Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2015

D. Hayes
University of Oslo
D. Ros
Telecom Bretagne
L.L.H. Andrew
CAIA Swinburne University of Technology
S. Floyd
ICSI
July 3, 2014

Common TCP Evaluation Suite
draft-irtf-iccrg-tcpeval-00

Abstract

This document presents an evaluation test suite for the initial assessment of proposed TCP modifications. The goal of the test suite is to allow researchers to quickly and easily evaluate their proposed TCP extensions in simulators and testbeds using a common set of well-defined, standard test cases, in order to compare and contrast proposals against standard TCP as well as other proposed modifications. This test suite is not intended to result in an exhaustive evaluation of a proposed TCP modification or new congestion control mechanism. 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.

Table of Contents

1 Introduction

This document describes a common test suite for the initial assessment of new TCP extensions or modifications. 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 to help researchers in evaluating their proposed modifications to TCP. The evaluation suite 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. A specific target is that the evaluations should be able to be completed in a reasonable amount of time by simulation, or with a reasonable amount of effort in a testbed.

It is not possible 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 document to define this initial set of test cases.

These scenarios generalize current characteristics of the Internet such as round-trip times (RTT), propagation delays, and buffer sizes. It is envisaged that as the Internet evolves these will need to be adjusted. In particular, we expect buffer sizes will need to be adjusted as latency becomes increasingly important.

The scenarios specified here are intended to be as generic as possible, i.e., not tied to a particular simulation or emulation platform. However, when needed some details pertaining to implementation using a given tool are described.

This document has evolved from a "round-table" meeting on TCP evaluation, held at Caltech on November 8-9, 2007, reported in [1]. This document is the first step in constructing the evaluation suite; the goal is for the evaluation suite to be adapted in response to feedback from the networking community. It revises draft-irtf-tmrg-tests-02.

Information related to the draft can be found at:
http://riteproject.eu/ietf-drafts

2 Traffic generation

Congestion control concerns the response of flows to bandwidth limitations or to the presence of other flows. Cross-traffic and reverse-path traffic are therefore important to the tests described in this suite.

Such traffic can have the desirable effect of reducing the occurrence of pathological conditions, such as global synchronization among competing flows, that might otherwise be mis-interpreted as normal average behaviours of those protocols [2,3]. This traffic must be reasonably 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.

2.1 Desirable model characteristics

Most scenarios use traffic produced by a traffic generator, with a range of start times for user sessions, connection sizes, and the like, mimicking the traffic patterns commonly observed in the Internet. 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 open-loop, packet-level generation of traffic where generated traffic does not respond to the behaviour of other present flows is 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 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), greedy traffic (e.g., P2P and long file downloads), as well as long-lived but non-greedy, non-interactive flows (or thin streams).

This approach models protocol reactions to the congestion caused by other flows in the common paths, although it fails to model the reactions of users themselves to the presence of congestion. A model that includes end-users' reaction to congestion is beyond the scope of this draft, but we invite researchers to explore how the user behavior, as reflected in the connection sizes, user wait times, and number of connections per session, might be affected by the level of congestion expe-

rienced within a session [4].

2.2 Tmix

There are several traffic generators available that implement a similar approach to that discussed above. For now, we have chosen to use the Tmix [5] traffic generator. Tmix is available for the NS2 and NS3 simulators, and can generate traffic for testbeds (for example GENI [6]).

Tmix represents each TCP connection by a connection vector consisting of a sequence of (request-size, response-size, think-time) triples, thus

representing bi-directional traffic. Connection vectors used for traffic generation can be obtained from Internet traffic traces.

2.2.1 Base Tmix trace files for tests

The traces currently defined for use in the test suite are based on campus traffic at the University of North Carolina (see [7] for a description of construction methods and basic statistics).

The traces have an additional "m" field added to each connection vector to provide each direction's maximum segment size for the connection. This is used to provide the packet size distribution described in section 2.4.

These traces contain a mixture of connections, from very short flows that do not exist for long enough to be "congestion controlled", to long thin streams, to bulk file transfer like connections.

The traces are available at:
http://hosting.riteproject.eu/tcpevaltmixtraces.tgz

2.3 Loads

While the protocols being tested may differ, it is important that we maintain the same "load" or level of congestion for the experimental scenarios. For many of the scenarios, such as the basic ones in section 4, each scenario is run for a range of loads, where the load is varied by varying the rate of session arrivals.

2.3.1 Varying the Tmix traffic load

To adjust the traffic load for a given scenario, the connection start times for flows in a Tmix trace are scaled as follows. Connections are actually started at:

experiment_cv_start_time = scale * cv_start_time

where cv_start_time denotes the connection vector start time in the Tmix traces and experiment_start_time is the time the connection starts in the experiment. Therefore, the smaller the scale the higher (in general) the traffic load.

2.3.1.1 Notes

Changing the connection start times also changes the way the traffic connections interact, potentially changing the "clumping" of traffic bursts.

Very small changes in the scaling parameter can cause disproportionate changes in the offered load. This is due to possibility of the small change causing the exclusion or inclusion of a CV that will transfer a very large amount of data.

2.3.2 Dealing with non-stationarity

The Tmix traffic traces, as they are, offer a non-stationary load. This is exacerbated for tests that do not require use of the full trace files, but only a portion of them. While removing this non-stationarity does also remove some of the "realism" of the traffic, it is necessary for the test suite to produce reliable and consistent results.

A more stationary offered load is achieved by shuffling the start times of connection vectors in the Tmix trace file. The trace file is logically partitioned into n-second bins, which are then shuffled using a Fisher-Yates shuffle [8], and the required portions written to shuffled trace files for the particular experiment being conducted.

2.3.2.1 Bin size

The bin size is chosen so that there is enough shuffling with respect to the test length. The offered traffic per test second from the Tmix trace

files depends scale factor (see section 2.3.1), which is related to the capacity of the bottleneck link. The shuffling bin size (in seconds) is set at: b = 500e6 / C where C is the bottleneck link's capacity in bits per second, and 500e6 is a scaling factor (in bits).

