

Chapter 6. Design and Implementation of the Experiment

6.1. Physical Test Setup

The test set-up included two ATM switches (Olicom CrossFire 9100 and 9200), two PCs (450 MHz Pentium III processors with 128 MB of RAM), and two NICs (RapidFire 6162 ATM 155 PCI) connected by OC-3 multimode optical fiber links. Furthermore, two variable attenuators were inserted between the two switches and the same set of tests was executed for different levels of attenuation. The photo of the actual network set-up is shown in Figure 6.1.



Figure 6.1 Laboratory Setup

6.2. Hardware Configuration

The ATM switches were configured to run LANE, where the 9200 operated as the primary LANE administrator and the 9100 was the secondary. The emulated protocol was Ethernet (IEEE 802.3). This configuration offered a SONET/ATM/LANE-Emulated Ethernet/IP/TCP/Application protocol stack. All hardware was given a separate LANE IP and had the same subnet mask of 255.0.0.0, as shown in Figure 6.2.

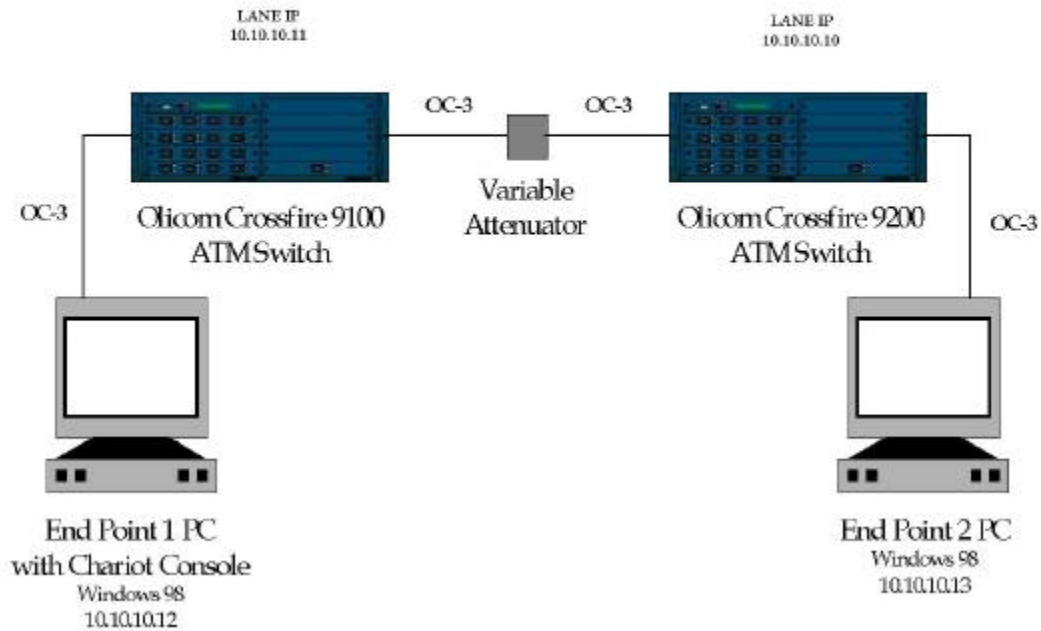


Figure 6.2 Block Diagram of Test Setup in the Laboratory

The ATM NIC's use UBR service category to relay the data traffic across the network. In the case where only the two PCs are connected to the network, the entire OC-3 bandwidth of 155.52 Mbps is available to UBR. This is an

important detail since generally UBR never has all the network bandwidth at its disposal.

Configuration parameters for the PCs are presented in Table 6.1, where E1 designation refers to Endpoint 1, and E2 corresponds to Endpoint 2.

Table 6.1 PC Configuration Parameters

Endpoint 1 Type	Endpoint 1 Value	Endpoint 2 Type	Endpoint 2 Value
E1 Version	3.1	E2 Version	3.1
E1 Build Level	403	E2 Build Level	403
E1 Product Type	Retail	E2 Product Type	Retail
E1 Operating System	Windows 98	E2 Operating System	Windows 98
OS Version (major)	4	OS Version (major)	4
OS Version (minor)	10	OS Version (minor)	10
OS Build Number	1998	OS Build Number	1998
CSD Version		CSD Version	
Memory	130572 (KB)	Memory	130572 (KB)
APPC Default Send Size	32763	APPC Default Send Size	32763
IPX Default Send Size	1391	IPX Default Send Size	1391
SPX Default Send Size	4096	SPX Default Send Size	4096
TCP Default Send Size	4096	TCP Default Send Size	4096
UDP Default Send Size	8183	UDP Default Send Size	8183
WinSock API	Microsoft	WinSock API	Microsoft
WinSock Stack Version	2.2	WinSock Stack Version	2.2
WinSock API Version Used	2.2	WinSock API Version Used	2.2

