Network Management

Lecture 11

Network Management System Design

Functional Requirements

- Scalability
- Heterogeneity
- Geographic spread
- Bursty Load
- Real-time response
- Batch processing
- Diverse users
- Local and remote management
- Ease of use
- Security
- Data management

Notes

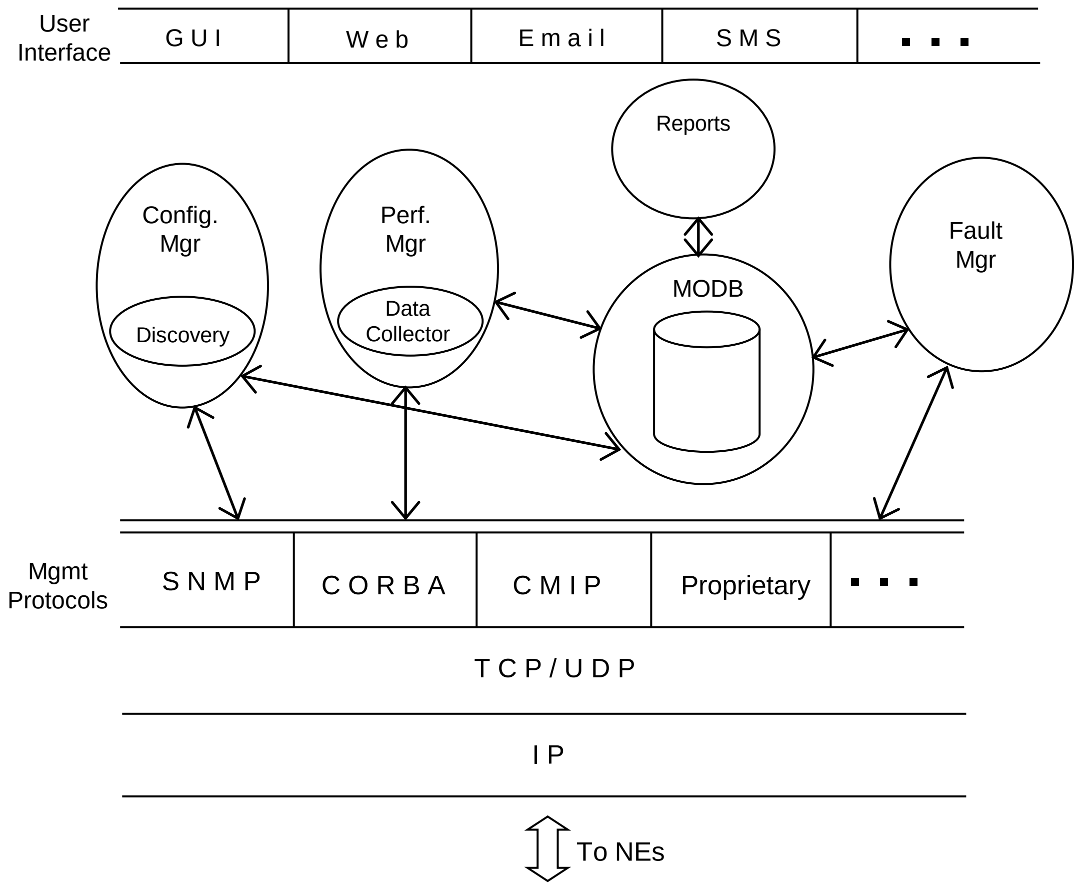
NM Software Components

- NMS Server
  - Centralized
  - Distributed
    - Local NMS / EMS
    - Integrated NMS / MoM

- NMS Client
  - Local / Console
  - Remote
    - Dedicated
    - Browser-based

Notes

NMS Server Architecture



Notes

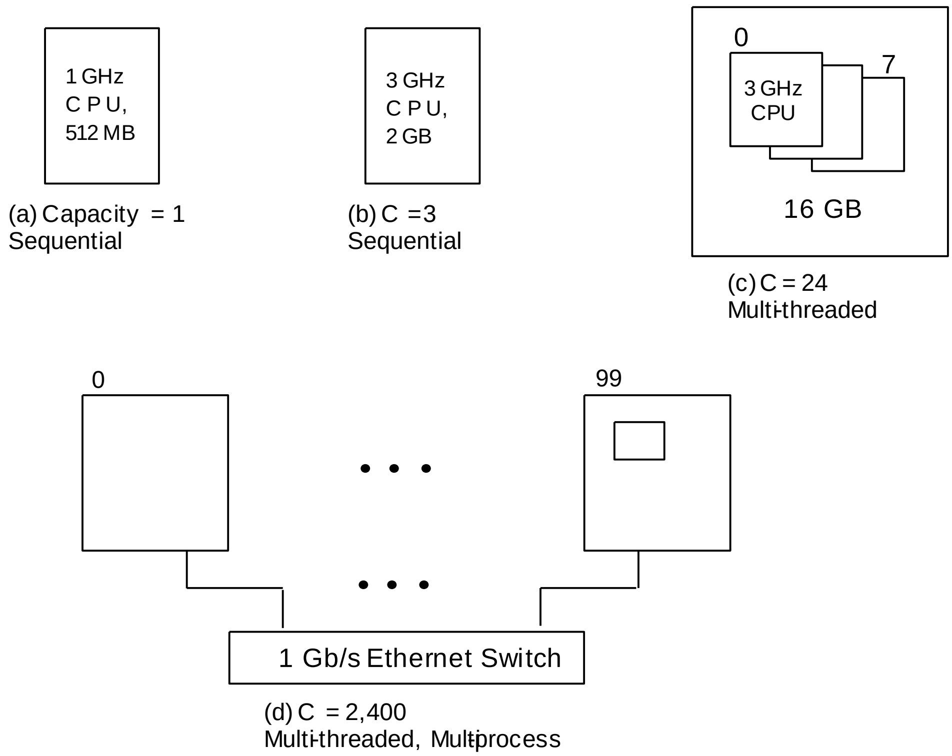
- Modular architecture
    - Managed object database
    - Configuration Manager / Discovery Module
    - Fault Manager
    - Performance Manager
- Module Layers
    - Lower (core) layer: Performs business logic
    - Upper layer: Graphical User Interface (GUI)

Server Key Design Decisions

- Modular
- Object-oriented
- Scalable
- Real-time response for handling notifications; Priority
- Software structure: Threads and Processes
- Relational DBMS: Oracle, DB2, SQL
- GUI: Browser-based.
- User-friendly interface

Notes

Design Decisions: Scalability



Notes

- Limited scalability achieved with more powerful hardware: Compare (a) and (b)
- Scalability increased with multiple CPUs sharing memory (SMP): (c) above
- As SMP gets expensive, redesign software with cluster of MSPs interconnected by high-speed LAN: (d) above

Discovery Module

- Discovers topology of network
- Auto-discovery
- Manual configuration
- Discovery queries for NEs in specified range of IP
- Found IP NEs are queried for specific details
- Discovery process is configures with parameters given in the above table

Notes

Discovery Parameters

Parameter	Value	Description
IP addresses	10.0.0.1 – 10.0.0.254, 192.168.0.0 / 24	A range or list of IP addresses
Wait Interval	10 secs	Waiting time between discovery of successive IPs to minimize load on the network
SNMP version	v1	v1, v2c, or v3
SNMP community	“public”	A commonly-used value
Discover types	Router, server, switch	Only elements of these types are added to the MODB
Ignore types	PC, UPS	Elements of these types are not added to the MODB

Notes

- Discovery parameters are set in the discovery configuration file
- Discovery parameters chosen to manage the scope of NEs discovered and the time for discovery run

