Benefits and tradeoffs of application-specific WAN acceleration in different bandwidth, latency and loss scenarios.

Research Report for RP2
University of Amsterdam
MSc in System and Network Engineering

Class of 2005-2006

Dirk-Jan van Helmond, Marc Smeets
{dirkjan,msmeets}@os3.nl

July 4, 2006

Abstract

Corporate collaboration software as used in the Microsoft platform is focused on functioning in LAN environments, where the communication characteristics of the network typically show low delay, high bandwidth and none or very low loss. But nowadays, many users want to access their files and mail from anywhere, anytime. Therefor this data often has to traverse WAN links. These WAN links have very different characteristics than LAN links, e.g. low bandwidth, higher delay and sometimes even considerable amounts of loss. When the protocols that are used are not optimized for these types of connections, the impact on the throughput can be severe.

Several manufacturers have developed products dedicated to increase the performance of these LAN protocols over WAN connections by eliminating protocol inefficiencies. Juniper Networks offers such application aware acceleration in her WX platforms. These appliances are dedicated to improve application performance over WAN links with a number of transport and application-specific optimizations. The goal of this project is to evaluate the performance benefits and tradeoffs encountered when deploying the WX platform over links with different bandwidth, delay and loss characteristics.

Tests done for this research show that the inefficiency inherent with the CIFS and MAPI protocol as used in older Outlook clients can be improved dramatically with the use of acceleration by the WX platform. We've measured noticeable throughput increase in all types of links with a delay of at least 10 to 30 milliseconds. The throughput increase in high bandwidth links was noticeable better than on low bandwidth links, mainly because the low bandwidth links were earlier saturated. Overall, improvements in throughput between 200% and 600% were found very common. On very high delay links the throughput increase even reached over 5000% in some settings.

Contents

1 Preface

This research is done as part of our Master of Science study in System and Network Engineering at the University of Amsterdam. The research is done on the field of WAN acceleration and done on behalf of Juniper Networks, Schiphol-Rijk.

2 Research goal

The goal of this project is to evaluate the performance benefits, and possible tradeoffs, encountered when deploying the Juniper WX platform over links with different bandwidth, delay and loss characteristics. In order to be complete, we need to deliver the following three things:

- A fully functional lab setup that can be used for testing several different scenarios with different link characteristics
- A setup with application servers that can be used over the lab setup
- A test plan and a report with the results from several realistic network scenario tests

3 Theory of acceleration

Before there can be spoken about WAN acceleration in-depth, there needs to be a descent knowledge of the theory behind acceleration. This topic explains the different approaches. But first there is a need for proper definitions of terms used throughout this paper.

3.1 Definitions

A couple of terms will be used throughout the paper. Although the definition might seem straightforward, confusion might arise when the definitions are not clear. Therefore these definitions will be stated here:

Bandwidth is the amount of data that can be send through a system (mostly a wire if we are talking about WAN) in a given time. Bandwidth is mostly expressed in the form of the amount of *bits per second*. Much used quantities are given a name (T1, E1, OC-1, et cetera). Important to notice is that the bandwidth of a (WAN)link is expressed many times as the maximum amount a wire is physically capable of, or limited to.

Delay is the amount of time it takes for a PDU[1] to be delivered. More accurate: it is the time needed for a PDU to be *put on the wire*, a switch or router needs for processing it and make forwarding decisions, travel a geographical long distance, et cetera, until it is delivered at the receiver.

[1] PDU – Protocol Data Unit

Latency is many times confused with delay. Latency represents the time caused by delay plus the time needed for processing the packet. Many times, latency is measured as the round-trip-latency where it stands for the time it takes before the source of the packet receives an answer back. Round-trip latency can be measured easily with the use of the `ping` program as it acts on the TCP/IP stack of the hosts and needs CPU processing time.

Loss stands for the loss of a packet traversing the network. There can be many different reasons for loss but the most common are collision and errors in routing. Recovery from loss due to collision is self controlled because of the protocols operating at different lower OSI layers. Recovery of loss due routing errors aren't always auto-corrected. In case of a (D)DoS attack you even want the packets to be lost. This intentional discarding of packets is called *null-routing*. Loss can be a serious problem of which the results aren't always easy to predict and can have serious impact on the throughput on a link.

Retransmission happens when a packet needs to be sent again. The cause for this retransmission can be actual loss of the packet, in which the sending side can decide to send the packet, or can be corruption of the packet in which the receiving side send a request for retransmission. In both cases the impact on the actual throughput of a link can be significant.

Throughput The actual amount of bandwidth that is being recorded over a link. This is the maximum bandwidth, minus the loss of latency and delay, loss of packets and retransmission. The difference between bandwidth and throughput can be significant, and is mostly bigger on WAN links than on LAN links.

Traffic Shaping is the general term of different ways of prioritization of traffic. The way traffic is categorized for making prioritization decisions, as well as the way traffic is handled, differs per used method for bandwidth throttling and rate limiting. The two predominant ways of traffic shaping are Leaky Bucket and Token Bucket. Sometimes traffic shaping is done by devices that also support different queueing mechanisms like First In First Out, Fair Queueing and Weighted Fair Queueing.

Leaky Bucket is a mechanism for shaping bursty traffic in a way that it comes out of the FIFO queue in a nice steady stream of packets. The size of the queue is limited. Any incoming packet while the queue is full will be lost.

Token Bucket only differs from the Leaky Bucket in the way that it has ways for incoming traffic to bypass the traffic shaping and to burst out of the queue. An administrator can define tokens in the bucket. Per token the output rate is customizable. That way, the Token Bucket mechanism allows to shape all the incoming traffic but also let some specific traffic exit the bucket at a higher rate.

LAN or Local Area Network is the network in the local surroundings. This can be a small network at home, or a office network spread over a couple of floors. Although not strict, the limit in span of a LAN is at one building,

or a few when they are near each other. In a LAN speed and bandwidth are high while latency, delay and loss are low.

WAN or Wide Area Network is the network that spans a wider distance. It could be the uplink to the Internet via an ISP, or it could be a POTS or xDSL connection to branch offices of the same company. In a WAN, speed and bandwidth are significant lower, and latency, delay and loss are higher compared to a LAN.

WAFS Wide Area Filesystem is a not a specific file system but the name of the concept of having a file server appear to be local (in both available files and response time) while it is in fact separated by a WAN connection. This concept can be realized by different techniques like caching and acceleration. If successful, WAFS give offices the opportunity to let employes from all the different offices access the same file server over the WAN connection like it is situated locally.

Any other definitions might be explained in-line, or are expected to be common knowledge.

3.2 Acceleration theory

During the years it has become clear that many protocols we use are not being used in the most efficient way. The reasons for this inefficiency lies in designing errors, implementation errors and a changing environment in which the protocols are being used. With the ever increasing need to access information from everywhere, every time, people have started to use protocols that were typically designed for LAN environments over WAN links.

Different link characteristics

WAN links have very different link characteristics than LAN links. In a LAN environment, connections are currently standardized on 100 Mbit/second and 1Gbit/second. The bandwidth on dedicated WAN links is usually much lower and the available bandwidth of high speed WAN links (excluding the academic networks) are usually shared between several companies. Also WAN links show a much larger delay in round-trip-time due tot the simple fact that the propagation of information is bound by the speed of light. Links over longer distances, often with several hops in between can show delays in excess of 250 ms, which is almost an eternity for LAN protocols as they where designed with low delay links in mind.

Protocols that suffer on WAN links

The applications that suffer the most on WAN links are applications that expect a low round-trip-time and cope bad with the large round-trip-times that is induced by WAN links, for example protocols that expect an acknowledgment for every sent PDU. This problem can be compensated to some level by making the PDU's really large, so that the *bandwidth*delay* product (BDP [6]) is still large enough to fully utilize a WAN link.

Some protocols that are notoriously bad when it comes to WAN performance are the *Common Internet File System* (CIFS), the *Network File System*

(NFS) and the *Messaging Application Programming Interface*, the protocol used for communication between Microsoft Outlook clients and Microsoft Exchange servers. These protocols expect an acknowledgment for every read or written block, before sending the next block. This is mainly for reliable writing to network storage but it imposes a tremendous burden on the BDP, making the protocol thus terribly inefficient.

TCP optimizations

TCP has been designed a long time ago. Since the 4.2BSD release in 1982 where TCP made it's appearance, it has been a much used protocol. Since around the mid-1990's TCP/IP has been the major protocol suite on the Internet. Link characteristics have changed a lot since the early days of TCP. During the years, the different TCP/IP implementations have been optimized and fine-tuned many times. But actual changes in the design to make use of the newer link characteristics have not happened that much. The most important changes in TCP of the last few years that are accepted as Internet standards are:

Selective Acknowledgment [7] is a new packet that informs the sender of data that has been received. This has tremendous benefits in situations where multiple packets from the same window (X) are lost because there is no need for sending information about a single lost packet per ACK, resulting in less round-trip-time (X-1).

Increasing TCP's Initial Window [8] proposes to increase the permitted initial window from two segments to roughly 4K bytes.

Limited Transmit [9] is a mechanism that can be used to more effectively recover lost segments when a connection's congestion window is small, or when a large number of segments are lost in a single transmission window.

Appropriate Byte Counting [10] is a way of enlarging the window size used in the session in a faster way based on the number of previously not acknowledged bytes in each ACK packet.

Early Retransmit [11] is a mechanism that reduces the amount of duplicate acknowledgments needed for a fast retransmission, in a situation with a small congestion window.

Extensive tests done in 2004 [12] have shown that, although some of these changes date back from 1996 and 2000, not all of these are implemented yet. 50% Of the tested servers does not support Limited Transmit, and only 30% of the server that advertise the SACK option actually use it. Unfortunately does not every improvement results in the suggested performance improvements. For example does SACK not offer any noticeable improvements. But not all changes are bad as showed that Limited Transmit option has significant improvements on transfer times. There must be noted that improvements of a protocol can't be fully used until the majority of systems fully uses the options.

Caching

Locally caching of content that otherwise needs to be retrieved via a slow WAN connection is perhaps the easiest way of improving the throughput. The benefit

is tremendous, but there are two drawbacks caching can not overcome. First isn't all content suitable for caching. Caching traffic like VoIP, IM, and others that are constantly changing has no purpose. A lot, if not most, of the traffic is therefor not suitable for caching. Secondly is caching of encrypted traffic impossible. As every raw stream of encrypted traffic has a different set of bits, caching of that stream of bits has no use.

But caching does have benefits for some specific protocols. Perhaps the most widely known is DNS. As the content of DNS traffic does not change rarely, and isn't encrypted in the lookup traffic, DNS is a clear example of where caching makes sense. Other protocols that can be used in caching are HTTP and email protocols. A caching box for the latter needs to have deep knowledge of the specific protocol used, but the benefit can be large as email destined for several hosts on the same LAN can be a couple MB. HTTP traffic can be cached as some (parts of) pages are static.

Of the proposed protocols usable for caching only DNS has build in capabilities for caching. HTTP and email have need for smart application aware caching boxes. This function is some times used in proxy servers.

4 Acceleration in the Juniper WX platform

Many vendors are starting to develop appliances that speed up performance over WAN links. Cisco had bought Actona [16], a small startup in the field of Wide Area File Systems. Juniper Networks has jumped into the *acceleration market* by acquiring Peribit [17]. As explained in chapter 3.2 acceleration is typically based on a couple of simple principles:

Compression and caching Keep a dictionary of recurring patterns to improve bandwidth and delay performance.

