MODELS OF NETWORK AND SYSTEM ADMINISTRATION

Dr.G.Ushadevi
Associate Professor
School of Information Technology and Engineering
VIT University
Vellore

Models of network and system administration

- **System interaction.**
  - *Systems involve layers of interacting (cooperating and competing) components that interdepend on one another.*
  - *Just as communities are intertwined with their environments, so systems are complex ecological webs of cause and effect.*
  - *Ignoring the dependencies within a system will lead to false assumptions and systemic errors of management.*

- **Adaptability.**
  - *An adaptable system is desirable since it can cope with the unexpected.*
  - *When one's original assumptions about a system fail, they can be changed.*
  - *Adaptable systems thus contribute to predictability in change or recovery from failure.*

- **System management's role.**
  - *The role of management is to secure conditions necessary for a system's components to be able to carry out their function.*
  - *It is not to direct and monitor (control) every detail of a system.*

Summary

- The structuring of organizational information in directories.
- The deployment of services for managing structural information.
- The construction of basic computing and management infrastructure.
- The scalability of management models.
- Handling inter-operability between the parts of a system.
- The division of resources between the parts of the system.

Information models and directory services

- **Directory service.**
  - *A collection of open systems that cooperate to hold a logical database of information about a set of objects in the real world.*
  - *A directory service is a generalized name service*

Differences

- Directories
  - organized in a structured fashion
  - hierarchical (tree structure)
  - employing an object-oriented model.
- Directory services
  - employ a common schema
  - for what can and must be stored about a particular object
  - so as to promote inter-operability.

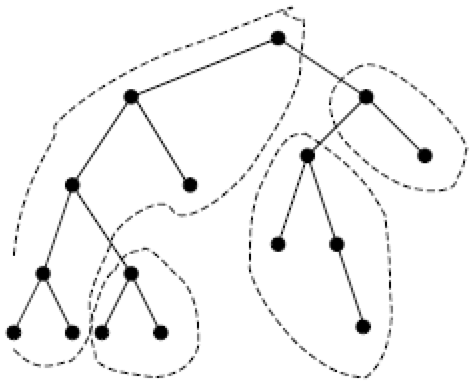
X.500 information model

- X.500
  - ISO9594 creating an standard for directories called X.500
  - Another standard (ASN) is to define formatted protocols
  - Directory Access Protocol (DAP) for addressing a hierarchical directory, with powerful search functionality.
  - Since DAP is an application layer protocol, it requires the whole OSI management model stack of protocols in order to operate.

Directories

- Attributes
  - Entries are name-value pairs
  - Single-valued or Multi-valued
  - It has a syntax
  - Set of sub-attributes
  - Includes matching-rules
- Entries of X.500 are arranged in hierarchical tree forming a Directory Information Tree (DIT)
- entry is identified by its *Distinguished Name (DN)*
- *joining a Relative Distinguished Name (RDN) with those of all its parents, back to the top class*

- Directory Information Tree is partitioned into smaller regions, each of which is a connected subtree, which does not overlap with other subtree partitions
- *A master server within each partition keeps master records and these are replicated on slave systems*
- The partitioning of a distributed directory. Each dotted area is handled by a separate server



- **Directory User Agent (DUA)**
  - *A program or subsystem that queries a directory service on behalf of a use*

Unix legacy directories

- Unix hosts stored directory information in the /etc file directory
- bind hosts together with a common directory for all hosts in a Local Area Network.
- Network Information Service(NIS) directory
  - very popular
  - but was both primitive, non-hierarchical and lacked an effective security model
- NIS+
  - add strong authentication to queries
  - allow modernized and more flexible schema
- Open standard LDAP

OpenLDAP

- Unix-like systems
- Directory information can be accessed through a variety of agents
- can be added to the Unix name server list via nsswitch.conf and Pluggable Authentication Modules (PAM)
- The strength of LDAP
  - versatility and interoperability with all operating systems
- Disadvantage
  - somewhat arbitrary and ugly syntactical structure
  - its vulnerability to loss of network connectivity.

