



HIGH-AVAILABILITY

NetMon

Reliable Network Connectivity

NetMon – Non-Clustered

Technology Introduction

Every device that connects to the Internet needs some method of connection, either to a corporate style network, which in turn connects to a point-of-presence (POP) or direct to a POP via a telecommunications provider.

In a server environment, the most common form of connection is to an Ethernet Local Area Network (LAN). Each server is fitted with an Ethernet port, which connects into a hub, switch or other similar device. All traffic to and from the server is typically via one or more of these connections.

Ethernet is available in a number of data rates, including 10Mega bits/sec (Mbs), 100Mbs and 1Gbs. Servers may be fitted with an 'internal' and 'external' interface, to allow traffic separation for security against 'attack' or for administrative reasons.

Why Do You Need NetMon?

Do you:

- Depend on network connectivity?
- Have servers with only one Internet facing interface?
- Want to remove single points of failure?
- Have limited IP address space?

Single Attached Servers

Servers normally have only one connection to the outside world (Internet) and in the event that this single connection fails they are no longer visible. Given suitably high-speed connections to the Internet then, a single network link is normally adequate for even high-powered hosts. Most servers are unable to continuously flood a 1Gbs network connection with data. The figure below shows a typical single server configuration, which is common in many small organisations.

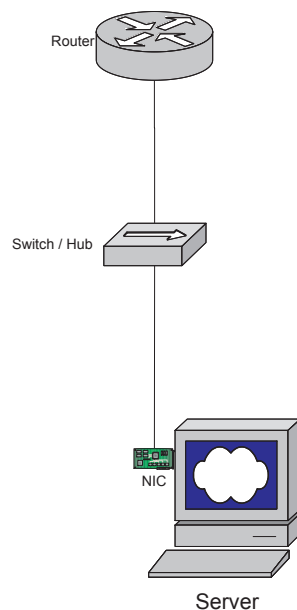


Figure 1.
Single Server with Single
Network Interface Card

This design while simple and cost effective has a number of problems.

- Multiple Single points of failure
- Limited performance
- Maintenance requires service interruption

For these reasons it is very unusual to use a single server in this way for providing a commercial web site.

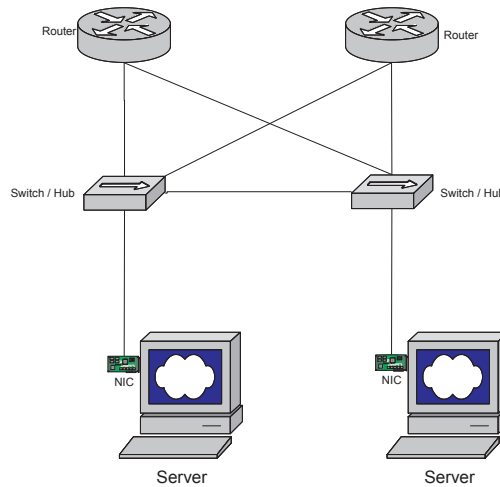
Multiple Servers

The most obvious single point of failure is the server. The network also has single points of failure but we will solve this problem at the same time as implementing another server as shown in the following diagram.

This implementation has no single points of failure. Indeed if two components (of different type) fail then the system should continue to work.



Figure 2.
Multiple Server
Configuration

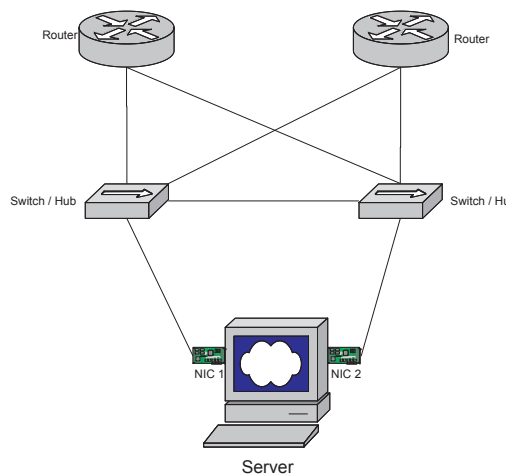


This design is also scalable, with many servers being deployed connected to one of the hubs. For sites with very high volumes of traffic it may be necessary to have a hundred or more servers to deal with the load generated. Using the design above 50 servers might be connected to each switch; load-balancing devices would be installed at the router or switch level to spread the load across all of the available devices. An added benefit of these load-balancing devices is that they usually detect server failure and 'steer' traffic to other servers if available.

This high capacity and fault resilient design has a further failing. In the event that a switch fails then half of the servers will remain connected. A solution to this would be to double a maximum number of servers required for peak volumes. This is clearly a high cost investment but does have the likely benefit of providing very good performance most of the time and adequate performance in the event of a failure.