TCP optimization Improve the *bandwidth*delay* product so that more data can be in transit at any given time.

Application specific acceleration Intervene in protocol specific headers or functionality to improve the performance of specific protocols.

The technical implementation of these principles is pretty much the same between the (mostly proprietary implementations) of most vendors, therefor, we will explain them in a little more detail by means of the Juniper implementation in the WX platform.

The WX platform supports several different kinds of (proprietary) acceleration technologies to increase the efficiency of the available bandwidth [18].

4.1 Compression and caching

The WX platform supports data stream compression through the use of what Juniper calls *Molecular Sequence Reduction* (MSR) and *Network Sequence Caching* (NSC). MSR and NSC are pattern matching and caching algorithms that look for patterns in data sequences and replaces them with tokens. Because the WX on each side communicate with each other, pattern dictionaries can be easily maintained and managed. MSR operates in memory and is primarily focused

upon reduction of small, often recurring patterns in the range from 50 bytes up to 500 kilobytes. NSC operates with dictionaries on disk and supports larger, less often recurring patterns from 100 kilobytes up to whole files.

4.2 TCP optimization

The WX platform performs several technologies to increase the performance of TCP over WAN links. This is called *Packet Flow Acceleration* by Juniper Networks. *Fast Connection Setup* improves the performance of short-lived sessions by locally acknowledging session requests for active destinations. This reduces the setup time by one round-trip-time. TCP windowing problems are circumvented by the *Active Flow Pipelining* technology. The local and remote WX devices terminate the TCP connection locally after which the data stream is transported in a more efficient transport protocol for WAN links between the WX devices. A third technology can by used on lossy links with relative high bandwidth and is called *Forward Error Correction*. This technology sends an extra stream of error correction information along with the data stream. This enables the remote WX device to reconstruct packets that were lost during transmission.

4.3 Application specific acceleration

Juniper's WX platform has the ability to perform acceleration at the application layer (*Application-Specific Acceleration*). Currently the WX platform supports the protocols CIFS, HTTP and MAPI . Of MAPI the latest (2003) version is not supported at the moment.

The bottleneck of the *CIFS* and *MAPI* protocol is that the client and server wait for an acknowledgment for every requested block. Read performance acceleration by the WX device is performed by requesting N blocks, following the one requested by the client. Write performance acceleration by the WX device is performed through acknowledging and caching blocks sent by the client, and discarding the acknowledgments it receives from the server [19]. HTTP acceleration is performed by caching and pre-fetching static content from websites requested by the client [20].

5 Lab setup

The tests are performed on a specially designed lab to test specific parts of the acceleration technology of the WX platform. The lab is designed in a way there is the least possible impact of different parts of the network. The used hardware, software and configured setup are explained below.

5.1 Physical configuration

The physical configuration is setup to simulate a connection from a remote office to a central office with a WAN link in between, with a service on one side of the link and a client using the specific service on the other side of the link. The link itself can be configured to simulate several different WAN scenarios. In this way, we can test performance of the application over different types of links. The measurement will be done outside of the two systems used, by a separate protocol analyzer that gets a copy of all client-to-server traffic through a span port on the switch at the client side. This setup is showed in figure 1.

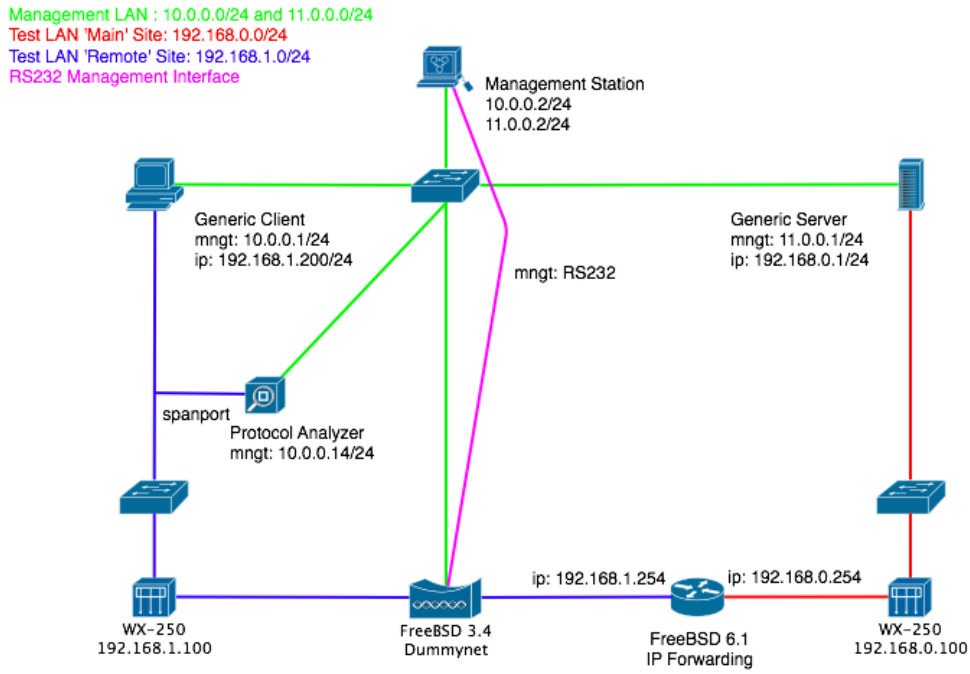

Figure 1: Layout of the test lab

Client and Server hardware

The client and server are recent Dell server systems with plenty of CPU power and memory to exclude these parts as bottlenecks in the tests. The NICs are 1Gbit/second full duplex interfaces. Because we connected the servers with FastEthernet switches to the WAN accelerators, the link will be automatically negotiated at 100 MBit Full duplex; the maximum available link speed.

WAN accelerator hardware

The WAN accelerator hardware used are two Juniper Networks WX-250 platforms [1]. These appliances are placed inline with the WAN link.

5.2 Software environment

The acceleration of the WX platforms is focused on the Microsoft Windows[2] platform. The lab setup will therefor be designed around a Microsoft Windows Client/Server environment suited for file sharing and the Microsoft Exchange[3] collaboration service.

Client and Server software

The primary focus of these tests is the *application-specific* acceleration. Because the WX platform primarily accelerates the Microsoft CIFS and MAPI protocols, the choice for server and client operating system are Microsoft Windows 2003 Server and Microsoft Windows XP client software. For the CIFS [4] tests the standard Windows File Sharing system can be used. For the MAPI [5] tests, we use the Microsoft Exchange 2003 Server system. Microsoft has changed the MAPI protocol for communication between Exchange 2003 and Outlook 2003. Because the WX platform cannot fully accelerate the new MAPI protocol [22], we will be testing MAPI performance both with the older Microsoft Outlook XP and Microsoft Outlook 2003 client software.

WAN Accelerator configuration

The WAN Accelerator supports several different kinds of acceleration. For our tests we have divided the acceleration techniques in two groups. First, there are the more traditional acceleration techniques like caching, compression and acceleration at TCP layer by optimization of the window size to increase the *bandwidth*delay* product. The second acceleration technique is a Juniper proprietary technology at the application layer. Our goal is to measure the impact of this application specific acceleration. Therefor variables in the lab setup will be in a way that we can test this specific acceleration. The results of the tests can be compared with a baseline test and the *regular* acceleration, achieved through compression and TCP acceleration to signify the performance impact.

WAN link simulator

The WAN link simulator is a dedicated FreeBSD Dell server configured with the *Dummynet* [24] software to simulate characteristics of a WAN link. The parameters delay and bandwidth can be configured with this tool. The Dummynet software was selected best out of a couple of available products with similar functionality. The software was calibrated to measure the impact of the simulation hardware and the actual accuracy of the simulation software. Appendix A shows the tests done and it's results.

WAN routing

For acceleration, the source and destination system had to be on different subnets, otherwise the WAN accelerators didn't mark the traffic for acceleration.

To accomplish this, there is a router placed in the WAN link. This router is a dedicated Dell server configured with FreeBSD and IP forwarding enabled. IP forwarding has impact on delay characteristics. This impact will be compensated because all results will be relative to a baseline test.

6 Getting results from the tests

To get results from the tests, there is need for a separate system that measures the time it takes for a particular file transfer. This way it is possible to calculate the actual transfer rate. For this purpose we have placed a protocol analyzer on a span port that receives a copy of all client-to-server traffic.

6.1 Empirical testing

Please keep in mind that all results are gathered by empirical testing. This means that the results are measurements done in an actual lab environment with external influences of all kind, e.g. hard disk performance, background operations in servers and appliances, and real loss and delay on the links. Many tests have been performed just once or twice because of limited time. We've strived for the best circumstances by using fast and reliable hardware. Some results deviated from what we expected. This could be because of a measurement error, a bug of a configuration error. Because of limited time, we haven't seen the time to do some serious investigation into these anomalies. Therefor, it can be possible that some results in this paper are wrong by some degree when compared to extensive and repetitive testing.

6.2 Keeping the link clean

To make sure that the results are *clean*, without interference of other protocols, we have disabled all other protocols on the WAN link. Management of the systems and appliances are all performed over a separate management LANs or through serial ports to make sure that only the desired traffic traverses that WAN link.

6.3 Measurement

The difference in time between several tests is of importance to us. Therefor, we have to make sure that time is measured accurately. The measurement of the tests will be performed through packet inspection. A protocol analyzer will be placed in the path of the client and the local WX device. The protocol analyzer that we used is *Ethereal 0.10.13* [14]. We chose for an older, less cutting-edge version and not for the latest version, which is called *Wireshark* 0.99.1pre1 [13].

Time

CIFS and MAPI, the two protocols that we use for testing, communicate in blocks. Time measurement is performed by counting the difference in time between the moment of the client request for the block with offset *0* and the acknowledgment of the last TCP segment of the last block. The first block the the block containing the data transfer request, the last block is the block in

the same TCP stream that contains the last part of the transferred data. The PCAP file will also be checked if there are many retransmissions of blocks due to loss or other problems. If a PCAP file had more than 1% of undesired traffic (e.g. other protocols, retransmissions) the test was performed again.

Volume

The volume of each request will be the same within specific protocol tests. All test will be done with files of exactly the same size. For low bandwidth links we will use 10 MB binary files, for high bandwidth links 100 MB binary files will be used.

Data used for transmission

The files that are used in the scenarios are binary files generated from */dev/urandom*. The files are all used only once. This is done because the acceleration technologies used in the WX platform can only be enabled in conjunction with the caching and compression algorithms. In our tests, we only want to measure the performance of the acceleration technologies and not the compression and caching algorithms. By using binary files with random data, the compression functionality is circumvented, because random data cannot be compressed. By using the files only once, the caching functionality is circumvented.

The size data to be transferred depends on the *bandwidth*delay* product. Very short transfers are unreliable to measure. The data transmission has to be long enough so that the throughput can stabilize to get a good throughput reading. This usually happens within a couple of seconds. Therefor as a rule of thumb, we make sure that all the transfers are at least 5 seconds long. For the fast links, we will use a 100 MB file, the slow links can be tested with 10 MB files.

7 Test setup

The tests are performed in a predefined way. The goal of the tests and the actual tests performed are described here.