In order to provide the best environment for obtaining consistent results, the PCs obtained for this project have identical hardware and software features, are configured in the same way, were purchased at the same time, and use the same NIC cards. The table also shows that both PCs are running identical Operating Systems (OS), Windows 98, and all corresponding parameters have the same values on both computers. The most important configuration

parameters to note are the TCP send size of 406 bytes and UDP send size of 8183 bytes.

6.3. Software Configuration

The application layer and network performance tests were done by Ganymede's Chariot software [3]. For this project, four main tests were selected and divided in two groups. Test 1 and Test 2 emulated Bader benchmark or classic transactions, and internet applications, respectively, and both used TCP as the transport layer. Test 3 and Test 4 emulated multimedia data and used UDP as the transport protocol. It is worth noting that multimedia data is generally transported over UDP protocol. The connectionless, "best effort" service of UDP and IP are necessary for effective multicast and streaming applications, which are an integral part of transporting multimedia over the Internet. Since Chariot emulates the "real world" applications, there is no option for running multimedia application scripts over TCP.

Generally, the test parameters were adjusted for achieving the highest possible throughput between the two end points. The four tests were run for a duration of one hour over nine different power levels. This duration was sufficient to obtain statistically valid data, which can be observed through the 95% Confidence Interval and Relative Precision parameters. The following parameters have been observed and compared for different attenuation levels: throughput, transaction rate, and response time for Tests 1 and 2, and throughput, percent bytes lost and number of datagrams lost between the endpoints, for Tests 3 and 4.

6.4. Testing

All test parameters were set at levels consistent with stressing the network. The goal was to cause a large amount of network traffic that would really stress the network hardware and software.

Run for a fixed duration: all tests were run for one hour, and certain script parameters were changed in order to generate a number of timing records large enough for statistical calculations, but small enough to prevent overloading the console with too many timing records.

Report timings using real-time: real-time reporting causes extra network traffic since the timing records flow across the network and are being reported as they are being generated. Batch file reporting waits until the tests are completed and then reports the results.

Regular polling of the endpoints: polling of endpoints causes additional flows outside the pattern of scripts and timing records, thus further stressing the network. In this project polling was done every minute for the duration of the test.

Validation of data upon receipt: data validation was important in order to see if there were any problems with data transferred across the network under the stress conditions. Data validation was especially important considering that the physical link was gradually degraded during this test.

Random SLEEP times: Chariot suggests using uniform distribution of sleep times to emulate many users. In this project the uniform distribution of 0 to 50 ms was used.

Setting the SEND data type to NOCOMPRESS: in addition to simply sending more data, the data that was sent is the toughest data to compress, thereby keeping any network components that use compression busy trying to find patterns in this data.

In addition to these parameters, the tests were designed to always use a new random seed when starting a new test, and the console was instructed to stop the test after the first pair failed.

6.4.1. Test One

Test one consisted of 4 pairs running simultaneously across the network from Endpoint 1 to Endpoint 2 for a sustained overall throughput of around 115 Mbps. The test used two of the benchmark scripts. Two pairs ran FILERCV scripts which emulated a request for a file and getting it back. The other two pairs ran FILESND scripts which emulate sending a file across the network and then getting a confirmation back.

The run options for Test 1 were chosen according to guidelines described at the beginning of Section 6.4 and are presented in Table 6.2. In addition to the duration, reporting type, polling, and data validation it shows that the test was set to stop if there was an initialization failure between any of the four pairs, regardless of when it happened during the test. It also shows that a new seed for random variables was used for every run.

Table 6.2 Test 1 Configuration Parameters

End type	Run for a fixed duration
Duration	01:00:00
Reporting type	Real-time
Automatically poll endpoints	Yes
Polling interval (minutes)	1
Stop run upon initialization failure	Yes
Connect timeout during test (minutes)	0
Stop test after this many running pairs fail	1
Validate data upon receipt	Yes
Use a new seed for random variables on every run	Yes

The details specifying the pairs and their respective endpoint's addresses, application scripts, and the network protocol used for the test setup, are shown in Table 6.3. It should be mentioned that Chariot uses only TCP to communicate with Endpoint 1, regardless of which network protocol is being used by the applications scripts between the endpoints.

