GridFTP Traffic Modelling

BUPT, ISCAS – November 2008

1. Background

The Globus Toolkit [1] is an open source software toolkit used for building Grid applications and the GridFTP tool is one of the most important components provided by Globus for moving large amounts of data in bulk. GridFTP is based on FTP, the highly-popular Internet file transfer protocol. Given the characteristics of Grid traffic - often a mixture of short, sporadic service calls and bulk data transfers - a GridFTP simulation scenario differs from other traffic models and is therefore important for testing Grid-specific network mechanisms.

Most of the existing literature for GridFTP focuses on analyzing the transmission speed, security, robustness and so on in the high-speed Grid [2], but we believe that research on GridFTP traffic characteristics is also very important, especially for the modelling of Grid traffic. We have analyzed the GridFTP testbed traffic measurements and developed a traffic simulator using ns2 to model GridFTP traffic characteristics. The following sections will detail our analysis and modelling efforts.

2. Modelling

Figure 1 shows our model of a GridFTP testbed which includes five nodes installed with the Globus Toolkit. Three nodes are located in a LAN at BUPT, and the other two are at ISCAS. In our model, each node plays an equal role in the system so traffic captured from any node can be used for modelling.

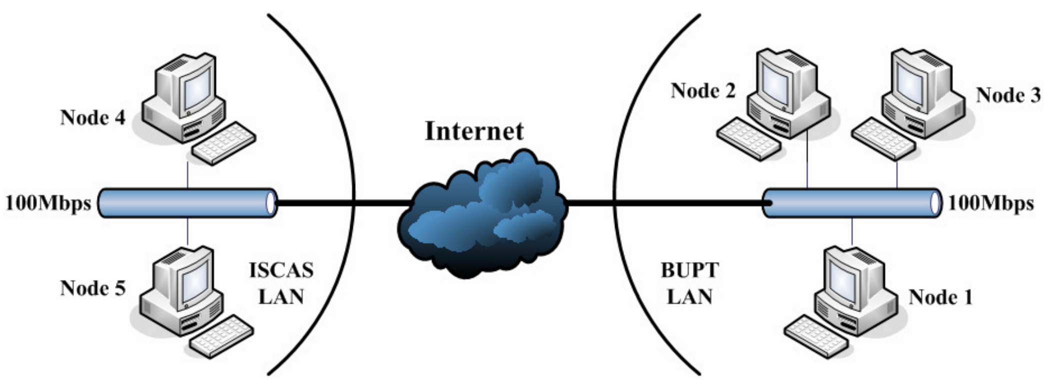


Figure 1 - GridFTP traffic model

The GridFTP modelling process can be described as follows:

- Transfer data between two fixed nodes with GridFTP and FTP respectively, and capturing all the traffic produced. From the captured data, we then analyze the similarity of GridFTP and FTP traffic, and explore the possibility of simulating GridFTP using FTP.
- Code a basic GridFTP connection with ns2 FTP flows.

- Set the GridFTP testbed topology, and retrieve real GridFTP traffic.
- Compute the input parameters of the GridFTP simulator according to the real traffic and gather the simulation results.
- Validate the correctness of the simulator by comparing real traffic data with the simulation results.

3. Metrics in the GridFTP Traffic Model

A summary of the GridFTP traffic modelling parameters are as follows:

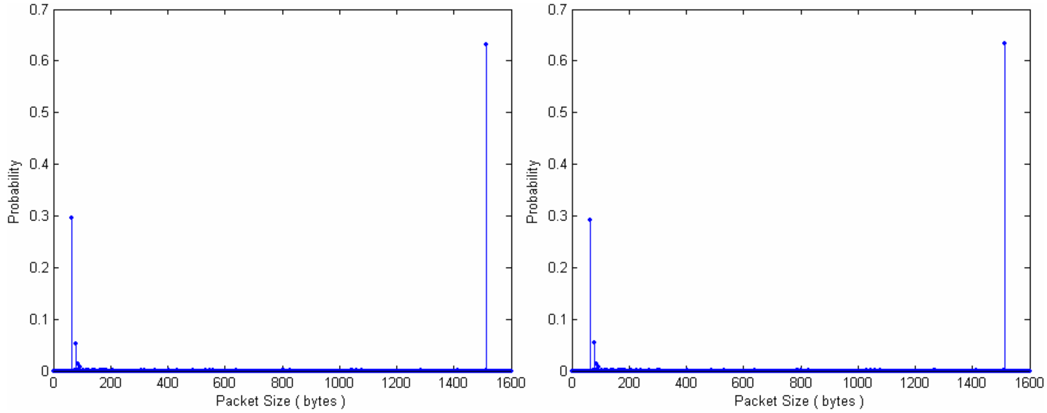
- Network location: BUPT LAN and ISCAS LAN in Beijing connected via the Internet.
- Testbed bandwidth: Bandwidth inside the LAN is 100Mbps, while typical Internet bandwidth is about 700Kbps-1Mbps.
- Transport protocol: TCP
- Measurement and processing tools:
    - TcpDump (version 3.9.8)
    - Matlab (version 6.5)
    - Gawk
    - EasyFit
- Packet size (include header)
    - Target: data packet
    - Behaviour: packet size for each transfer
    - Level of observation: packet-level
    - Observation duration: the whole transmission period
    - Adopted probability distribution: to be released
- Throughput
    - Target: one node in BUPT LAN
    - Behaviour: total bytes per time unit seen from the monitoring node
    - Level of observation: packet-level
    - Observation duration: the whole transmission period
    - Adopted probability distribution: to be released

