Research White Paper

WHP 178

January 2010

GRIDFT Server Simple Performance Measurements

David Butler

BRITISH BROADCASTING CORPORATION

BBC Research White Paper WHP 178

GRIDFT Server Simple Performance Measurements

David Butler

Abstract

GRIDFTP is an open source and open standard high speed file transfer service, employing digital certificates and extensions to the existing File Transfer Protocol (FTP). GRIDFTP can be deployed as part of the Globus Toolkit or as a stand alone service.

The inherent TCP/IP delay problem of FTP is overcome by employing parallelisation and other techniques, to provide an efficient, secure, high speed content movement service.

This white paper documents the performance of gsiftp globus-url-copy over an emulated Wide Area Network, with latency and packet loss.

Many thanks to Belfast e-Science Centre, for instructions on GRIDFTP server installation and setup, and to Chris Chambers and Peter Brightwell for their advice on optimising TCP settings.

This document was originally published in 2010.

Additional key words: GRID, Globus, Cloud Computing, X.509, IETF

White Papers are distributed freely on request.

Authorisation of the Chief Scientist is required for publication.

BBC Research White Paper WHP 178

GRIDFT Server Simple Performance Measurements

David Butler

1 Introduction

GRIDFTP is a set of extensions to FTP, implemented by the GRID community for the Globus Alliance [Ref. 1]. Originally intended as part of a GRID middleware, GRIDFTP can be used independently, to provide high speed file transfers.

The GRIDFTP protocol is implemented either as a GRIDFTP server or GRIDFTP client. GRIDFTP supports a variety of transfer and security mechanisms for server to server and server to client file movement and is able to employ digital certificates for server and user authentication.

A GRIDFTP server implementation is more complex and less platform independent, but supports more transfer mechanisms. A GRIDFT client implementation is simpler and more platform independent, but supports fewer transfer mechanisms. A physical client on a network could employer either a GRIDFTP server or GRIDFTP client implementation to access a network of GRIDFTP servers. The tests documented in this white paper are for GRIDFTP server-to-server transfers, without the use of a Globus middleware.

2 GRIDFTP Protocol

The GRIDFTP Protocol is s a set of IETF [Ref. 2] RCS and draft extensions to FTP, as well a specific Globus extensions.

2.1 GRIDFTP Standards

The following were listed at GlobusWORLD 2005 [Ref. 3] and on the Open GRID Forum [Ref. 4]

IEFT FTP Recommendations:

RFC 959: File Transfer Protocol [Ref. 5]

RFC 2228: FTP Security Extensions [Ref. 6]

RFC 2389: Feature Negotiation of File Transfer Protocol [Ref. 7]

RFC 2428: FTP Extensions for IPv6 and NATs [Ref. 8]

GRIDFTP Open GRID Forum Protocol Extensions:

GFD.20: GRIDFTP: Protocol Extensions to FTP for the Grid [Ref. 9]

GFD.21: GRIDFTP Protocol Improvements [Ref. 10]

GFD.47: GRIDFTP v2 Protocol Description [Ref. 11]

2.2 FTP and GRIDFTP Overview

Both FTP and GRIDFTP employ control and data channels, using the TCP/IP protocol. The control channel is maintained by the Control Channel Interpreter and data channel is maintained by the Data Protocol Interpreter, as shown in Figure 2.1.

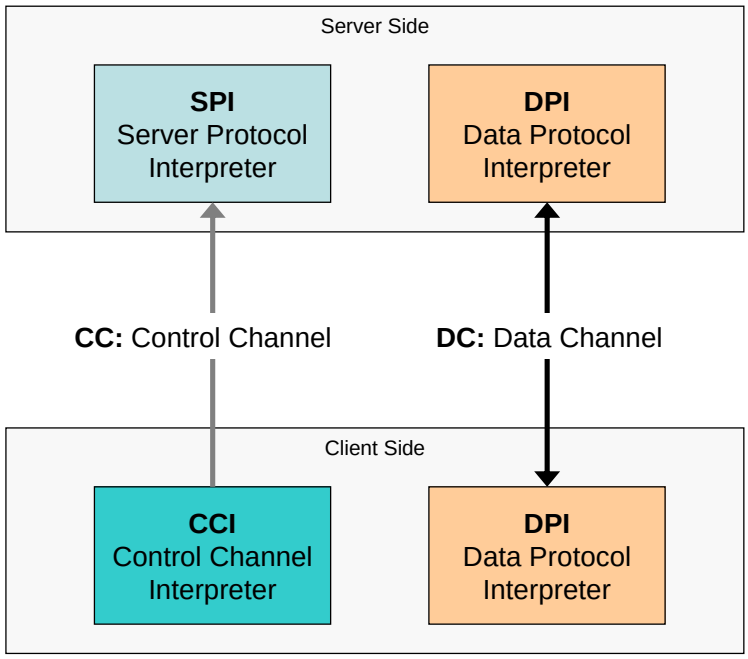


Figure 2.1: FTP/GRIDFTP Control and Data Channels

2.3 FTP Functionality

A FTP data transfer is limited by the maximum size of TCP/IP packet and the acknowledged (ACK) reception of each data packet, as shown in Figure 2.2, below.

In WANs, FTP is affected by latency, as the transmission delay of DATA (data packet) and the ACK (acknowledge) reduces the data transfer rate.

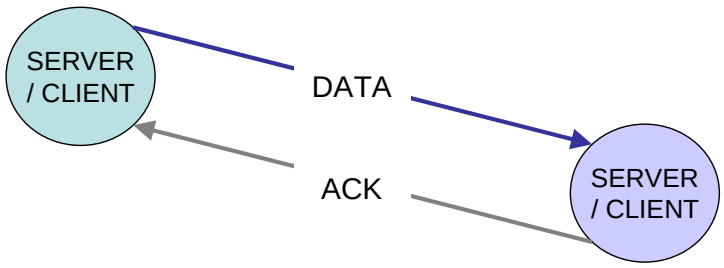


Figure 2.2: FTP Transfers

2.4 GRIDFTP Functionality

GRIDFTP provides authentication based security and mechanisms for overcoming the inherent TCP/IP packet size and acknowledge latency problems, as shown in Figures 2.3 and 2.4.