Discovery Procedure

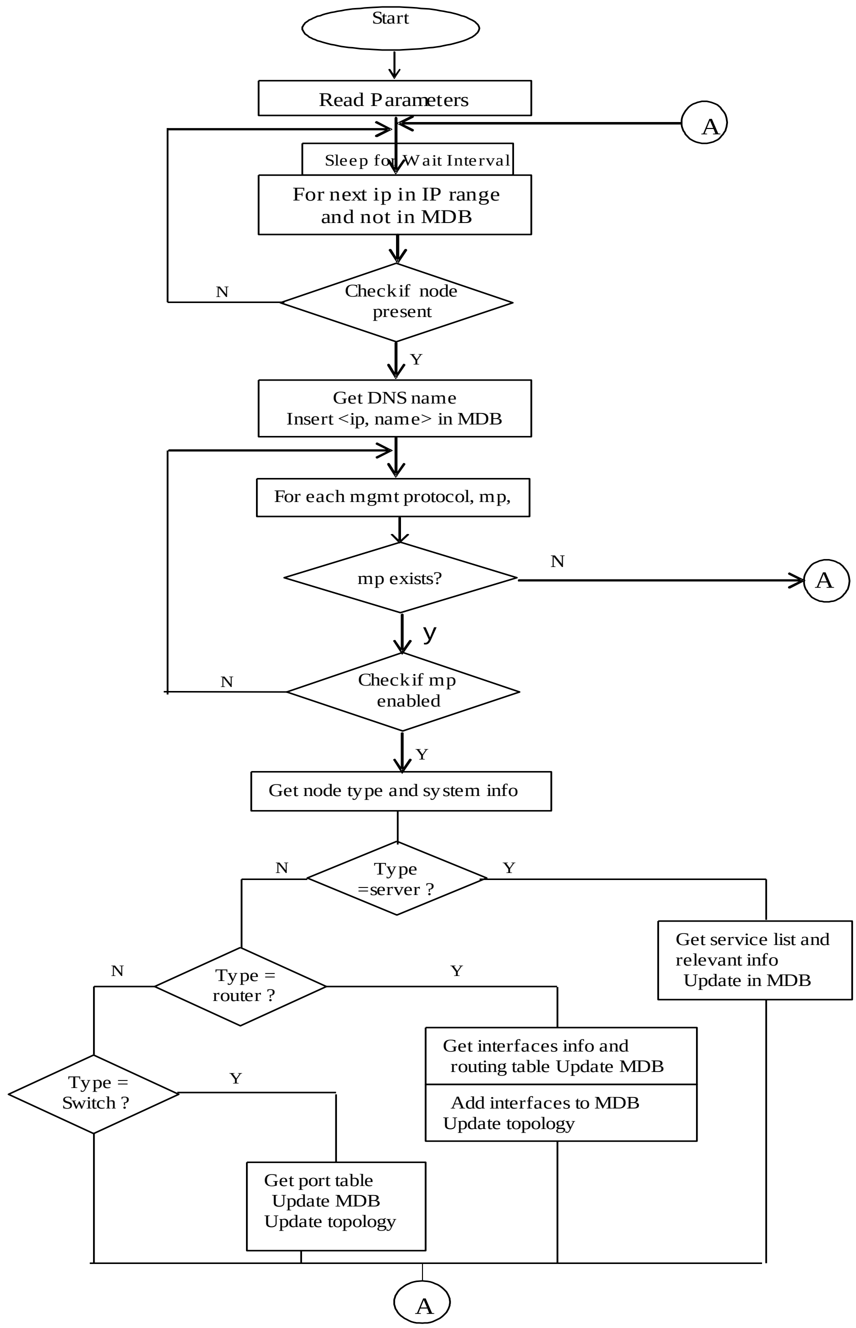


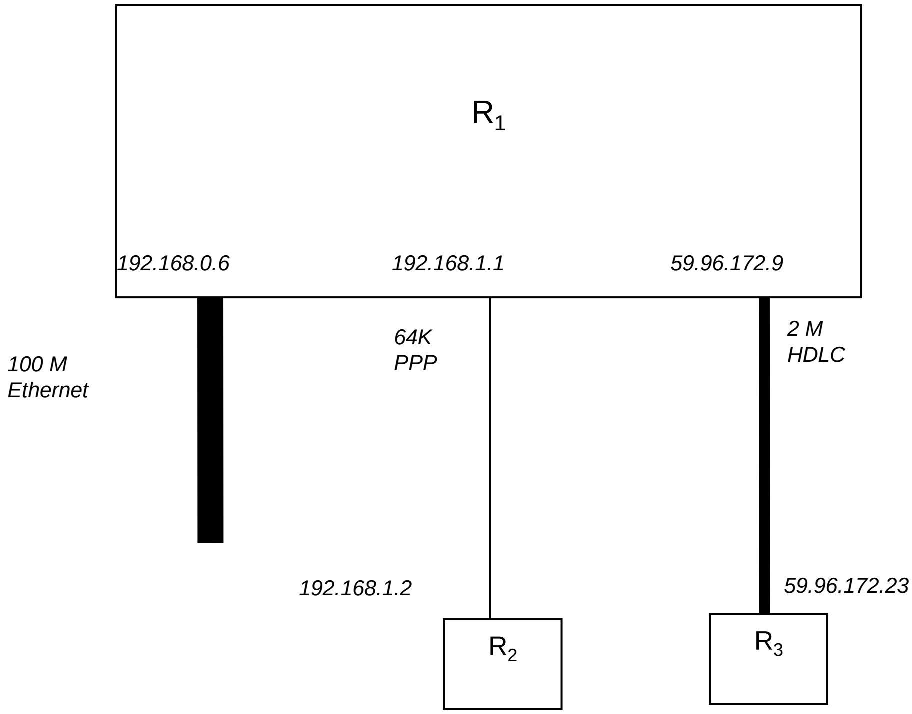
Figure 9.24 Discovery Procedure

Rediscovery

- Periodic check by NMS NEs in the network with MDB for changes in the network
- Deleted NEs are detected by scheduled polling and to be manually deleted by the operator
- New NEs are discovered by simpler discovery process and added to MDB
- Discovered components are compared to the ones in MDB and the difference reported for appropriate action

Notes

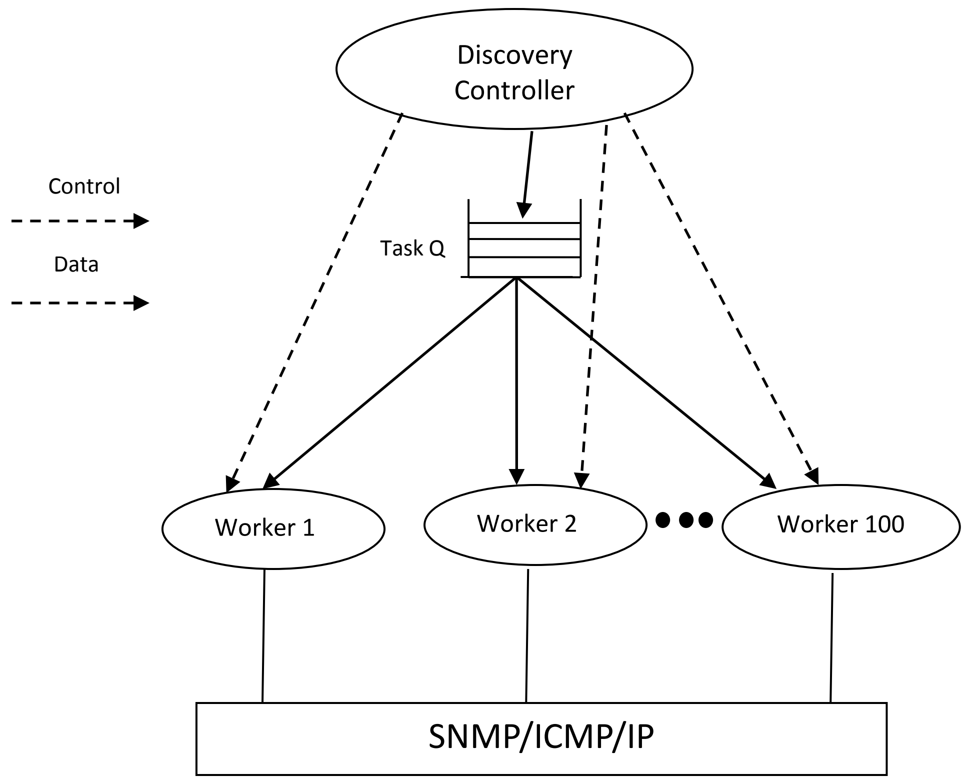
Topology



Notes

- Topology shows the nodes and interconnections
- Nodes are routers and switches; Hubs are non-managed
- Router information collected from router MIB variables
    - *ipAdEntAddr & ipAdEntIfIndex (ipAddressTable)*
    - *ipForwardNextHop ipForwardIfIndex (ipForwardingTable)*
    - *ifType & ifSpeed (ifTable)*
- Switch information (RFC 1493)
    - *dot1dBaseNumPorts*
    - *dot1dStpPortTable*
    - *dot1dTpfDbTable*

Thread Pool for Speedy Discovery



Notes

- Single threaded discovery is slow for large network discovery
- Lower % IP address in use, larger the discovery time
- Use concurrent discovery using pool of worker threads
- Discovery controller creates task queue, each task with 1 or more Ips
- It then creates a number of worker threads to do parallel processing

Single Thread vs Multithread discovery

9.7 Time to Complete Discovery Sequentially
(Note: Numbers in italics are days, normal font in hours)

% Addresses in Use / **Total Assignable Addresses**	**5%**	**20%**	**50%**	**100%**
1 Class C ~ 250	2.0 hr	1.7 hr	1.2hr	0.3 hr
10 Class C ~ 2,500	20.0 hr	17.4 hr	12.2 hr	3.5 hr
1 Class B ~ 65,000	*21.6 dy*	*18.8 dy*	*13.2 dy*	*3.8 dy*
16 Class B ~ 1,000,000	*332.8 dy*	*289.4 dy*	*202.5 dy*	*57.9 dy*

Table 9.8 Time to Complete Discovery with 10 Concurrent Threads
(Note: Numbers in italics are hours, normal font in secs)

% in Addresses Use / **Total Assignable Addresses**	**5%**	**20%**	**50%**	**100%**
1 Class C ~250	72 s	63 s	44 s	13 s
10 Class C ~2,500	719 s	625 s	438 s	125 s
1 Class B ~65,000	*5.2 hr*	*4.5 hr*	*3.2 hr*	*0.9 hr*
16 Class B ~1,000,000	*79.9 hr*	*69.4 hr*	*48.6 hr*	*13.9 hr*

Performance Manager

- Two major functions
  - Data Collection
  - Analysis and report generation
- Data Collection
  - Offline: Use local data collector and periodic batch transmittal to NMS. Should have NE id, OID, Value and timestamp
  - Online: For real-time performance analysis:
  - Issues
    - Overloading of the server
    - Overloading the network
    - Overloading the agent
    - Poll configuration
    - Database schema

Notes

Poll Configuration

- Frequency of polling
  - Fast (backbone traffic) - 30secs to 2 minutes
  - Slow (free disk space) - once an hour
- Dynamic polling
  - Increase polling frequency during critical time (Congestion in traffic)
  - Keep the network load within limits
- Concurrency of polling in large networks
- Mitigate overrun (delay) in scheduled polling

Notes

Database Schema

Table 9.9 Typical Growth of Collected Data (56 B / Record)