4. GridFTP Testbed Data Analysis

Seen from the application layer, GridFTP is a secure and efficient data transfer protocol for the Grid environment, and extends the FTP protocol greatly. However, we have found that they are both very similar in terms of traffic features at the network layer. To gather the initial traffic data, we first transfer a file of 200MB using FTP then GridFTP inside a LAN and compare their traffic in terms of packet size and throughput distribution. Then we do the same work over the WAN.

4.1 Packet Size Distribution

Figure 2 (a) shows the packet size distribution of transferring the 200MB file in the LAN with FTP, and (b) gives the distribution of transferring the same file in the LAN with GridFTP. Here we use the ‘extended block’ GridFTP transfer mode, with 3 parallel streams (writing as ‘-p 3’).



(a) FTP (b) GridFTP (-p 3)

Figure 2 - Packet size distribution graph in LAN

The specific probability values of different packet sizes are given in Table 1. From the figures and tables, we see that the packet size distribution using FTP and GridFTP within a LAN are approximately equal.

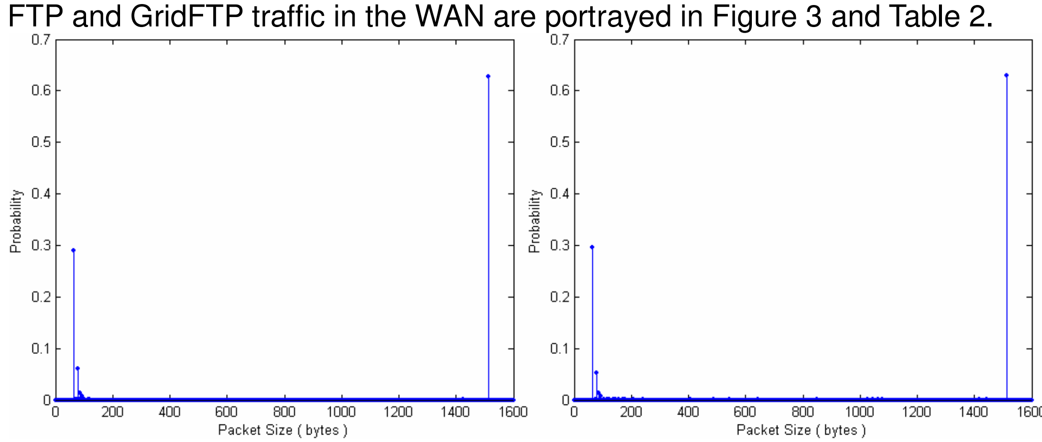
Table 1 (a) FTP packet size probability values in LAN

Packet Size (bytes)	1514	66	78	86	94	others
Probability	0.6306	0.2959	0.0517	0.0137	0.0075	0.0006

Table 1 (b) GridFTP (-p 3) packet size probability values in LAN

Packet Size (bytes)	1514	66	78	86	94	others
Probability	0.6329	0.2916	0.0533	0.0142	0.0076	0.0006

The corresponding packet size distribution graphs and probability values for FTP and GridFTP traffic in the WAN are portrayed in Figure 3 and Table 2.



(a) FTP (b) GridFTP (-p 3)

Figure 3 - Packet size distribution in WAN

Table 2 (a) FTP packet size probability values in WAN

Packet Size (bytes)	1514	66	78	86	94	others
Probability	0.6282	0.2898	0.0598	0.0141	0.0080	0.0001

