Turbo King: Framework for Large-Scale Internet Delay Measurements

Derek Leonard

Joint work with Dmitri Loguinov

Internet Research Lab
Department of Computer Science
Texas A&M University, College Station, TX 77843

April 15, 2008

Agenda

- Introduction
  - Related Work
- Understanding King
- Turbo King
  - Measurement Algorithm
  - Discovering Nameservers
- Evaluation
- Conclusion

Introduction I

- Distance estimation in the Internet has recently evolved into a large field
  - The goal is to estimate or measure latency (delay) between hosts
- Can be used to provide better service to end-users and construct more efficient networks
  - Increasing responsiveness for online games
  - Quickly locating the closest server in a CDN
  - Creating topologically-aware P2P networks

Introduction II

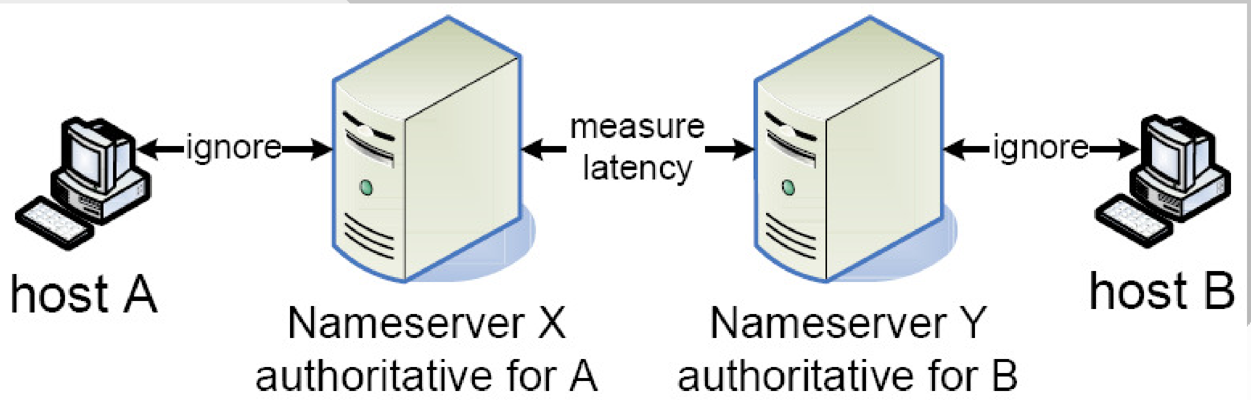
- Focus of our work
  - Create a method for distance estimation that requires no infrastructure to be deployed throughout the Internet
  - Allow for the generation of a much larger Internet latency matrix than previous work
- Requirements
  - Produce accurate latency estimates
  - Minimize the impact of our measurements on the network

Agenda

- Introduction
  - Related Work
- Understanding King
- Turbo King
  - Measurement Algorithm
  - Discovering Nameservers
- Evaluation
- Conclusion

King I

- King uses existing DNS infrastructure to estimate the latency between two arbitrary hosts in the Internet
  - Assumes end-hosts are within close proximity to the authoritative nameserver responsible for their IP addresses



King II

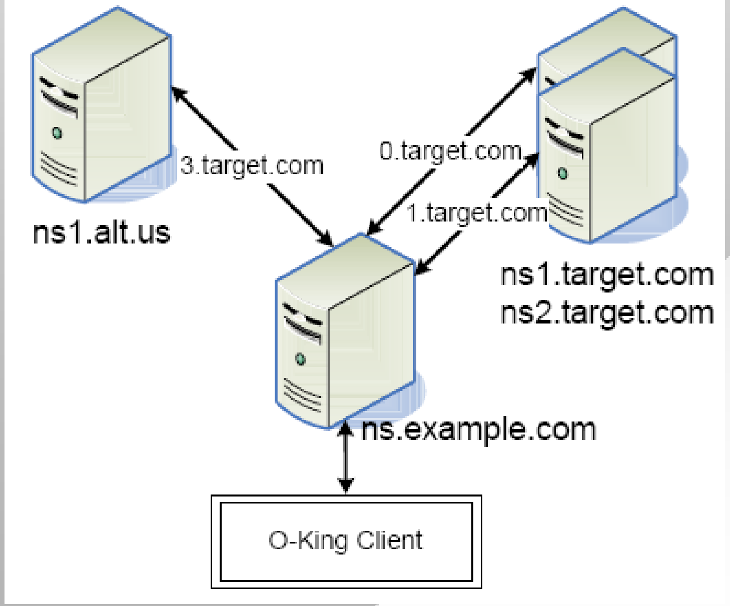
- Original King (O-King)
  - Main algorithm proposed by Gummadi *et al.*
  - Uses queries for authoritative zone data to measure the latency between two remote DNS nameservers
- Direct King (D-King)
  - Also proposed by Gummadi *et al.*, but not fully implemented or evaluated in the literature
  - Alternative to O-King that forces a nameserver to query an arbitrary target server

