Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 19, 2010

B. Constantine
JDSU

Oct. 19, 2009

# TCP Throughput Testing Methodology
## draft-constantine-ippm-tcp-throughput-tm-00

## Status of this Memo



## Copyright Notice

## Abstract

This memo describes a methodology for measuring TCP throughput performance in an end-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.

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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. Ultimately, network providers have come to the realization that a successful RFC2544 test result does not guarantee end-user satisfaction.

Therefore, the network provider community desires to measure network throughput performance at the TCP layer. Measuring TCP throughput will provide a much more meaningful measure that the network can meet 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.). These are all very complex topics to the general network community, and there is a strong interest to standardize a test methodology at the TCP layer (especially from an end-end perspective).

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 TCP layer performance, and guidelines for expected TCP throughput results that should be observed in the network under test. Network providers (and NEMs) are wrestling with end-end complexities of the above (queuing, active proxy devices, etc.); they desire to standardize the methodology to validate end-end TCP performance, as this is the precursor to acceptable end-user application performance.

## 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. This is important to note, since this methodology clearly does not intend to benchmark underlying mechanisms within various flavors of TCP OS implementations:

- 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

- This methodology will not drill into the detailed inner behavior of TCP implementations nor dissect the various timers and state machines that can affect TCP performance

Concerning the goals of this methodology, it is clearly from the perspective of a user that needs to conduct a structured, end-end assessment of TCP performance within a managed business class IP network. A key goal is that with the collective minds of this working group, to establish a set of "best practices" that a user should apply when validating the ability of a managed network to carry end-user TCP applications. Some specific goals are to:

- Provide the logical, next-step testing methodology so that a provider can test the network at Layer 4 (beyond the current Layer 2/3 RFC2544 testing approach)

- 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

- For networks that have been "tuned" with proper shaping and queuing control mechanisms, it is desirable to define a TCP layer test condition that can validate if the end-end network is tuned as expected.

- 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 all stateless traffic. Further more, many networks will not tolerate

TCP "traffic blasting" to emulate the TCP application traffic.

## 3. TCP Throughput Testing Methodology

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

### 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-end topology may be well enough defined.

#### 3.1.1 Techniques to Measure Round Trip Delay

This is not meant to provide an exhaustive list, but summarizes some of the more common ways to determine round trip delay (RTD) through the network. The desired resolution of the measurement (i.e. msec versus usec) may dictate whether the RTD 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 attempt 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 RTD 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 RTD based upon the SYN -> SYN-ACK handshakes within the TCP connection set-up.

- ICMP Pings may also be adequate to provide round trip delay 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 should also be able to track round trip delay of the TCP connection(s) during the test. This can provide insight into the potential effects of congestive delay to the throughput achieved for the TCP layer test.

## 3.2. Single TCP Connection Throughput Tests

With a reasonable representation of round trip delay 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 = RTD 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 RTD. The BDP would equal ~1,125,000 bits and the ideal TCP window would equal ~140,000 bytes.

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 set and the achieved throughput is measured, either uni-directionally or bi-directionally. For higher BDP situations in lossy networks (satellite links, etc.), TCP options such as Selective Acknowledgment should be considered and also become part of the window size / throughput characterization.

Ideally, the single connection TCP throughput test should be run over a a long duration and results logged at the desired interval. The test should record RTD and TCP retransmissions at each interval. The combination of throughput, RTD, and retransmissions per interval can provide valuable insight into the resulting TCP throughput results.

One important aspect of this test relates to the capabilities of the end test tools or hosts. Variation in host hardware performance can cause the results to be erroneous. Underperforming host hardware may not be able to transfer payload fast enough to the NIC card, and create the unintended bottleneck to TCP throughput 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.

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. If network conditions (RTD and retransmissions) are determined to be acceptable during the test interval and the TCP throughput is not, then this MAY point to active devices within the network that are altering window size (or other) TCP parameters.

## 3.3. Multiple TCP Connection Throughput Tests

After baselining the network under test with a single TCP connection (Section 3.2), the notional 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 will be 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.

### 3.3.1 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 RTD, 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, RTD, 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.

### 3.3.2 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 RTD 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.

### 3.4. Varying MSS per Connection

This test step can be run either on a single TCP connection test (Section 3.2) or a multiple TCP connection test (section 3.3).

By varying the MSS size of the TCP connection(s), 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.).

The single connection testing listed in Section 3.2 should be repeated first, using the appropriate window size and collecting throughput measurements per various MSS sizes. It would be reasonable to run single TCP connection tests with MSS sizes of 24, 88, 216, 472, 984, and 1460 bytes. These tests should also be run over a predetermined test interval and the throughput, retransmissions, and RTD logged during the entire test interval.

### 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 in the midst of 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 sizes 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 RTD 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 + RTD 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

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

Author's Address

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