Novell Directory Service – NDS

- Novell Netware / Network Operating System (NOS)
- centralized sharing service with common disk and a common printer
- thus allowing expensive hardware to be shared amongst desktop PCs.
- The Novell directory keeps information about all devices and users within its domain: users, groups, print queues, disk volumes and network services
- In Novell, a directory does not merely assist the organization:
- the organization *is a directory that directly implements the information model of* the organization

Active Directory – AD

- directory service introduced with and integrated into Windows 2000
- replaces the Domain model used in NT4, and is based on concepts from X.500
- LDAP compatible
- smallest LDAP partition area in Active Directory is called a *domain to provide a point of departure for NT4 users*
- *The schema in Active Directory differ slightly from the X.500 information model*

System infrastructure organization

- a network is a community of cooperating and competing components
- A system administrator has to choose the components and assign them their roles on the basis of the job which is intended for the computer system
- There are two aspects:
  - machine aspect
    - relates to the use of computing machinery to achieve a functional infrastructure
  - human aspect
    - about the way people are deployed to build and maintain that infrastructure.

1.Team work and communication

- essential bi-directional communications taking place in a variety of forms:
  - Between computer programs and their data,
  - Between computers and devices,
  - Between collaborating humans (in teams),
  - Between clients and servers,
  - Between computer users and computer systems,
  - Between policy decision-makers and policy enforcers,
  - Between computers and the environment

2. Homogeneity

- **Homogeneity/Uniformity I.**
  - *System homogeneity or uniformity means that all hosts appear to be essentially the same.*
  - *This makes hosts predictable for users and manageable for administrators.*
  - *It allows for reuse of hardware in an emergency*

3. **Load balancing**

- **Delegation II.**
- *For large numbers of hosts, distributed over several locations,*
- *a policy of delegating responsibility to local administrators*
- *with closer knowledge of the hosts' patterns of usage minimizes the distance between administrative center and zone of responsibility.*
- *Zones of responsibility allow local experts to do their jobs*

4. Mobile and ad hoc networks

- An ‘ad hoc’ network (AHN) is defined to be a networked collection of mobile objects, each of which has the possibility to transmit information.
- The union of those hosts forms an arbitrary graph that changes with time.
- The nodes, which include humans and devices, are free to move randomly
- thus the network topology may change rapidly and unpredictably.

5. Peer-to-peer services

- **Peer-to-peer application.**
- *A peer-to-peer network application is one in which each node, at its own option, participates in or abstains from exchanging data with other nodes, over a communications channel*

Network administration models

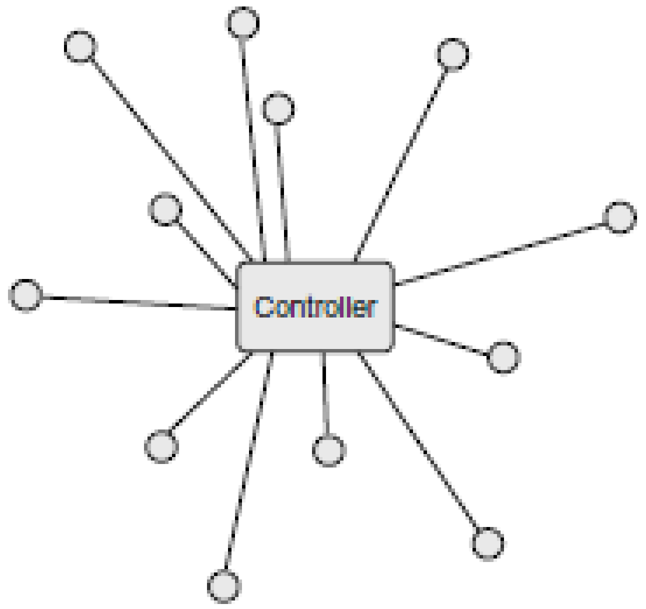
Model.1Central management 'star' model

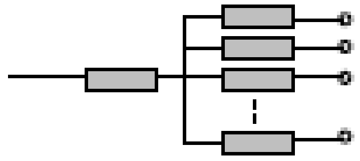
- Idealized model of host configuration based on remote management
- Central manager decides and implements policy from a single location and all networks and host are reliable.
- Manager monitors the whole network using bidirectional communication (N:1)
- reliability in any component of system
- Request service capacity of the controller is

  $I_{controller} = I_1 + I_2 + I_3 \ldots\ldots\ldots + I_N$

- Controller current cannot exceed its capacity Cs
- Assume controller puts flow of repair instructions at its full capacity