	1 day	1 week	1 month	1 year
Number of records	29 m	201 m	863 m	10, 501m
Database size	1.6GB	11.3GB	48.3 GB	588 GB

Table 9.10 Comparison of Different DBMS Schema Designs

	1 table	1 table/NE	1 table/OID	1 table/day
Space / record	28 B	20 B	20 B	20 B
Number of open tables	1	Very large	Large	Few
Insert	Slow	Fast	Fast	Fast
Reports	1 table, conflict with insert	1 table, less conflict	Several tables, some conflict	1 table, Conflict for online reports, No Conflict for offline reports
Data safety	Poor	Good	Good	Very Good

Notes

- Database grows rapidly
- Database should be easily and rapidly accessible
- Careful design needed based on size and application needs
- Polled data can be stored in multiple tables
  - One table per NE
  - One table per variable
  - One table per day / period

Fault Manager

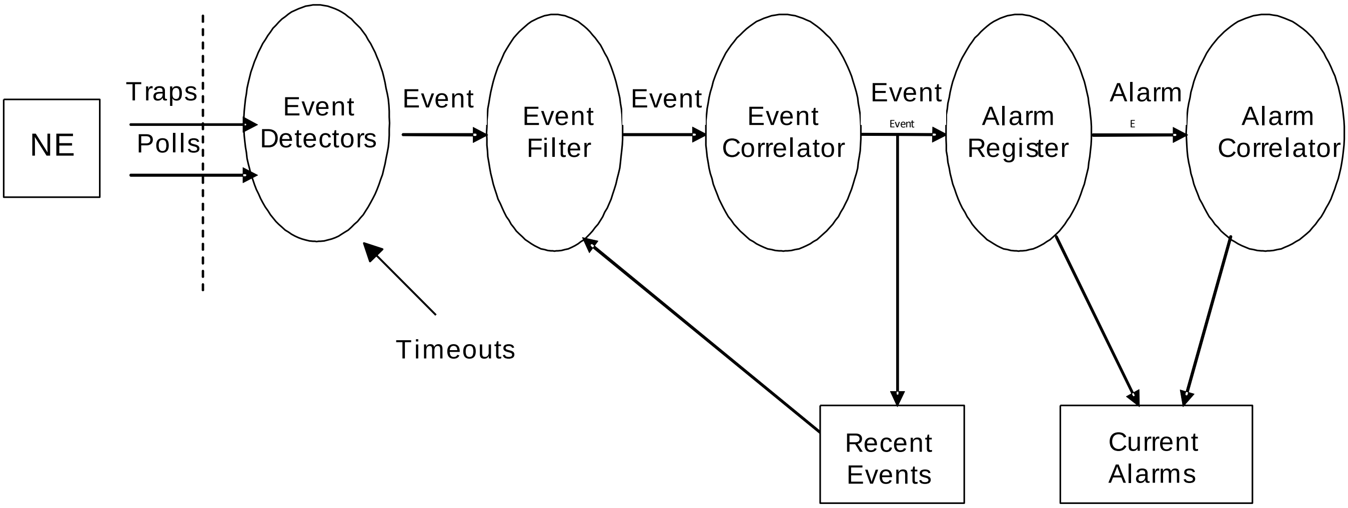


Figure 9.31 Path of an Event through the Fault Manager

Notes

- Event generation:
  - Notification or trap indicator of fault
  - Multiple consecutive status polls fail to receive response
  - Threshold crossing, e.g. performance limit
  - Internal escalation of fault resolution
- Event filtering of unwanted events
- Event correlation of repeated receipt of same fault event
- Conversion of event to alarm by alarm register
- Root cause analysis of multiple alarms done alarm correlator
- Alarm indications: Visual, audio, SMS / phone call, email, log

Alarm Severity and Priority

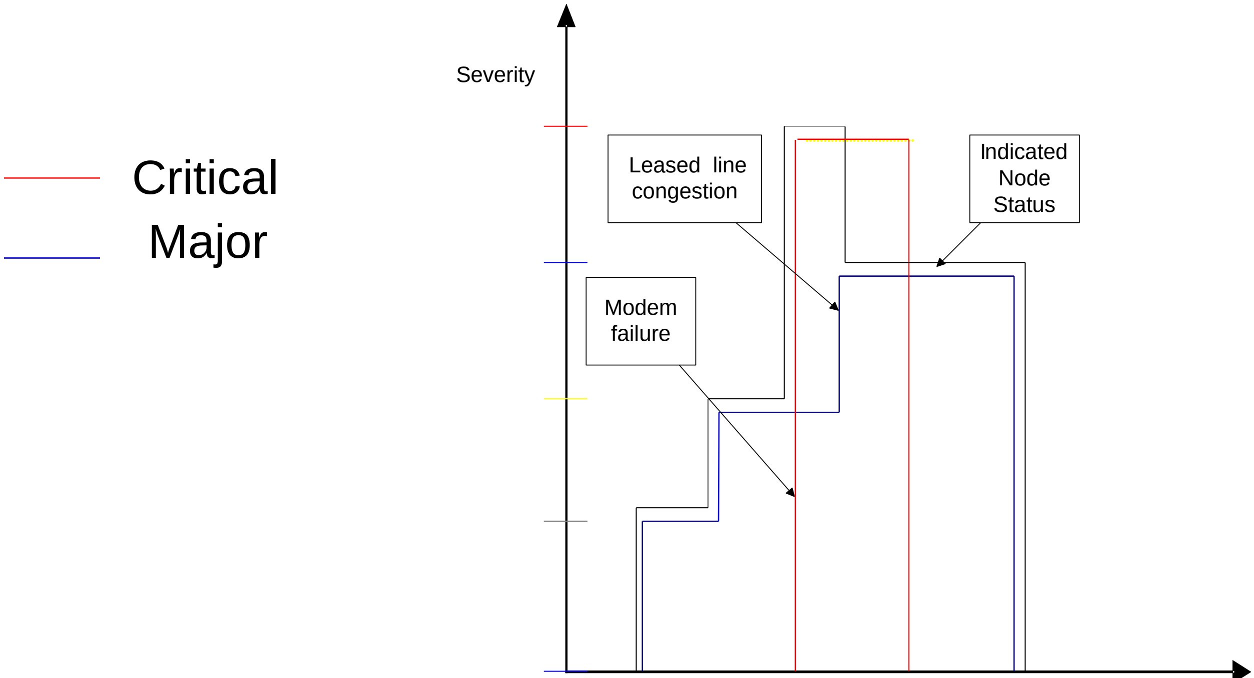
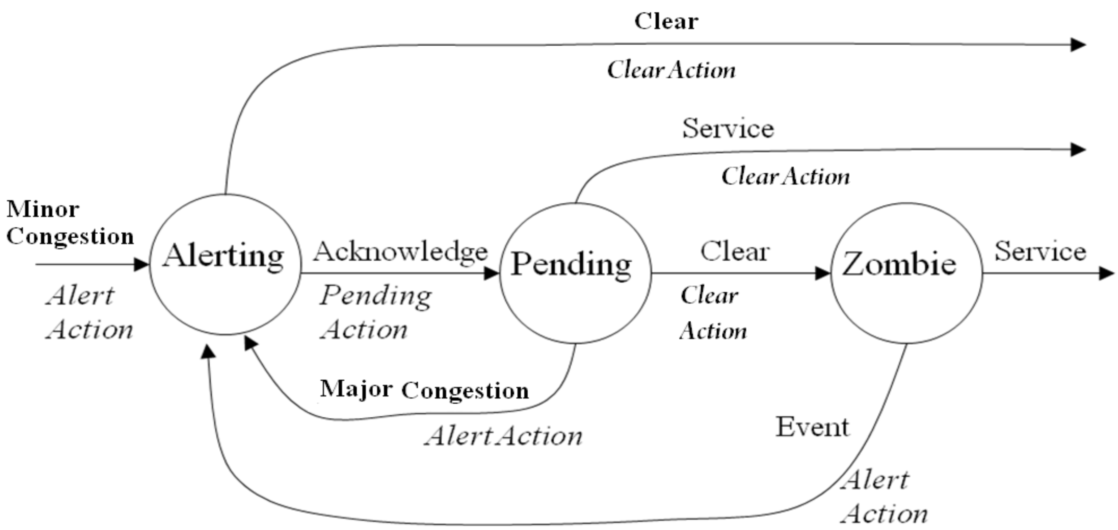


Figure 9.32 Alarm Indication for a Router with Two Simultaneous Faults

Notes

1. For each <NE, subsystem, event>, set the severity
2. For each <NE, subsystem, event>, set the priority
3. For each <NE, subsystem, event, severity>, set the alarm indication(s
4. For each <NE, subsystem, event, priority>, set the alarm indication(s)

