3. MONITORING AND TESTING THE ETHERNET NETWORK

3.1 Introduction

The following parameters are covered by the Ethernet performance metrics:

- **Latency** (delay) – the amount of time required for a frame to travel from source to destination.
- **Jitter** – a measure of the deviation of the latency from its average value.
- **Loss rate** – the probability that an individual packet is lost (dropped) during the transmission.
- **Throughput** – the amount of digital data transferred per time unit.
- **Error rate** – the ratio of the number of erroneous units of data to the total number of units of data transmitted.
- **Bit error rate** – the ratio of the number of incorrectly received bits to the total number of transmitted bits.

There are many free and commercial applications available for common Operating Systems such as MS Windows, Linux and OS X that can provide varying LAN throughput performance measurement. There are also self contained hardware based solutions that provide the most accurate LAN performance measurement / stress testing.

Software applications can cost up to several hundred euros. Hardware based solutions range from a few hundred to tens of thousands of euros and are typically used by manufacturers, carriers and professional IT consulting organizations.

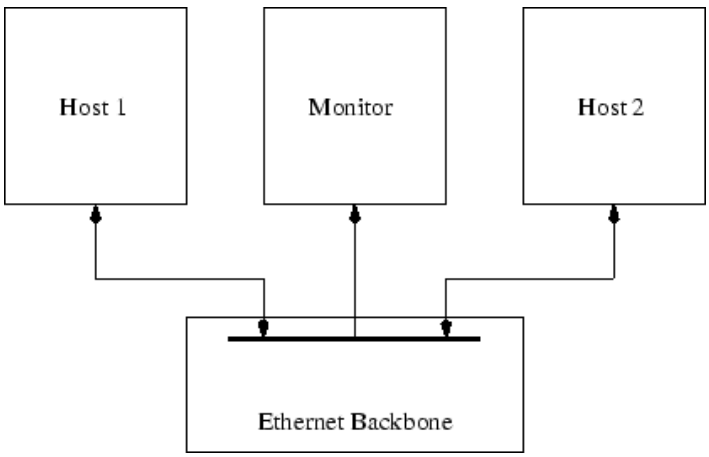

Fig 3.1. Basic Ethernet measurement setup

In its most basic configuration Ethernet is relatively simple to measure. The base configuration is shown in Figure 3.1. As shown here, Ethernet is a broadcast technology, where all but the sending machine can see all of the traffic on the network. The Ethernet

backbone could either be a coaxial cable string, as in 10base2 and 10base5, or a passive hub using a 10 or 100baseT. In all of these situations every machine sees all of the traffic at the same time, excluding cable delays. This means that all the monitor needs to do is capture every packet from the network and timestamp it.

Unfortunately fewer and fewer networks are using such a simple broadcast system any more. In order to obtain higher levels of performance and security, most networks use some sort of active device for the backbone. A common example of this is a network switch, where packets are sent only to the destination machine. This has the advantage of being much more secure; only the destination machine can look at the traffic, and can have higher performance. This is because if two pairs of machines are using the network at once, in a broadcast system they all have to share the available bandwidth equally. In a switch, however, each pair has the full available bandwidth between them. The possible bandwidth between hosts only drops off when multiple hosts all want to transmit to the same destination.

Both of these properties make passive measurement difficult. If the network traffic is no longer broadcast, how can the monitor watch the traffic? There are two methods that can allow this. First, it is often the case that an Ethernet switch will contain one port that copies all the data. The problem with this port is that the switch must interfere with the data. Normally two packets could be trying to flow through the switch at exactly the same time, providing the source and destinations are different. If this occurs then the switch must delay and queue one of the packets before it is sent to the port, introducing inaccuracies in the timestamps on these packets. One other problem is that the data rates inside a switch can exceed the data rates of a single line. In this case the switch must `clip' the data that is sent to the accumulated port, dropping packets from the trace.

