Improving Speed Tests

Srikanth Sundaresan (Princeton), Amogh Dhamdhere and k claffy (CAIDA)

AIMS 2017
Speed tests have not changed in years

- They still just run TCP stream(s) between two hosts and report a number
- None of the popular tools try to do anything more
  - No attempt at any type of diagnosis
    - Where did congestion occur (if it occurred)?
    - Was it the access link or the wireless link or something else?
Very little needs to change to be able to answer (some of) these questions

• Packet captures at servers can tell us about RTT
  – Which in turn can tell us about the conditions that the flow encounters

• The TCP flow has already punched a hole in the NAT
  – Which ought to let us probe the path all the way to the end host
Very little needs to change to be able to answer (some of) these questions

• Packet captures at servers can tell us about RTT
  – Which in turn can tell us about the conditions that the flow encounters

• The TCP flow has already punched a hole in the NAT
  – Which ought to let us probe the path all the way to the end host
What *sort* of congestion did a TCP flow encounter?

- **Self-induced congestion?**
  - Clear path, the flow itself induced congestion
  - Access links with plan rates

- **Already congested path?**
  - Low available capacity
  - Congested interconnect

- **Cannot distinguish using just throughput numbers**
  - Plan rates vary widely
TCP Congestion Signatures

• Self-induced congestion fills up an empty buffer during slow start
  – This causes the RTT to increase (Max RTT – Min RTT)
  – Also increases variability (Coeff. Of Variation of RTT)

• Simple Decision Tree Model Using the RTT Parameters
Does it work?

- Extensive validation using controlled experiments testbed
  - Build model using testbed data
  - Minimize complexity
“Validation” using M-Lab data

- Time-span – Cogent interconnection issue (~Feb 2014)
  - Coarse ground truth
  - The two event periods clearly stand out
Very little needs to change to be able to answer (some of) these questions

• Packet captures at servers can tell us about RTT
  – Which in turn can tell us about the conditions that the flow encounters

• The TCP flow has already punched a hole in the NAT
  – Which ought to let us probe the path all the way to the end host
Very little needs to change to be able to answer (some of) these questions

- Packet captures at servers can tell us about RTT
  - Which in turn can tell us about the conditions that the flow encounters

- The TCP flow has already punched a hole in the NAT
  - Which ought to let us probe the path all the way to the end host
Probing the TCP Path Using BufferTrace

• The Idea: Send TTL-limited packets *within a TCP flow*
  – Observe the buildup of buffers
  – Trace the path that the flow actually takes
  – Send zero-payload TCP packets so as to not break the application layer
  – Encode hop ID in the sequence number
    • Some NATs rewrite the IPID field
Demo

https://github.com/ssundaresan/buffertrace
[Private repo, ping me for access]

Based on:
https://github.com/robertswiecki/intrace
Drawbacks

• Both techniques depend on buffering
  – How much?

• Lack of solid ground truth for congestion signatures
  – Any labeled data source for interconnect congestion?

srikanths@princeton.edu