Agenda

- Introduction
  - Related Work
- Understanding King
- Turbo King
  - Measurement Algorithm
  - Discovering Nameservers
- Evaluation
- Conclusion

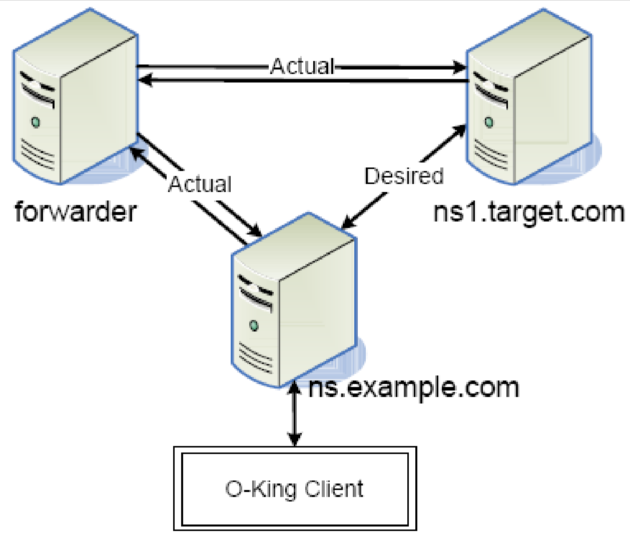
Understanding O-King I

- While O-King’s simplicity is attractive, it has certain drawbacks
- Zones with multiple authoritative nameservers
  - Recommended by DNS RFC
- Suggested to be placed in different networks
- 33% of zones have at least one server in a different network



Understanding O-King II

- DNS Forwarders
  - Server that aggregates DNS queries initiated from within a network targeting external destinations
  - Recursive nameservers receiving requests for zones not under their control often use forwarders to reduce their own load
- End-user not notified
- Unnoticed by O-King
- Potentially introduces non-trivial extra latency



Understanding O-King III

- Cache pollution: insertion of DNS zone data that has not been requested by a local user into the cache of a nameserver
  - The purpose of a DNS cache is to reduce latency mainly for local users
  - Local users are those that rely principally on the nameserver to resolve queries
- O-King seed queries pollute the cache with two records for each target nameserver
- At large scale records inserted by O-King would dominate local caches
  - Would likely be viewed as intrusive by admins

Understanding D-King I

- Additional Complexity
  - A domain name and extra infrastructure (DNS server with dedicated IP) are required
  - O-King is so simple, is D-King worth it?
- Forwarders
  - D-King does not detect or avoid forwarders
- Cache Pollution
  - D-King only requires caching records for one server
  - However, the cached records are completely useless to others

Agenda

- Introduction
  - Related Work
- Understanding King
- Turbo King
  - Measurement Algorithm
  - Discovering Nameservers
- Evaluation
- Conclusion

Turbo King I

- Turbo King (T-King) basics
  - Stand-alone service
  - Accepts as arguments the IP addresses of two end-hosts A and B
  - Returns estimated latency from host A to B
- Two modes of operation
  - Passive: waits for requests before generating latency estimates
  - Active: preemptively make latency estimates to eventually obtain an entire $N \times N$ matrix

