TBIT: TCP Behavior Inference Tool

Jitendra Padhye

Sally Floyd

AT&T Center for Internet Research at ICSI

(ACIRI)

http://www.aciri.org/tbit/

Outline of talk

- Motivation
- Description of the tool
- Results
- Future work

Motivation

- TCP handles a majority of today’s Internet traffic
- Understanding TCP behavior is important: OS vendors, ISPs
- RFCs and other documents specify how TCP *should* behave

Needless to say ....

Implementations do not always match specifications!

Example

- Initial window used by TCP: amount of data sent out in a “burst” before any ACKs are received.
- RFC 2414: min (4*MSS, max (2*MSS, 4380 bytes))
- MSS 512 $\Rightarrow$ burst of 2000 bytes
- We have found TCPs (www.uwm.edu) that send 8000+ bytes with MSS of 512!
- Large bursts of packets $\Rightarrow$ buffering problems, loss, delays.

How to detect misbehaving TCPs

- Passive detection: Vern Paxson analyzed thousands of `tcpdump` traces and detected several anomalies (1996-97)

- Passive detection has limitations

- TBIT *actively* probes TCP stacks at web servers to test behavior

How it works: The basic idea

- Send “fabricated” TCP packets over raw IP sockets.
- Host firewall prevents kernel from seeing response packets.
- BPF delivers blocked packets to user process.
- Net effect: a user-level, user-controllable TCP, without kernel changes.

Based on “Sting” project at Univ. of Washington by Stefan Savage

Example

Determine TCP initial window used by a web server.

- Send SYN. Wait to receive SYN-ACK.
- Send HTTP GET request for “/”
- Do not ACK any incoming packets.
- Wait until first retransmission.
- Initial window $=$ Max. sequence number received.

Can check with several MSS values!

Tests implemented so far

- **Handshake tests:** Timestamp used? SACK-capable?

- **Congestion response:** Reduce congestion window? NewReno/Reno/Tahoe?

- **SACK:** Construct SACKs correctly? Respond to SACKs correctly?

- **Other:** Initial window? ECN-capable?

Results: Background

- Two lists of web sites:
  - 100hot.com: approx. 200 unique IP addresses.
  - Trace from an ISP proxy (courtesy Dax Kelson): approx. 27,000 unique IP addresses.
- Tests repeated at least twice at different times.
- Results reported only if consistent across runs.
- Not allowed to run NMAP: hard to correlate with OS

Initial Window

- 638 tests from Proxy list. 10/12/00. MSS 512.
- Results:
  - 4 hosts had initial windows of 8000+ bytes (17 packets with MSS 512, 80 packets with MSS 100). www.uwm.edu(2), endeavor.med.nyu.edu, www.monash.com.
  - 12% hosts reported initial windows of > 4 packets.

Timestamps

- Timestamps enable better estimation of RTO
- 136 completed tests from Hot list. 7/15/00.
- 25% of the servers tested did not use timestamps.

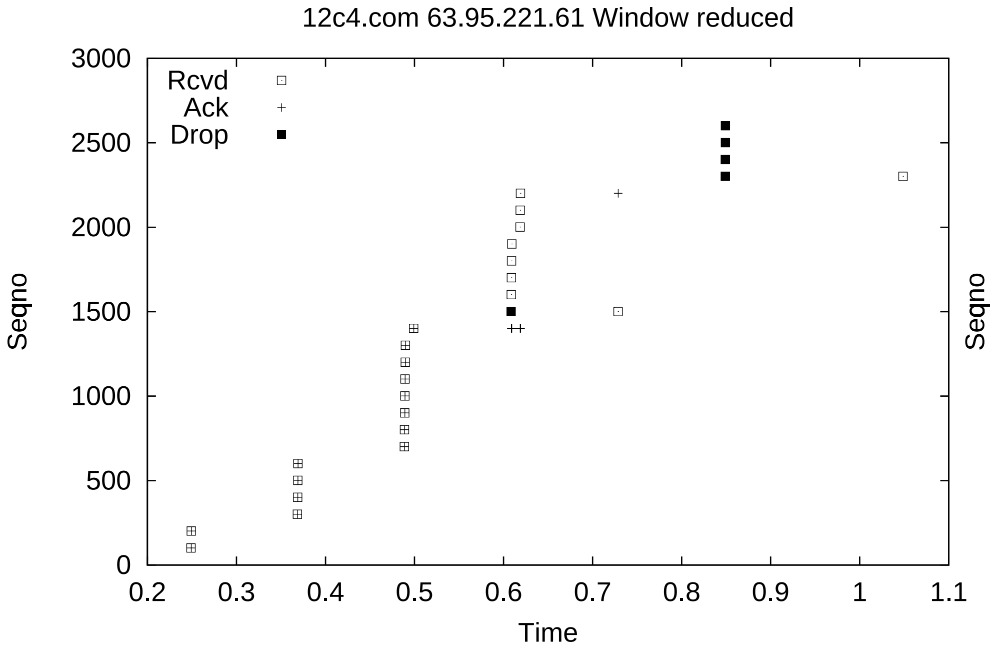
  For example: www.ebay.com, www.hp.com
- AIX hosts send garbage. Problem reported to IBM,