Table 6.3 Test 1 Setup from Endpoint 1 to Endpoint 2

Group/ Pair	Endpoint 1	Endpoint 2	Network Protocol	Service Quality	Script Name
All Pairs					
Pair 1	10.10.10.12	10.10.10.13	TCP		filercvltest1.scr
Pair 2	10.10.10.12	10.10.10.13	TCP		filercvltest1.scr
Pair 3	10.10.10.12	10.10.10.13	TCP		filesndltest1.scr
Pair 4	10.10.10.12	10.10.10.13	TCP		filesndltest1.scr

6.4.1.1. File Receive, Long Connection Script

As can be seen in Table 6.3, the first two pairs ran the `filercvltest1.scr` script, which performs the same function as the `FILERCVL.SCR` but with a few changed parameters in order to provide the higher throughput. The changed script was named `filercvltest1` and it is presented in Table 6.4.

Table 6.4 `filercvltest1.scr` Script

ENDPOINT 1	ENDPOINT 2
<pre> SLEEP initial_delay=u[0,50] CONNECT_INITIATE port_number=AUTO LOOP Number of timing records=100 START TIMER LOOP transactions_per_record=1 SEND size_of_record_to_send=100 size_of_record_to_send=100 send_datatype=NOCOMPRESS send_data_rate=UNLIMITED RECEIVE file_size=10000000 receive_buffer_size=DEFAULT INCREMENT_TRANSACTION END_LOOP END_TIMER SLEEP transaction_delay=0 END_LOOP DISCONNECT </pre>	<pre> CONNECT_ACCEPT port_number=AUTO LOOP Number of timing records=100 LOOP Transactions per record=1 RECEIVE size_of_record_to_send=100 size_of_record_to_send=100 SEND file_size=10000000 send_buffer_size=DEFAULT send_datatype=NOCOMPRESS send_data_rate=UNLIMITED END_LOOP END_LOOP DISCONNECT </pre>

Following the guidelines, and in order to maximize the throughput, a few script variables were changed: Initial delay was set to have uniform distribution values between 0 and 50 ms, and the file size was increased. The detailed description of the script parameters, their values, and descriptions are given in Table 6.5.

Table 6.5 Description of Variables for filercvlttest1.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	100	How many timing records to generate
transactions_per_record	1	Transactions per timing record
size_of_record_to_send	100	Amount of data to be sent
file_size	10000000	How many bytes in the transferred file
send_buffer_size	DEFAULT	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
transaction_delay	0	Milliseconds to pause
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.1.2. File Send, Long Connection Script

The second two pairs ran the filesndltest1.scr script, which performs the same function as the FILESNDL.SCR but with a few changed parameters in order to provide the higher throughput. The changed script was named *filesndltest1* and it is presented in Table 6.6, and the individual description of parameters used in the script can be found in Table 6.7.

Table 6.6 filesndtest1.scr Script

ENDPOINT 1	ENDPOINT 2
SLEEP initial_delay=u[0,50] CONNECT_INITIATE port_number=AUTO LOOP number_of_timing_records=10 START_TIMER LOOP transactions_per_record=1 SEND file_size=10000000 send_buffer_size=DEFAULT send_datatype=NOCOMPRESS send_data_rate=UNLIMITED CONFIRM_REQUEST INCREMENT_TRANSACTION END_LOOP END_TIMER SLEEP transaction_delay=0 END_LOOP DISCONNECT	CONNECT_ACCEPT port_number=AUTO LOOP number_of_timing_records=10 LOOP transactions_per_record=1 RECEIVE file_size=10000000 receive_buffer_size=DEFAULT CONFIRM_ACKNOWLEDGE END_LOOP END_LOOP DISCONNECT

Table 6.7 Description of Variables for filesndtest1.scr Script

VARIABLE NAME	VALUE	DESCRIPTION
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	10	How many timing records to generate
transactions_per_record	1	Transactions per timing record
file_size	10000000	How many bytes in the transferred file
send_buffer_size	DEFAULT	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
transaction_delay	0	Milliseconds to pause
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.2 Test Two

Test 2 consisted of 3 pairs running simultaneously across the network from Endpoint 1 to Endpoint 2 for a sustained overall throughput of around 108 Mbps. The test used three of the Internet scripts, which were slightly modified for this project: FTPPUT, HTTPGIF and HTTPTEXT.

