Towards a Common TCP Evaluation Suite

Lachlan Andrew	Cesar Marcondes	Sally Floyd	Lawrence Dunn	Romaric Guillier	Wang Gang	Lars Eggert	Sangtae Ha and Injong Rhee
Caltech	UCLA	ICSI	Cisco Systems	INRIA	NEC China	Nokia	NCSU

I. Introduction

This document describes a common evaluation suite for the initial evaluation of new TCP extensions. It defines a small number of evaluation scenarios, including traffic and delay distributions, network topologies, and evaluation parameters and metrics. The motivation for such an evaluation suite is that it allows researchers that are proposing new TCP extensions and variants to quickly and easily evaluate their proposals against a common set of well-defined, standard test cases, in order to compare and contrast their proposal against standard TCP as well as other proposed extensions. It will also enable independent duplication and verification of reported results by others, which is an important aspect of the scientific method that is not often put to use by the networking community.

It is important to stress that this evaluation suite is not intended to result in an exhaustive evaluation of a TCP extension. Instead, the focus is on quickly and easily generating an initial evaluation report that allows the networking community to understand and discuss the behavioral aspects of a new proposal, in order to guide further experimentation that will be needed to fully investigate the specific aspects of a new proposal. A specific target was that the evaluations should be able to be completed in three days of simulations.

This paper is the outcome of a "round-table" meeting on TCP evaluation, held at Caltech on November 8–9, 2007.

II. Traffic generation

Congestion control concerns the response of flows to bandwidth limitations or to the presence of other flows. A realistic testing of a congestion control protocol must be conducted in a presence of a normal amount of cross-traffic. The flows of the protocols being tested must also be generated in a realistic fashion mimicking the traffic patterns commonly observed in the Internet. Cross-traffic has the desirable effect of reducing the occurrence of pathological conditions such as global synchronization among competing flows that might be interpreted as normal average behaviours of those protocols. This traffic must be realistic for the tests to predict the behaviour of congestion control protocols in real networks, and also well-defined so that statistical noise does not mask important effects.

It is important that the same "amount" of congestion or cross-traffic be used for the testing scenarios of different congestion control algorithms. This is complicated by the fact that packet arrivals and even flow arrivals are influenced by the behavior of the algorithms. For this reason, a pure packet-level generation of traffic where generated traffic does not respond to the behaviour of other present flows are not suitable. Instead, emulating application or user behaviours at the end points using reactive protocols such as TCP in a *closed-loop* fashion results in a closer approximation of cross-traffic, where user behaviours are modeled by well-defined parameters for source inputs (e.g., request sizes for HTTP), destination inputs (e.g., response size), and pause times (e..g, think times) between pairs of source and destination inputs. By setting appropriate parameters for the traffic generator, we can emulate non-greedy user-interactive traffic (e.g., HTTP 1.1, SMTP and Telnet) as well as greedy traffic (e.g., P2P and long file downloads). This approach models protocol reactions to the congestion caused by other flows in the common paths, but it fails to model the reactions of users themselves to the presence of the congestion.

While the protocols being tested may differ, it is important that we maintain the same "load" or level of congestion for the experimental scenarios. To enable this, we use a hybrid of open-loop and close-loop approaches. For this test suite, network traffic consists of *sessions* corresponding to individual users. Because users are independent, these session arrivals are well modeled by an *open-loop* Poisson process. A session may consist of a single greedy TCP flow, multiple greedy flows separated by user "think" times, or a single non-greedy flow with embedded think times. The session arrival process forms a Poisson process [1]. Both the think times and burst sizes have heavy-tailed distributions, with the exact distribution based on empirical studies. The think times and burst sizes will be chosen independently. This is unlikely to be the case in practice, but we have not been able to find any measurements of the joint distribution. We invite researchers to study this joint distribution, and future revisions of this test suite will use such statistics when they are available.

There are several traffic generators available that implement a similar approach to that discussed above. For now, we are using the Tmix [2] traffic generator. Tmix represents each Internet flow in terms of a sequence of *connection vectors* where a connection vector corresponds to a user. Connection vectors used for traffic generation can be obtained from Internet traffic traces. By taking measurement from various points of the Internet such as campus networks, DSL access links, and IPS core backbones, we can obtain sets of connection vectors for different levels of congested links. We plan to publish these connection vectors as part of this test suite.