Alarm Finite State Machine

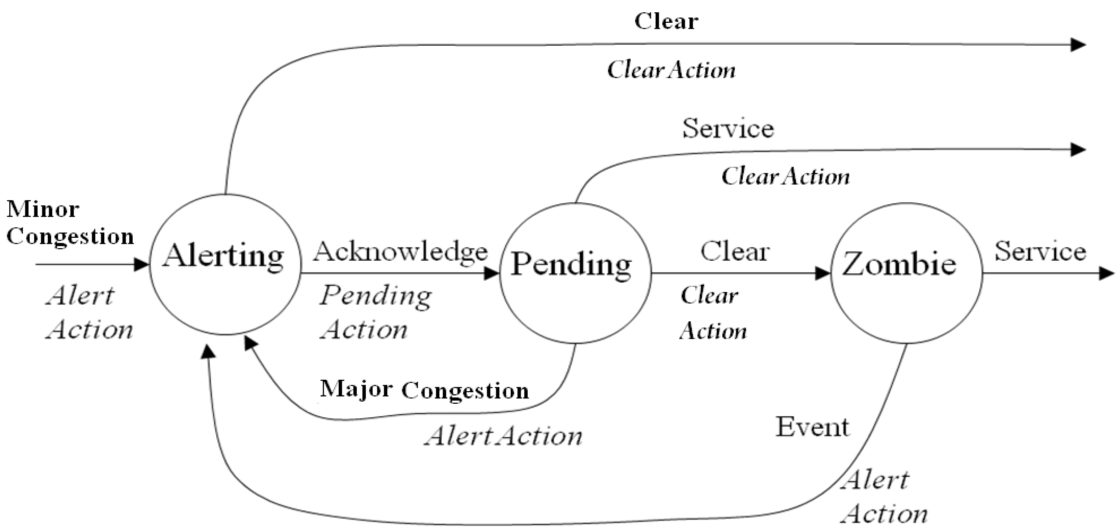


Notes

Table 9.11 List of Alarm States

State	Description
Initial	A new, unused alarm object
Altering	The alarm is registered in the list of active alarms. The FM operator has not yet noticed the alarm. The FM uses various alarm indications to catch the attention of the operator.
Pending	The operator has noticed the alarm. The FM is awaiting the results of corrective actions.
Zombie	The FM is aware that the fault condition is clear. The icon color is normal. The operator has yet to indicate the corrective action taken.
Closed	The operator and FM are both aware that the fault is closed. The alarm object is returned to the pool of unused alarms and is effectively in the initial state.

Alarm Finite State Machine

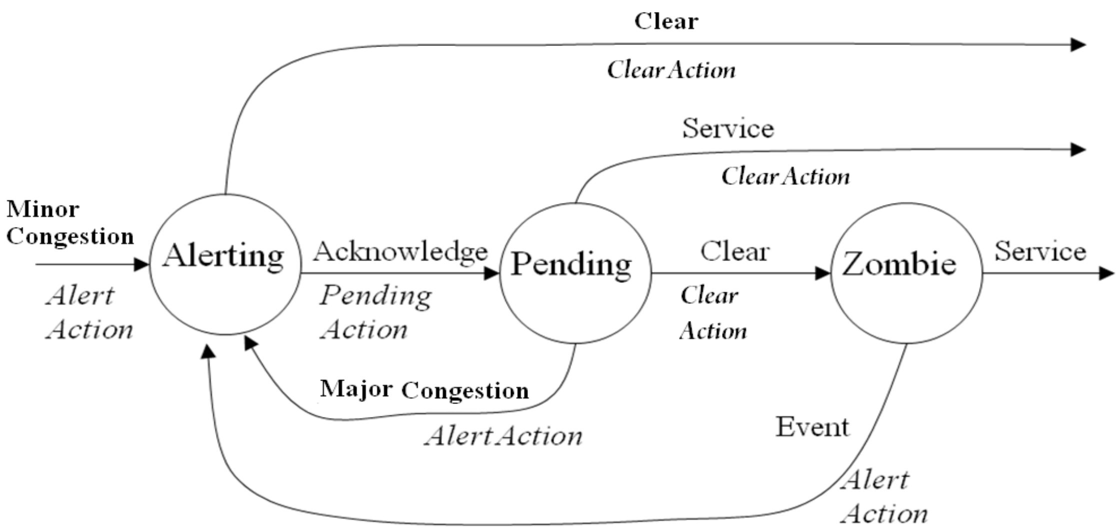


Notes

Table 9.12 List of FSM Events

Events	Description
Minor congestion	Utilization on a link exceeds 90%
Major congestion	Utilization or a link exceeds 95%
Clear	Utilization or a link exceeds 95%
Acknowledge	Operator invokes an FM menu or button to indicate awareness of a new alarm
Service	Operator invokes an FM menu or button to indicate corrective action taken

Alarm Finite State Machine

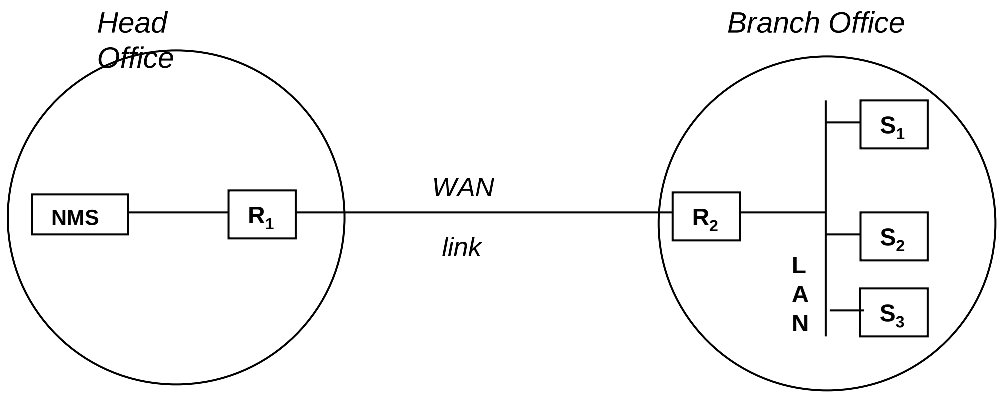


Notes

Table 9.13 Transition Table for Alarm FSM in Figure 9.33

	Initial	**Alerting**	**Pending**	**Zombie**
Minor Congestion	*Alert action* Alerting state	*Null*	*Null*	*Alert action* Alerting state
Major Congestion	*Alert action* Alerting state	*If alarm severity is Minor then: Alerting action* Alerting state	*If alarm severity is Minor then: Alerting action* Alerting state	*Alert action* Alerting state
Acknowledge	NA	*Pending action* Pending state	NA	NA
Service	NA	NA	*Clear action* Closed state	*Clear action* Closed state
Clear	Null	*Clear action* Closed state	*Clear action* Zombie state	*Null*

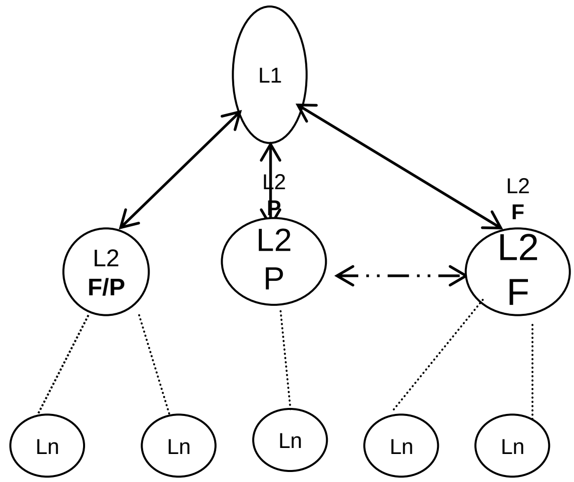
Root Cause Analysis



Notes

- RCA shows one primary alarm for one fault; Suppresses secondary alarms
- RCA based on dependency, i.e., reachability by NMS
- Status: *Up, Down, Unreachable / Unknown*
- R_1 failure primary node (shown red);
  Link, R_2, S_1, S_2, and S_3 are secondary failures (shown grey)

Distributed Management



Notes

- Large networks managed by geographically distributed configuration
- Several managers running on different servers
- Classification
    - Weakly distributed: Hierarchical, vertical delegation
    - Strongly distributed: Horizontal delegation allowed
- Multitier architecture: Trade-off between Flexibility vs Ease of implementation and deployment
- Topology: One parent NMS for each cluster of lower level NMSs

Server Platforms

- Operating System Requirements
  - Support threads, processes and shared-memory multiprocessors
  - Large disk, RAM and peripherals
  - Variety of programming and scripting languages
  - Supports 3rd party tools, components, and libraries
  - Examples: UNIX (Sun Solaris, IBM AIX, HPUX MacOS X, FreeBSD) and Microsoft Windows

Notes

- Database Management System (DBMS) Requirements
  - Support large amount of data
  - Support transactions
  - Good administrative tools
    - Create indices
    - Recover from table corruption
    - Backups
    - Performance tuning
  - Examples:
    - Open source: MySQL and PostgreSQL
    - Commercial: Oracle, IBM, db2, Microsoft SQL Server, Sybase

NMS Client Design

- Three approaches
  - Terminal client
  - Rich client
  - Browser or Web client

Notes

Terminal Client

- “dumb” character-oriented terminal
- Terminal emulation software (xterm, Putty) on PC
- Connection to NMS via telnet or ssh over TCP/IP
- No GUI

Notes

Rich Client

- PC runs special client application that works with server
- GUI interface
- Runs NMS functionalities
- Usually written in object-oriented languaage
- Drawbacks
  - Lack of portability
  - Requires update with NMS server update
  - High degree of incompatibility with multiple servers

Notes

Browser or Web Client

- de-facto standard now
- GUI capability
- Minimal NMS-dependent software in the client
- Portability between different browsers still a problem, but minimized
- Popular usage: Microsoft IE and Mozilla Firefox

