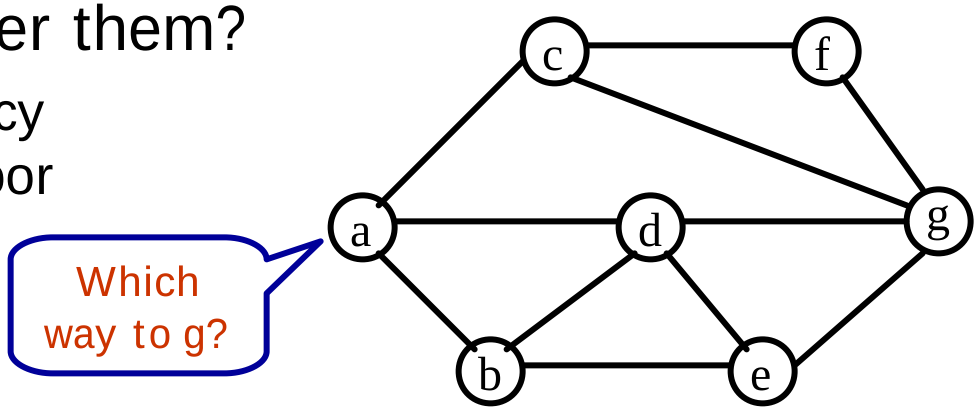
Inferring Link Weights using End-to-End Measurements

Ratul Mahajan
Neil Spring
David Wetherall
Tom Anderson

University of Washington

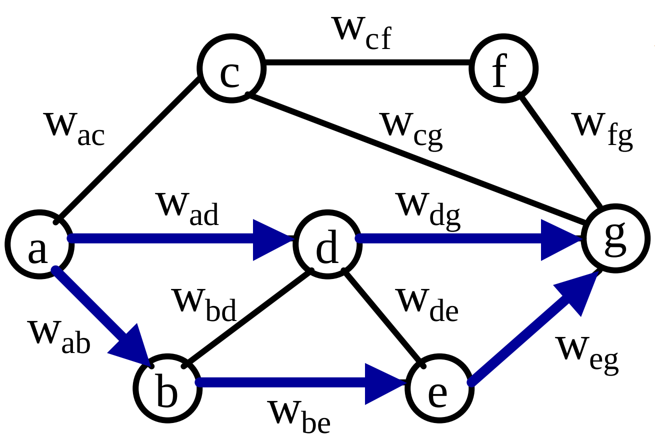
Motivation: topology → routing

- Accurate and detailed ISP topologies are now available
- But how to route over them?
  - Hop count and latency based models are poor



- Obtain a link weight based routing model
  - Most common model (OSPF, IS-IS, RIP)
  - Disclaimer: these are not the real weights!
- Also helpful in understanding intra-domain traffic engineering

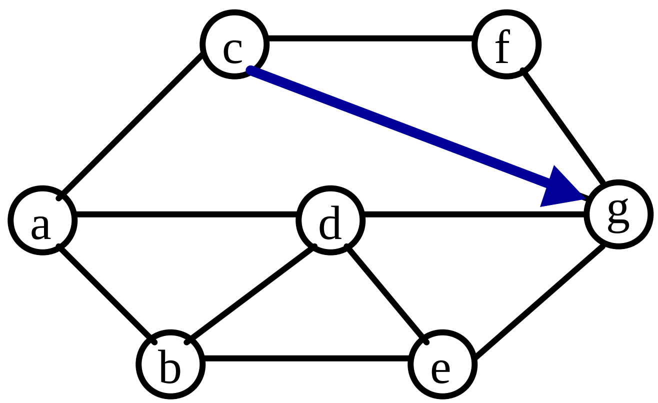
Problem definition, basic solution



- Given:
  - Map of a network w/ weighted shortest path routing
  - Routing – *chosen* paths between node pairs
- Wanted:
  - Weights that characterize routing

- Keys to the solution
  - All chosen paths between a node-pair have the same weight (ECMP)
  - This weight is less than that of other possible paths
- A constraint-based solution
  1. $w_{ad} + w_{dg} = w_{ab} + w_{be} + w_{eg}$ [ADG=ABEG]
  2. $w_{ad} + w_{dg} < w_{ac} + w_{cg}$ [ADG<ACG]
  3. $w_{ad} + w_{dg} < w_{ac} + w_{cf} + w_{fg}$ [ADG<ACFG]
  4. $w_{ad} + w_{dg} < w_{ab} + w_{bd} + w_{dg}$ [ADG<ABDG]
  5. $w_{ad} + w_{dg} < w_{ad} + w_{de} + w_{eg}$ [ADG<ADEG]
  6. $w_{ad} + w_{dg} < w_{ab} + w_{bd} + w_{de} + w_{eg}$ [ADG<ABDEG]

Making it tractable



- Problem: too many constraints
  - Exponential in number of nodes
- Solution: use knowledge of chosen paths between other node-pairs to remove redundant constraints

- Example
  - CG is a chosen path
  - The following exists in the system
    - $w_{cg} < w_{cf} + w_{fg}$

1. $w_{ad} + w_{dg} = w_{ab} + w_{be} + w_{eg}$
2. $w_{ad} + w_{dg} < w_{ac} + w_{cg}$
3. $w_{ad} + w_{dg} < w_{ac} + w_{cf} + w_{fg}$
4. $w_{ad} + w_{dg} < w_{ab} + w_{bd} + w_{dg}$
5. $w_{ad} + w_{dg} < w_{ad} + w_{de} + w_{eg}$
6. $w_{ad} + w_{dg} < w_{ab} + w_{bd} + w_{de} + w_{eg}$

Hello, real world!