Thus for the access link scenario described in section 4.4.2, the bin size for shuffling will be 5 seconds.

2.3.2.2 NS2 implementation specifics

The tcl scripts for this process are distributed with the NS2 example test suite implementation. Care must be taken when using this algorithm, so that the given random number generator and the same seed are employed, or else the resulting experimental traces will be different.

2.4 Packet size distribution

For flows generated by the traffic generator, 10% of them use 536-byte packets, and 90% 1500-byte packets. The base Tmix traces described in section 2.2.1 have been processed at the *connection* level to have this characteristic. As a result, *packets* in a given test will be roughly, but not be exactly, in this proportion. However, the proportion of offered traffic will be consistent for each experiment.

2.4.1 Potential revision

As Tmix can now read and use a connection's Maximum Segment Size (MSS) from the trace file, it will be possible to produce Tmix connection vector trace files where the packet sizes reflect actual measurements.

3 Achieving reliable results in minimum time

This section describes the techniques used to achieve reliable results in the minimum test time.

3.1 Background

Over a long time, because the session arrival times are to a large extent independent of the transfer times, load could be defined as:

A=E[f]/E[t],

where E[f] is the mean session (flow) size in bits transferred, E[t] is the mean session inter-arrival time in seconds, and A is the load in bps.

It is important to test congestion control protocols in "overloaded" conditions. However, if A>C, where C is the capacity of the bottleneck link, then the system has no equilibrium. In long-running experiments with A>C, the expected number of flows would keep on increasing with time (because as time passes, flows would tend to last for longer and longer, thus "piling up" with newly-arriving ones). This means that, in an overload scenario, some measures will be very sensitive to the duration of the tests.

3.2 Equilibrium or Steady State

Ideally, experiments should be run until some sort of equilibrium results can be obtained. Since every test algorithm can potentially change how long this may take, the following approach is adopted:

1. Traces are shuffled to remove non-stationarity (see section 2.3.2.)

2. The experiment run time is determined from the traffic traces. The shuffled traces are compiled such that the estimate of traffic offered in the second third of the test is equal to the estimated traffic offered in the second third of the test is equal to the estimate of traffic offered in the final third of the test, to within a 5% tolerance. The length of the trace files becomes the total experiment run time (including the warm up time).

3. The warmup time until measurements start, is calculated as the time at which the NS2 simulation of standard TCP achieves "steady state". In this case, warmup time is determined as the time required so the measurements have statistically similar first and second half results. The metrics used as reference are: the bottleneck raw throughput, and the average bottleneck queue size. The latter is stable when A>>C and A<<C, but not when AapproxC. In this case the queue is not a stable measure, and just the raw bottleneck throughput is used.

3.2.1 Note on the offered load in NS2

The offered load in an NS2 simulation using one-way TCP will be higher than the estimated load. One-way TCP uses fixed TCP segment sizes, so all transmissions that would normally use a segment size less than the maximum segment size (in this case 496B or 1460B), such as at the end of a block of data, or for short queries or responses, will still be sent as a maximum segment size packet.

3.3 Accelerated test start up time

Tmix traffic generation does not provide an instant constant load. It can take quite a long time for the number of simultaneous TCP connections, and thus the offered load, to build up when using Tmix to generate the load. To accelerate the system start up, the system is "prefilled" to a state close to "steady state", as follows.

Connections that start before t=prefill_t are selected with a bias toward longer sessions. Only connections which are estimated to continue past the long_flow_bias time (see figure 1) are selected.

The prefill_t (in seconds) calculation has been automated, based on the following heuristic: prefill_t = 1.5 * targetload * maxRTT where maxRTT is the median maximum RTT in the particular topology, and targetload is given as a percentage. The long_flow_bias threshold is set at long_flow_bias = prefill_t / 2 These values are not optimal, but have been experimentally determined to give reasonable results.

These selected connections are then started at an accelerated rate so that the estimated resulting load over the accelerated start up time is the target load for this experiment: prefill_si = total_pfcb / (C * TL / 100.0) where prefill_si is the interval of time for the accelerated start up, total_pfcb is the total number of bits estimated to be sent by the prefill connections, C is the capacity of the bottleneck link, and TL is the target offered load as a percentage.

This procedure has the effect of quickly bringing the system to a loaded state. From this point the system runs until t=warmup (as calculated in

section 3.2), after which moment statistics are computed.

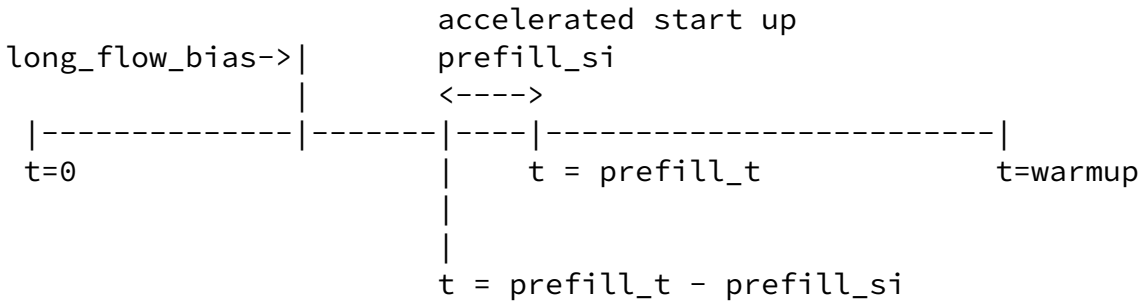


Figure 1: prefilling

4 Basic scenarios

The purpose of the basic scenarios is to explore the behavior of a TCP extension over different link types. These scenarios use the dumbbell topology described in section 4.1.

