Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 18, 2010

B. Constantine
JDSU
G. Forget
Bell Canada (Ext. Consultant)
L. Jorgenson
Apparent Networks
Reinhard Schrage
Schrage Consulting
June 8, 2010

TCP Throughput Testing Methodology
draft-ietf-ippm-tcp-throughput-tm-03.txt

Abstract

This memo describes a methodology for measuring sustained TCP throughput performance in an end-to-end managed network environment. This memo is intended to provide a practical approach to help users validate the TCP layer performance of a managed network, which should provide a better indication of end-user application level experience. In the methodology, various TCP and network parameters are identified that should be tested as part of the network verification at the TCP layer.

Status of this Memo

Table of Contents

1. Introduction

Even though RFC2544 was meant to benchmark network equipment and used by network equipment manufacturers (NEMs), network providers have used it to benchmark operational networks in order to verify SLAs (Service Level Agreements) before turning on a service to their business customers. Testing an operational network prior to customer activation is referred to as "turn-up" testing and the SLA is generally Layer 2/3 packet throughput, delay, loss and jitter.

Network providers are coming to the realization that RFC2544 testing and TCP layer testing are required to more adequately ensure end-user satisfaction. Therefore, the network provider community desires to measure network throughput performance at the TCP layer. Measuring TCP throughput provides a meaningful measure with respect to the end user's application SLA (and ultimately reach some level of TCP testing interoperability which does not exist today).

The complexity of the network grows and the various queuing mechanisms in the network greatly affect TCP layer performance (i.e. improper default router settings for queuing, etc.) and devices such as firewalls, proxies, load-balancers can actively alter the TCP settings as a TCP session traverses the network (such as window size, MSS, etc.). Network providers (and NEMs) are wrestling with end-end complexities of the above and there is a strong interest in the standardization of a test methodology to validate end-to-end TCP performance (as this is the precursor to acceptable end-user application performance).

So the intent behind this draft TCP throughput work is to define a methodology for testing sustained TCP layer performance. In this document, sustained TCP throughput is that amount of data per unit time that TCP transports during equilibrium (steady state), i.e. after the initial slow start phase. We refer to this state as TCP Equilibrium, and that the equalibrium throughput is the maximum achievable for the TCP connection(s).

One other important note; the precursor to conducting the TCP tests test methodlogy is to perform "network stress tests" such as RFC2544 Layer 2/3 tests or other conventional tests (OWAMP, etc.). It is highly recommended to run traditional Layer 2/3 type test to verify the integrity of the network before conducting TCP testing.

2. Goals of this Methodology

Before defining the goals of this methodology, it is important to clearly define the areas that are not intended to be measured or analyzed by such a methodology.

- The methodology is not intended to predict TCP throughput behavior during the transient stages of a TCP connection, such as initial slow start.

- The methodology is not intended to definitively benchmark TCP implementations of one OS to another, although some users may find some value in conducting qualitative experiments

- The methodology is not intended to provide detailed diagnosis of problems within end-points or the network itself as related to non-optimal TCP performance, although a results interpretation section for each test step may provide insight into potential issues within the network

In contrast to the above exclusions, the goals of this methodology are to define a method to conduct a structured, end-to-end assessment of sustained TCP performance within a managed business class IP network. A key goal is to establish a set of "best practices" that an engineer should apply when validating the ability of a managed network to carry end-user TCP applications.

Some specific goals are to:

- Provide a practical test approach that specifies the more well understood (and end-user configurable) TCP parameters such as Window size, MSS, # connections, and how these affect the outcome of TCP performance over a network

- Provide specific test conditions (link speed, RTT, window size, etc.) and maximum achievable TCP throughput under TCP Equilbrium conditions. For guideline purposes, provide examples of these test conditions and the maximum achievable TCP throughput during the equilbrium state. Section 2.1 provides specific details concerning the definition of TCP Equilibrium within the context of this draft.