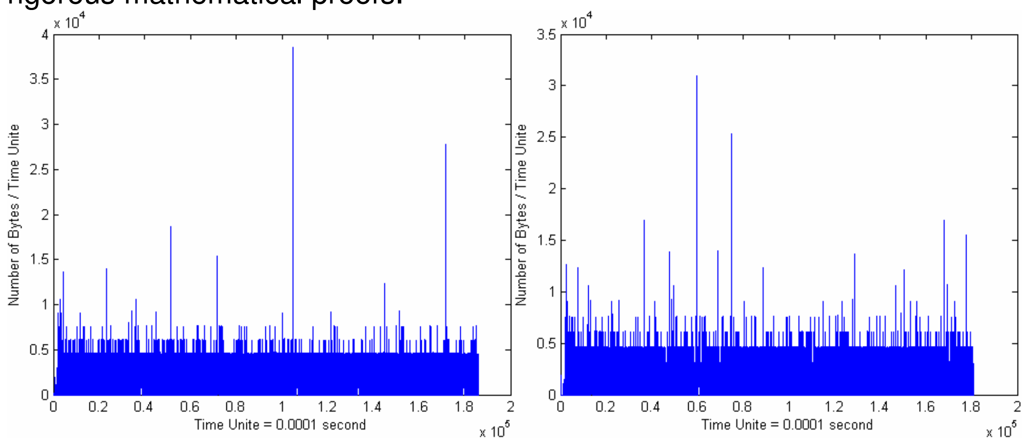
Table 2 (b) GridFTP (-p 3) packet size probability values in WAN

Packet Size (bytes)	1514	66	78	86	94	others
Probability	0.6298	0.2966	0.0522	0.0131	0.0081	0.0002

By comparing the packet size distribution graphs and probability values of the FTP and GridFTP traffic, we can reach the conclusion that they both have similar packet size characteristics.

4.2 Throughput Distribution

Figure 4 shows the FTP and GridFTP throughput of transferring a 200MB file in the LAN. In the graphs, the x axis represents the time (0.0001 second units), and the y axis represents the number of bytes per time unit. From initial impressions, we see that they look very similar but, of course, we need more rigorous mathematical proofs.



(a) FTP (b) GridFTP (-p 3)

Figure 4 - Throughput in LAN

Next, we fit the distribution of the FTP and GridFTP LAN traffic throughput to some common distribution such as normal, logistic, log-logistic, generalized pareto distribution, etc. The Kolmogorov-Smirnov test can then be used to measure the goodness-of-fit. These results show that the 'generalized pareto' is the best fit of all the potential distributions. The K-S values and fitting parameters are shown in Table 3 and the fitting graphs are illustrated in Figure 5. Of course, even the 'best' K-S values are relatively large, that is, the goodness-of-fit is not ideal. However, we can say that the FTP and GridFTP throughput have similar distributions because the K-S values and fitting parameters of FTP and GridFTP are very similar.

Table 3 K-S values and fitting parameters of the LAN traffic

	K-S Test	**Parameters**
FTP	0.28012	k=-0.21512, σ=1953.0, μ=-390.38
GridFTP	0.29053	k=-0.21468, σ=1966.8, μ=-492.46