4.1 Basic topology

Most tests use a simple dumbbell topology with a central link that connects two routers, as illustrated in Figure 2. Each router is also connected to three nodes by edge links. In order to generate a typical range of round trip times, edge links have different delays. Unless specified otherwise, such delays are as follows. On one side, the one-way propagation delays are: 0ms, 12ms and 25ms; on the other: 2ms, 37ms, and 75ms. 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 1. These RTTs are computed for a dumbbell topology assuming a delay of 0ms for the central link. The delay for the central link that is used in a specific scenario is given in the next section.

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

4.2 Traffic

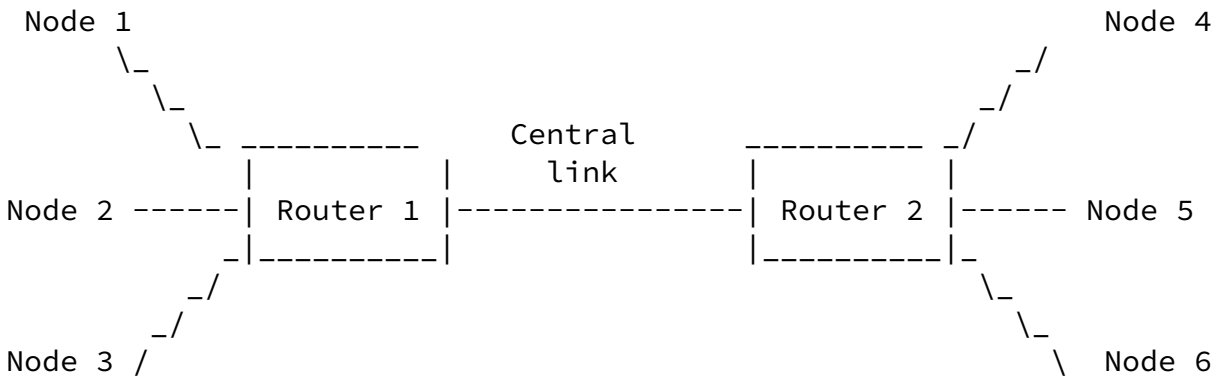


Figure 2: A dumbbell topology

Path	RTT	Path	RTT	Path	RTT
1-4	4	1-5	74	1-6	150
2-4	28	2-5	98	2-6	174
3-4	54	3-5	124	3-6	200

Table 1: Minimum RTTs of the paths between two nodes, in milliseconds.

In all of the basic scenarios, *all* TCP flows use the TCP extension or modification under evaluation.

In general, the 9 bidirectional Tmix sources are connected to nodes 1 to 6 of figure 2 to create the paths tabulated in table 1.

Offered loads are estimated directly from the shuffled and scaled Tmix traces, as described in section 3.2. The actual measured loads will depend on the TCP variant and the scenario being tested.

Buffer sizes are based on the Bandwidth Delay Product (BDP), except for the Dial-up scenario where a BDP buffer does not provide enough buffering.

The load generated by Tmix with the standard trace files is asymmetric, with a higher load offered in the right to left direction (refer to fig-

ure 2) than in the left to right direction. Loads are specified for the

higher traffic right to left direction. For each of the basic scenarios, three offered loads are tested: moderate (60%), high (85%), and overload (110%). Loads are for the bottleneck link, which is the central link in all scenarios except the wireless LAN scenario.

The 9 tmix traces are scaled using a single scaling factor in these tests. This means that the traffic offered on each of the 9 paths through the network is not equal, but combined at the bottleneck produces the specified offered load.

4.3 Flows under test

For these basic scenarios, there is no differentiation between "cross-traffic" and the "flows under test". The aggregate traffic is under test, with the metrics exploring both aggregate traffic and distributions of flow-specific metrics.

4.4 Scenarios

4.4.1 Data Center

The data center scenario models a case where bandwidth is plentiful and link delays are generally low. All links have a capacity of 1 Gbps. Links from nodes 1, 2 and 4 have a one-way propagation delay of 10 us, while those from nodes 3, 5 and 6 have 100 us [9], and the central link has 0 ms delay. The central link has 10 ms buffers.

load	scale	experiment time	warmup	test_time	prefill_t	prefill_si
60%	0.56385119	156.5	4.0	145.0	7.956	4.284117
85%	0.372649	358.0	19.0	328.0	11.271	6.411839
110%	0.295601	481.5	7.5	459	14.586	7.356242

Table 2: Data center scenario parameters

4.4.1.1 Potential Revisions

The rate of 1 Gbps is chosen such that NS2 simulations can run in a reasonable time. Higher values will become feasible as computing power increases, however the current traces may not be long enough to drive simulations or test bed experiments at higher rates.

The supplied Tmix traces are used here to provide a standard comparison across scenarios. Data Centers, however, have very specialised traffic which may not be represented well in such traces. In the future,

specialised Data Center traffic traces may be needed to provide a more realistic test.

4.4.2 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 4.1. 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. The Central link buffer size is 100 ms, which is equivalent to the BDP (using the mean RTT).

load	scale	experiment time	warmup	test_time	prefill_t	prefill_si
60%	4.910115	440	107.0	296.0	36.72	24.103939
85%	3.605109	920	135.0	733.0	52.02	23.378915
110%	3.0027085	2710	34.0	2609.0	67.32	35.895355

Table 3: Access link scenario parameters (times in seconds)

4.4.2.1 Potential Revisions

As faster access links become common, the link speed for this scenario will need to be updated accordingly. Also as access link buffer sizes shrink to less than BDP sized buffers, this should be updated to reflect these changes in the Internet.

4.4.3 Trans-Oceanic Link

The trans-oceanic scenario models a test case where mostly lower-delay edge links feed into a high-delay central link. Both the central and all edge links are 1 Gbps. The central link has 100 ms buffers, and a one-way propagation delay of 65 ms. 65 ms is chosen as a "typical number". The actual delay on real links depends, of course, on their length. For example, Melbourne to Los Angeles is about 85 ms.

