Preliminary Empirical Study of BTC Tools

Mark Allman, NASA GRC/BBN

IPPM WG Meeting

August 2000

Background

- We have a draft BTC framework around which different BTC methodologies can be built:
  - TReno draft (Mathis)
  - *cap* – no draft yet
- The basic idea of a BTC tool is to measure the throughput a flow utilizing standard congestion control could obtain if flowing over the given network path.
  - A tool that does not rely on the underlying TCP is very attractive because quirks in TCP stacks do not impact the results.

But, A Question...

- A question remains as to whether or not a tool producing packets according to TCP's congestion control algorithms can predict TCP performance.
  - Intuitively – yes!
  - Empirically – not sure yet

cap Overview

- Consists of sender (*cap*) and receiver (*capd*) processes.
- Use UDP for both "data" and "ACK" packets
- Advantages:
  - Allows good control over all behavior (sender loss recovery strategy, delayed ACK behavior, etc.)
  - The "ACKs" are cumulative, just as in TCP
    - Data loss/reordering can be disambiguated from ACK loss
- Disadvantages:
  - Must have access to the receiver to run *capd*

TReno Overview

- Consists of only a sending process
- Can use UDP or ICMP packets to induce the receiver into "ACKing" (ala ping or traceroute)
- Advantages:
  - Does not require access to the receiver host
- Disadvantages:
  - ACK loss is the same as data loss since only specific data segments are ACKed (i.e., no cumulative ACK)
  - No control over the receiver's behavior
    - We can emulate things like delayed ACKs, SACKs, etc.
    - The receiver cannot do things like take bandwidth estimates (although this is not currently a problem)

Methodology

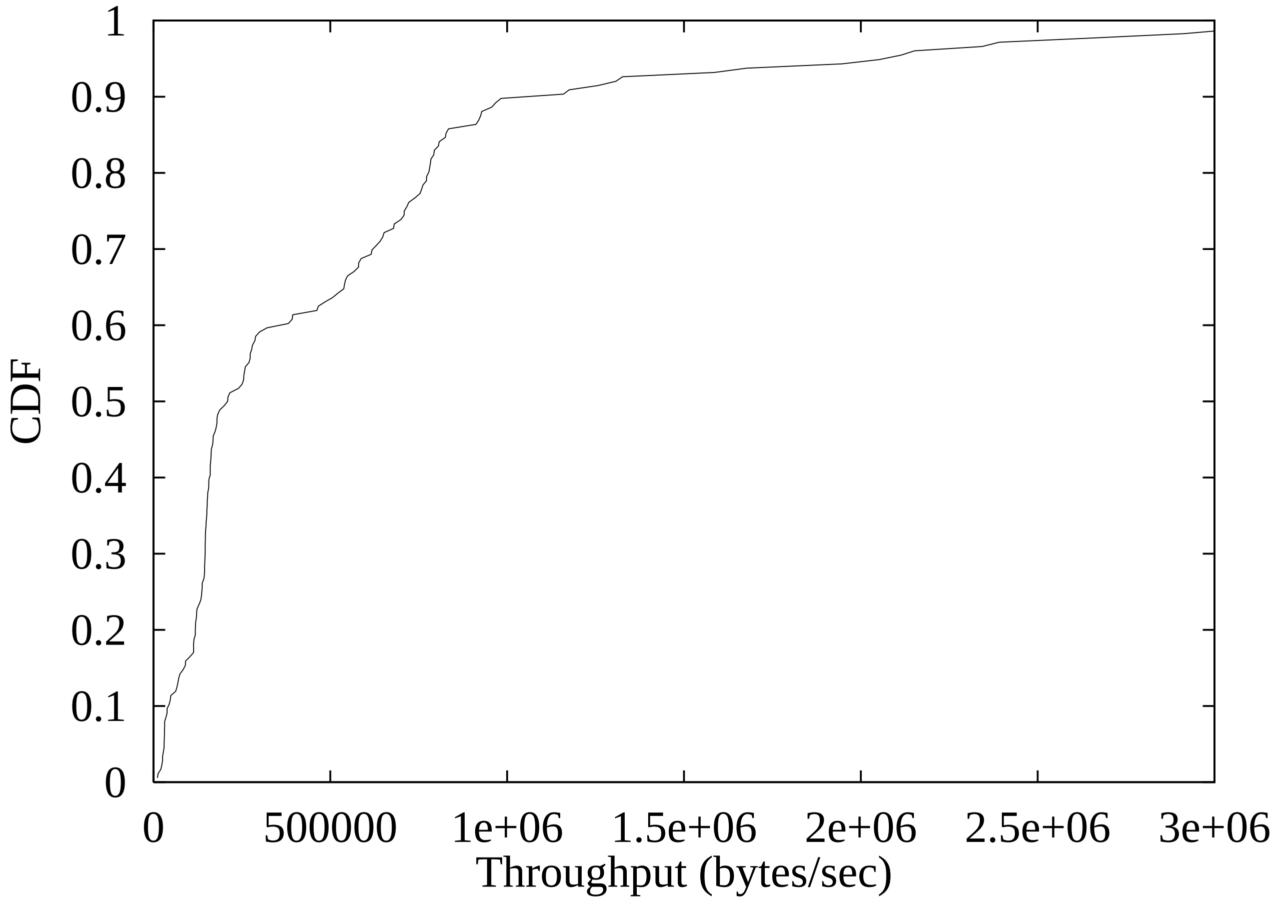
- Used a subset of the NIMI sites
  - Used 31 sites (mostly FreeBSD, a couple NetBSD)
  - Hosts excluded due to configuration issues, not network issues.
- One measurement consists of two back-to-back transfers
  - Each transfer is 30 seconds
  - We randomly pick TCP, *cap* or TReno for each transfer
- We have XXX measurements over the course of roughly 3 days

