RESEARCH ARTICLE OPEN ACCESS

# QUALITY OF SERVICE IN NETWORK ADMINISTRATIONS

T.Sankar[1]
Assistant Professor in Computer Science, Department of Computer Science,
Vinayaka Missions University, Salem.

## Abstract:

Today's computer networks have gone from typically being a small local area network, to wide area networks, where users and servers are interconnected with each other from all over the world. This development has gradually expanded as bandwidth has become higher and cheaper.But when dealing with the network traffic, bandwidth is only one of the important properties. Delay, jitter and reliability are also important properties for the quality of network connection. This is because different applications has different needs, and therefore require different properties from the network. System administrators are in an increasing degree involved with the troubleshooting of solving network problems concerning the quality of service for the different applications.This research paper analyzed techniques for measuring, analyzing, presenting and interpreting the different properties for the administration of remote computer network.

*Keywords*— **Bandwidth, Delay, Jitter, Reliability.**

## INTRODUCTION:

The most used network architecture is the client-server architecture. In a client-server architecture the server passively waits for a request, until the client actively sends a request to the server. The server then executes the request and sends the reply back to the client.

One of the first computer networks were isolated local area networks (LANs), with client-server architecture. The clients were cheap terminals, attached to a screen and a keyboard. At the time, the clients required low network bandwidth. The only data transmitted was the keyboard activity sent to the server, and the screen updates sent back to the client.

The terminals used in these networks are classified as thin clients. This is because most of the processing is done at the server, while the client typically process keyboard input and screen output.

## THEORETICAL BACKGROUND:

Quality of Service

The stream of packet between two nodes in a network is called a flow. This flow will in a connection-oriented network follow the same route, but in a connectionless network, the packet may take different routes [1][14][15].

| Application | Reliabil | Delay | Jitter | Bandwi |
|---|---|---|---|---|
| E-mail | High | Low | Low | Low |
| File transfer | High | Low | Low | Medium |
| Web access | High | Mediu | Low | Medium |
| Remote login | High | Mediu | Mediu | Low |
| Audio on | Low | Low | High | Medium |
| Video on | Low | Low | High | High |
| Telephony | Low | High | High | Low |
| Videoconferen | Low | High | High | High |

The problem with a connectionless network is that the routes may have different properties. The four main properties for a network connections are [1][15]:

1. Bandwidth
2. Delay
3. Jitter

4. Reliability

These four properties define the quality of service (QoS), that the flow requires. The QoS for the routes may not matter for some applications, but it may be crucial for others. Table 1.1 shows the stringency of several common applications [1].

Table 1.1: Quality-of-service requirement's stringent

**Providing Quality of Service:**
**1. Bandwidth**

**Theory**

Bandwidth is a term used to describe the capacity of a link. It is the transmission rate for the link. A link able to transmit at 100 Mbps, has a bandwidth of 100 Mbps[16]. Table 1.2 lists some of the typical bandwidths provided by the most common medium access control technologies. Table 1.3 lists categories of wide area network connections provided by internet service providers [1].

Even though bandwidth is what is provided by the internet service provider, it is the throughput of the connection that is of interest for the customer. "Throughput is a measure of the amount of data that can be sent over a link in a given amount of time [16]".

The throughput is determined by the formula:

$$Throughput = \frac{Data\ Transferred}{Time} \qquad (3.1)$$

The throughput is expressed in bits per second or packets per second. But when expressed in bits per second, the more the typical expression is kilobits ($10^3$ bits), megabits ($10^6$ bits) or gigabits ($10^9$ bits) per second, depending on the connection throughput.

The difference between throughput and bandwidth is that throughput measurements may be affected by considerable overhead that is not included in bandwidth measurements. And therefore throughput is a more realistic estimator of the actual performance for the connection [16].