4.4.4 Geostationary Satellite

The geostationary satellite scenario models an asymmetric test case with a high-bandwidth downlink and a low-bandwidth uplink [10,11]. The scenario modeled is that of nodes connected to a satellite hub which has an asymmetric satellite connection to the master base station which is

load	scale	experiment time	warmup	test_time	prefill_t	prefill_si
60%	tbd	tbd	tbd			
85%	tbd	tbd	tbd			
110%:	tbd	tbd	tbd			

Table 4: Trans-Oceanic link scenario parameters

connected to the Internet. The capacity of the central link is asymmetric - 40 Mbps down, and 4 Mbps up with a one-way propagation delay of 300 ms. Edge links are all bidirectional 100 Mbps links with one-way delays as given in Section 4.1. The central link buffer size is 100 ms for downlink and 1000 ms for uplink.

Note that congestion in this case is often on the 4 Mbps uplink (left to right), even though most of the traffic is in the downlink direction (right to left).

load	scale	experiment time	warmup	test_time	prefill_t	prefill_si
60%	tbd	tbd	tbd			
85%	tbd	tbd	tbd			
110%:	tbd	tbd	tbd			

Table 5: Trans-Oceanic link scenario parameters

4.4.5 Wireless LAN

The wireless LAN scenario models WiFi access to a wired backbone, as depicted in Figure 3.

The capacity of the central link is 100 Mbps, with a one-way delay of 2 ms. All links to Router 2 are wired. Router 1 acts as a base station for a shared wireless IEEE 802.11g links. Although 802.11g has a peak bit rate of 54 Mbps, its typical throughput rate is much lower, and decreases under high loads and bursty traffic. The scales specified here are based on a nominal rate of 6Mbps.

The Node_[123] to Wireless_[123] connections are to allow the same RTT distribution as for the wired scenarios. This is in addition to delays on the wireless link due to CSMA. Figure 3 shows how the topology should look in a test bed.

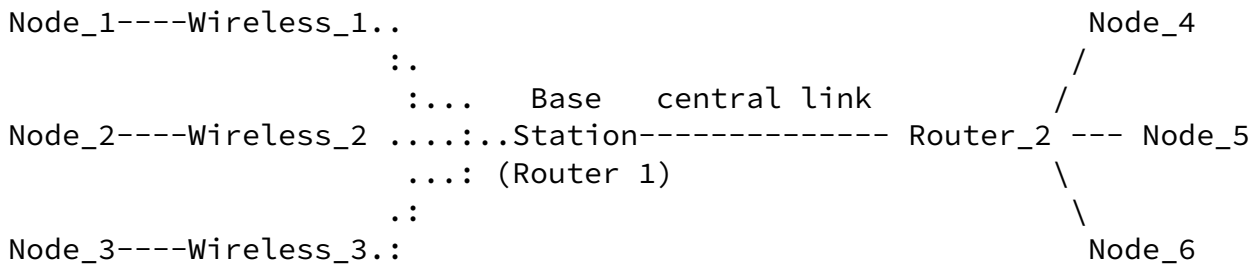


Figure 3: Wireless dumbell topology for a test-bed. Wireless_n are wireless transceivers for connection to the base station

load	scale	experiment time	warmup	test_time	prefill_t	prefill_si

60%	117.852049	14917	20.0	14917	0	0
85%	85.203155	10250.0	20.0	10230	0	0
110%:	65.262840	4500.0	20.0	4480	0	0

Table 6: Wireless LAN scenario parameters

The percentage load for this scenario is based on the sum of the estimate of offered load in both directions since the wireless bottleneck link is a shared media. Also, due to contention for the bottleneck link, the accelerated start up using prefill is not used for this scenario.

Note that the prefill values are zero as prefill was found to be of no benefit in this scenario.

4.4.5.1 NS2 implementation specifics

In NS2, this is implemented as depicted in Figure 2 The delays between Node_1 and Wireless_1 are implemented as delays through the Logical Link layer.

Since NS2 don't have a simple way of measuring transport packet loss on the wireless link, dropped packets are inferred based on flow arrivals and departures (see figure 4). This gives a good estimate of the average loss rate over a long enough period (long compared with the transit delay of packets), which is the case here.

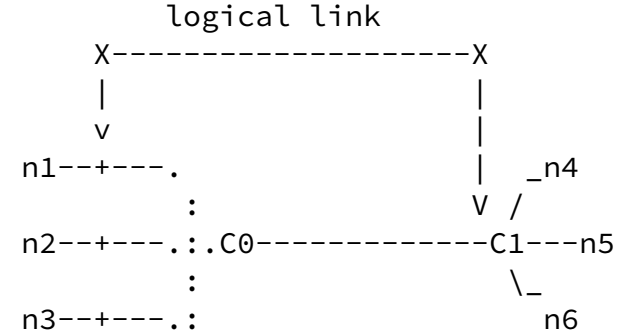


Figure 4: Wireless measurements in the ns2 simulator

4.4.5.2 Potential revisions

Wireless standards are continually evolving. This scenario may need updating in the future to reflect these changes.

Wireless links have many other unique properties not captured by delay and bitrate. In particular, the physical layer might suffer from propagation effects that result in packet losses, and the MAC layer might add high jitter under contention or large steps in bandwidth due to adaptive modulation and coding. Specifying these properties is beyond the scope of the current first version of this test suite but may make useful additions in the future.

Latency in this scenario is very much affected by contention for the media. It will be good to have end-to-end delay measurements to quantify this characteristic. This could include per packet latency, application burst completion times, and/or application session completion times.