A less costly solution would be to fit each server with more than one interface (Network Interface Card – NIC) as shown below.

Figure 3.
Multi-homed Server



This solution allows each server to theoretically use either network interface providing a solution with, not only no single points of failure but also, optimum use of resources.

Multi-Homed Server Issues

The physical design of the above solution is essentially flawless but the server is typically unable to satisfactorily able to determine which link is actually working. This is because the IP 'stack' and underlying technologies typically implemented in servers are unable to detect a cable being disconnected from the primary route. With a server being unable to detect a failure it is unable to determine the correct time to 'switch' to the 'good' interface.

To enable a server to detect a failure of an interface it is necessary to constantly monitor each interface. Whilst the server is able to constantly monitor the number of packets received on each interface, and hence infer that the server is still connected to the network, the server is typically unable to establish that other hosts are able to receive data from the server. However, if another server is sent a request and that then responds the original server can genuinely infer that the server is properly connected.

The server's testing procedures need not only to cater for the possibility that a partial failure is detected but that a false detection of failure does not occur. Multiple other devices should be contacted in an attempt to reduce the likelihood of a 'false' detection occurring.

Example Of A Possible Configuration

The server below is configured with normal public addresses for each interface or as shown with private addresses and an additional 'service' address that is public. The routers (which also have an integral switch/hub, or separate ones, which are not shown here) are configured with private addresses as well if required or simply the normal public addresses. This option allows you to make best use of the limited number of public addresses that are available whilst being able to properly test all connectivity.

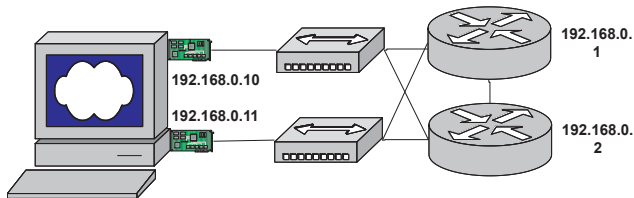


Figure 4.
Example Details

NetMon is installed on the server and configured to test for the presence of both routers using both of the server's interfaces. In the event that one router fails, reboots or is otherwise unavailable then NetMon will normally take no immediate action, as NetMon is still able to contact the other router and hence infer that both interfaces are working correctly.

Remove the cable to one of the interfaces and NetMon will detect the failure of the interface and if the interface is currently the primary (traffic bearing) then it will re-configure the machine so that traffic is now directed through the other interface. This move is achieved by moving the advertised IP address (not 192.168.0.10 or 192.168.0.11) to the other interface and issuing a 'promiscuous' ARP; (I'm over here now), which the routers will see and redirect traffic.

What Is NetMon?

NetMon is a program that runs on a server and tests for connectivity to the server's interfaces. It must be configured for the local network and must have some external devices to be able to contact, to enable the testing to take place.

NetMon has no kernel module components and runs completely in user space but for most configurations must be run with root level privileges to enable the program to attach to the correct internal components when running.

NetMon can be configured to operate at different speeds, with detection and switch over of traffic occurring in less than five seconds. In some installations it is safe to configure NetMon to respond in sub-second times but this should be used with caution as it increases the likelihood of a false failure detection on a machine with diminishing resources (CPU etc).

How Does NetMon Work?

Introduction

In principle NetMon needs to send a request and receive a response and both request and response should traverse the same defined path to enable NetMon to establish if the 'path' is functioning correctly. A number of different possible techniques can be used to test connectivity, some routable and some non-routable.

In the event of a failure of the interface, or other component, then the NetMon program should, if possible, switch the traffic to another interface. NetMon will continue to test all interfaces but will only switch traffic in the event that the active interface fails.

All methods allow multiple remote servers to be tested so that NetMon does actually reduce the number of single points of failure.

Routed

Test methods that can be routed are an integral part of IP or those that are built on top of IP. IP is routable; packets transmitted on Ethernet segments can be forwarded to other segments by routers.

Advantage	Disadvantage
No requirement for locally attached device that supports service required for testing.	Reduced certainty of path taken.
	May require a special Firewall filter.
Visibility beyond local network can test if machine can 'see' user space.	Reduced control and hence certainty about health of remote machines / network.



Non-Routed

Test methods that are non-routable as described in relation to NetMon are those which do not send IP packets. ARP traffic, whilst relating to IP, is not IP but Ethernet frames.

Advantage	Disadvantage
Controlled environment where all tests are local.	Machines on the local network must support the test service.
Is invisible to most general snoop mechanisms.	Harder to debug install problems, may require additional local node configuration.