A. Loads

Scaling the connection inter-arrival times by a constant varies the load. We invite other researchers to explore how the user behavior, as reflected in the connection-size distribution, might be affected by the level of congestion at the time.

Because the connection arrival times are specified independent of the file transfer times, one way to specify the load is as $A = \mathbb{E}[f]/\mathbb{E}[t]$ where $\mathbb{E}[f]$ is the mean connection size (in bits), $\mathbb{E}[t]$ is the mean connection inter-arrival time in seconds, and A is the load in bps.

It is important to test congestion control in "overloaded" conditions. However, if $A > c$, where c is the capacity of the bottleneck link, then the system has no equilibrium. The expected number of flows will increase without bound. This means that the measured results will be very sensitive to the duration of the simulation.

Instead, we measure the load as the "mean queue size of an M/G/1 queue using processor sharing." For small loads, say 10%, this is essentially equal to the fraction of the bandwidth. However, for overloaded systems, the fraction of the bandwidth used will be much less than this measure of load.

B. Equilibrium

In order to minimize the dependence of the results on the experiment durations, scenarios should be as stationary as possible. To this end, experiments will start with a number of active cross-traffic flows equal to the mean size of an M/G/1 queue using processor sharing, with traffic of the specified load.

Note that the distribution of the durations of the active flows at a given time is (often significantly) different from the distribution of flow durations, skewed toward long flows.

C. Packet size distribution

For flows generated by the traffic generator, 10% use 536-byte packets, and 90% 1500-byte packets. The packet size of each Tmix flow will be specified along with the start time and duration, to maximize the repeatability.

III. Round Trip Times

Most tests use a simple dumb-bell topology with a central link that connects two routers, as illustrated in Figure 1. Each router is connected to three nodes by edge links. In order to generate a typical range of round trip times, edge links have different delays. On one side, the one-way propagation delays are: 0 ms, 12 ms and 25 ms; on the other: 2 ms, 37 ms, and 75 ms. Traffic is uniformly shared among the nine source/destination pairs, giving a distribution of per-flow RTTs in the absence of queueing delay shown in Table I. These RTTs are computed for a dumb-bell topology with a delay of 0 ms for the central link. The delays for the edge links are based on those in [3]. The delay for the central link is given in the specific scenarios in the next section.

For dummynet experiments, delays can be obtained by specifying the delay of each flow.

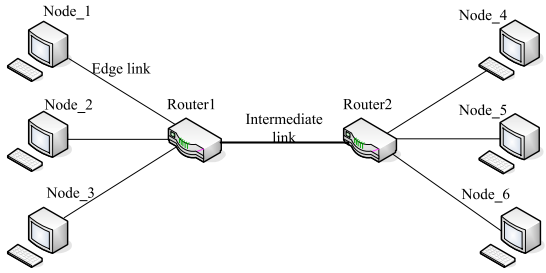

Fig. 1. A dumb-bell topology.

Path	RTT	Path	RTT	Path	RTT
1–4	4	1–5	74	1–6	150
2–4	26	2–5	98	2–6	174
3–4	54	3–5	124	3–6	200

TABLE I
RTTs OF THE PATHS BETWEEN TWO NODES, IN MILLISECONDS.

IV. Scenarios

It is difficult to provide TCP researchers with a complete set of scenarios for an exhaustive evaluation of a new TCP extension; especially because the characteristics of a new extension will often require experiments with specific scenarios that highlight its behavior. On the other hand, an exhaustive evaluation of a TCP extension will need to include several standard scenarios, and it is the focus of the test suite described in this section to define that basic set of test cases.

A. Basic scenarios

The purpose of the *basic scenarios* is to explore the behavior of a TCP extension over different link types. The scenarios use the dumb-bell topology of Section III, with the link delays modified as specified below.

This basic topology is used to instantiate several basic scenarios, by appropriately choosing capacity and delay parameters for the individual links. Depending on the configuration, the bottleneck link may be in one of the edge links or the central link. The basic scenarios are:

1) Data Center: The *data center* scenario models a case where bandwidth is plentiful and link delays are generally low. It uses the same configuration for the central link and all of the edge links. All links have a capacity of either 1 Gbps, 2.5 Gbps or 10 Gbps, and all links have a one-way propagation delay of either 1 ms or 10 ms [4].