4.4.6 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 as typical in Africa, with many users sharing a single low-bandwidth dial-up link. Central link buffer size of 1250 ms

4.4.6.1 Note on parameters

The traffic offered by tmix over a low bandwidth link is very bursty. It takes a long time to reach some sort of statistical stability. For event

load	scale	experiment time	warmup	test_time	prefill_t	prefill_si
60%	10176.2847	1214286	273900	273900	0	0
85%	7679.1920	1071429	513600	557165	664.275	121.147563

110%:	5796.7901	2223215	440.0	2221915	859.65	180.428

Table 7: Dial-up link scenario parameters

based simulators, this is not too much of a problem, as the number of packets transferred is not prohibitively high, however for test beds these times are prohibitively long. This scenario needs further investigation to address this.

4.4.6.2 Potential revisions

Modems often have asymmetric up and down link rates. Asymmetry is tested in the Geostationary Satellite scenario (section 4.4.4), but the dial-up scenario could be modified to model this as well.

4.5 Metrics of interest

For each run, the following metrics will be collected for the central link in each direction:

1. the aggregate link utilization,

2. the average packet drop rate, and

3. the average queueing delay.

These measures only provide a general overview of performance. The goal of this draft is to produce a set of tests that can be "run" at all levels of abstraction, from Grid500's WAN, through WAN-in-Lab, testbeds and simulations all the way to theory. Researchers may add additional measures to illustrate other performance aspects as required.

Other metrics of general interest include:

1. end-to-end delay measurements

2. flow-centric:

    1. sending rate,

    2. goodput,

3. cumulative loss and queueing delay trajectory for each flow, over time,

4. the transfer time per flow versus file size

3. stability properties:

1. standard deviation of the throughput and the queueing delay for the bottleneck link,

2. worst case stability measures, especially proving (possibly theoretically) the stability of TCP.

4.6 Potential Revisions

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.

5 Latency specific experiments

5.1 Delay/throughput tradeoff as function of queue size

Performance in data communications is increasingly limited by latency. Smaller and smarter buffers improve this measure, but often at the expense of TCP throughput. The purpose of these tests is to investigate delay-throughput tradeoffs, *with and without the particular TCP extension under study*.

Different queue management mechanisms have different delay-throughput tradeoffs. It is envisaged that the tests described here would be extended to explore and compare the performance of different Active Queue Management (AQM) techniques. However, this is an area of active research and beyond the scope of this test suite at this time. For now, it may be better to have a dedicated, separate test suite to look at AQM performance issues.

5.1.1 Topology

These tests use the topology of Figure 4.1. They are based on the access link scenario (see section 4.4.2) with the 85% offered load used for this test.

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 base RTT flow (the average base RTT in the access link dumbell scenario is 100 ms).

5.1.1.1 Potential revisions

Buffer sizing is still an area of research. Results from this research may necessitate changes to the test suite so that it models these changes in the Internet.

AQM is currently an area of active research. It is envisaged that these tests could be extended to explore and compare the performance of key AQM techniques when it becomes clear what these will be. For now a dedicated AQM test suite would best serve such research efforts.

5.1.2 Flows under test

Two kinds of tests should be run: one where all TCP flows use the TCP modification under study, and another where no TCP flows use such modification, as a "baseline" version.

The level of traffic from the traffic generator is the same as that described in section 4.4.2.

5.1.3 Metrics of interest

For each test, three figures are kept, the average throughput, the average packet drop rate, and the average queueing delay over the measurement period.

Ideally it would be better to have more complete statistics, especially for queueing delay where the delay distribution can be important. It would also be good for this to be illustrated with 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 with and without the TCP mechanism under evaluation. 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 traffic load. Examples of delay/throughput tradeoffs appear in Figures 1-3 of[12] and Figures 4-5 of[13].

5.2 Ramp up time: completion time of one flow

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

5.2.1 Topology and background traffic

The ramp up time test uses the topology shown in figure 5. Two long-lived test TCP connections are used in this experiment. Test TCP connection 1 is run between T_n1 and T_n3, with data flowing from T_n1 to T_n3, and test TCP source 2 runs between T_n2 and T_n4, with data flowing from T_n2 to T_n4. The background traffic topology is identical to that used in the basic scenarios (see section 4 and Figure 2); i.e., background flows run between nodes B_n1 to B_n6.

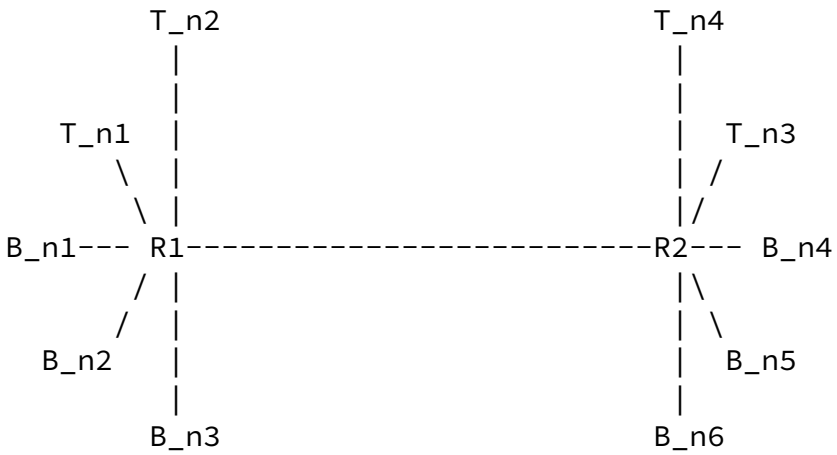


Figure 5: Ramp up dumbbell test topology

Experiments are conducted with capacities of 56 kbps, 10 Mbps and 1 Gbps for the central link. The 56 kbps case is included to investigate the performance using low bit rate devices such as mobile handsets or dial

up modems.

For each capacity, three RTT scenarios should be tested, in which the existing and newly arriving flow have RTTs of (80,80), (120,30), and (30,120) respectively. This is made up of a central link has a 2 ms delay in each direction, and test link delays as shown in Table 5.2.1.