7.1 Test targets

The goal of these tests is to test the **Application Specific Acceleration** performance of the WX appliances. Besides that, we want to know the performance increase of the WX platform over the protocol optimizations put in place by Microsoft in the new MAPI protocol.

Primary target

Our goal for these tests is measurement of the benefits and tradeoffs of the application specific acceleration techniques (layer 7 acceleration) implemented by the Juniper Networks WX productline (AppFlow technology) for the MAPI and CIFS protocol. Therefor, measures are taken to exclude the caching and compression by the WX platform. We have performed several tests in different environments with different bandwidth, delay and loss characteristics. The

results of these tests are be compared against a baseline and traditional acceleration techniques like TCP optimization.

Secondary target

Microsoft, the main vendor of software that uses the MAPI protocol, and others are evaluating their own products for usage over WAN links. E.g. communication between Exchange 2003 and Outlook 2003 is being optimized by Microsoft. Microsoft used some of the same techniques as used in the WX platform, i.e. bigger blocks and compression. This makes Microsoft a competitor to the WX product. A second goal is to determine if Microsoft can accelerate their own protocol enough to make specific hardware like the WX platform unnecessary.

7.2 Protocols

The acceleration is specific for the CIFS and MAPI protocol in a typical Microsoft environment. Our tests will be focused on these protocols.

CIFS

The CIFS protocol is the protocol used for file and printer sharing between Microsoft Windows systems. These tests will be performed by transmitting a file from a share on a server to the local disk on a client.

MAPI with Outlook XP client

The MAPI protocol is the protocol used by Microsoft Exchange to communicate with email clients. The WX platform is able to accelerate the MAPI that is used by Outlook XP client and older.

MAPI with Outlook 2003 client

The MAPI protocol is the protocol used by Microsoft Exchange to communicate with email clients. The WX platform is unable to accelerate the MAPI that is used by the newer Outlook 2003 client, because of adaptations done my Microsoft on the protocol. The tests with this client will be performed with the Microsoft acceleration technique enabled and disabled to measure the performance increase of the adaptations done by Microsoft.

7.3 Acceleration tests

For every protocol in every scenario, there will be three tests done, to measure the performance impact of the acceleration. With the MAPI test, we will perform an extra test to compare the Exchange optimization with the WX acceleration.

Establishing a baseline

All tests are started by establishing a baseline for the used equipment for every protocol in every scenario. The baseline test is done in the lab environment with all systems in place, but any form of acceleration disabled. This shows the

performance of a normal environment, that can be compared to the tests that are performed with acceleration enabled. Table 1 shows an overview of the tests for this baseline measurement.

Acceleration	**Enabled**
Caching	no
Compression	no
TCP Acceleration	no
CIFS/MAPI Acceleration	no
Exchange Acceleration	no

Table 1: Baseline Configuration

Typical acceleration techniques

In this test the compression, caching and TCP acceleration are enabled. By optimizing the *bandwidth*delay* product, we could exclude transmission problems that occur on layers below the transport layer. Caching and compression need to be enabled to enable the other acceleration techniques. Because we are not interested in the performance of caching and compression, we measures described in chapter 6.3 are taken to exclude the impact of these techniques. The acceleration of Exchange 2003 with Outlook 2003 is enabled by default. Therefor we had to manually disable it. This can be done with a registry tweak within the Exchange server. [23]. Table 2 shows the tests done.

Acceleration	**Enabled**
Caching	yes
Compression	yes
TCP Acceleration	yes
CIFS/MAPI Acceleration	no
Exchange Acceleration	no

Table 2: Typical Acceleration Configuration

Application specific acceleration techniques

The last test is the test with the proprietary AppFlow technology enabled. Throughput performance measured in these tests are compared with the earlier tests to signify the acceleration. Figure 3 shows the tests done.

Acceleration	**Enabled**
Caching	yes
Compression	yes
TCP Acceleration	yes
CIFS/MAPI Acceleration	yes
Exchange Acceleration	no

Table 3: Application Specific Acceleration Configuration

Exchange Acceleration

The MAPI protocol between Microsoft Exchange 2003 and Outlook 2003 is optimized for links with lower bandwidths and higher delays by increasing the block size and enabling compression. For the MAPI tests we performed an extra test to compare the performance increase of the WX platform with the performance increase Microsoft's adaptations in the protocol. These extra tests are shown in table 4

Acceleration	**Enabled**
Caching	no
Compression	no
TCP Acceleration	no
CIFS/MAPI Acceleration	no
Exchange Acceleration	yes

Table 4: Exchange Acceleration Configuration

7.4 Test scenarios

We have designed several different scenarios with different loss and delay characteristics to simulate a real WAN environment.

Impact in different bandwidth scenarios

All tests are performed in two different bandwidth scenarios. For our scenarios we have chosen to simulate the common occuring WAN types of T1 (1,544Mbit) and OC-1 (51,84Mbit), as showed in table 5. We would have preferred to perform more tests with bandwidth values in between, but due to the short time span in which we could perform the tests, we chose to perform just these two tests.

The choice for a low bandwidth link is because the link is easily filled with traffic, even over larger distances, with larger delays. Therefor, we expect to see less performance increase over these kinds of low bandwidth links. The high bandwidth link on the other hand is much more difficult to fill. Especially over larger distances with high delay the performance of the link can drop dramatically. Therefor we expect to see a larger performance increase over these kind of links.

	Bandwidth
Scenario 1	T1-like connection (1,544Mb/sec)
Scenario 2	OC-1-like connection (51,84Mb/sec)

Table 5: Bandwidth Scenarios

Impact in different delay scenarios

We have chosen to simulate the delay with four different values: 0 ms, 30 ms, 100 ms and 250 ms, as showed in table 6. The 0 ms scenario is chosen to baseline the performance of the link as if it was local. The values for delay are derived

from average values for short (30 ms) and longer WAN (100 ms) links. The 250 ms value is added to test the scenarios in an extreme environment, to make any performance increase very visible.

	Delay
Scenario 1	Local LAN (0 ms)
Scenario 2	MAN/small WAN (30 ms)
Scenario 3	Trans-Atlantic line (100 ms)
Scenario 4	Extreme delay (250 ms)

Table 6: Delay Scenarios

Impact in different loss scenarios

Because of limited time, we haven't tested all scenarios in a packet loss environment. We have taken three scenarios in which we generated loss on a link. These scenarios are picked from scenarios with bandwidth and delay restrictions. This gives us the opportunity to compare these results against previously measured results.

The scenarios we picked for the loss scenario are all high speed links with OC-1 throughput. This is done because the error correction consumes extra bandwidth. In a low bandwidth scenario error correction would impact the performance more than it would improve it. For the protocols we chose MAPI with the Outlook XP client, and did an accelerated test over a delayed link (250 ms) and a non-accelerated test over a non-delayed link (0 ms). This way it can be made clear if error correction has impact on acceleration. We also did a CIFS test over a non-delayed and non-accelerated test. Table 7 shows the tests done.

	Bandwidth	**Delay**	**Protocol**	**Acceleration**
Scenario 1	OC-1	0 ms	MAPI	None
Scenario 2	OC-1	250 ms	MAPI	Application Specific
Scenario 3	OC-1	0 ms	CIFS	None

Table 7: Bandwidth/Delay scenarios used for loss tests

We generated increasing amount chance of loss in several scenarios over a link to measure the performance impact, as show in table 8.

	Chance of loss
Scenario 1	0,001
Scenario 2	0,005
Scenario 3	0,01
Scenario 4	0,05
Scenario 5	0,1

Table 8: Loss Scenarios

8 Test methodology

All of the performed tests are described in detail in this chapter. Each protocol has a separate set of tests that focuses on the particular scenario and acceleration that is possible.

8.1 CIFS test procedure

The CIFS tests are performed in a Microsoft Windows environment, with the server having Microsoft Windows 2003 Server installed, and the client running Microsoft Windows XP. The client accessed a share on the server from which it copies a file that contains 10 megabytes of random binary data. This test are performed several times with different acceleration settings and with different link characteristics to measure performance increase in several different configurations. The CIFS procedure has a few little points that we had to be aware about. The mounting of the share and opening it in a new window needs to be done before the network analyzer is started. We performed a ping to the server before and after the transfer is completed so we could see the start and end of the actual copying in the PCAP file.

Note

Microsoft Windows 2003 Server is configured to automatically sign SMB blocks in client/server communication. This functionality has to be disabled because its incompatible with the acceleration techniques of the WX platform. In all tests, the SMB block singing is disabled. To disable the SMB singing see the WX Operations Manual [21] page 206.

Pre-test procedure

These steps are taken whenever the used acceleration techniques are changed.

- Step 1: Configure the WX platform with the appropriate settings
- Step 2: Reboot the WX devices to make sure their cache is empty
- Step 3: Mount a network share from the server on the client
- Step 4: Open the share in a new window

It is important to open the share in a new window before starting the test, otherwise the opening of the window spawns a request to the server that could possibly influence the measurement.

Test procedure

Every test is started with reconfiguring the WAN link simulator with the desired characteristics. The changes work instant, so the test can be performed immediately after.

- Step 1: Configure the WAN link simulator with the appropriate settings
- Step 2: Start the traffic analyzer, write captured data to a file

- Step 3: Ping the server
- Step 4: Initiate the file transfer of the file from the network share to the local disk
- Step 5: Ping the server
- Step 6: Stop the traffic analyzer

The pings to the server before and after the file transfer are used to make it easier to identify the file transfer in the PCAP file.

Post-test procedure

After all tests are done, we analyzed the PCAP file with the use of a traffic analyzer. To measure the complete transfer, we took the request for the block with offset 0 as a starting point, and the acknowledgment for the last TCP segment of the last block as an endpoint.

- Step 1: Open the PCAP file
- Step 2: Note the time ($t1$) of the request for the SMB[2] with offset *0*
- Step 3: Note the time ($t2$) of the *ack* for the last tcp segment of the last SMB
- Step 4: Calculate Δt ($t2 - t1$)
- Step 5: Calculate the transfer speed by dividing the transferred bytes (10 of 100 MB) with Δt

8.2 MAPI test procedure

The MAPI tests are performed in a Microsoft Windows environment, with the server having Microsoft Windows 2003 Server installed, and the client running Microsoft Windows XP. The server also has Microsoft Exchange installed. The client accessed the server by use of the Outlook client it has installed. There are tests performed with Outlook XP and Outlook 2003 to differentiate between the acceleration of the two versions of the MAPI protocol. The mailbox of the clients contains several mails with 10 MB attachments that are downloaded by the client on queue. This test is performed several times with different acceleration settings and with different link characteristics to measure performance increase in several different configurations. The MAPI procedure also has a few points that we needed to address when performing the tests.

Pre-test procedure

These steps are be taken whenever the used acceleration techniques are changed.

- Step 1: Configure the WX platform with the appropriate settings
- Step 2: Mail enough 10 MB files to the account needed for testing

[2]Server Message Block

- Step 3: Disable mail preview in Outlook
- Step 4: Reboot the WX devices to make sure their cache is empty

The 10 MB files are mailed to the account before the WX is rebooted to clean its cache. This is done as the caching engine could otherwise influence the test results if the same file were to travel the WAN link twice. Mail preview is disabled, otherwise the Outlook client could start to pre-fetch attachments, which would cloud the test results.