- In test situations where the recommended procedure does not yield the maximum achievable TCP throughput result, this draft provides some possible areas within the end host or network that should be considered for investigation (although again, this draft is not intended to provide a detailed diagnosis of these issues)

2.1 TCP Equilibrium State Throughput

TCP connections have three (3) fundamental congestion window phases as documented in RFC2581. These states are:

- Slow Start, which occurs during the beginning of a TCP transmission or after a retransmission time out event

- Congestion avoidance, which is the phase during which TCP ramps up to establish the maximum attainable throughput on an end-end network path. Retransmissions are a natural by-product of the TCP congestion avoidance algorithm as it seeks to achieve maximum throughput on the network path.

- Retransmission phase, which include Fast Retransmit (Tahoe) and Fast Recovery (Reno and New Reno). When a packet is lost, the Congestion avoidance phase transitions to a Fast Retransmission or Recovery Phase dependent upon the TCP implementation.

The following diagram depicts these states.

```
         |        ssthresh
TCP      |           |
Through- |           |       Equilibrium
put      |           |\      /\/\/\/\/\  Retransmit          /\/\ ...
         |           | \    /         |  Time-out           /
         |           |  \  /          |  _______          _/
         |  Slow   _/    |/           | /       | Slow  _/
         | Start _/      Congestion   |/        |Start_/   Congestion
         |     _/         Avoidance   Loss      |   _/     Avoidance
         |   _/                       Event     | _/
         | _/                                   |/
         |/___________________________________________________________
                                                         Time
```

This TCP methodology provides guidelines to measure the equilibrium throughput which refers to the maximum sustained rate obtained by congestion avoidance before packet loss conditions occur (which would cause the state change from congestion avoidance to a retransmission phase). All maximum achievable throughputs specified in Section 3 are with respect to this Equilibrium state.

2.2 Metrics for TCP Throughput Tests

This draft focuses on a TCP throughtput methodology and also provides two basic metrics to compare results of various throughput tests. It is recognized that the complexity and unpredictability of TCP makes it impossible to develop a set of metrics that account for the myriad of variables (i.e. RTT variation, loss conditions, TCP implementation, etc.). However, these two basic metrics are useful to compare network traffic management techniques, especially in section 3.4 of this document (Traffic Management Tests).

The TCP Efficiency metric is the percentage of bytes that were not retransmitted and is defined as:

```
             Transmitted Bytes - Retransmitted Bytes
             ---------------------------------------  x 100
                        Transmitted Bytes
```

This metric is easy to understand and provides a comparative measure between various QoS mechanisms such as traffic management, congestion avoidance, and also various TCP implementations (i.e. Reno, Vegas, etc.).

The second measure is also basic and is the TCP Transfer Time, which is simply the time is takes to transfer a block of data across simultaneous TCP connections. The concept is useful to benchmark traffic management tests, where multiple connections are required and it simplifies comparing results of different approaches. An example would be the bulk transfer of 10 MB upon 8 separate TCP connections (each connection uploading 10 MB). Each connection may achieve different throughputs during a test and the overall throughput rate is not always easy to determine (especially as the number of connections increases). But by defining the Transfer Time as that of the successful transfer of 10MB over all 8 connections, the single transfer time metric is very useful means to rate various traffic management techniques (i.e. FiFO, WFQ queuing, WRED, etc.).

3. TCP Throughput Testing Methodology

This section summarizes the specific test methodology to achieve the goals listed in Section 2.

As stated in Section 1, it is considered best practice to verify the integrity of the network by conducting Layer2/3 stress tests such as RFC2544 or other methods of network stress tests. If the network is not performing properly in terms of packet loss, jitter, etc. then the TCP layer testing will not be meaningful since the equalibrium throughput would be very difficult to achieve (in a "dysfunctional" network).

The following provides the sequential order of steps to conduct the TCP throughput testing methodology:

1. Identify the Path MTU. Packetization Layer Path MTU Discovery or PLPMTUD (RFC4821) should be conducted to verify the minimum network path MTU. Conducting PLPMTUD establishes the upper limit for the MSS to be used in subsequent steps.