2) Access Link: The *access link* scenario models an access link connecting an institution (e.g., a university or corporation) to an ISP. The central and edge links are all 100 Mbps. The one-way propagation delay of the central link is 2 ms, while the edge links have the delays given in Section III. Our goal in assigning delays to edge links is only to give a realistic distribution of round-trip times for traffic on the central link.

3) Trans-Oceanic Link: The *trans-oceanic* scenario models a test case where mostly lower-delay edge links feed into a high-delay central link. The central link is 1 Gbps, with a one-way propagation delay of 65 ms. The edge links have the same

bandwidth as the central link, with the one-way delays given in Section III. An alternative would be to use smaller delays for the edge links, with one-way delays for each set of three edge links of 5, 10, and 25 ms.[1]

4) Geostationary Satellite: The *geostationary satellite* scenario models an asymmetric test case with a high-bandwidth downlink and a low-bandwidth uplink [5], [6]. The capacity of the central link is 40 Mbps with a one-way propagation delay of 300 ms. The downlink capacity of the edge links is also 40 Mbps, but their uplink capacity is only 64 kbps. Edge one-way delays are as given in Section III. Note that "downlink" is towards the router for edge links attached to the first router, and away from the router for edge links on the other router.

5) Wireless Access: The *wireless access* scenario models a network with wireless edge links and a wired central link. The capacity of the wired link is 100 Mbps with 50 ms of one-way delay. The shared wireless link capacity is 11 Mbps (to model IEEE 802.11b links) or 54 Mbps (to model IEEE 802.11a/g links) with a one-way delay of either 1 ms or 10 ms.

Note that wireless links have many other unique properties not captured by delay and bitrate. Specifying these properties is beyond the scope of the current first version of this test suite.

6) Dial-up Link: The *dial-up link* scenario models a network with a dial-up link of 64 kbps and a one-way delay of 5 ms for the central link. This could be thought of as modeling a scenario reported at typical in Africa, with many users sharing a single low-bandwidth dial-up link.

As with all of the scenarios in this document, the basic scenarios could benefit from more measurement studies about characteristics of congested links in the current Internet, and about trends that could help predict the characteristics of congested links in the future. This would include more measurements on typical packet drop rates, and on the range of round-trip times for traffic on congested links.

For each of the basic scenarios, the goal is to run simulations or experiments in three cases: uncongested; mild congestion, and moderate congestion.[2] In the default case, the reverse path has a low level of traffic (10% load). The buffer size at the two routers is set to the maximum bandwidth-delay-product for a 100 ms flow (i.e., a maximum queueing delay of 100 ms), with drop-tail queues in units of packets. Each run will be for at least 100 seconds. (Testbeds might use longer run times, as should simulations with high bandwidth-delay products.)

For the access link scenario, more extensive simulations or experiments will be run, with both drop-tail and RED queue management, with drop-tail queues in units of both bytes and packets, and with RED queue management both in byte mode and in packet mode. Specific TCP extensions may require the evaluation of associated AQM mechanisms. For the access link scenario, simulations or experiments will also include runs with a reverse-path load equal to the forward-path load. For the access link scenario, additional experiments will use a range of buffer sizes, including 20% and 200% of the bandwidth-delay product for a 100 ms flow.

7) Outputs: For each run, the following metrics will be collected, for the central link in each direction: the aggregate link utilization, the average packet drop rate, and the average queueing delay[3], all over the second half of the run.

Other metrics of interest for general scenarios can be grouped in two sets: flow-centric and stability. The flow-centric metrics include the sending rate, good-put, cumulative loss and queueing delay trajectory for each flow, over time[4], and the transfer time per flow versus file size. Stability properties of interest include the standard deviation of the throughput and the queueing delay for the bottleneck link and for flows [4]. The worst case stability is also considered.

B. Delay/throughput tradeoff as function of queue size

Different queue management mechanisms have different delay-throughput tradeoffs. E.g., Adaptive Virtual Queue [7] gives low delay, at the expense of lower throughput. Different congestion control mechanisms may have different tradeoffs, which these tests aim to illustrate.