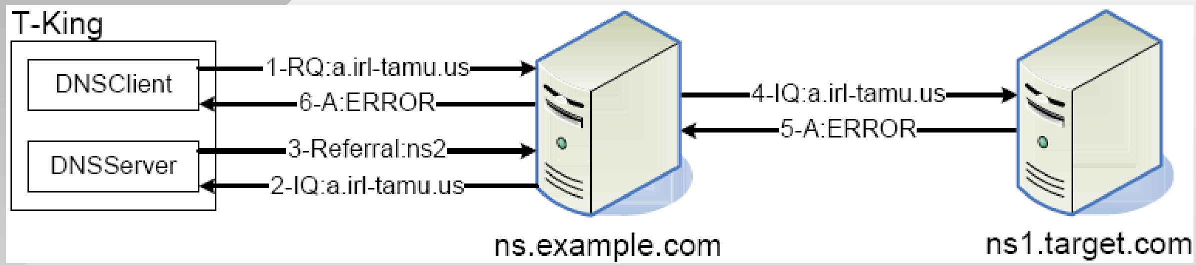
Turbo King II

- Server selection is a major difference between King and Turbo King
- Turbo King maintains a large list S of N nameservers
  - Currently both recursive nameservers and other authoritative nameservers
- Use BGP data to match servers from S to both A and B
  - Negates the assumption that the authoritative nameservers for end-host IP addresses are closest

Agenda

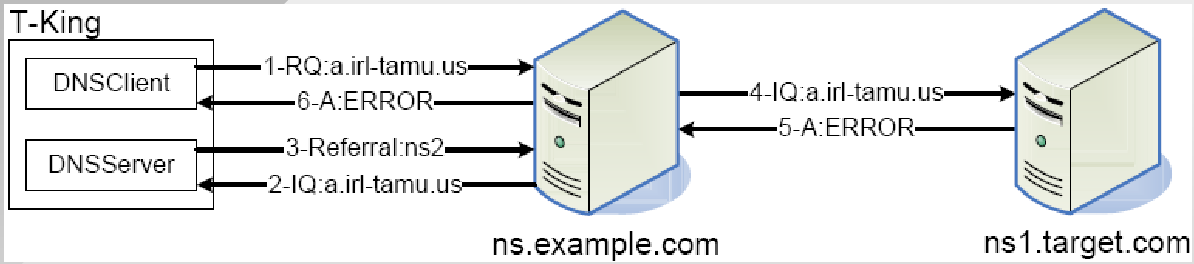
- Introduction
  - Related Work
- Understanding King
- Turbo King
  - Measurement Algorithm
  - Discovering Nameservers
- Evaluation
- Conclusion

Measurement Algorithm I



- Measurement algorithm
  - Multi-threaded application with client and server operations communicating together
  - Timestamps taken for every packet sent/received by T-King
- All answers returned by DNSServer have zero TTL
- Let d_{ij} be the delay between steps i and j

Measurement Algorithm II



- Local latency sample $L_i = d_{12}$
- Remote latency sample $R_i = d_{36}$
- Latency estimate: $min\{R_i\}$ - $min\{L_i\}$
- Detection and avoidance of forwarders
  - Compares IP addresses used to contact T-King
  - If different, exclude the original query IP and add the newly discovered one to S for later use

Agenda

- Introduction
  - Related Work
- Understanding King
- Turbo King
  - Measurement Algorithm
  - Discovering Nameservers
- Evaluation
- Conclusion

Discovering Nameservers I

- Discovering Nameservers
  - T-King is most effective when its list S of nameservers is large
  - Ideally S would contain a nameserver for each /24 network in the Internet
- The current version builds S by exhaustively crawling the reverse DNS (IP to domain name) tree
  - Only accepts authoritative responses
  - Maximizes depth of the crawl and subsequently the number of discovered nameservers

Discovering Nameservers II

- Results from the reverse DNS crawl

	T-King	ISC [18]
Month run	Nov. 2006	Jul. 2006
Duration (hours)	33.8	240
Queries/Sec	5, 300 (2.3 mb/s)	751 (0.3 mb/s)
Queries Completed	649, 270, 000	N/A
IPs Discovered	439, 431, 355	439, 286, 364
Nameservers	216, 843	89, 592
Recursive Nameservers	117, 817	N/A