The second alternative is to insert a passive electronic tap into the wire between a host and switch. This situation ends up looking very similar to the ATM situation due to the full duplex host to hub connection present on modern 100baseT systems, requiring 2 receiver interfaces on the monitor.

3.2 Hardware based measurement

Hardware solutions try to correct the many limitations of a software system. In a full hardware solution, as much of the monitoring system is carried out on the custom interface (card) as possible, and the system is used for storage and formatting of the data.

The important features that can be carried out on the card are:

- -Timestamping
- -Clock synchronization
- -Dropped packet counting
- -Traffic filtering

When recording network traffic, accurate timestamping of the arrival of packets is essential for the subsequent analysis of performance metrics. Ideally the packet should be timestamped as soon as it has arrived on the card, before any buffering or queuing takes place.

Clock synchronization is important if there are multiple interfaces in a machine. If packets are being captured by two interfaces, the timestamps need to be consistent between interfaces.

In a hardware system, a check can be placed on any buffers to detect packet loss. A hardware solution will not necessarily be perfect, but a well designed system will be able to provide a definite count of any drops that have occurred in the system. For this reliability to carry through to the final network trace, checks must be placed at every point in the system that packet loss can occur. This includes any software components that run on the host PC to deal with the storage of the data as it is delivered by the monitor card.

Hardware solutions also allow a monitor to deal with larger data rates than software solutions could. It is quite often the case that a limited amount of data will be saved in a monitor, however in a software solution the operating system and possibly monitoring application have to see all of the traffic, then decide if it should be kept. If the hardware becomes intelligent, then it can carry out this function, discarding unwanted data at the earliest possible point.

The major disadvantage of hardware monitoring systems has always been cost. Until recently there have been limited amounts of hardware designed specifically for network monitoring. This meant that any group wanting high accuracy network monitoring had to design such a system from scratch. These solutions are often too costly for some measurement work, so they tend to be used for projects requiring very high accuracy or high speed systems.

Fig. 3.2. Validator NT 955

Test-Um NT 955 Validator NT (Fig. 3.2) is an example of all-in-one network management tool with a 4-inch color LCD screen. The NT955 Validator NT certifies, identifies, configures and documents the Ethernet network.

The Test-Um NT955 Validator NT can:

- Determine fault locations, cable length and delay or noise conditions.
- Produce and print cable test schedules and cable test results.
- Qualify lines for VoIP usage.
- Identify - active components of your network on the other end of the cable
- Identify all types of equipment and port service discovery with advertised speed ratings and DHCP negotiation.
- Access IP addresses, ping equipment and flash hubs/switches for positive port location. · Configure - links between nodes at Gigabit speed.
- Check IP addresses on netmask, Gateway/routers and domain name servers.
- Confirm links between equipment for changes or upgrades.

3.3 Software based measurement

All measurement systems will require some software components, the important distinction is that in a software system, no special assistance for measurement is provided by any hardware.

The simplest and quickest passive measurement system is to run the Unix program *tcpdump*. This program is supported under most Unix systems. It will listen on a specified network interface, and capture all traffic seen on that link. Regardless of its name, it will dump all data sourced from and destined for that interface, as well as any other traffic seen on that section of the network, and is not restricted to TCP. The *tcpdump* can run in two modes, either capture traffic to a file, or display text information on every packet on the screen. Also built in are filtering rules that allow capture of a specified subset of data, for instance TCP traffic, on port 80, between host x and y.

The *tcpdump* is a not only a very common application, but provides a good example of techniques used in software-based solutions. A software based monitor requires assistance from the operating system kernel. In most operating systems the kernel contains the network stack that provides an interface between the applications and the network card. In normal operation there are two levels of filtering occurring. First the network card will only transmit to the network stack packets that have to be dealt with by this machine. For point to point network systems, this will be all packets, for broadcast systems such as Ethernet, this will be only a subset of packets. Next the kernel splits up packets, deals with some itself, and sends on the others to the appropriate application to deal with.

A software monitor such as *tcpdump* will run as an application program. Without a special interface to the network stack, this program would be unable to capture any packets that were destined for other machines or other applications on the monitor. This would mean that passive measurement would be impossible.

