Model Based Metrics

IPPM Working Group
IETF 88, Vancouver BC, Nov 2012

Matt Mathis <mattmathis@google.com>
Al Morton <acmorton@att.com>

Outline (DRAFT)

- Document status update & major changes
- Quick overview (new spin)
- Some concrete examples
- Model tweaks & parameter derating
- Future work & open issues

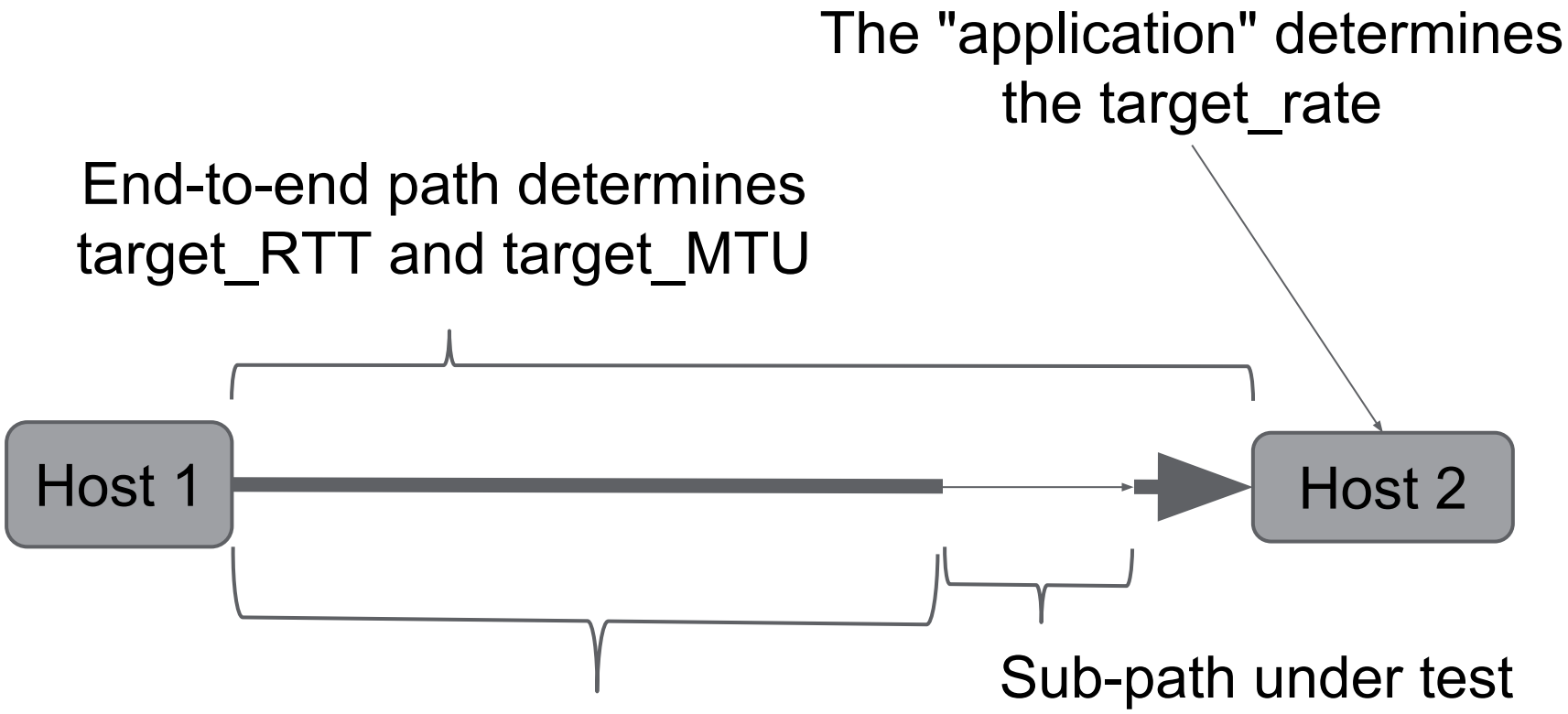
Document Status (-01)

- Completed previously missing sections
  - (Model) Validation (was Calibration)
  - Alternate model derivations
  - **The document is now structurally complete**
- Narrowed scope
  - Focuses on test traffic patterns and delivery statistics
  - All other measurement detail are (or to be) abstracted away
    - (Some obsolete text lingers)
- New key concepts
  - Targeted Diagnostic Suite
    - A set of tests with specified traffics patterns and delivery statistics
  - Fully Specified Targeted Diagnostic Suite
    - ...plus type-p and other out-of-scope measurement details