2. Baseline Round-trip Delay and Bandwidth. These measurements provide estimates of the ideal TCP window size, which will be used in subsequent test steps.

3. Single TCP Connection Throughput Tests. With baseline measurements of round trip delay and bandwidth, a series of single connection TCP throughput tests can be conducted to baseline the performance of the network against expectations.

4. Traffic Management Tests. Various traffic management and queuing techniques are tested in this step, using multiple TCP connections. Multiple connection testing can verify that the network is configured properly for traffic shaping versus policing, various queuing implementations, and RED.

Important to note are some of the key characteristics and considerations for the TCP test instrument. The test host may be a standard computer or dedicated communications test instrument and these TCP test hosts be capable of emulating both a client and a server. As a general rule of thumb, testing TCP throughput at rates greater than 250-500 Mbit/sec generally requires high performance server hardware or dedicated hardware based test tools.

Whether the TCP test host is a standard computer or dedicated test instrument, the following areas should be considered when selecting a test host:

- TCP implementation used by the test host OS, i.e. Linux OS kernel using TCP Reno, TCP options supported, etc. This will obviously be more important when using custom test equipment where the TCP implementation may be customized or tuned to run in higher performance hardware

- Most importantly, the TCP test host must be capable of generating and receiving stateful TCP test traffic at the full link speed of the network under test. This requirement is very serious and may require custom test equipment, especially on 1 GigE and 10 GigE networks.

3.1. Determine Network Path MTU

TCP implementations should use Path MTU Discovery techniques (PMTUD). PMTUD relies on ICMP 'need to frag' messages to learn the path MTU. When a device has a packet to send which has the Don't Fragment (DF) bit in the IP header set and the packet is larger than the Maximum Transmission Unit (MTU) of the next hop link, the packet is dropped and the device sends an ICMP 'need to frag' message back to the host that originated the packet. The ICMP 'need to frag' message includes the next hop MTU which PMTUD uses to tune the TCP Maximum Segment Size (MSS). Unfortunately, because many network managers completely disable ICMP, this technique does not always prove reliable in real world situations.

Packetization Layer Path MTU Discovery or PLPMTUD (RFC4821) should be conducted to verify the minimum network path MTU. PLPMTUD can be used with or without ICMP. The following sections provide a summary of the PLPMTUD approach and an example using the TCP protocol. RFC4821 specifies a search_high and search_low parameter for the MTU. As specified in RFC4821, a value of 1024 is a generally safe value to choose for search_low in modern networks.

It is important to determine the overhead of the links in the path, and then to select a TCP MSS size corresponding to the Layer 3 MTU. For example, if the MTU is 1024 bytes and the TCP/IP headers are 40 bytes, then the MSS would be set to 984 bytes.

An example scenario is a network where the actual path MTU is 1240 bytes. The TCP client probe MUST be capable of setting the MSS for the probe packets and could start at MSS = 984 (which corresponds to an MTU size of 1024 bytes).

The TCP client probe would open a TCP connection and advertise the MSS as 984. Note that the client probe MUST generate these packets with the DF bit set. The TCP client probe then sends test traffic per a nominal window size (8KB, etc.). The window size should be kept small to minimize the possibility of congesting the network, which could induce congestive loss. The duration of the test should also be short (10-30 seconds), again to minimize congestive effects during the test.

In the example of a 1240 byte path MTU, probing with an MSS equal to 984 would yield a successful probe and the test client packets would be successfully transferred to the test server.

Also note that the test client MUST verify that the MSS advertised is indeed negotiated. Network devices with built-in Layer 4 capabilities can intercede during the connection establishment process and reduce the advertised MSS to avoid fragmentation. This is certainly a desirable feature from a network perspective, but can yield erroneous test results if the client test probe does not confirm the negotiated MSS.