1) Topology and background traffic: These tests use the topology of Section III. This test is run for the access link scenario in Section IV-A.

For each Drop-Tail scenario set, five tests are run, with buffer sizes of 10%, 20%, 50%, 100%, and 200% of the Bandwidth Delay Product (BDP) for a 100 ms flow. For each AQM scenario (if used), five tests are run, with a target average queue size of 2.5%, 5%, 10%, 20%, and 50% of the BDP, with a buffer equal to the BDP.

2) Flows under test: The level of traffic from the traffic generator will be specified so that when a buffer size of 100% of the BDP is used with Drop Tail queue management, there is a moderate level of congestion (e.g., 1–2% packet drop rates when Standard TCP is used). Alternately, a range of traffic levels could be chosen, with a scenario set run for each traffic level (as in the examples cited below).

3) Outputs: For each test, three figures are kept, the average throughput, the average packet drop rate, and the average queueing delay over the second half of the test.

For each set of scenarios, the output is two graphs. For the delay/bandwidth graph, the x-axis shows the average queueing delay, and the y-axis shows the average throughput. For the drop-rate graph, the x-axis shows the average queueing delay, and the y-axis shows the average packet drop rate. Each pair of graphs illustrates the delay/throughput/drop-rate tradeoffs for this congestion control mechanism. For an AQM mechanism, each pair of graphs also illustrates how the throughput and average queue size vary (or don't vary) as a function of the

[1]Tests in simulators might have to use a smaller bandwidth for the trans-oceanic link, in order to run in a feasible amount of time. In testbeds, one of the metrics might be the number of timeouts in servers, due to implementation issues when running at high speed.

[2]The exact traffic loads and run times for each scenario will be specified later.

[3]This metric could be difficult to gather in emulated testbeds since routers statistics of queue utilization are not always reliable and depend on time-scale.

[4]Testbeds could use monitors in the TCP layer (e.g., Web100) to estimate the queueing delay and loss.

traffic load. Examples of delay/throughput tradeoffs appear in Figures 1–3 of [8] and Figures 4–5 of [9].

C. Convergence times: completion time of one flow

These tests aim to determine how quickly existing flows make room for new flows.

1) Topology and background traffic: Dumbbell. At least three capacities should be used, as close as possible to: 56 kbps, 10 Mbps and 1 Gbps. As always, 56 kbps is included to investigate the performance using mobile handsets.

For each capacity, three RTT scenarios should be tested, in which the existing and newly arriving flow have RTTs of (80 ms, 80 ms), (120 ms, 30 ms) and (30 ms, 120 ms).

Throughout the experiment, there is also 10% bidirectional cross traffic, as described in Section II, using the mix of RTTs described in Section III. All traffic is from the new TCP extension.

2) Flows under test: Traffic is dominated by two long lived flows, because we believe that to be the worst case, in which convergence is slowest.

One flow starts in "equilibrium" (at least having finished normal slow start). A new flow then starts; slow-start is disabled by setting the initial slow-start threshold to the initial CWND. Slow start is disabled because this is the worst case, and could happen if a loss occurred in the first RTT.

The experiment ends once the new flow has run for 5 minutes. Both of these flows use 1500-byte packets.

3) Outputs: The output of these experiments are the time until the $1500(10^n)$th byte of the new flow is received, for $n = 1, 2, \ldots$. This measures how quickly the existing flow releases capacity to the new flow, without requiring a definition of when "fairness" has been achieved. By leaving the upper limit on n unspecified, the test remains applicable to very high-speed networks.

A single run of this test cannot achieve statistical reliability by running for a long time. Instead, an average over *at least three* runs should be taken. Each run must use different pseudo-random cross traffic, as specified in Section II.

D. Transients: release of bandwidth, arrival of many flows

These tests investigate the impact of a sudden change of congestion level.

1) Topology and background traffic: The network is a single bottleneck link, with bit rate 100 Mbps, with a buffer of 1024 packets (120% BDP at 100 ms).

The transient traffic is generated using UDP, to avoid overlap with the scenario of Section IV-C and isolate the behavior of the flows under study. Three transients are tested:

1) step decrease from 75 Mbps to 0 Mbps,
2) step increase from 0 Mbps to 75 Mbps,
3) 30 step increases of 2.5 Mbps at 1 s intervals, simulating a "flash crowd" effect.

