OpenView and Hostname Resolution

Integration Concepts

DNS and OpenView

  In order for any network and systems management product such as HP OpenView Network Node Manager (NNM) to function properly there are two services that must be properly configured. These two services are Domain Name Service (DNS) and Simple Network Management Protocol (SNMP).

  This paper will give a basic outline on how Network Node Manager uses these two services, how additional products are affected when they are properly and improperly configured, and how the services should be properly configured. The paper is based on the writers' experience as an OpenView Consultant and a DNS hostmaster.

Introduction

In the today's Information Technology market, more IT managers are spending hundreds of thousands of dollars, some millions of dollars, in order to "manage" their networks. Managers are insipid because the chosen products hostnames are not "synchronized". The management of IP based networks require proper hostname and address resolution. Within an IP managed network there are three ways for systems to resolve hostnames and IP addresses. They are:

- Host file
- Network Information Service (NIS)
- Domain Name Service (DNS)

This document instructs how and why to properly configure any of the three for use in proper hostname resolution. The aim is to compare how each resolves hostnames and IP addresses and how, when improperly configured, it affects network and systems management products.

Acknowledgement.

This document has been achieved in collaboration with the following folks:
- Paul A. Weber, Senior Technical Consultant, Hewlett-Packard Consulting
- Christopher Devita, Hewlett-Packard Response Center Escalation Team

Scope

We discuss the implications of improper DNS configuration and how it impacts Network Node Manager (NNM), IT/Operations (ITO), any product that synchronizes with Network Node Manager's object database, as well as the affect it has on network security.

Network Node Manager and Discovery

The majority of all systems or network devices on a network come with an Simple Network Management Protocol (SNMP) agent in order to "manage" the devices from a central console. These SNMP agents are SNMP MIB-II compliant. To be MIB-II compliant an agent must provide specific information about the node. This includes certain information about the system, interface, tcp, ip, ip routing, and more. Network Node Manager as well as software specifically developed by a vendor for the management of its products use this information within the MIB-II tree as a starting point for management. The system Object ID (sysObjectId) falls under the MIB-II tree. The sysObjectId can be used by the vendor to determine the type of node, the vendor, and/or the operating system.

Network Node Manager is the "baseline" product used by many management products. The majority of vendors use NNM's discovery process to discover the nodes on the network and determine what types of nodes they are. This is done by netmon reading the arp cache of the local router (via SNMP) and determining if those nodes are available on the network. NNM (i.e. netmon) will try to determine if the nodes are available by sending an ICMP ping to the node and if it responds netmon will try to determine the type of node (via SNMP) by reading the nodes' sysObjectId. Once the sysObjectId is determined via SNMP, NNM will compare the sysObjectId to files named oid_to_type and oid_to_sym located in the $OV_CONF directory. Both files can be updated manually, but are generally updated by applying third party applications that integrate with NNM for the specific management of a vendors' products.

The oid_to_type file is used to map the SNMP sysObjectId of a node into the correct IP topology attributes, as well as the correct vendor and SNMPAgent values for use with ovw and ovwdb. The node can be assigned specific attributes via the file such as bridge, gateway, hub and how specific attributes are to be handled: ignore the device with regard to SNMP, the device is

not a gateway even if it has multiple interfaces. For more information on this file and additional attributes, see the oid_to_type man page.

The oid_to_sym file is used by the ipmap process to map the SNMP sysObjectId of a node to the default ovw symbol type used to represent the node of the IP topology submaps it displays. Symbols are arranged in classes and sub-classes within NNM. NNM also queries the SNMP Agent for the interface information of the node. NNM takes the information on the interfaces, their types (FDDI, IP, X.25, etc.), their IP addresses, netmask, etc. and associates these objects with the node object. All of these together create a single node within the ovw map.

After a node is discovered, its configuration gathered via SNMP, and created within the map with its associated interfaces, it must be given its hostname. The hostname is derived by an inverse address lookup of an interface via the local host file or the Domain Name System (DNS) in the following order: lowest numbered IP address on the node, or SysName via SNMP. If NNM cannot retrieve a name via the hosts file or the DNS it will use the SNMP system.SysName. NNM takes the fully qualified hostname and truncates it from the left to the first dot and uses it for the label of the node object on the ovw map. This brings us to the importance of the accuracy and proper configuration of the hosts file, and DNS. They must both resolve to the same fully qualified hostname. If a valid hostname cannot be found using any of these methods, NNM will use the IP address as the hostname.

