Path Stitching: Internet-Wide Path and Delay Estimation from Existing Measurements

DK Lee, *Keon Jang*, Changhyun Lee, Sue Moon, Gianluca Iannaccone*

CAIDA-WIDE-CASFI Workshop @ LA
Aug 15, 2008

Division of Computer Science, KAIST
Intel Research, Berkeley*

Motivation behind Path Stitching

- Distributed applications are popular in today's Internet
  - P2P file sharing, content distribution networks, multi-player online games
- These applications benefit from information about the Internet path between their nodes
  - Nearest neighbor discovery, leader node selection, distribution tree construction
- Our goal is a DNS-like system that provides network information

Key idea behind Path Stitching

- Internet separates inter-domain and intra-domain routing
  - Path stitching splits paths into path segments , and stitches path segments together using BGP routing information to predict a new path

- Many measurement data are available already, and we use them and do no additional measurement

Talk outline

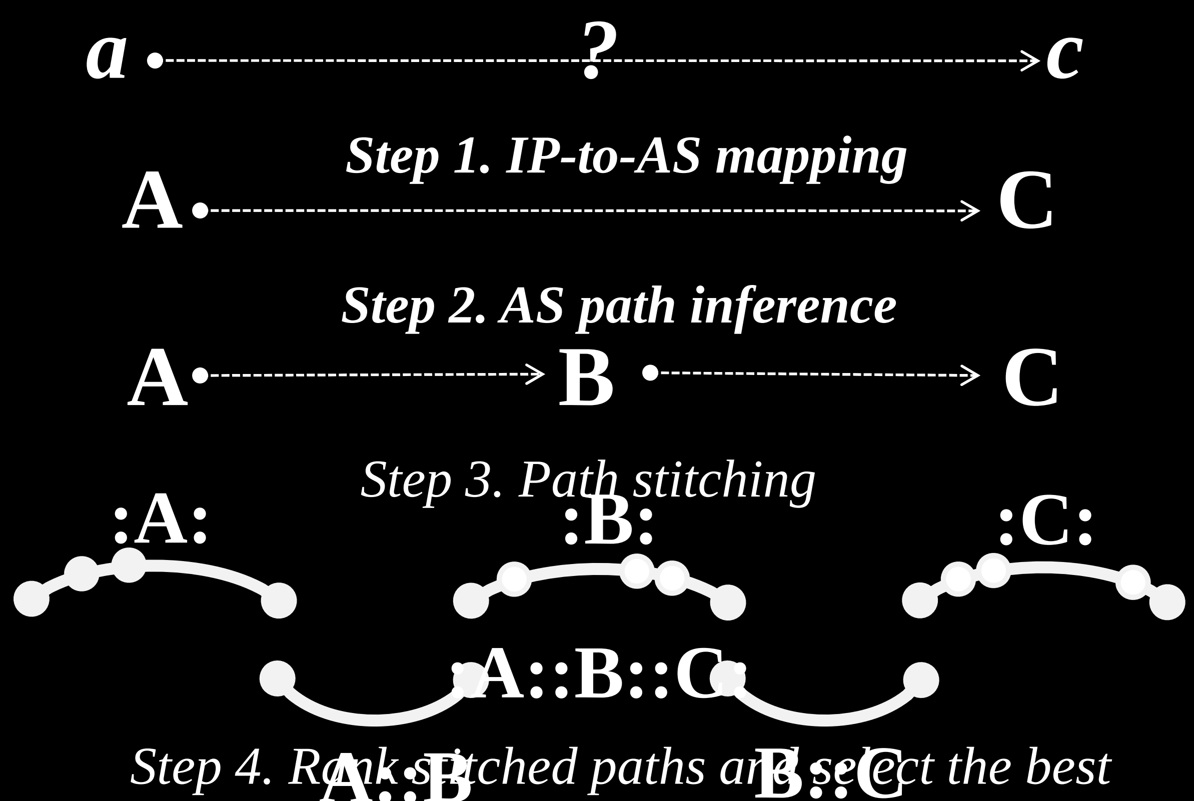
- Path Stitching algorithm
- When Path Stitching produces no stitched path
  - Approximation heuristics
- When Path Stitching produces multiple paths
  - Preference rules
- Evaluation
- Conclusion and Future Work

Data set

- *CAIDA Ark's traceroutes*
  - One round of *traceroute* outputs from 18 sources to every /24 prefix
  - 14 millions of *traceroute* outputs
- BGP routing tables
  - University of Oregon, *RouteViews*' BGP listener
  - *RIPE RIS*' 14 monitoring points (rrc00 ~ rrc07, rrc10 ~ rrc15)
- Notations
  - **:X:** Intra-domain paths of AS **X**
  - **X::Y** Inter-domain edges between AS **X** and **Y**
  - **:X: + X::Y + :Y: = :X::Y:**
    - » Internet forwarding paths from AS **X** to **Y**