These transients occur after the flow under test has exited slow-start, and remain until the end of the experiment.

There is no TCP cross traffic as described in Section II in this experiment, because flow arrivals/departures occur on timescales long compared with these effects.

2) Flows under test: There is one flow under test: a long-lived flow in the same direction as the transient traffic, with a 100 ms RTT.

3) Outputs: For the decrease in cross traffic, the metrics are (i) the time taken for the flow under test to increase its window to 80% of the BDP, and (ii) the maximum change of the window in a single RTT while the window is increasing to that value.

For cases with an increase in cross traffic, the metric is the time taken for the flow under test to reduce its window by a factor of 3. This is chosen to be a greater change than the common response to a *single* loss, but less than the reduction in equilibrium rate.

E. Impact on standard TCP traffic

Many new TCP proposals achieve a gain, G, in their own throughput at the expense of a loss, L, in the throughput of standard TCP flows sharing a bottleneck, as well as by increasing the link utilization. This scenario quantifies this tradeoff.

1) Topology and background traffic: The dumb-bell of Section III is used with the same capacities as for the convergence tests (Section IV-C). All traffic in this scenario comes from the flows under test.

2) Flows under test: The scenario is performed by conducting pairs of experiments, with identical flow arrival times and flow sizes. Within each experiment, flows are divided into two *camps*. For every flow in camp A, there is a flow with the same size, source and destination in camp B, and vice versa. The start time of the two flows are within 2 s.

The file sizes and start times are as specified in Section II, with start times scaled to achieve loads of 50% and 100%. In addition, both camps have a long-lived flow. The experiments last for 1200 seconds.

In the first experiment, called BASELINE, both camp A and camp B use standard TCP. In the second, called MIX, camp A uses standard TCP and camp B uses the new TCP extension.

The rationale for having paired camps is to remove the statistical uncertainty which would come from randomly choosing half of the flows to run each algorithm. This way, camp A and camp B have the same loads.

3) Outputs: The gain achieved by the new algorithm and loss incurred by standard TCP are given by

$$G = \frac{T(B)_{MIX}}{T(B)_{BASELINE}} \qquad L = \frac{T(A)_{MIX}}{T(A)_{BASELINE}}$$

where $T(x)$ is the throughput obtained by camp x, measured as the amount of data acknowledged by the receivers (that is, "goodput"), and taken over the last 8000 seconds of the experiment.

The loss, L, is analogous to the "bandwidth stolen from TCP" in [10] and "throughput degradation" in [11].

A plot of G vs L represents the tradeoff between efficiency and loss.

4) Suggestions: Other statistics of interest are the values of G and L for each quartile of file sizes. This will reveal whether the new proposal is more aggressive in starting up or more reluctant to release its share of capacity.

As always, testing at other loads and averaging over multiple runs are encouraged.

F. Intra-protocol and inter-RTT fairness

These tests aim to measure bottleneck bandwidth sharing among flows of the same protocol with the same RTT, which represents the flows going through the same routing path. The tests also measure inter-RTT fairness, the bandwidth sharing among flows of the same protocol where routing paths have a common bottleneck segment but might have different overall paths with different RTTs.

1) Topology and background traffic: The topology, the capacity and cross traffic conditions of these tests are the same as in IV-C. The bottleneck buffer is varied from 25% to 200% BDP for a 100 ms flow, increasing by factors of 2.

2) Flows under test: We use two flows of the same protocol for this experiment. The RTTs of the flows range from 10 ms to 160 ms[5] (10 ms, 20 ms, 40 ms, 80 ms, and 160 ms) such that the ratio of the minimum RTT over the maximum RTT is at most 1/16.

Intra-protocol fairness: For each run, two flows with the same RTT, taken from the range of RTTs above start randomly within the first 10% of the experiment. The order in which these flows start doesn't matter. An additional test of interest, but not part of this suite, would involve two extreme cases — two flows with very short or long RTTs (e.g., the delay less then 1–2 ms represents communication happen in the data-center and the delay larger than 600 ms considers communication over satellite).