$I_{repair} = c_{s / N}$

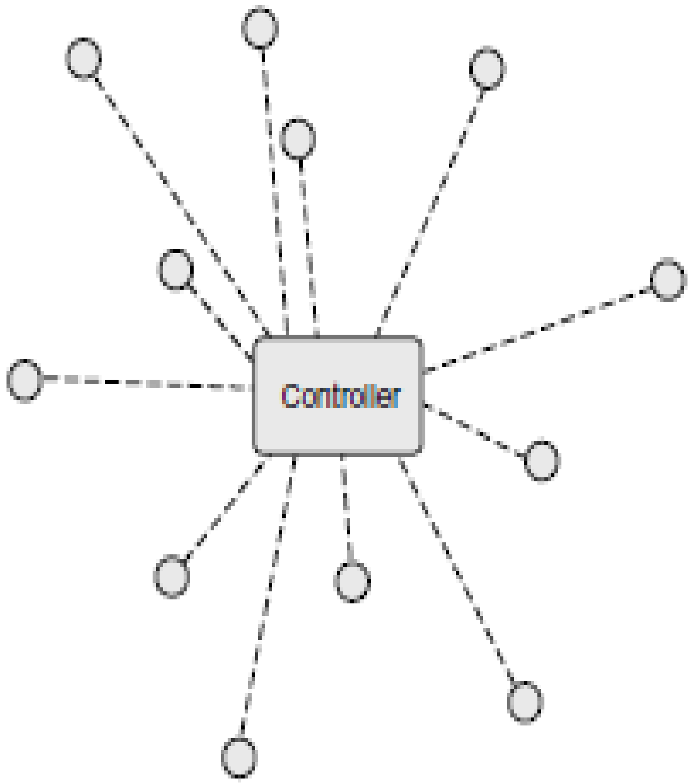




Model 1: the star network. A central manager maintains bi-directional communication with all clients. The links are perfectly reliable, and all enforcement responsibility lies with the central controller.

Model 1 and 2 : Star model in intermittently connected environment

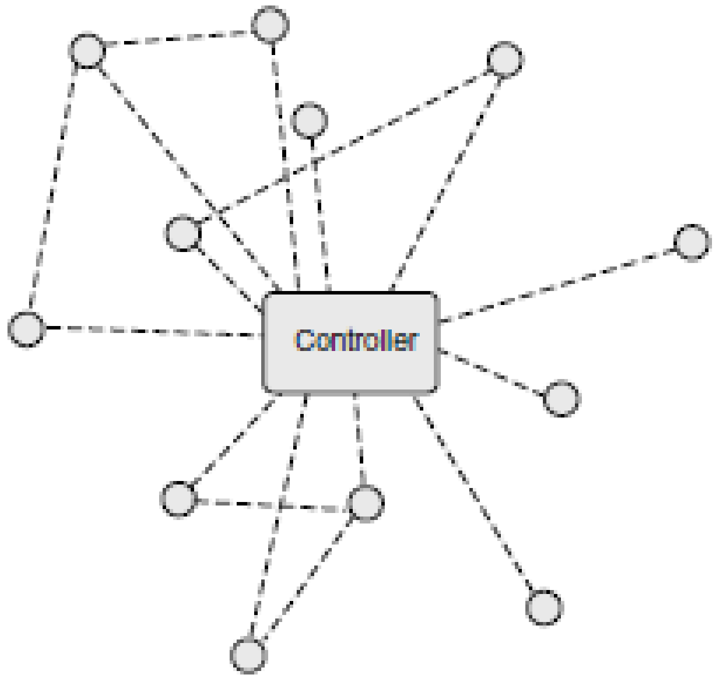
- Realistic centralized management takes account the unreliability of environment
- Partially reliable links , a remote communication model bears risk of not reaching every host
- Capacity of central manager Cs is shared between average no. of host(N)
- Model fails on average at threshold value N as model 1



Model 2: a star model with built-in unreliability. Enforcement is central as in Model 1.