Since both Test 1 and Test 2 run over TCP, the run options for this test template are the same as presented in Table 6.2. The details for the test setup from Endpoint 1 to Endpoint 2, such as the network protocol and the script name for the group of pairs 1, 2, and 3, are presented in Table 6.8.

Table 6.8 Test Setup from Endpoint 1 to Endpoint 2

Group/ Pair	Endpoint 1	Endpoint 2	Network Protocol	Service Quality	Script Name
All Pairs					
Pair 1	10.10.10.12	10.10.10.13	TCP		<i>ftpputtest2.scr</i>
Pair 2	10.10.10.12	10.10.10.13	TCP		<i>httpgifttest2.scr</i>
Pair 3	10.10.10.12	10.10.10.13	TCP		<i>httptexttest2.scr</i>

6.4.2.1. FTP Put Script

The first pair ran an FTPPUT script, which emulated the TCP/IP's FTP application, and where Endpoint 1 is placing a file onto Endpoint 2. The modified FTPPUT script, *ftpputtest2*, is shown in Table 6.9, and detailed descriptions of the changed variables are presented in Table 6.10.

Table 6.9 ftpptest2.scr Script

Endpoint 1	Endpoint 2
<p>SLEEP initial_delay=u[0,50] LOOP number_of_repetitions=1 CONNECT_INITIATE port_number=auto RECEIVE login_size=15 control_buffer_size=DEFAULT</p> <p>SLEEP user_delay=0 SEND login_size=15 control_buffer_size=DEFAULT control_datatype=trans.cmp send_data_rate=UNLIMITED RECEIVE login_size=15 control_buffer_size=DEFAULT</p> <p>SLEEP user_delay=0 SEND login_size=15 control_buffer_size=DEFAULT control_datatype=trans.cmp send_data_rate=UNLIMITED RECEIVE 200 control_buffer_size=DEFAULT</p> <p>DISCONNECT LOOP number_of_timing_records=100 START TIMER LOOP transactions_per_record=1 CONNECT_INITIATE port_number=AUTO SEND file_control_size=30 control_buffer_size=30 control_datatype=trans.cmp</p>	<p>LOOP number_of_repetitions=1 CONNECT_ACCEPT port_number=auto SEND login_size=15 control_buffer_size=DEFAULT control_datatype=trans.cmp send_data_rate=UNLIMITED</p> <p>RECEIVE login_size=15 control_buffer_size=DEFAULT</p> <p>SEND login_size=15 control_buffer_size=DEFAULT control_datatype=trans.cmp send_data_rate=UNLIMITED</p> <p>RECEIVE login_size=15 control_buffer_size=DEFAULT</p> <p>SEND 200 200 control_datatype=trans.cmp send_data_rate=UNLIMITED</p> <p>DISCONNECT LOOP number_of_timing_records=100</p> <p>LOOP transactions_per_record=1 CONNECT_ACCEPT port_number=AUTO RECEIVE file_control_size=30 control_buffer_size=30</p>

<pre> send_data_rate=UNLIMITED RECEIVE file_control_size=30 control_buffer_size=DEFAULT SEND file_control_size=30 control_buffer_size=DEFAULT control_datatype=trans.cmp send_data_rate=UNLIMITED RECEIVE file_control_size=30 control_buffer_size=DEFAULT SLEEP delay_before_responding=0 SEND size_of_record_to_send=1000000 send_buffer_size_4096 send_datatype=NOCOMPRESS Send_data_rate=UNLIMITED RECEIVE file_control_size=30 control_buffer_size=DEFAULT DISCONNECT INCREMENT_TRANSACTION END_LOOP END_TIMER SLEEP transaction_delay=0 END_LOOP CONNECT_INITIATE port_number=AUTO SEND 6 6 control_datatype=trans.cmp send_data_rate=UNLIMITED RECEIVE 25 control_buffer_size=DEFAULT DISCONNECT END_LOOP </pre>	<pre> SEND file_control_size=30 control_buffer_size=DEFAULT control_datatype=trans.cmp send_data_rate=UNLIMITED RECEIVE file_control_size=30 control_buffer_size=DEFAULT SEND file_control_size=30 control_buffer_size=DEFAULT control_datatype=trans.cmp send_data_rate=UNLIMITED RECEIVE file_control_size=30 control_buffer_size=DEFAULT SEND file_control_size=30 control_buffer_size=DEFAULT control_datatype=trans.cmp send_data_rate=UNLIMITED DISCONNECT END_LOOP END_LOOP CONNECT_ACCEPT port_number=AUTO RECEIVE 6 control_buffer_size=DEFAULT SEND 25 25 control_datatype=trans.cmp send_data_rate=UNLIMITED DISCONNECT END_LOOP </pre>
---	--