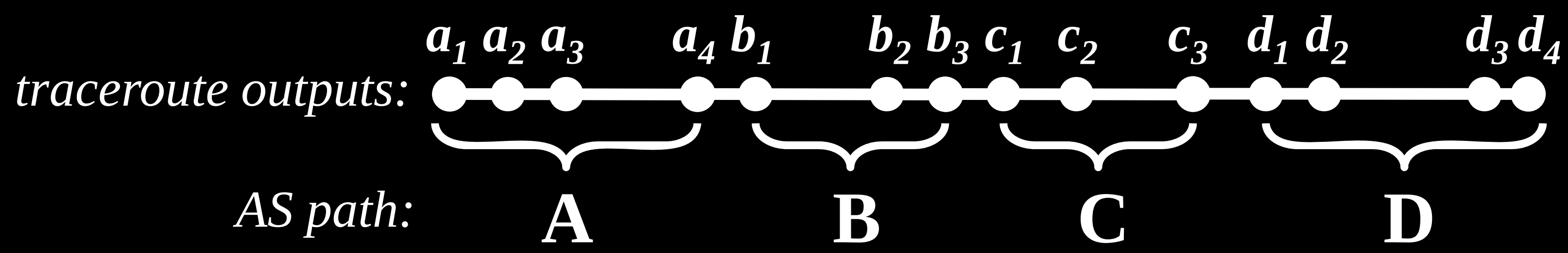
Overview of Path Stitching

- What are Internet forwarding paths and end-to-end delay between two arbitrary Internet host ***a*** and ***c*****?**



Index building

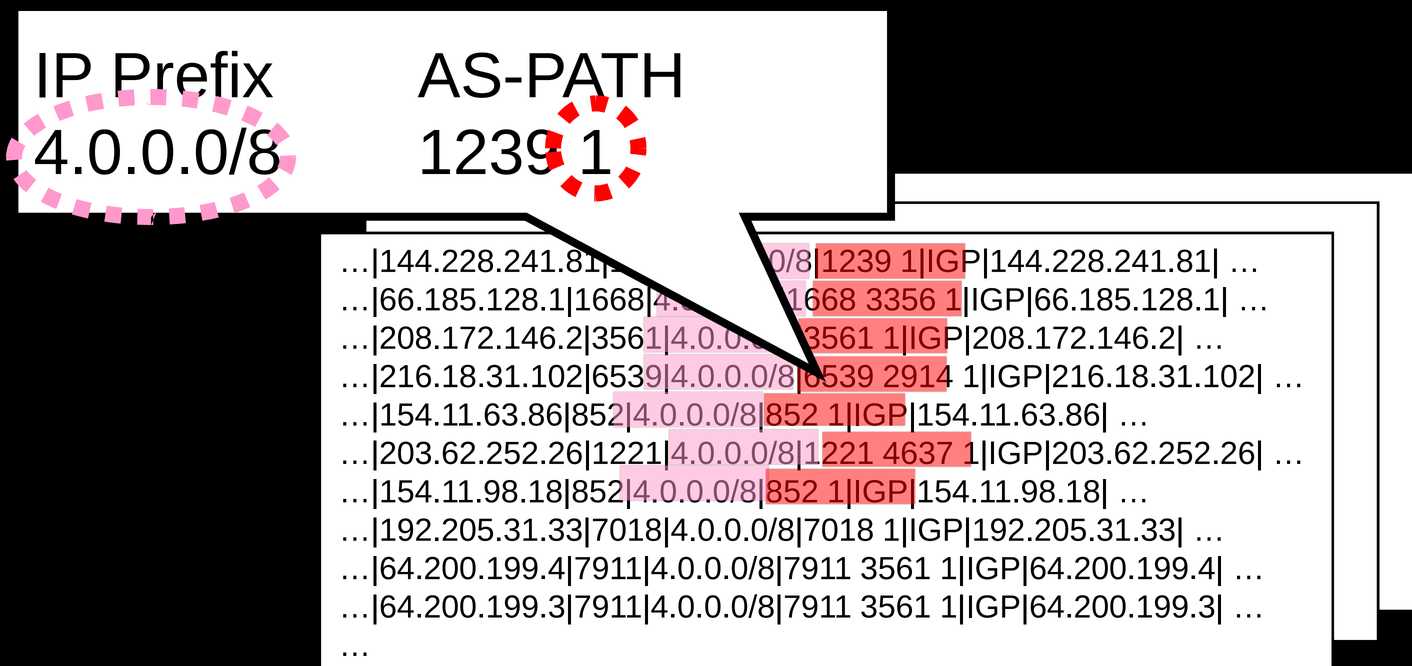
- In order to make a huge number of *traceroute* measurements *searchable*,



- Choices
  - Build indices for all possible partial paths
    - **ABCD, ABC, BCD, AB, BC, CD, CD, A, B, C, D**
    - Requires O(l^2) space
  - Build indices for intra AS and inter AS segments
    - **A, B, C, D, AB, BC, CD**
    - Requires O(l) space

Step 1. IP to AS mapping

- Use BGP routing table snapshots:
  - An IP address is mapped to the *longest matching IP prefix* in a table,
  - Take the *last hop in the AS-PATH* as the origin AS