Limitations of routing information gathered using traceroute

- Problem: some observed paths are non-chosen paths
  - Due to transient events such as failures
  - Renders the constraint system inconsistent
  - Solution: use error variables, minimize the weighted sum of errors

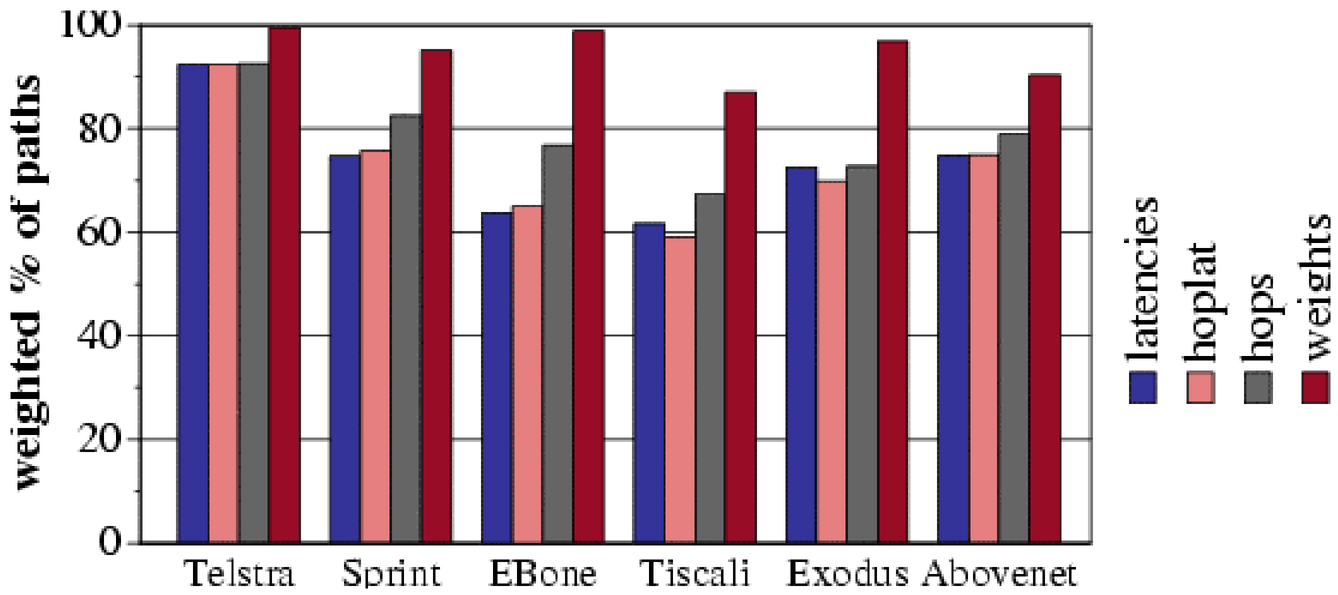
  1. $w_{ad} + w_{dg} - e_{adg} = w_{ab} + w_{be} + w_{eg} - e_{abeg}$
  2. $w_{ad} + w_{dg} - e_{adg} < w_{ac} + w_{cg}$

- Problem: all chosen paths between a node-pair may not be observed
  - Due to a small number of measurements between the node-pair
  - $w_{ad} + w_{dg} - e_{adg} < w_{ac} + w_{cg}$ (but ACG may also be a chosen path for a→g)
  - Solution: $w_{ad} + w_{dg} - e_{adg} <= w_{ac} + w_{cg}$

Evaluation

- Dataset: backbone topologies collected by Rocketfuel
  - 600+ vantage points, 9-200K+ traceroutes
  - Telstra (au), Ebone, Tiscali (eu), Abovenet, Exodus, Sprint (us)
- Compare the inferred weights with three alternate models
  - *Hops:* Minimum hop count path
  - *Latency:* Minimum latency (geographical) path
  - *HopLat:* Minimum latency minimum hop count path
- Criteria
  1. What fraction of all observed paths fit?
  2. What fraction of dominant paths fit
  3. What is the accuracy of multi-path prediction?