Security / Authentication
Robust and flexible authentication, integrity, and confidentiality for file access and transfer, supporting multiple authentication options:

- Anonymous use
- Username and password.
- Grid Security Infrastructure (GSI)
- GSSAPI authentication mechanisms defined by RFC 2228 [Ref. 6]

Parallel Data Transfer
GRIDFTP supports parallel data transfer through FTP command extensions and data channel extensions.

The use of multiple TCP streams in parallel (even between the same source and destination) improves the aggregate bandwidth over a single TCP stream.

Automatic Negotiation of TCP Buffer/Window Sizes
Using optimal settings for TCP buffer/window sizes improves performance, especially in WANs.

GRIDFTP extends the standard FTP command set and data channel protocol to support both manual setting and automatic negotiation of TCP buffer sizes for large files and for large groups of small files.

Striped Data Transfer
Data may be striped or interleaved across multiple servers, as in a DPSS network disk cache or a striped file system.

GRIDFTP includes extensions that initiate striped transfers, which use multiple TCP streams to transfer data that is partitioned among multiple servers. Striped transfers provide further bandwidth improvements over those achieved with parallel transfers.

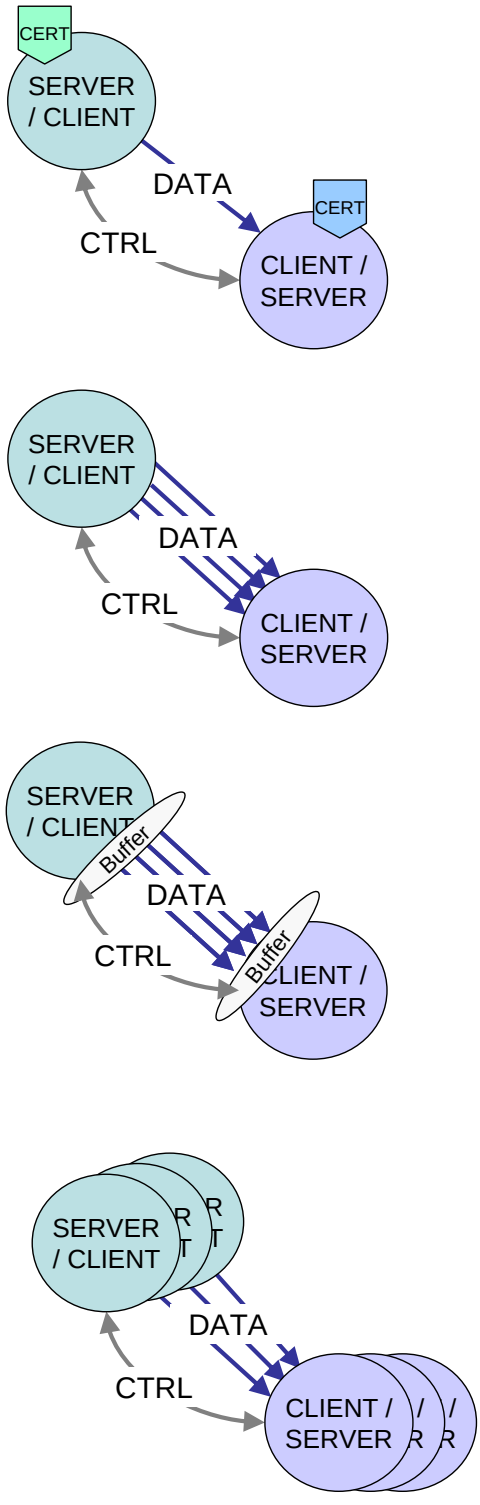
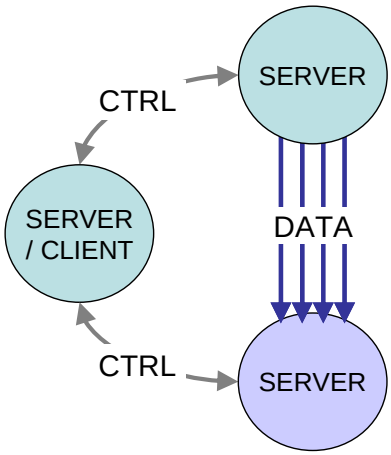


Figure 2.3: GRIDFTP Transfers

Third-party Control of Data Transfer
To manage large amounts of distributed data, authenticated third-party control of data transfers between storage servers is required.

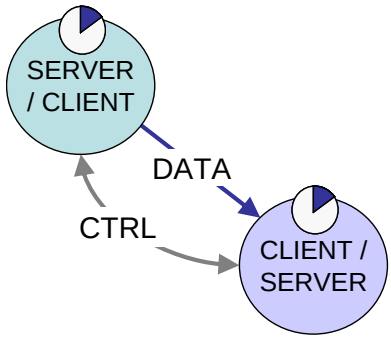
A third-party operation allows a user or application at one site to initiate, monitor and control a data transfer operation between two other sites, the source and destination for the data.

A third party transfer can only be made between GRIDFTP servers. A GRIDFTP client can control a third party transfer, but cannot be the source or destination.



Partial File Transfer
Some applications can benefit from transferring portions of files rather than complete files.

Standard FTP protocol allows the transfer of the remainder of a file starting at a particular offset. GRIDFTP provides commands to support transfers of arbitrary subsets or regions of a file.



Reliable and Re-startable Data Transfers
GRIDFTP incorporates fault tolerant features to handle transient network failures, server outages, etc.

The FTP standard includes a basic feature for restarting failed transfers, but these are not widely implemented. GRIDFTP exploits these features and extends them to cover the new data channel protocol.

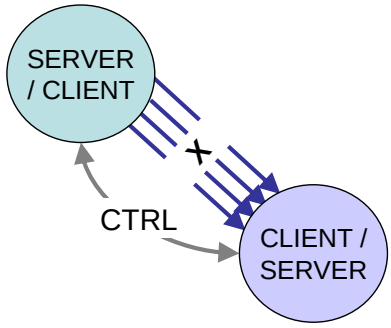


Figure 2.4: GRIDFTP Transfers (continued)

3 GRIDFTP Test Setup

The performance of GRIDFTP transfers between two GRIDFTP servers was measured against latency and packet loss. One-way latencies of up to 50ms (100ms RTT) were tested, the equivalent of large European WANs. The latency and packet loss were introduced using a Shunra Network Emulator [Ref. 12], as shown in Figure 3.1. All network connections are 1 Gigabit Ethernet.