The following sections explain the operation and some of the pros and cons of each mechanism.

Echo

Echo is a very simple standard service provided by many systems, which is typically used for testing connectivity. When data is sent to the service all data sent is sent back verbatim. TCP or UDP can be used by NetMon making a 'connection' to port 7.

As echo is built on top of UDP or TCP and IP it can be used in a routed environment.

Time

Time is a simple standard service provided by many systems, which is used for setting the systems clock of one system by getting the current time from another. The client sends a request and the reply contains the number of seconds since the epoch, January 1st 1970. As time is built on top of UDP or TCP and IP it can be used in a routed environment.

NTP

Network Time Protocol (NTP) is a fairly complex protocol in detailed implementation implemented by many hosts, which is used for more accurate time synchronisation between machines. NTP also has a number of security features that can be used if required. As NTP is built on top of TCP and IP it can be used in a routed environment.

Ping

ICMP echo request and reply, which is commonly known as ping, is an extremely simple service which is a mandatory requirement for every implementation of IP. Used solely for testing purposes (except by malicious attackers) it operates in a similar fashion to the echo service, returning verbatim all data it is sent. Ping is an integral component of IP and as such is routable, although some Firewalls are configured to block it.

ARP

Address Resolution Protocol (ARP) enables the IP protocol to discover the hardware Media Access Carrier (MAC) address of a server (or route to) that a server wishes to communicate with. All servers implementing IP with Ethernet (and other network types) must implement ARP. An ARP request is broadcast to all nodes on the LAN, requesting that any device that is or can reach the destination respond.

NetMon Configuration

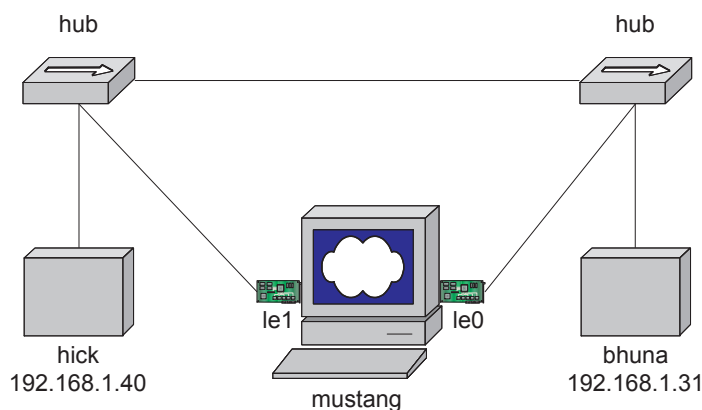


Figure 5.
Exmaple Configuration

Remote Connectivity (ISP Failover)

Routing Table

Default Route - 192.168.1.1
Static Route for 168.40.0.0 Gateway is 192.168.1.1
Static Route for 172.38.0.0 Gateway is 192.168.1.2

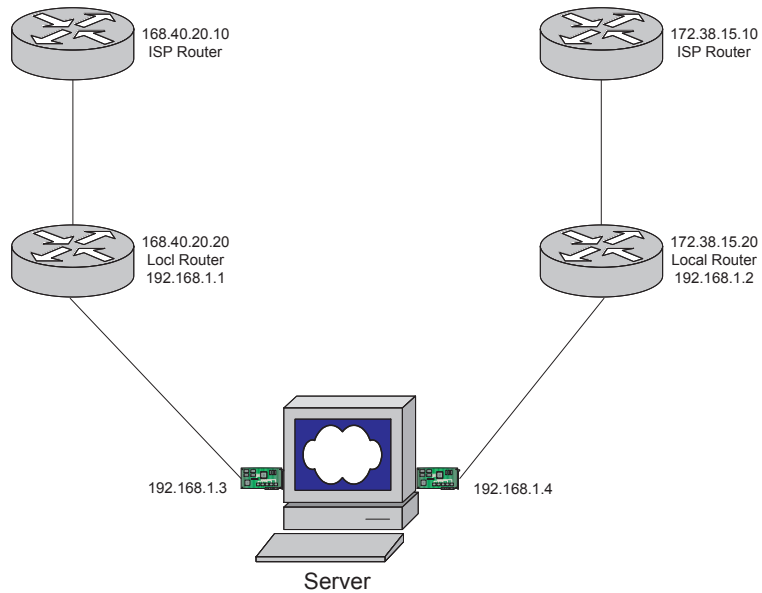


Figure 6.
ISP Switching

NetMon tests the ISP's router directly connected to our default router.

If the default router fails due to the ISP's router failing, ResMon will re-configure the default route to be 192.168.1.2.