If an SNMP agent is not running on a node, NNM cannot determine any information about the node. All it knows it that an interface appeared in an ARP cache in a router that it is managing on the network. If this address belongs to a node that has multiple interfaces (multi-homed) there is only one other way for NNM to determine that this interface is in a node that is multi-homed. This is can only be done via the DNS. If the DNS is not configured in such a manner that shows that all interfaces belong to the same host, then individual interfaces will appear as different nodes in the NNM databases, with no correlation between them.

Third Party Product Integration

This very problem will propagate itself into third party OpenView applications that integrate with NNM and synchronize with the OpenView database. These products too will have multiple nodes in their respective product database that actually belong to the same node. This is due to some products, after synchronization with NNM, translating a different interfaces' IP to a hostname than NNM used to retrieve the hostname. Since the DNS does not reflect that the IP addresses have the same fully qualified host name, no product has any way of knowing the IP addresses actually reside on the same node. Even when the node has an SNMP agent and all the interfaces are associated within NNM as the same node, the hostname in one product will most likely be different than NNM, or even another product. Therefore all automation, replication, etc. is out of sync, and product databases are virtually useless. Customers turn off NNM auto-discovery and third party application synchronization of the OpenView NNM database and go to "manual" operation for each products' database because the products cannot stay in sync. This will continue to occur for each product added to the management system that integrates and synchronizes with the NNM database. The out of sync databases have nothing to do with the products themselves, but is all due to the improper configuration of the DNS for multi-homed systems.

This problem is greatly exacerbated on NNM collection stations. If name resolution on different collection stations does not return the same name and address (because that collection station is using its own /etc/hosts, NIS Domain, or DNS lookups) as the Management Server, then collection station synchronization will not be able to complete.

Network Node Manager, /etc/hosts file, and DNS

Now that the node has been discovered via SNMP and NNM has the IP addresses for the interfaces it its database NNM will assign a hostname via the inverse address lookup of the lowest numbered IP address assigned to an interface of the discovered node. There are can actually be several outcomes to this but the proper configuration host files and the DNS will be dealt with first. Later in this paper are examples of improperly configured hosts files and DNS's and the results.

Host File Configuration

The proper configuration of the UNIX management server is to setup the nsswitch.conf file to use the host file first, then if the lookup is not found use the DNS. One could also use NIS, but since an NIS lookup will be a lookup within the hosts file map, the results can easily be determined from configuration of the hosts file.

The host file generally will generally contain only the host names of the node itself and possibly the default router. Proper configuration of the host file on the management server as well as managed nodes is most important, especially when utilizing IT/Operations in conjunction with the Advance Networking Security Extensions (ANSE). The hosts file can also be used to ensure that agents take a private network path. Within the host file as well as the DNS all IP addresses of the node must resolve to a single (the same) fully qualified hostname. On any UNIX system the host file should be setup using the following standard layout:

IP Address   Fully qualified hostname   short hostname and aliases

For example, an HP-UX system named itov5 in the domain super.weber.net has three interfaces, lan0, lan1, and lan2. Lan0 is the built-in 100BT interface, lan1 is an ATM interface, and lan2 and FDDI interface. In HP-UX 10.x and 11.x all interface IP's and subnet masks as well as the short hostname are configured in /etc/rc.config.d/netconf where Solaris configures interfaces via the hostname.interface-device-name in the /etc directory. The host file for the HP-UX operating system should look like this:

```
10.1.0.11        itov5.super.weber.net        itov5        itov5-lan0
10.1.1.11        itov5.super.weber.net        itov5        itov5-lan1
10.1.2.11        itov5.super.weber.net        itov5        itov5-lan2
```

For Solaris, the host file would look just the same except one would change the interface alias in order that Solaris will configure the interfaces properly during boot time.

```
10.1.0.11        itov5.super.weber.net        itov5        itov5-hme0
10.1.1.11        itov5.super.weber.net        itov5        itov5-hme1
10.1.2.11        itov5.super.weber.net        itov5        itov5-hme2
```

With Solaris, by placing the alias interface name in the appropriate /etc/hostname.interface file, i.e. hostname.hme0 contains itov5-hme0, the operating system will configure each interface with the appropriate IP address. The operating system will find itov5-hme0 in the host file, go to the first column and retrieve the IP address and set the IP to the hme0 interface. This works the same for each interface configured on the Solaris system. Solaris uses the /etc/nodename file to set the short hostname of the system.