Table 1.2: Bandwidths provided in Local Area Networks

| Description | Bits | Bytes |
|---|---|---|
| Ethernet (10base-X) | 10 Mb/s | 1,25 MB/s |
| Fast Ethernet (100base-X) | 100 Mb/s | 12,5 MB/s |
| FDDI | 100 Mb/s | 12,5 MB/s |
| Gigabit Ethernet (1000base- | 1.000 Mb/s | 125 MB/s |

Table 1.3: Bandwidth technologies in Wide Area Networks

| SONET | | SDH | Data Rate (Mbps) | | |
|---|---|---|---|---|---|
| Electric | Optic | Optic | Gross | SPE | User |
| STS-1 | OC-1 | STM- | 51,84 | 50,112 | 49,536 |
| STS-3 | OC-3 | 1 | 155,52 | 150,336 | 148,608 |
| STS-9 | OC-9 | STM- | 466,56 | 451,008 | 445,824 |
| STS-12 | OC- | 3 | 622,08 | 601,34 | 594,432 |
| STS-18 | 12 | STM- | 933,12 | 4 | 891,648 |
| STS-24 | OC- | 4 | 1244,1 | 902,01 | 1188,86 |
| STS-36 | 18 | STM- | 6 | 6 | 4 |
| STS-48 | OC- | 6 | 1866,2 | 1202,68 | 1783,29 |
| STS- | 24 | STM- | 4 | 8 | 6 |

**Data Collection**

The two factors affecting the throughput are, the amount of data transferred, and the time it took to transfer that data.

Determining the throughput can be done in two ways:

1. Measuring the time it takes to transfer a predetermined amount of data.
2. Measuring the amount of data transferred in a predetermined amount of time.

**Analysis**

The throughput measurements provide useful statistical information about the throughput of the node. If the node has used active measurements, the measurements show the throughput for the connection, while passive measurement tools show the utilization of the bandwidth.

Trends are identified by presenting the measured data in a time series diagram, where the time is for a long duration of time. The longer the duration, the easier the trend may be to recognize.

The throughput distribution can be viewed in a histogram diagram.

**2. Delay**

**Theory**

The delay is the time it takes to send a packet or frame from a source node to a destination node. The delay is the product of thee delays, these are called [2][16][22]:

- Transmission delay
- Propagation delay
- Queuing delay

Transmission delay is the amount of time it takes to put the signal onto the cable. This depends on the transmission rate (or interface speed) and the size of the frame 16].

Propagation delay is the amount of time it takes for the signal to travel across the cable. This depends on the type of media used and the distance involved [16].

Queuing delay is the time it takes to process the packet or frame at intermediate devices such as routers and switches. It is called the queuing delay because most of time is spent in queues within the device [16].

The transmission delay and the propagation delay are quite predictable and stable delays, but queuing delays can introduce considerable variability [2][16]. The delay can be calculated by the following method:

$$d_1 = t_1 + p_1 + q_1$$
$$d_2 = t_2 + p_2 + q_2$$
$$\dots$$
$$d_N = t_N + p_N + q_N$$

Where $d_{1\text{-}N}$ is the delay for each node on the path, $t_{1\text{-}N}$, is the transmission delay between each of the nodes on the path, $p_{1\text{-}N}$ is the propagation delay between each of the nodes on the path, and $q_{1\text{-}N}$ is the queuing delay for each node on the path.

The total delay is then:

$$Delay = \sum_{i=1}^{N} d_i$$

Where $d_{1\text{-}N}$ is the delay for each node on the path and N is the number of nodes on the path to the destination.

The delay is expressed in time, and since the delay usually is quite small, it is expressed in ms.

**Data Collection**

To measure the delay, it is necessary to send a packet to a destination node, and somehow measure the delay between the nodes. The ICMP are designed to handle these kinds of operations, but this may create an unreliable result as the ICMP packets are usually prioritized compared to the IP packets.

