The End-to-End Effects of Internet Path Selection

Stefan Savage
Andy Collins, Eric Hoffman, John Snell
Tom Anderson

Department of Computer Science and Engineering
University of Washington

Motivation

- Routing is a black box
  - Packets follow some path chosen for you
  - That path has a delay and a loss rate
  - Maybe that was the best path… probably not
- Our goal:
  - Quantify and understand the impact of *path selection* on end-to-end performance

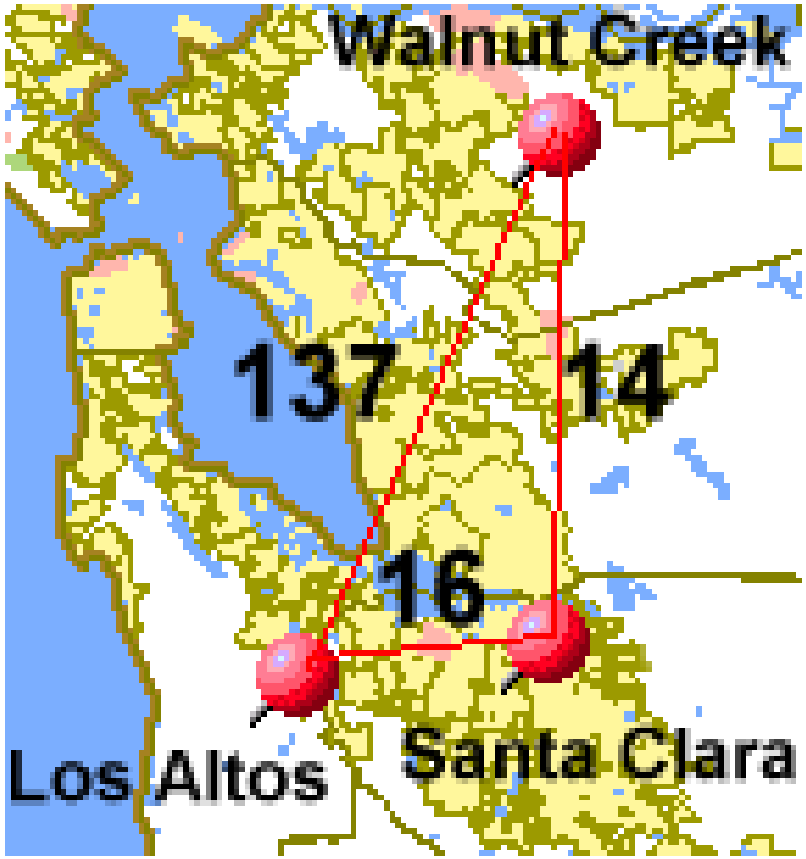
Anecdotal evidence

Does path selection impact performance?

YES

(sometimes a lot)

How often, how much, and why?



Measured round-trip times (in ms)

Quantifying the impact of path selection

- Basic metric:

  Let X = performance of default path

  Let Y = performance of best path

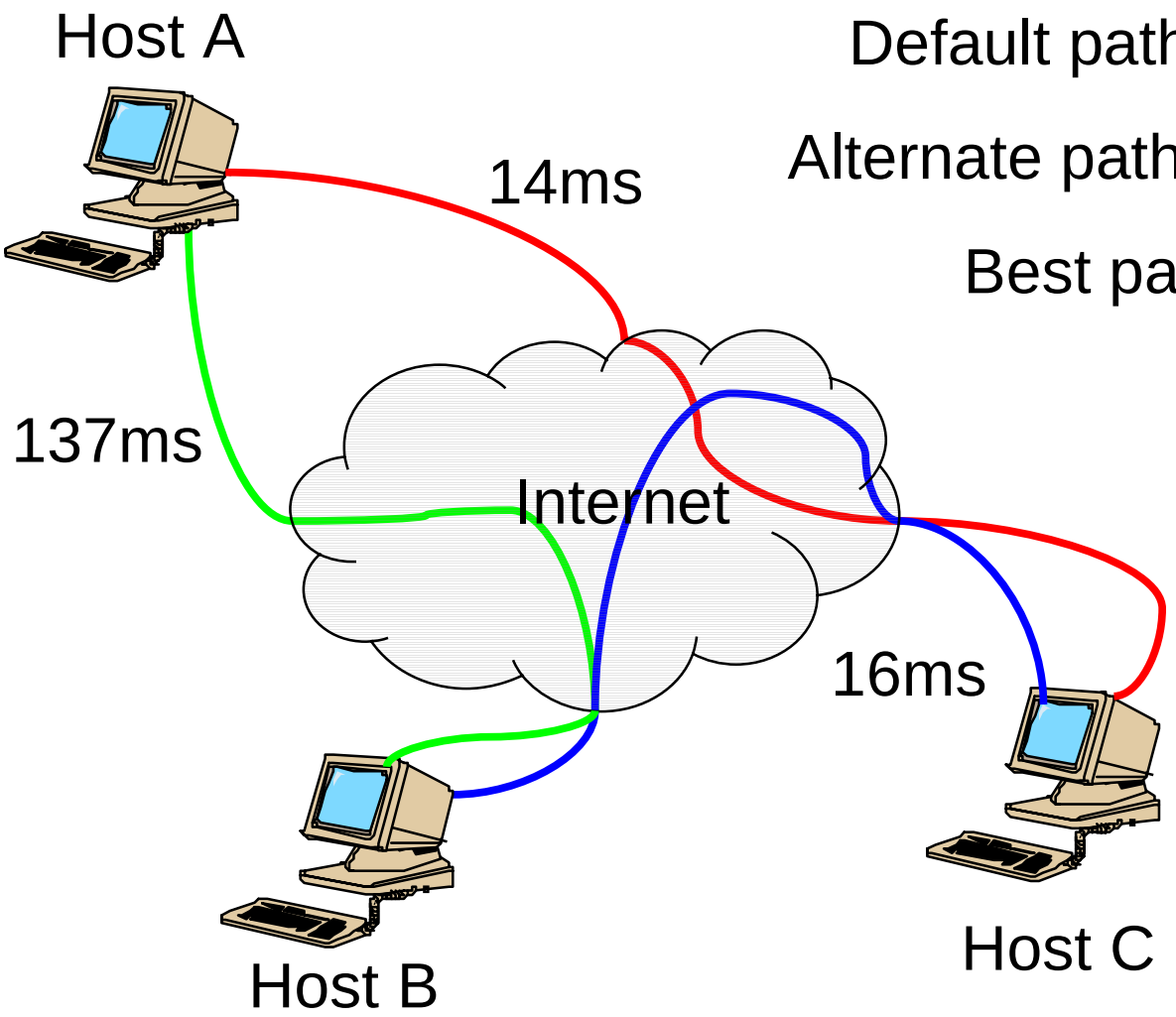
  Y-X = cost of using default path

- Technical problems
  - How to find the best path?
  - How to measure the best path?