Test procedure

Every test is started with reconfiguring the WAN link simulator with the desired characteristics. The changes work instant, so the test can be performed immediately after.

- Step 1: Configure the WAN link simulator with the appropriate settings
- Step 2: Open an unread mail with an attachment
- Step 3: Start the traffic analyzer, write captured data to a file
- Step 4: Ping the server
- Step 5: Save the attachment to disk
- Step 6: Ping the server
- Step 7: Stop the traffic analyzer
- Step 8: Close the mail

It is important that the mail is opened before the traffic analyzer is started. This is because it is difficult to distinguish the opening of the mail and the download of the attachment afterwards in the PCAP file.

Post-test procedure

After all tests are done, we analyzed the PCAP file with a traffic analyzer. To measure the complete transfer, we taok the request for the block with offset 0 as a startpoint, and the acknowledgment for the last TCP segment of the last block as an endpoint.

- Step 1: Open the PCAP file
- Step 2: Note the time ($t1$) of the request for the first RPC[3]
- Step 3: Note the time ($t2$) of the *ack* for the last tcp segment of the last block
- Step 4: Calculate Δt ($t2 - t1$)

[3]Remote Procedure Call

8.3 Overview of scenarios

All protocols will are tested with all combinations of environments. Table 9 shows a all inclusive overview of all the tests that are performed.

B - Baseline test, no acceleration
F - Forward Error Correction, no acceleration
T - Caching, Compression and TCP acceleration enabled
A - Caching, Compression, TCP and Application specific acceleration enabled
E - Exchange acceleration enabled (Only with Outlook 2003 client)

Bandw.	**Delay**	**CIFS**	**MAPI (XP)**	**MAPI (2003)**
T1	0 ms	B, T, A	B, T, A	B, T, A, E
T1	30 ms	B, T, A	B, T, A	B, T, A, E
T1	100 ms	B, T, A	B, T, A	B, T, A, E
T1	250 ms	B, T, A	B, T, A	B, T, A, E
OC-1	0 ms	B, T, A	B, T, A, F	B, T, A, E
OC-1	30 ms	B, T, A	B, T, A	B, T, A, E
OC-1	100 ms	B, T, A	B, T, A	B, T, A, E
OC-1	250 ms	B, T, A	B, T, A, F	B, T, A, E

Table 9: Scenario Overview (total of 80 tests)

9 Test results

This sections shows the tests done and the outcome of these tests. The result of every test is showed in two ways: a table showing the actual numbers, and a diagram showing a graphical overview of the results. Each table shows from left to right: the delay used, the throughput of the baseline, the throughput of the TCP acceleration, the benefit of TCP acceleration compared to the baseline, the throughput of the application specific acceleration (called AFA[4]) and the benefit of AFA compared to the baseline. In the MAPI 2003 tests the table is extended with the throughput of acceleration done by Microsoft Exchange and it's benefit compared to the baseline. All values of throughput are expressed in KB/sec. All values of benefit are expressed in percentages of the baseline of that delay. Some of the diagrams have got a logarithmic scale. This is done for better understanding of the figure. Where used, this is always noted in the text explaining that diagram.

9.1 Acceleration of the CIFS protocol

Low bandwidth link

One of the first things to notice in figure 2 is that the actual performance increase of TCP acceleration is not resulting in a higher throughput comparing to the baseline. Although the throughput of TCP acceleration decreases less quickly than the baseline, advantage of acceleration at TCP level is not being

[4]AFA — Application Flow Acceleration

achieved in delays from 0 to 250 ms. Most likely, TCP acceleration will only boost performance in situations where the delay exceeds 250 ms. But, the actual throughput will go towards the 20KB/sec and will therefor likely not be sufficient for situation where CIFS is being used.

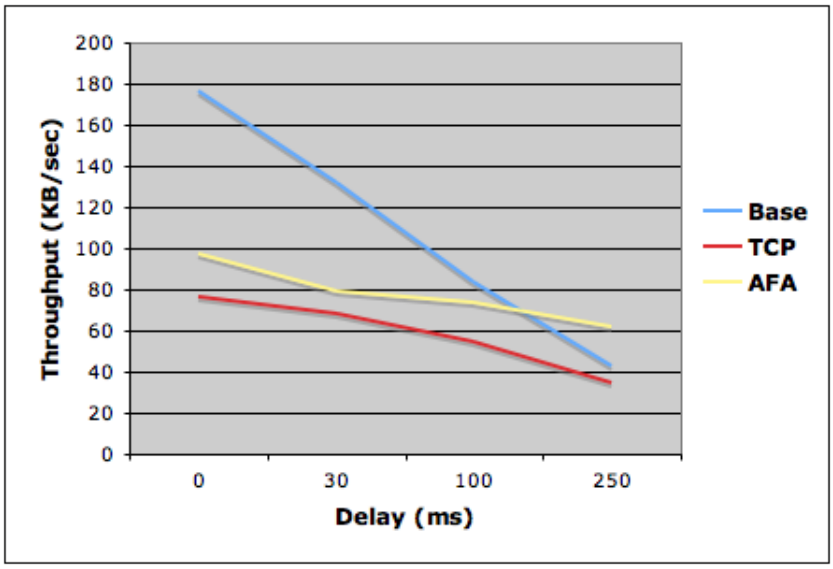

Figure 2: Throughput of CIFS acceleration at 1.544 Mbit/sec

Acceleration at the application layer does result in a throughput increase as can be seen at the yellow line in figure 2. But important to notice is that the actual increase of throughput can only be achieved at delays higher than 150 ms. At 250 ms AFA has a higher throughput of 20KB/sec. Although 20KB is 45.4% of the throughput of the baseline at that delay, and therefore is a noticeable acceleration, the actual throughput is only 62KB/sec. Again, this is most likely not sufficient for situations where CIFS is used.

Delay	Baseline	TCP	Benefit	AFA	Benefit
0ms	176.6	76.5	- 56.7%	97.5	- 44.7%
30ms	131.8	67.8	- 48.5%	79.4	- 39.7%
100ms	83.6	54.6	- 29.8%	73.7	- 11.7%
250ms	42.7	34.2	- 19.7%	62.0	45.4%

Table 10: Throughput in KB/sec of CIFS acceleration at 1.544 Mbit/sec

The increase of throughput AFA generates at high latency is much less than the loss made at lower delays. At 0 and 30 ms the loss is around 80 and 50 KB/sec. A quick glance at table 10 shows that the benefits are negative almost everywhere. Only AFA at 250 ms has a positive benefit. Overall, although depending on the situation, TCP acceleration has no benefits and application specific acceleration could not be enough for a proper use of CIFS.

High bandwidth link

Figure 3 reveals the strength of CIFS application acceleration. The only difference with the previous figure is the bandwidth of the link, which has now increased to the amount of 51.84 Mbit/sec. But the increase of throughput is huge. Not only does the benefit of AFA appear much earlier in the delay spectrum (around 20 ms), also is the actual benefit of the acceleration much more significant. The scale of figure 3 is logarithmic, but table 11 gives us clear understanding of the numbers of performance benefit.

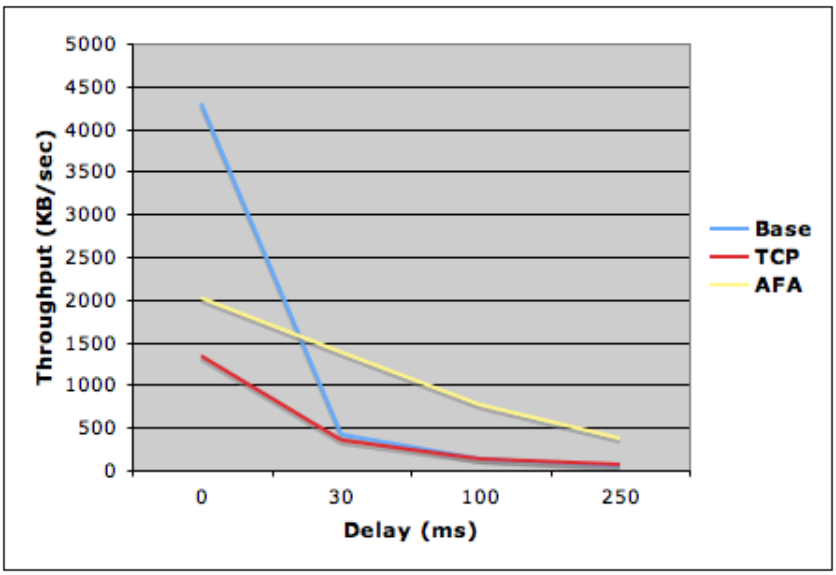

Figure 3: Throughput of CIFS acceleration at 51.84 Mbit/sec

Looking at the numbers of table 11, one can conclude that the acceleration of TCP does not have a true benefit. The overhead at the lower delays (around 70KB/sec at 30 ms) can not be turned into a significant benefit. Starting from around 80 ms TCP acceleration is a bit faster than the baseline with a 15KB/sec higher throughput at 100 ms and a 7KB/sec at 250 ms. So starting from 30 ms, TCP acceleration and the baseline are roughly the same.

Delay	Baseline	TCP	Benefit	AFA	Benefit
0ms	4294.6	1347.6	- 68.6%	2008.7	- 53.2%
30ms	425.4	357.6	- 15.9%	1373.4	222.8%
100ms	126.1	139.5	10.7%	774.7	514.3%
250ms	52.3	59.7	14.2%	375.4	617.4%

Table 11: Throughput in KB/sec of CIFS acceleration at 51.84 Mbit/sec

AFA shows that acceleration at the application level can be very significant. The turning point is around 20 ms, from which AFA is much faster than the baseline of no acceleration. At 30, 100 and 250 ms the application acceleration

has a benefit in throughput of respectively 222.8%, 514.3% and 617.4%. The overhead has a negative benefit of 50% at 0 ms, but still is almost 2MB/sec at that delay. Therefore one can conclude that application acceleration of CIFS at OC-1 has serious benefits.

9.2 Acceleration of the MAPI protocol (Outlook XP)

Low bandwidth link

Figure 4 demonstrates the throughput of acceleration of the MAPI protocol as used in communication between an Outlook XP client and Exchange 2003 server at 1.544 Mbit/sec. Turning point in this graph is around 15-20 ms, as from there the baseline has less throughput than both TCP and MAPI acceleration. Noticeable is that the yellow AFA line decrease slowly, while the lines of both the baseline and TCP are decreasing more heavily and not in the same steady form.