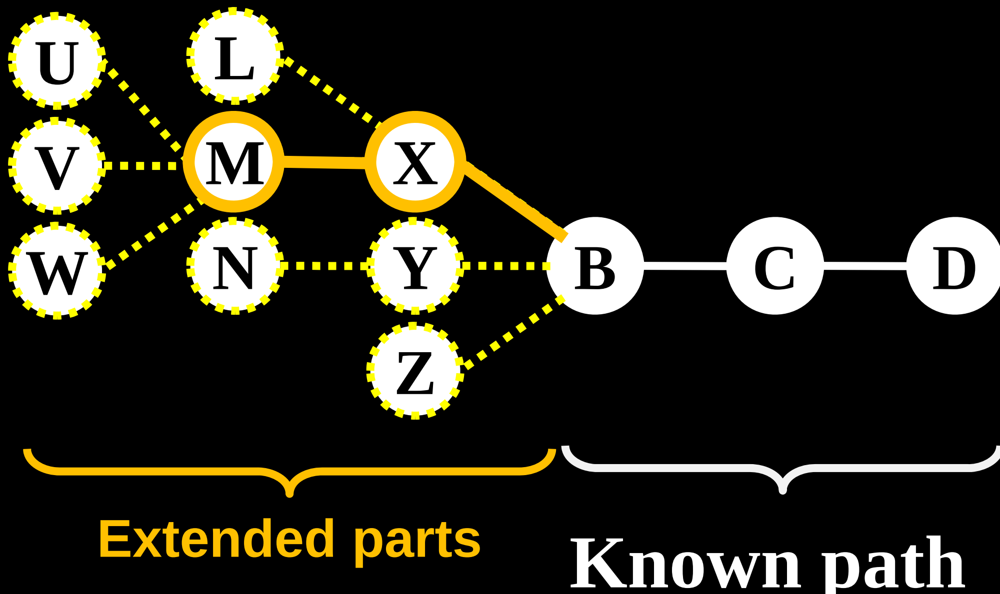
BGP Routing table snapshots.

Errors in IP to AS mapping

- Single origin AS mismatch
  - Mao et al reported that inaccurate mapping result in
    - Missing AS hop, extra AS hop, substitute AS hop, two hop AS loops
  - 8.9% AS paths contain two-hop AS loops
  - If we use the same IP-to-AS mapping for a query, the outcome would be consistent although mismatched.
- Multiple origin AS (MOAS)
  - 2,651,387 traceroutes have MOAS conflicts
  - 22.61% of MOAS are caused by Internet exchange prefixes
  - Infer AS paths from all MOASes

