HighSpeed TCP for Large Congestion Windows

draft-floyd-tcp-highspeed-00.txt

Sally Floyd

July 17, 2002

TSVWG, Yokohama IETF

HighSpeed TCP:

Joint work with Sylvia Ratnasamy and Scott Shenker at ICIR.

Additional investigations with Evandro de Souza and Deb Agarwal.

One experimental implementation in progress from Tom Dunigan.

URL: http://www.icir.org/floyd/hstcp.html

HighSpeed TCP: The problem.

- TCP's average congestion window is roughly $1.2/\sqrt{p}$ packets.

- Given 1500-byte packets and a 100 ms RTT, filling a 10 Gbps pipe would correspond to a congestion window of $W = 83,333$ packets.
  - At least 1.6 hours between packet drops.

- We can do better, even with only the current feedback from routers.

HighSpeed TCP: Is this a pressing problem?

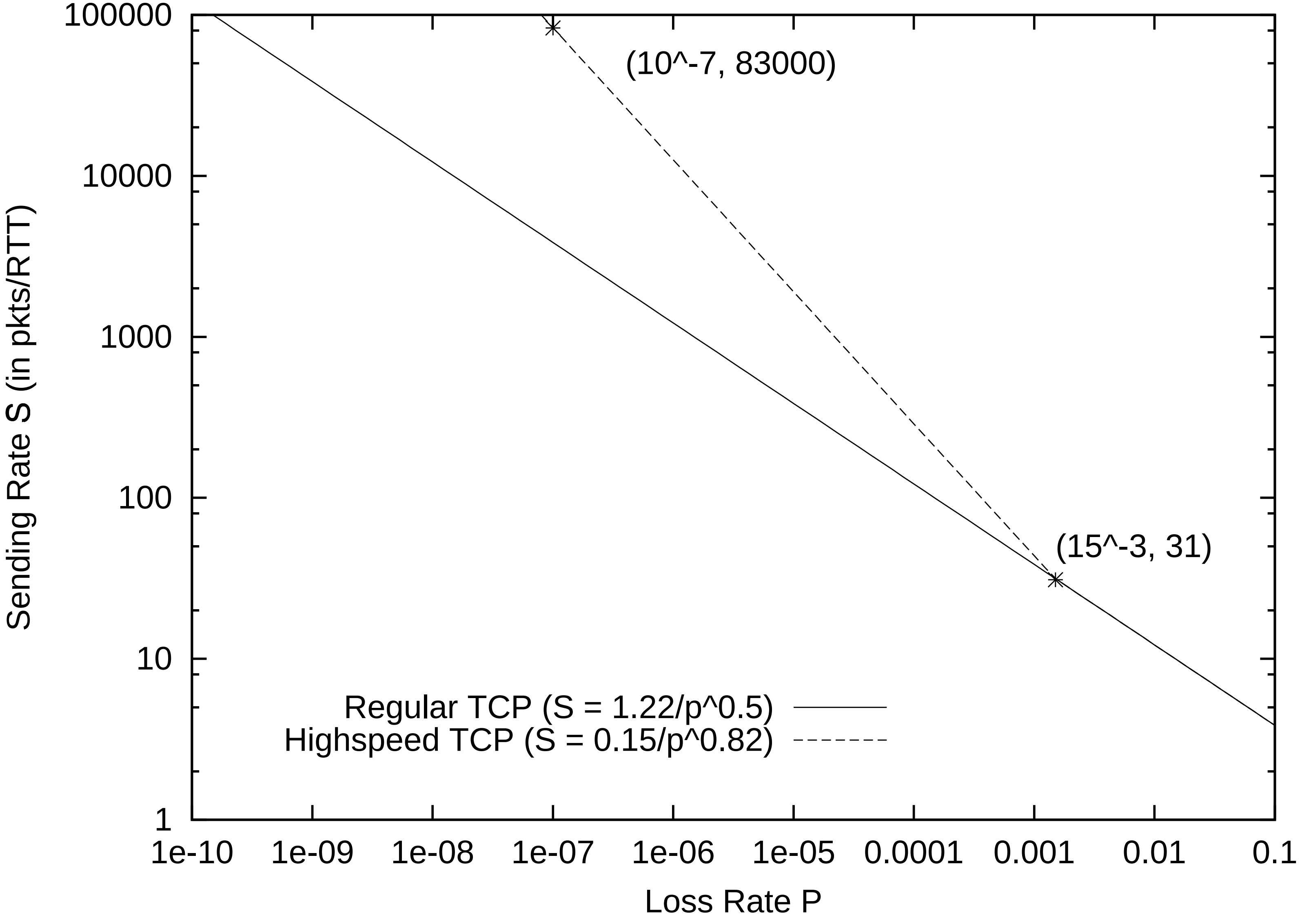
- Nope. In practice, users do one of the following:
  - Open up N parallel TCP connections; or
  - Use MulTCP (roughly like an aggregate of N virtual TCP connections).