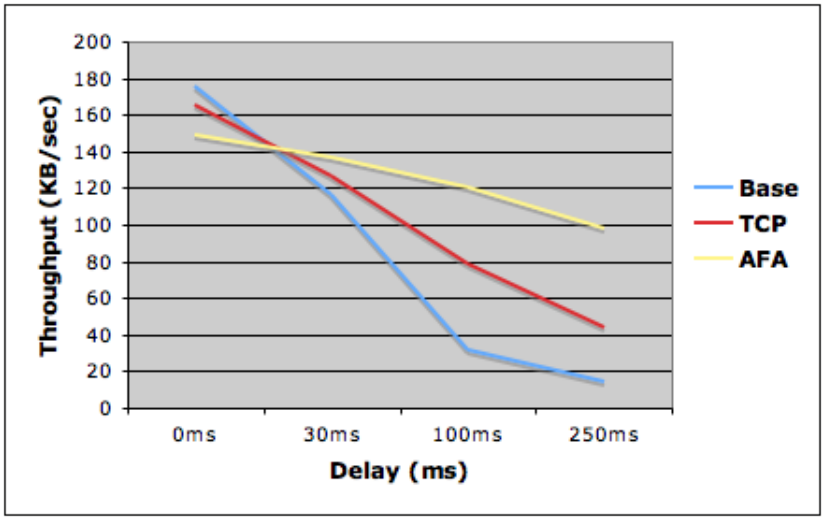

Figure 4: Throughput of MAPI (XP) acceleration at 1.544 Mbit/sec

Looking at the data of figure 4 and table 12 it is easy to see that there is a benefit from acceleration of both TCP and MAPI. At 0 ms the penalty for acceleration is 5.3% for TCP and 14.9% for MAPI acceleration. Starting from 30 ms there is a benefit of 8.8% rising to 208.2% for TCP acceleration at 250 ms. MAPI acceleration has an even higher benefit ranging from 17.9% at 30 ms to 585.7% at 250 ms.

High bandwidth link

Figure 5 and table 13 represent the acceleration of the MAPI protocol as used in Office XP at the speed of 51.84 Mbit/sec. Figure 5 has got a logarithmic scale.

Delay	Baseline	TCP	Benefit	AFA	Benefit
0ms	175.1	165.7	- 5.3%	148.9	- 14.9%
30ms	116.0	126.1	8.8%	136.7	17.9%
100ms	32.1	78.8	145.5%	120.1	274.1%
250ms	14.3	44.1	208.2%	98.2	585.7%

Table 12: Throughput in KB/sec of MAPI (XP) acceleration at 1.544 Mbit/sec

This scale gives a better view on the enormous benefit of acceleration at 250 ms, but makes it a bit harder to see the actual turning point. In this situation this is around 15 ms when application acceleration is performing better than the baseline, and around 30 ms when TCP acceleration is better than the baseline.

Delay	Baseline	TCP	Benefit	AFA	Benefit
0ms	3688.3	1191.4	- 67.7%	1969.9	- 46.6%
30ms	243.9	320.4	31.4%	1949.9	652.5%
100ms	44.3	126.2	185.0%	1706.2	3752.9%
250ms	17.9	15	- 16.4%	953.6	5205.5%

Table 13: Throughput in KB/sec of MAPI (XP) acceleration at 51.84 Mbit/sec

Looking at the raw numbers of table 13 it is noticeable that application acceleration has a tremendous higher throughput at 250 ms of 5205.5% regarding to the baseline. Also remarkable is that although there is a performance penalty at 0 ms, the actual throughput of AFA is roughly the same at 0 and 30 ms. The throughput drops only slightly when the delay is stretched to 100 ms, but at that delay there is already a benefit of 3752.9% compared to the baseline.

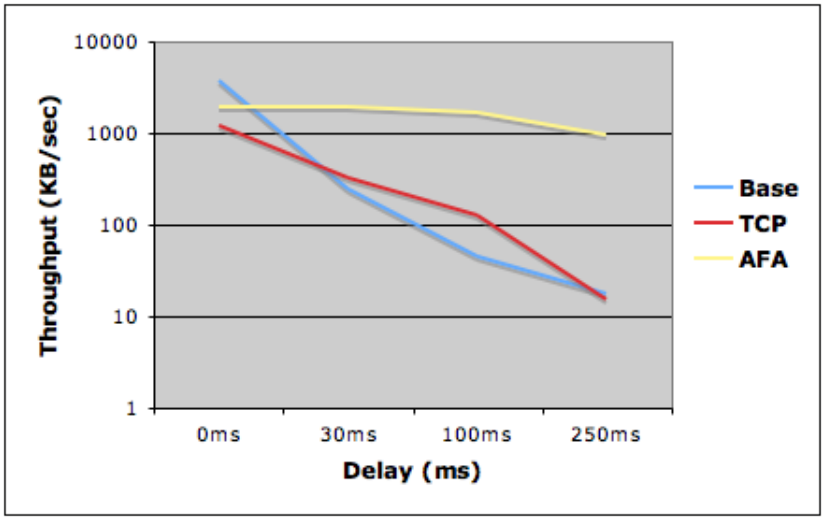

Figure 5: Throughput of MAPI (XP) acceleration at 51.84 Mbit/sec

Looking at TCP acceleration the most noticeable thing is that there is benefit when the delay of the link is between 30 ms and 220 ms. The peak of the benefit is around 100 ms, which results in a 185.0% higher throughput. But, the loss at 0 ms is 67.7%. This is a even higher loss than application acceleration. At 250 ms the loss of TCP acceleration is 16.4%, but one might expect to loose even more when higher delays are taken into account. This makes TCP acceleration not a clear choice, while application acceleration has a very clear benefit.

9.3 Acceleration of the MAPI (Outlook 2003)

Low bandwidth

Since Exchange 2003, Microsoft has putt effort in the acceleration of the MAPI protocol on the server side. This acceleration can only be benefitted from by using the 2003 version of Outlook. The authors have tested this acceleration at the speed of 1.544 Mbit, using different delays and comparing it to the acceleration done by the Juniper WX. One must take in account that Juniper's official statement is that the newest version of the MAPI protocol is not supported for acceleration at this time. The results of this acceleration, and the acceleration done by Microsoft can be seen in table 14 and figure 6.

Delay	Baseline	TCP	Benefit	AFA	Benefit	Exchange	Benefit
0ms	174.0	153.8	- 11.6%	164.9	- 5.2%	170.8	- 1.8%
30ms	116.1	110.1	-5.2%	124.5	7.2%	111.7	- 3.8%
100ms	59.1	57.8	- 2.2%	79.4	34.3%	58.7	- 0.9%
250ms	27.7	26.4	- 4.6%	44.5	60.5%	27.2	- 1.8%

Table 14: Throughput in KB/sec of MAPI (2003) acceleration at 1.544 Mbit/sec

Table 14 shows the throughput of no acceleration, TCP acceleration, MAPI acceleration done by the Juniper WX and acceleration by the use of Microsoft's new capabilities in Exchange 2003 and Outlook 2003. The graphical representation of these numbers gives a clear view on the fact that the acceleration done by Exchange is not noticeably different than the baseline of no acceleration. There are two things that stand out from this graph. TCP acceleration is slightly behind the others in situations where delays ranges from 0 ms to 30 ms. From 30 ms on it performs about the same as no acceleration and the acceleration done by Exchange. Note that both these accelerations actually suffer a small penalty resulting in a lower throughput of a few percent. The second thing to notice is that starting from 10 ms, the acceleration done by the Juniper WX outperforms all the others. The actual benefit of that last acceleration is 34.3% at 100 ms and 60.5% at 250 ms. It appears that the AFA acceleration benefits from higher delays.

Overall, the officially not supported AFA acceleration is the only acceleration that performs better than the baseline at this speed and does this better at higher delays.

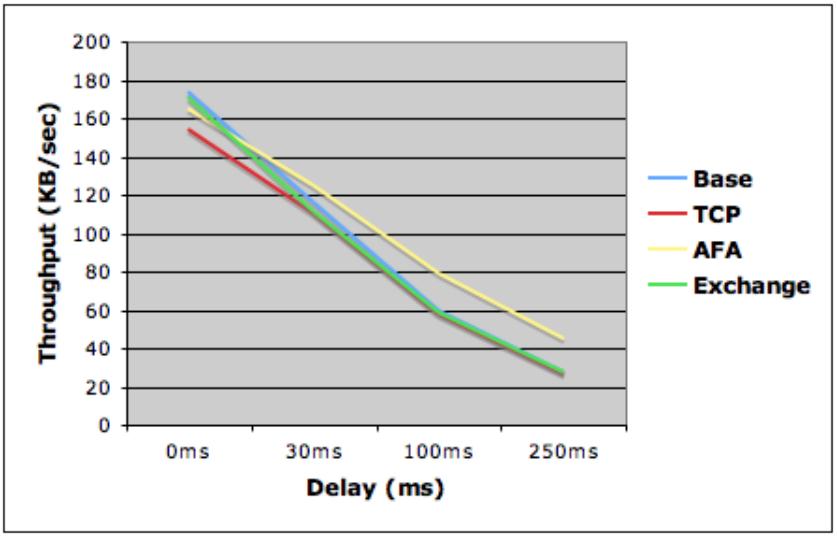

Figure 6: Throughput of MAPI (2003) acceleration at 1.544 Mbit/sec

High bandwidth link

At the speed of 51.84 Mbit/sec the results are slightly different from the results at 1.54 Mbit/sec. AFA still is the type of acceleration that has the highest throughput and benefits even more when the delay increases. But, when the the delay reaches 250 ms, both TCP acceleration and acceleration done by Exchange are performing better than the baseline, being Exchange the winner of the two as shown in table 15.

Delay	Baseline	TCP	Benefit	AFA	Benefit	Exchange	Benefit
0ms	3447.1	829.6	- 75.9%	1234.4	- 64.2%	2600.5	- 24.6%
30ms	244.9	183.3	- 25.2%	337.8	37.9%	243.4	- 0.6%
100ms	72.8	68.8	- 5.5%	131.2	80.1%	74.5	2.3%
250ms	25.5	28.7	12.5%	57.8	125.9%	29.9	17.2%

Table 15: Throughput of MAPI (2003) acceleration at 51.84 Mbit/sec

Figure 7 shows that, roughly, Exchange and the baseline follow the same path. But, although Exchange performs better starting from 100 ms, that increase in throughput is not as big as the loss made at 0 ms. At 0 ms Exchange has a 24.6% lower throughput and evens around 30 ms. From there on Exchange performs better when the delay increases, with being 17.2% faster at 250 ms. TCP acceleration performs worse by being 75.9% slower than the baseline, and leveling around 150 ms. Eventually it performs 12.5% better at 250 ms.

Acceleration done by the Juniper WX is much better, but also suffers a penalty. This significant penalty is about 64.2% or 2.2Mbit/sec at 0 ms. The throughput of AFA drops less heavily, as it equals to the baseline around 20 ms. From thereon, it's performance benefits increase as the delay increases. It tops out being 125.9% faster at 250 ms. Starting from around 20 ms, AFA acceleration also is faster than acceleration done by Exchange itself. At every

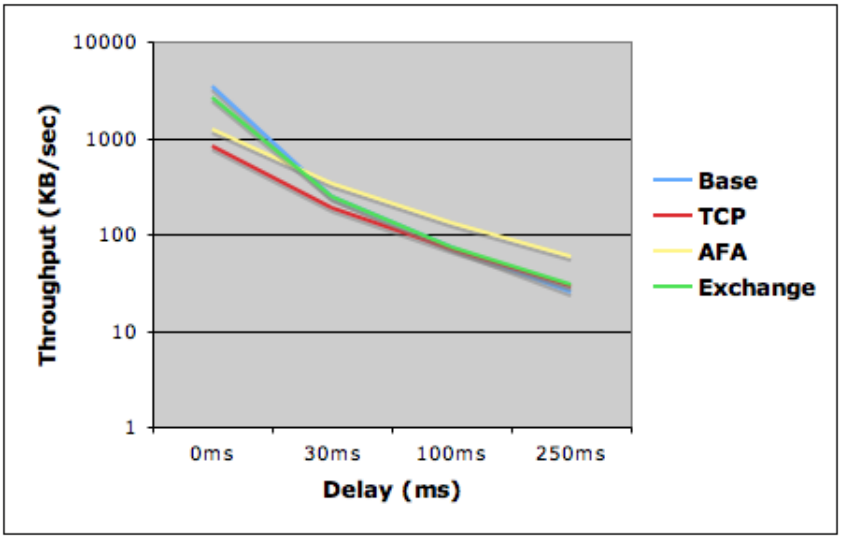

Figure 7: Throughput of MAPI (2003) acceleration at 51.84 Mbit/sec

delay, TCP acceleration is noticeable slower.