There are two methods for measuring the delay between two nodes:

- One way delay
- Round trip delay

The one way delay is the time it takes the packet or frame to travel from the source to the destination node. The problem with measuring this value is the time synchronization needed to get a reliable result. If the clock is not perfectly synchronized for both hosts, the timestamps provide a false result. Time synchronization is provided through the network time protocol (NTP) and GPS. Tools like sting [23] provides one way delay measurements.

**Analysis**

The delay measurements can provide information as statistics of the connection that has been monitored. This can provide useful, when troubleshooting applications. By presenting the measurements in diagrams, the following information can be identified:

- Trends for the round trip time, for the measured connection. The trends can be viewed in a time series diagram.

- The distribution of the round trip time. This can be viewed in a histogram diagram.
- The congestion of the connection. Congestion is determined by a phase plot diagram.

In a phase-plot of a given measurement period there are three congestion regions as shown in figure 2.1[22]:

- Region I contain probe pairs that see empty queues and experience minimum RTT plus minor random overheads.
- Region III contains probe pairs that always see a queue. This is the region of persistent congestion.
- Region II contains probe pairs where one of the probes experiences queuing delay but the other does not, i.e., there is a transition in congestion state between adjacent probes. This is the region of transient congestion.

## 3. Jitter

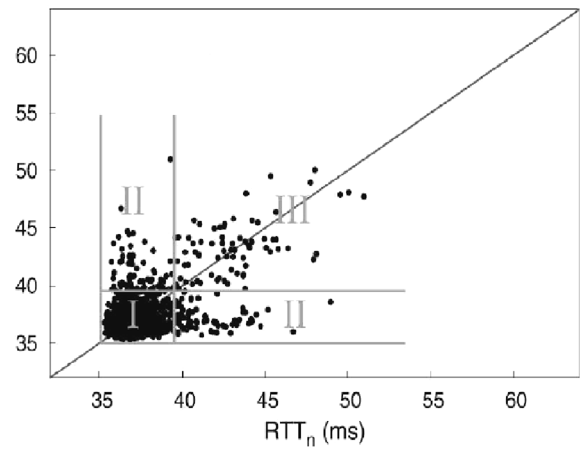
### Theory

Jitter is the variation in arrival times of successive packet from a source to a destination. And is determined by the difference experienced by subsequent packets, $RTT_I$ and $RTT_{I+1}$[2][16][22]. The mathematical formula is:

$$Jitter_I = \sqrt{\frac{1}{2}(RTT_I - RTT_{I+1})} \qquad (3.3)$$

### Data Collection

The jitter can be measured by monitoring the round trip time for packets between two nodes. This can be done by passive measurement tools like tcpdump that taps into the network and stores the relevant data. Other tools can then extract information from tcpdump which can then be analyzed. Active measurement tools include all of those used to measure the round trip time.




### Analysis

Showing the jitter in a time series diagram, shows the jitter during that time period, but reveals little information about the jitter itself. More interesting information about the jitter comes from the distribution of the jitter. This can be viewed in a histogram chart. When the distribution of the jitter is mainly within a few seconds, the jitter can be qualified as low, but this depends on the requirement of the application. The desired distribution would be an exponential distribution. If the distribution is not within a few seconds, but rather spread across several second, the connection is unpredictable and has a high jitter value.

## 4. Reliability

### Theory

Reliability is defined as "An attribute of any system that consistently produces the same results, preferably meeting or exceeding its specifications"[28].

A method to describe reliability is to use the failure rate, which describes how frequently something fails. A failure in network is when the packet does not reach its destination, before the time expire [29][30].

The failure rate (λ) has been defined as "The total number of failures within an item population, divided by the total time expended by that population, during a particular measurement interval under stated conditions. (MacDiarmid, et al.)". It has also been defined mathematically as the probability that a failure per unit time occurs in a specified interval, which is often written in terms of the reliability function, R(t), as,

$$\lambda = \frac{R(t_1) - R(t_2)}{(t_2 - t_1)R(t_1)} \quad (3.4)$$

where, t1 and $t_2$ are the beginning and ending of a specified interval of time, and R(t) is the reliability function, i.e. probability of no failure before time t.