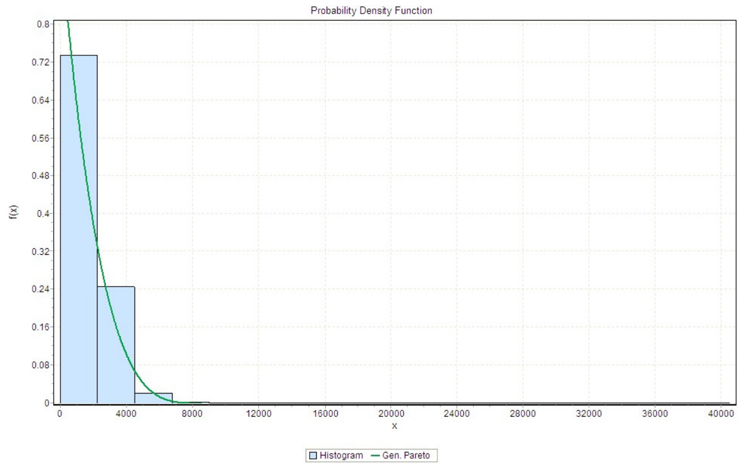


Figure 5 (a) - Fitting graph of the FTP traffic in LAN

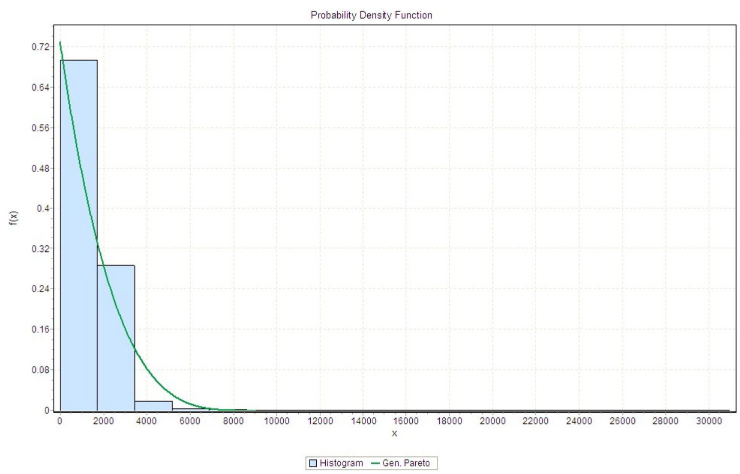
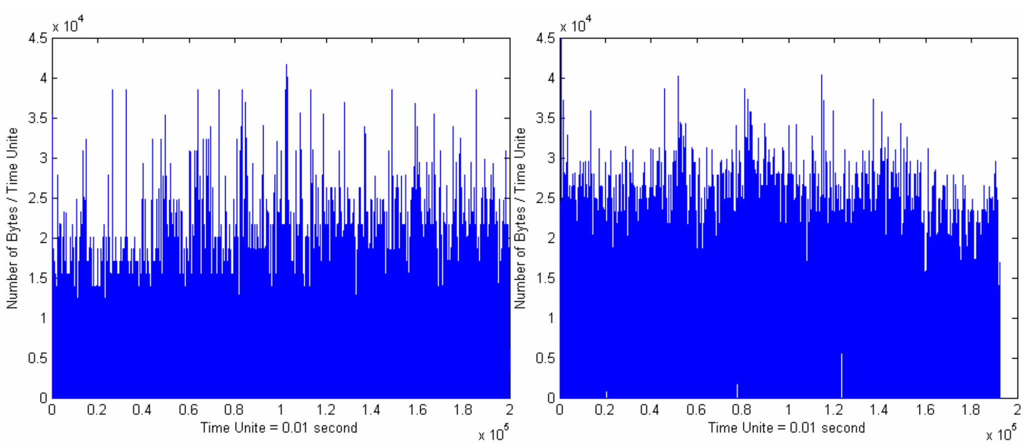


Figure 5 (b) - Fitting graph of the GridFTP (-p 3) traffic in LAN

The same analysis work was also done for the WAN FTP and GridFTP traffic. Figure 6 shows the FTP and GridFTP throughput in WAN.



(a) FTP (b) GridFTP (-p 3)

Figure 6 - Throughput in WAN

We then fit the WAN throughput to the same ‘generalized pareto’ distribution as described for the LAN traffic analysis and we get the following K-S values and fitting parameters shown in Table 4.

Table 4 K-S values and fitting parameters of the WAN traffic

	K-S Test	Parameters
FTP	0.52197	k=0.75983, σ=310.35, μ=-120.07
GridFTP	0.5220	k=0.76181, σ=296.53, μ=-118.19

By comparing the intuitional throughput graphs and distribution fitting results of the FTP and GridFTP traffic both in LAN and in WAN, we get the conclusion that they have similar throughput characteristics.

5. GridFTP Simulator Development

The analysis results from section 4 consolidate our belief in simulating GridFTP with FTP flows in ns2 based on their similarity. This section will present the detailed simulator development.

5.1 General Description

The basic GridFTP simulator has three input parameters:

- Bandwidth of the network
- Number of parallel streams
- Ratio of the data transferred by each stream

The output of the simulator is a traffic throughput graph. Figure 7 gives the main flowchart of the simulator.

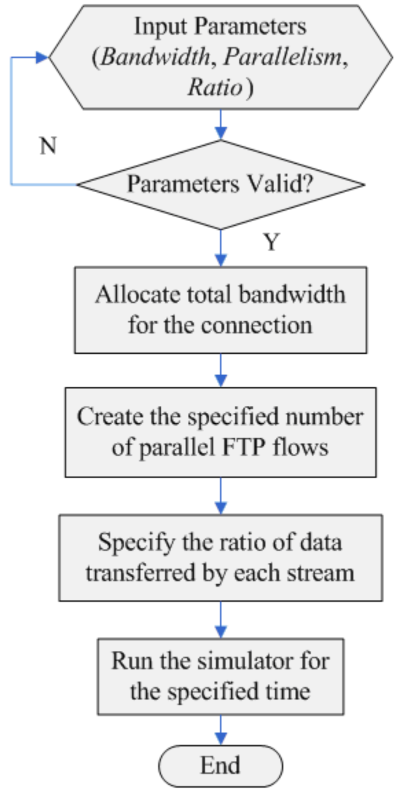


Figure 7 - Flowchart of coding basic GridFTP connection

5.2 Simulator Development

The GridFTP simulator is developed with the OTCL language and is embedded in a gridftp.tcl file. As outlined above, the three major parameters defined for the simulator are:

- Bandwidth: this parameter is used to set the total bandwidth of the link. By default, this parameter is set to 1.0Mbps. With this and the ratio parameter, we can determine the "rate_" parameter for each FTP instance.
- Parallel: this parameter is used to set the parallel GridFTP streams. By default, this is set to 4. Since each GridFTP stream can be simulated by FTP, this parameter will actually set the number of FTP instances for the GridFTP simulator.
- Ratio: this parameter is used to set the throughput ratio among the parallel streams. By default, this is set to 1:1:1:1 which means each stream will transmit packets at an equal speed.