The same host file structure applies to NT as well. The host file on NT is located in BOOTDISK:\winnt\system32\drivers\etc\hosts.

Hosts File and an Out-of-Band Network

Some may wish to manage their systems via an out-of-band network in which all management traffic will traverse. All managed nodes and the management server reside on a private network in addition to another network. All hosts on the network are now multi-homed. In order to force an agent such as IT/O or Omniback to use the out-of-band network for communication a line is added to the hosts file with the out-of-band IP and fully qualified name of the management server(s).

The opcinfo file contains the hostname of the management server. When the agent communicates with the server the system will determine the IP of the hostname via the hosts file (due to the configuration of nsswitch.conf). The hosts file contains the IP address of the management servers out-of-band network interface and thus communication will take place from client to server via the out-of-band network. This scenario can be used for backup via an out-of-band network as well.

Configuring DNS

The configuration of the Domain Name Server is most critical. All nodes within a network use this service and both forward and inverse resolution must be correct. In order for a network and systems management product such as HP OpenView Network Node Manger to function properly, all IP addresses for a particular node must inversely resolve to the same fully qualified host name. In addition all forward lookups for the fully qualified host name must return all the IP address assigned on that host.[1]

An alias for particular interface, such as itov5-lan0 should have an additional address record inserted within the DNS. This allows for the forward lookup of the alias for the specific interface on the host. For example, when a telnet session to that alias (itov5-lan0.super.weber.net) is issued, the DNS will return the address record for that specific interface and the user would connect to that interface. Configuring the DNS in this method allows for the connection to a specific interface just like naming all the interfaces to a separate hostname.

Service addresses, as some call them, are addresses that are assigned hostnames that point to a particular service such as an Oracle database server or a DNS. They are assigned either as an IP alias on a particular interface or can be setup as a disparate IP network on a loopback or actual interface. These can be setup in the same manner as an alias for a particular interface. An additional address record is placed in the DNS for a particular service name. The pointer record for the address points to the actual fully qualified host name of the system. As long as the client is configured to use the service name the client will lookup the service host name and the DNS will return that specific IP.

Example: On our node itov5.super.weber.net we have configured a loopback interface lo0:1 with an IP address of 10.1.3.11. The service name is dns2.super.weber.net. Within the DNS it would look like this:

```
db.super.weber.net:
itov5         IN      A       10.1.0.11
itov5-lan0    IN      A       10.1.0.11
itov5         IN      A       10.1.1.11
itov5-lan1    IN      A       10.1.1.11
itov5         IN      A       10.1.2.11
itov5-lan2    IN      A       10.1.2.11
itov5         IN      A       10.1.3.11
itov5-lo0-1   IN      A       10.1.3.11
dns           IN      A       10.1.3.11


db.10:
11.0.1        IN      PTR     itov5.super.weber.net.
11.1.1        IN      PTR     itov5.super.weber.net.
11.2.1        IN      PTR     itov5.super.weber.net.
11.3.1        IN      PTR     itov5.super.weber.net.
```

With the DNS setup in this manner, it does not matter which interface is chosen by NNM to resolve the hostname because the DNS will always be the same host name. It also means that third-party add-on products to NNM will function properly and hostnames will be in sync between all products no matter what interface the product uses for IP to hostname resolution. Depending on the product, a client session will try connect via the first address returned from the DNS, if it is unavailable, it will round-robin to the next until it gets a connection, or go through all the IP's returned and give up if the system was actually unavailable.