The failure rate data can be obtained in several ways. The most common methods are [30]:

- Historical data about the device or system under consideration.
- Government and commercial failure rate data.
- Testing.

When monitoring a network connection or a node, packet loss is a measurement for measuring the fraction of packets sent from a measurement node to a destination node for which the measurement node does not receive an acknowledgment from the destination node [30].

**Data Collection**

The active measurement tools used for measuring the delay and jitter, are also suitable for measuring the failure rate.

**Analysis**

The interesting information gathered from the raw data conserving the reliability, is how many error there where during the measurement period. This is known as the error rate. It shows the number of errors divided on the time interval.

To retrieve reliable failure rate data, the testing should be performed over a relative large period of time. This removes uncertainty in the result.

## PROPOSED METHOD:

The state of a network link can be determined by measuring the throughput, the delay, the jitter and the packet loss. These four properties represent the quality of the link. In the following three case studies, methods for measuring these and more properties will be shown.

**Case One: Network Traffic**

**Motivation**

By monitoring the network traffic for a node, information about the state of that node can be determined. This may provide the network administrator, with enough information to optimize the system performance, by removing bottlenecks. The system can represent the node, a subnet, or the whole network.

There are especially two locations that are of interest, when performing passive network measurements:

- The state of a service host, that provides a network service. Examples of network services are the DNS, DHCP, HTTP, and FTP services.
- The state of a network node, performing a routing or forwarding functions. Examples of such nodes are firewalls, virtual private networks (VPN), and routers.

**Objective:**

By using a passive network measurement tool, two nodes are to be monitored for one day. The data gathered from these nodes are to be analyzed, and the state of the nodes is of interest.

Resources:

In this experiment, the resources located in table 1.4 have been utilized.

Table 1.4: The resources utilized in Case One.

| Description | Node One | Node Two |
|---|---|---|
| Processor | Intel | AMD Athlon(tm) |
| Processor Mhz | 549,947Mhz | 1852,314Mhz |
| Memory | 640 MB | 1024 MB |
| Network MAC | Fast Ethernet | Ethernet FD |
| Network Link | 100 Mb/s | 10 Mb/s |
| Internet Service | UNINETT | Bredbandsbolaget |
| IP Address | 128.39.73.19 | 83.227.111.133 |

**Description of Node One**

Node one is a host located in a test lab at Oslo University College, in Oslo Norway. The operating system is GNU Linux Debian, where the version of Debian is "Sarge". The host is also running the following services: PostgreSQL, SMTP, HTTP, HTTPS, FTP, and SSH.

The host is connected to a local area network, which shares a 155Mb/s internet connection with the rest of the school. But as the local area network, and the host network interface card only supports Fast Ethernet, the maximum network speed is about 100Mb/s.

**Description of Node Two**

Node Two is a host serving as a firewall for a local area network, located in Oslo Norway. The operating system is GNU Linux Debian, where the version of Debian is "Woody". The host serves the firewall, which protects a local area network with services like HTTP, FTP, and SSH.

The host is connected to an Ethernet connection, providing 10 Mb/s internet connections. The ISP has a direct connection from Norway to Bredbandsbolaget's backbone network, which is located in Sweden.

**Tools**

To perform the passive measurements, a program has to run on the node that is monitored. The SNMP service is an alternative method for collecting the measurements. But it may also generate traffic on the network, if the collecting node is located on the monitored network. Another approach would have been to log all network traffic, with the help of tcpdump or an equitant program, and then later process the saved data. But this generates a lot of data, requires a lot of disk space, and it lacks the function to measure the state of the node.

A more suited program for the measurements conducted in this experiment, where tcpstat. tcpstat is a highly configurable program that measures some data, and may generate some statistics if wanted. Examples of the data that can be gathered are: bits per second, bytes since last measurement, ARP packets since last measurement, TCP packets since last measurement, ICMP packets since last measurement, etc.