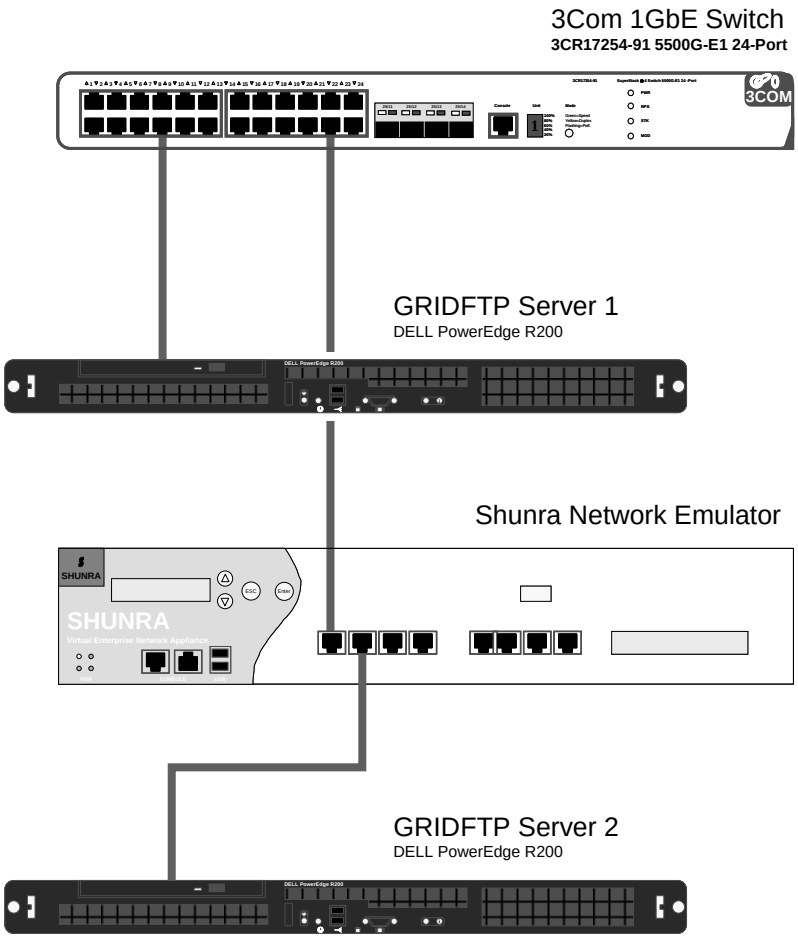


Figure 3.1: GRIDFTP Test setup

Physical server specifications:

Quad Core Intel® Xeon® X3320, 2.5GHz, 2x3MB, Cache, 1333MHz FSB
4GB DDR2 SDRAM 667MHz Memory
2 x 250GB SATA (7,200rpm) Hard Drive

GRIDFTP server software install:

Server 1: Ubuntu 7.04; Linux 2.6.20-15-server #2 SMP Sun Apr 15 2007 i686 GNU/Linux
Server 2: Ubuntu 8.04.1; Linux 2.6.24-21-server #1 SMP Tue Oct 21 2008 x86_64 GNU/Linux
GT 4.2.1 GridFTP, gt4.2.1-all-source-installer
http://www-unix.globus.org/toolkit/survey/index.php?download=gt4.2.1-all-source-installer.tar.gz

4 GRIDFTP Globs-url-copy Configuration

Established guide values for optimum parallelisation and TCP Buffer Size values are provided in the globus-url-copy command parameters [Ref. 13], the Globus GRIDTP tutorials [Ref. 14] and the GT4 GRIDFTP user's tutorial [Ref .15]. These suggest employing a parallelisation value (-p) of 4 to 8 and TCP window size (-tcp-bs) of 2MB or more.

Formulas for calculating the optimum TCP buffers size (TCP-BS) and parallelisation values are also provided in the globus-url-copy command parameters and GT4 GRIDFTP users tutorial.

$$TCP-BS(MB) = \frac{NetworkBandwidth(Mb/s) \times RoundTripDelay(ms) \times 1000}{8 \times 10^6} \qquad \text{[Ref. 13]}$$

$$TCP-BS(MB) = \frac{NetworkBandwidth(Mb/s) \times RoundTripDelay(ms) \times 1000}{8 \times 10^6 \times (Parallelisation - 1)} \qquad \text{[Ref. 15]}$$

The Round Trip Delay is equal to 2 x the one-way latency. These equations provide the theoretical TCP-BS values against latency shown in Figure 4.1.

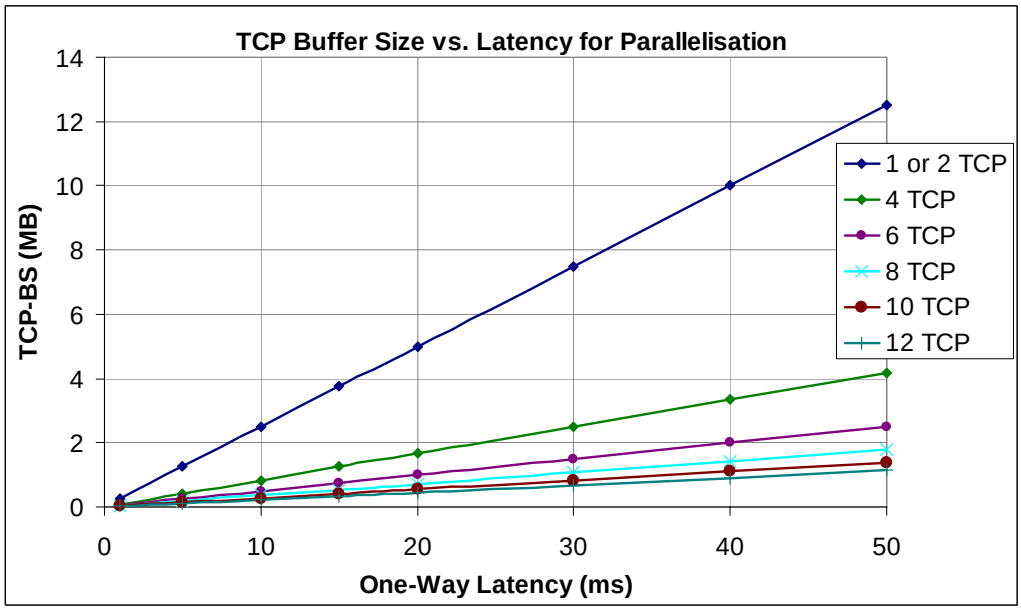


Figure 4.1: Theoretical TCP-BS vs. One-way Latency

Figure 4.1 shows the optimum TCP buffer size decreasing with parallelisation, but increasing with latency.