The M/C ServiceGuard floating IP is an IP address that is assigned to a specific ServiceGuard package. This package is able to run a node within an M/C ServiceGuard cluster. When the package is running on a cluster node it places an IP address (or IP addresses) on an interface card that is (are) used with the package. These IP addresses are "relocateable" in that they relocate to the node where the package is running. This poses a peculiar scenario because these IP addresses can move from node to node. If a DNS Pointer record is set for this floating IP to point to a particular node as stated earlier, the Pointer record will be invalid if the package is run on a different node within the ServiceGuard Cluster. The writer makes this the

ONLY exception to the rule and the floating IP address record and pointer record may be different node name. *It is HIGHLY recommended that floating IP's be assigned to the higher numbered IP address than those actually assigned to the interface of the nodes to prevent NNM from renaming or finding a new hostname for the node. If the system is a multi-homed host, assign the floating IP's to higher numbered IP address than those assigned to the interface and use the higher numbered IP network on the node.*

[1] The writer of this paper is aware of this setup violating the RFC that states that each interface should have a unique host name within the DNS. This writer also knows that RFC's are NOT standards in themselves, but are actually *documents from which standards are created*. The writer will readdress this issue when the DNS RFC has been given an official standard from a standards body such as POSIX or IEEE versus only an RFC number.

Traceroute

As stated earlier, IT managers spend a lot of money on hardware and software in order to manage their systems; some spend millions of dollars to accomplish this task. By configuring host files and DNS's as per this document, it will increase the reliability of host naming across all Network and Systems Management applications that integrate and synchronize data with HP OpenView NNM. Security will also be enhanced.

There is an argument that prevails between the network folks and the management folks about the setup of the DNS and the functionality of the traceroute command. With the DNS setup as described within this document the traceroute command does not cease to function, but the inverse address lookups always return the same hostname for each interface when traceroute traverses through interfaces of the same router or gateway. This can be replaced by traceroute via SNMP in the NNM product. When used through the NNM graphical user interface, the route is highlighted on the NNM Map including the interfaces the route traversed.

The questions to IT managers: "Do you want hundreds of thousands of dollars of network and systems management hardware and software to function properly without a lot of manual intervention?" or "Do you want one freebee, unsupported command to function properly?"

IT/O Advance Networking Security Extension (ANSE)

When using ANSE it is imperative that hostname resolution within both /etc/hosts and DNS are configured in the manner described in this paper. When the IT/O agent on the client sends a message to the management server, it includes the hostname of the client. This hostname is derived from a hostname lookup on the client. The server takes this hostname from the message and uses it to resolve the IP. If the address matches a node in the node bank, decryption occurs, if there is no match, the message is discarded.

If the client's host file is incorrectly configured i.e. the short hostname is in the second column then the client sends this short hostname in the packet. The management server retrieves the short hostname from the packet and tries to resolve the hostname. The management server most likely will not have all its clients in the host file and will be depending on the DNS for resolution. The DNS will ALWAYS return a fully qualified name, therefore there is no authentication because the short hostname and the fully qualified name do not match (they are not equal to each other). Communication between the two systems is denied.

Actual Customer Implementations and Implications

Now that the interactions between NNM, SNMP, and DNS are known they will be applied to actual customer environments.

Customer #1
It was discovered that not all the system administrators run the SNMP agent that comes with Solaris 2.6 and above. Many administrators disable the agent and some run the agent. Each administrator has reasons for the decision made.

As described earlier, without an SNMP agent running on the node, NNM cannot determine the number of interfaces, sysObjectId, operating system, vendor, etc. In turn, if this node is to be managed via IT/Operations, it will be dragged from the NNM map into the IT/Operations Node Bank. IT/O will try to determine the system type via SNMP or query the NNM database for the information. Since the SNMP agent is not running IT/Operations cannot determine the system type it will default to a machine type of "other" and operating system of "other" for the node. The IT/O administrator will have to manually change the agent configuration to the correct machine type and operating system before pushing the agent to the node for installation.

Without an SNMP agent running on the system, none of the default MIB-II traps will be sent to the management server, therefore none will be forwarded into IT/Operations. The default SNMP traps are cold start, warm start, link down, link up, egp down and authentication failure. No vendor specific traps will be sent either.