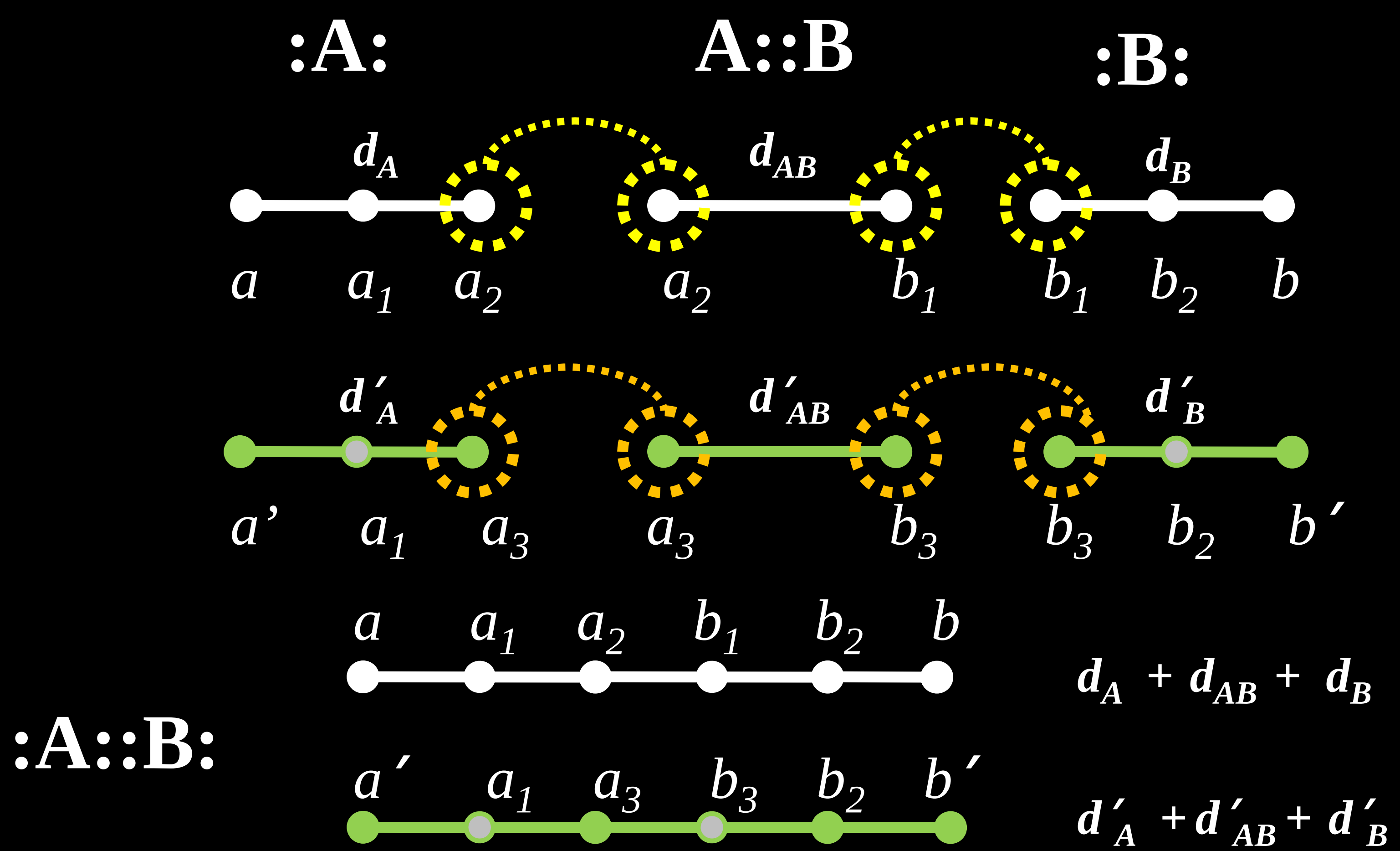
Step 2. AS path inference

- Qiu and Gao's methodology [GLOBECOM'06]
  - Exploits the AS paths, *known paths*, appeared in BGP routing tables.
  - Infer AS paths that satisfying *valley-free property* [L.Gao, TON'00]



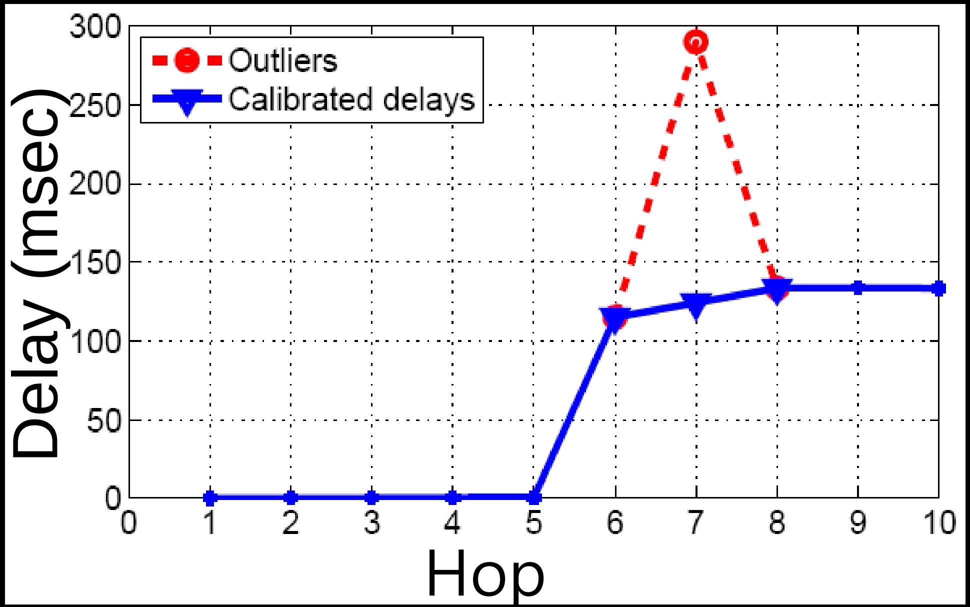
Choose shortest path with low ***unsure length*** and high ***frequency index***
Accuracy of 60% reported

Step 3. Stitching path segments



Sources of error – *traceroute*

- Dynamic nature of the Internet
  - » Record all reported measurement per path segment.
  - » Report the most recent or median of the past known history.
- Non-decreasing delay principle



When Path Stitching produces no stitched path

Case #1: No path segments in source/destination AS

The source or the destination is not in the same AS with any measurement data

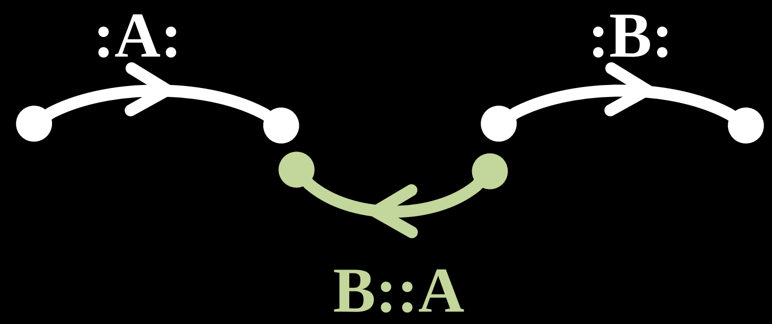
Data type	*Total AS*	*Transit AS*	*Stub AS*
Ark	14,378	4,418	***9,960***
BGP	28,244	4,847	***23,397***

- For 90% of undiscovered AS in Ark, the traceroute did not reach to AS
- ASes not covered by Ark accounts for only 110M or 5.8% of IP addresses in BGP