Throughout the experiment, the offered load of the background (or cross) traffic is 10% of the central link capacity in the right to left direction. The background traffic is generated in the same manner as for the basic scenarios (see section 4).

RTT scenario	T_n1 (ms)	T_n2 (ms)	T_n3 (ms)	T_n4 (ms)
1	0	0	38	38
2	23	12	35	1
3	12	23	1	35

Table 8: Link delays for the test TCP source connections to the central link

Central link	scale	experiment time	warmup	test_time	prefill_t	prefill_s
56	kbps					
10	Mbps					
1	Gbps	3.355228	324		9.18	2.82020

Table 9: Ramp-up time scenario parameters (times in seconds)

All traffic for this scenario uses the TCP extension under test.

5.2.2 Flows under test

Traffic is dominated by the two long lived test 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 five minutes. Both of the flows use 1500-byte packets. The test should be run both with Standard TCP and with the TCP extension under test for comparison.

5.2.2.1 Potential Revisions

It may also be useful to conduct the tests with slow start enabled too, if time permits.

5.2.3 Metrics of interest

The output of these experiments are the time until the 1500times10n th byte of the new flow is received, for n = 1,2,... . 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 cross traffic. Different cross traffic can be generated using the standard tmix trace files by changing the random number seed used to shuffle the traces.

5.3 Transients: release of bandwidth, arrival of many flows

These tests investigate the impact of a sudden change of congestion level. They differ from the "Ramp up time" test in that the congestion here is caused by unresponsive traffic.

Note that this scenario has not yet been implemented in the NS2 example test suite.

5.3.1 Topology and background traffic

The network is a single bottleneck link (see Figure 6), with bit rate 100 Mbps, with a buffer of 1024 packets (i.e., 120% of the BDP at 100 ms).

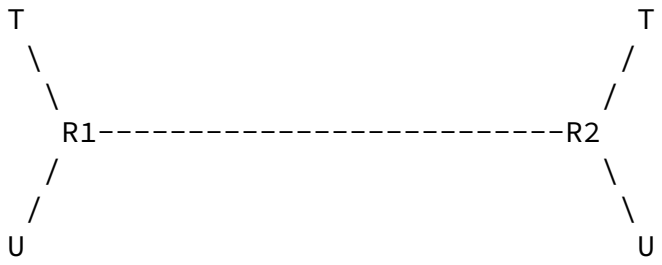


Figure 6: Transient test topology

The transient traffic is generated using UDP, to avoid overlap with the ramp-up time scenario (see section 5.2) 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.

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 in this experiment.

5.3.2 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. The test should be run both with Standard TCP and with the TCP extension under test for comparison.

5.3.3 Metrics of interest

For the decrease in cross traffic, the metrics are

1. the time taken for the TCP flow under test to increase its window to 60%, 80% and 90% of its BDP, and

2. 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 number of *cross traffic* packets dropped from the start of the transient until 100 s after the transient. This measures the harm caused by algorithms which reduce their rates too slowly on congestion.

6 Throughput- and fairness-related experiments

6.1 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. In this context a "standard TCP flow" is defined as a flow using SACK TCP [14] but without ECN [15].

The intention is for a "standard TCP flow" to correspond to TCP as commonly deployed in the Internet today (with the notable exception of CUBIC, which runs by default on the majority of web servers). This scenario quantifies this trade off.

6.1.1 Topology and background traffic

The basic dumbbell topology of section 4.1 is used with the same capacities as for the ramp-up time tests in section 5.2. All traffic in this scenario comes from the flows under test.

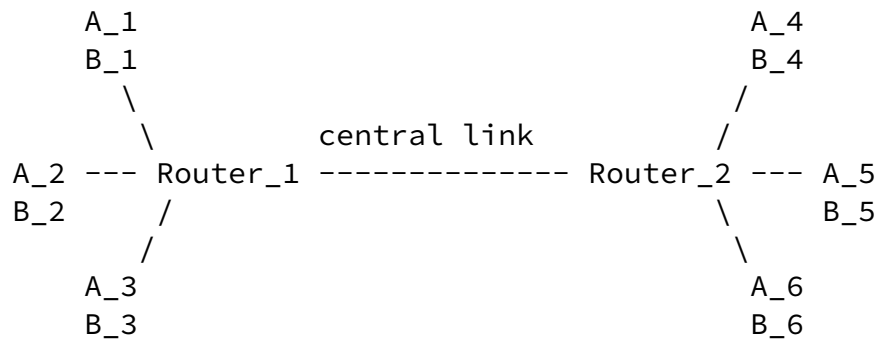


Figure 7: Impact on Standard TCP dumbbell

6.1.2 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.

These experiments use duplicate copies of the Tmix traces used in the basic scenarios (see section 4). Two offered loads are tested: 50% and 100%.

Two experiments are conducted. A BASELINE experiment where 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 under evaluation.

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.

6.1.3 Metrics of interest

load	scale	experiment time	warmup	test_time	prefill_t	prefill_si
50%	13.780346	660	104.0	510.0	45.90	14.262121
100%	5.881093	720	49.0	582.0	91.80	23.382947

Table 10: Impact on Standard TCP scenario parameters

The gain achieved by the new algorithm and loss incurred by standard TCP are given, respectively, by G=T(B)_Mix/T(B)_Baseline and L=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").

The loss, L, is analogous to the "bandwidth stolen from TCP" in [16] and "throughput degradation" in [17].

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

6.1.3.1 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 is encouraged.

6.2 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.

6.2.1 Topology and background traffic

The topology, the capacity and cross traffic conditions of these tests are the same as in section 5.2. The bottleneck buffer is varied from 25% to 200% of the BDP for a 100 ms base RTT flow, increasing by factors of 2.

6.2.2 Flows under test

We use two flows of the same protocol variant for this experiment. The RTTs of the flows range from 10 ms to 160 ms (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.