  fix in works.
- Have not tested if timestamps are used correctly.

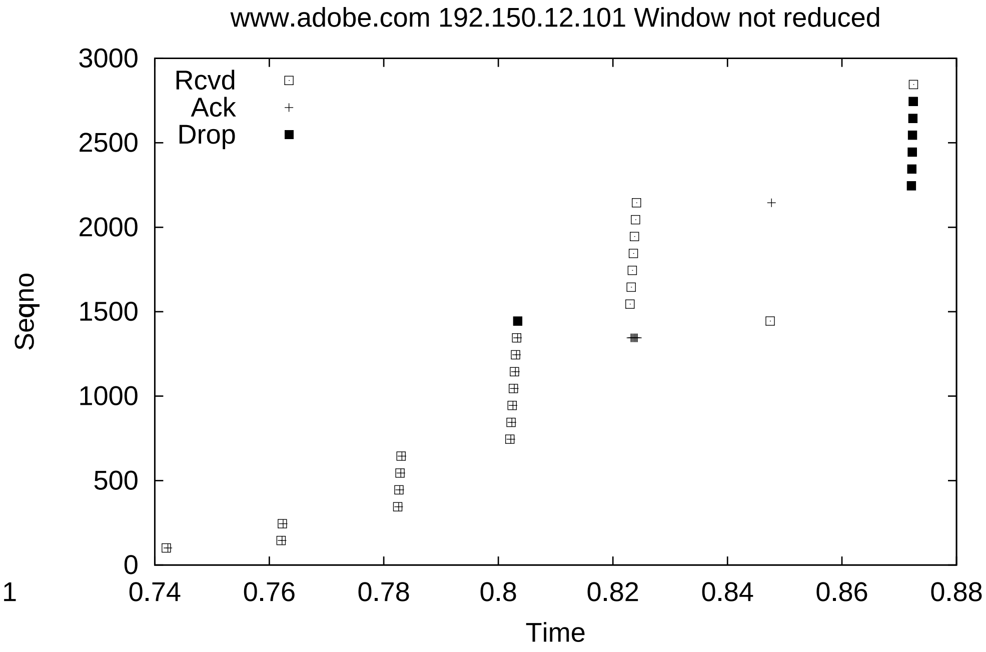
Congestion window reduction

- TCP expected to cut sending rate in half on packet drop. Essential to the stability of the Internet!
- 6485 tests from Proxy list. 10/19/00. MSS 100.
- Drop one packet when window reaches 8, and count outstanding packets.
- Results: 72 hosts (1.11%) reduced congestion window to 7 packets. For example: www.adobe.com, members.zdnet.com