Model 3: Mesh topology with centralised policy and local enforcement

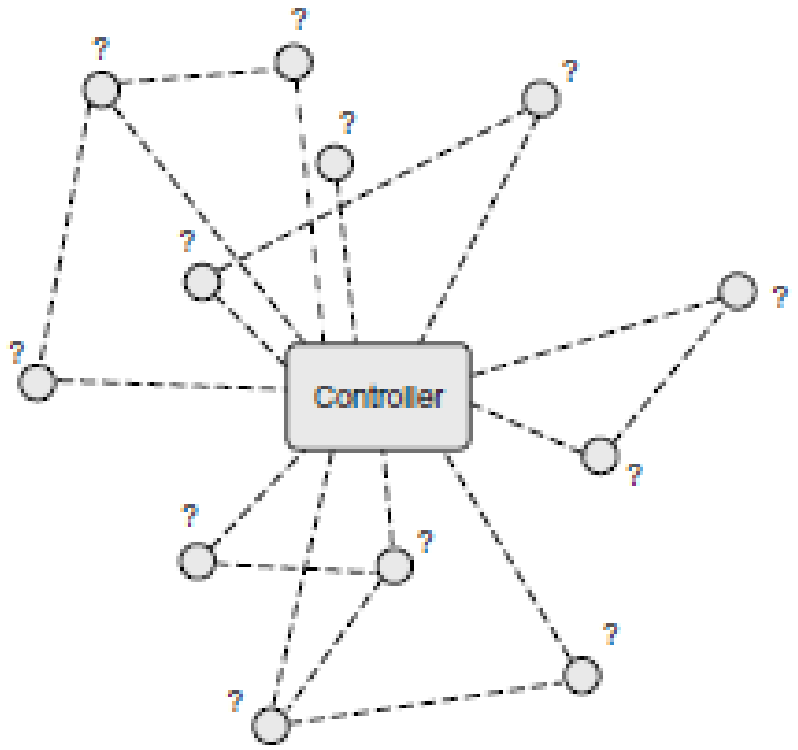
- Download a summary of policy to each host and empower the host itself to enforce it
- Each host carries the responsibility of configuring itself
- Two issues:
  - Update of policy : pull model
  - Enforcement of policy
- Policy to contain self- referential rule for updating
- Load on controller is smaller
- Nodes can cooperate in diffusing policy updates via flooding



Model 3 mesh topology. Nodes can learn the centrally-mandated policy from other nodes as well as from the controller. Since the mesh topology does not assure direct connection to the controller, each node is responsible for its own policy enforcement.

Model 4 : Mesh Topology, partial host autonomy and local enforcement

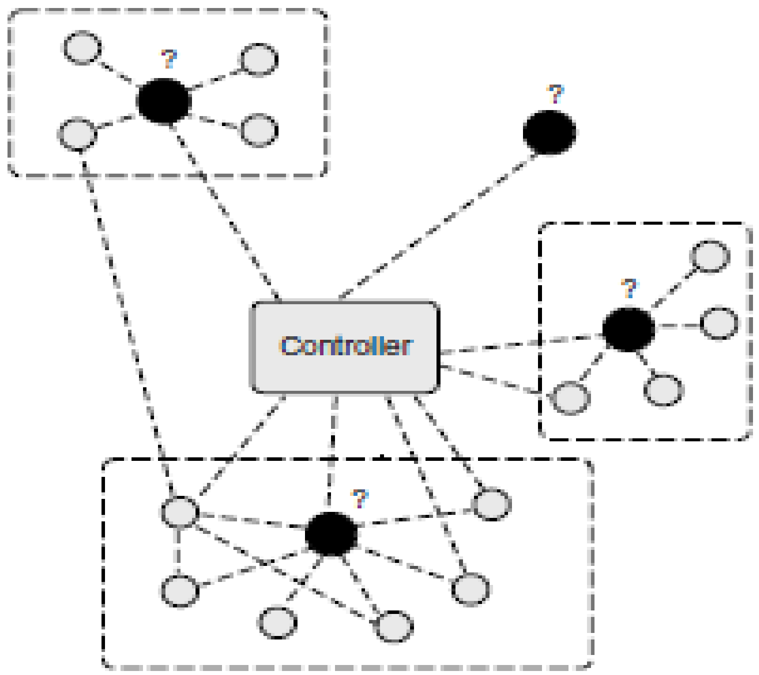
- Allows host to decide their own policy, instead of being dictated
- Host can choose not to receive policy from central authority if it conflicts with local interests
- Suggestions from central authority in the form of latest version of the policy
- Communication and enforcement use distinct channels
- Works to arbitarily large N
- Used by cfengine



Model 4. As in Model 3, except the hosts can choose to disregard or replace aspects of policy at their option. Question marks indicate a freedom of hosts to choose.