Approximating the best path

- Key idea
  - Use end-to-end measurements to extrapolate potential alternate paths
- Rough algorithm
  - Measure paths between pairs of hosts
  - Generate *synthetic* topology – full N^2 mesh
  - Find best alternate path through this graph
- Conservative approximation of *best* path

Example:
Lowest latency path from A to B



Default path AB = 137ms

Alternate path AC-CB = 30ms

Best path is 30ms

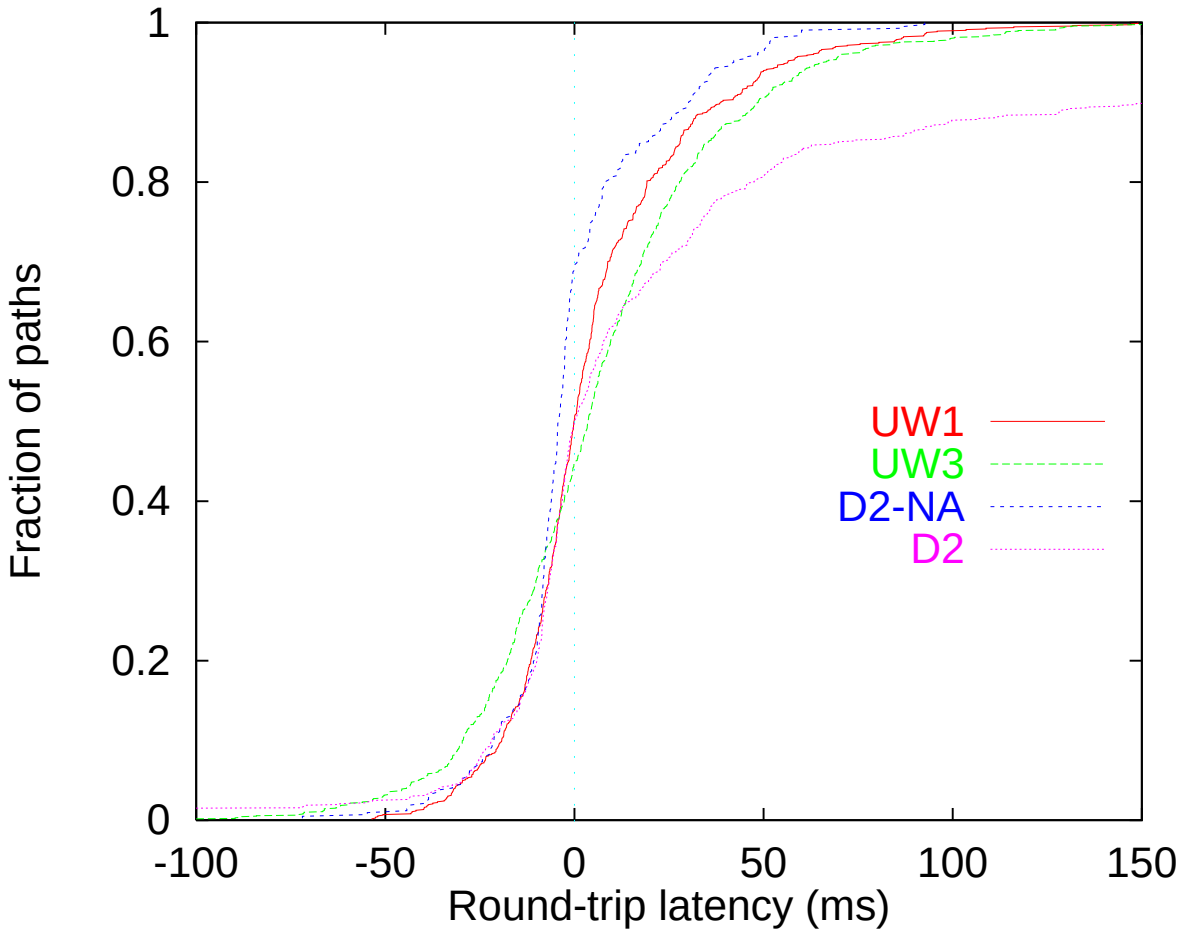
Visualizing “routing efficiency”

- For each pair of hosts, calculate
  - Average round-trip time
  - Average loss rate
  - Average bandwidth
- Generate synthetic alternate paths based on long-term averages
- For each pair of hosts, graph difference between default path and best alternate

Calculating synthetic path metrics

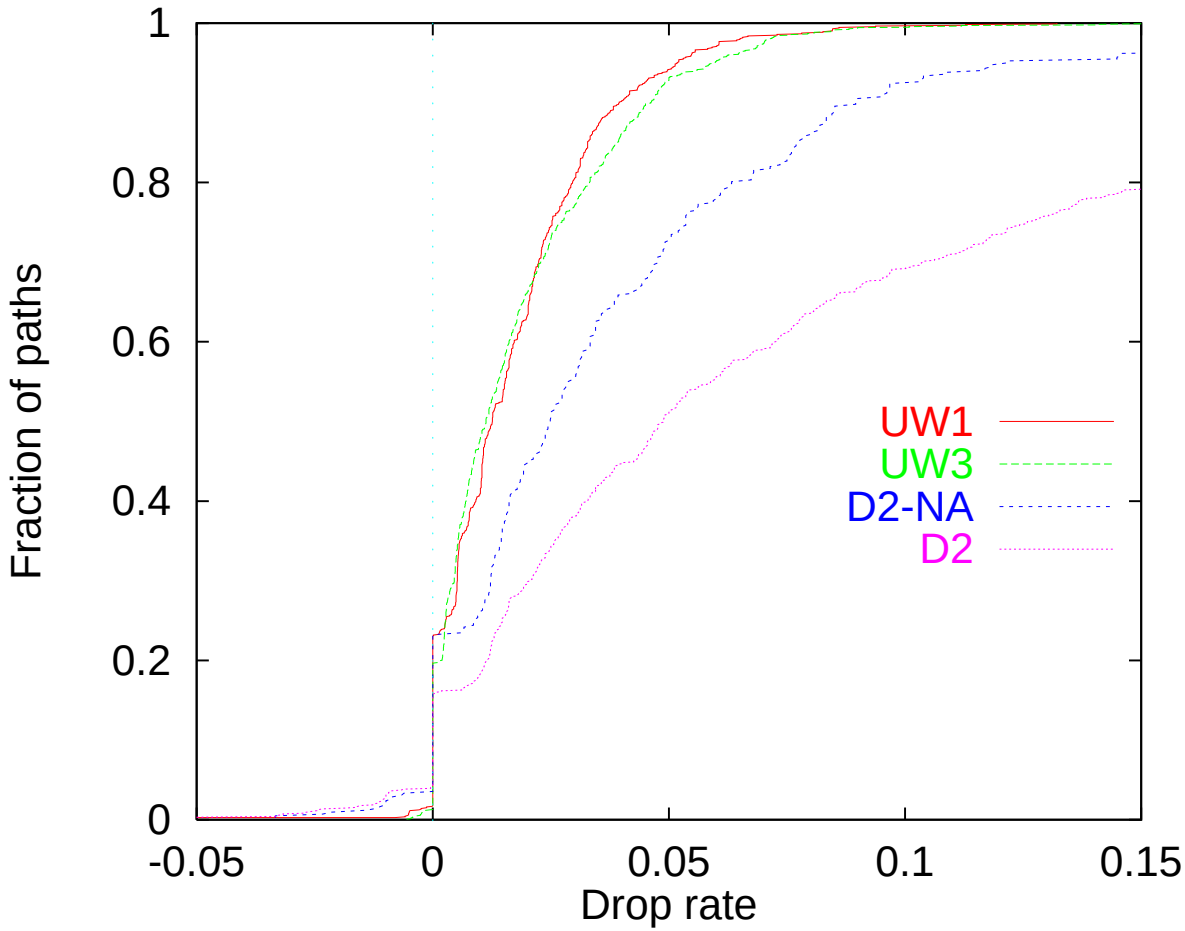
- Round-trip time: AB + BC = ABC
- Loss rate: 1-((1-AB)*(1-BC)) = ABC
- Bandwidth
  - Pessimistic: same as above
  - Optimistic: MAX(AB,BC) = ABC
  - Solve using [Mathis97] approximation for TCP bandwidth
- Use distribution convolution for medians