TCP settings in the operating system TCP stack will also impact the performance of GRIDFTP. However, increasing the TCP maximum window size could be detrimental to the performance of other server applications.

5 GRIDFTP Performance Measurements

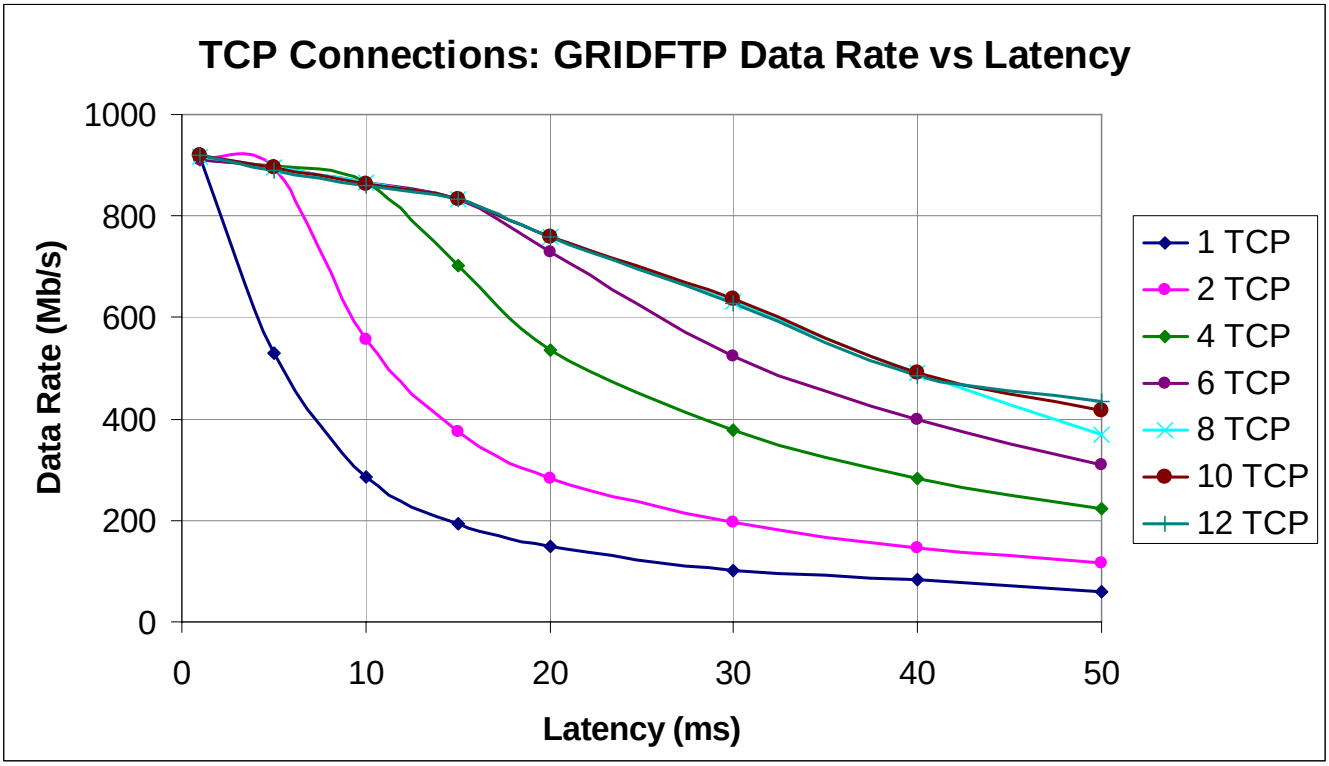
Performance measurements were made for different parallelisation (number of TCP connections) and TCP buffer sizes, for non-optimised and optimised TCP maximum window sizes.

The transfer data rate at the network interface was measured against both latency and packet loss for different globus-url-copy parallelisation settings. The transfer data rate was also measured against latency for different globus-url-copy TCP buffer sizes.

All file transfers employ a timed gobus-url-copy PULL of a 1GB (1073741824 bytes) data file from GRIDFTP server 2 to a ///dev/null on GRIDFTP server 1. A null file destination was employed to remove the effect of hard disk writing speed on the measurements.

5.1 Parallelisation Performance vs Latency with Non-Optimised TCP Window Sizes

The GRIDFTP date rate was measured against one-way latency for different globus-url-copy parallelisation settings, with non-optimised TCP window settings in the server TCP stack. The results can be seen in Figure 5.1 and Table 5.1.



globus-url-copy -p X gsiftp://gridftpserver2:2811/home/griduser/1GB.dat file:///dev/null

Figure 5.1: GRIDFTP Performance vs. One-Way Latency for Number of Parallel TCP Connections

Parallelisation (TCP)	Measured Bit Rate (Mb/s) vs. Network One-Way Latency (ms)							
	1ms	5ms	10ms	15ms	20ms	30ms	40ms	50ms
1	915.22	530.02	286.34	194.89	147.85	100.35	82.44	60.57
2	914.35	894.44	555.64	376.14	282.90	196.43	146.16	117.54
4	911.24	897.90	865.16	702.93	534.94	376.64	283.99	221.90
6	911.92	896.87	866.38	833.67	729.38	524.20	399.89	310.47
8	916.69	897.15	865.34	832.70	758.09	631.04	489.65	369.83
10	918.35	894.35	862.81	832.94	759.37	636.56	489.73	417.48
12	920.03	890.55	860.83	831.97	758.56	627.54	485.80	433.19

Table 5.1: GRIDFTP Performance vs. Latency for Number of Parallel TCP Connections

For these measurements, the TCP window settings for GRIDFTP Server 1 were not optimised. GRIDFTP Server 2 TCP window settings were set to preferred values. The server TCP values are set in the /etc/sysctl.conf file.