Inter-RTT fairness: For each run, one flow with a fixed RTT of 160 ms starts first, and another flow with a different RTT taken from the range of RTTs above, joins afterward. The starting times of both two flows are randomly chosen within the first 10% of the experiment as before.

3) Outputs: The output of this experiment is the ratio of the average throughput values of the two flows. The output also includes the packet drop rate for the congested link.

G. Multiple bottlenecks

These experiments explore the relative bandwidth for a flow that traverses multiple bottlenecks, and flows with the same round-trip time that each traverse only one of the bottleneck links.

1) Topology and background traffic: The topology is a "parking-lot" topology with three (horizontal) bottleneck links and four (vertical) access links. All links have a one-way delay of 10 ms. The bottleneck links have a rate of 100 Mbps, and the access links have a rate of 1 Gbps. All flows cover three links, so all flows have a round-trip time of 60 ms.

[5]In case the testbed doesn't support up to 160 ms RTT, we can scale down the RTTs in proportion to the maximum RTT supported in that environment.

Throughout the experiment, there is 10% bidirectional cross traffic on each of the three bottleneck links, as described in Section II. The cross-traffic flows all traverse two access links and a single bottleneck link.

All traffic uses the new TCP extension.

2) Flows under test: In addition to the cross-traffic, there are four flows under test, all with traffic in the same direction on the bottleneck links. The multiple-bottleneck flow traverses no access links and all three bottleneck links. The three single-bottleneck flows each traverse two access links and a single bottleneck link, with one flow for each bottleneck link. The flows start in quick succession, separated by approximately 1 second. These flows last at least 5 minutes.

An additional test of interest would be to have a longer, multiple-bottleneck flow competing against shorter single-bottleneck flows.

3) Outputs: The output for this experiment is the ratio between the average throughput of the single-bottleneck flows and the throughput of the multiple-bottleneck flow, measured over the second half of the experiment. Output also includes the packet drop rate for the congested link.

V. Conclusion

An initial specification of an evaluation suite for TCP extensions has been described. Future versions will include: detailed specifications, with modifications for simulations and testbeds; more measurement results about congested links in the current Internet; alternate specifications; and specific sets of scenarios that can be run in a plausible period of time in simulators and testbeds, respectively.

References

[1] N. Hohn, D. Veitch, and P. Abry, "The Impact of the Flow Arrival Process in Internet Traffic," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, vol. 6, pp. 37–40, 2003.

[2] M. C. Weigle, P. Adurthi, F. Hernández-Campos, K. Jeffay, and F. D. Smith, "Tmix: a tool for generating realistic TCP application workloads in ns-2," *SIGCOMM Computer Communication Review (CCR)*, vol. 36, no. 3, pp. 65–76, 2006.

[3] S. Floyd and E. Kohler, "Internet Research Needs Better Models," *SIGCOMM Computer Communication Review (CCR)*, vol. 33, no. 1, pp. 29–34, 2003.

[4] D. Wei, P. Cao, and S. Low, "Time for a TCP benchmark suite?" 2005.

[5] T. Henderson and R. Katz, "Transport Protocols for Internet-Compatible Satellite Networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 17, no. 2, pp. 326–344, 1999.

[6] A. Gurtov and S. Floyd, "Modeling Wireless Links for Transport Protocols," *SIGCOMM Computer Communication Review (CCR)*, vol. 34, no. 2, pp. 85–96, 2004.

[7] S. Kunniyur and R. Srikant, "Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management," *Proc. SIGCOMM'01*, pp. 123–134, 2001.

[8] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED," ICIR, Tech. Rep., 2001. [Online]. Available: http://www.icir.org/floyd/papers/adaptiveRed.pdf

[9] L. L. H. Andrew, S. V. Hanly, and R. G. Mukhtar, "Active Queue Management for Fair Resource Allocation in Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 7, 2008.

[10] E. Souza and D. Agarwal, "A HighSpeedTCP Study: Characteristics and Deployment Issues," LBNL, Technical Report LBNL-53215, 2003.

[11] H. Shimonishi, M. Sanadidi, and T. Murase, "Assessing Interactions among Legacy and High-Speed TCP Protocols," *Proc. Workshop on Protocols for Fast Long-Delay Networks (PFLDNet)*, 2007.