The GridFTP simulator consists of two classes. One is the GridFTP class and the other is the GridFTPSink class. We also override two methods for the basic Simulator class, *attach-agent* and *connect*, with which the GridFTP instance can be attached to the network node and be connected to the GridFTPSink instance.
Due to the similarity between GridFTP and FTP, we provide similar interfaces in the GridFTP class as those of FTP as follows.

- start {}: start continuously sending packets and the underlying TCP agents are always getting more packets to send
- stop {}: stop sending packets
- send {n}: send n bytes of data
- produce {n}: send n packets
- producemore {n}: send n more packets

In addition, three interfaces for setting the parameters are defined below.

- setParallel {n}: this interface sets n parallel GridFTP streams, which will actually create n FTP instances and bind them to n TCP agents.
- setRatio {args}: this interface parses the args parameter and extracts the throughput ratio for each GridFTP stream.
- setBandwidth {n}: this interface sets nMbps bandwidth to the link and assigns the "rate_" parameter to each FTP instance appropriately.

For the GridFTPSink class, the most important interface is setParallel {n}, which will create n TCPSink instances.

We also override the following two methods of the Simulator class.

- attach-agent {node agent}: this interface attaches the GridFTP instance with the network node, which will in turn attach the underlining TCP agents to the network node.
- connect {src dst}: this interface connects the GridFTP instance with the GridFTPSink instance, which will in turn connect each pair of the TCP agent and the TCPSink agent.

6. Simulation

In order to verify the efficiency of the simulator, we define four scenarios and acquire some real traffic data from the testbed. Based on this data, we determine the input parameters of the simulator and get the simulation results.

6.1 Test Scenario Setup and Real Traffic Captured

The four GridFTP test scenarios are described as follows:

(1) 3 nodes / single stream transmission (writing as '*node3-p1*')

- Node_1 and node_2 are located in the BUPT LAN.
- Node_3 is in the ISCAS LAN.
- Node_3 transfers a file of 200MB to node_1 with a single GridFTP stream.
- At the time point of 600 seconds from the beginning of the above transmission, node_2 transfers a 200MB file to node_1 with a single GridFTP stream. At 1200 seconds, node_2 transfers to node_1 again.

(2) 3 nodes / 8 parallel stream transmission (writing as '*node3-p8*')

- All the configurations are the same as '*node3-p1*', except that the GridFTP are 8 parallel streams.

(3) 5 nodes / single stream transmission (writing as '*node5-p1*')

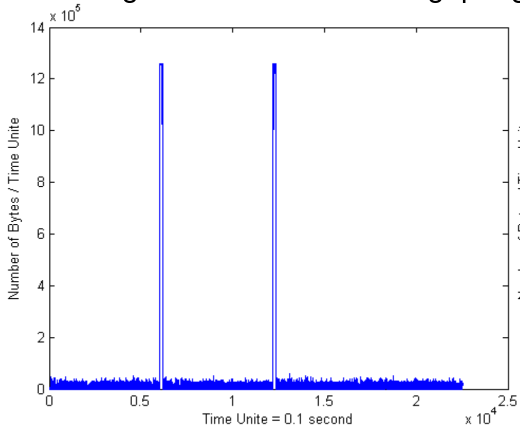
- Node_1, node_2 and node_3 are located in the BUPT LAN.
- Node_4 and node_5 are in the ISCAS LAN.

- Node_4 and node_5 transfer 100MB files to node_1 simultaneously.
- At the time point of 600 second, node_2 and node_3 transfer 200MB files to node_1 simultaneously with single GridFTP stream. At 1200 second, node_2 and node_3 transfer to node_1 simultaneously again.

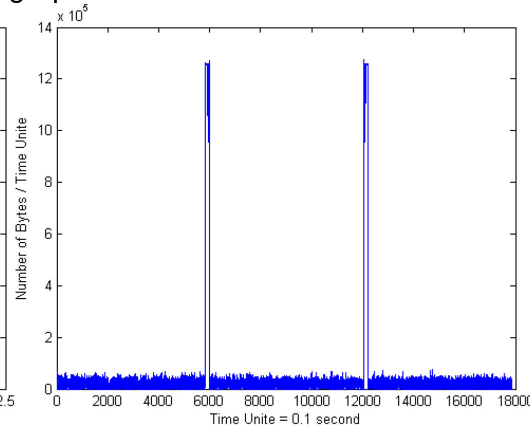
(4) 5 nodes / 8 parallel stream transmission (writing as '*node5-p8*')

- All the configurations are the same as '*node5-p1*', except that the GridFTP are 8 parallel streams.