Case #2: No segments in the middle of inferred AS path

No inter-domain path segment

- Incorporating the reverse inter-domain segments

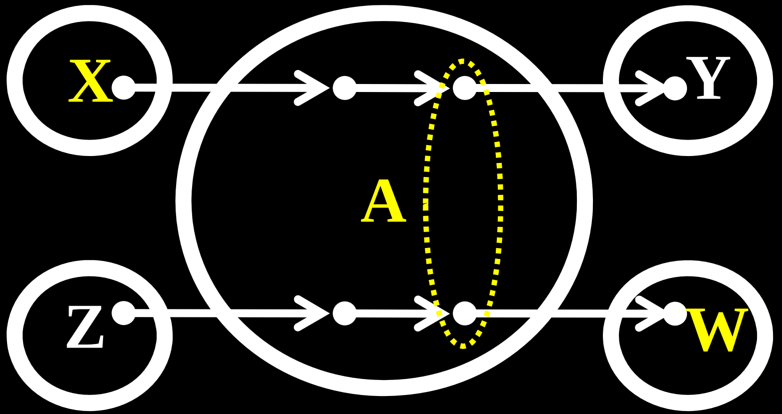


No intra-domain path segment

- No solution yet

Case #3: Segments does not rendezvous at the same address

For all ASes along the path has segments, but they do not rendezvous at the same address



X::A::W = ?

- Clustering heuristics:
  - Identifying IP address of *the same router*
  - Clustering IP addresses *in a single Point-of-presence (PoP)*
  - Clustering two ending points based on their ***IP prefix proximity***

When Path Stitching produces multiple paths

- Rank stitched paths using preference rules
- Same destination bound path segments
  - The more same destination bound path segments in a stitched path, the more this path is close to the real path
- Closeness to source and destination
  - For 20% of ASes, delay difference of path segments in an AS is larger than 100ms

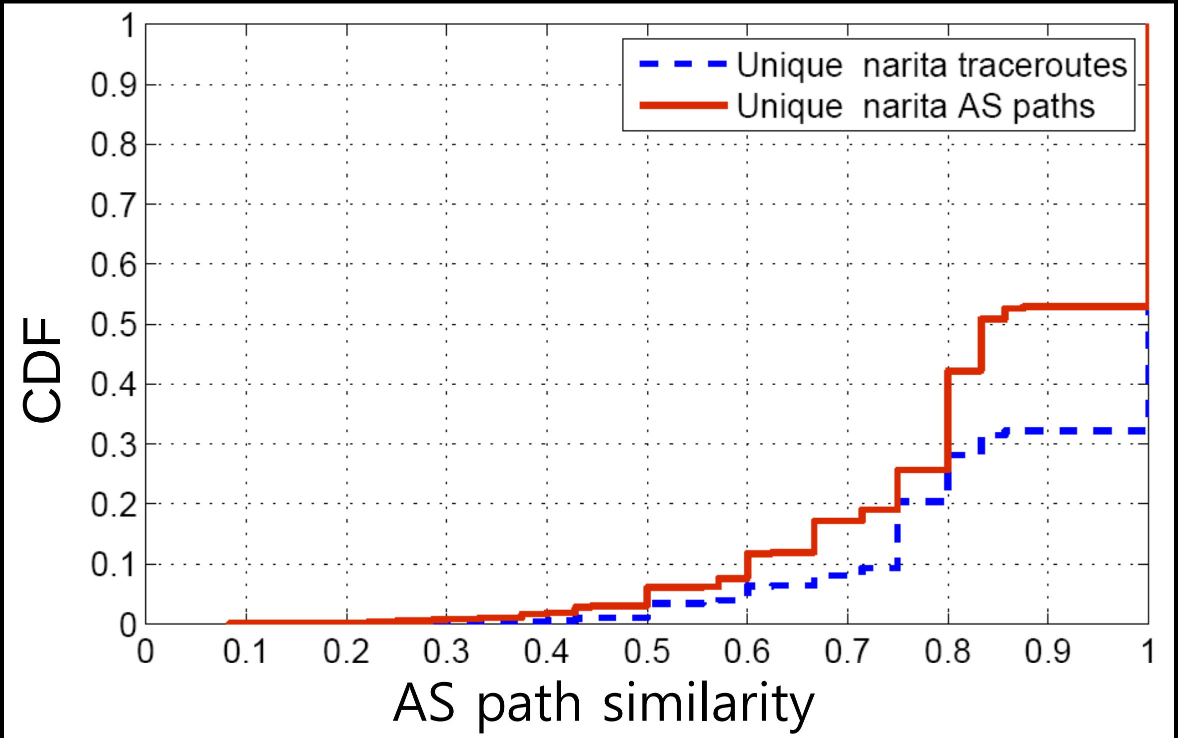
Evaluation

- Evaluate:
  1. Similarity between inferred AS path and AS path mapped from traceroutes
  2. Effectiveness of approximation heuristics

- Data set for evaluation:
  - **`narita`** : traceroute outputs from Ark monitor *nrt-jp* (Collected on April 11)

AS path similarity

- How close is inferred AS path to the AS path from traceroutes?



