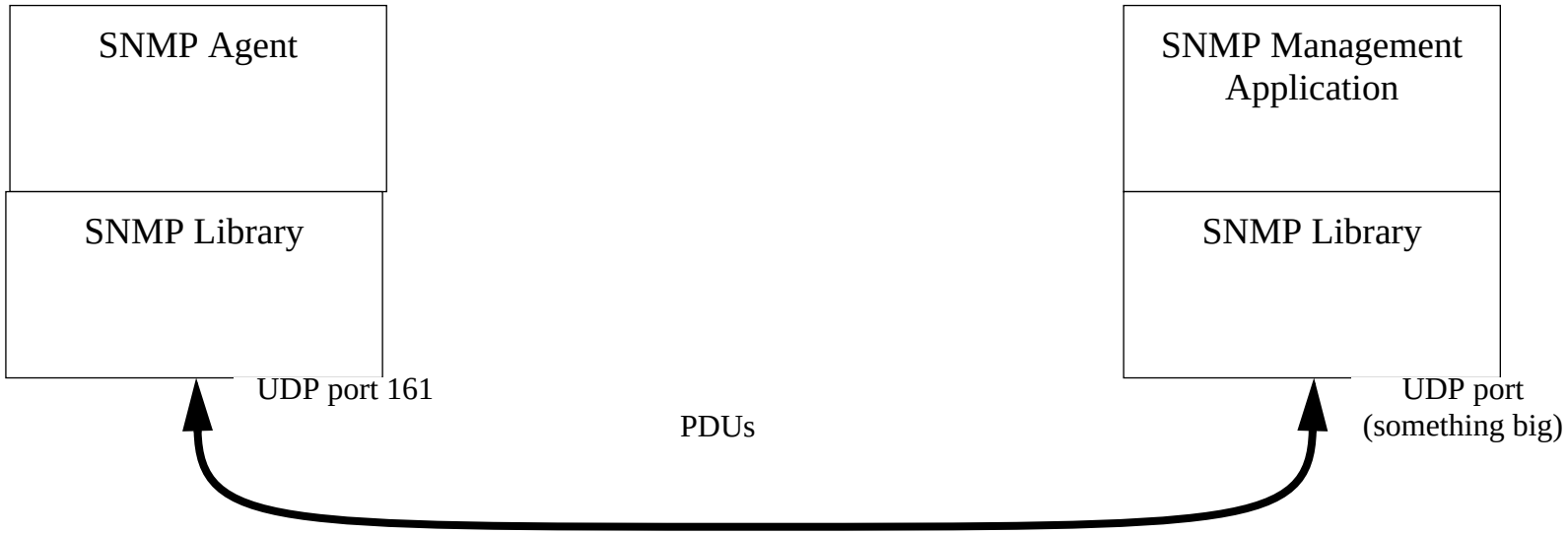
MIT

SunNet Manager

(notes on) SNMP implementations

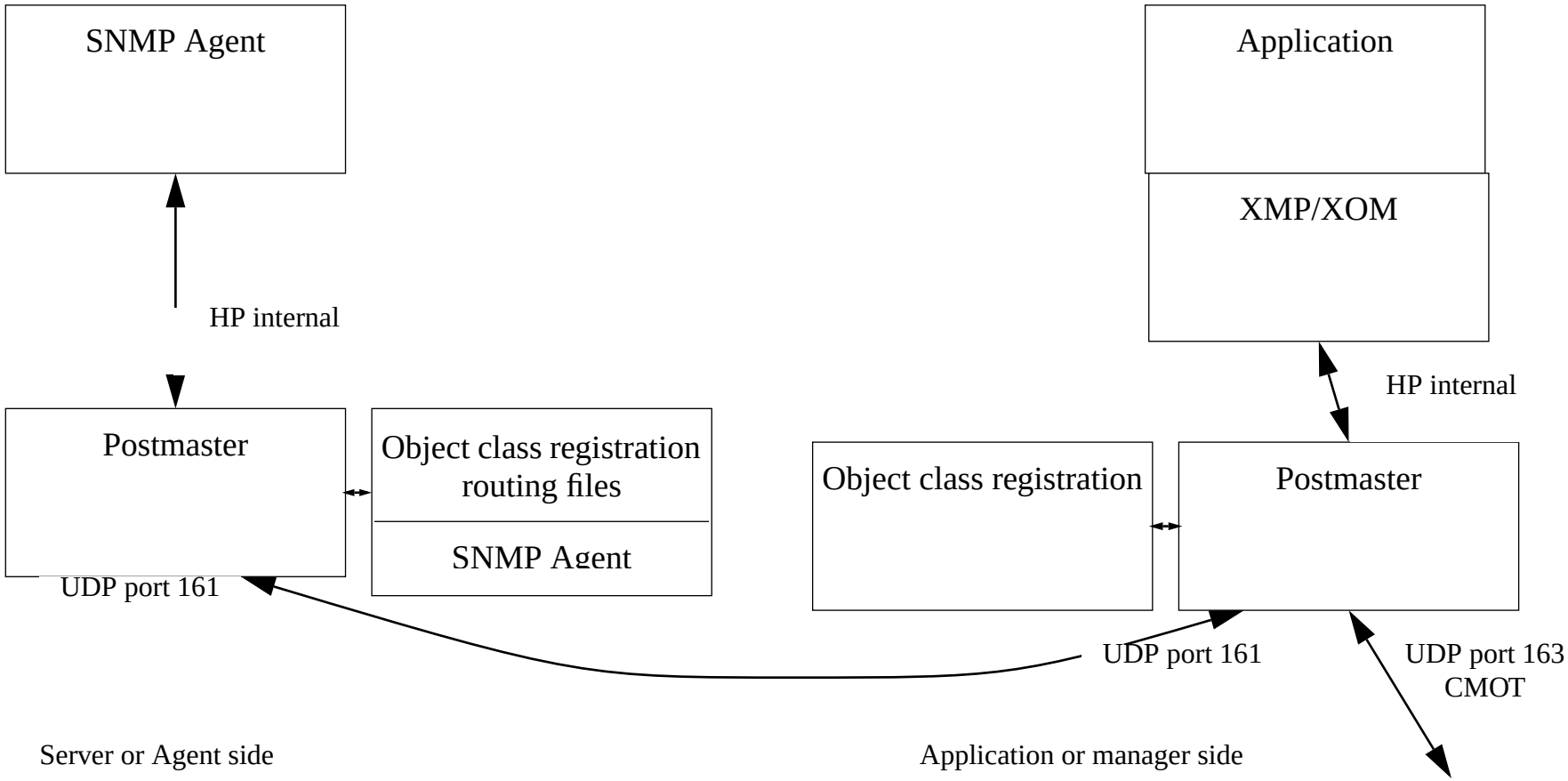
There are a multitude of SNMP APIs. There is one with the both CMU packages, one with the MIT package. There are a couple of standard ones, the XMP / XOM which also encompasses CMIP. The HP implementation, or a variant is in OpenView, IBM NetView / 6000, and DME. There is also the WINSNMP standard which is a Windows API standard that a number of the PC SNMP platforms are supporting.

Simple SNMP


SNMP Agent
SNMP Library
UDP port 161
PDUs
SNMP Management Application
SNMP Library
UDP port (something big)
Server or Agent side
Application or manager side