Methodology (cont.)

- The TCP used was the stock version used by the particular operating system
  - We increased the socket buffer sizes to roughly 200 KB
    - I.e., we used window scaling and timestamps (also used in *cap* and TReno)
  - We disregarded measurements made with smaller socket buffer sizes.

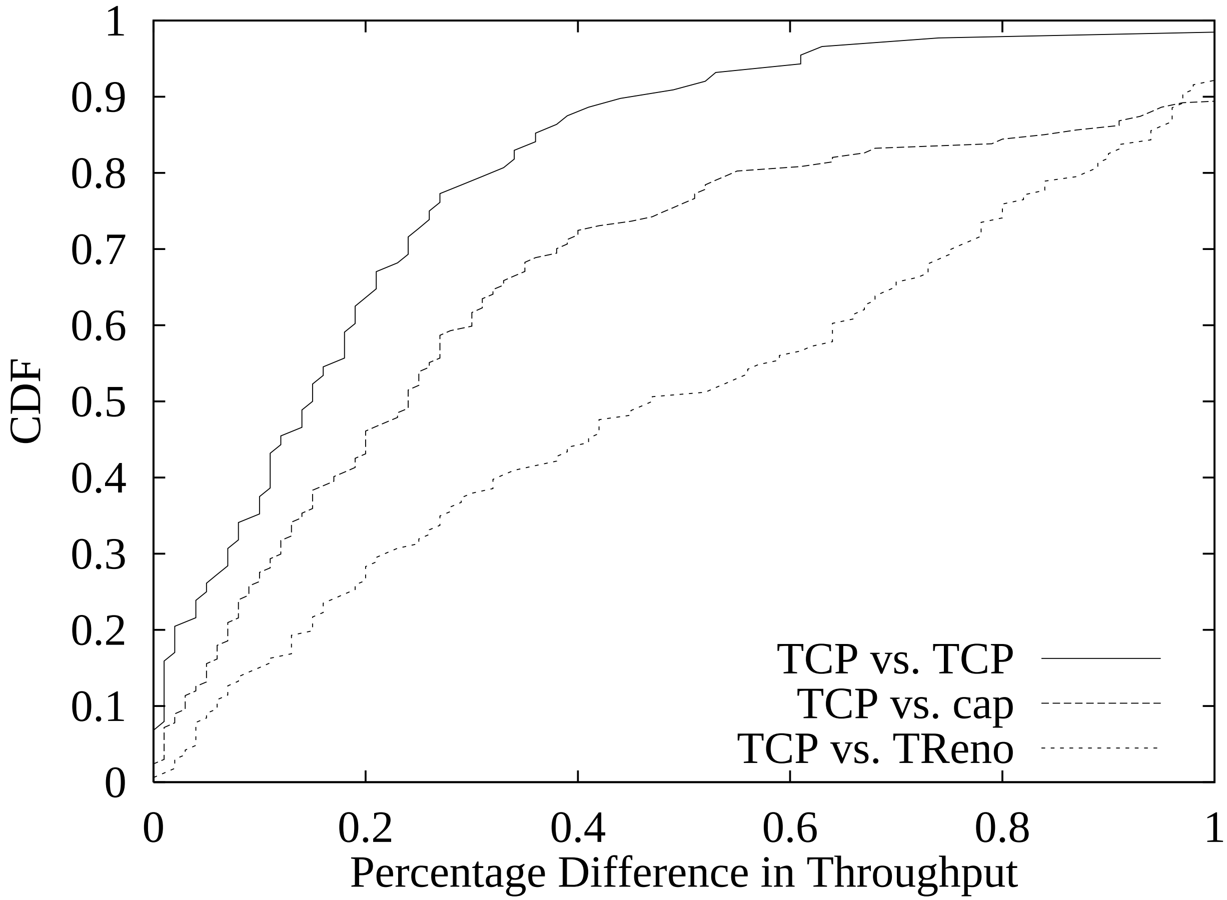
Results

- Throughput distribution of TCP transfers:



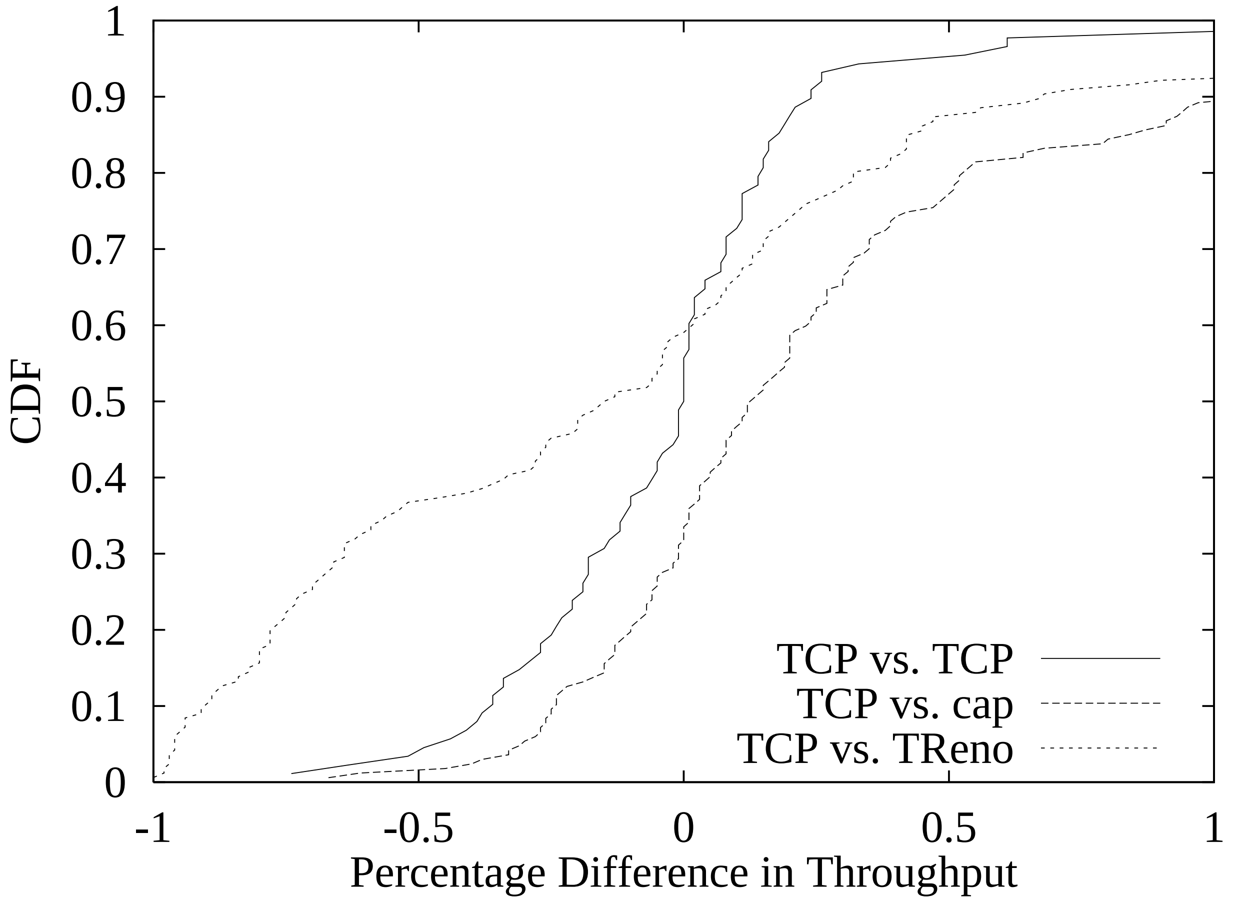
Results (cont.)

- Difference in throughput, take 1:



Results (cont.)

- Difference in throughput, take 2:

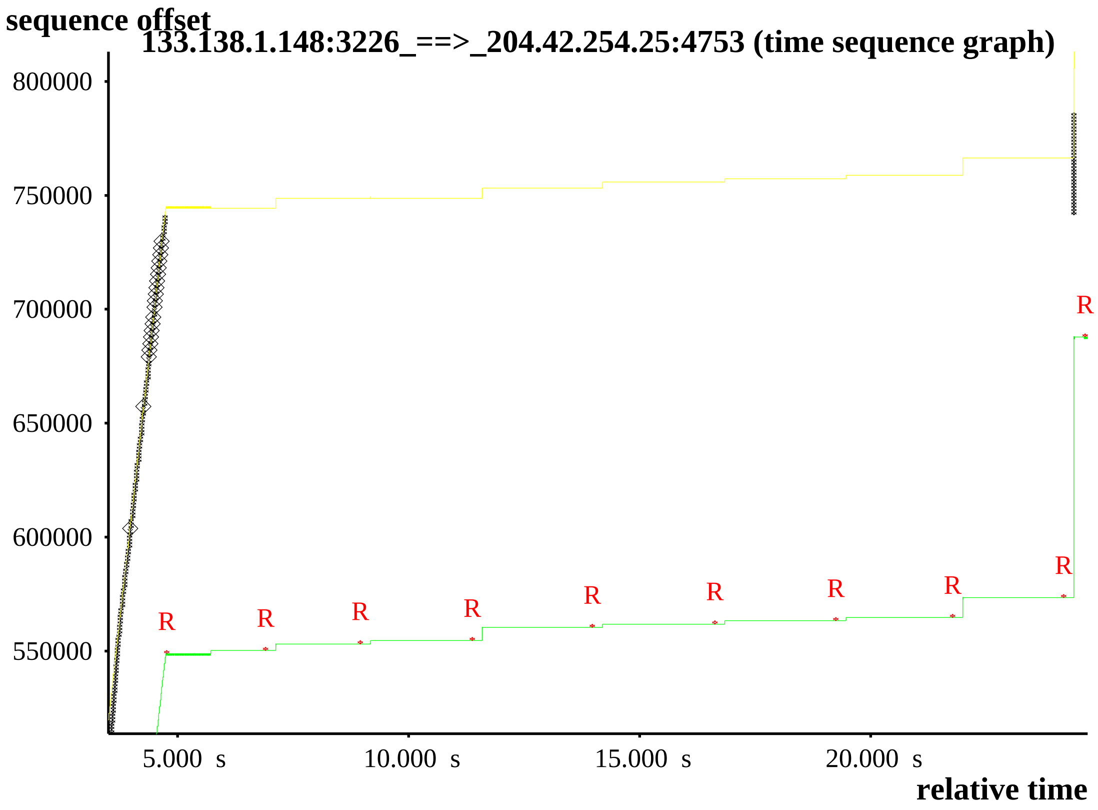


Results (cont.)

- Why the difference?
  - *cap*'s initial RTO is different from TCP's (3 secs as opposed to 6 secs)
  - *cap*'s RTO ends up being a bit longer than TCP's in some cases
    - Likely indicating a bug in *cap*'s heartbeat timer emulation code
  - BSD TCP bugs

Results (cont.)

- BSD TCP bug:



Conclusions

- Not definite conclusions... just leanings...
  - BTC is likely possible with a sender/receiver measurement methodology.
  - Whether or not we can make a sender-only methodology work is an open question.

Future Work

- Continue to crunch the data to determine to what degree *cap* and/or TReno need to be fixed to better emulate TCP behavior
  - Keeping in mind that some of the differences might not be bugs, but rather legal diversity, as allowed by RFC 2581.
- Run some measurements using different TCP stacks to figure out what sort of variation exists between currently existing implementations.
  - I.e., *cap* and/or TReno might be different from BSD TCP, but no more so than another implementation of TCP.

Future Work (cont.)

- Give *cap* the ability to work as a sender-side only tool.
  - Allow a more direct comparison between the sender-only approach and the sender/receiver approach currently employed.

IPPM Implications

- What do we do with BTC in IPPM?
  - *I* believe the framework is essentially sound at this point and should be forwarded to the IESG after a light editing pass.
  - I think a document based around the BTC framework and the current *cap* tool is appropriate in the near-term.