- Our crawler is approximately seven times faster than a previous effort and discovered 2.4 times more nameservers
- We found that 32% use a forwarder for queries

Discovering Nameservers III

- Coverage of Internet by discovered servers

	All	Recursive	Total
Countries	190	174	232 [17]
AS	13, 017	10, 895	23, 773 [16]
BGP Prefixes	48, 196	31, 059	219, 110 [38]
IPs covered	1, 031, 736, 562	828, 675, 500	1, 642, 441, 178
Web servers	3, 192, 918	2, 659, 379	3, 638, 433
Gnutella peers	1, 734, 483	1, 338, 217	3, 534, 300

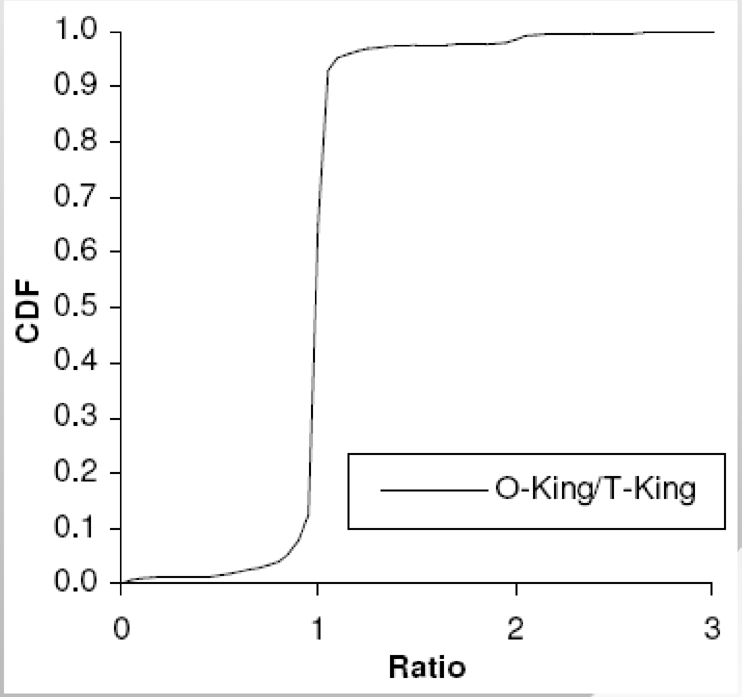
- We found that 49% of Gnutella peers and 88% of web servers are in a BGP prefix that contains at least one nameserver in T-King
- For recursive servers, 37% of Gnutella and 73% of web servers are covered

Agenda

- Introduction
  - Related Work
- Understanding King
- Turbo King
  - Measurement Algorithm
  - Discovering Nameservers
- Evaluation
- Conclusion

Evaluation I

- We took 2,450 measurements for 50 nameservers using O-King and T-King
- Ratio of O-King to T-King used to highlight the differences
- Four samples per estimate
- Conclusion: 15% of O-King estimates are more than 10% different from T-King and 8% of O-King estimates are more than 20% different