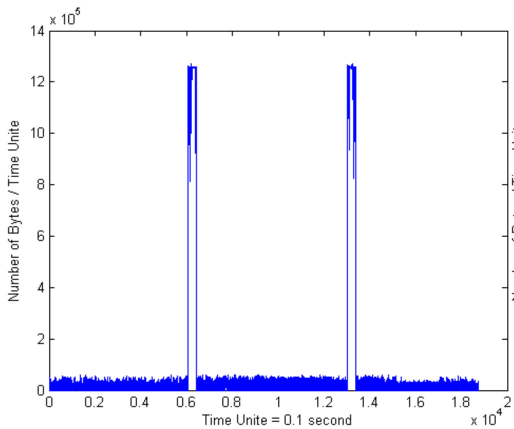
The data captured from the above scenarios are all mixtures of LAN and WAN traffic. Figure 8 shows the throughput graphs.



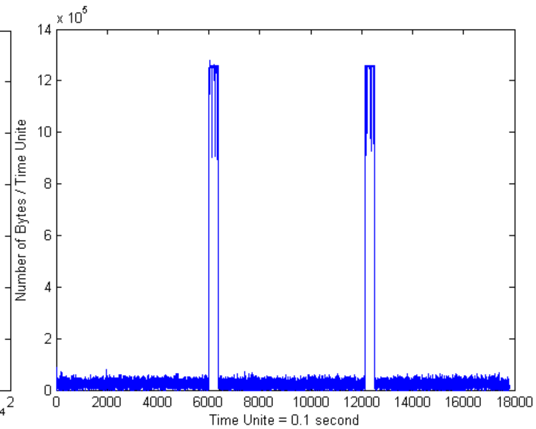
(a) node3-p1



(b) node3-p8



(c) node5-p1



(d) node5-p8

Figure 8 - Throughput graphs of the four scenarios

6.2 Simulation Results

6.2.1 Computing the Input Parameters

For simulating the proposed scenarios with our simulator, firstly we should compute the input parameters according to the real traffic. Table 5(a) gives the 'Bandwidth' computational result of each connection in every scenario, and Table 5(b) shows the 'Runtime' parameters (including starting time and stopping time of each connection). The data transferred by each stream are approximately equal, so the 'Ratio' parameter can be fixed to '1:1:1:…'.

Table 5 (a) 'Bandwidth' parameters computed from real traffic

Scenario	**Bandwidth**			
	LAN		**WAN**	
node3-p1	94.168 Mbps		802.485 Kbps	
node3-p8	92.834 Mbps		1448.732 Kbps	
node5-p1	50.264 Mbps	46.423 Mbps	563.357 Kbps	557.980 Kbps
node5-p8	52.087 Mbps	46.415 Mbps	848.543 Kbps	848.423 Kbps

Table 5 (b) 'Runtime' parameters computed from real traffic

Scenario	**Runtime** (second)			
	LAN		**WAN**	
node3-p1	600.0~618.1; 1200.0~1218.2		0.0~2256.7	
node3-p8	600.0~618.6; 1200.0~1218.4		0.0~1787.2	
node5-p1	600.0~633.8; 1200.0~1232.9	600.0~636.7; 1200.0~1236.9	0.0~1860.5	0.0~1872.0
node5-p8	600.0~632.7; 1200.0~1231.7	600.0~636.8; 1200.0~1236.9	0.0~1779.4	0.0~1773.8

6.2.2 Simulation Codes

We give the detailed simulation method of the '*node5-p8*' scenario as follows, the code for the other three scenarios are similar and can be easily derived.

Firstly we should include "gridftp.tcl", the tcl file of the GridFTP simulator.

```
# Import the GridFTP class
source gridftp.tcl
```

The next step is to create the simulator object and trace file.

```
# Create a simulator object
set ns [new Simulator]
# Open the trace file
set nd [open node5-p8.tr w]
$ns trace-all $nd
```