Model 5: Mesh ,with partial autonomy and hierarchical coalition

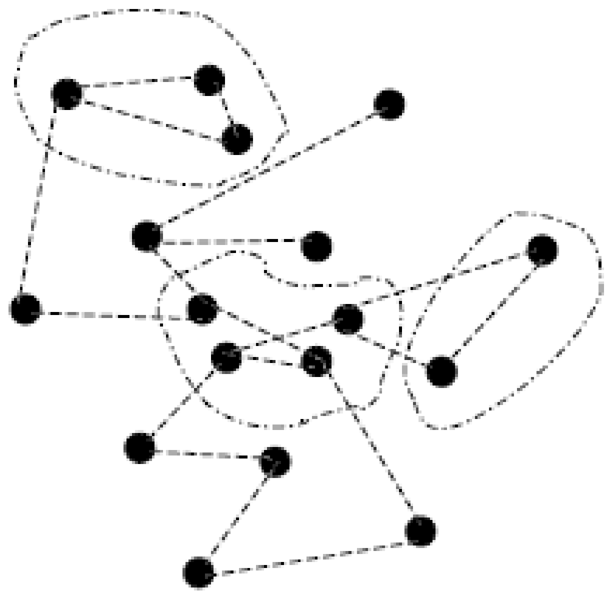
- To allow local group of host to form policy coalitions
- If policies are public then scaling argument of Model 3 applies since any host cache any policy, now policy must be assembled from several sources
- Distribute policy to avoid contention in bottlenecks
- Central source is protected from maximum loading by delegating local policy



Model 5. Communication over a mesh topology, with policy choice made hierarchically. Sub-controllers (dark nodes) edit policy as received from the central controller, and pass the result to members of the local group (as indicated by dashed boxes).

Model 6: Mesh , with partial autonomy and inter-peer policy exchange

- Free exchange of information between arbitrary hosts
- Example of collaborative network is the open source community
- No center , scale free by design; all interactions are local
- Can be made to work at small system size and will also work at any large size
- Have the greatest freedom to explore the space of possible policies
- No immediate scaling problems with respect to communication and enforcement



Model 6. Free exchange of policies in a peer-to-peer fashion; all nodes have choice (dark). Nodes can form spontaneous, transient coalitions, as indicated by the dashed cells. All nodes can choose; question marks are suppressed

SNMP TOOLS

Installing SNMP Tools

- need a set of tools that can make SNMP queries
  - Query a variable and view the response
  - Set a variable and determine if it was successful
  - Query entire tables with get-next-request
  - Receive traps

- widely used set of very good SNMP tools is the net-snmp package
- there are binary distributions available of version 5 for Linux and binary distributions of version 4 available for Linux, Solaris, HPUX, FreeBSD, Irix, and Windows

Building from Source

```
#gtar zxvf net-snmp-5.0.8.tar.gz
#./configure
```

- Default SNMP Version.
  - 1, 2, or 3. The default answer is version 3.
- System Contact Information.
  - It is often set to the email address of the administrator.
- System Location.
  - It should be set to the physical location of the device.
- Logfile Location.
  - This specifies the name of the file to which net-snmp will send logging information and error messages.
- Snmpd Persistent Storage Location.
  - This is the name of the directory where net-snmp will keep statefull configuration files.

Build and Install

# make

- to build the entire package.

#./configure --prefix=/usr/local/mydirectory

# make install

- By default, this will install into the following directories in /usr/local/:
  - bin/: net-snmp applications
  - sbin/: net-snmp daemons (snmpd, snmptrapd)
  - share/snmp/: net-snmp configuration data
  - share/snmp/mibs/: MIB files
  - lib/: net-snmp programming libraries
  - include/: net-snmp programming includes
  - man/: net-snmp man pages

Using SNMP Tools

- the program name is executed without a full pathname, assuming that /usr/local/bin/ and /usr/local/sbin/ are in your path.
- If they are not, you will need to type the full path to the program, as in:

```
#/usr/local/bin/snmpget -h
```

Snmpget

- The snmpget program built by net-snmp can be used to retrieve the value of an SNMP variable.

```
#snmpget -v 1 -c public switch.example.com \
system.sysUpTime.0
SNMPv2-MIB::sysUpTime.0 = Timeticks:
(405064255) 46 days, \ 21:10:42.55
```

Options Common to snmpget, snmpset, and snmpwalk

Argument	Function
-h, --help	Print the help message.
-H	Print configuration file options relevant to this program.
-V, --version	Print net-snmp software version number.
-v version	Specify SNMP version. Must be 1, 2c, or 3.
-c community	Specify the SNMP community.
-r retries	Specify the number of times to retry requests.
-t timeout	Specify the number of seconds to wait for a response.
-d	Dump SNMP packets in hexadecimal.
-D module[,...]	Debugging for the specified modules. Try ALL.
-m mib[:mib...]	Load named MIBs. Use ALL for everything.
-M dir[:dir...]	Include named directories when looking for MIBs.
-P suboptions	Set options for MIB parsing.
-O suboptions	Set output options.
-I suboptions	Set input options.
-C suboptions	Set application-specific options.