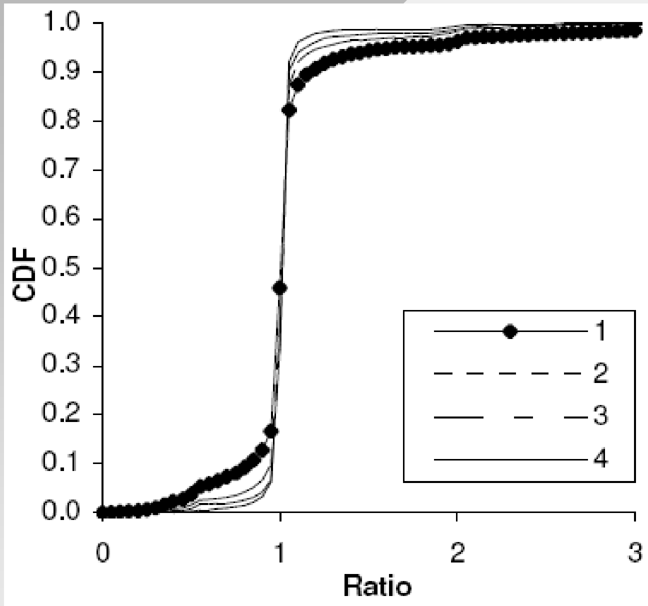


Computer Science, Texas A&M University

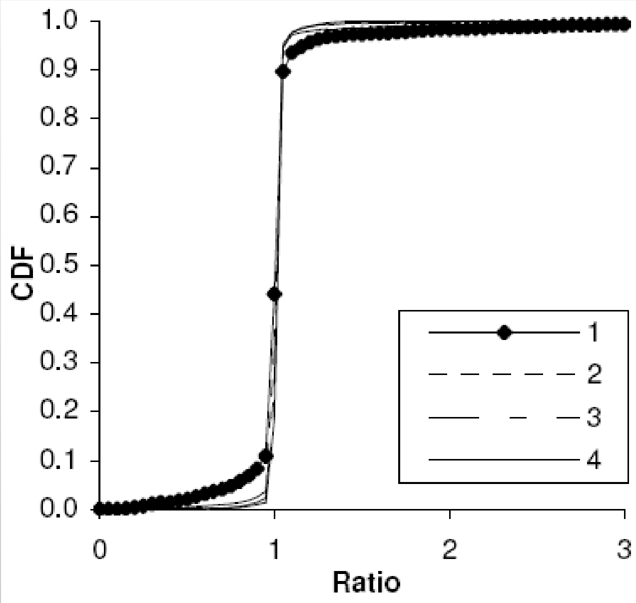
Evaluation II

- We next compare the convergence properties of T-King vs. O-King
  - Consistent samples are the goal so that an accurate estimate requires fewer samples
- To do this, we repeated the previous 2,450 latency estimates using sample sizes ranging from one to four
  - We then calculated the ratio of O-King to O-King and T-King to T-King and plotted the CDF
  - More consistent samples are centered at one on the x-axis

Evaluation III



(a) O-King/O-King by sample size



(b) T-King/T-King by sample size

- Conclusion: The original suggestion that O-King use 4 samples is sound
- Conclusion: T-King converges to a consistent estimate in only 2 samples

Evaluation IV

- The overhead associated with each method is critical when considering a large-scale measurement
  - Consider the generation of a hypothetical latency matrix of 100,000 $\times$ 100,000 hosts
  - Requires 10 billion estimates to complete
- Consider 4 samples per estimate for O-King, 2 samples for both D-King and T-King
- Examine both total number of queries sent and total cache pollution entries

Evaluation V

- Network overhead to complete 10 billion estimates
  - Turbo King: 70 billion queries
  - D-King: 100 billion queries (1.43 times T-King)
  - O-King: 150 billion queries (2.14 times T-King)
- Conclusion: The increased accuracy and lack of seeding required by T-King results in a significant reduction in bandwidth usage

Evaluation VI

- Cache pollution created for 10 billion estimates
  - Two entries for every nameserver entered into cache
- Total polluted entries in DNS caches
  - O-King: 48 billion entries (2.4 nameservers per zone measured from our reverse crawl)
  - D-King: 20 billion entries
  - T-King: 200,000 entries (0.0004% of O-King and 0.0001% of D-King)
- Conclusion: T-King is much more suitable for large-scale measurements in this regard

Agenda

- Introduction
  - Related Work
- Understanding King
- Turbo King
  - Measurement Algorithm
  - Discovering Nameservers
- Evaluation
- Conclusion

Conclusion

- We proposed Turbo King as a framework to perform large-scale Internet latency measurements
  - More accurate than both O-King and D-King
  - Requires fewer samples than O-King
  - Much more scalable in terms of both bandwidth usage and cache pollution
- Our next step is to generate an $N \times N$ latency matrix using T-King
- Please see the paper for more details