Then the input parameters are specified, which have been computed from the real traffic. For example, the variable 'bandwidth41' stands for the bandwidth between node_4 and node_1 (848.543Kbps), 'parallel31' for the number of parallel streams between node_3 and node_1 (8), and 'start21_2' for the start time of the second transmission between node_2 and node_1 (the 1200.0th second).

```
set bandwith21  52.087
set bandwith31  46.415
set bandwith41  [ expr 848.543/1024 ]
set bandwith51  [ expr 848.423/1024 ]
```

```
set parallel21  8
set parallel31  8
set parallel41  8
set parallel51  8
```

```
set start21_1 600.0
set stop21_1 632.7
set start21_2 1200.0
set stop21_2 1231.7
set start31_1 600.0
set stop31_1 636.8
set start31_2 1200.0
set stop31_2 1236.9
set start41 0.0
set stop41 1779.4
set start51 0.0
set stop51 1773.8
```

After the parameter inputting, the finish procedure should be defined.

```
# Define the 'finish' procedure
proc finish {} {
  global nd ns
 close $nd
 exit 0
}
```

Then the network topology is defined. Here five nodes are included, which are connected by four links.

```
# Create five nodes
set node1 [$ns node]
set node2 [$ns node]
set node3 [$ns node]
set node4 [$ns node]
set node5 [$ns node]
# Create links between the nodes
$ns duplex-link $node1 $node2 [expr $bandwith21]Mb 2ms DropTail
$ns duplex-link $node1 $node3 [expr $bandwith31]Mb 2ms DropTail
$ns duplex-link $node1 $node4 [expr $bandwith41]Mb 10ms DropTail
$ns duplex-link $node1 $node5 [expr $bandwith51]Mb 10ms DropTail
```

The following codes show the creation of the GridFTP generator ‘gridftp2’ using ‘Application/GridFTP’ defined in ‘gridftp.tcl’. The other three generators ‘gridftp3’~’gridftp5’ are similar to this. These generators are attached to node_2~node5 respectively.

```
# Create an GridFTP application, then attach it to node2, node3, node4 and node5
set gridftp2 [new Application/GridFTP]
$gridftp2 setParallel $parallel21
$gridftp2 setPacketSize 1474
$gridftp2 setRatio 1:1:1:1:1:1:1:1
$gridftp2 setWindows 20
$ns attach-agent $node2 $gridftp2
```

The GridFTP sink agent ‘gridftpsink21’ is created using ‘Agent/GridFTPSink’ defined in ‘gridftp.tcl’ and is attached to node1. The other three sink agents are created and attached in the same way.

```
# Create a GridFTP sink agent and attach it to node1
set gridftpsink21 [new Agent/GridFTPSink]
$gridftpsink21 setParallel $parallel21
$gridftpsink21 setPacketSize 66
$ns attach-agent $node1 $gridftpsink21
$ns connect $gridftp2 $gridftpsink21
```

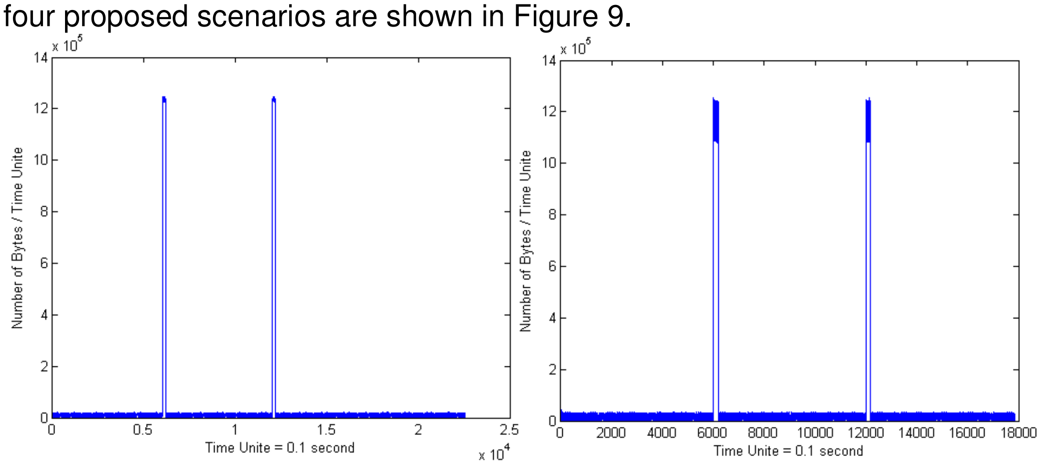
Now it’s time to define the start and the stop procedures of the simulation, and finally run the simulation.

```
# Schedule events for all the connections
$ns at $start41 "$gridftp4 start"
$ns at $start51 "$gridftp5 start"
$ns at $start21_1 "$gridftp2 start"
$ns at $start31_1 "$gridftp3 start"
$ns at $stop21_1 "$gridftp2 stop"
$ns at $stop31_1 "$gridftp3 stop"
$ns at $start21_2 "$gridftp2 start"
$ns at $start31_2 "$gridftp3 start"
$ns at $stop21_2 "$gridftp2 stop"
$ns at $stop31_2 "$gridftp3 stop"
$ns at $stop41 "$gridftp4 stop"
$ns at $stop51 "$gridftp5 stop"
```