Congestion window reduction: Examples



Window reduced



Window not reduced

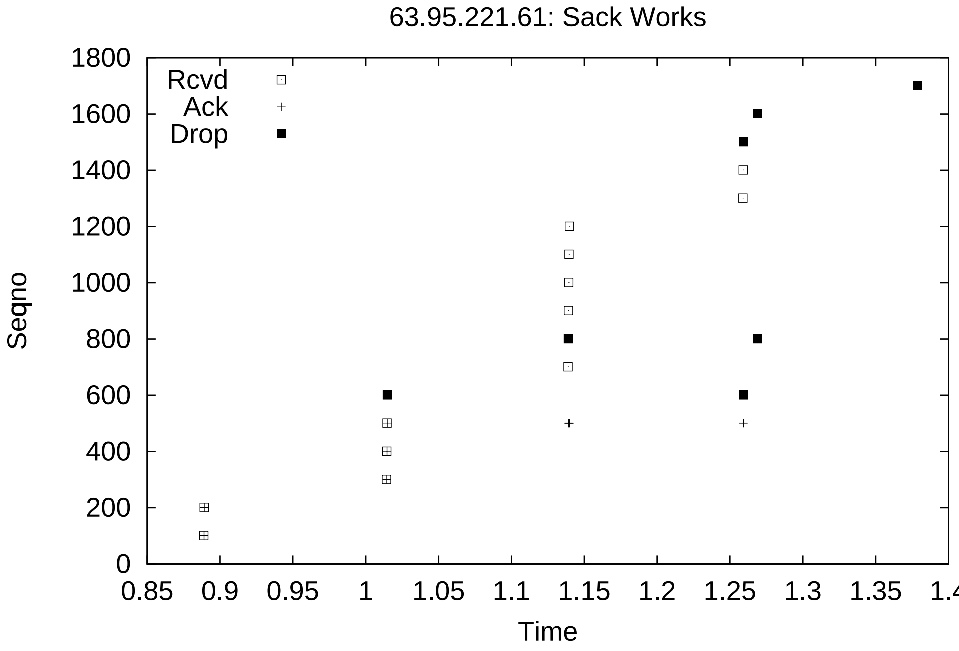
Claim SACK-capable

- SACK (Selective Acknowledge Ment) reduces RTOs, improves performance.
- 136 tests from Hot list. 7/15/00.
- Results:
  - 42% not SACK-capable. For example: home.netscape.com, www.cnn.com
  - Many SACK-capable hosts do not seem to use SACKs correctly.

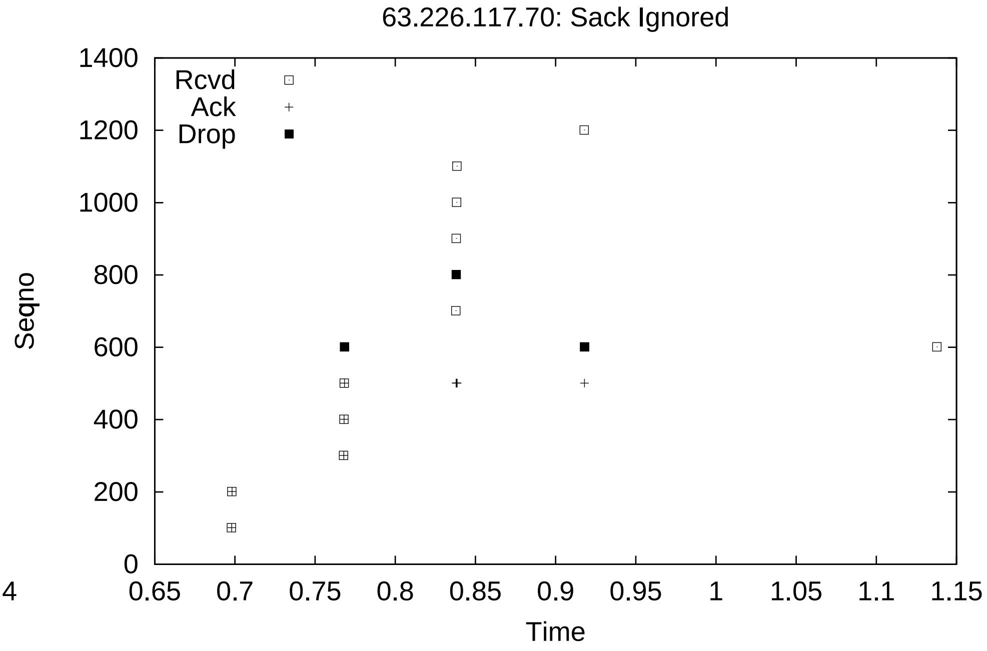
Correct SACK usage

- 2278 tests from Proxy list. 10/18/00. MSS 100.
- Drop packets 6 and 8, and see if they are retransmitted together.
- Results: Only about 6% of the hosts used SACK correctly.

