
Determining When to Use UDT (Sterling File Accelerator) for Connect:Direct for UNIX

Sterling File Accelerator offers a UDP-based Data Transport (UDT) option for Connect:Direct for UNIX 4.0.00 or later. This document summarizes when UDT should be used instead of TCP, based on experimental tests conducted by Sterling Commerce.

Introduction

UDT is an alternative transport layer protocol to TCP that is designed for high latency with high-bandwidth connections. Over this type of connection, UDT achieves higher data transfer rates than TCP. For more information about UDT, refer to <http://udt.sourceforge.net>.

High-latency connections have always existed, but high-latency, high-bandwidth connections are a more recent development. One situation in which this type of connection may occur is the connection between geographically distant data centers. In this case, latencies can be 50ms or more, and bandwidths may fall in the Gbps range.

When to Use UDT

The following table summarizes which of UDT and TCP achieves the greater data transfer rate over a range of round-trip latencies and bandwidths. The results apply to a connection with little or no packet loss.

Bandwidth (Mbps)	Round-trip Latency (ms)			
	0	10	20	30 and above
45	TCP			
155	TCP		UDT	

Bandwidth (Mbps)	Round-trip Latency (ms)			
	0	10	20	30 and above
1,000	TCP	UDT		

At zero latency, TCP is better than or equal to UDT at all bandwidths. At 45 Mbps, UDT and TCP achieve the same data transfer rate. At 155 Mbps, UDT outperforms TCP above 20ms of latency, and at 1Gbps, UDT outperforms TCP above 10ms of latency.

CPU Usage

A UDT session consumes approximately twice the CPU resources as a TCP session. This is an important consideration on a CPU-bound system.

Measuring Latency and Bandwidth

Round-trip latency is reported by many implementations of the “ping” command. Bandwidth was determined using the Iperf tool (<http://sourceforge.net/projects/iperf>).

Using Iperf

Iperf must be installed on the systems on both ends of the connection that will be instrumented. On one system, Iperf is launched as a server with the following command:

```
iperf -s
```

On the other system, Iperf is run as a client using the following command to communicate with the Iperf server:

```
iperf -c <server system address>
```

Below are sample Iperf runs from the two ends of a connection. The endpoints have hostnames "NodeA" and "NodeB" and IP addresses 10.20.43.12 and 10.20.43.11, respectively. For each endpoint, the command used to launch Iperf and the output from Iperf are shown.

```

NodeA# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 10.20.43.12 port 5001 connected with 10.20.43.11 port 55819
[ 4] 0.0- 9.4 sec   900 MBytes   807 Mbits/sec

NodeB# iperf -c NodeA
-----
Client connecting to arslles-ps, TCP port 5001
TCP window size: 16.0 KByte (default)
-----
[ 3] local 10.20.43.11 port 55819 connected with 10.20.43.12 port 5001
[ 3] 0.0-10.0 sec   900 MBytes   807 Mbits/sec

```

In this case, Iperf has determined that the bandwidth of the connection is 807 Mbits/sec.

Experimental Results

The following results were obtained with two instances of Connect:Direct for UNIX version 4.0 running on Linux systems separated by an IP Impairment device. The experiments consisted of copying a 5GB file between the two systems.