Table 6.10 Description of Variables for ftpptest2.scr Script

VARIABLE NAME	VALUE	DESCRIPTION
initial_delay	u[0, 50]	Pause before the first transaction
number_of_repetitions	1	How many times to repeat the script
number_of_timing_records	100	How many timing records to generate
transactions_per_record	1	Transactions per timing record
size_of_record_to_send	10000000	Amount of data to be sent
user_delay	0	Pause before answering
transaction_delay	0	Milliseconds to pause
delay_before_responding	0	Milliseconds to wait before responding
file_control_size	30	How many bytes are in the control flows
login_size	15	How many bytes are in the login flows
control_buffer_size	DEFAULT	Buffer size for control flows
send_buffer_size	4096	How many bytes of data in each SEND
receive_buffer_size	4096	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
control_datatype	trans.cmp	What type of control data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.2.2. HTTP GIF Script

The second pair ran an HTTPGIF script, which emulates a transfer of an image in a GIF format from an HTTP server. The modified HTTPGIF script, httpgifest2, is shown in Table 6.11, and detailed descriptions of the changed variables are presented in Table 6.12.

Table 6.11 *httptest2.scr* Script

Endpoint 1	Endpoint 2
<p>SLEEP initial_delay=0</p> <p>LOOP number_of_timing_records=50 START_TIMER LOOP transactions_per_record=1 CONNECT_INITIATE port_number=AUTO SEND size_of_record_to_send=300 size_of_record_to_send=300 control_datatype=trans.cmp send_data_rate=UNLIMITED</p> <p>RECEIVE file_size=10000000 receive_buffer_size=DEFAULT</p> <p>DISCONNECT INCREMENT_TRANSACTION END_LOOP END_TIMER SLEEP transaction_delay=0 END_LOOP</p>	<p>LOOP number_of_timing_records=50</p> <p>LOOP transactions_per_record=1 CONNECT_ACCEPT port_number=AUTO RECEIVE size_of_record_to_send=300 size_of_record_to_send=300</p> <p>SLEEP delay_before_responding=0 SEND file_size=10000000 send_buffer_size=DEFAULT send_datatype=lena.cmp send_data_rate=UNLIMITED DISCONNECT</p> <p>END_LOOP</p> <p>END_LOOP</p>

Table 6.12 Description of Variables for *httpgifttest2.scr* Script

Variable Name	Value	Description
initial_delay	0	Pause before the first transaction
number_of_timing_records	50	How many timing records to generate
transactions_per_record	1	Transactions per timing record
size_of_record_to_send	300	Amount of data to be sent
file_size	10000000	How many bytes in the transferred file
send_buffer_size	DEFAULT	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
delay_before_responding	0	Milliseconds to wait before responding
transaction_delay	0	Milliseconds to pause
send_datatype	lena.cmp	What type of data to send
control_datatype	trans.cmp	What type of control data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endp

6.4.2.3. HTTP Text Script

The third one ran an HTTPTEXT script, which emulates a transfer of text files from an HTTP server. The modified HTTPTEXT script, httpstest2, is shown in Table 6.13, and detailed descriptions of the changed variables are presented in Table 6.14.

Table 6.13 httpstest2.scr Script

Endpoint 1	Endpoint 2
<pre> SLEEP initial_delay=u[0,50] LOOP number_of_timing_records=50 START_TIMER LOOP transactions_per_record=1 CONNECT_INITIATE port_number=AUTO SEND size_of_record_to_send=300 size_of_record_to_send=300 control_datatype=trans.cmp send_data_rate=UNLIMITED RECEIVE file_size=10000000 receive_buffer_size=DEFAULT DISCONNECT INCREMENT_TRANSACTION END_LOOP END_TIMER SLEEP transaction_delay=0 END_LOOP </pre>	<pre> LOOP number_of_timing_records=50 LOOP transactions_per_record=1 CONNECT_ACCEPT port_number=AUTO RECEIVE size_of_record_to_send=300 size_of_record_to_send=300 SLEEP delay_before_responding=0 SEND file_size=10000000 send_buffer_size=DEFAULT send_datatype=news.cmp send_data_rate=UNLIMITED DISCONNECT END_LOOP END_LOOP </pre>