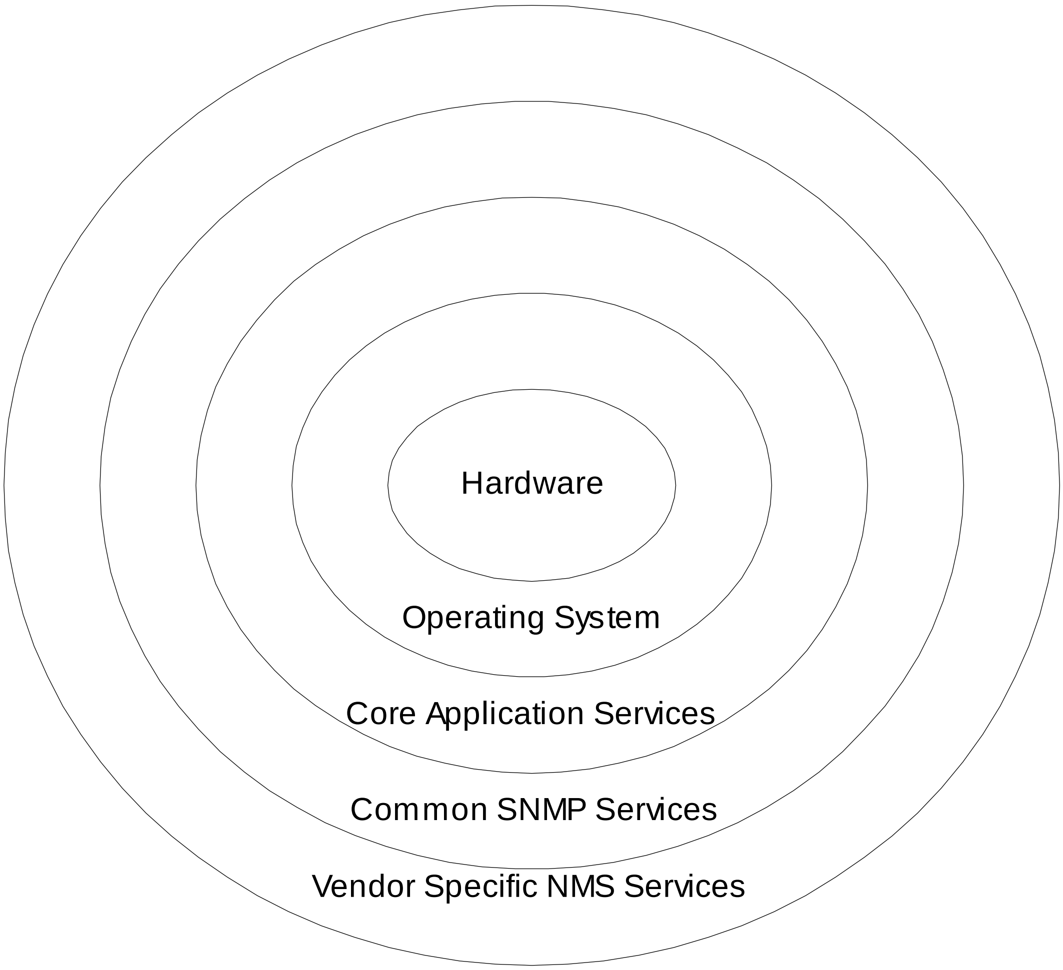
Notes

NMS for Enterprise & Service Providers

- Management of data transport
  - IBM Netview, IBM Microcuse,
    HP OpenView, NMSWorks CygNet
- Systems management
  - CA Unicenter and Tivoli TME
- Network and systems management
  - Partnerships
- Telecommunications management
  - NMSWorks CygNet, Agilent CNMS
- Service management and policy management

Notes

NMS Components



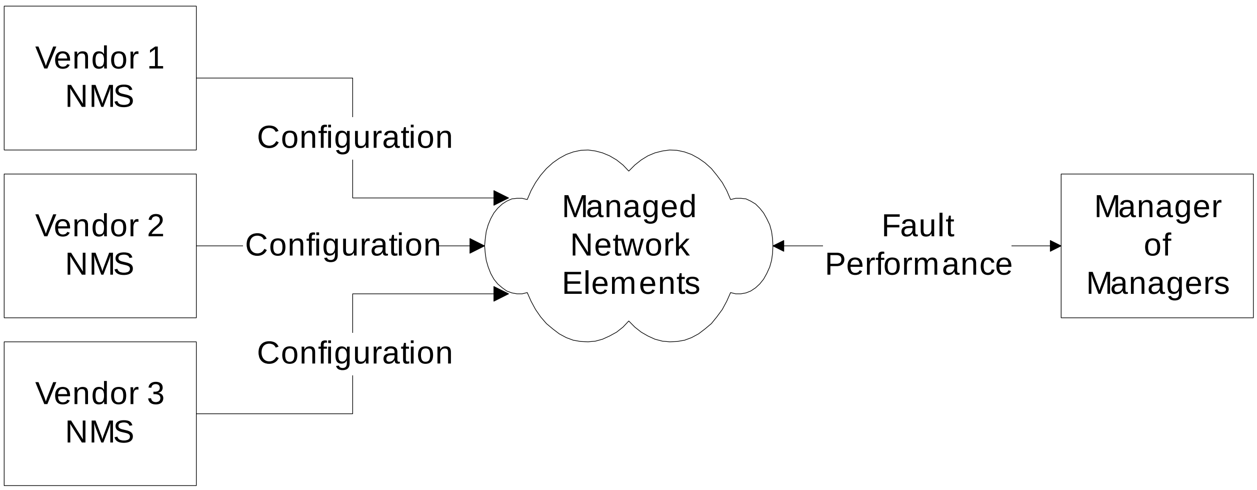
Notes

NMS Components

Component	Service
Hardware	Processor Monitor Mouse and Keyboard Communications
Operating system	OS services
Core application services	Display GUI Database Report generation Communication services
Common SNMP services	SNMPv1 messages SNMPv2 messages MIB management Basic SNMP applications 3rd party NMS API
Vendor-specific NMS services	MIB management SNMP applications Config. management Physical entity display

Notes

Multi-NMS Configuration



Notes

Network Configuration

- Configure agents
- Configure management systems
- Community administration parameters
  - Community name
  - MIB view
  - Trap targets
- Autodiscovery : Scope