  - 68% of inferred paths match the `narita` paths exactly.
  - 24% of inferred paths are shorter than `narita` paths.

Effectiveness of approximation heuristics

- No stitched path without approximation

959(11%) path segment missing on inferred AS path

1453(16%) no stitched path

1724(20%) /28 clustering

2492(29%) Router/PoP clustering

2051(24%) Reverse segments

» Router/PoP clustering and /28 IP prefix clustering significantly *enlarge the coverage*.

Conclusions

- Path and latency prediction by combining traceroutes and BGP data
- Our approach uses existing measurement data and do no additional measurement
- Evaluation results are preliminary, but promising

Future Work

- Devise a mechanism to select a best path amongst many stitched paths
- Incorporate more datasets to improve coverage and accuracy
- Include performance metrics to include bandwidth and loss rate
- Build and deploy DNS-like system in the real-world

Thank you!

- Any question?

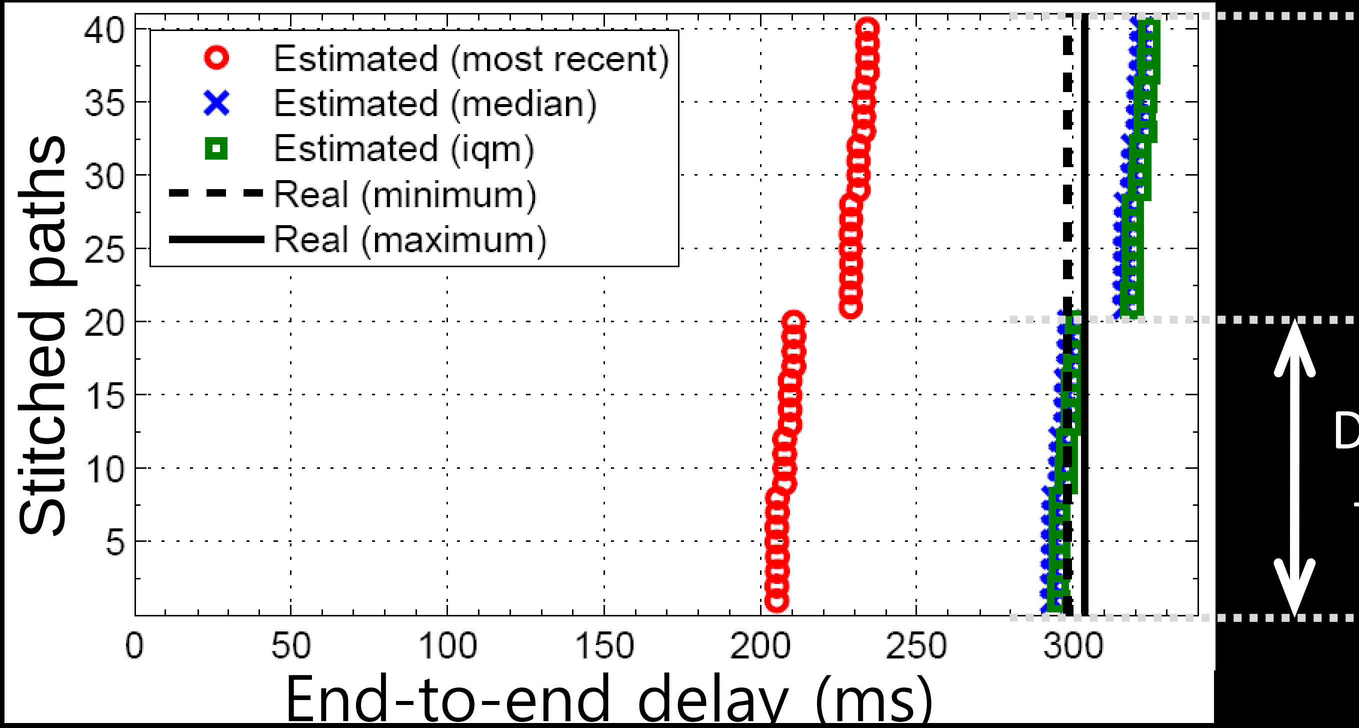
- For more question:

  keonjang@an.kaist.ac.kr

Same destination-bound preference

- planetlab2.xeno.cl.***cam.ac.uk***

  → pl1-higashi.ics.es.***osaka-u.ac.jp***

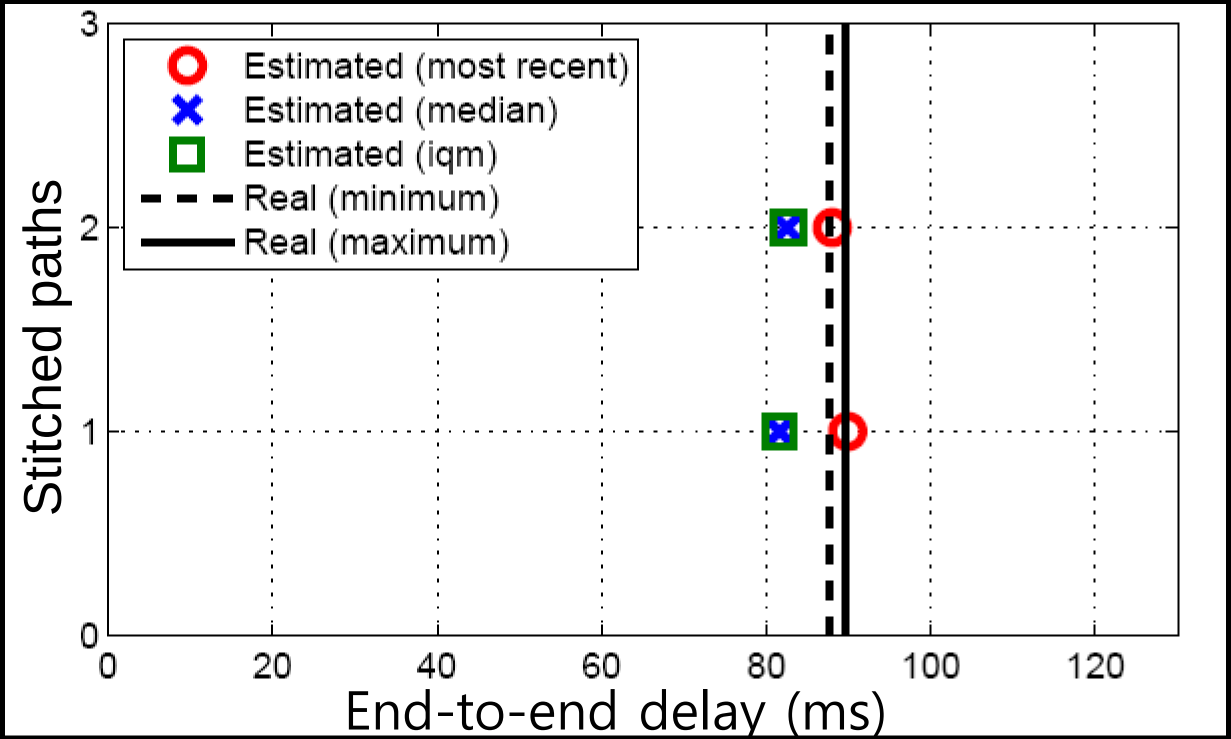


» Preference ***to the same destination-bound*** path segments

Closeness to source and destination

- Planetlab2.csil.***mit.edu***

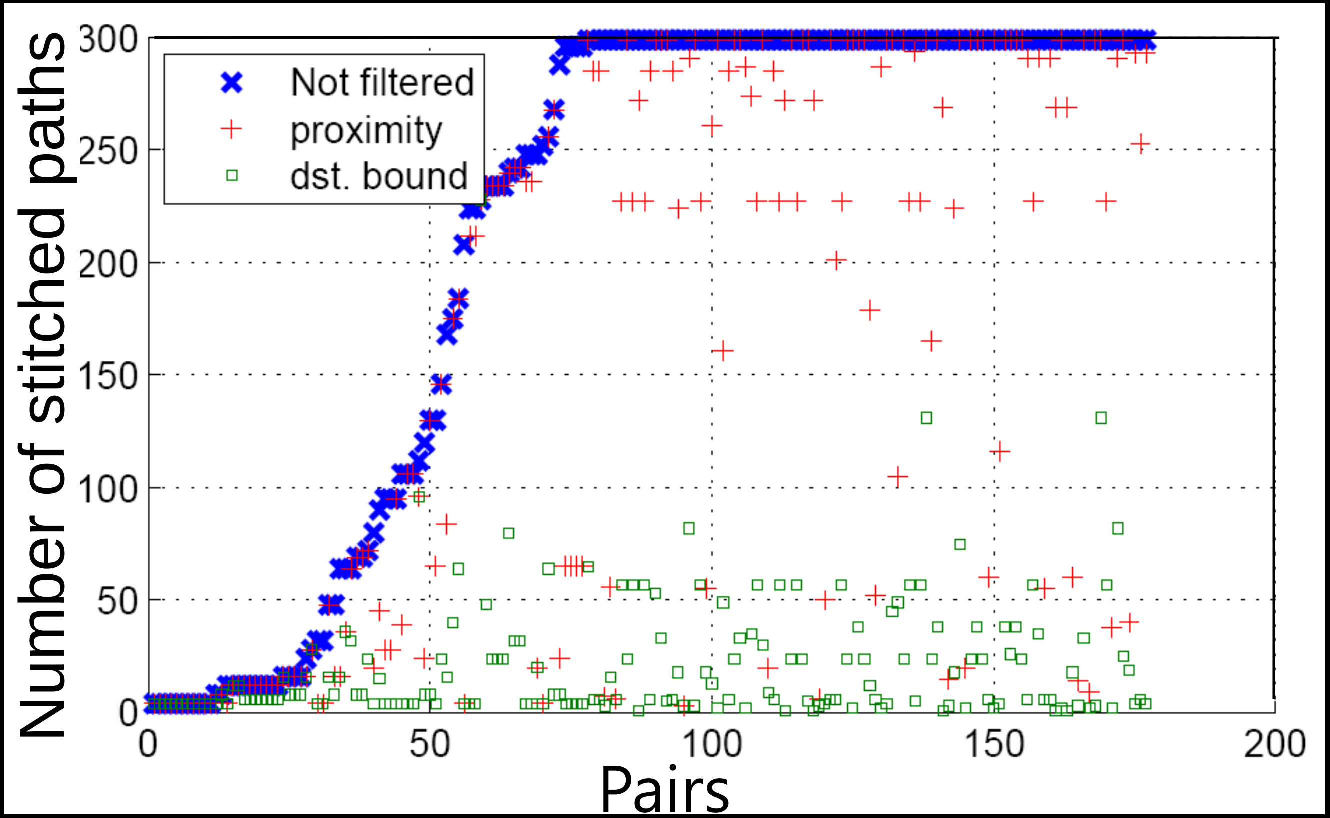