Without SNMP agent running on the node, NNM cannot determine if the IP addresses it has discovered are individual nodes or if nodes are multi-homed because of the configuration of the customers' DNS servers. The customer appears to be using the DNS as an inventory system of interfaces on nodes rather than host name to address and address to host name mapping. The customer has devised the following scheme of naming their hosts:

> lan-interface.hostname.domain

On a Sun Solaris system with two interfaces, hme0 and hme1 with a host name of itov5 it looks as two separate hosts within the DNS:

> hme0.it0v5.idc1.oss.weber.net and hme1.itov5.idc1.oss.weber.net

Without an SNMP agent running on this system and the DNS configured this way, it appears as two separate nodes within NNM. If one of these nodes is added to the node bank of IT/Operations and the other interface fails, there will be no SNMP event alert from NNM into the IT/O message browser stating the failure. This is because NNM cannot determine that they are the same system via the DNS or SNMP.

The customer's UNIX architecture uses a loopback interface (lo0:1) and assigns the official hostname of the node to this interface i.e. itov5.idc1.level3.com. By assigning the IT/O agent to bind to this address for nodes within the Node Bank of IT/Operations, there will be no SNMP messages from NNM on the status of the interfaces of the node. This once again is because the interfaces are not associated with each other within the DNS or via SNMP. Within the DNS this is defined as an individual host.

Without an SNMP agent running on each node one is limited to the types of discovery, map, and topology filters that can be written. Nodes on the NNM map have to be manually entered and different attributes manually set. Available attributes and capabilities for nodes are set by field registration files and are vendor specific. Without the SNMP agent, only certain attributes can be set for a node within the NNM map. Capabilities are only set by NNM via the registration files once the SNMP agent type is determined. Furthermore if these addresses are moved or reassigned to other devices NNM has no idea that this has happened and cannot adjust the configuration automatically. It will up to the administration staff to manually alter the configuration related to the changed IP address.

With the DNS host name entries being in the convention interface-name.hostname.domain, every node within the NNM map is given the short hostname of the interface name, i.e. hme0, qfe1, lan0, lan1, etc. as the label of the object. This is due to NNM truncating to the right of the first dot from the beginning of the fully qualified host name to retrieve the short hostname. This too could easily be replicated into third party applications synchronizing with the NNM database (if the product tries to configure a short hostname). All the nodes within these databases will have a short hostname of the lan interface name and may be indistinguishable from each other.

Customer #2

The customer required installation and integration of several OpenView products and DNS configuration for the entire company. There were seven sites in which OpenView was installed. These seven sites reported their NNM nodes to the central management server. The OpenView products were Network Node Manager, CiscoWorks, NerveCenter, Compaq Insight Manager for Unix, NG SniffMaster, and Optivity. There was no systems management software installed.

The DNS was setup as described above. The IP addresses for multi-homed systems (routers) all resolved to the same hostname for that node. Aliases were inserted into the DNS with an address record for that alias. The aliases were setup with the naming convention -- host-interface.domain. This provided the ability to telnet to the specific interface.

This worked well with the synchronization of CiscoWorks (v4.0) and the NNM database. All sites were configured to use the CiscoWorks database at the central management server. All the names within the CiscoWorks configuration database were accurate. When setting up polling and data collection via CiscoWorks all reports were accurate.
Synchronization of NerveCenter and NNM was accurate as well. Node names within NNM, CiscoWorks, and NerveCenter were all correct and reflected the specific node.

Then came a re-organization of the company. The responsibility of the DNS moved to a different group within the company. This group decided to transition the DNS from the standard db-files to the Quadratech IP Management (QIP) product. This seemed like a good idea. It allowed for IP address ranges to be assigned and managed by individuals throughout the company. These individuals could in turn insert their own hostnames into DNS when needed via the QIP product. It made for less work for a single hostmaster.

Over time, strange things began to happen to the names within the NNM map and the whole OpenView suite of tools. The labels were no longer the short hostname, but host-interface. CiscoWorks now showed "new" routers with host-interface.domain as the hostname in its configuration database with no configuration history. NerveCenter showed some routers as nodes twice, once with the correct hostname and another with host-interface.