Table 6.14 Description of Variables for *httptexttest2.scr* Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	50	How many timing records to generate
transactions_per_record	1	Transactions per timing record
size_of_record_to_send	300	Amount of data to be sent
file_size	10000000	How many bytes in the transferred file
send_buffer_size	DEFAULT	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
delay_before_responding	0	Milliseconds to wait before responding
transaction_delay	0	Milliseconds to pause
send_datatype	news.cmp	What type of data to send
control_datatype	trans.cmp	What type of control data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3. Test Three

Test three consisted of 10 pairs running simultaneously across the network from Endpoint 1 to Endpoint 2 for a sustained overall throughput of around 14 Mbps. The test used six multimedia scripts: There were 5 pairs emulating Microsoft's NetMeeting Audio, Microsoft's NetMeeting Video, Microsoft's NetShow application, RealAudio by RealNetworks, and RealMedia by RealNetworks. The other five pairs all ran Voice-Over-IP (VOIP) scripts. All of the applications ran over UDP.

Test 3, with ten pairs running simultaneously, was extremely CPU intensive and was unable to generate significant overall throughput between the endpoints. However, it did show that the attenuation changes do not have much effect on network performance if the network is significantly underutilized.

The configuration parameters for Test 3 were also chosen according to guidelines described at the beginning of Section 6.4, and are presented in Table

6.15. In addition to the options such as duration, reporting type, polling, data validation, stop test after initialization failure, and new seed for random variables -- which are the same run options on all tests -- scripts running over UDP must specify the datagram window size, datagram retransmission time-out, datagram number of retransmissions before aborting the test, receive time-out, and time-to-live. Time-to-live was able to be set at one hop due to the fact that all network elements used in this project were on the same subnet.

Table 6.15 Test 3 Configuration Parameters

End type	Run for a fixed duration
Duration	01:00:00
Reporting type	Real-time
Automatically poll endpoints	Yes
Polling interval (minutes)	1
Stop run upon initialization failure	Yes
Connect timeout during test (minutes)	0
Stop test after this many running pairs fail	1
Validate data upon receipt	Yes
Use a new seed for random variables on every run	Yes
Datagram window size (bytes)	1500000
Datagram retransmission timeout (milliseconds)	200
Datagram number of retransmits before aborting	50
Receive Timeout (milliseconds)	10000
Time To Live (Hops)	1

The details specifying the pairs and their respective endpoints addresses, applications scripts that were used, and network protocol used for the test are shown in Table 6.16.

Table 6.16 Test 3 Setup from Endpoint 1 to Endpoint 2

Group/ Pair	Endpoint 1	Endpoint 2	Network Protocol	Service Quality	Script Name
All Pairs					
Pair 1	10.10.10.12	10.10.10.13	UDP		netmtgatest3.scr
Pair 2	10.10.10.12	10.10.10.13	UDP		netmtgvttest3.scr
Pair 3	10.10.10.12	10.10.10.13	UDP		netshowutest3.scr
Pair 4	10.10.10.12	10.10.10.13	UDP		realaudtest3.scr
Pair 5	10.10.10.12	10.10.10.13	UDP		realmedtest3.scr
Pair 6	10.10.10.12	10.10.10.13	UDP		voipstest3.scr
Pair 7	10.10.10.12	10.10.10.13	UDP		voipstest3.scr
Pair 8	10.10.10.12	10.10.10.13	UDP		voipstest3.scr
Pair 9	10.10.10.12	10.10.10.13	UDP		voipstest3.scr
Pair 10	10.10.10.12	10.10.10.13	UDP		voipstest3.scr

6.4.3.1. NetMeeting Audio Stream Script

The first pair ran a NETMTGA script, which emulates sending an audio stream, using Microsoft’s NetMeeting Version 2.1, over a 100Mbps Ethernet LAN. The modified script, netmtgatest3, is shown in Table 6.17.