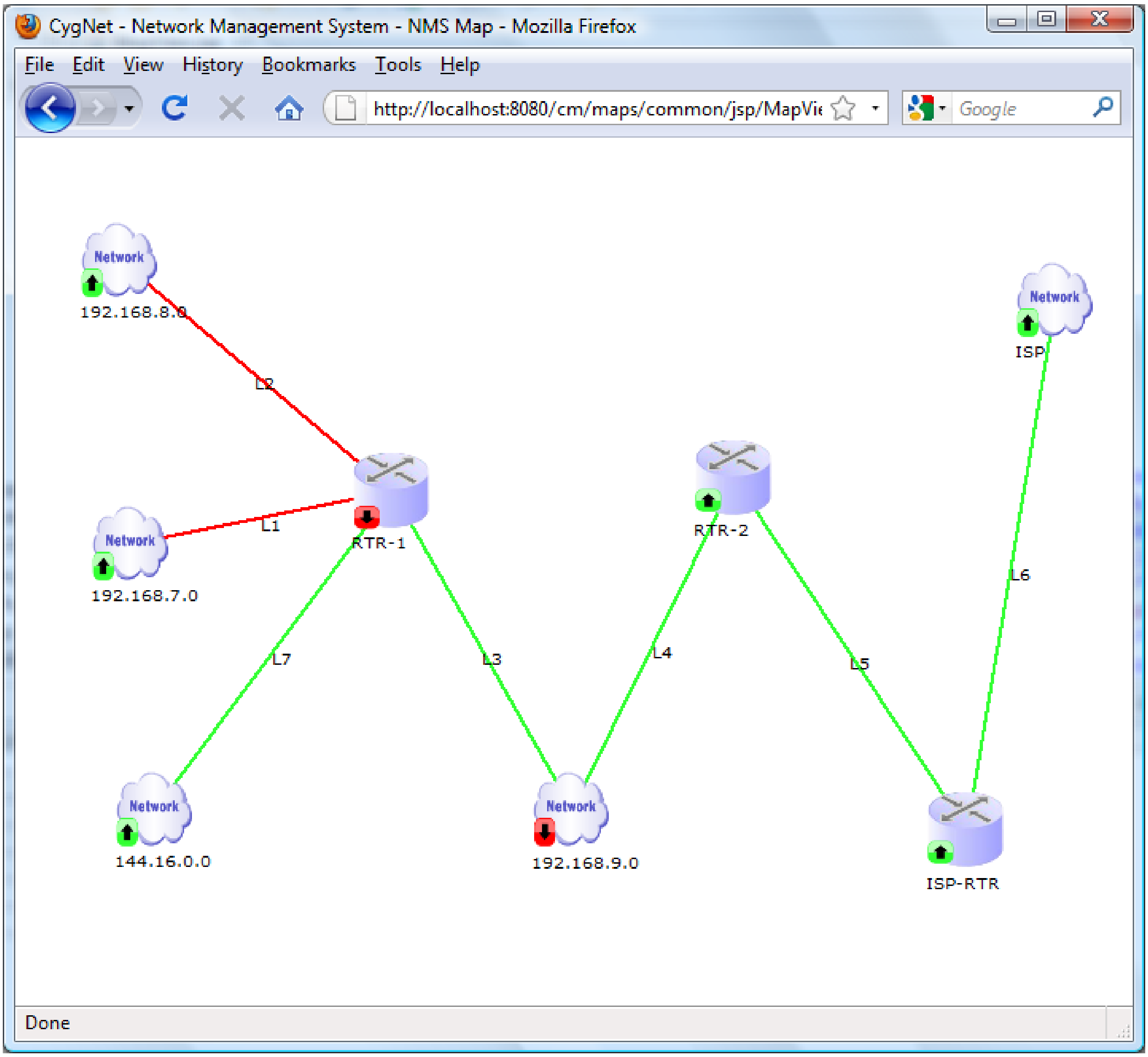
Notes

Network Monitoring

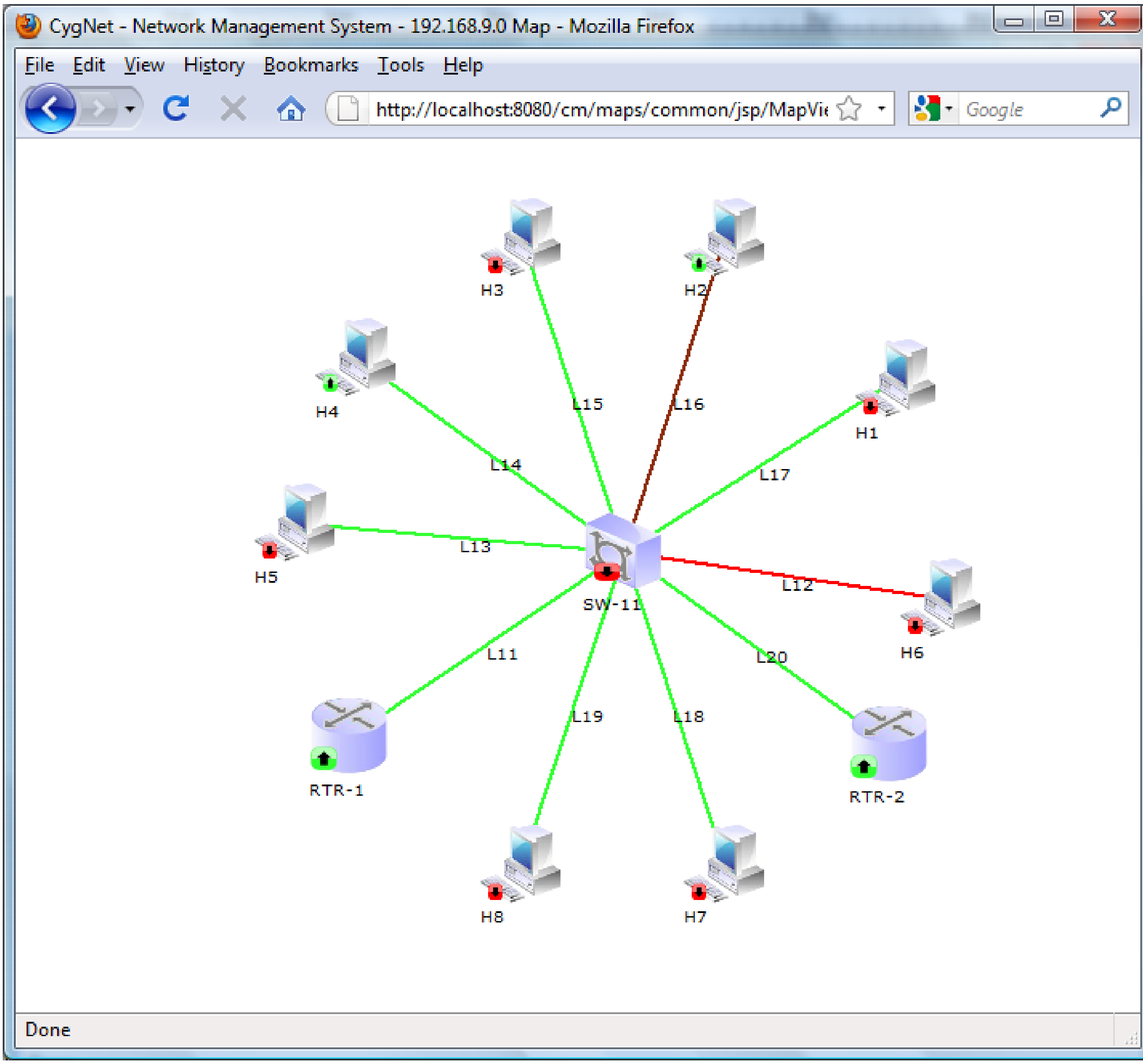
- By polling
- By traps
- Failure indicated by pinging or traps
- Ping frequency optimized for network load vs. quickness of detection
- trap messages: *linkdown, linkUp, coldStart, warmStart, etc.*
- Network topology discovered by auto-discovery
- Monitoring done at multiple levels - “drilling”

Notes

Global View

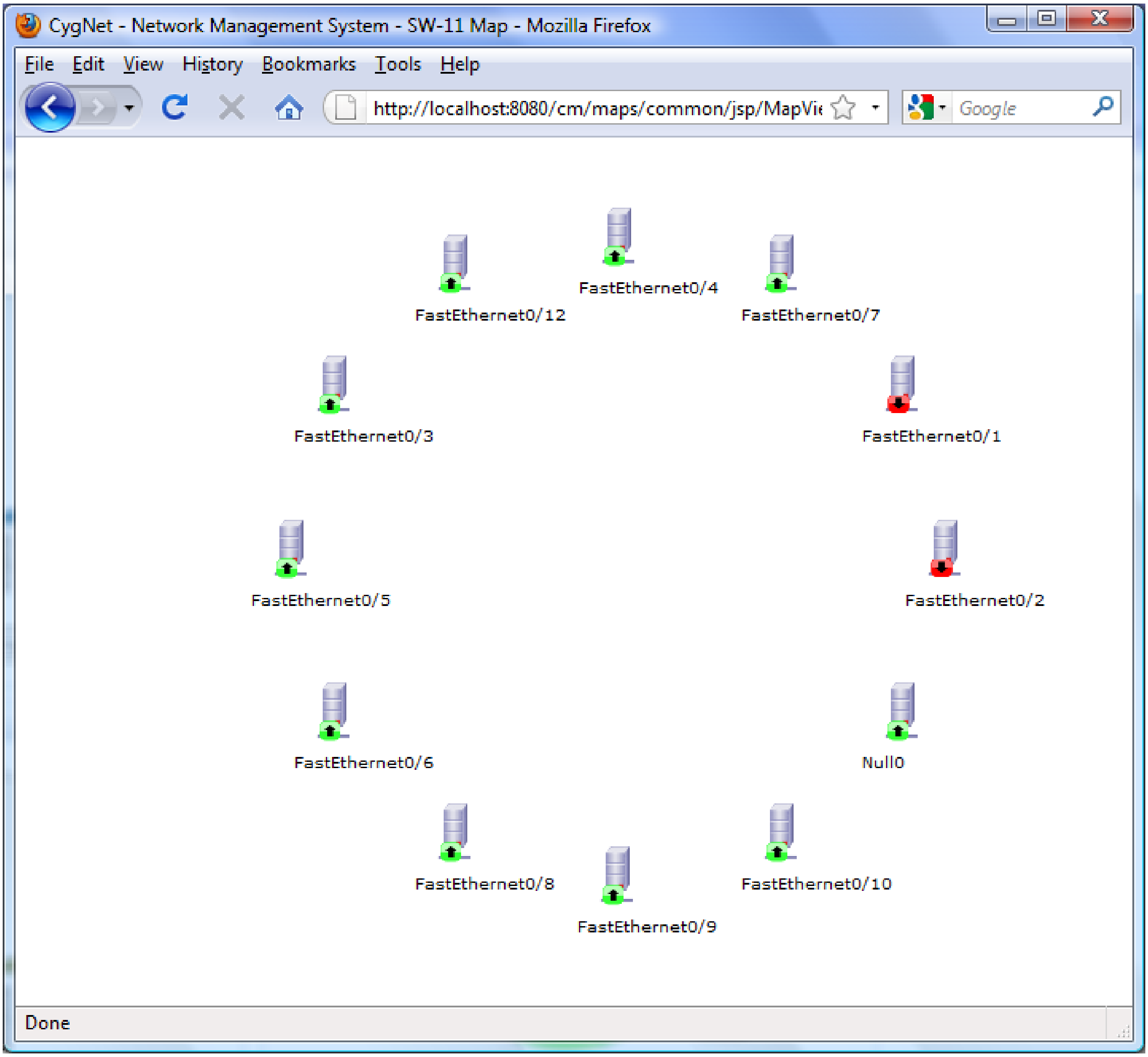

CygNet - Network Management System - NMS Map - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/cm/maps/common/jsp/MapVie
Google
Network
192.168.8.0
Network
ISP
L2
RTR-1
RTR-2
Network
192.168.7.0
L1
L6
L7
L3
L4
L5
Network
144.16.0.0
Network
192.168.9.0
ISP-RTR
Done

Domain View



CygNet - Network Management System - 192.168.9.0 Map - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/cm/maps/common/jsp/MapVie
Google
H3
H2
H4
H1
H5
H6
SW-11
RTR-1
RTR-2
H8
H7
L11
L12
L13
L14
L15
L16
L17
L18
L19
L20
Done

Interfaces View



Commercial / Open Source NMS & System Solutions

- Low End NMS
- Enterprise NMS
  - Centralized
  - Distributed
- System and Network Management

Notes

- Enterprise NMS
  - Hewlett-Packard OpenView
  - IBM Micromuse
  - NMSWorks CygNet
- Low End NMS
  - SNMPc
  - Open NMS
  - Nagios
- System & Network Management
  - Computer Associates Unicenter TNG
  - Tivoli TME / Netview
  - AdventNet Operations Manager
  - Big Brother
  - Spong

HP OpenView
Network Node Manager

- Autodiscovery and mapping
- Drill-down views
- Fault monitoring
- Event monitoring
- MIB Browser
- SNMP tools
- Traffic monitoring
- 3rd party integration