9.4 Loss of the CIFS protocol

One of the other functions of the Juniper WX is optimization of lossy links. The authors have tested the performance of this Fast Error Correction (FEC) by measuring the throughput of this option on different lossy links. The first of this measurement is done on a OC-1 link with no delay, and using the CIFS protocol. The baseline for this test is the result of the CIFS test at OC-1 without delay and without acceleration of any kind as tested in section 9.1. The throughput of this baseline is 4294.6 KB/sec.

Tabel 16 shows the results of the tests done. One of the first thing to notice is the lack of data in the row of 0.1 chance of loss. This is the result of the fact that is was impossible to successfully transfer a file via a link that is that lossy. Remarkable is that this was even impossible with the Fast Error Correction option enabled. The column called '% of base' shows the percentage throughput left over from the baseline when the specific chance of packet loss occurs. Remarkable here is that the chance of 0.005 generated less lost packets than the chance of 0.001. The reason for this is most likely coincidence.

As the second '% of base' column shows, there is no positive benefit of the FEC option in loss ranging from 0.1 to 0.005. Only at 0.001 is there a slight improvement of throughput, being 2.9%. This proves that packet loss doesn't occur only at lossy links, but also on normal "clear" links. The last row shows the benefit FEC generates compared to when FEC is turned off on the same lossy link. As one can see, FEC is mostly a negative influence. Only at the lowest tested chance is FEC actually improving the throughput by 54.7%.

Chance of loss	No FEC	FEC	Benefit
0.001	2856.0	4418.0	54.7%
0.005	3263.3	2090.6	- 36.0%
0.010	1202.8	1159.4	- 3.6%
0.050	190.3	153.0	- 19.6%

Table 16: Lossy non-accelerated CIFS at 51.84 Mbit/sec with 0 ms delay

Figure 8 represents the data in a graphical way. The blue line shows the baseline, being steady at every chance of loss as it is the maximum throughput that can be theoretical achieved. At a high chance of loss, the FEC option has no effect. Only at the two lowest tested chances, the lines differ.

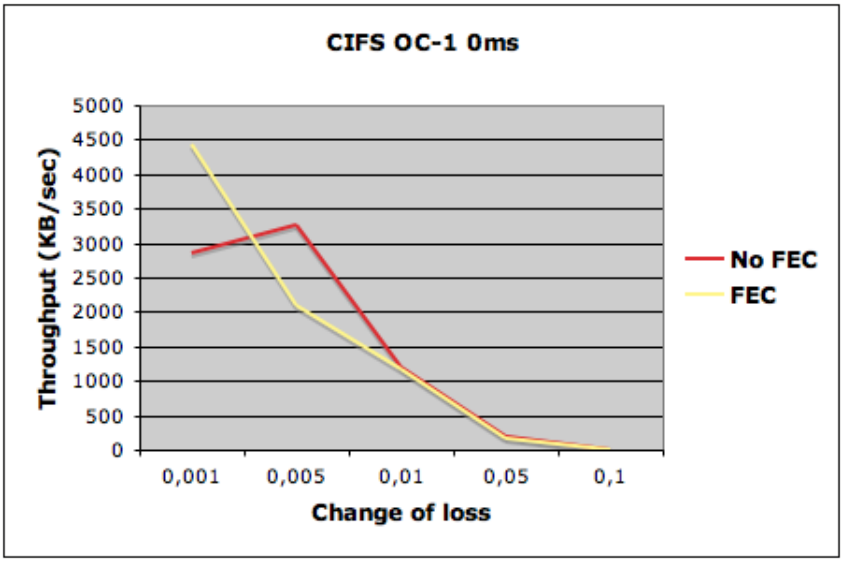

Figure 8: Lossy non-accelerated CIFS 51.84 Mbit/sec with 0 ms delay

9.5 Loss of the MAPI protocol (Outlook XP)

Low delay link

The second test of FEC is done with the baseline of the MAPI (XP) throughput at 51.84Mbit/sec without delay and without any acceleration. The throughput of this baseline, as tested in section 9.2, is 3688,317KB/sec. On of the conclusions drawn from figure 9 is that the throughput of FEC and no FEC are fairly the same at every chance of loss. Only at a achange of 0.01 FEC has a 40.9% increase in performance over No FEC. But, the throughput at that point is only 27.3% of the baseline, which puts it in perspective.

Looking at table 9 it appears that once again it was impossible to actually use the MAPI protocol at a 0.1 chance of packet loss. When testing at this chance of loss, Outlook responded that it was unable to contact the Exchange server, and therefor made testing impossible.

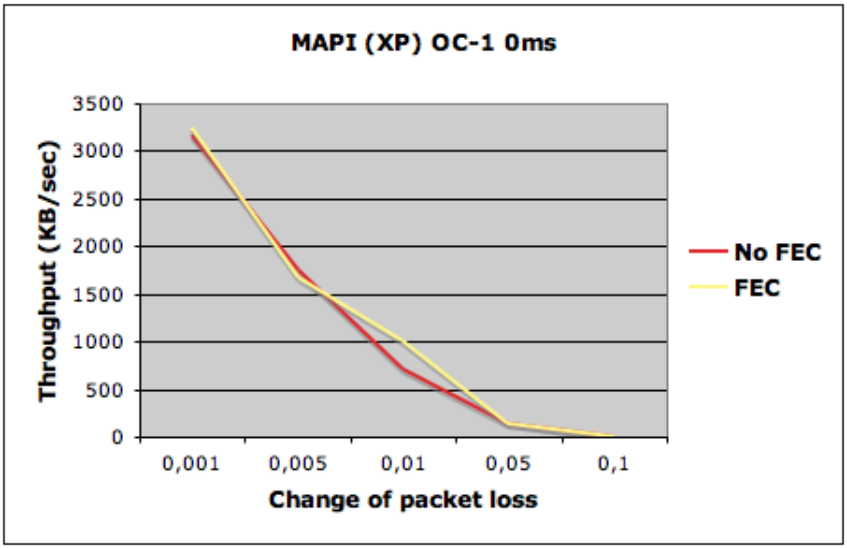

Figure 9: Lossy non-accelerated MAPI (XP) at 51.84 Mbit/sec with 0 ms delay

Chance of loss	No FEC	FEC	Benefit
0.001	3175.2	3253.2	2.5%
0.005	1752.4	1671.6	- 4.6%
0.010	715.4	1008.1	40.9%
0.050	134.7	131.3	- 2.5%

Table 17: Lossy non-accelerated MAPI (XP) 1.544 Mbit/sec with 0 ms delay

Overall does the FEC option seem to have a positive influence on the throughput. Compared to disabling the FEC option, FEC performs 40.9% better at a 0.01 chance of packet loss. This relatively large amount of performance improvement is most likely the result of a lucky situation of packet loss. Although the large value of the increase should be taken lightly, the fact that the value is positive proves FEC results in a performance increase. At a much lower rate of 0.001 FEC also has a positive influence. In the other situations. enabling FEC had a negative impact. While being 4.6% and 2.5%, this penalty is not severe.

High delay link

This test shows the use of FEC in application accelerated traffic on a lossy link. This is done in the situation of the MAPI (XP) protocol at 51.84Mbit/sec with a delay of 250 ms. The baseline used herein is the AFA accelerated throughput of 953,609 KB/sec taken from section 9.2.

Table 18 shows that this is the only test done where a chance of packet loss of 0.1 actually resulted in data. Noticeable is that the percentages at that rate, being 17.0% for No FEC and 16.3% for FEC, are higher compared to lower

Chance of loss	No FEC	FEC	Benefit
0.001	718.6	735.5	2.3%
0.005	618.8	588.0	- 5.0%
0.010	511.2	534.2	4.5%
0.050	299.5	277.0	- 7.5%
0.100	162.4	155.3	- 4.3%

Table 18: Lossy accelerated MAPI (XP) at 51.84 Mbit/sec with 250 ms delay

chances of packet loss in the two other tests.This is most likely the result of the fact that this test was done in the situation of a 250 ms delay.

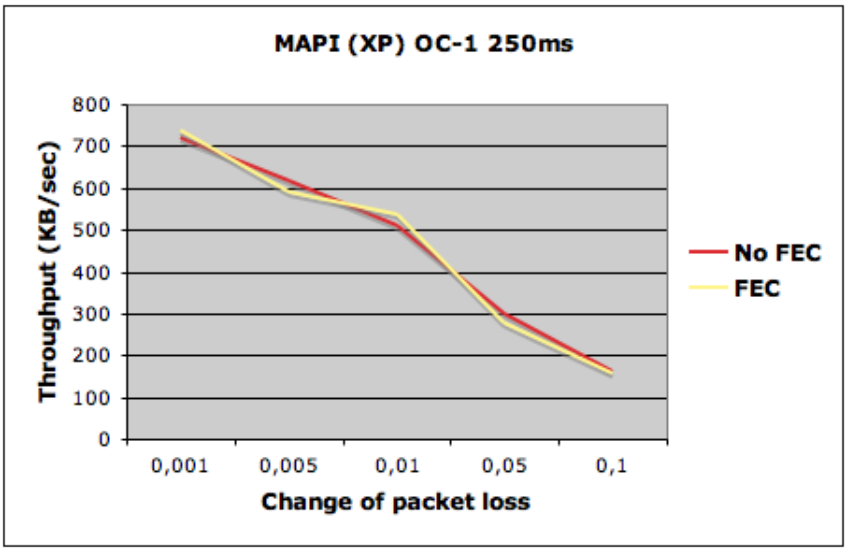

Figure 10: Lossy accelerated MAPI (XP) 51.84 Mbit/sec with 250 ms delay

Looking at the performance of FEC, one can conclude form figure 10 that at every chance of loss the throughput of FEC is about the same as when FEC is disabled. Only at a chance of 0.01 and 0.001, FEC is performing slightly better. At the two highest chances, FEC has a negative impact. Overall, it appears FEC has no specific benefit in this situation and might even have a slight negative benefit.

10 Other protocols

Protocols that are best accelerated are old protocols that were never designed for current LAN or WAN characteristics, protocols that are based upon older protocols and protocols that were just badly designed. This chapter lists some other protocols that could possibly be optimized for current LAN and WAN characteristics, and even be incorporated into the WX platform.

NFS

NFS, the Unix *Network File System*, is essentially the same protocol as CIFS for Microsoft Windows. The NFS protocol suffers from the same *design flaw* as the CIFS protocol in the way that it waits for acknowledgment for every block. Therefor, it should be relatively easy to adapt the WX platform in to make it accelerate the NFS protocol.