SACK Usage examples


63.95.221.61: Sack Works

Correct usage


63.226.117.70: Sack Ignored

SACK info ignored

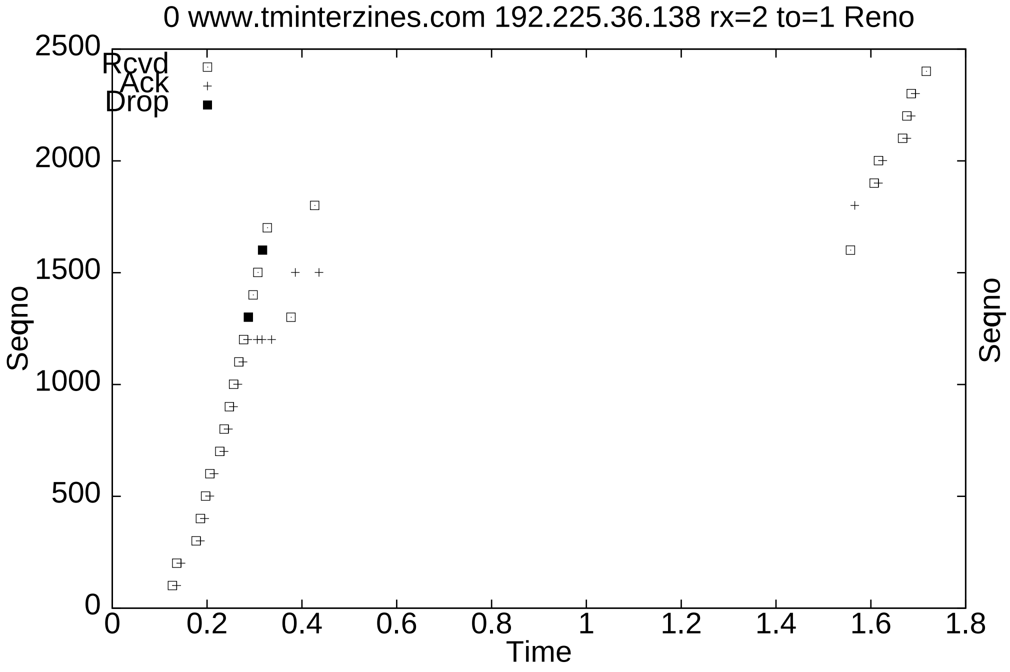
ECN

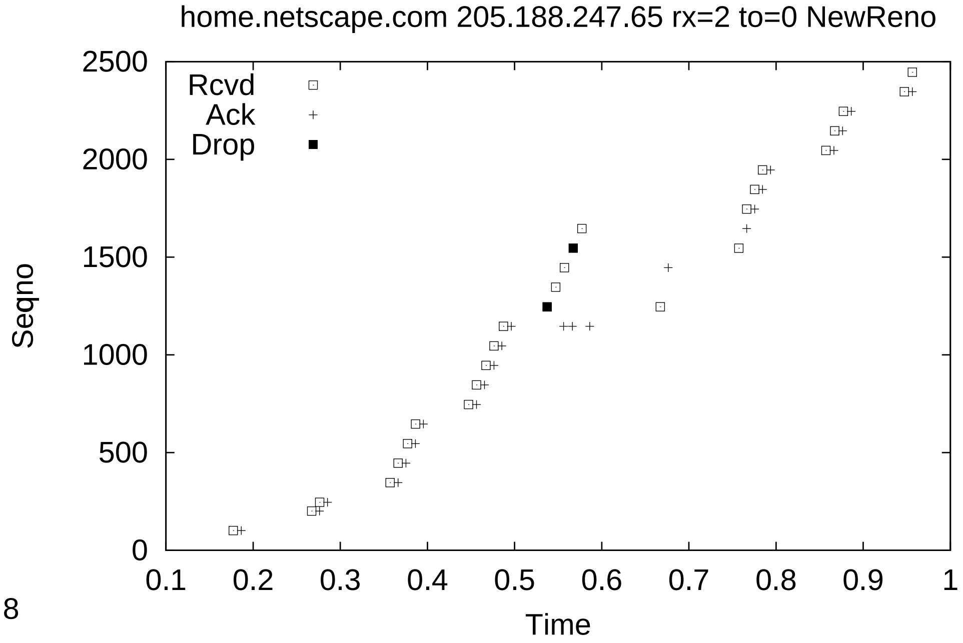
- Negotiated during SYN/ACK exchange.
- 26,447 tests from Proxy list.
- 8% of web servers unreachable from ECN-capable clients.
- Sometimes, problem with Cisco Local Director (Dax Kelson). Fixed.

TCP flavor

- 136 tests from Hot list. 7/15/00. MSS 100.
- Results:
  - 61% NewReno, 22% Reno, rest Tahoe.
  - Microsoft servers took timeout for every packet loss for small transfers. Problem reported to Microsoft, fix will be available in next version of Windows 2000.