The next test probe would use the search_high value and this would be set to MSS = 1460 to correspond to a 1500 byte MTU. In this example, the test client would retransmit based upon time-outs (since no ACKs will be received from the test server). This test probe is marked as a conclusive failure if none of the test packets are ACK'ed. If any of the test packets are ACK'ed, congestive network may be the cause and the test probe is not conclusive. Re-testing at other times of the day is recommended to further isolate.

The test is repeated until the desired granularity of the MTU is discovered. The method can yield precise results at the expense of probing time. One approach would be to reduce the probe size to half between the unsuccessful search_high and successful search_low value, and increase by increments of 1/2 when seeking the upper limit.

3.2. Baseline Round-trip Delay and Bandwidth

Before stateful TCP testing can begin, it is important to baseline the round trip delay and bandwidth of the network to be tested. These measurements provide estimates of the ideal TCP window size, which will be used in subsequent test steps. These latency and bandwidth tests should be run over a long enough period of time to characterize the performance of the network over the course of a meaningful time period.

One example would be to take samples during various times of the work day. The goal would be to determine a representative minimum, average, and maximum RTD and bandwidth for the network under test. Topology changes are to be avoided during this time of initial convergence (e.g. in crossing BGP4 boundaries).

In some cases, baselining bandwidth may not be required, since a network provider's end-to-end topology may be well enough defined.

3.2.1 Techniques to Measure Round Trip Time

We follow in the definitions used in the references of the appendix; hence Round Trip Time (RTT) is the time elapsed between the clocking in of the first bit of a payload packet to the receipt of the last bit of the corresponding acknowledgement. Round Trip Delay (RTD) is used synonymously to twice the Link Latency.

In any method used to baseline round trip delay between network end-points, it is important to realize that network latency is the sum of inherent network delay and congestion. The RTT should be baselined during "off-peak" hours to obtain a reliable figure for network latency (versus additional delay caused by congestion).

During the actual sustained TCP throughput tests, it is critical to measure RTT along with measured TCP throughput. Congestive effects can be isolated if RTT is concurrently measured

This is not meant to provide an exhaustive list, but summarizes some of the more common ways to determine round trip time (RTT) through the network. The desired resolution of the measurement (i.e. msec versus usec) may dictate whether the RTT measurement can be achieved with standard tools such as ICMP ping techniques or whether specialized test equipment would be required with high precision timers. The objective in this section is to list several techniques in order of decreasing accuracy.

- Use test equipment on each end of the network, "looping" the far-end tester so that a packet stream can be measured end-end. This test equipment RTT measurement may be compatible with delay measurement protocols specified in RFC5357.

- Conduct packet captures of TCP test applications using for example "iperf" or FTP, etc. By running multiple experiments, the packet captures can be studied to estimate RTT based upon the SYN -> SYN-ACK handshakes within the TCP connection set-up.

- ICMP Pings may also be adequate to provide round trip time estimations. Some limitations of ICMP Ping are the msec resolution and whether the network elements respond to pings (or block them).

3.2.2 Techniques to Measure End-end Bandwidth

There are many well established techniques available to provide estimated measures of bandwidth over a network. This measurement should be conducted in both directions of the network, especially for access networks which are inherently asymmetrical. Some of the asymmetric implications to TCP performance are documented in RFC-3449 and the results of this work will be further studied to determine relevance to this draft.

The bandwidth measurement test must be run with stateless IP streams (not stateful TCP) in order to determine the available bandwidth in each direction. And this test should obviously be performed at various intervals throughout a business day (or even across a week). Ideally, the bandwidth test should produce a log output of the bandwidth achieved across the test interval AND the round trip delay.

And during the actual TCP level performance measurements (Sections 3.3 - 3.5), the test tool must be able to track round trip time of the TCP connection(s) during the test. Measuring round trip time variation (aka "jitter") provides insight into effects of congestive delay on the sustained throughput achieved for the TCP layer test.

3.3. Single TCP Connection Throughput Tests

This draft specifically defines TCP throughput techniques to verify sustained TCP performance in a managed business network. Defined in section 2.1, the equalibrium throughput reflects the maximum rate achieved by a TCP connection within the congestion avoidance phase on a end-end network path. This section and others will define the method to conduct these sustained throughput tests and guidelines of the predicted results.

With baseline measurements of round trip time and bandwidth from section 3.2, a series of single connection TCP throughput tests can be conducted to baseline the performance of the network against expectations. The optimum TCP window size can be calculated from the bandwidth delay product (BDP), which is:

BDP = RTT x Bandwidth

By dividing the BDP by 8, the "ideal" TCP window size is calculated. An example would be a T3 link with 25 msec RTT. The BDP would equal ~1,105,000 bits and the ideal TCP window would equal ~138,000 bytes.

The following table provides some representative network link speeds, latency, BDP, and associated "optimum" TCP window size. Sustained TCP transfers should reach nearly 100% throughput, minus the overhead of Layers 1-3 and the divisor of the MSS into the window.

For this single connection baseline test, the MSS size will effect the achieved throughput (especially for smaller TCP window sizes). Table 3.2 provides the achievable, equalibrium TCP throughput (at Layer 4) using 1000 byte MSS. Also in this table, the case of 58 byte L1-L4 overhead including the Ethernet CRC32 is used for simplicity.

Table 3.2: Link Speed, RTT and calculated BDP, TCP Throughput

Link Speed*	RTT (ms)	BDP (bits)	Ideal TCP Window (kbytes)	Maximum Achievable TCP Throughput(Mbps)
T1	20	30,720	3.84	1.20
T1	50	76,800	9.60	1.44
T1	100	153,600	19.20	1.44
T3	10	442,100	55.26	41.60
T3	15	663,150	82.89	41.13
T3	25	1,105,250	138.16	41.92
T3(ATM)	10	407,040	50.88	32.44
T3(ATM)	15	610,560	76.32	32.44
T3(ATM)	25	1,017,600	127.20	32.44
100M	1	100,000	12.50	90.699

100M	2	200,000	25.00	92.815

Link Speed*	RTT (ms)	BDP (bits)	Ideal TCP Window (kbytes)	Maximum Achievable TCP Throughput (Mbps)
100M	5	500,000	62.50	90.699
1Gig	0.1	100,000	12.50	906.991
1Gig	0.5	500,000	62.50	906.991
1Gig	1	1,000,000	125.00	906.991
10Gig	0.05	500,000	62.50	9,069.912
10Gig	0.3	3,000,000	375.00	9,069.912

* Note that link speed is the minimum link speed throughput a network; i.e. WAN with T1 link, etc.

Also, the following link speeds (available payload bandwidth) were used for the WAN entries:

- T1 = 1.536 Mbits/sec (B8ZS line encoding facility)
- T3 = 44.21 Mbits/sec (C-Bit Framing)
- T3(ATM) = 36.86 Mbits/sec (C-Bit Framing & PLCP, 96000 Cells per second)

The calculation method used in this document is a 3 step process :

1 - We determine what should be the optimal TCP Window size value based on the optimal quantity of "in-flight" octets discovered by the BDP calculation. We take into consideration that the TCP Window size has to be an exact multiple value of the MSS.
2 - Then we calculate the achievable layer 2 throughput by multiplying the value determined in step 1 with the MSS & (MSS + L2 + L3 + L4 Overheads) divided by the RTT.
3 - Finally, we multiply the calculated value of step 2 by the MSS versus (MSS + L2 + L3 + L4 Overheads) ratio.

This gives us the achievable TCP Throughput value. Sometimes, the maximum achievable throughput is limited by the maximum achievable quantity of Ethernet Frames per second on the physical media. Then this value is used in step 2 instead of the calculated one.

There are several TCP tools that are commonly used in the network provider world and one of the most common is the "iperf" tool. With this tool, hosts are installed at each end of the network segment; one as client and the other as server. The TCP Window size of both the client and the server can be maunally set and the achieved throughput is measured, either uni-directionally or bi-directionally. For higher BDP situations in lossy networks (long fat networks or satellite links, etc.), TCP options such as Selective Acknowledgment should be considered and also become part of the window size / throughput characterization.

The following diagram shows the achievable TCP throughput on a T3 with the default Windows2000/XP TCP Window size of 17520 Bytes.

```
           45|
             |
           40|
TCP          |
Throughput 35|
in Mbps      |
           30|
             |
           25|
             |
           20|
             |
           15|         _______ 14.48M
             |         |     |
           10|         |     |         +-----+ 9.65M
             |         |     |         |     |        _______ 5.79M
            5|         |     |         |     |        |     |
             |_________+_____+_________+_____+________+____ +___________
                          10              15             25
                                RTT in milliseconds
```

The following diagram shows the achievable TCP throughput on a 25ms T3 when the TCP Window size is increased and with the RFC1323 TCP Window scaling option.

```
           45|
             |                                              +-----+42.47M
           40|                                              |     |
TCP          |                                              |     |
Throughput 35|                                              |     |
in Mbps      |                                              |     |
           30|                                              |     |
             |                                              |     |
           25|                                              |     |
             |                                ______ 21.23M |     |
           20|                                |    |        |     |
             |                                |    |        |     |
           15|                                |    |        |     |
             |                                |    |        |     |
           10|               +----+10.62M     |    |        |     |
             |  _______5.31M |    |           |    |        |     |
            5|  |     |      |    |           |    |        |     |
             |__+_____+______+____+___________+____+________+_____+___
                     16           32           64              128
                              TCP Window size in Kilo Bytes
```

The single connection TCP throughput test must be run over a a long duration and results must be logged at the desired interval. The test must record RTT and TCP retransmissions at each interval.

This correlation of retransmissions and RTT over the course of the test will clearly identify which portions of the transfer reached TCP Equilbrium state and to what effect increased RTT (congestive effects) may have been the cause of reduced equilibrium performance.

Host hardware performance must be well understood before conducting this TCP single connection test and other tests in this section. Dedicated test equipment may be required, especially for line rates of GigE and 10 GigE.

3.3.1 Interpretation of the Single Connection TCP Throughput Results

At the end of this step, the user will document the theoretical BDP and a set of Window size experiments with measured TCP throughput for each TCP window size setting. For cases where the sustained TCP throughput does not equal the predicted value, some possible causes are listed:

- Network congestion causing packet loss
- Network congestion not causing packet loss, but effectively increasing the size of the required TCP window during the transfer
- Intermediate network devices which actively regenerate the TCP connection and can alter window size, MSS, etc.

3.4. Traffic Management Tests

After baselining the network under test with a single TCP connection (Section 3.3), the nominal capacity of the network has been determined. The capacity measured in section 3.3 may be a capacity range and it is reasonable that some level of tuning may have been required (i.e. router shaping techniques employed, intermediary proxy like devices tuned, etc.).

Single connection TCP testing is a useful first step to measure expected versus actual TCP performance and as a means to diagnose / tune issues in the network and active elements. However, the ultimate goal of this methodology is to more closely emulate customer traffic, which comprise many TCP connections over a network link.

3.4.1 Multiple TCP Connections - below Link Capacity

First, the ability of the network to carry multiple TCP connections to full network capacity should be tested. Prioritization and QoS settings are not considered during this step, since the network capacity is not to be exceeded by the test traffic (section 3.5.2 covers the over capacity test case).

For this multiple connection TCP throughput test, the number of connections will more than likely be limited by the test tool (host vs. dedicated test equipment). As an example, for a GigE link with 1 msec RTT, the optimum TCP window would equal ~128 KBytes. So under this condition, 8 concurrent connections with window size equal to 16KB would fill the GigE link. For 10G, 80 connections would be required to accomplish the same.

Just as in section 3.3, the end host or test tool can not be the processing bottleneck or the throughput measurements will not be valid. The test tool must be benchmarked in ideal lab conditions to verify it's ability to transfer stateful TCP traffic at the given network line rate.

For this test step, it should be conducted over a reasonable test duration and results should be logged per interval such as throughput per connection, RTT, and retransmissions.

Since the network is not to be driven into over capacity (by nature of the BDP allocated evenly to each connection), this test verifies the ability of the network to carry multiple TCP connections up to the link speed of the network.

3.4.2 Multiple TCP Connections - over Link Capacity

In this step, the network bandwidth is intentionally exceeded with multiple TCP connections to test expected prioritization and queuing within the network.

All conditions related to Section 3.3 set-up apply, especially the ability of the test hosts to transfer stateful TCP traffic at network line rates.

Using the same example from Section 3.3, a GigE link with 1 msec RTT would require a window size of 128 KB to fill the link (with one TCP connection). Assuming a 16KB window, 8 concurrent connections would fill the GigE link capacity and values higher than 8 would over-subscribe the network capacity. The user would select values to over-subscribe the network (i.e. possibly 10 15, 20, etc.) to conduct experiments to verify proper prioritization and queuing within the network.

3.4.3 Interpretation of Multiple TCP Connection Test Restults

Without any prioritization in the network, the over subscribed test results could assist in the queuing studies. With proper queuing, the bandwidth should be shared in a reasonable manner. The author understands that the term "reasonable" is too wide open, and future draft versions of this memo would attempt to quantify this sharing in more tangible terms. It is known that if a network element is not set for proper queuing (i.e. FIFO), then an oversubscribed TCP connection test will generally show a very uneven distribution of bandwidth.

With prioritization in the network, different TCP connections can be assigned various QoS settings via the various mechanisms (i.e. per VLAN, DSCP, etc.), and the higher priority connections must be verified to achieve the expected throughput.

4. Acknowledgements

The author would like to thank Gilles Forget, Loki Jorgenson, and Reinhard Schrage for technical review and contributions to this draft-03 memo.

Also thanks to Matt Mathis and Matt Zekauskas for many good comments through email exchange and for pointing us to great sources of information pertaining to past works in the TCP capacity area.

5. References

[RFC2581] Allman, M., Paxson, V., Stevens W., "TCP Congestion Control", RFC 2581, June 1999.

[RFC3148] Mathis M., Allman, M., "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2001.

[RFC2544] Bradner, S., McQuaid, J., "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, June 1999

[RFC3449] Balakrishnan, H., Padmanabhan, V. N., Fairhurst, G., Sooriyabandara, M., "TCP Performance Implications of Network Path Asymmetry", RFC 3449, December 2002

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., Babiarz, J., "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008

[RFC4821] Mathis, M., Heffner, J., "Packetization Layer Path MTU Discovery", RFC 4821, June 2007

draft-ietf-ippm-btc-cap-00.txt Allman, M., "A Bulk Transfer Capacity Methodology for Cooperating Hosts", August 2001

[MSMO]   The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm Mathis, M.,Semke, J, Mahdavi, J, Ott, T July 1997 SIGCOMM Computer Communication Review, Volume 27 Issue 3

[Stevens Vol1] TCP/IP Illustrated, Vol1, The Protocols Addison-Wesley

Authors' Addresses

Barry Constantine
JDSU, Test and Measurement Division
One Milesone Center Court
Germantown, MD 20876-7100
USA

Phone: +1 240 404 2227
Email: barry.constantine@jdsu.com

Gilles Forget
Independent Consultant to Bell Canada.
308, rue de Monaco, St-Eustache
Qc. CANADA, Postal Code : J7P-4T5

Phone: (514) 895-8212
gilles.forget@sympatico.ca

Loki Jorgenson
Apparent Networks

Phone: (604) 433-2333 ext 105
ljorgenson@apparentnetworks.com

Reinhard Schrage
Schrage Consulting

Phone: +49 (0) 5137 909540
reinhard@schrageconsult.com