It was our intention to test if the WX platform could accelerate NFS in the same way as CIFS. Unfortunately, we haven't gotten round to it due to the short amount of time we had at our disposal. T

DNS

DNS is a protocol that is used on every network. Almost every network connection is preceded by a DNS lookup. Slow lookups can have a severe impact on network performance. Especially in an Active Directory environment, where the client is configured with the DNS servers of the domain that is on a WAN link, lookups over this WAN link can have an enormous impact on the client performance. An 'inline' caching DNS server that could cache requests for a certain time could speed up performance, because all following requests could be served from the cache of the near WX appliance. This technology would be different from a local caching DNS server, because a local server would mean that the DNS settings of the client would be adapted. Because the WX appliance is already in the data stream from the client to the server, it could easily filter the request from the client, send a cached answer and discard the request for the server. The advantage would be that the client won't need DNS reconfiguration.

11 Tradeoffs of inline protocol optimization

The functionality of the WX devices and the appliances from other vendors with the same functionality are based upon 'tinkering' with protocol parameters without knowledge of this from either of the involved nodes. These appliances exist by the grace of bad protocols. This means that the best thing that could happen for a systems administrator, actually would be the worst thing that could happen for the WX platform, i.e. total redesign the protocol to mitigate the problems involved.

11.1 Vendor Optimization

Microsoft has already taken steps to optimize their MAPI protocol for faster networks with larger delay. They have done this by making the blocks larger and enabling compression on a link. Also the CIFS protocol is several times adjusted in the past by Microsoft. We've seen in our tests that the impact of these changes is quite low and because we didn't test caching and compression, the main optimizations performed by Microsoft are not very well portrayed our the results. But when Microsoft would decide that the windowing properties of these these protocols should be reviewed (the primary cause of the bad performance over long fat networks) this could mean a significant performance increase from

within the protocol and even nullify the justification of buying specific hardware for protocol optimization.

11.2 Acceleration and the OSI model

Rules of orthogonality say that network devices can only read in the header of the respective layer that they operate on, and adapt the headers of the lower layers. A switch may adapt the layer 1 header, while a router rewrites the layer 1 and 2 header. The interaction of the WX device, that is actually a layer 2 device, with the client and the server on layer 7 is therefor actually a *Very Bad Thing*. It completely disregards the layering that is made in the networking model. Systems that intervene in layers above their own operating layer were designed to resolve performance or usability issues on the short run. In the long run, they've become a burden for the operation or design of networks. When these technologies become mainstream (e.g. NAT) it could even becomes impossible to phase them out. Therefor, the authors believe the best solution would be to perform the protocol optimization in the client and the server, keeping the layers separated.

12 Conclusion

All the results that were gathered in this report are based upon empirical testing. This conclusion is based upon the results from these tests. All the tests were done with great care. Although there can be errors in the measured results, the authors believe the general trends and conclusions drawn of the results are clear and significant less error prone.

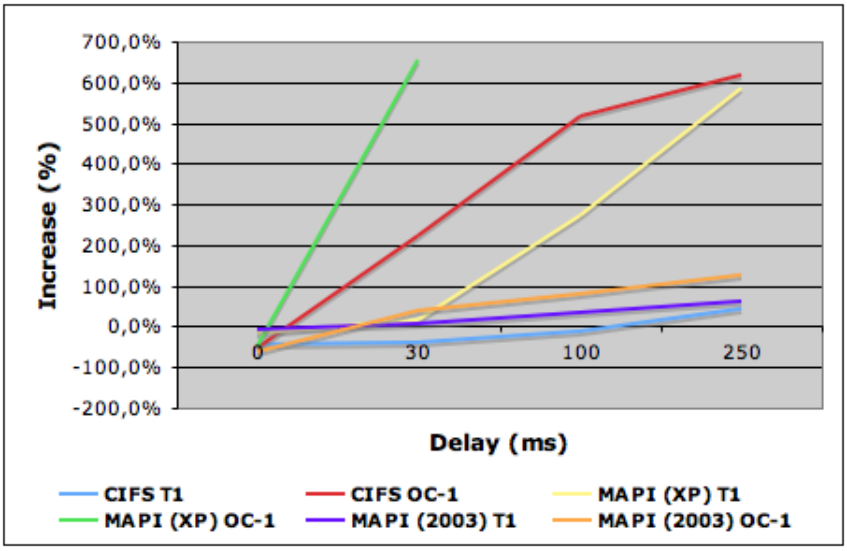

Figure 11: Acceleration of applications

Performance increase CIFS and MAPI

Mainly performance decrease induced by delay can be mitigated with acceleration hardware. In all high-delay situations we've seen performance increase regardless of bandwidth and/or loss. But because the acceleration appliances are placed inline in the data stream, therefor will always be an impact in the throughput and delay. This impact is noticeable on links where the delay is lower than 10 to 15 ms for all bandwidth and loss. Implementation of an acceleration device in situations where these characteristics apply is unwise. The impact of the device will outweigh the performance increase of the acceleration. On links with higher delay, the performance increase of the protocols differs widely, as can be seen in figure 11.

The CIFS protocol, used for file and printer sharing in a Microsoft environment sees a performance increase on low bandwidth links from about 100 ms delay. The increase of the performance ranges from -40% in low delay links upto 40% in high delay links. The small amount of increase in <100 ms delay links is mainly because the *bandwidth*delay* product for this link is small enough for transmission of single CIFS blocks. Links with delay in excess of 100 ms are too large to be completely filled with one block. Therefor, for these links there is a performance increase. On high bandwidth links, the saturation of the *bandwidth*delay* product is reached earlier. The performance impact on these links is much higher, ranging from 200% up to 600% relative to the baseline.

The MAPI protocol is there in two flavors; the older version with Outlook XP that can be accelerated, and the newer version in Outlook 2003 with adaptations in the protocol. The increase of throughput for the older MAPI protocol on low bandwidth links pays off for all delay situations above about 10 ms. The increase of performance is between 40% and 120%, depending on the delay on the link. The high bandwidth links increase the performance of the MAPI protocol even more, from 500% to 5000%, a 50 fold increase of throughput utilization relative to the baseline. The newer MAPI protocol does show an increase in performance, but not as high as the increase of the older protocol. On low bandwidth links the throughput increases 10% to 60% relative to the baseline. On a high bandwidth link, the performance increase is a little higher, namely 40% to 120%. Compared to the acceleration of the older MAPI protocol, the increase is negligible.

Overall, the performance increase of the acceleration hardware can be summed up with the following: with CIFS and MAPI links with high bandwidth and high delay profit the most from protocol optimization. Although low bandwidth connections do show improvements, the best results are gathered with high bandwidth links. This is because of the space that is left in the *bandwidth*delay* product to send more blocks than one, while waiting for an acknowledgment.

Performance increase Forward Error Correction

The results from our error correction tests showed that there is no real performance increase on lossy links. The reason of this is unknown to us, and due to lack of time, we were not able to investigate the source of the problem with these results.

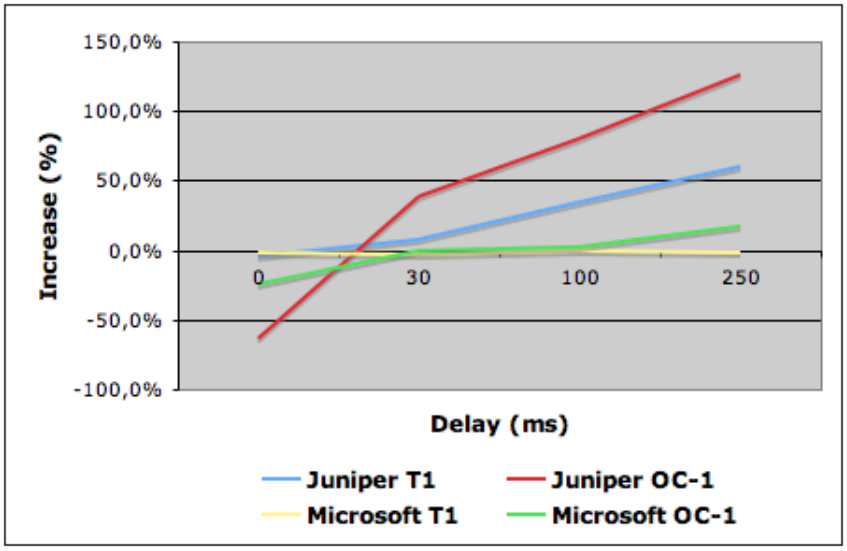

Figure 12: Acceleration of MAPI (2003)

Juniper vs. Microsoft MAPI acceleration

The optimization of the MAPI protocol by Microsoft did not show a very large increase in throughput relative to the baseline, as can be seen in figure 12. On the low bandwidth link, there was no acceleration whatsoever. The high bandwidth link showed an increase of about 20% on links with very high delay. The Juniper acceleration of the MAPI 2003 protocol was in the range of 30% to 120% better than the acceleration by Microsoft, even though the MAPI 2003 protocol is officially not supported by Juniper. We have to remark that Microsoft claims that the performance increase is done by compression as well as bigger blocks. Because of the test setup, compression was excluded from the results. Therefor these results only indicate the performance increase by enlarging the blocks.

Tradeoffs of inline application acceleration

The use of these type of acceleration techniques would certainly be a good choice to mitigate slow throughput on high delay links, but we have to remember that the use of these appliances goes against the layered principle of the OSI model. A layer 2 device that is put inline with the traffic reads and adjusts data in headers on layers where it is not supposed to be. We've seen previously that these kinds of techniques (e.g. NAT) are very well suitable to mitigate performance or scalability issues, but when these techniques become ubiquitous, they obfuscate the workings of a layered network, and severely trouble the work of a network designer. The real solution to bad performance of protocols over a WAN link is to optimize the protocols themselves.