The following command was executed on both nodes, and ran on the nodes for one day.

```
tcpstat -i eth0 \
        -o "%S %A %C %V %I %T %U %a %d %b %p %n %N %l \n"
```

The explanation of logged data can be found in the result chapter.

**Predictions**

The predictions for the result are:

- Node one, will never fully utilize the available bandwidth, but will probably utilize 100% of the processing power.
- Node Two, will fully utilize the bandwidth, but will probably not utilize the processing power.
- For both nodes, IP will dominate the network layer protocols, and TCP will dominate the transport layer protocols.

**Case Two: Throughput**

**Motivation**

By performing active measurements from one node to another node with equal link speed, the state of the connection can be determined. If the link speed is not as expected, countermeasures can be taken to locate and remove the bottleneck.

**Objective**

By using a active measurement tool, the connection between two nodes are to be benchmarked and analyzed. The test should provide enough information to see trends in the network, and determine if the node manage to utilize the available bandwidth. To remove uncertainties in the results, benchmarking should be executed from one node, to two other nodes.

**Description of Node One**

Node one is a host located in a test lab at Oslo University College, in Oslo Norway. The operating system is GNU Linux Debian, where the version of Debian is "Sarge". The host is also running the following services: PostgreSQL, SMTP, HTTP, HTTPS, FTP, and SSH.

Table 1.5: The resources utilized in Case Two

| Description | Node One | Node Two | Node |
|---|---|---|---|
| Processo | Intel P III | Intel P | Intel P III |
| Processor Mhz | 549,947M | 167,047M | 447,699M |
| Memory | 640 MB | 96 MB | 923 MB |
| Network MAC | Fast | Fast | Fast |
| Network Link | 100Mb/s | 100Mb/s | 100Mb/s |
| Internet Service | UNINETT | UNINETT | UNINETT |
| IP Address | 128.39.73. | 158.38.88. | 128.39.74. |

The host is connected to a local area network, which shares a 155Mb/s internet connection with the rest of the school. But as the local area network, and the host network interface card only supports Fast Ethernet, the maximum network speed is about 100Mb/s.

Description of Node Two

Node Two is a host located in the student housings at Molde University College, in Norway. The operating system is GNU Linux Red Hat, where the version of Red Hat is "9.0".

The host is connected to a local area network, which shares a 155Mb/s internet connection with the rest of the school. But as the local area network, and the host network interface card only supports Fast Ethernet, the maximum network speed is about 100Mb/s.

The path from Node One to Node Two is shown in table 1.6.

Table 1.6: Path from Node One to Node Two.

| Path ID | IP | Location |
|---|---|---|
| 01 | 128.39.73.1 | Oslo, Norway |
| 02 | 158.36.84.21 | Oslo, Norway |
| 03 | 128.39.0.73 | Oslo, Norway |
| 04 | 128.39.46.249 | Oslo, Norway |
| 05 | 128.39.46.2 | Trondheim, Norway |
| 06 | 128.39.46.102 | Trondheim, Norway |
| 07 | 128.39.47.102 | Ålesund, Norway |
| 08 | 128.39.47.130 | Molde, Norway |
| 09 | 158.38.0.66 | Molde, Norway |
| 10 | 158.38.88.147 | Molde, Norway |

**Description of Node Three**

Node Three is a host located in the student network at Oslo University College, in Oslo Norway. The operating system is GNU Linux Debian, where the version of Debian is "Woody". The host is also running the following services: MYSQL, NTP, SMTP, HTTP, FTP, and SSH.

The host is connected to a local area network, which shares a 155Mb/s internet connection with the rest of the school. But as the local area network, and the host network interface card only supports Fast Ethernet, the maximum network speed is about 100Mb/s.

The path from Node One to node two is shown in table 1.7.