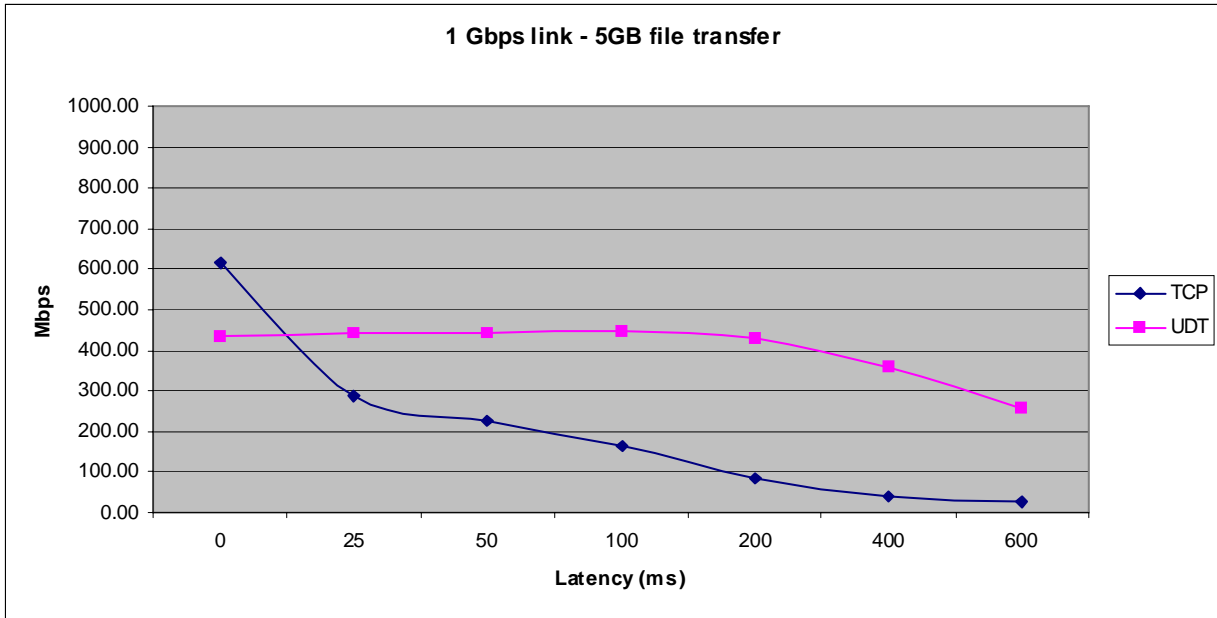
Test Systems

Both test systems ran the SuSE Enterprise 10 OS. The sending system had one dual-core, 2.1 GHz AMD Opteron 8218 CPU and 2G of memory. The receiving system had two 2.4 GHz Intel Xeon CPUs and 3.7G of memory.

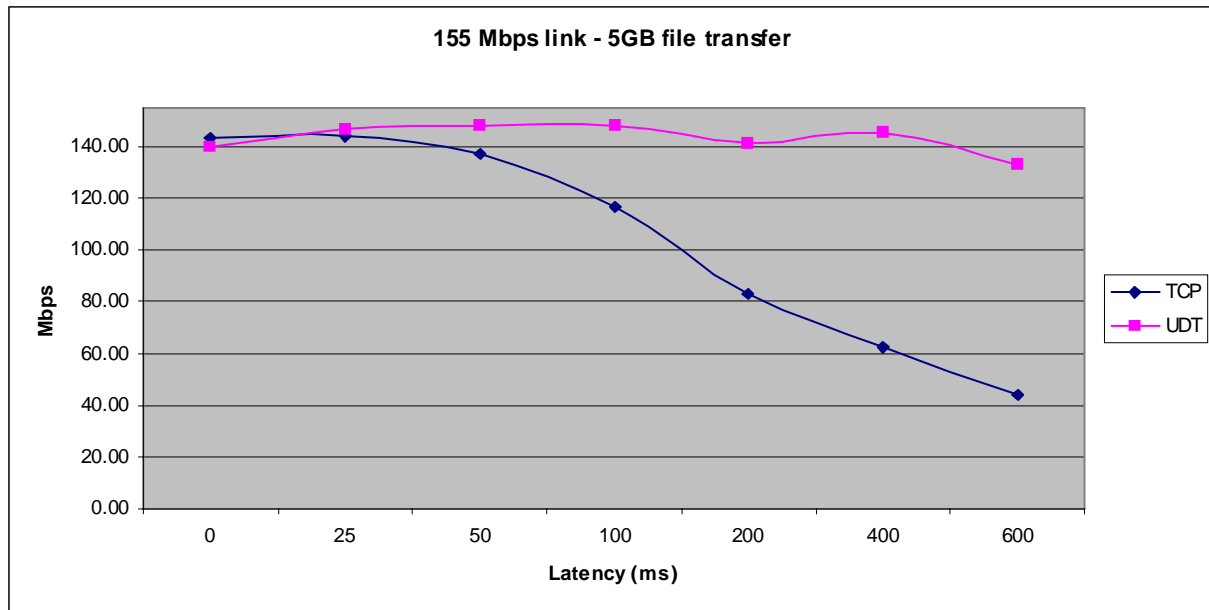
TCP Tuning

TCP often requires a non-default configuration to function effectively in the presence of latency. In the case of systems that support TCP Windows Size auto-tuning, such as the more recent releases of Linux, the maximum TCP Window Size must be set to a value that is at least the Bandwidth Delay Product of the connection. All the experiments were performed with the TCP Windows Size tuned appropriately.

1 Gbps Bandwidth



155 Mbps Bandwidth



45 Mbps Bandwidth