References

[1] **Juniper Networks WX Application Acceleration platforms**
http://www.juniper.net/products/appaccel/wan/wx/

[2] **Microsoft Windows Platform**
http://www.microsoft.com/windows/default.mspx

[3] **Microsoft Exchange Server**
http://www.microsoft.com/exchange/default.mspx

[4] **CIFS: A Common Internet File System**
http://www.microsoft.com/mind/1196/cifs.asp

[5] **MAPI in Outlook/Exchange Development**
http://www.outlookcode.com/d/mapi.htm

[6] **SG FAQ :: What is the Bandwidth * Delay Product?**
http://www.speedguide.net/faq_in_q.php?category=89&qid=185

[7] **RFC 2018 - TCP Selective Acknowledgment Options**
http://www.ietf.org/rfc/rfc2018.txt

[8] **RFC 3390 - Increasing TCP's Initial Window**
http://www.ietf.org/rfc/rfc3390.txt

[9] **RFC 3042 - Enhancing TCP's Loss Recovery Using Limited Transmit**
http://www.ietf.org/rfc/rfc3042.txt

[10] **RFC 3465 - TCP Congestion Control with Appropriate Byte Counting**
http://www.ietf.org/rfc/rfc3465.txt

[11] **Early Retransmit for TCP and SCTP**
http://bgp.potaroo.net/ietf/idref/draft-allman-tcp-early-rexmt/

[12] **On the Prevalence and Evaluation of Recent TCP Enhancements**
http://www.cis.udel.edu/∼iyengar/publications/2004.globecom-draft.ladha.pdf

[13] **Wireshark: A Network Protocol Analyzer**
http://www.wireshark.org/

[14] **Ethereal: A Network Protocol Analyzer**
http://www.ethereal.com/

[15] **Riverbed RiOS Application Acceleration**
http://riverbed.com/technology/app_streamlining/index.php

[16] **Cisco Systems to Acquire Actona Technologies**
http://newsroom.cisco.com/dlls/2004/corp_062904.html

[17] **Juniper Networks to Acquire Peribit Networks [...]**
http://www.juniper.net/company/presscenter/pr/2005/pr-050426a.html

[18] **Juniper WX Application Acceleration platforms - Technologies**
http://www.juniper.net/products/appaccel/wan/wx/

[19] **WX/WXC Operators Guide - Release 5.3, page 201**
http://www.juniper.net/techpubs/hardware/wx/srs/53/wxog_53.pdf

[20] **WX/WXC Operators Guide - Release 5.3, page 202**
http://www.juniper.net/techpubs/hardware/wx/srs/53/wxog_53.pdf

[21] **WX/WXC Operators Guide - Release 5.3, page 206**
http://www.juniper.net/techpubs/hardware/wx/srs/53/wxog_53.pdf

[22] **WX/WXC Operators Guide - Release 5.3, page 207**
http://www.juniper.net/techpubs/hardware/wx/srs/53/wxog_53.pdf

[23] **Modify Remote Procedure Call Compression in Exchange Server 2003**
http://support.microsoft.com/?kbid=825371

[24] **Dummynet**
http://info.iet.unipi.it/∼luigi/ip_dummynet/

[25] **Dummynet: A Simple approach... (1997) - Rizzo, Pisa University**
http://iet.unipi.it/∼luigi/dummynet.ps.gz

[26] **PicoBSD, the Small BSD.**
http://people.freebsd.org/∼picobsd/picobsd.html

[27] **Iperf - The TCP/UDP Bandwidth Measurement Tool**
http://dast.nlanr.net/Projects/Iperf/

[28] **MasterShaper - Easy traffic shaping and QoS with Linux**
http://www.mastershaper.org/

[29] **PragaOrgAr : Dev Praga htb-gen**
http://www.praga.org.ar/wacko/DevPraga/htbgen/

A WAN link simulator

A.1 Selection Criteria

The tool that we want to use to simulate several WAN links has to meet several criteria. For the purpose of this project, we looked at the network characteristics of the most common types of network links that are used with the *application-specific WAN acceleration appliances*. With these links in mind, we set the following criteria.

Bandwidth of the link needs to be configurable in a range from 64 Kbit/sec (typical DS0 link) upto 51.84 Mbits/sec (typical OC-1 link).

Delay of a round-trip-time needs to be configurable in a range from 0ms upto 500ms to simulate transatlantic links.

Loss of packets during transit needs to be configurable in a range from 0,01% upto 0,1%.

The *application-specific WAN acceleration appliances* accelerates traffic by manipulating in the application layer or the transport layer header. Therefor we had to set the following criteria to make sure that the *application-specific WAN acceleration appliances* and the WAN link simulator won't intervene with each others functions.

Application of the simulation must take place at the network layer or lower in the stack.

A.2 Short-list

After a search for tools that can simulate some or several of the criteria stated above, we came up with this shortlist.

Tool	Bandwidth	Delay	Loss	Implementation (based upon)
Dummynet[24]	y	y	y	Layer 3 (ipfw, BSD)
Mastershaper[28]	y	n	n	Layer 3 (iproute2, Linux)
htb-gen[29]	y	n	n	Layer 3 (iproute2, Linux)

Table 19: Simulation tool shortlist

A.3 Selection

For the simulation of the WAN links, from all the tools we had at hand, we chose `dummynet` [24], because `dummynet` is the only tool that complied with al the criteria stated. *'`dummynet` works by intercepting communication of the protocol layer under test and simulating the effects of finite queues, bandwidth limitations and communication delays.'* [25] The author of the article has made an implementation of `dummynet` available under the BSD style license. Dummynet works by turning the available NICs in a system in a bridge, and forcing the packets through a firewall ruleset. The ruleset is optimized for the functionality needed. Bandwidth limitation is performed by shaping the traffic, while

delay limitation is performed by WFQ2+[5] queuing technology. Loss simulation is performed by randomly dropping packets.

`dummynet` comes in an image with the picoBSD [26] operating system, based on a FreeBSD 3.4 kernel. The hardware setup is a Dell system with two Intel Fast Ethernet NICs inline in the link between the two *application-specific WAN acceleration appliances*. Before using the tool, we calibrated the tool in a test environment to see if `dummynet` results match the configuration.

A.4 Calibration

For calibration of the WAN link simulator we created several scenarios in which we could single out the performance penalty induced by the software, and the accuracy of the simulation software.

Tests

The bandwidth test is performed with the *iperf* [27] tool that measures tcp throughput performance between a client and a server. In each scenario there are 10 tests, of which the average result is taken. The delay test is performed by performing 1000 pings, and taking the average round-trip-time. Loss performance is tested with 10000 pings. To measure significant results with the 0,1% loss, 1000 pings is not enough. Additional, we have to take in account that the loss statistics of ping are measured over the round-trip. Because the WAN link simulator applies the loss to both the ping request as the ping reply, we have to check the loss statistics at the remote end. This can be done with a *tcpdump* setup where we count the ping requests that are received.

Baseline

We first performed a baseline test with a regular desktop switch and a baseline test with the WAN link simulator that has no configuration, so we can isolate the performance impact of the WAN link simulator. The baseline tests included a bandwidth test to see what the practical utilization of throughput between the systems is, a delay test to see what the delay there was between the systems and a loss test, to see if the baseline performed optimal. The baseline bandwidth tests were performed with 10 MBit full duplex and 100 MBit full duplex MII [6] settings (i.e. speed restricted through hardware).

test	desktop switch	WAN link simulator
Bandwidth (10M)	9,373 Kbit/sec (93,73%)	9,394 Kbit/sec (93,94%)
Bandwidth (100M)	94382,08 Kbit/sec (94,38%)	94330,88 Kbit/sec (94,33%)
Delay	0,44 msec (rtt)	0,50 msec (rtt)
Loss	0,000%	0,000%

Table 20: Baseline results

[5] Weighted Fair Queuing
[6] Media Independent Interface

Bandwidth accuracy

For the bandwidth accuracy tests, we have performed several tests in which we configured the WAN link simulator to simulate a typical WAN speed. Then we measured the actual performance with *iperf*. In this test, the speed was restricted trough the WAN link simulation software. The accuracy is calculated by dividing the utilization with the average utilization of the hardware restricted tests *((93,94% + 94,33%)/2 = 93,64%)*. Each test is performed ten times.

Speed	WAN link simulator	utilization	accuracy
64K (DS1)	59,56 Kbit/sec	93,06%	99,3%
256K	238,90 Kbit/sec	93,32%	99,6%
1024K	935,80 Kbit/sec	91,38%	97,6%
4096K	3827,71Kbit/sec	93,45%	99,7%
10M	9,491 Mbit/sec	94,91%	101,4%
51,84M (OC-1)	49,112 bit/sec	94,77%	101,2%
100M	93,194 Mbit/sec	93,19%	99,5%

Table 21: Bandwidth results

The bandwidth tests show a fairly accurate result of the bandwidth restriction functionality of the WAN link simulator. All of the results are within acceptable margin of the hardware restricted tests. This accuracy is more than enough for our goal.

Delay accuracy

For the delay accuracy tests, we have performed several tests in which we configured the WAN link simulator to simulate delay. Then we measured the actual delay with *pings*. Each test is performed with ten thousand pings.

Configured delay	Actual delay	accuracy
0ms	0,268ms	—
2ms	1,876ms	93,80%
10ms	9,887ms	98,87%
40ms	39,88ms	99,70%
100ms	99,89ms	99,89%
200ms	199,85ms	99,92%
2000ms	1999,63ms	99,98%

Table 22: Delay results

The delay tests show a fairly accurate result of the delay generation functionality of the WAN link simulator. All of the results are within acceptable margin. This accuracy is enough for our goal.

Loss accuracy

For the loss accuracy tests, we have performed several tests in which we configured the WAN link simulator to simulate loss. Then we measured the actual loss with *pings*. Each test is performed with ten thousand pings.

Configured loss	Actual loss	accuracy
0%	0%	—
0,05%	0,03%	60%
0,1%	0,11%	110%
0,5%	0,48%	96%
1%	0,91%	91%
5%	4,74%	94%

Table 23: Loss results

The loss tests show a fairly accurate result of the loss generation functionality of the WAN link simulator. All of the results are within acceptable margin. This accuracy is enough for our goal.

A.5 Conclusion

The `dummynet` tool matches all the criteria specified. In calibration tests is made clear that the tool performed within reasonable margins of the specified values for bandwidth, delay and loss. Therefor `dummynet` is the right tool to use in our scenario.

B Lists

B.1 List of Figures

List of Figures

B.2 List of Tables

List of Tables

C Research Plan

General Information

Contact

Student:	Marc Smeets	msmeets@os3.nl
Student:	Dirk-Jan van Helmond	dirkjan@os3.nl
UvA General Contact:	Cees de Laat	delaat@science.uva.nl
Juniper General Contact:	Hans Rinsema	hrinsema@juniper.net
Juniper Technical Contact:	Saverio Pangoli	spangoli@juniper.net

Project

Start date:	Tuesday 6 June 2006
End date:	Friday 30 June 2006
Location:	Juniper EMEA JTAC, Schiphol-Rijk

Project Definition

Growing bandwidth demand by corporate users has created a market for application-aware "WAN acceleration platforms", i.e. devices that help make optimal use of all available bandwidth by eliminating protocol inefficiencies. The Juniper WX platform helps improve application performance over WAN links with a number of transport and application-specific optimizations. The goal of this project is to evaluate the performance benefits and tradeoffs encountered when deploying the WX platform over links with different bandwidth, delay, jitter and loss characteristics.

Project Planning

Week 1

- Get familiar with the WX platform from a theoretical and practical point of view
- Create a shortlist of applications that will be used over the 'WAN' link for testing
- Create a shortlist of tools that can simulate network link characteristics e.g. bandwidth, delay, retransmission
- Test the tools for simulating network link characteristics for reliability and accuracy and choose the tools to be used

Week 2 and 3

- Create several realistic network characteristic scenarios. Actual link characteristics as well as theoretical link characteristics are possible

- **Create a WX configuration shortlist with application specific optimization (deliverable)**
- **Write a test plan to execute, similar to *application shortlist * configuration shortlist * network scenarios* (deliverable)**
- Design a lab configuration to be used for testing the scenarios
- **Build the actual lab configuration (deliverable)**
- Execute the test plan.
- Analysis of measured results

Week 4

- **Write-up of the results (deliverable)**

Deliverables

The students are required to deliver the following results:

- A fully functional lab setup that can be used for testing several different scenarios with different link characteristics
- A setup with application servers that can be used over the lab setup
- A test plan and a report with the results from several realistic network scenario tests

Equipment

Juniper will provide a lab setup consisting of the WX-series hardware, a few servers that can be used to simulate realistic application load, and the use of the available JTAC traffic generators and analyzers. In addition to these, the candidates are encouraged to use any of the freely avaliable network analisys and simulation tools.