Round-trip time



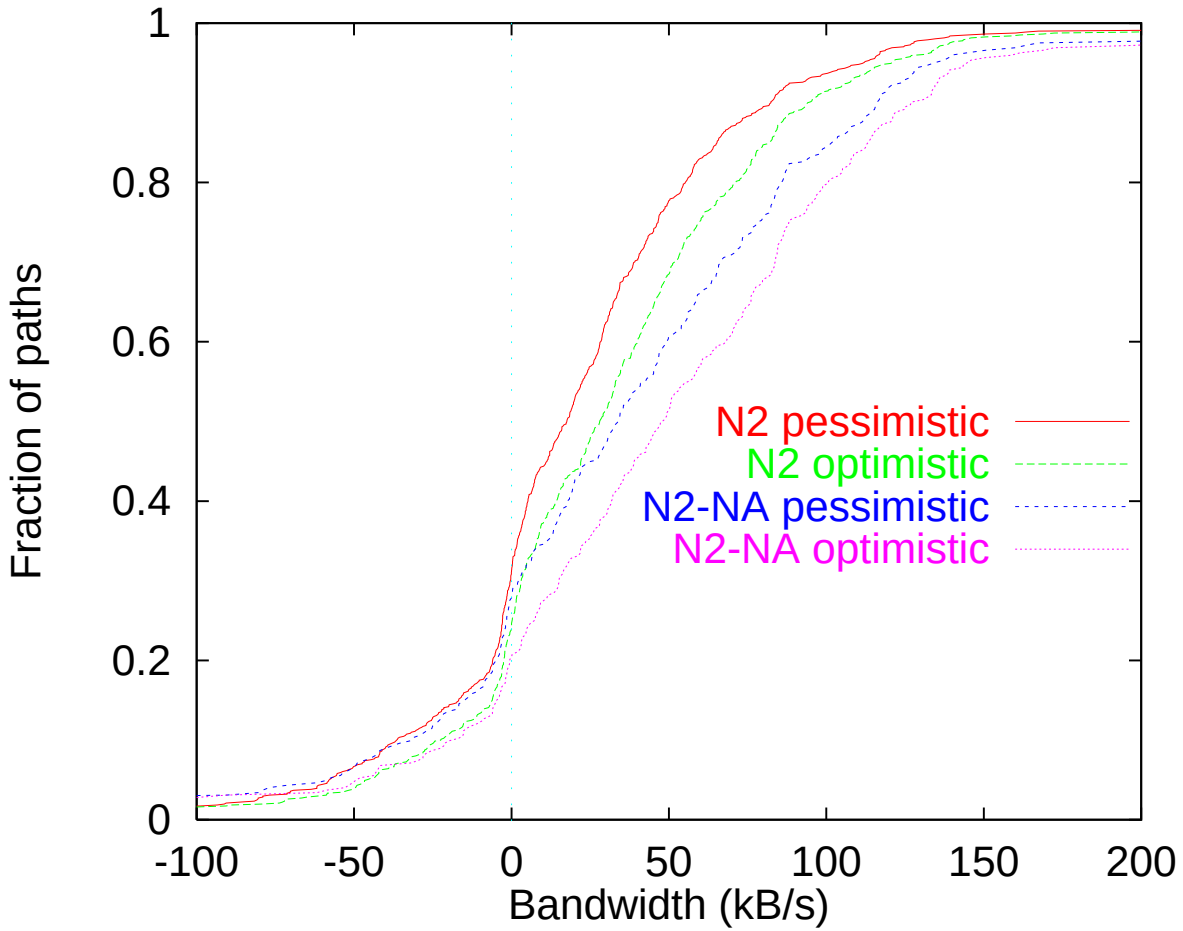
30%-55% of default paths have longer round-trip times

Loss rate



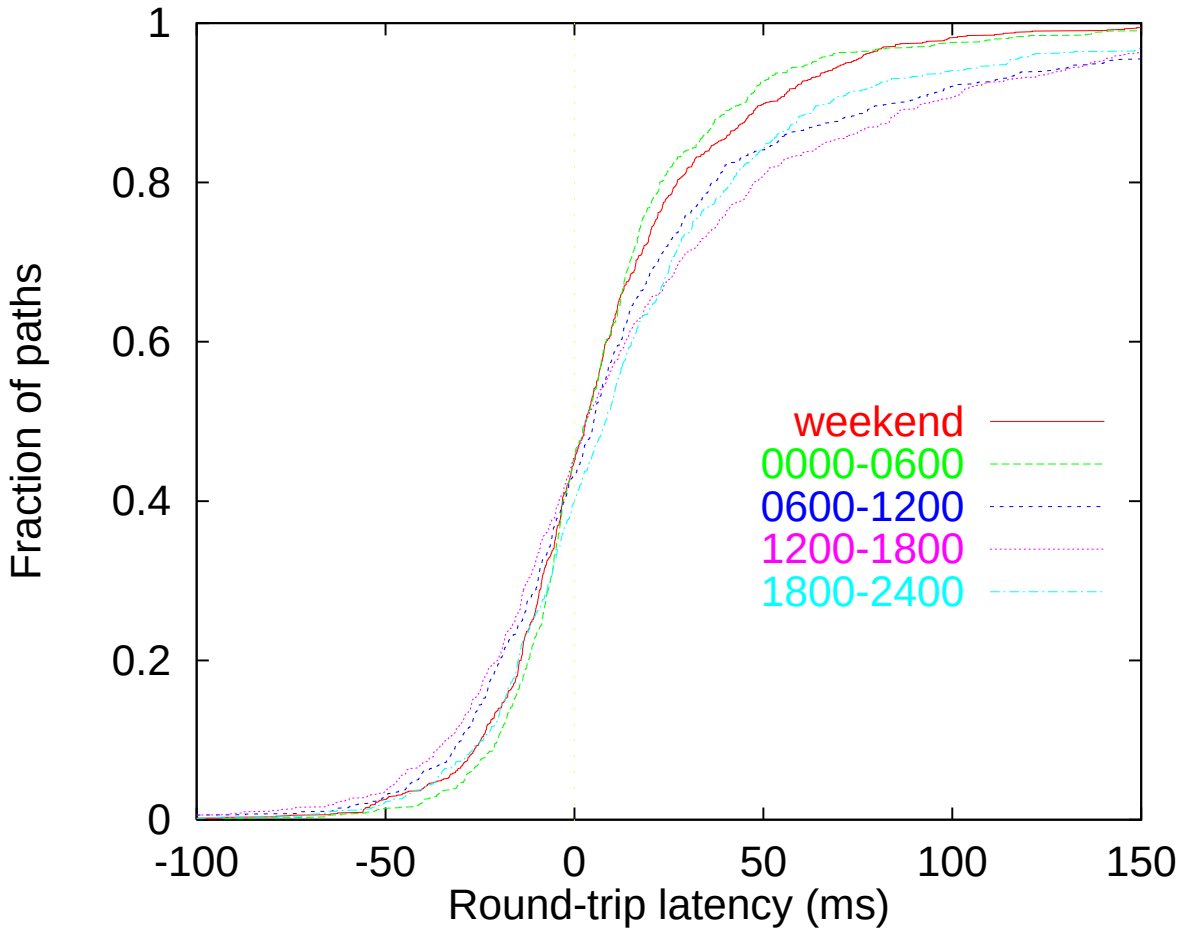
75%-85% of default paths have higher loss rates