TCP flavor: NewReno vs. Reno

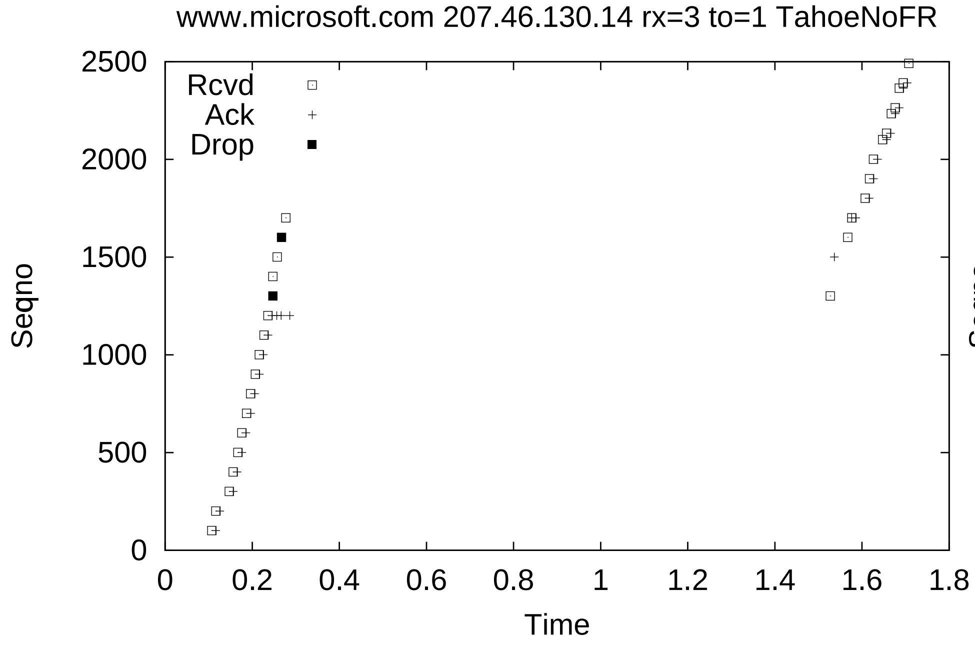




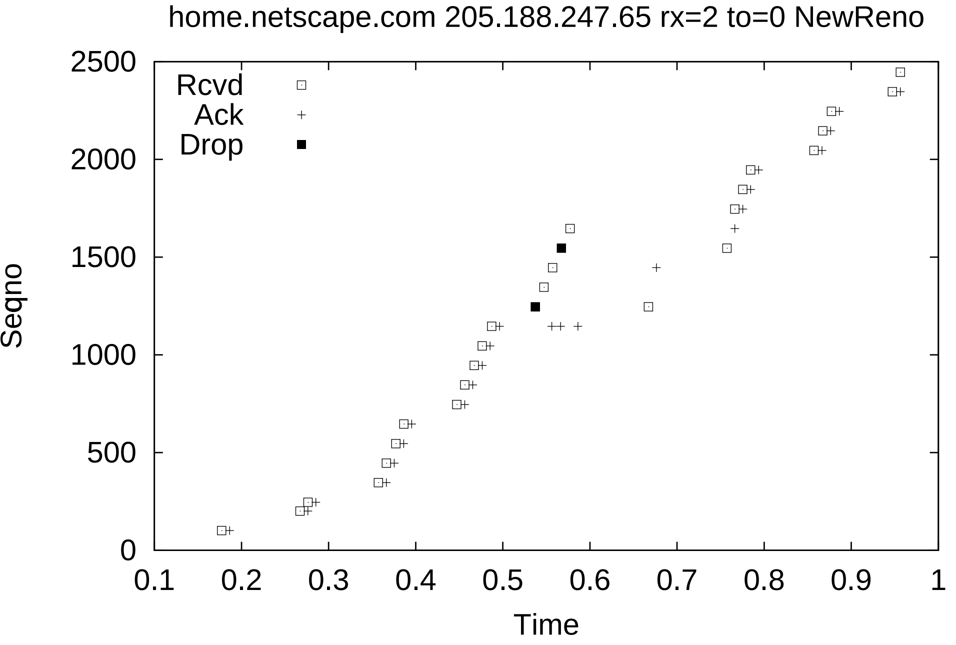
Reno

NewReno

TCP flavor: NewReno vs. Tahoe



Tahoe (No Fast Retransmit)



NewReno

Difficulties

- Too few packets: set smaller MSS?
- Lost packets: repeat test multiple times.
- Multiple hosts answering same IP address:
  non-repeatable results?
- No easy way to test without a web server.

Future Work

- Full conformance checking for TCP.
- Automatic generation of simulator models.
- Extend this approach to investigate other behaviors of the Internet infrastructure
- Suggestions? Beyond TCP?
- Run NMAP?

Finally ....

- Source code, detailed results and a preliminary report are available: **http://www.aciri.org/tbit/**
- We encourage people to use the software and add their own tests.