Server 1: Default (Un-optimised) TCP maximum window size settings:

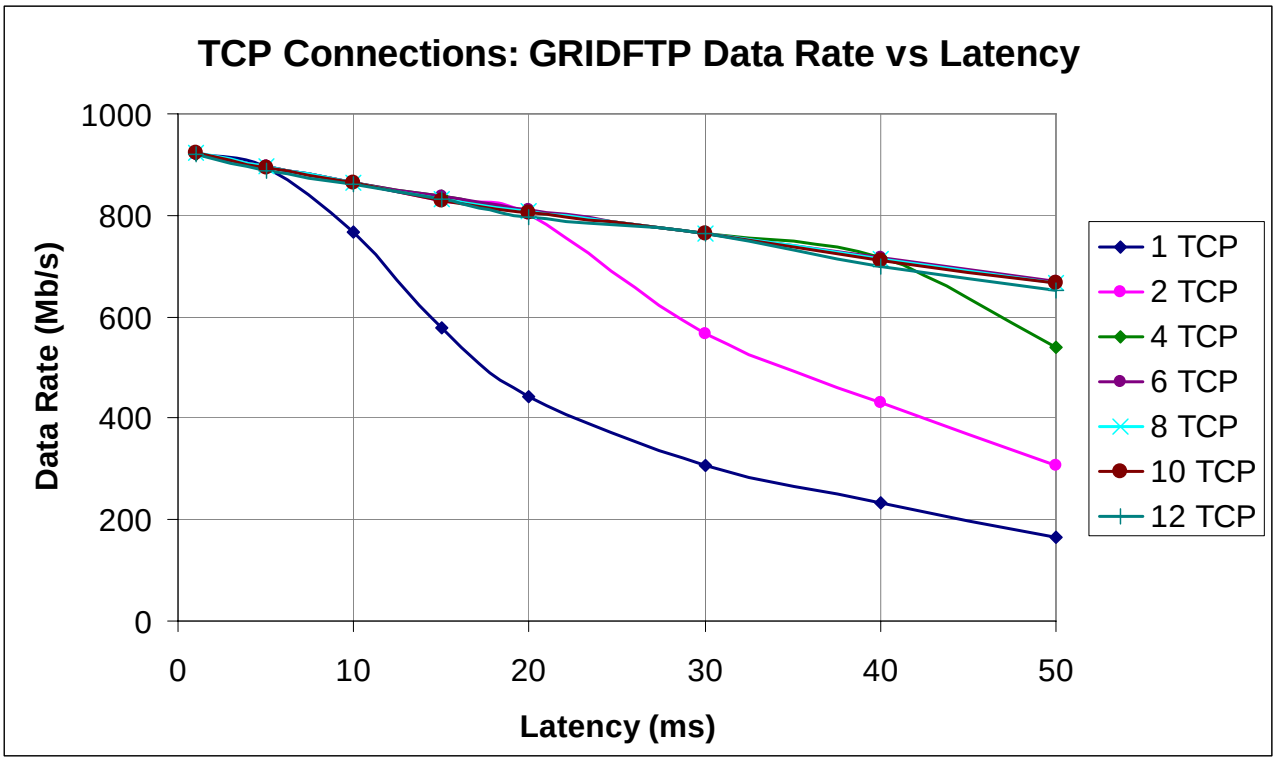
```
# Auto tuning TCP buffer limits
net.ipv4.tcp_rmem =   4096    87380  1048576
net.ipv4.tcp_wmem = 4096    16384  1048576
sysctl net.ipv4.tcp_window_scaling = 1
sysctl net.ipv4.tcp_timestamps = 1
sysctl net.ipv4.tcp_sack = 1
# TCP max buffer size setable using setsockopt()
net.core.wmem_max = 131071
net.core.rmem_max =131071
```

Server 2: Preferred (Optimised) TCP maximum window size settings of 4MB:

```
# Auto tuning TCP buffer limits
net.ipv4.tcp_rmem =   4096       87380   4194304
net.ipv4.tcp_wmem = 4096       65536   4194304
sysctl net.ipv4.tcp_window_scaling = 1
sysctl net.ipv4.tcp_timestamps = 1
sysctl net.ipv4.tcp_sack = 1
# TCP max buffer size setable using setsockopt()
net.core.wmem_max = 4194304
net.core.rmem_max = 4194304
```

5.2 Parallelisation Performance vs Latency with Optimised TCP Window Sizes

The GRIDFTP date rate was measured against one-way latency for different globus-url-copy parallelisation settings, with optimised TCP maximum window settings of 4MB. The results can be seen in Figure 5.2 and Table 5.2.



globus-url-copy -p X gsiftp://gridftpserver2:2811/home/griduser/1GB.dat file:///dev/null

Figure 5.2: GRIDFTP Performance vs. One-Way Latency for Number of Parallel TCP Connections