6.2.2.1 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 duration. 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., a delay less than 1-2 ms representing communication happening in a data-center, and a delay larger than 600 ms representing communication over a satellite link).

6.2.2.2 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.

6.2.3 Metrics of interest

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.

6.3 Multiple bottlenecks

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

6.3.1 Topology and traffic

The topology is a "parking-lot" topology with three (horizontal) bottleneck links and four (vertical) access links. The bottleneck links have a rate of 100 Mbps, and the access links have a rate of 1 Gbps.

All flows have a round-trip time of 60 ms, to enable the effect of traversing multiple bottlenecks to be distinguished from that of different round trip times.

This can be achieved in both a symmetric and asymmetric way (see figures 8 and 9). It is not clear whether there are interesting performance differences between these two topologies, and if so, which is more typical of the actual internet.

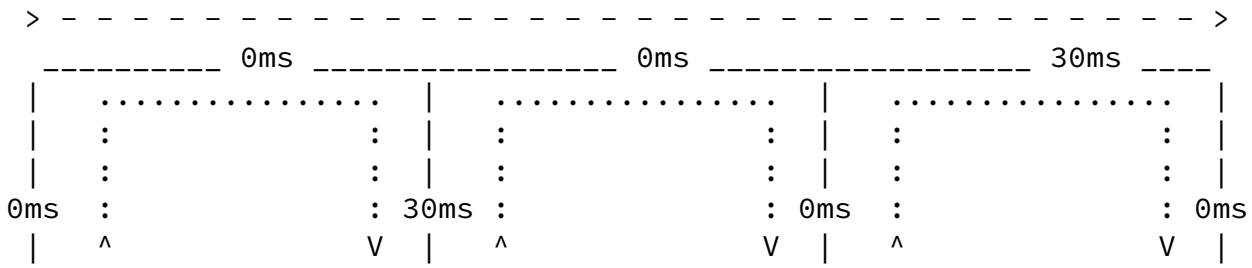


Figure 8: Asymmetric parking lot topology

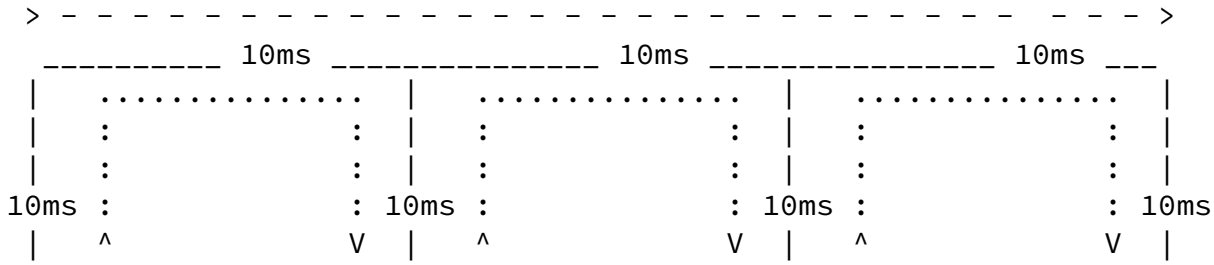


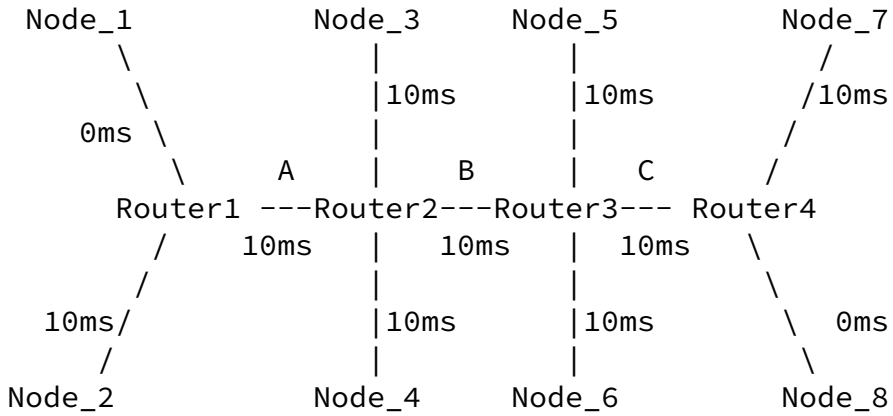
Figure 9: Symmetric parking lot topology

The three hop topology used in the test suite is based on the symmetric topology (see figure 10). Bidirectional traffic flows between Nodes 1 and 8, 2 and 3, 4 and 5, and 6 and 7.

The first four Tmix trace files are used to generate the traffic. Each Tmix source offers the same load for each experiment. Three experiments are conducted at 30%, 40%, and 50% offered loads per Tmix source. As two sources share each of the three bottlenecks (A,B,C), the combined offered loads on the bottlenecks is 60%, 80%, and 100% respectively.

All traffic uses the new TCP extension under test.

6.3.1.1 Potential Revisions



Flow 1: Node_1 <--> Node_8
Flow 2: Node_2 <--> Node_3
Flow 3: Node_4 <--> Node_5
Flow 4: Node_6 <--> Node_7

Figure 10: Test suite parking lot topology

load scale 1 prefill_t prefill_si scale 2 prefill_t prefill_si scale 3 prefill_t prefill_si total time warmup test_time

50%	tbd	tbd	tbd	tbd	tbd	tbd	t
100%	tbd	tbd	tbd	tbd	tbd	tbd	t

Table 11: Multiple bottleneck scenario parameters

Parking lot models with more hops may also be of interest.

6.3.2 Metrics of interest

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 after the warmup period. Output also includes the packet drop rate for the congested link.

7 Implementations

At the moment the only implementation effort is using the NS2 simulator. It is still a work in progress, but contains the base to most of the test, as well as the algorithms that determined the test parameters. It is being made available to the community for further development and verification through ***** url ***

At the moment there are no ongoing test bed implementations. We invite the community to initiate and contribute to the development of these test beds.