(notes on) Simple SNMP

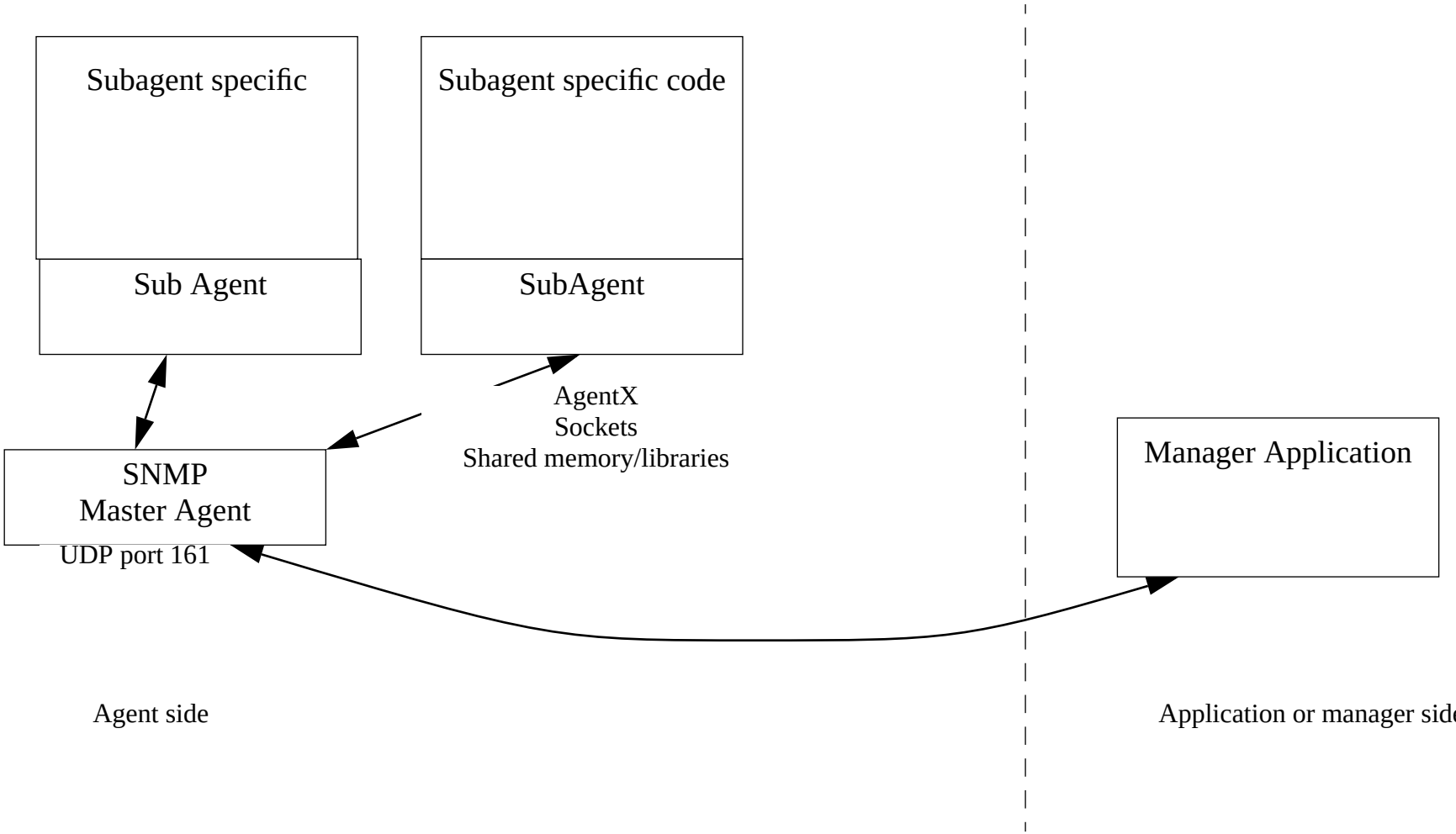
HP Postmaster (DME) -- SNMP, CMIP, CMOT



(notes on) HP Postmaster (DME) -- SNMP, CMIP, CMOT

In the current OpenView, HP provide an SNMP agent that can operate in either mode.

Subagents


Subagent specific
Subagent specific code
Sub Agent
SubAgent
AgentX
Sockets
Shared memory/libraries
SNMP
Master Agent
UDP port 161
Manager Application
Agent side
Application or manager side

Notes on Subagents

The singe port, 161, is a bit of a restriction, when multiple vendors or applications all want to be the SNMP agent. So the concept of subagents was introduced. In this a master agent acts as a funnel for all SNMP requests to the device, but the MIB itself is managed by subagents.

At subagent startup, these register subtrees or attributes with the master, who will be responsible for delegation. When a request comes in, the master forwards the request to the subagent to process through some implementation specific means.

AgentX - Standardised messages to another process

Emanate - Messages or shared memory - used by Sun & HP

CiTR - Shared memory, with loadable shared libraries for API

References

- **OSI & TCP/IP Impact on Software Vendors and Users William Stallings [(Video) TK5105.5.S73]**
- **The simple book : an introduction to management of TCP/IP-based internets -- Marshall T. Rose [TK5105.5.R681991]**
- **“munnari.oz.au” ftp server “/rfc” [1]**
  - **rfc-index**
  - **rfc1212 (concise MIB definition)**
  - **rfc1213 (MIB-II)**
  - **rfc1157 (SNMP)**
  - **rfc1901 - rfc1910 SNMPv2**
  - **rfc2011-rfc2013 (IP, TCP, UDP MIBs)**
  - **rfc2271-rfc2275 SNMPv3**
  - **rfc2257 - AgentX**
- **/home/horton/doc/SNMP_Presentation.pdf**

1. /citr/Information_Centre/RFC/

(notes on) References