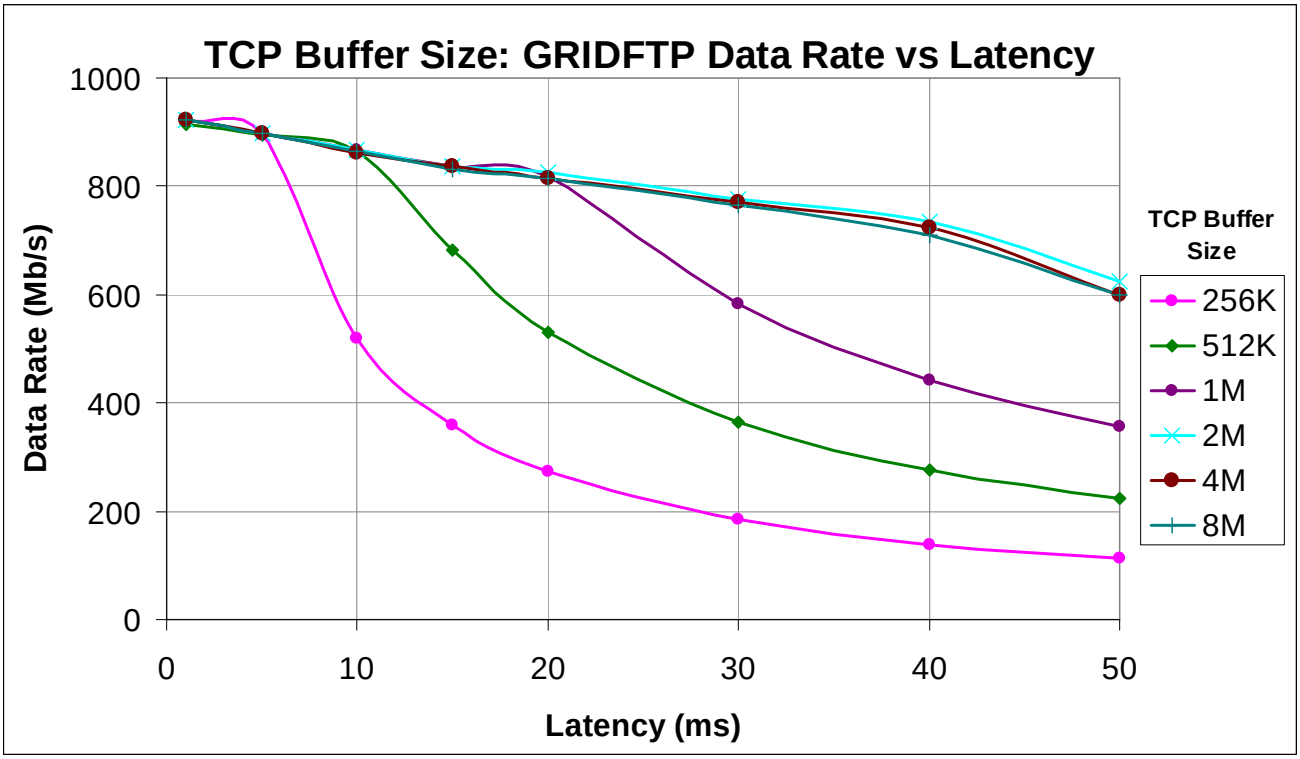
Parallelisation (TCP)	Measured Bit Rate (Mb/s) vs. Network One-Way Latency (ms)							
	1ms	5ms	10ms	15ms	20ms	30ms	40ms	50ms
1	919.93	893.14	766.35	577.76	442.09	305.41	231.98	165.60
2	921.01	897.72	864.20	829.40	803.56	566.45	431.30	307.65
4	919.93	897.34	864.46	837.33	807.26	763.83	717.08	540.77
6	921.51	895.28	865.16	836.92	811.00	764.91	717.68	668.57
8	921.90	895.94	864.73	832.05	806.81	763.69	713.21	666.65
10	922.30	895.19	863.94	829.32	806.12	763.89	709.67	665.57
12	921.61	889.17	860.65	830.60	795.15	764.64	698.36	651.48

Table 5.2: GRIDFTP Performance vs. Latency for Number of Parallel TCP Connections

Both GRIDFTP Server 1 and Server 2 sysctl.conf files were set with optimised TCP window settings.

5.3 TCP Buffer Size Performance vs Latency for a Fixed Parallelisation

The GRIDFTP date rate was measured against one-way latency for different globus-url-copy TCP buffer sizes with a fixed parallelisation setting of 4. The TCP maximum window settings in the TCP stack sysctl.conf were set to 16MB, for both servers. This was to allow a wider range of control with the -tcp-bs command line option. The results can be seen in Figure 5.3 and Table 5.3.



globus-url-copy -p 4 -tcp-bs X gsiftp://gridftpserver2:2811/home/griduser/1GB.dat file:///dev/

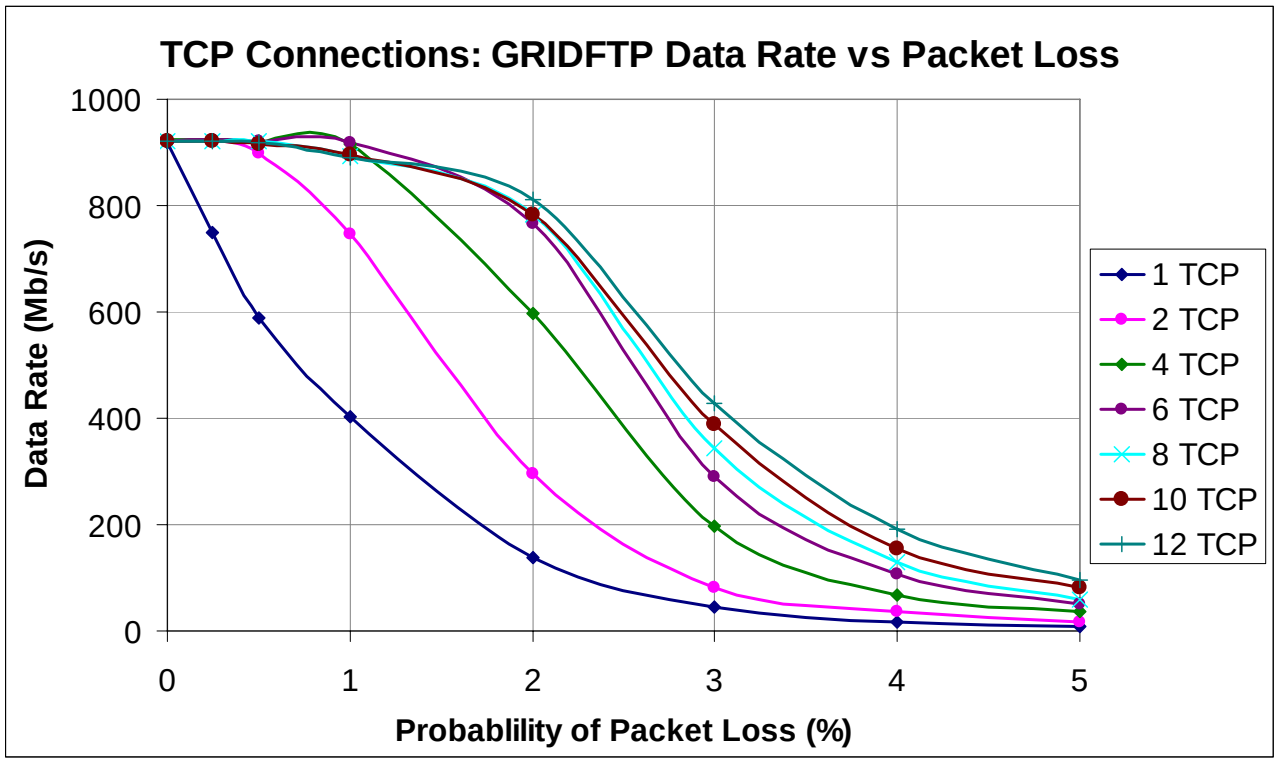
Figure 5.3: GRIDFTP Performance vs. Latency for TCP Buffer Size

