CS551
End-to-End Internet Packet Dynamics
[Paxson99b]

Bill Cheng

http://merlot.usc.edu/cs551-f12

End-to-end Packet Dynamics

- How do you measure Internet performance?
  - Why do people want to know?
  - Are ISPs willing to tell you?
- What kinds of packet dynamics are observed in the network?
- Does there exist a *typical* Internet path?

Key Ideas

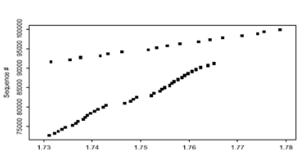
- Measure Internet traffic
  - active measurements
  - N^2 paths
  - lots of details out of TCP
- Evaluate dynamics
  - pathologies (out-of-order, duplication, corruption)
  - bandwidth
  - loss
  - delay

Methodology

- Previous studies
  - Focused on a small number of paths
  - Used unrealistic traffic (pings etc.)
- Paxson's study
  - Examined nearly 1000 paths
  - Used TCP traffic
    - routers designed to handle TCP as common case
    - congestion-adaptive (both good and bad)
  - Was extraordinarily careful
    - used statistically valid sampling to reduce bias
    - *looked at the wire* to get most confidence
    - adjusted for TCP implementation idiosyncrasies

Pathologies: Reordering

- Reordering: packets arrive at receiver in a different order than they were sent
- Evidence:
  - Significant (non-trivial) occurrence (10-30% connections)
  - Strongly-site dependent
  - Most egregious instances correlated with route flutter
    - Different packets sent along different routes
- Other curious effects
  - Router forwarding lulls (i.e., stops forwarding as if it has gone to sleep)



Impact of Reordering

- On TCP fast retransmit and recovery
  - Which assume packet loss upon receiving dup-ACKs
  - But packets may actually have been reordered
- Can we avoid this by:
  - Waiting before sending ACK
    - yes, about 20ms waits would have detected most reordering events
  - Reducing the dup-ACK threshold
    - possibly, to 2
  - But, these require server and client side change
    - bottom line: current techniques work

Other Pathologies

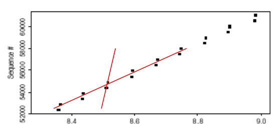
- Packet duplication
  - Link layer retransmissions
  - Happens, but very infrequently
- Packet corruption
  - About 1 in 5000 ($2x10^{-4}$)
  - Is TCP 16-bit checksum enough to protect against this?
    - maybe not
- Found one out of 300K ACKs corrupted, so maybe not

Bottleneck Bandwidth Estimation

- How do you compute the bottleneck path bandwidth?
  - Bottleneck BW: max possible rate
  - Available bandwidth: reasonable share
- Packet pair
  - Send two packets, each size S, closely spaced
  - At bottleneck, the packets are separated by a time T
  - Bottleneck bandwidth Qb = S/T
- Where to measure? Sender (RTT) or receiver (OTT)?
  - If inference done at sender, can be error-prone because of
    - ACK compression
    - bandwidth asymmetry, which causes noise in reverse path

Packet Pair Problems and Fixes

- Clock granularity (fix: measure multiple packets)
- Route changes (fix: measure several, take mode)
- Out of order delivery (fix: filter out)
- Multi-channel links, route spraying (fix: measure for multiple packets)



Fix? Packet-bunch Modes

- Compute estimates from *bunches* of packets each sent closely spaced to the next (also known as ***packet trains***)
- Get ***modes*** from the distribution of estimates
  - If two modes widely separated in trace-> route change
  - If two modes for different bunch sizes-> multi-channel links
  - Bunches also eliminate clock granularity problems

Packet Loss

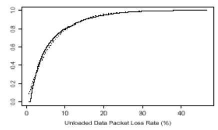
- Fairly high rates (3% or 5%)
  - much higher on some links, ex. US to Europe
- But many connections are loss-free (30- 66%)

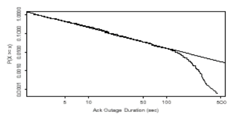
Is Loss Predictive?

- short-time-scale: packet a to b (stream)
  - define ***queued*** and ***unqueued*** pkts
    - queued := packet i queued behind i-1 at bottleneck link
    - else unqueued (sufficent spacing that no self-queueing)
  - queued packets have much higher loss rates
- long-time scale: hours or days
  - zero/non-zero is predictive (data not in paper)
  - actual loss ***rate*** is not predictive
  - allows ***traffic engineering***

Loss Patterns

- Data vs ACK loss
  - Data loss across connections well-modeled by exponential
  - Not so for ACKs
- Bursts
  - Loss are *not* independent
  - Burst sizes are heavy-tailed



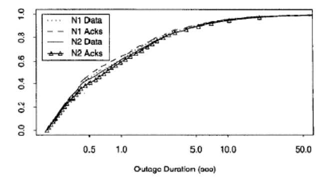


Burst Loss

- Conditional loss definition
  - P[pkt i lost | pkt i-1 was lost]
  - conditinal loss rates are much higher
- Why
  - drop-tail routers
- Implications
  - losses are *not* i.i.d

TABLE II
UNCONDITIONAL AND CONDITIONAL LOSS RATES

Type of loss	P_l^u $\mathcal{N}_1$	P_l^u $\mathcal{N}_2$	P_l^c $\mathcal{N}_1$	P_l^c $\mathcal{N}_2$
Queued data pkt	2.8%	4.5%	49%	50%
Unqueued data pkt	3.3%	5.3%	20%	25%
Ack	3.2%	4.3%	25%	31%



Overall Loss Characteristics

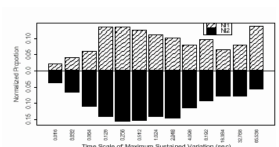
- ACK loss is the correct determinant of network conditions
  - In measuring, must be careful to account for tcpdump losses
- Doubling of average loss in one year
- Loss rates don't have predictive power
  - But whether a connection suffers loss or not can be used for prediction
- Existence of
  - Dual network states (*quiescent* vs. *busy*)
  - Diurnal variations
  - Geographical diversity in loss patterns
  - No *typical* loss rate
- Avoiding unnecessary retransmissions
  - Correct RTO implementation
  - SACK

Delays

- ACK and data timing compression should *not* happen
- ACK compression
  - A flight of ACKs queued behind cross traffic
  - Happens quite infrequently
    - although most connections experienced one
    - durations are small and number of such events is small
  - Packet pair techniques can account for this by rejecting outliers
- Data timing compression
  - Much more infrequent than ACK compression
  - Possibly due to specific routers

Delays

- Queueing time scales
  - Measured by variations in one-way transit times
  - Show wide variability, so we cannot design for a particular regime
- Available bandwidth
  - Approximated by variations in delay experienced due to own loading
  - Again, shows wide variability
  - Most between 0.1 - 1 sec



Questions?

- Do you think this study is valid today?
- What has happened since 1995?
- Dialup->broadband
- Better connectivity
- Higher backbone speeds