Overall Methodology

- Choose Target Parameters
  - Target_data_rate, target_RTT, and target_MTU
- Compute common model parameters
  - target_pipe_size - required average window size
  - target_run_length - required spacing between losses/ECN marks, etc
- Generate a Targeted Diagnostic Suite (TDS)
  - Pass/Fail/Inconclusive tests of all important IP properties
    - Average spacing between losses (run length)
    - Sufficient buffering at the dominant bottleneck
    - Sufficient tolerance for IF rate bursts
    - Appropriate treatment of standing queues (AQM, etc)
- FSTDS fully specifies all of the rest of the test details
  - Type-p, timing and cross traffic tolerances, etc
  - Out of scope for this document, but can reuse existing metrics

Overview


The "application" determines the target_rate
End-to-end path determines target_RTT and target_MTU
Host 1
Host 2
Sub-path under test
The rest of path is modeled as though it is effectively ideal
Each sub-path must pass all IP diagnostic tests of a Target Diagnostic Suite (TDS).

Example: HD Video at moderate range (50 mS)

- Target: 5 Mb/s (payload) rate; 50 mS RTT; 1500 Byte MTU
- Model:
  - Target_pipe_size = 22 packets
  - Target_run_lenght = 1452 packets
- Computed TDS:
  - Run length longer than 1452 packets (no more than 0.069% loss)
  - Tolerates 44 packet slowstart bursts (twice the actual bottleneck rate)
    - (Peak queue occupancy is expected to be 22 packets)
  - Tolerates 22 packet bursts at server interface rate
    - (Peak bottleneck queue also expected to be 22 packets)
  - Standing queue test:
    - First loss/ECN is more than 1452 packets after the onset of queueing
    - First loss/ECN is no later than 3*1452(?) packets after queueing onset
      - Precise success criteria still under evaluation

Tieing it all together

- Assertion:
  - If every sub-path passes all TDS tests (IP layer!)
  - Then the end-to-end path will meet the target TCP performance

- Observation:
  - The Fully Specified TDS must be concrete
  - The derivation from the targets does not have to be as concrete

An easier test procedure

- Fold most of the TDS into a single combined test
  - Downside: symptoms become ambiguous
- Send 22 packet server rate bursts every 50 mS
  - Must average <1 loss/ECN every 66 bursts (1452 packets)
  - This has the same average data rate
  - ...same stress on the primary bottleneck (although more frequent)
  - ...same or higher stress on the rest of the path
- This test may actually be too conservative
  - A path that can withstand this test is likely to meet a higher target
  - This was the motivation for "derating"

Quasi-passive

- Diagnosis as a side effect of delivering real content
  - e.g. using RFC 4898 - TCP ESTATS MIB
- Requires non-throughput maximizing traffic
  - To avoid self inflicted congestion
  - E.g. any streaming media < target_rate
- Requires serving RTT < target_RTT
- Compute test_window = target_data_rate*serving_RTT
- Clamp serving cwnd to test_window
  - Average rate over any full RTT will be smaller than target_rate
  - All bursts will be smaller than test_window (also target_pipe_size)
  - Compute run length from actual delivery statistics

The concept of parameter derating

- Original idea was to partially offset overly conservative models
  - Replace theory with **empirically derived models**
- Proven via "Validation" experiments
  - Must justify alternate models and assumptions
  - Construct a real network that infinitessimally passes the TDS
  - Demonstrate that a real application can still meet the targets
  - **The validation has to be public to the same extent as the results**
- The potential for parameter creep
  - MBM will implicitly separate network and transport responsibility
    - See ICCRG @ IETF86
  - Tweaking models may make it easier to pass dubious gear but
  - also gives transport designers permission to be more aggressive

Parameter derating in the current draft

- Greatly reduced prominence
  - Removed all test specific derating, except:
- Tests where we do not have strong models
  - Use derating to include weakly justified "rules of thumb"
    - For server rate bursts:
      - Do they have to be full window, or are partial windows sufficient?
      - Should they be permitted to have "slightly" higher losses?
    - For standing queue tests:
      - How late is still ok for the first loss (AQM test)
      - Signatures of channel arbitration problems

Next steps

- Start a separate research paper
  - Move much of the background material out of the draft
    - Also rational and "paths not chosen", etc
- Finish and evaluate prototype MBM tools
  - Plan a full measurement and validation study in the research paper
- -01bis is already open
  - Currently receiving a tight editorial pass
  - No content changes yet
    - Although significant portions are already tagged to move elsewhere
- Possible new text:
  - TDS completeness and coverage rules
  - Discussion of budgeting loss and delay (RTT) across subpaths
  - Testing in idealized environments (this is not the right language)
- Open a comment tracker(?)