Table 6.17 netmtgatest3.scr Script

Endpoint 1	Endpoint 2
SLEEP initial_delay=u[0,50] CONNECT_INITIATE port_number=AUTO LOOP number_of_timing_records=1 SEND file_size=1760000 send_buffer_size=44 send_datatype=NOCOMPRESS send_data_rate=UNLIMITED END_LOOP DISCONNECT	CONNECT_ACCEPT port_number=AUTO LOOP number_of_timing_records=1 START_TIMER RECEIVE file_size=1760000 receive_buffer_size=DEFAULT END_TIMER END_LOOP DISCONNECT

Detailed descriptions of the variables used in this script and their values are presented in Table 6.18.

Table 6.18 Description of Variables of netmtgatest3.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	1760000	How many bytes in the transferred file
send_buffer_size	44	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.2. NetMeeting Video Stream Script

The second pair ran a NETMTGV script, which emulates sending a video stream, using Microsoft's NetMeeting Version 2.1, over a 100Mbps Ethernet LAN. The modified NETMTGV script, netmtgvtest3, is shown in Table 6.19.

Table 6.19 netmtgvtest3.scr Script

ENDPOINT 1	ENDPOINT 2
SLEEP initial_delay=u[0,50] CONNECT_INITIATE port_number=AUTO LOOP number_of_timing_records=1 SEND file_size=20880000 send_buffer_size=522 send_datatype=NOCOMPRESS send_data_rate=UNLIMITED END_LOOP DISCONNECT	CONNECT_ACCEPT port_number=AUTO LOOP LOOP number_of_timing_records=1 START_TIMER RECEIVE file_size=20880000 receive_buffer_size=DEFAULT END_TIMER END_LOOP DISCONNECT

Descriptions of the variables used in the script are presented in Table 6.20.

Table 6.20 Description of Variables for netmtgvttest3.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	20880000	How many bytes in the transferred file
send_buffer_size	522	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.3. NetShow Script

The third pair ran a NETSHOW script, which emulated Microsoft's NetShow application. The modified NETSHOW script, netshowtest3, is presented in Table 6.20, and detailed descriptions of the variables used in the script are presented in Table 6.21.

Table 6.21 netshowutest.scr3 Script

Endpoint 1	Endpoint 2
SLEEP initial_delay=u[0,50] CONNECT_INITIATE port_number=AUTO LOOP number_of_timing_records=2 SEND file_size=10520000 send_buffer_size=526 send_datatype=NOCOMPRESS send_data_rate=UNLIMITED END_LOOP DISCONNECT	CONNECT_ACCEPT port_number=AUTO LOOP number_of_timing_records=2 START_TIMER RECEIVE file_size=10520000 receive_buffer_size=DEFAULT END_TIMER END_LOOP DISCONNECT

Table 6.22 Description of Variables for netshowutest3.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	2	How many timing records to generate
file_size	10520000	How many bytes in the transferred file
send_buffer_size	526	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.4. RealAudio CD Streaming Script

The fourth pair ran a REALAUD script, which emulates the Real Audio application by RealNetworks, such as CD streaming. The modified REALAUD script, realaudtest3, is presented in Table 6.23, and detailed descriptions of the variables used in the script are presented in Table 6.24.

Table 6.23 realaudtest3.scr Script

Endpoint 1	Endpoint 2
SLEEP initial_delay=u[0,50] CONNECT_INITIATE port_number=AUTO LOOP number_of_timing_records=1 SEND file_size=14040000 send_buffer_size=351 send_datatype=NOCOMPRESS send_data_rate=UNLIMITED END_LOOP DISCONNECT	CONNECT_ACCEPT port_number=AUTO LOOP number_of_timing_records=1 START_TIMER RECEIVE file_size=14040000 receive_buffer_size=DEFAULT END_TIMER END_LOOP DISCONNECT

Table 6.24 Description of variables for realaudtest3.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	14040000	How many bytes in the transferred file
send_buffer_size	351	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.5. RealMedia Stream Smart Script

The fourth pair ran a REALMED script, which emulates the Real Audio application by RealNetworks, such as their “smart” audio-video stream. The modified REALMED script, realmedtest3, is presented in Table 6.25, and detailed descriptions of the variables used in the script are presented in Table 6.26.

