Network Working Group B. Constantine, Ed.
Internet-Draft JDSU
Intended status: Informational G. Forget
Expires: August 21, 2010 Bell Canada (Ext. Consultant)
L. Jorgenson
Apparent Networks
Reinhard Schrage
Schrage Consulting
Feb 21, 2010

TCP Throughput Testing Methodology
draft-constantine-ippm-tcp-throughput-tm-01

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

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Creation date Feb 21, 2010.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on Septmber 19, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

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 provide SLAs (Service Level Agreements) to their business customers. 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).

Before RFC2544 testing existed, network providers and NEMs deployed a variety of ad hoc test techniques to verify the Layer 2/3 performance of the network. RFC2544 was a huge step forward in the network test world, standardizing the Layer 2/3 test methodology which greatly improved the quality of the network and reduced operational test expenses. These managed networks are intended to be predictable, but therein lies the problem. It is difficult if not impossible, to extrapolate end user application layer performance from RFC2544 results and the goal of RFC2544 was never intended to do so.

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 RFC2544 related Layer 2/3 tests. It is highly recommended to run traditional RFC2544 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, RTD, 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)

- Testing end-end prioritization of services is a key goal. This draft proposes the use of stateful TCP connections in the midst of stateless background traffic such as UDP which is a very common service condition in managed provider networks. The ability to test stateful TCP with stateless traffic (with proper prioritization for each) is a more thorough and realistic test than simply testing with only stateless traffic. Further more, many networks will not tolerate TCP "traffic blasting" to emulate the TCP application traffic.

2.1 TCP Equilibrium State Throughput

TCP connections have three (3) fundamental congestion window phases as documented in RFC-TBD. 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.

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 from a Layer2/3 perspective by first conducting RFC2544 type testing. If the network is not performing properly in terms of packet loss, jitter, etc. when running RFC2544 tests, 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. Baseline Round-trip Delay and Bandwidth. These measurements provide estimates of the ideal TCP window size, which will be used in subsequent test steps.

2. 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.

3. TCP MSS Throughput Testing. By varying the MSS size of the TCP connection, the ability of the network to sustain expected TCP throughput can be verified.

4. Multiple TCP Connection Throughput Tests. Single connection TCP testing is a useful first step to measure expected versus actual TCP performance. The multiple connection test more closely emulates customer traffic, which comprise many TCP connections over a network link.

5. TCP Sessions with Stateless Background Traffic. This step assesses the ability of the network to carry end user TCP application traffic concurrent with stateless background traffic. The prioritization must be configured properly throughout the network to allow either the TCP traffic or the stateless traffic to receive the expected priority.

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.

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. 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.

3.1.1 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 / end points respond to pings (or block them).

3.1.2 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 3.2 - 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.

3.2. 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 3.1, 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-L3 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 Overheads) divided by the RTT.
3 - Finally, we multiply the calculated value of step 2 by the MSS versus (MSS + L2 & L3 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
```

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.

3.2.1 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
- Network fragmentation at the IP layer
- Intermediate network devices which actively regenerate the TCP connection and can alter window size, MSS, etc.

3.3. TCP MSS Throughput Testing

This test setup should be conducted as a single TCP connection test. By varying the MSS size of the TCP connection, the ability of the network to sustain expected TCP throughput can be verified. This is similar to frame and packet size techniques within RFC2-2544, which aim to determine the ability of the routing/switching devices to handle loads in term of packets/frames per second at various frame and packet sizes. This test can also further characterize the performance of a network in the presence of active TCP elements (proxies, etc.), devices that fragment IP packets, and the actual end hosts themselves (servers, etc.).

3.3.1 TCP Test for Network Path MTU

TCP implementations should use Path MTU Discovery techniques (PMTUD), but this technique does not always prove reliable in real world situations. Since PMTUD relies on ICMP messages (to inform the host that unfragmented transmission cannot occur), PMTUD is not always reliable since many network managers completely disable ICMP.

Increasingly network providers and enterprises are instituting fixed MTU sizes on the hosts to eliminate TCP fragmentation issues in the application.

Packetization Layer Path MTU Discovery or PLPMTUD (RFC4821) should be conducted to verify the minimum network path MTU. Conducting the PLPMTUD test establishes the upper limit upon the MTU, which in turn establishes the upper limit for the MSS testing of section 3.3.2. MSS refers specifically to the payload size of the TCP packet and does not include TCP or IP headers.

3.3.2 MSS Size Testing Method

The single connection testing listed in Section 3.2 should be repeated, using the appropriate window size and collecting throughput measurements per various MSS sizes.

The following are the typical sizes of MSS settings for various link speeds:

- 256 bytes for very low speed links such as 9.6Kbps (per RFC1144).
- 536 bytes for low speed links (per RFC879) .
- 966 bytes for SLIP high speed (per RFC1055).
- 1380 bytes for IPSec VPN Tunnel testing
- 1452 bytes for PPPoE connectivity (per RFC2516)
- 1460 for Ethernet and Fast Ethernet (per RFC895).
- 8960 byte jumbo frames for GigE

Using the optimum window size determined by conducting steps 3.1 and 3.2, a variety of window sizes should be tested according to the link speed under test. Using Fast Ethernet with 5 msec RTT as an example, the optimum TCP window size would be 62.5 kbytes and the recommended MSS for Fast Ethernet is 1460 bytes.

Link		Achievable TCP Throughput (Mbps) for					
Speed	RTT(ms)	MSS=1000	MSS=1260	MSS=1300	MSS=1380	MSS=1420	MSS=1460
T1	20	1.20	1.008	1.040	1.104	1.136	1.168
T1	50	1.44	1.411	1.456	1.335	1.363	1.402
T1	100	1.44	1.512	1.456	1.435	1.477	1.402
T3	10	41.60	42.336	42.640	41.952	40.032	42.048
T3	15	42.13	42.336	42.293	42.688	42.411	42.048
T3	25	41.92	42.336	42.432	42.394	42.714	42.515
T3(ATM)	10	32.44	33.815	34.477	35.482	36.022	36.495
T3(ATM)	15	32.44	34.120	34.477	35.820	36.022	36.127
T3(ATM)	25	32.44	34.363	34.860	35.684	36.022	36.274
100M	1	90.699	89.093	91.970	86.866	89.424	91.982
100M	2	92.815	93.226	93.275	88.505	90.973	93.442

100M	5	90.699	92.481	92.697	88.245	90.844	93.442

For GigE and 10GigE, Jumbo frames (9000 bytes) are becoming more common. The following table adds jumbo frames to the possible MSS values.

Link Speed	RTT(ms)	Achievable TCP Throughput (Mbps) for MSS=1260	MSS=1300	MSS=1380	MSS=1420	MSS=1460	MSS=8960
1Gig	0.1	924.812	926.966	882.495	894.240	919.819	713.786
1Gig	0.5	924.812	926.966	930.922	932.743	934.467	856.543
1Gig	1.0	924.812	926.966	930.922	932.743	934.467	927.922
10Gig	0.05	9248.125	9269.655	9309.218	9839.790	9344.671	8565.435
10Gig	0.3	9248.125	9269.655	9309.218	9839.790	9344.671	9755.079

Each row in the table is a separate test that should be conducted over a predetermined test interval and the throughput, retransmissions, and RTT logged during the entire test interval.

3.3.3 Interpretation of TCP MSS Throughput Results

For cases where the predicted TCP throughput does not equal the predicted throughput predicted for a given MSS, some possible causes are listed:

- TBD

3.4. Multiple TCP Connection Throughput Tests

After baselining the network under test with a single TCP connection (Section 3.2), the nominal capacity of the network has been determined. The capacity measured in section 3.2 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. This methodology inevitably seeks to provide the framework for testing stateful TCP connections in concurrence with stateless traffic streams, and this is described in Section 3.5.

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 3.3.2 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 3.2, 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 3.2, 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.

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.

3.5. TCP Sessions with Stateless Background Traffic

The ultimate intent of this methodology is to more accurately assess the ability of the network to carry end user TCP application traffic concurrent with stateless background traffic. The background traffic may be of a lower priority than the TCP traffic (i.e. background is best effort Internet), or the background traffic may be of higher priority (i.e.UDP representing voice). The prioritization must be configured properly throughout the network to allow either the TCP foreground traffic or the stateless background traffic to receive the expected priority.

For this test, each stateful TCP connection must be able to have a unique prioritization. Depending upon the network prioritization scheme (i.e. VLAN, DSCP, MPLS, etc.), the test system must allow for unique identification of each connection. The same applies for the stateless background streams. The individual background streams must also be tagged or identified based upon the prioritization mechanism.

3.5.1. TCP Foreground Traffic Control

In addition to the prioritization of each individual TCP connection, the desired bandwidth for each TCP connection must be configurable. Depending upon the capabilities of the test system, this bandwidth rate may be discretely controlled (or shaped) by the test system or may also be controlled by limiting the window size of each connection. The sophistication of the test system will dictate which bandwidth control mechanism is used for the TCP connections. The window based approach allows the TCP traffic to reach the maximum achievable within the bandwidth capacity of the link (BDP), which may be the intended test configuration. Each TCP foreground connection should also have configurable MSS size as well.

3.5.2 Stateless Background Traffic Control

Each stateless background traffic stream must be configurable in terms of the offered network bandwidth. The frame / packet sizes of the background traffic streams should be individually configurable. Ideally, the test system should allow for the stateless background traffic to ramp up from a configurable starting bandwidth to the final bandwidth setting (i.e. start at 10 Mbps, incrementing by 5 Mbps, until reaching 50 Mbps). The time step of the ramping function should also be programmable. Ramping of the background traffic facilitates the study of the effect of stateless background traffic on foreground TCP traffic.

3.5.3. Test Methodology for TCP + Stateless Background Traffic

Depending upon the prioritization within the provider's network, there are many permutations of prioritization between TCP sessions, background streams, and combinations between. This memo will summarize two common use cases: 1) higher priority TCP stateful traffic in the midst of best effort background traffic; 2) higher priority stateless traffic (i.e. VoIP) in the midst of lower priority TCP stateful traffic.

3.5.3.1. Prioritized Stateful TCP Traffic Test

In this test, the intent is to verify that a business class data service (i.e. thin client, web-based application traffic) is given proper priority in times of high utilization. For this test case, the TCP connections are given priority via the prioritization mechanism used in the network (VLAN, DSCP, etc..) and the stateless background traffic is given lower priority (or best effort).

By one of the traffic control techniques listed in Section 3.5.1, the stateful TCP connections are allocated bandwidth within the test system and ideally the background traffic will perform a ramp traffic function. The results of this test should show that the prioritized stateful TCP traffic reaches the designated throughput and is not disturbed when the best effort background traffic exceeds the link capacity. The test results should be logged during the test interval, with each TCP connection's throughput, retransmissions, and RTT recorded (along with the background traffic levels).

3.5.3.2. Prioritized Stateless Traffic Test

The corollary to section 3.5.3.1 is the case where the stateless traffic is higher priority than the stateful TCP traffic. This may be the case where a network provider is offering VoIP services in addition to regular IP data service. Even though the TCP traffic is

prioritized lower than the stateless traffic, it is important to determine how the TCP traffic reacts in the presence of over subscription (this can again point to non-optimized queuing techniques in the network for the TCP traffic as discussed in Section 3.3.2.).

With the TCP offered bandwidth set by one of the traffic control mechanisms listed in Section 3.5.1., the higher priority stateless traffic should be ramped up to exceed the link capacity. The results of this test should show that the higher priority stateless traffic achieves the designated bandwidth and that the TCP connection bandwidth is reduced. By studying the logged TCP and stateless traffic throughput over the test interval (and the retransmissions + RTT for the TCP traffic), the manner in which the TCP connections shared the remaining bandwidth may provide insight into possible queuing optimizations in the network.

3.5.3.3. Other Traffic Test Cases

Sections 3.5.3.2 and 3.5.3.3 lay out the basic foundation for testing the prioritization effects of TCP traffic and background stateless traffic. There are many hybrids that can also be pertinent, dependant upon the network provider's offering. An example would be strictly an IP service type test. In this case, the network provider seeks to test various prioritizations of each stateful TCP connection and verify that the higher priority TCP connection(s) achieve the SLA bandwidth while the others do not. Regardless of the prioritization profile of the TCP connections and the background streams, the same test results should be recorded across the entire test interval (as specified in Section 3.5.3.1 and 3.5.3.2)

4. Acknowledgements

The author would like to thank Gilles Forget, Mike Hamilton, and Reinhard Schrage for technical review and contributions to this draft-00 memo.

Also thanks to Matt Mathis and Matt Zekauskas for many good comments through email exchange and for pointing me 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, April 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, March 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] 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