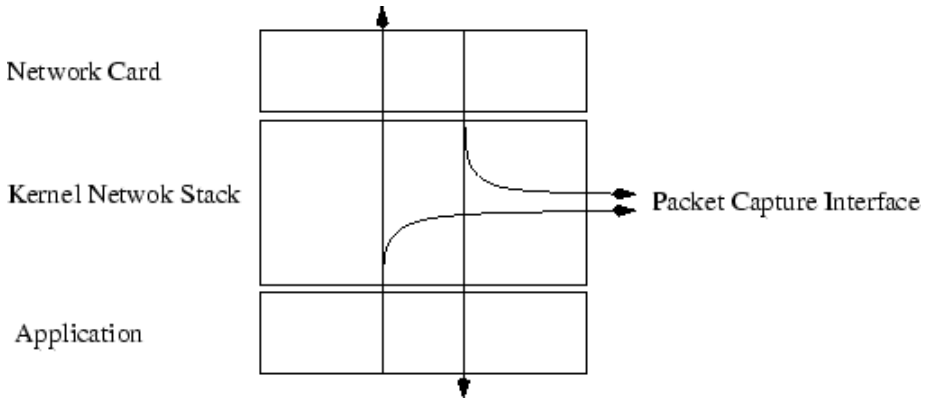

Figure 3.3. Software monitor setup

Figure 3.3 shows a representation of the network layering present in many Unix kernels. The interface that allows programs such as *tcpdump* to work, is the packet capturing interface. This interface, referred to often as the *pcap* interface, is accessed through a C library called *libpcap*. This library also allows reading and writing to files, providing uniform access to a program for trace files and live networks. *tcpdump* is a higher level

interpretation and filtering program that sits on top of libpcap. For this reason it is common to refer to files created by *libpcap* as *tcpdump* files.

libpcap provides the method for a software program to capture packets that are destined for other applications. To allow capture of all packets seen on a network, it is often necessary to put a network card in promiscuous mode. In this mode a network card will provide the network stack with all packets on the network, not only the ones to be dealt with locally. These packets will normally be discarded at the kernel level, but will be forwarded to the pcap interface if requested.

The problem with software monitoring solutions is the fact that none of the components have been designed or optimized for this use. There are many delays, and buffers in a software solution that reduce the accuracy of timestamps and increase the possibility of packet loss.

The first place this occurs is at the hardware level. A network card may not be reliable enough to capture every packet on a network. In normal use these dropped packets would be corrected for by the TCP stack for a reliable connection. In a monitoring situation, undetected packet loss could cause problems with analysis, or just reduce the accuracy of results. Another common problem is caused by attempting to increase performance; many network cards will buffer a stream of packets, the card will then only interrupt the host once and deliver a collection of packets, as opposed to doing this for each packet. This increases host performance, but means that the timestamps on successive packets may not represent the actual inter-packet time.

The timestamping process on a *libpcap* system occurs in the kernel. This timestamp can be delayed if the host system is under load, and the interrupt service for a packet is delayed. Also the buffers that store the packet before delivery to the libpcap application may be unchecked. If an application program does not clear the buffer quickly enough, then packets can be silently dropped.

Software based systems are not limited to using the *pcap* interface. A user could write a custom network card driver to provide an interface to the network card for use with passive measurement only. This solution would remove the ability to use the network card as a standard network interface, but it could still suffer from many of the problems of a *libpcap* system.

Software solutions have one major advantage. They are likely to be much cheaper and quicker to setup than a specialist hardware solution. They offer only limited accuracy of the data measured.

3.4 Further reading

For more information visit:

http://www.ixiacom.com

http://www.testersandtools.com

http://www.wand.net.nz

http://www.ethernetextender.com

Sources:

Figure [3.1], [3.2] http://www.wand.net.nz

Figure [3.3] http://www.elnex.pl

Authors: P.Bogdanski, W. Grega