Table 1.7: Path from Node One to Node Three.

| Path ID | IP | Location |
|---|---|---|
| 01 02 | 128.39.73.1<br>128.39.74.16 | Oslo, Norway<br>Oslo, Norway |

**Tools**

There are multiple tools that perform about the same function when performing active measurements. Known network throughput benchmarking tools are: netperf, iperf, ttcp, and ftp.

The tool chosen to test the throughput is netperf. To execute the experiment, a server node and a client node has to be installed on each of the nodes.

For the experiment, the server program was installed on Node Two and three, and the client software was installed on Node One. This setup was chosen, so that the process of benchmarking could be controlled from Node One. This minimizes the probability for interference from each measurement.

**Predictions**

The predictions for the result are as follows:

- As all nodes are attached to an over dimensioned network, the connection itself should not be a problem. And it should be possible to achieve full link utilization.
- Node Two could have a problem to achieve 100 Mb/s as the processing power is a bit low.
- All the nodes are connected to a school network. This will probably mean that the

link has the highest load during the day. This is why there is a higher chance to achieve full link utilization during the night or weekends.

**Case Three: Delay, Jitter and Packet Loss**

**Motivation**

- By using an active measurement tool that measures the delay between two nodes, the jitter and packet loss can be determined by using mathematical methods.
- The delay can be measured as the time it takes for one packet to be sent from a host, until it is received at the destination. But as this requires that the clocks are perfectly synchronized, an alternative method is mostly used. This is to measure the delay in form of the round trip time.
- The round trip time is measured as the time it takes for one packet to be sent from a node to a destination node, until another packet is received from the destination node.

**Objective**

- The previous cases showed methods for measuring the throughput for the link. In this last case study, the delay of a network link is to be measured, and based on those measurements; the jitter and packet loss is to be determined.
- The round trip time, from one node to three other nodes are to be measured for one week. This should provide enough information to make reasonable decisions about the link state.

**Resources**

In this experiment, the resources located in table 1.8 have been utilized.

Table 1.8: The resources utilized in Case Three.

| Description | Node One |
|---|---|
| Processor Model | Intel Pentium III |
| Processor Mhz | 549,947Mhz |
| Memory | 640 MB |
| Network MAC | Fast Ethernet FD |
| Network Link | 100 Mb/s |
| Internet Service Provider | UNINETT |
| IP Address | 128.39.73.19 |

In addition, the link and processing power of three remote nodes has been utilized. As the active measurements does not require any installation or configuration of the destination nodes, the hardware configuration of Node Three and four are not known. The hardware configuration of Node Two can be found in table 1.5 (Node Two).

**Description of Node One**

Node one is a host located in a testlab at Oslo University College, in Oslo Norway. The operating system is GNU Linux Debian, where the version of Debian is "Sarge". The host is also running the following services: PostgreSQL, SMTP, HTTP, HTTPS, FTP, and SSH.

The host is connected to a local area network, which shares a 155Mb/s internet connection with the rest of the school. But as the local area network, and the host network interface card only supports fast ethernet, the maximum network speed is about 100Mb/s.

**Description of Node Two**

Node Two is a host located in the student housings at Molde University College, in Molde Norway. The operating system is GNU Linux Red Hat, where the version of Red Hat is "9.0".

The host is connected to a local area network, which shares a 155Mb/s internet connection with the rest of the school. But as the local area network, and the host network interface card only supports Fast Ethernet, the maximum network speed is about 100Mb/s.

**Description of Node Three**

Node Three is a node that is a part of a cluster that serves the "www.vg.no" domain. This domain belongs to a Norwegian news paper called "Verdens Gang". The path from Node One to Node Three can be view in table 1.9.

Table 1.9: Path from Node One to Node Three.

| Path ID | IP | Location |
|---|---|---|
| 01 02 | 128.39.73.1 | Oslo, Norway |
| 03 04 | 158.36.84.21 | Oslo, Norway |
| 05 06 | 128.39.0.73 | Oslo, Norway |
| 07 08 | 193.156.120.3 | Oslo, Norway |
| | 193.75.3.6 | Oslo, Norway |
| | 193.75.3.1 | Oslo, Norway |
| | 193.69.165.11 | Oslo, Norway |
| | 193.69.165.11 | Oslo, Norway |

**Description of Node Four**

Node Four is a node that is a part of a cluster that serves the "www.kernel.org" domain. This domain belongs to the official GNU Linux kernel. The path from Node One to node four can be view in table 1.10.