- However, we think it is possible to do much better, with:
  - Better flexibility (no N to configure);
  - Better scaling;
  - Better slow-start behavior;
  - Competing more fairly with current TCP

(for environments where TCP is able to use the available bandwidth).

- HighSpeed TCP can be thought of as only behaving as an aggregate of N TCP connections at higher congestion windows.

HighSpeed TCP: use a modified response function.



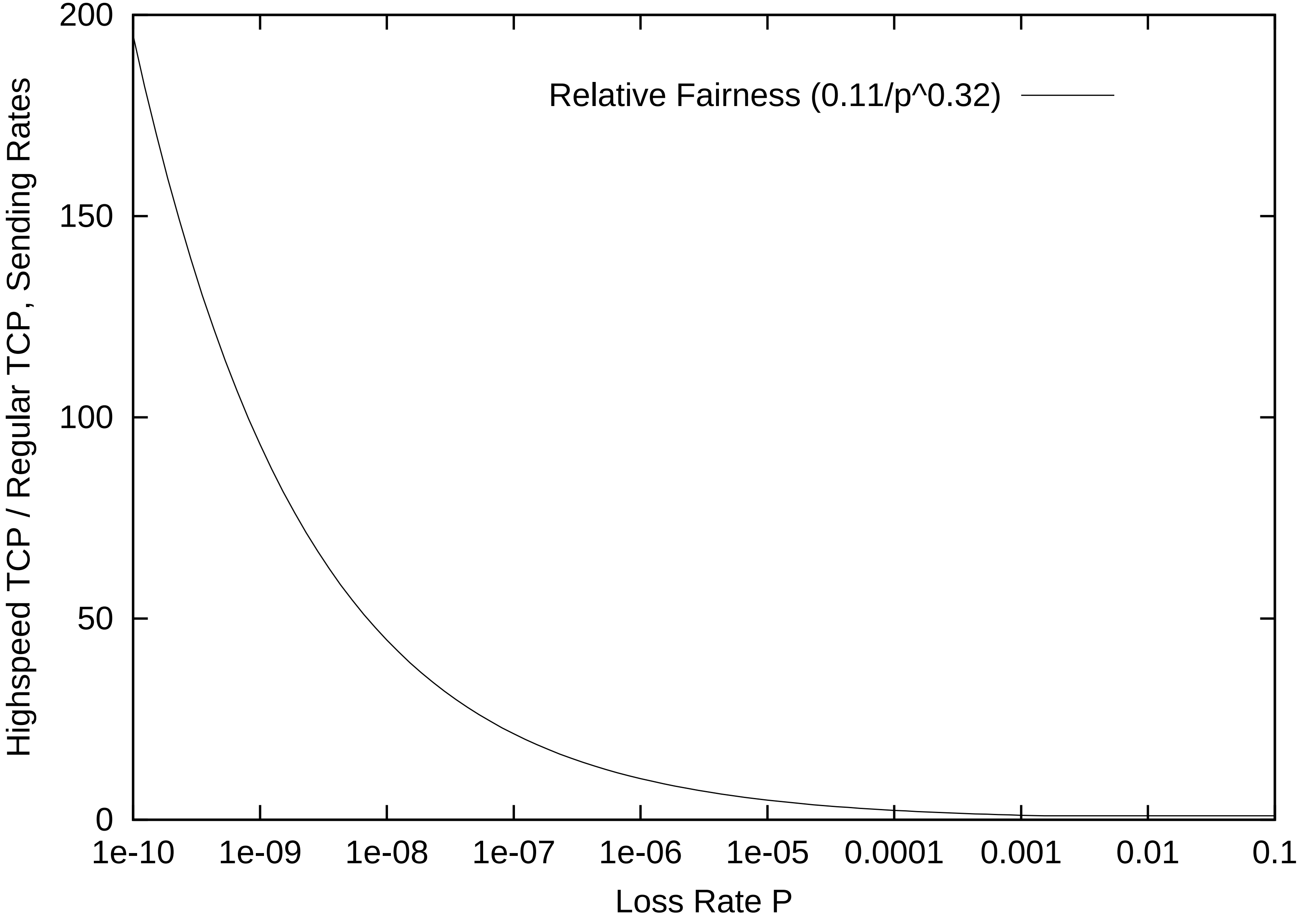
HighSpeed TCP: Simulations in NS.