Output Options for snmpget, snmpset, and snmpwalk

Arg	Function
a	Print all strings in ASCII format
b	Do not break OID indexes down
e	Print enums numerically
E	Escape quotes in string indices
f	Print full OIDs on output
n	Print OIDs numerically
q	Quick print for easier parsing
Q	Quick print with equal-signs
s	Print only last symbolic element of OID
S	Print MIB module-id plus last element
t	Print timeticks unparsed as numeric integers
T	Print human-readable text along with hex strings
u	Print OIDs using UCD-style prefix suppression
U	Don't print units
v	Print values only (not OID = value)
x	Print all strings in hex format
X	Extended index format

Snmpset

- The snmpset command can be used to set the value of a writable SNMP variable.
- For example, if we wish to set the system contact on a device

```
# snmpset -v 1 -c really-secret switch.example.com \
system.sysContact.0 s admin@example.com SNMPv2-
MIB::sysContact.0 = STRING: admin@example.com
```

Value Type Arguments for snmpset

Arg	Variable Type
i	Integer
u	Unsigned integer
t	Timeticks
a	IP address
o	Object ID
s	String
x	Hexadecimal string
d	Decimal string
b	Bits
U	Unsigned 64 bit integer
I	Signed 64 bit integer
F	Float
D	Double

Snmpwalk

- The snmpwalk command provides a useful way to retrieve a contiguous segment of variables from a device.
- It uses the get-next-request PDU type to continue requesting the next variable until the entire segment is retrieved.
- For example, the entire system group can be obtained with:

```
# snmpwalk -v 1 -c public switch.example.com system
SNMPv2-MIB::sysDescr.0 = STRING: Cabletron Systems, Inc. ... SNMPv2-
MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises ... SNMPv2-
MIB::sysUpTime.0 = Timeticks: (690848548) 79 days, ... SNMPv2-
MIB::sysContact.0 = STRING: admin@example.com SNMPv2-
MIB::sysName.0 = STRING: switch.example.com
SNMPv2-MIB::sysLocation.0 = STRING: 5-125T
SNMPv2-MIB::sysServices.0 = INTEGER: 71
```

```
#snmpwalk -v 1 -c public router.example.com \
  ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaPhysAddress
#snmpwalk -v 1 -c public router.example.com
  enterprises
```

Snmptrapd

- The snmptrapd program is a daemon that listens for SNMP traps and either logs the messages to syslog or stores them in a file.
- If run with no arguments, it will send the messages to syslog by default.

```
# snmptrapd
# snmptrapd -o /var/tmp/trapd.log
```

Other Tools

- snmpgetnext:
  - Perform a single get-next-request query
- snmpbulkget:
  - Perform a bulk-get request (not for SNMPv1)
- snmpbulkwalk:
  - Perform a bulk-get walk (not for SNMPv1)
- snmpd:
  - An SNMP listening daemon
- snmpdelta:
  - Real-time monitor for integer values
- snmpdf:
  - Retrieve disk information from remote workstations

- snmptrap, snmpinform:
  - Send an SNMP trap
- snmpnetstat:
  - Produce network information, as in netstat
- snmpstatus:
  - Print general device status information
- snmptable:
  - Produce a nicely formatted SNMP table
- snmptest:
  - For SNMP debugging
- snmptranslate:
  - Print detailed MIB information

```
#snmpnetstat -r -v 1 -c public router.example.com
#snmpdelta -c public -v 1 switch.example.com ifInUcastPkts.6
#snmptable -v 1 -c public switch.example.com ipNetToMediaTable
```

Maintaining SNMP Tools

- The net-snmp package requires little maintenance.
- Occasionally, you may wish to add a MIB or update the software, but other than that, there is no routine maintenance necessary

Network Management Technologies

- **SNMP network Management**
  - Simple gateway monitoring protocol
  - SNMP exists in 3 versions
  - SNMP architecture is based on 2 entities : managers and agents
    - Managers execute management application
    - Agents mediate access to management variables
  - Management Information Base : Set of all variables on a managed system

- Version 3 - strong encryption methods
- SNMP supports 3 operations : read, write and notify
- Since SNMP is a ‘peek –poke’ protocol for simple values it depends on ability of MIB to correctly characterize the state of devices and how agents translate values into real actions
- Variables in MIB are defined in MIB modules that are written in a language called Structure of Management Information (SMI)
  - SMI provides generic and extensible namespace for identifying MIB variables
- SNMP request specifies information it wants to read/write by giving name of an instance of variable to read/write in request
  - Standard modules for address translation tables ,TCP/IP statistics and so on