Table 1.10: Path from Node One to Node Four.

| Path ID | IP | Location |
|---|---|---|
| 01 | 128.39.73.1 | Oslo, Norway |
| 02 | 158.36.84.21 | Oslo, Norway |
| 03 | 128.39.0.73 | Oslo, Norway |
| 04 | 128.39.46.249 | Oslo, Norway |
| 05 | 193.10.68.101 | Oslo, Norway |
| 06 | 193.10.68.29 | Stockholm, Sweeden |
| 07 | 213.242.69.21 | Stockholm, Sweeden |
| 08 | 213.242.68.201 | Stockholm, Sweeden |
| 09 | 212.187.128.25 | London, England |
| 10 | 4.68.128.106 | New York, USA |
| 11 | 64.159.1.130 | San Jose, USA |
| 12 | 4.68.114.158 | San Jose, USA |
| 13 | 209.245.146.251 | San Jose, USA |
| 14 | 192.5.4.233 | San Jose, USA |
| 15 | 204.152.191.5 | San Jose, USA |

**Tools**

There are multiple tools for measuring the round trip time, or one-way delays. The tools vary in methods for collecting and measuring the data. The biggest difference is what sort of packet which is used. The available packet formats can be ICMP, TCP or UDP.

The tool used to measure the data, is a modified Perl script that utilized the ping command which is available for most operating systems. The Perl script is a part of the pinger measurement package, which is used to measure round trip times from links all around the world. The modified Perl scripts together with other scripts are available in the appendix.

**Predictions**

The predictions for the result are as follows:

- Node Three has the least amount of hops, and is located in Oslo, Norway, this node will probably have the lowest delay, the lowest jitter, and the lowest packet loss.
- Node Two has the second least amount of hops, and it is located in Norway, so this node will probably have a low delay, a predictable jitter, and a low packet loss.
- Node four is located in the United States of America. This will probably result in a high delay, with at times unpredictable jitter. The packet loss should however be relative low, with today's network link properties.

## CONCLUSION

The objective is to assist network and system-administrators in administration of remote computer networks.

This was primary done by identifying the properties for securing the remote computer networks, and the properties that are important for the quality of the services. The properties provide quality of service for the connection between the remote computer networks.

Secondary, some simple methods for analyzing and presenting the measured data was identified. These methods simplify the interpretation part of the administration of remote computer networks.

The three case studies were created to demonstrate the functionality for some of the tools used to measure the four properties in quality of service.

In Case One, the objective was to make use of passive throughput measurement tools, to monitor the traffic on two different nodes for one day. The tcpstat tool successfully measured the data on both nodes, and provided enough information to create a good understanding of what had happened on the network for the last 24 hours. The only mistake in these two experiments was that the filter functionality in tcpstat should been used to filter input and output traffic. But as the objective was demonstrate, the experiments can still be classified as successful.

In Case Two, the objective was to make use of active throughput measurement tools, to benchmark the network connection between two different nodes for one week. The netperf tool successfully measured the throughput from one node, to two other nodes situated at different computer networks. In this case, the results did not match the predictions, and this had probably something to do with limited hardware resources at the nodes. It would have been interesting to remove that bottleneck, and have really tested the network throughput between two high performance nodes, or at least include the CPU of the measurement nodes during the measurement. But again the objective was to demonstrate the active throughput tools, and since netperf performed as expected, the experiment can still be classified as successful.