8 Acknowledgments

This work is based on a paper by Lachlan Andrew, Cesar Marcondes, Sally Floyd, Lawrence Dunn, Romaric Guillier, Wang Gang, Lars Eggert, Sangtae Ha and Injong Rhee [1].

The authors would also like to thank Roman Chertov, Doug Leith, Saverio Mascolo, Ihsan Qazi, Bob Shorten, David Wei and Michele Weigle for valuable feedback and acknowledge the work of Wang Gang to start the NS2 implementation.

This work has been partly funded by the European Community under its

Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700), by the Aurora-Hubert Curien Partnership program "ANT" (28844PD / 221629), and under Australian Research Council's Discovery Projects funding scheme (project number 0985322).

9 Bibliography

[1] L. L. H. Andrew, C. Marcondes, S. Floyd, L. Dunn, R. Guillier, W. Gang, L. Eggert, S. Ha, and I. Rhee, "Towards a common TCP evaluation suite," in Protocols for Fast, Long Distance Networks (PFLDnet), 5-7 Mar 2008.

[2] S. Floyd and E. Kohler, "Internet research needs better models," SIGCOMM Comput. Commun. Rev., vol. 33, pp. 29--34, Jan. 2003.

[3] S. Mascolo and F. Vacirca, "The effect of reverse traffic on the performance of new TCP congestion control algorithms for gigabit networks," in Protocols for Fast, Long Distance Networks (PFLDnet), 2006.

[4] D. Rossi, M. Mellia, and C. Casetti, "User patience and the web: a hands-on investigation," in Global Telecommunications Conference, 2003. GLOBECOM

[5] M. C. Weigle, P. Adurthi, F. Hernandez-Campos, K. Jeffay, and F. D. Smith, "Tmix: a tool for generating realistic TCP application workloads in ns-2," SIGCOMM Comput. Commun. Rev., vol. 36, pp. 65--76, July 2006.

[6] G. project, "Tmix on ProtoGENI."

[7] J. xxxxx, "Tmix trace generation for the TCP evaluation suite." http://web.archive.org/web/20100711061914/http://wil-ns.cs.caltech.edu/benchmark/traffic/.

[8] Wikipedia, "Fisher-Yates shuffle." http://en.wikipedia.org/wiki/Fisher-Yates_shuffle.

[9] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center tcp (dctcp)," in Proceedings of the ACM SIGCOMM 2010 conference, SIGCOMM '10, (New York, NY, USA), pp. 63--74, ACM, 2010.

[10] T. Henderson and R. Katz, "Transport protocols for internet-compat-

ible satellite networks," Selected Areas in Communications, IEEE Journal on, vol. 17, no. 2, pp. 326--344, 1999.

[11] A. Gurtov and S. Floyd, "Modeling wireless links for transport protocols," SIGCOMM Comput. Commun. Rev., vol. 34, pp. 85--96, Apr. 2004.

[12] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED," tech. rep., ICIR, 2001.

[13] 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, pp. 231--246, Feb. 2008.

[14] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP." RFC 2883 (Proposed Standard), July 2000.

[15] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP." RFC 3168 (Proposed Standard), Sept. 2001. Updated by RFCs 4301, 6040.

[16] E. Souza and D. Agarwal, "A highspeed TCP study: Characteristics and deployment issues," Tech. Rep. LBNL-53215, LBNL, 2003.

[17] H. Shimonishi, M. Sanadidi, and T. Murase, "Assessing interactions among legacy and high-speed tcp protocols," in Protocols for Fast, Long Distance Networks (PFLDnet), 2007.

[18] N. Hohn, D. Veitch, and P. Abry, "The impact of the flow arrival process in internet traffic," in Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on, vol. 6, pp. VI-37--40 vol.6, 2003.

[19] F. Kelly, Reversibility and stochastic networks. University of Cambridge Statistical Laboratory, 1979.

A Discussions on Traffic

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, a single non-greedy flow with embedded think times, or many non-greedy "thin stream" flows. process forms a Poisson process [18]. 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.

For most current traffic generators, the traffic is specified by an arrival rate for independent user sessions, along with specifications of connection sizes, number of connections per sessions, user wait times within sessions, and the like. Because the session arrival times are specified independently of the transfer times, one way to specify the load would be as A = E[f]/E[t], where E[f] is the mean session size (in bits transferred), E[t] is the mean session inter-arrival time in seconds, and A is the load in bps.

Instead, for equilibrium experiments, we measure the load as the "mean number of jobs in an M/G/1 queue using processor sharing," where a job is a user session. This reflects the fact that TCP aims at processor sharing of variable sized files. Because processor sharing is a symmetric discipline [19], the mean number of flows is equal to that of an M/M/1 queue, namely rho/(1-rho), where rho=lambda S/C, and lambda [flows per second] is the arrival rate of jobs/flows, S [bits] is the mean job size and C [bits per second] is the bottleneck capacity. For small loads, say 10%, this is essentially equal to the fraction of the capacity that is used. However, for overloaded systems, the fraction of the bandwidth used will be much less than this measure of load.

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 rho/(1-rho) active cross-traffic flows, with traffic of the specified load.

Authors' Addresses

David Hayes
University of Oslo
Department of Informatics, P.O. Box 1080 Blindern
Oslo N-0316
Norway

Email: davihay@ifi.uio.no

David Ros
Institut Mines-Telecom / Telecom Bretagne
2 rue de la Chataigneraie
35510 Cesson-Sevigne
France

Email: david.ros@telecom-bretagne.eu

Lachlan L.H. Andrew
CAIA Swinburne University of Technology
P.O. Box 218, John Street
Hawthorn Victoria 3122
Australia

Email: lachlan.andrew@gmail.com

Sally Floyd
ICSI
1947 Center Street, Ste. 600
Berkeley CA 94704
United States