TCP Buffer Size	Measured Bit Rate (Mb/s) vs. Network One-Way Latency (ms)							
	1ms	5ms	10ms	15ms	20ms	30ms	40ms	50ms
256KB	920.52	894.63	518.34	359.08	272.97	184.93	139.07	112.25
512KB	914.35	894.63	864.12	682.22	531.00	365.00	277.54	223.96
1MB	922.89	897.90	867.61	836.43	818.34	583.73	442.36	357.49
2MB	923.09	898.28	866.73	837.16	825.26	777.45	734.18	624.30
4MB	921.90	896.69	860.91	838.22	815.00	771.09	724.15	600.72
8MB	921.41	898.09	864.64	831.89	814.23	766.07	708.62	600.30

Table 5.3: GRIDFTP Performance vs. Latency for TCP Buffer Size

5.4 Parallelisation Performance vs Packet Loss with Optimised TCP Window Sizes

The GRIDFTP date rate was measured against percentage packet loss for different globus-url-copy parallelisation settings. Optimised TCP settings were used for both servers, with a TCP maximum window setting of 4MB. The results can be seen in Figure 5.4 and Table 5.4.



globus-url-copy -p X gsiftp://gridftpserver2:2811/home/griduser/1GB.dat file:///dev/null

Figure 5.4: GRIDFTP Performance vs. Percentage Packet Loss for Parallel TCP Connections

Parallelisation (TCP)	Measured Bit Rate (Mb/s) vs. Percentage Packet Loss (%)							
	0%	0.25%	0.5%	1%	2%	3%	4%	5%
1	919.63	748.71	588.61	402.67	138.32	44.14	17.71	8.18
2	922.00	922.40	899.31	746.89	295.24	81.26	36.32	16.24
4	923.19	924.38	921.71	916.88	598.00	196.88	68.19	35.26
6	920.42	923.29	922.00	917.37	767.31	290.14	106.69	49.93
8	921.31	921.21	919.83	892.03	783.05	342.85	128.40	59.64
10	920.82	921.31	916.59	895.10	783.47	388.00	156.04	81.58
12	921.80	921.41	917.37	891.01	812.30	429.49	192.12	96.97

Table 5.4: GRIDFTP Performance vs. Percentage Packet Loss for Parallel TCP Connections

6 Analysis of Results

The results for parallelisation and TCP buffer size vs. latency match the established guide values of 4 to 8 TCP connections and 2MB, from the globus-url-copy command line parameters [Ref. 13] and GT4 user tutorial [Ref. 15]. Increasing the TCP parallelisation and optimising the TCP maximum window size offered significant improvements over a single TCP connection with an un-optimised TCP maximum window size.

As can be seen in Figures 5.1 and 5.2, optimum TCP maximum window size settings in the server sysctl.conf file improves the data rate for all parallelisation settings. However, if the TCP window size is optimised, then the improvement offered by increased parallelisation is less noticeable for higher parallelisation values. In Figure 5.1, with no TCP optimisation, there is no improvement in transfer rate for more than 8 TCP connections. In Figure 5.2, with TCP optimisation, there is only a small improvement for more than 4 TCP connections.

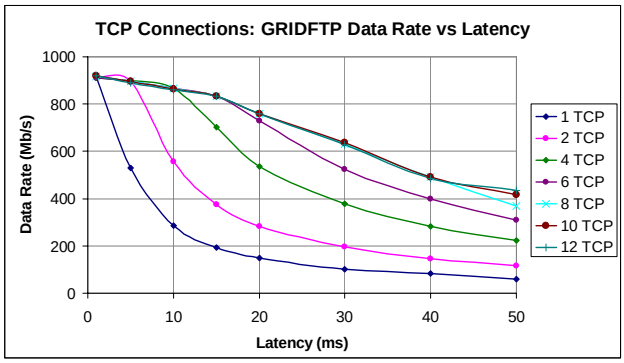


Figure 5.1 (page 7)

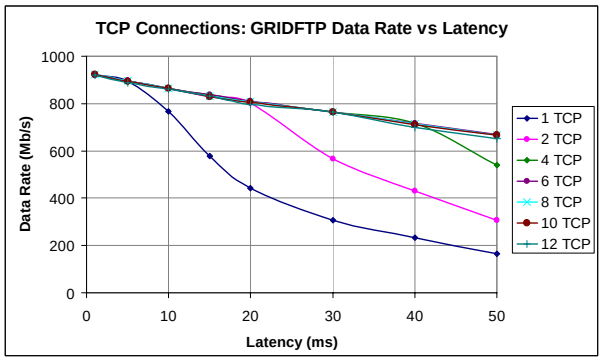


Figure 5.2 (page 9)

In Figure 5.3, the TCP buffer size parameter (-tc-bs) offers little improvement above 2MB. This matched established guide values rather than the formula based results from Figure 4.1.

Modern TCP stacks have some form of automatic optimisation of the TCP window size. Setting the parallelisation (-p) and TCP maximum window size in the server sysctl.conf file has more effect that adjusting the globus-url-copy TCP buffer size (-tc-bs).

As shown in Figure 5.4, increased parallelisation offers improved tolerance to packet loss. Data is spread over multiple TCP flows, such that the loss rate per flow is lower. The loss resistance of individual TCP flows is not improved.

The improvement is less noticeable above 8 TCP connections, but still continues to improve. The packet loss is spread over multiple TCP connections. For 8 TCP connections, a single packet loss at any one time only effects 1 of the 8 connections.