Notes

- OpenView is Hewlett-Packard’s platform for Network Management
- Many NMSs use OpenView Platform: CiscoWorks, CA TNG, Transcend
- NNM is HP NMS on OpenView
- Drill-down Map Hierarchy

HP OpenView Platform

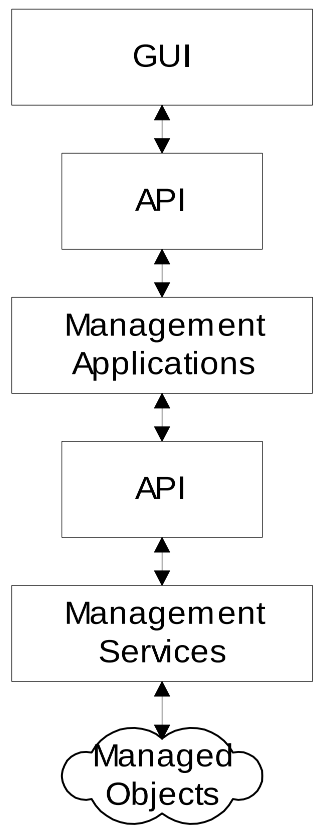
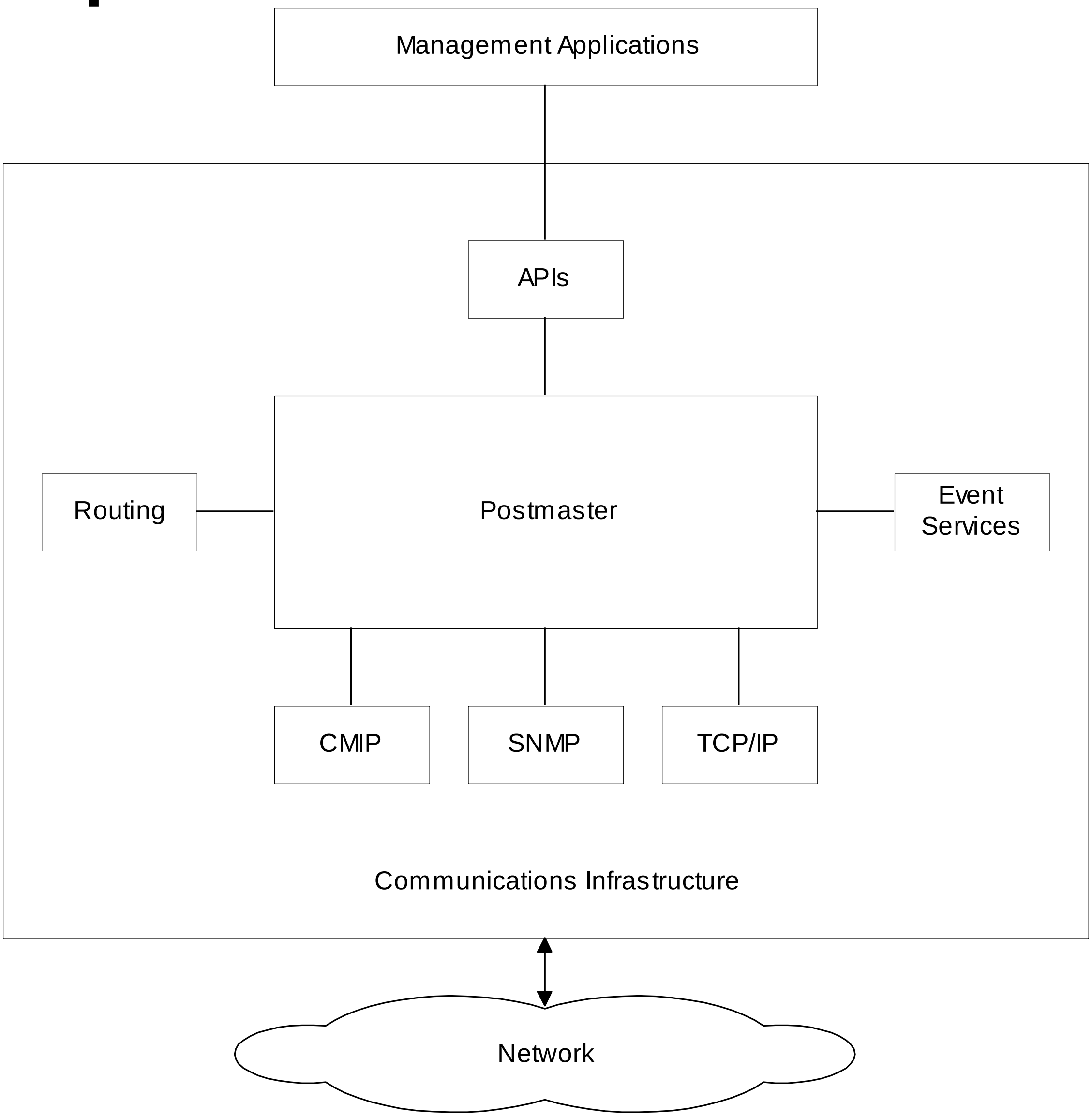


Figure 9.25 HP OpenView Platform Architecture

Notes

- Open, modular, and distributed architecture
- Object-oriented design; TNM can be implemented
- Open API-based architecture
- Easy vendor-specific NMS integration by 3rd party

OpenView Distributed Platform



Notes

- Postmaster integrates all management services
  - multiple protocol stacks: SNMP, CMIP, TCP/IP
- Routing enables distributed message routing
- Event services control event and alarm messages
- Management services interface with management applications via APIs