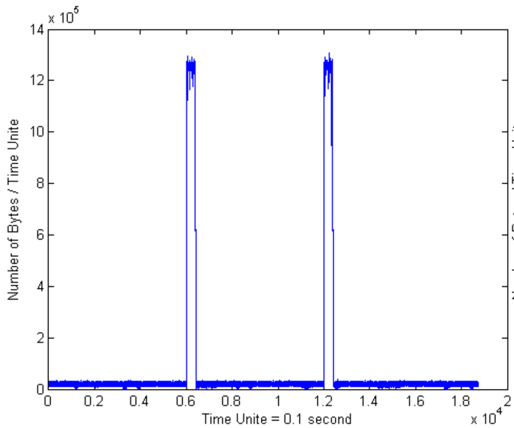
```
# Call the finish procedure
$ns at 1780 "finish"
# Run the simulation
$ns run
```

6.2.3 Simulation Evaluation

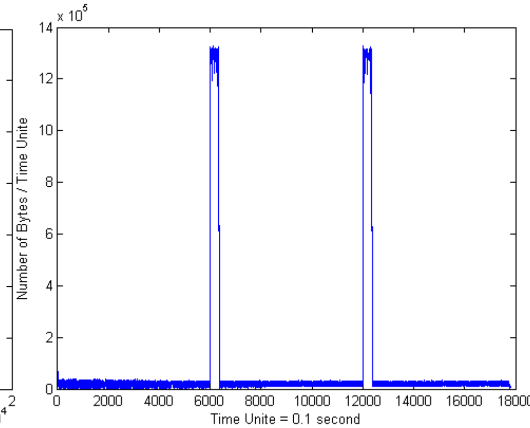
In order to evaluate the simulation results, we analyze the trace files generated by the above ns2 code. The throughput simulation graphs of the four proposed scenarios are shown in Figure 9.



(a) node3-p1 (b) node3-p8



(c) node5-p1



(d) node5-p8

Figure 9 - Throughput simulation results of the four scenarios

By comparing the real throughput (Figure 8) to the simulation throughput (Figure 9), we see that they both look very similar. We can then provide some more rigorous mathematical proofs using the goodness-of-fit comparison. Again, we fit the real traffic and simulation traffic to the 'generalized pareto' distribution, and get the following K-S values and fitting parameters as shown in Table 6.

Table 6 K-S values and fitting parameters of the real traffic and simulation traffic

Scenarios	**Real Traffic**		**Simulation Traffic**	
	K-S values	**Parameters**	**K-S values**	**Parameters**
node3-p1	0.20305	k=0.76627, σ=7137.0, μ=-839.43	0.22643	k= 0.77417, σ= 6968.4, μ=-928.37
node3-p8	0.14957	k=0.72833, σ=10270.0, μ= 3917.6	0.13277	k= 0.73085, σ=8799.5, μ= 4148.0
node5-p1	0.13759	k= 0.79812, σ= 12713.0, μ= -922.52	0.14301	k= 0.78026, σ= 10964.3, μ= -875.56
node5-p8	0.15904	k= 0.78885, σ= 10244.0, μ= 5054.3	0.16551	k= 0.79062, σ= 9885.4, μ= 5627.0

From Figure 10 and Table 6, we believe that the model proposed is indeed suitable for simulating GridFTP traffic.

7. Conclusion

In this work, a testbed for GridFTP traffic testing and analyzing is established and some useful modelling methods are defined. Packet size probability and traffic distribution fitting indicate that GridFTP is similar to FTP in terms of real network traffic so we conclude that we can effectively simulate GridFTP with FTP flows in ns2. We implement two classes, 'Application/GridFTP' and 'Agent/GridFTPSink', for the basic GridFTP connection simulation. Then, four scenarios are presented to validate the correctness of the simulator by comparing real and simulation traffic. Our experimental results show the efficiency of the model proposed.

8. Future Work

This work represents the basic implementation of the GridFTP simulator. In the future, we plan to improve the model in the following two aspects: (1) The user gives the max bandwidth of the connection measured by a specific tool (such as 'Iperf'), then the model could adaptively simulate different transmission bandwidths using a number of parallel streams. (2) The simulator could adaptively compute the ratio of the data carried by each stream in a parallel transmission mode.

9. References

[1] The Globus Alliance. http://www.globus.org/

[2] Allcock B, Bester J, Bresnahan J, et al. Data management and transfer in high-performance computational grid environments [J]. Parallel Computing, 2002, 28(5): 749-771.