Bandwidth



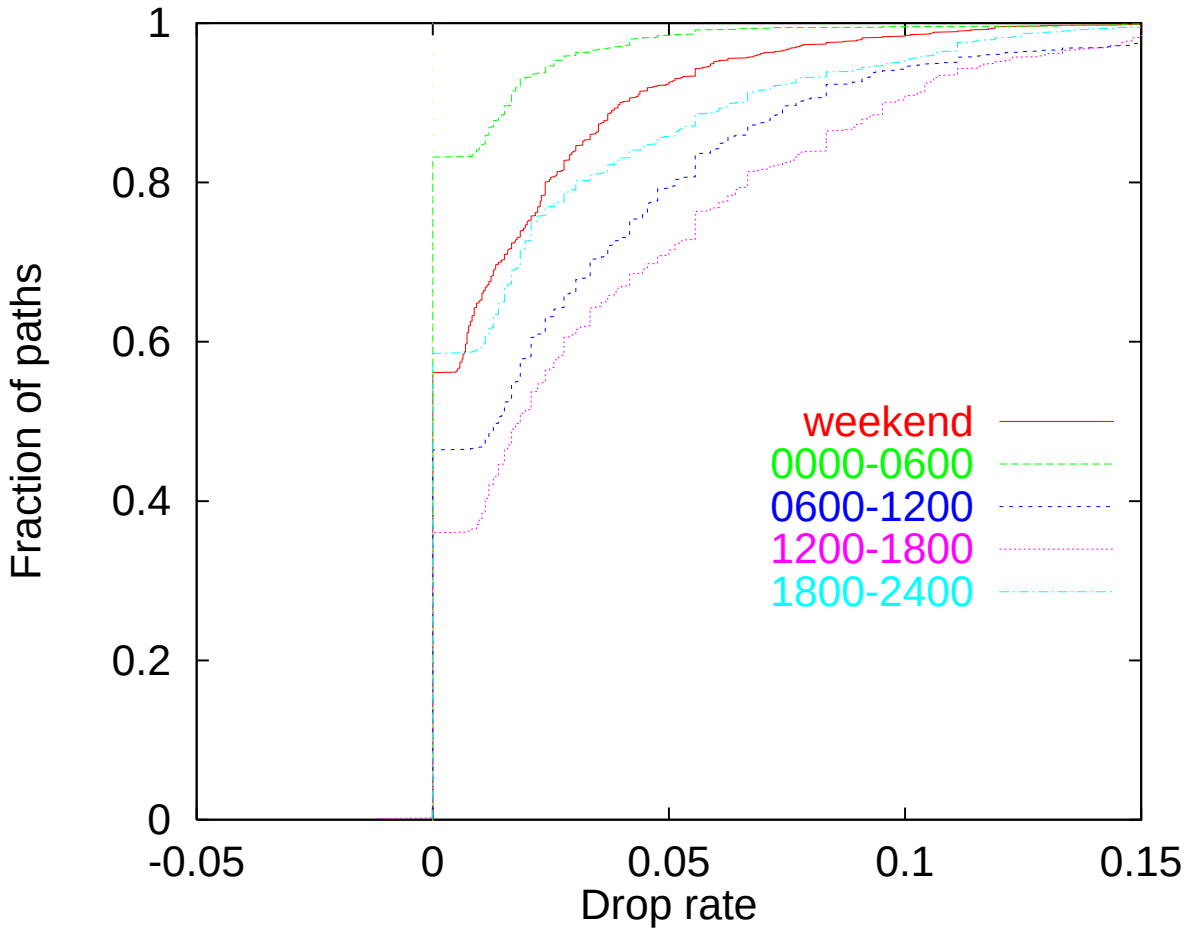
70%-80% of default paths have lower bandwidth

Time-of-day variation (latency)



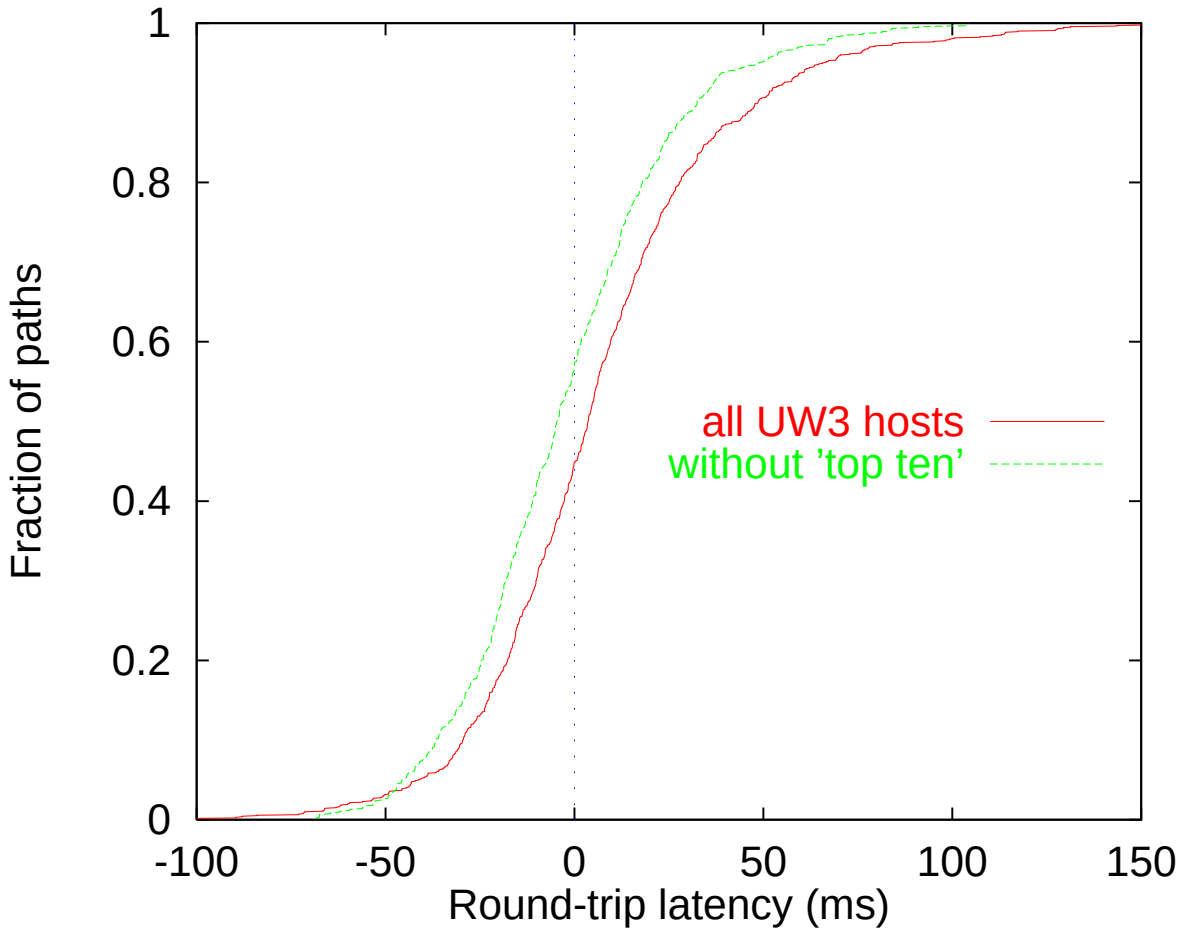
Effect stronger during “peak” hours

Time-of-day variation (loss)



Even stronger peak effect for loss

Are there a few bad or good hosts?



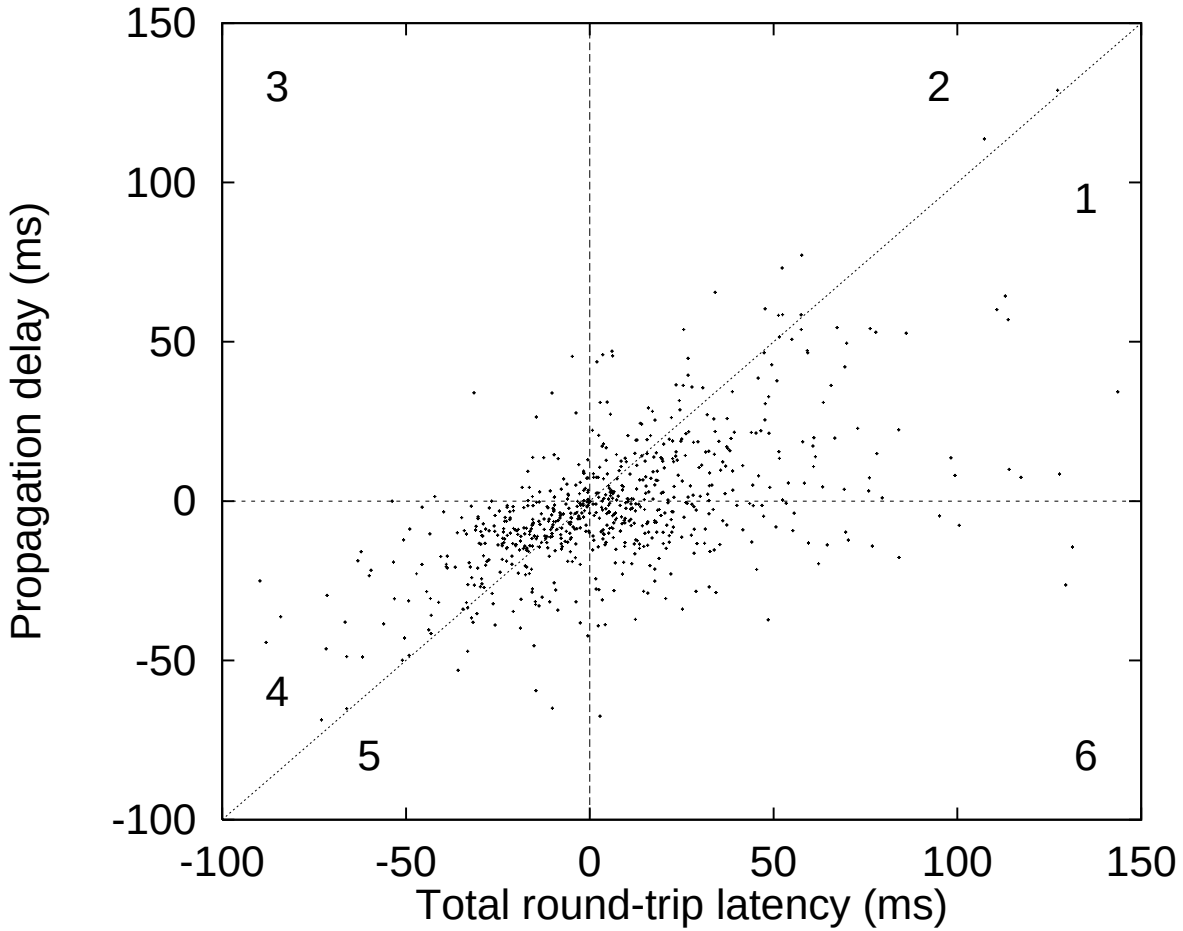
Quick summary of results

- The default path is usually not the best
  - True for latency, loss rate, and bandwidth
  - **In spite** of synthetic end-host transiting
- Many alternate paths are much better
- Effect stronger during peak hours
- Better paths can be shorter, less congested, or both

What makes a better path better?

- Possibilities
  - Avoids congested queues
  - Shorter propagation delay
- Answer seems to be: **both**
- Visualizing propagation and congestion
  - Estimate propagation delay (10^{th} percentile)
  - Queuing delay = RTT – propagation delay
  - Graph improvement in propagation delay vs improvement in RTT

Propagation delay vs congestion



Why path selection isn't "perfect"

- Technical reasons
  - Single-path routing
  - Non-topological route aggregation
  - Coarse routing metrics (AS_PATH)
  - Local policy decisions
- Economic reasons
  - Disincentive to offer transit
  - Minimal incentive to optimize transit traffic

Conclusions

- We can roughly quantify the impact of path selection on performance
- Routing is clearly a significant part of the end-to-end performance equation

Datasets characteristics

Name	Year	Duration (days)	Hosts	Number of measurements
UW1	1998	34	36	54034
UW3	1998	7	39	94420
UW4-A	1999	14	15	216928
UW4-B	1999	14	15	9169
D2 (NA)	1995	48	33 (22)	35109 (14896)
N2 (NA)	1995	44	31(20)	18274 (7582)

Traceroute server placement (UW1)