Distributed OpenView NNMs

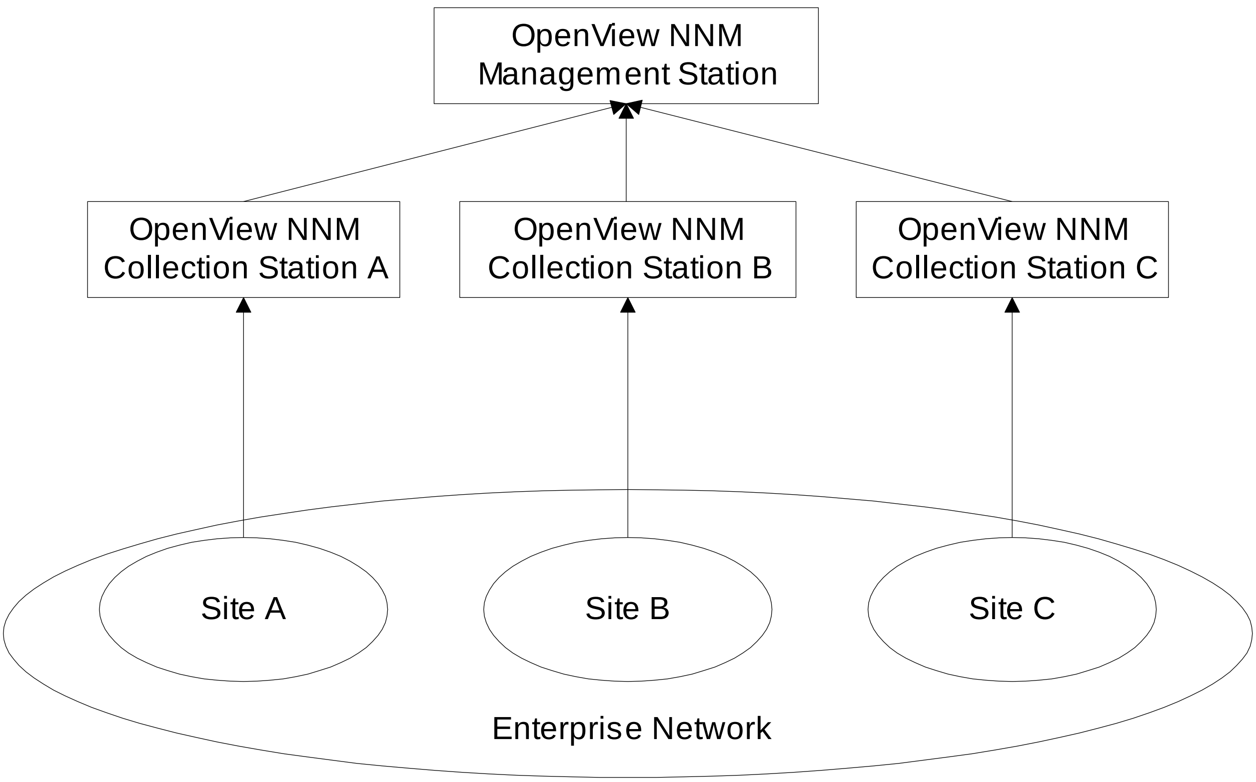


Figure 9.27 HP OpenView Distributed Network Management

Notes

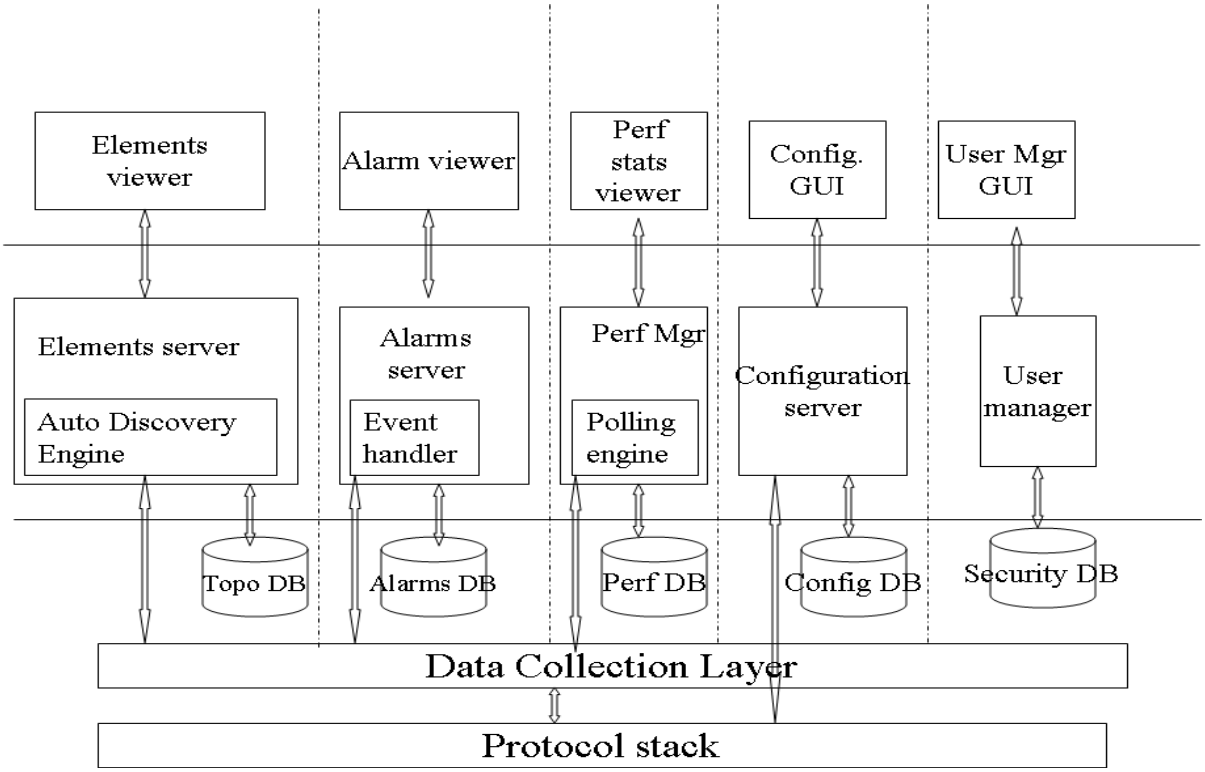
- An enterprise network managed by multiple NMSs
- Collection stations could be regular NNMs or entry-NNMs (100 nodes)
- Filtered information sent to MoM (top level NNM) reduces traffic
- MoM integrates and presents enterprise data

Telecommunications NMS

- Differs from Enterprise NMS – most based on IP
- Features specific to Telecom:
    - Management of network based on legacy equipment
    - Based on ITU-T standard vs. IP
    - NMS necessitated to interface to EMS, not NE
    - Multivendor, multitechnology equipment using diverse protocols
    - Provisioning, such as SDH / SONET optical network complex
    - Dynamic accurate inventory management critical for bandwidth provisioning
    - Importance of QoS / SLA in voice call management
    - Real-time Service Impact Analysis on network failures necessary
    - Telecom networks are large and hence more difficult to manage

Notes

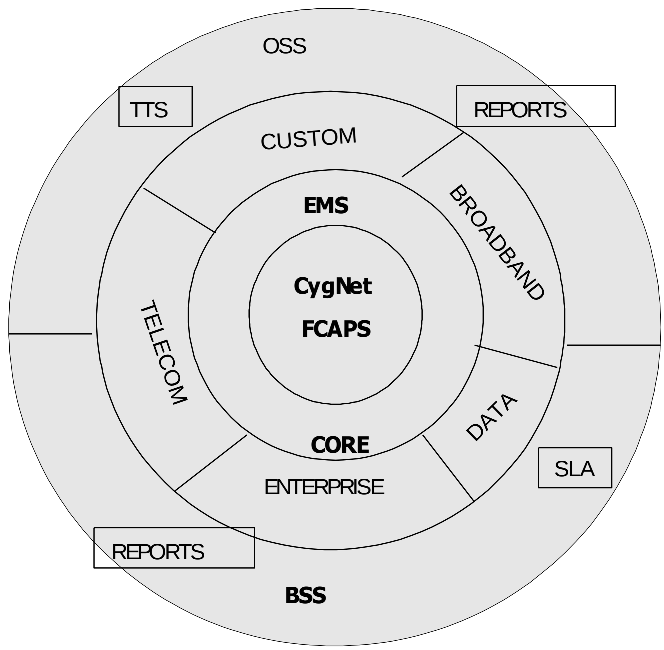
CygNet Core Architecture



Notes

- Elements server discovers NEs and links and stores them in Topo (Topology) database
- Alarm server receives events, does correlation, generates alarms, stores, and sends notifications
- Performance manager gathers telecommunication data such as call data, generates statistics, and stores them
- Configuration manager is a tool to set the configuration of NEs
- User manager system security operations

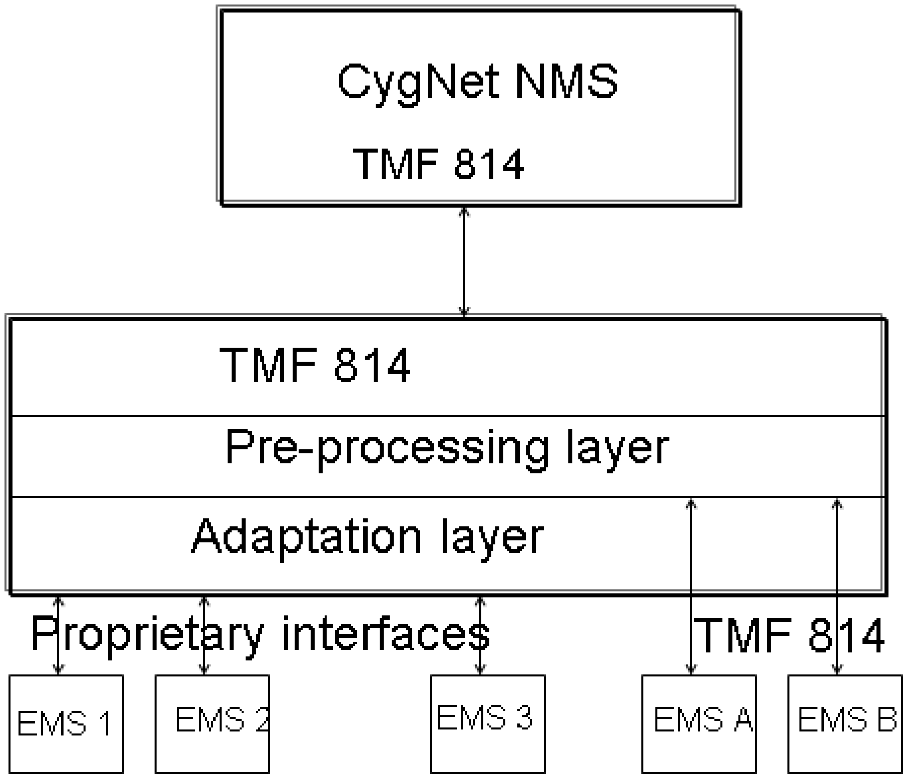
CygNet as Telecom NMS



Notes

- CygNet NMS for telecom management is an application on top of CygNet core module
- Custom module in the architecture enables customization needed for telecom NMS

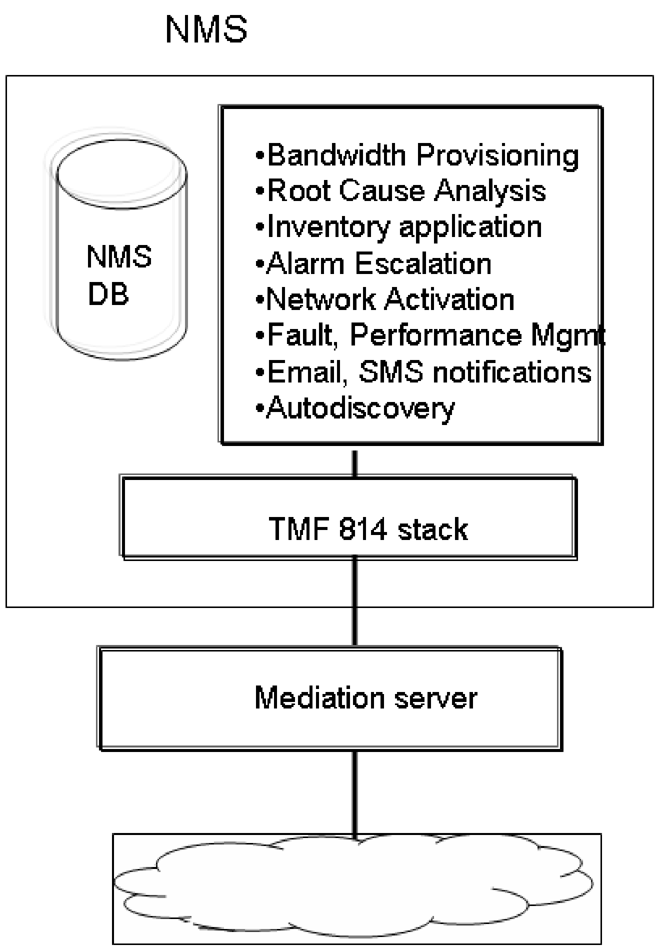
CygNet Mediation Server



Notes

- Mediation server enables handling of multivendor multitechnology NEs
- Cygnet NMS with south-bound standard TMF 814 protocol is transparent to the multivendor multi-technology due to the presence of mediation server

CygNet OTNMS Architecture



Notes

- OTNMS (Optical Transport NMS) manages multivendor multitechnology optical network from service and network fulfillment to service and network assurance