TMN and others

- International Telecommunications Union(ITU) has defined Telecommunications Management Network (TMN) for managing Telecommunications Network
- Common Object Request Broker Architecture(CORBA) adopted as middleware for TMN
- Distributed Management Task Force(DMTF) developed Desktop management Interface(DMI)
- These systems use an abstraction based on the concept of ‘managed objects’
- Cfengine and PIKT use descriptive languages to describe the attributes of many objects at same time and agents to enforce rules

OSI

- ISO Open system Interconnect (OSI) model consist of documents describing aspects of network communication and management
- Basic conceptual model for management of networked computers consist of
  - Configuration management : change control, hardware inventory mapping, software inventories
  - Fault management : events, alarms, problem identification, trouble shooting
  - Performance management : capacity planning, availability, response time, accuracy
  - Security management : policy, authorizations, exceptions , logging
  - Accounting management : asset management, cost controls

Java Management Extension (JMX)

- JMX is to deal with managed objects
- Similar to SNMP but the transport mechanisms are integrated into Java’s extensive middleware framework.
- MX defines a standard instrumentation model, called MBeans, for use in Java programs and by Java management applications
- Mbean : A Java object that implements a specific interface and conforms to certain design patterns
- JMX also specifies a set of complementary services that work with MBean instrumentation to monitor and manage Java-based applications

Jini and UPnP: management-free networks

- Jini
  - a Java derivative technology that is aimed at self-configuring ad hoc networks
  - It is middle-ware that provides application programming interfaces (API) and networking protocols for discovery and configuration of devices that have only partial or intermittent connectivity.
- Microsoft's Universal Plug'n'Play (UPnP)
  - a peer-to-peer initiative that uses existing standards like TCP/IP, HTTP and XML to perform a similar function.
  - The aim of these technologies is to eliminate the need for system administrators, by making devices configure themselves.
- Both address the area of device connectivity and ability to dynamically make use of new devices on the network

Creating infrastructure

- **Principles of stable infrastructure**
- **Scalability**
  - Any model of system infrastructure must be able to scale efficiently to large numbers of hosts (and perhaps subnets, depending on the local netmask).
- **Reliability**
  - Any model of system infrastructure must have reliability as one of its chief goals. Down-time can often be measured in real money.
- **Redundancy**
  - Reliability is safeguarded by redundancy, or backup services running in parallel, ready to take over at a moment's notice
- **Homogeneity/Uniformity II**
  - A model in which all hosts are basically similar is i) easier to understand conceptually both for users and administrators, ii) cheaper to implement and maintain, and iii) easier to repair and adapt in the event of failure.

- **Virtual machine model**
- a small step from viewing a multitasking operating system as a collaboration between many specialized processes, to viewing the entire network as a distributed collaboration between specialized processes on different hosts
- sites adopt specific policies and guidelines in order to create this seamless virtual environment by limiting the magnitude of the task
- Tools such as make, which have been used to jury-rig configuration schemes can now be replaced by more specific tools like cfengine

Creating uniformity through automation

- If we want hosts to have the same software and facilities, creating a general uniformity, we need to employ automation to keep track of changes
- formalize needs by writing them in the form of a policy, program or script, and have automatic reproducibility
- **Abstraction generalizes**
  - Expressing tasks in an operating system independent language reduces time spent debugging, promotes homogeneity and avoids unnecessary repetition.
- **Platform independent languages**
  - Use languages and tools which are independent of operating system peculiarities, e.g. cfengine, Perl, python

- Perl
  - Disadvantage:
    - it is a low-level programming language, which requires us to code with a level of detail which can obscure the purpose of the code.

- Cfengine
  - It is a very high-level interface to system administration.
  - It is also platform independent, and runs on most systems.
  - advantage
    - it hides the low-level details of programming, allowing us to focus on the structural decisions

Revision control

- created as a repository for files, where changes could be traced through a number of evolving versions
- introduced as a tool for programmers, to track bug fixes and improvements through a string of versions
- CVS system is an extended front-end to this system
- useful way of keeping track of text-file changes, but it does not help us with other aspects of system maintenance, such as file permissions, process management or garbage collection