DNS Performance and the Effectiveness of Caching

Jaeyeon Jung, Emil Sit, Hari Balakrishnan, Robert Morris

MIT Laboratory for Computer Science

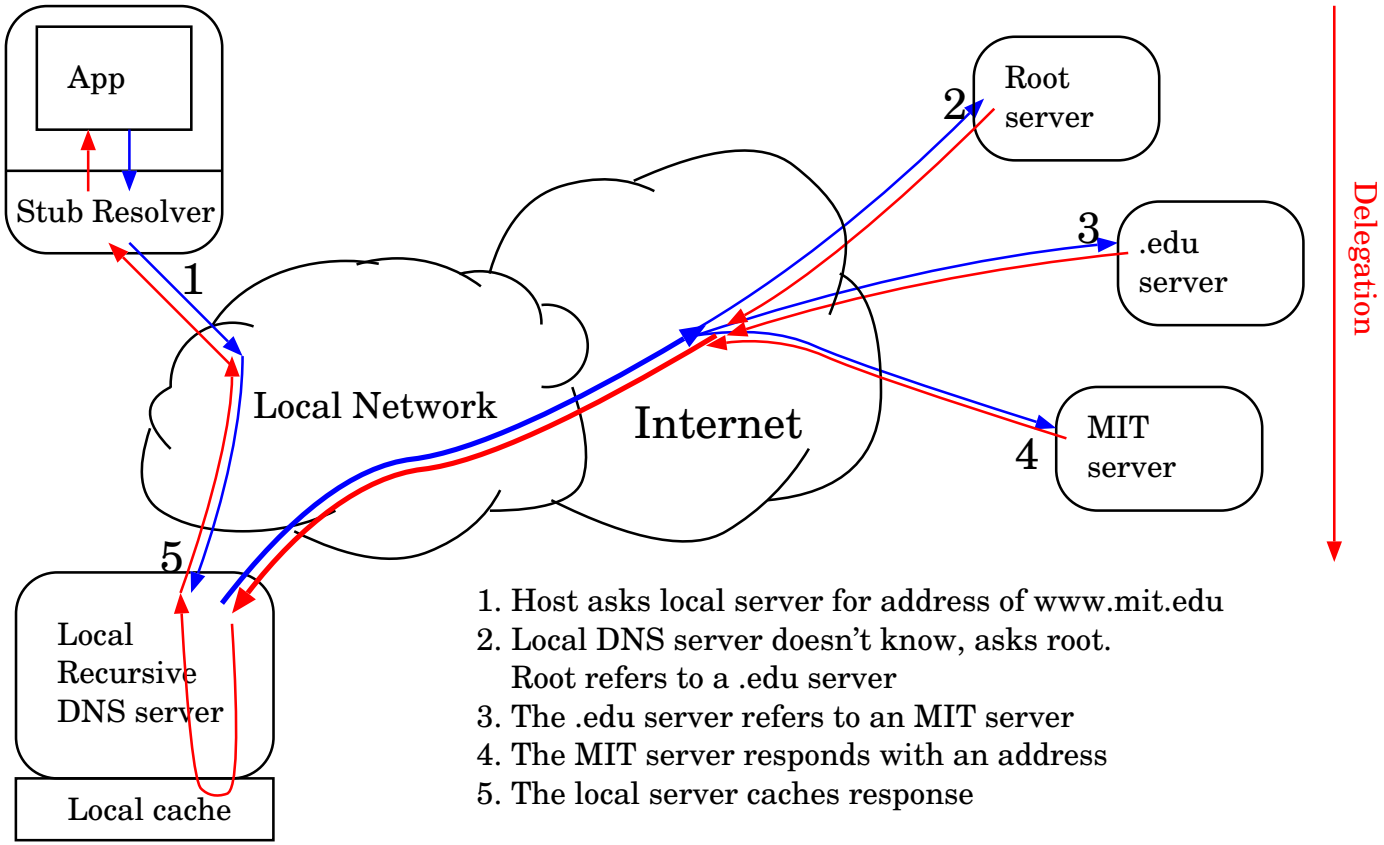
November 2001

`http://nms.lcs.mit.edu/dns/`

Motivation

- ✔ Identify the factors that affect client-perceived DNS performance
  - Response latency
  - Errors and failure modes of DNS
- ✔ Evaluate the effectiveness of DNS caching
  - How important is it for scalability?
  - Unanticipated uses of DNS (e.g. Web server selection)
  - 18% of flows were due to DNS in MCI wide-area backbone traces [Thompson97]

DNS Lookup Sequence



1. Host asks local server for address of www.mit.edu
2. Local DNS server doesn't know, asks root.
   Root refers to a .edu server
3. The .edu server refers to an MIT server
4. The MIT server responds with an address
5. The local server caches response

Terminology

- ✔ Mapping in the DNS name space
  - `A` record : name's IP address
  - `NS` record : name of DNS server
- ✔ Caching in DNS
  - *Time To Live* : expiration time set by the originator of a name
  - Negative caching

Questions

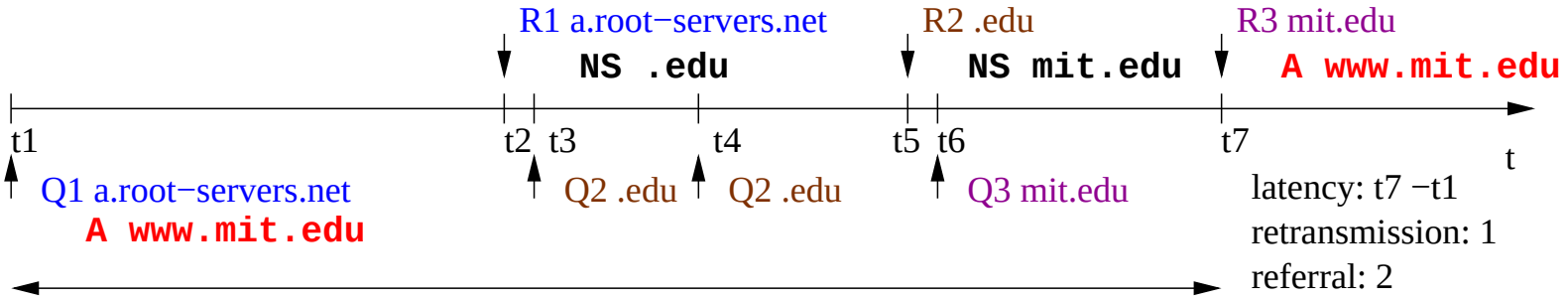
- ✔ What is the ratio of TCP connections to DNS A record lookups?
- ✔ What is the number of DNS queries per lookup?
- ✔ DNS errors
  - What percentage of lookups do never get an answer?
  - Performance of retransmission protocol
- ✔ DNS failures
- ✔ What is the effect of varying TTLs and degrees of caching sharing on cache hit rate?

Key Findings

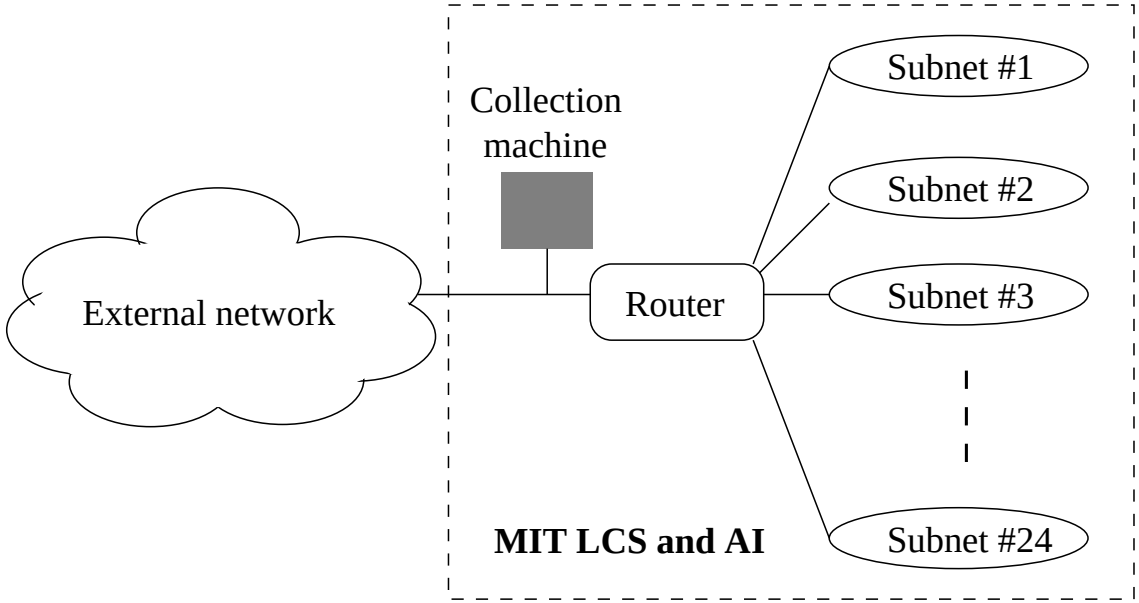
- ✔ TCP / DNS lookup ratio suggests that the hit rate of DNS caches inside MIT is between 70% and 80%
- ✔ 23% of all client lookups in the most recent MIT trace fail to elicit any answer
- ✔ 13% of lookups result in an answer that indicates a failure. Most of these failures indicate `NXDOMAIN`
- ✔ % of TCP connections made to names with low TTL values increased from 12% to 25% in 2000
- ✔ Setting all A-record TTL's to a value as small as 10 minutes is not likely to degrade the scalability of DNS

The Data

- ✔ Collection Methodology
  - Wide-area DNS query/response
  - Outgoing TCP connections: `SYN/FIN/RST`
  - Anonymized internal addresses
- ✔ Analysis Methodology
  - Sliding window of 60 seconds



Traced Network Topology



✔ MIT LCS/AI

- 24 internal subnetworks sharing the border router
- Data collected in January and December 2000

Basic Statistics

	mit-jan00	mit-dec00
Date	00/01/03-10	00/12/04-11
Total lookups	2,530,430	4,160,954
Unanswered	23.5%	22.7%
Answered with success	64.3%	63.6%
Answered with failure	11.1%	13.1%
Zero answer	1.0%	0.5%
Total query packets	6,039,582	10,617,796
TCP connections	4,521,348	5,347,003
#TCP : #valid `A` answers	4.62	3.53