Table 6.25 realmedtest3.scr Script

Endpoint 1	Endpoint 2
SLEEP initial_delay=u[0,50] CONNECT_INITIATE port_number=AUTO LOOP number_of_timing_records=1 SEND file_size=17240000 send_buffer_size=431 send_datatype=NOCOMPRESS send_data_rate=UNLIMITED END_LOOP DISCONNECT	CONNECT_ACCEPT port_number=AUTO LOOP number_of_timing_records=1 START_TIMER RECEIVE file_size=17240000 receive_buffer_size=DEFAULT END_TIMER END_LOOP DISCONNECT

Table 6.26 Description of Variables for realmedtest3.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	17240000	How many bytes in the transferred file
send_buffer_size	431	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.6. Voice-Over-IP Script

VOIPS script, which emulates voice-over-IP application, simultaneously ran over pairs 6, 7, 8, 9 and 10. This script is presented in Table 6.27, and descriptions of the variables used in the script are given in Table 6.28.

Table 6.27 voipstest3.scr Script

Endpoint 1	Endpoint 2
SLEEP initial_delay=0 CONNECT_INITIATE port_number=AUTO LOOP number_of_timing_records=1 SEND file_size=456000 send_buffer_size=40 send_datatype=NOCOMPRESS send_data_rate=UNLIMITED END_LOOP DISCONNECT	CONNECT_ACCEPT port_number=AUTO LOOP number_of_timing_records=1 START_TIMER RECEIVE file_size=456000 receive_buffer_size=DEFAULT END_TIMER END_LOOP DISCONNECT

Table 6.28 Descriptions of Variables for voipstest3.scr Script

Variable Name	Value	Description
initial_delay	0	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	456000	How many bytes in the transferred file
send_buffer_size	40	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.4. Test Four

Test 4 also consisted of 4 pairs running simultaneously across the network from Endpoint 1 to Endpoint 2 for a sustained overall throughput of around 120 Mbps. Two pairs were running two MPEG video applications. The other two pairs were running two MPEG audio scripts. Test 4 applications ran over UDP, which exhibited the most sensitivity to soft failures, and thus became the most significant test for this project. Since this test used UDP as the network protocol, as was the case in Test 3, the configuration parameters are the same as those in Table 6.15. The details about the Test 4 setup, including the application scripts used, are presented in Table 6.29.

Table 6.29 Test 4 Setup from Endpoint 1 to Endpoint 2

Group/ Pair	Endpoint 1	Endpoint 2	Network Protocol	Service Quality	Script Name
All Pairs					
Pair 1	10.10.10.12	10.10.10.13	UDP		iptvvstress.scr
Pair 2	10.10.10.12	10.10.10.13	UDP		iptvvstress.scr
Pair 3	10.10.10.12	10.10.10.13	UDP		iptvastress.scr
Pair 4	10.10.10.12	10.10.10.13	UDP		iptvastress.scr

6.4.4.1. IP/TV MPEG Video Stream Script

Pairs 1 and 2 ran IPTVV script, which emulates Cisco System's application for running an MPEG video stream over IP. It is presented in Table 6.30, and descriptions of the variables used are given in Table 6.31.

Table 6.30 iptvvstress.scr Script

Endpoint 1	Endpoint 2
SLEEP initial_delay=u[0,50] CONNECT_INITIATE port_number=AUTO LOOP number_of_timing_records=1 SEND file_size=7300000 send_buffer_size=18250 send_datatype=NOCOMPRESS send_data_rate=UNLIMITED END_LOOP DISCONNECT	CONNECT_ACCEPT port_number=AUTO LOOP number_of_timing_records=1 START_TIMER RECEIVE file_size=7300000 receive_buffer_size=DEFAULT END_TIMER END_LOOP DISCONNECT

Table 6.31 Description of Variables for iptvvstress.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	7300000	How many bytes in the transferred file
send_buffer_size	18250	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.4.2. IP/TV Audio Stream Script

Pairs 3 and 4 ran IPTVA script, which emulates Cisco System's application for running an MPEG audio stream over IP. The script used is presented in Table 6.32, and descriptions of the variables used are given in Table 6.33.

Table 6.32 *iptvastress.scr* Script

Endpoint 1	Endpoint 2
SLEEP initial_delay=u[0,50] CONNECT_INITIATE port_number=AUTO LOOP number_of_timing_records=1 SEND file_size=5112000 send_buffer_size=12780 send_datatype=NOCOMPRESS send_data_rate=UNLIMITED END_LOOP DISCONNECT	CONNECT_ACCEPT port_number=AUTO LOOP number_of_timing_records=1 START_TIMER RECEIVE file_size=5112000 receive_buffer_size=DEFAULT END_TIMER END_LOOP DISCONNECT

Table 6.33 *Description of Variables for iptvastress.scr* Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	5112000	How many bytes in the transferred file
send_buffer_size	12780	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints