Model Based Metrics

draft-mathis-ippm-model-based-metrics-00.txt

Matt Mathis
mattmathis@google.com

IETF 85 IPPM
6-Nov-2012

Bulk Transport Capacity

- Expected single stream TCP performance
  - What is realistic share of the network under load?
- NOT
  - Maximum raw capacity
    - E.g. if all other traffic gives way
    - Typical of multiple concurrent connections
  - Available capacity (idle or head room)
    - E.g. if measurement traffic gives way
- This was one of the main motivations for IPPM
  - BOF at IETF 32 (April 1995)
  - Known to be a very hard problem
  - Hints in the Charter, RFCs 2330 and 3148

BTC is hard for a reason

- TCP and all transports are complicated control systems
    - TCP causes self inflicted congestion
    - Governed by "equilibrium like" behaviors
    - Changes in one parameter are offset by others
- Every component effects performance
    - All sections of the path
    - End systems & middle boxes (TCP quality)
    - Routing anomalies and path length
- The Meta-Heisenberg problem
    - TCP "stiffness" depends on RTT
    - The effects of "shared congestion" depend on
        - Bottlenecks and RTT of the other cross traffic
    - Can't generally measure cross traffic with 1 stream

A another way to do BTC

- Need to "open loop" TCP
  - Prevent self inflicted congestion
  - Prevent circular dependencies between parameters
- Independently control traffic
  - Defeat congestion control (generally slow down)
  - Measure path properties section by section
  - Compare to properties required per models
  - E2E paths passes only if all sections pass all tests

An example

- Goal: 1 MByte/s BTC over a path that is
  - 10 Mb/s raw capacity (~1.2 MByte/s)
  - 20 ms, 1500 Byte MTU
  - Invert TCP performance model [MSMO97]

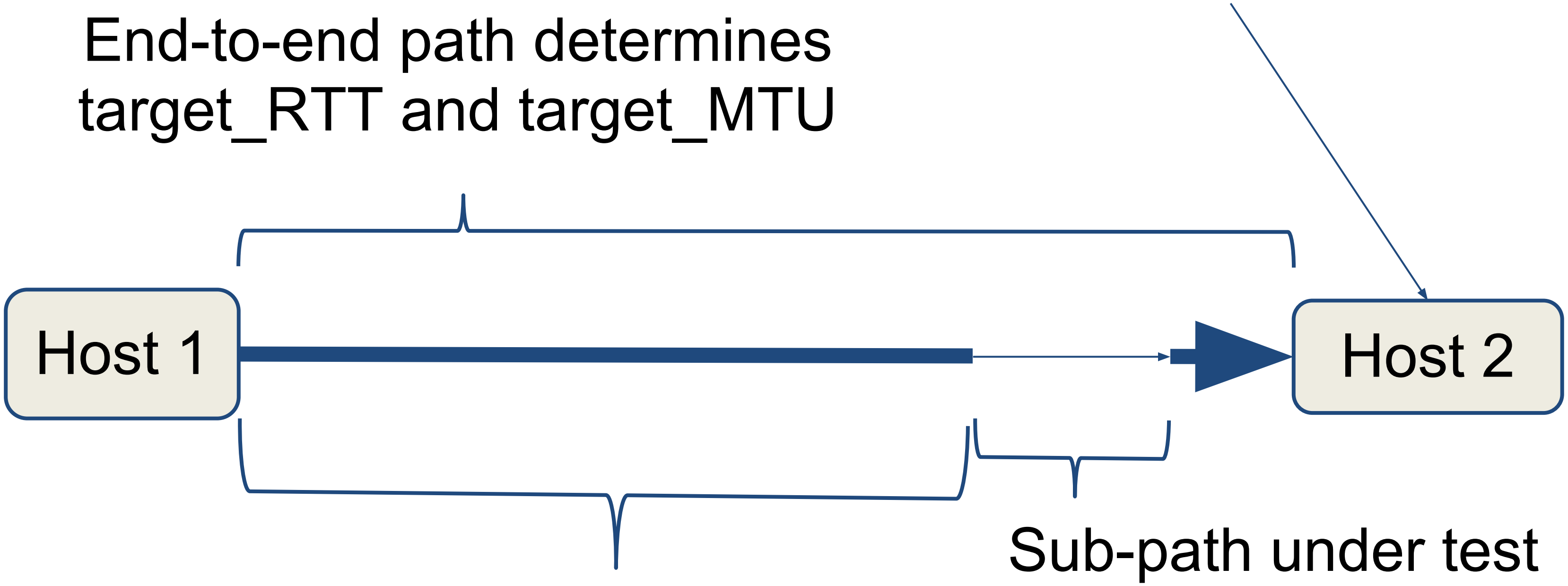
$$Rate = \left(\frac{MSS}{RTT}\right)\frac{C}{\sqrt{p}}$$

  - Yields loss probability budget less than 0.3%
  - Test each short section at 1 MByte/s
  - Fails if total loss probability is more than 0.3%
  - But passing this test is not sufficient
    - Because the link can still fail in other ways
- This is a pass/fail test, not a measurement

The pieces (simplified)

The "application" determines target_rate

End-to-end path determines target_RTT and target_MTU



Sub-path under test

Rest of path is assumed to be effectively ideal

Must meet constraints determined by models based on target_rate, target_RTT and target_MTU

Additional parameters

- Per sub-path
  - subpath_RTT and subpath_rate
- "run_length" number of packets between losses
  - e.g. 1/p
- Support for derating
  - Allow some parameters to be relaxed
  - Some models are overly conservative
  - Also a migration/bootstrapping strategy

Common Calculations

- target_pipe_size = target_rate*target_RTT/target_MTU
    - The # of packet to reach the knee
- reference_target_run_length = (3/2)(target_pipe_size^2)
    - The conservative # of packets between losses
- target_run_length = [Documented alternate model]
    - More pragmatic target run length

Property 1: CBR loss rate

- Send traffic at specified target_rate
  - measured_run_length > target_run_length
- Also support stealth mode e.g.
  - Send at 1% of target rate, monitor run_length
- To use TCP, clamp cwnd to control the rate
  - Use RFC 4898, etc to measure loss probability
  - Test is "inconclusive" if rate is not accurate
  - (If fail, then buggy TCP's cause false fails)

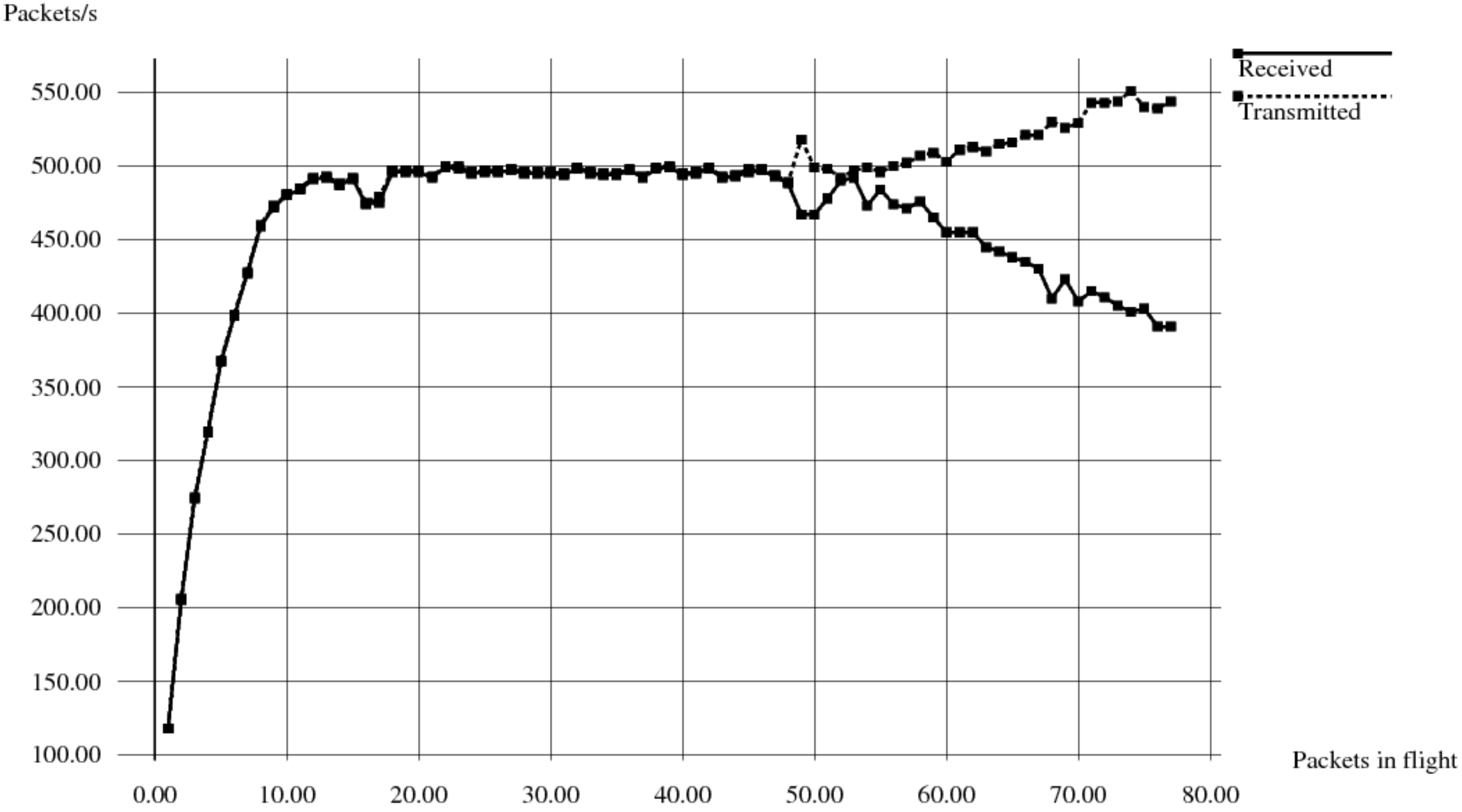
Property 2: Queue burst capacity

- Slowstart burst test:
  - Send target_pipe packets
  - At a rate 2*subpath_rate
  - Observed run_length < (derated)target_run_length
  - (Otherwise slowstart exits prematurely)
- NIC TSO burst test:
  - Send MIN(42, target_pipe) packets
  - At server interface rate (e.g. 10 Gb/s)
  - Observed run_length < (derated)target_run_length
  - (Otherwise ubiquitous TSO suffers)
- May need other burst size/rate scales too
  - e.g. TCP restart after idle

Property 3: Stable at onset of congestion

- Must be well behaved at the onset of congestion
    - Gradual onset of queueing delay and/or
    - Gradual onset of loss (e.g. AQM)
- See for example:
    - M. Mathis "Windowed Ping: An IP Level Performance Diagnostic", Proceedings of INET'94.
    - M. Mathis, J. Heffner, P. O'Neil, P. Siemsen, "Pathdiag: Automated TCP Diagnosis", PAM 2008.

Queuing example (From "Windowed Ping")



Additional test: Cross traffic/unidentified load

- E.g. Bots or viruses contaminating measurements
- SNMP using "trigger" technique from: B. Tierney et al, "Self Configuring Network Monitor (SCNM)"
    - UDP packet containing a "magic" pattern
    - Causes a SNMP report back to the sender
- Many other techniques might be possible

Possible additional tests

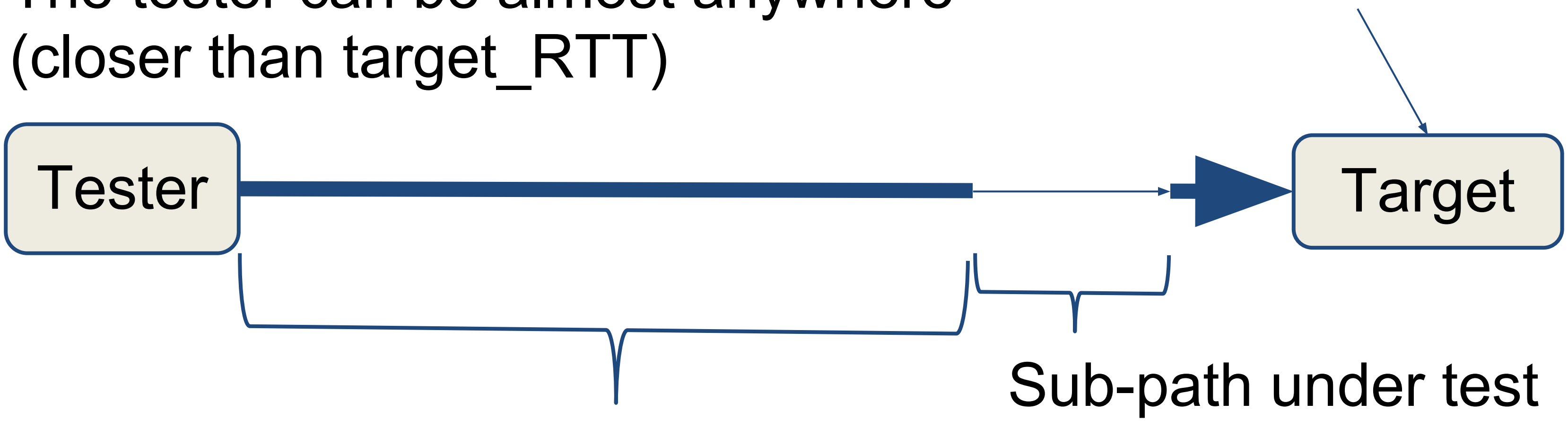
- Packet Reordering
  - But (I think) TCP should be more tolerant
  - "Equal cost multipath routing" should be ok
- Metrics to support Real Time
  - See the rmcat charter's mention of IPPM
  - Probably in a different document

Derating and Calibration

- Future draft will present multiple TCP models
    - Allow CUBIC and other TCP variants
    - Allow for (limited) multiple TCP streams
        - This is not without cost
            - Would be taken as a signal that this is ok
- Future section on calibration
    - Validate E2E performance with derated parameters
    - With a network infinitesimally failing all tests
    - (Think epsilon-delta proof, except a measurement)
- Address the failure cases
    - Paths that pass all single property tests, but fail E2E

The goal

Patterned after NPAD/Pathdiag



The tester can be almost anywhere (closer than target_RTT)

Minimal instrumented discard service

Tester

Target

Sub-path under test

The "core" portion of the path should not effect the results.

Must meet constraints determined by models based on target_rate, target_RTT and target_MTU