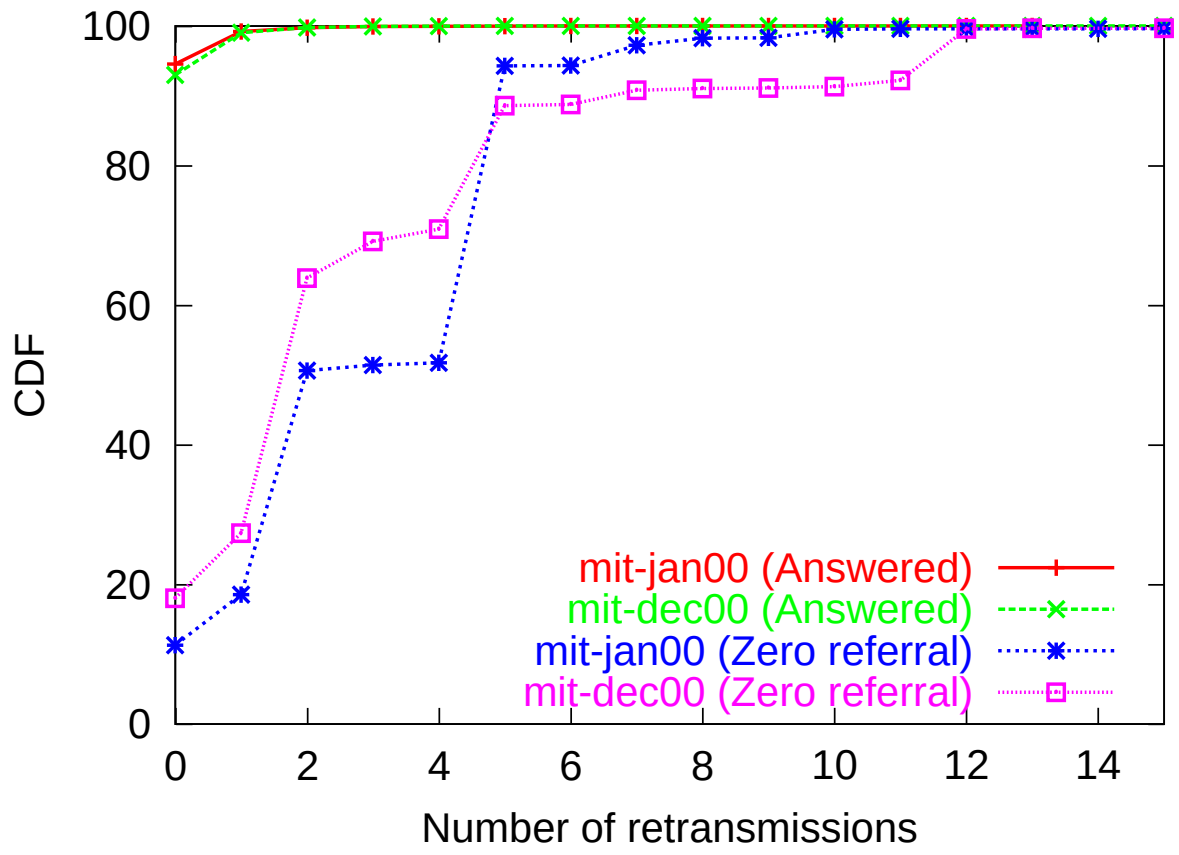
Unanswered Lookups

- Significant fraction of all DNS packets seen in the wide-area Internet:
  - 59% (mit-jan00), 63% (mit-dec00) of total query packets

	mit-jan00	mit-dec00
Zero referrals	5.5%	9.3%
Non-zero referrals	13.1%	10.3%
Loops	4.9%	3.1%

- Unanswered lookups classified by type -

Retransmissions



✔ 99.9% of answered lookups have $\leq$ 2 retransmissions

Suboptimal Retransmission Strategy

- ✔ Overly persistent retransmissions
  - Average # of retransmissions:

    3.5 (mit-jan00), 5 (mit-dec00)
  - # of retransmissions of worst 5%:

    6 (mit-jan00), 12 (mit-dec00)
- ✔ Inappropriate setting for the number of retries or excessive timeout value
  - No retransmissions within 60 seconds:

    12% (mit-jan00), 19% (mit-dec00)

Failures

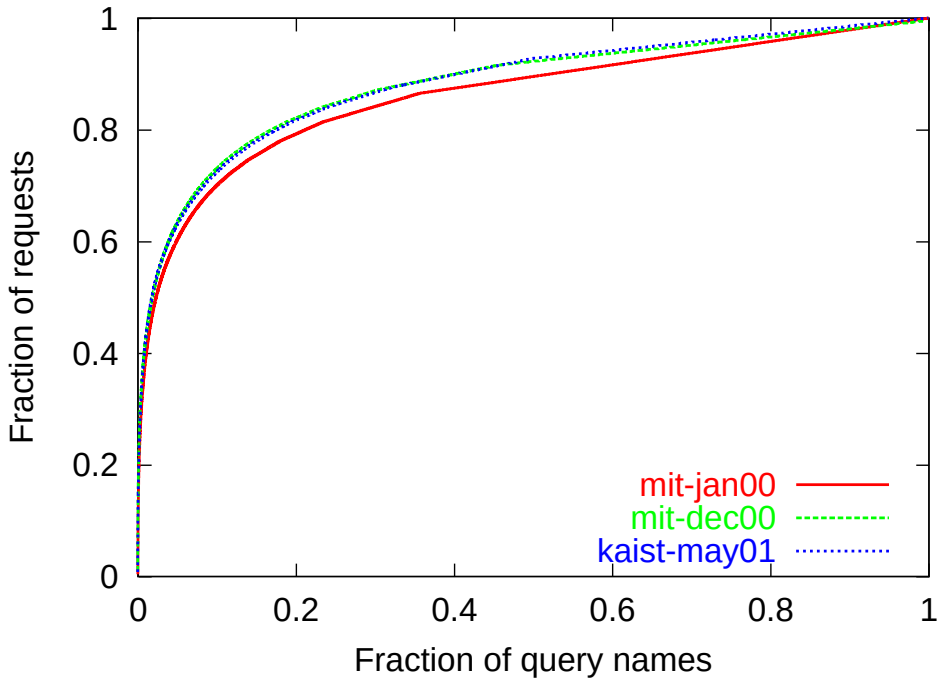
	mit-jan00	mit-dec00
NXDOMAIN	7.68%	11.14%
SERVFAIL	3.24%	1.96%

- ✔ Inverse lookups for IP addresses with no inverse mapping: `213.228.150.38.in-addr.arpa`
- ✔ Invalid queries : `ld;`
- ✔ Non-existent top-level domains: `loopback`, `index.htm`
- ➔ Large number of distinct names makes negative caching ineffective