- ./test-all-tcpHighspeed in tcl/test.

- The parameters specifying the response function:
  - Agent/TCP set low_window_ 31
  - Agent/TCP set high_window_ 83000
  - Agent/TCP set high_p_ 0.0000001

- The parameter specifying the decrease function at high_p_:
  - Agent/TCP set high_decrease_ 0.1

HighSpeed TCP: Relative fairness.



The Gory Details:

```
   w  a(w)  b(w)
----  ----  ----
  38     1  0.50
 118     2  0.44
 221     3  0.41
 347     4  0.38
 495     5  0.37
 663     6  0.35
 851     7  0.34
1058     8  0.33
1284     9  0.32
1529    10  0.31
1793    11  0.30
2076    12  0.29
2378    13  0.28
```

Conclusions:

- This is a proposal in its initial stages.

- We are presenting it here for feedback from the community.

- We would love feedback from experiments.

- My own view, not surprisingly, is that something like this is the fundamental correct thing to do, given the need for backwards compatibility and incremental deployment.
    - Another view would be that this is a mangled hack...

Limited Slow-Start for TCP with Large Congestion Windows

draft-floyd-tcp-slowstart-00.txt

Sally Floyd
July 17, 2002
TSVWG, Yokohama IETF

HighSpeed TCP: modifying slow-start:

- Slow-starting up to a window of 83,000 packets doesn't work well.
  - Tens of thousands of packets dropped from one window of data.
  - Slow recovery for the TCP connection.

- The answer:
  - Agent/TCP set max_ssthresh_ N
  - During the initial slow-start, increase the congestion window by at most N packets in one RTT.

The pseudocode:

```
For each arriving ACK in slow-start:
  If (cwnd <= max_ssthresh)
     cwnd += MSS;
  else
     K = int(cwnd/(0.5 max_ssthresh));
     cwnd += int(MSS/K);
```

Experimental results:

- Tom Dunigan has added the slow-start mod to the Linux 2.4.16 Web100 kernel.

- The initial results look promising.

- http://www.csm.ornl.gov/ dunigan/net100/floyd.html

Quick-Start for TCP and IP

draft-amit-quick-start-00.txt

Amit Jain and Sally Floyd.

From an initial proposal by Amit Jain.

Quick-Start: Larger Initial Sending Rates

- An IP option in the SYN packet gives the sender’s desired initial sending rate.
  - Routers on the path decrement a TTL counter,
  - and decrease the allowed initial sending rate, if necessary.

- If all routers on the path participated:
  - The receiver tells the sender the allowed initial sending rate in the SYN/ACK packet, in the transport header.

- The sender gets an RTT measurement when it gets the SYN/ACK from the receiver, and can set the initial congestion window.

The Quick-Start Request Option for IPv4

```
     0          1          2          3          4
+----------+----------+----------+----------+----------+
| Option   | Length=5 |  QS TTL  | Initial  | QS Nonce |
|          |          |          | Rate     |          |
+----------+----------+----------+----------+----------+
```

Assumptions:

- In order for best-effort connections to use initial windows higher than four segments, explicit feedback from all of the routers along the path would be required.

- A router should only allow an initial sending rate higher than the transport protocol’s default initial rate if the router is significantly underutilized.

- No per-flow state is kept at the router.

Questions:

- Would be benefits of the Quick-Start mechanism be worth the added complexity?
  - SYN and SYN/ACK packets not taking the fast path in routers.

- Is there a compelling need to add some form of congestion-related feedback from routers such as this (in addition to ECN).

- If so, are there other mechanisms that would be preferable to Quick-Start?

Architectural sub-themes:

- A goal of incremental deployment in the current Internet.

- Steps must go in the fundamantally correct, long-term direction, not be short-term hacks.

- Robustness in heterogeneous environments valued over efficiency of performance in well-defined environments.

- A preference for simple mechanisms, but a skepticism towards simple traffic and topology models.

- Learning from actual deployment is an invaluable step.

- The Internet will continue to be decentralized and fast-changing.