In Case three, the objective was to make use of active delay measurement tools, to benchmark the network connection between three different nodes for one week. The round trip time measurement that was done in the experiment can be used to find both the delay and the jitter of a connection. These three properties provide important information about the quality of service for the connection. And with the available information relative important assumptions could be done to describe the quality of service for these network connections. This helps the system administrator in designing the network for services that require different quality of service properties. This experiment can definitely be classified as successful.

## BIBLIOGRAPHY


[1] Andrew S. Tanenbaum. Computer Networks, Fourth Edition. Prentice Hall, 2003.

[2] Kevin Hamilton Kennedy Clark. Cisco LAN Switching (CCIE Professional Development). Cisco Press, 1999.

[3] Annabel Z. Dodd. The Essential Guide to Telecommunications, Second Edition. Prentice Hall PTR, 1999.

[4] Sergio Verdú. Wireless bandwidth in the making. IEEE, 2000.

[5] Gene Spafford Simson Garfinkel, Alan Schwartz. Practical Unix & Internet Security, 3rd Edition. O'Reilly, 2003.

[6] Fred Halsall. Data Communications, Computer Networks and Open Systems, Fourth Edition. Addison-Wesley, 1996.

[7] William Stallings. Local networks. ACM, 1984.

[8] Mahbub Hassan and Raj Jain. High Performance TCP/IP Networking. Pearson Prentice Hall, 2004.

[9] W. Richard Stevens. The Protocols (TCP/IP Illustrated, Volume 1). Addison-Wesley, 1993.

[10] Matt Bishop. Computer Security, Art and Science. Addison-Wesley, 2002.

[11] Cross-Industry Working Team. Internet service performance: Data analysis and visualization. Technical report, The Cross-Industry Working Team (XIWT), 2000.

[12] Mark Burgess. Analytical Network And System Administration. Wiley, 2004.

[13] B. Kleiner P.A. Tukey J.M. Chambers, W.S. Cleveland. Graphical Methods for Data Analysis. Duxbury Press, 1983.

[14] Lionel M. Ni Xipeng Xiao. Internet qos - a big picture. IEEE, 1999.

[15] Bruce S. Davie Larry L. Peterson. Computer Networks - A System Approach, Second Edition. Morgan Kaufmann, 2000.

[16] Joseph D. Sloan. Network Troubleshooting Tools. O'Reilly, 2001.

[17] Rick Jones. The netperf website. http://www.netperf.org/, May 2005.

[18] The Board of Trustees of the University of Illinois. The iperf website. http://dast.nlanr.net/Projects/Iperf/, May 2005.

[19] The ttcp website. http://www.pcausa.com/Utilities/pcattcp.htm, May 2005.

[20] The tcpdump website. http://www.tcpdump.org/, May 2005.
[21] Paul Herman. The tcpstat website. http://www.frenchfries.net/paul/tcpstat/, May 2005.
[22] Sugih Jamin Amgad Zeitoun, Zhiheng Wang. Rttometer: Measuring path minimum rtt with confidence. Technical report, The University of Michigan, Ann Arbor, 2003.
[23] Stefan Savage. The sting website. http://www.cs.washington.edu/homes/savage/sting/, May 2005.
[24] PING 127.0.0.1 Computer Services. The ping website. http://www.ping127001.com/pingpage.htm, May 2005.
[25] Zhiheng Wang Amgad Zeitoun.The rttometer website. http://idmaps.eecs.umich.edu/rttometer/, May 2005.
[26] The pinger website. http://www-iepm.slac.stanford.edu/ pinger / tools /software.html, May 2005.
[27] Tobias Oetiker. The smokeping website. http://people.ee.ethz.ch/ oetiker/webtools/smokeping/, May 2005.

[28] Dictionary.com. Dictionary.com’s website. http://dictionary.com/, May 2005.

[29] Daniel P. Siewiorek Jim Gray. High-availability computer systems. IEEE, 1991.

[30] Miroslaw Malek Allen M. Johnson. Survey of software tools for evaluating reliability, availability, and serviceability. ACM Computing Surveys, 1988.