If an interface went down, automatic ticket generation from NerveCenter into Remedy would sometimes occur twice, once for the official hostname and once for the host-interface hostname. Even though they were the same node. This was due to them being in the same group and running through the same states.

What happened was the Wide Area Network group did not like the having each interface name inversely resolve to the same hostname so the host-interface name was moved to the official hostname of the interface within QIP. This allowed them to use the traceroute command via the command line and determine the interfaces the packet went through to get to the destination. Once the DNS was changed and NNM completed a configuration check on the router, a new nodename was assigned to the router. When CiscoWorks synchronized with the NNM database, it did not use the same interface as NNM to resolve the hostname. Now a new router was entered into the configuration database.

The NerveCenter synchronization gave the same results as CiscoWorks. A new host was found because the product did an inverse lookup of an interface and came up with a new node name and entered it into the Cisco group. Sometimes there were three different names used within the OpenView system that represented the same node. Operators were not happy. LAN personnel were not happy because it was NerveCenter that passed tickets on to the Remedy system. The Remedy system in turn paged the appropriate personnel after a ticket was created.

Over $1 million dollars was spent on hardware and software and the company stopped synchronization of the products so the WAN group could use traceroute.

Customer #3

For political reasons another name resolution scheme was implemented that was not useful to the rest of the networking groups. As a work-around the ITO administrators pulled down the name database, built a hosts file and set the resolver to look at files then DNS at the primary server.

This white paper recommends an nsswitch file that points to /etc/hosts and then to DNS, but this host file contained nearly 1300 names. The configuration works fine and ITO processing performance is not affected on the six CPU K580. However, over a period of time template distributions started to fail and messages for the ITO operators started to take a long time to show up in the browser.

Here the problem relates to hosts and DNS diverging from a shared name space. A name or IP resolution is made in DNS and the inverse is found in the /etc/hosts file. This can be further exacerbated by the remote device using another version of the name space to resolve its own name, which is reported to OV via SNMP.

Play out following scenario:

- A new address is added to a device or a device is moved to another subnet and DNS is updated but not /etc/hosts.
- The device is discovered or added by NNM/ITO and it tries to resolve the name.
- Processes look in /etc/hosts and do not find the address so DNS is queried and there it finds a name. A name that was used before, a name that is in the /etc/hosts file.
- Later in time that name is resolved to an address and this time it finds it in hosts, with the old IP. It tries to query the device and gets no response. If that IP has since been assigned to another device it gets back the name (via SNMP) of the different device. At this point the DB has inconsistent data in it and it really has no idea of the status of the new device.
- Next imagine that the remote device also has a name in its SNMP configuration that does not reflect its name in DNS or the name that NNM/ITO knows it by (because it has a slightly different name space). Then the SNMP query response could come back with even a different name that in turn could resolve to a completely different IP.
- An added twist came when another managed node running NNM was inadvertently allowed to discover a network where it could resolve names but the ITO server could not. This NNM system would detect node up and down and then send via SNMP a trap to the ITO server. Opcmsgm would receive this trap and try to resolve the address three times and each time, because of the resolver configuration, it would take 80 seconds to time out (see table 6.1 of DNS and BIND by Paul Albitz and Cricket Liu) During this four minute interval opcmsgm waits and message processing to the browser or trouble ticket plug-in does not proceed.

Summary

In summary name resolution must be accurate, fast, and consistent across all systems and lookup methods on the network or the different processes and databases that rely on this crucial data will not perform properly. We recommend the following design for hostname resolution as standard practice.

- Set up the hosts file in the following manner:
  IP Address   Fully qualified hostname   short hostname   alias
- Limit the hosts file to as few hosts as possible, mostly local interfaces.
- Configure the nsswitch configuration file to look for hosts with the hosts file then the DNS.
- Ensure all IP addresses for a node resolve to the same hostname within the hosts file and DNS.
- Ensure hostname lookups via the DNS return all the addresses assigned to the host.
- Use additional address records within the DNS for hostname aliases.
- Run a DNS server on the management system. At minimum, a caching only name server. A primary or secondary name server for the managed domains is preferred.