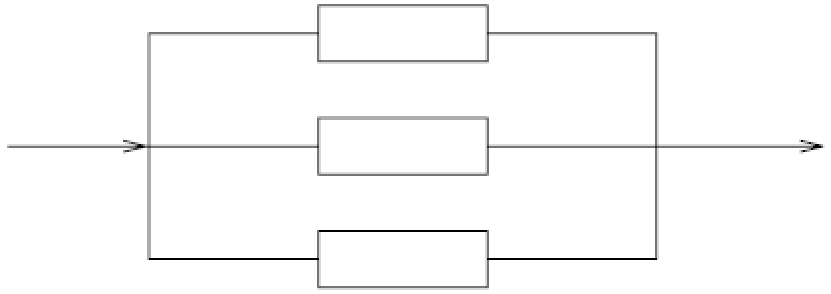
Push models and pull models

- two types of distribution mechanism
  - *1. Push:*
    - *The push model is epitomized by the rdist program.*
    - *pushing files* from a central location to a number of hosts is a way of forcing a file to be written to a group of hosts
    - central repository decides when changes are to be sent, and the hosts which receive the files have no choice about receiving them
    - **Advantage** : more easily optimized than a pull approach
    - **disadvantage** of a push model is that hosts have no freedom to decide their own fate.

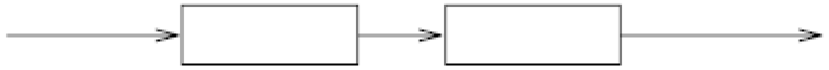
- 2. Pull :
  - represented by cfengine and rsync
  - each host decides to collect files from a central repository, of its own volition.
  - **Advantage** : there is no need to open a host to control from outside, other than the trust implied by accepting configuration files from the distributing host
  - **Disadvantage** : optimization is harder.
    - rsync addresses this problem by using an ingenious algorithm for transmitting only file changes, and thus achieves a significant compression of data,
    - cfengine uses multi-threading to increase server availability.

Reliability

- Failure can encompass hardware and software which includes downtime due to physical error (power, net cables and CPUs) and also downtime due to software crashes but its possible to *prevent failure with* pro-active maintenance
- Component failure can be avoided by parallelism, or redundancy.
- flow of service can continue when servers work in parallel, even if one or more of them fails.
- to employ a *fail-over capability (backup service)*



System components in parallel, implies redundancy



System components in series, implies dependency

System maintenance models

- Unix administrators have run background scripts to perform system checks and maintenance for many years
- Windows can be both easier and harder to administrate than Unix.
  - It can be easier because the centralized model of having a domain server running all the network services, means that all configuration information can be left in one place , and that each workstation can be made to configure itself from the server's files.
- **Reboot**
- **Manual administration**
- **Central control**
- **Immunology (Self-maintenance)**

Policy and configuration automation

- Cfengine and PIKT are system administration tools consisting of two elements: a language and a configuration engine.
- Cfengine : defining the way we want all the hosts on our network to be configured, and having them do the work themselves
- PIKT: Problem Informant/Killer Tool - allows a mixture of declarative and imperative programming to define host policy.
  - tools are for automation and for definition.
- Include language for describing system configuration at a high level, they can also be used to express system policy in formal terms
- Cfengine satisfies the principle of convergence
- Policy-based administration works from a central configuration, maintained from one location.
- The work of configuration and maintenance is performed by each host separately
- a cfengine or PIKT policy, scales trivially with the number of hosts, or put another way, the addition of extra hosts does not affect the ability of other hosts to maintain themselves

Integrating multiple OSs

1.Compatible naming

- Different operating systems use quite different naming schemes for objects
- The Internet URL naming scheme has created its own naming scheme for objects, which takes into account the service or communications channel used to access the object:
  - Channel://Object-name
- The Internet Protocol family uses a hierarchical naming scheme encoded into IP addresses.
- The general problem of naming objects in distributed systems has great importance to being able to locate resources and express their locations.

2.Filesystem sharing

- Sharing of file systems between different operating systems can be useful in a variety of circumstances

3. User IDs and passwords

- If sharing across such different operating systems, we need to have common usernames on both systems.

4. User authentication

- The password mechanisms for Unix and Windows are completely different and basically incompatible
- Passwords are incompatible between Windows and Unix for two reasons:
  - NT passwords can be longer than Unix passwords and the form of encryption used to store them is different.
  - The encryption mechanisms which are used to store passwords are one-way transformations, so it is not possible to convert one into the other.

- Configuration management is the administration of *state in hosts* or network hardware
- Host state is configured by a variety of methods:
  - Configuration text file
  - XML file
  - Database format (registry)
  - Transmitted protocol (ASN.1)