→ planet2.scs.***stanford.edu***



- In 20 % of Ases, delay difference within an AS is > 100 ms.

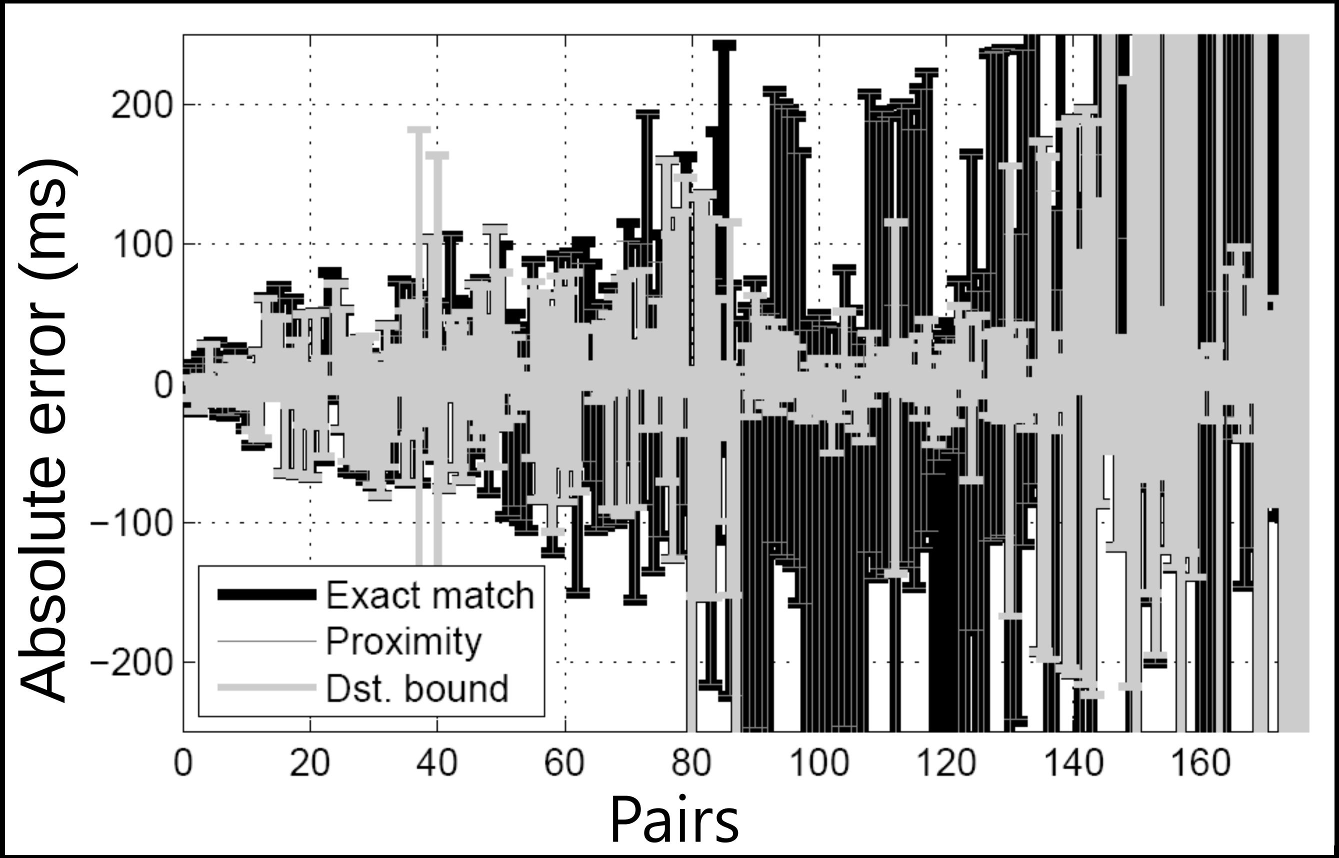
» Preference ***to the closest points*** in source and destination ASes

Preference rules



» Destination-bound and proximity rules prune large amounts of spurious paths

Preference rules



» Destination bound and proximity rules help to *improve accuracy*