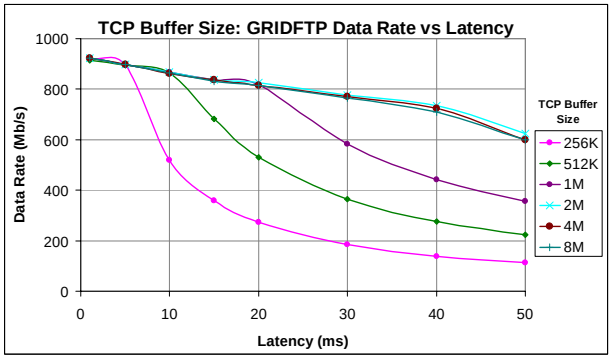


Figure 5.3 (page 10)

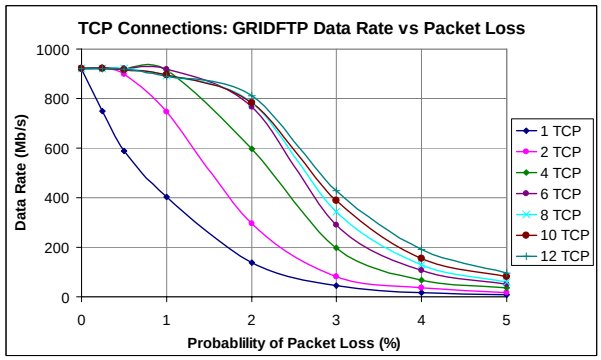


Figure 5.4 (page 11)

7 Conclusion

Using an optimised modern TCP stack in an uncongested lossless WAN, it is possible to achieve high data rates [Ref. 16] without employing GRIDFTP parallelisation techniques. With ideal stack optimisation and an uncongested lossless network, GRIDTP will not provide an improved performance.

Using an un-optimised TCP stack, in an uncongested WAN, it is not possible to achieve high data rates without employing parallelisation or other techniques. GRIDFTP would provide an improved performance.

Using either an optimised or un-optimised TCP stack in a congested network, it is not possible to achieve high data rates with or without GRIDFTP. Any GRIDFTP performance improvement would be at the expense of other traffic on the network. In some cases the use of parallelisation could further congest the network, reducing the GRIDFTP performance.

Real networks have a mixture clients and servers with different operating systems, running multiple applications, with both LAN and WAN traffic. The TCP stacks will not all be optimised for WAN traffic. In such a network, with low or moderate congestion, GRIDFTP performance will be greater with parallelisation, but will not improve beyond 4 to 8 TCP connections. For RTT delays of up 100ms, a TCP buffer size of 2MB will give the best data transfer performance, but smaller buffer sizes could be used.

In WiFi and other lossy networks, where packet loss is not due to congestion, GRIDFTP's tolerance of packet loss will provide improved performance.

GRIDFTP would be a useful protocol for file based production, for moving both compressed and un-compressed files across a mixture of LAN and WAN.

8 References

[1] Globus Alliance
A community of organisations and individuals, developing GRID technologies.
http://www.globus.org/

[2] IETF, Internet Engineering Task Force
An organised activity of the Internet Society (ISOC) is a not-for-profit organization founded to provide leadership in Internet related standards, education, and policy.
http://www.ietf.org/

[3] Globus Alliance
GT4 GRIDFTP for Developers: The New GRIDFTP Server
Bill Allcock, ANL, GlobusWORLD 2005, Feb 7-11, 2005
http://www.globus.org/toolkit/presentations/GlobusWorld_2005_Session_4c.pdf

[4] Open GRID Forum
OGF Grid Final Documents (GFDs)
http://www.ggf.org/gf/docs/

[5] IETF / W3 Network Working Group
RFC 959: File Transfer Protocol
J. Postel, J. Reynolds, October 1985
http://www.ietf.org/rfc/rfc959.txt
http://www.w3.org/Protocols/rfc959/

[6] IETF / W3 Network Working Group
RFC 2228: FTP Security Extensions
M. Horowitz, Cygnus Solutions; S. Lunt, Bellcore; October 1997
http://www.ietf.org/rfc/rfc2228.txt

[7] IETF / W3 Network Working Group
RFC 2389: Feature Negotiation of File Transfer Protocol
P. Hethmon, Hethmon Brothers; R. Elz, University of Melbourne; August 1998
http://www.ietf.org/rfc/rfc2389.txt

[8] IETF / W3 FTP Ext Working Group
RFC2428: FTP Extensions for IPv6 and NATs
Mark Allman, NASA Lewis/Sterling Software; Shawn Ostermann, Ohio University; Craig Metz, The Inner Net, May 1998
http://tools.ietf.org/html/draft-ietf-ftpext-ftp-over-ipv6-02
http://tools.ietf.org/html/rfc2428

[9] Open GRID Forum
GFD.20: GRIDFTP: Protocol Extensions to FTP for the Grid.
W. Allcock, Editor, Argonne National Laboratory, April 2003
http://www.ggf.org/documents/GFD.20.pdf

[10] Open GRID Forum
GFD.21: GRIDFTP Protocol Improvements
Igor Mandrichenko, FNAL, July 2003.
http://www-isd.fnal.gov/gridftp-wg/
http://www.ggf.org/documents/GFD.21.pdf

[11] Open GRID Forum
GFD.47: GRIDFTP v2 Protocol Description
I. Mandrichenko, FNAL; W. Allcock, ANL; T.Perelmutov, FNAL, May 2005
http://www.ggf.org/documents/GFD.47.pdf

[12] Shunra Software Ltd
Virtual Enterprise Network Appliance
http://www.shunra.com

[13] Globus Alliance
Globus-url-copy Multi-protocol Data Movement; command line parameters
http://www.globus.org/toolkit/docs/4.0/data/gridftp/rn01re01.html

[14] Globus Alliance
Globus GRIDFTP Tutorials
http://www.globus.org/toolkit/data/gridftp/tutorials/

[15] NeSC GT4 GRIDFTP For Users
http://www.nesc.ac.uk/talks/519/tue/3-GridFTP4Users.ppt
http://www.mcs.anl.gov/~kettimut/tutorials/NeSC05GridFTP4Users.pdf

[16] High Performance File Transfer over IP Networks
EBU TECHNICAL REVIEW – 2009 Q4
P. Brightwell, BBC Research & Development, November 2009.
http://tech.ebu.ch/docs/techreview/trev_2009-Q4_IP-Networks_Brightwell.pdf