DNS Caching

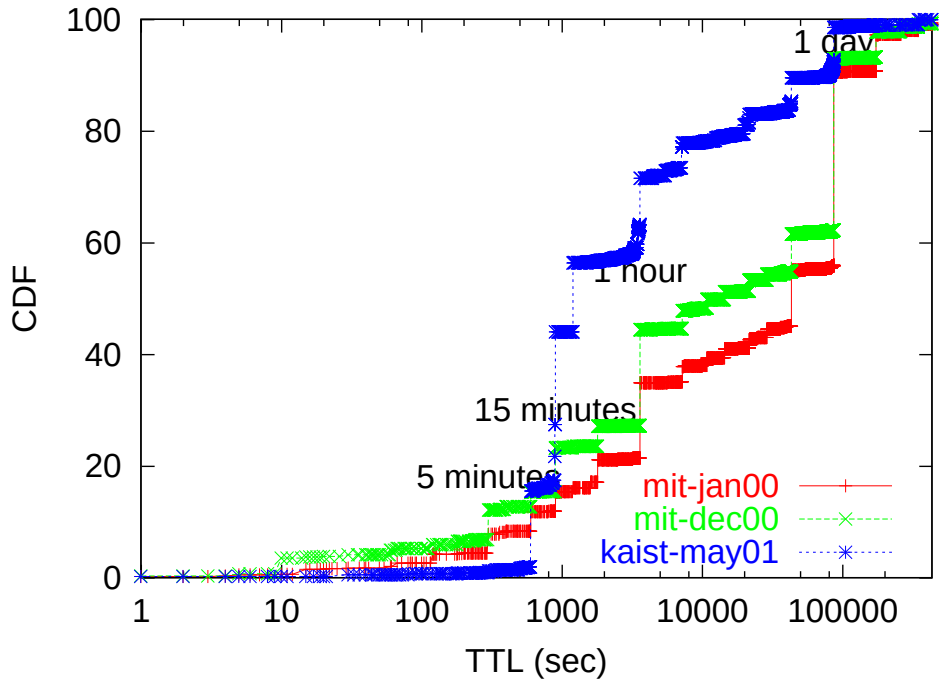
- ✔ How useful is it to share DNS caches among many client machines?
  - ➔ Locality of references among clients
- ✔ What is the likely impact of choice of TTL on caching effectiveness?
  - ➔ Locality of references in time

Name Popularity



- The top 10% account for more than 68% of total lookups
- A long tail : 9.0% are unique names

TTL Distribution

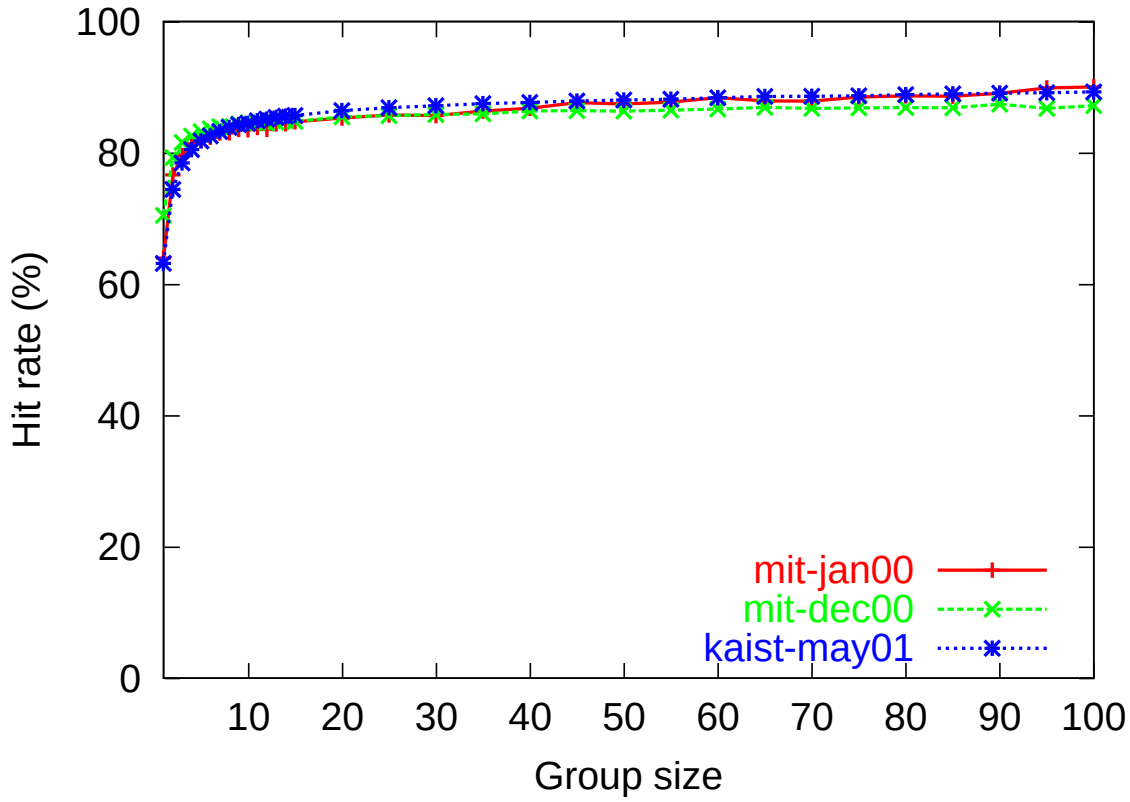


- ✔ The fraction of accesses to short TTLs has doubled
  - ➔ Increased deployment of DNS-based server selection

Trace-driven Simulation

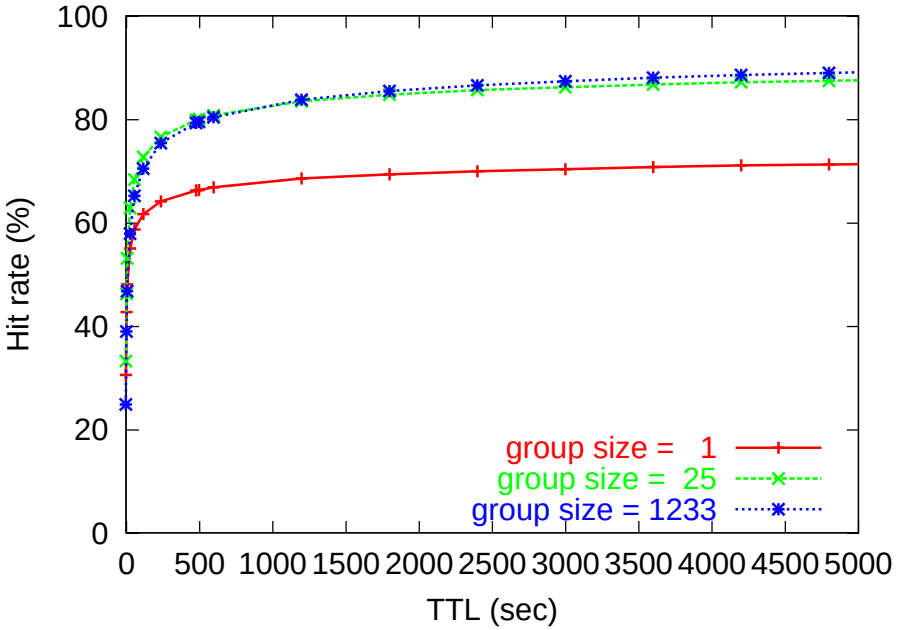
- ✔ TCP connection workload
- ✔ Database containing name/IP/TTL bindings
- ✔ Algorithm
  - Randomly divide TCP clients into groups of size *s*
  - For each new TCP connection, determine the group *g* and look for a name *n* in the cache of group *g*
  - If *n* exists and the cached TTL has not expired, record a *hit*. Otherwise record a *miss*

Effect of Sharing on Hit rates



- ✔ Most of the benefit of sharing is obtained with as few as 10 or 20 clients per cache

Impact of TTL on Hit rates



- Most of the benefit of caching is achieved with TTLs less than about 1000 seconds.
- 5-min TTLs would increase DNS traffic by factor of 1.5
- `NS` record caching is critical

Conclusions

- ✔ About a quarter of all DNS lookups never get an answer, which corresponds over 50% of DNS packets in the wide-area Internet
- ✔ The DNS retransmission protocol appears to be overly persistent, but in 10%-12% of cases, no retransmissions occur.
- ✔ Setting all A-record TTL's to a value as small as 10 minutes is not likely to degrade the scalability of DNS in any noticeable way.
- ✔ The cacheability of `NS` records enhances scalability by reducing load on the root and top-level name servers