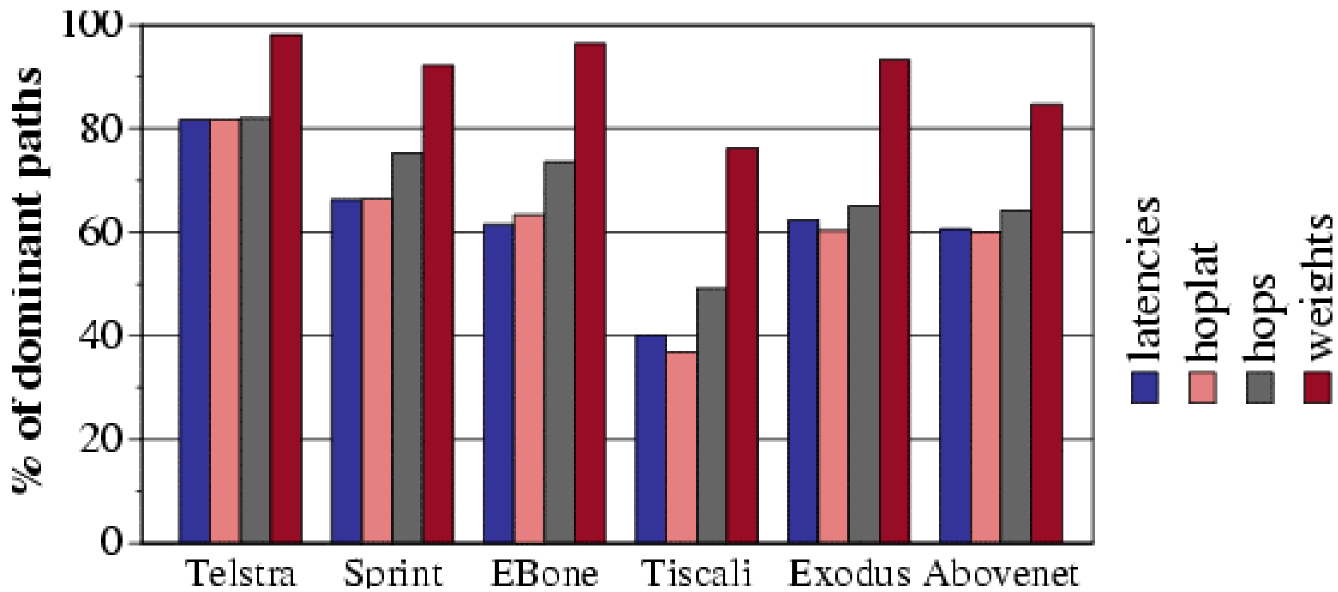
Fraction of all paths that fit



- Weights describe the routing well
  - Weights: 87-99%
  - Hops: 67-92% (best alternate metric)
    - Performance level of hops is misleading (2 slides away)

Fraction of dominant paths that fit

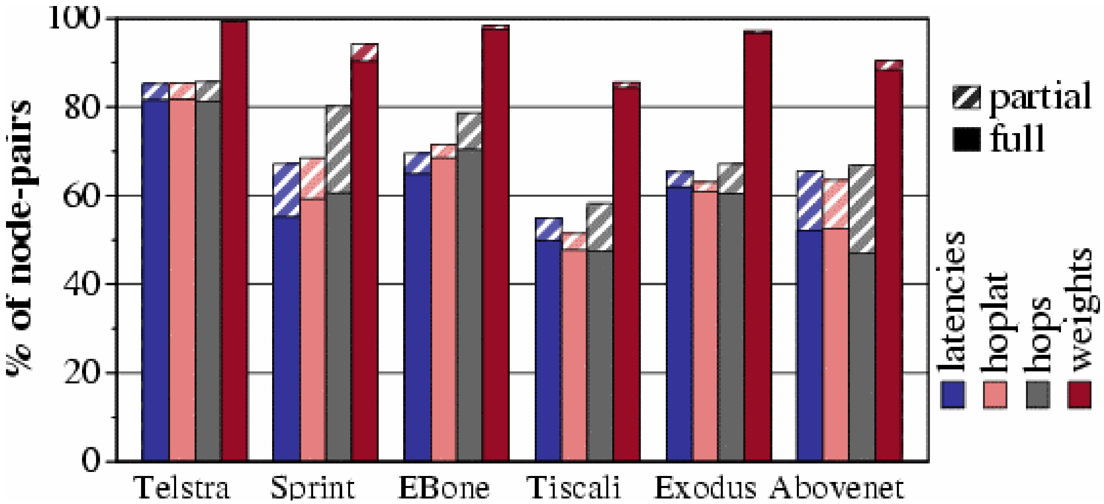
- Dominant path: most common path between a node-pair



- Weights fit more dominant paths
  - Weights: 76-98%
  - Hops: 49-82% (best alternate metric)

Accuracy of multi-path prediction

- Classify routing characterization between a node-pair as one of
  - Full: all predicted paths were observed (accurate)
  - Partial: some predicted path was not observed (over prediction)
  - None: none of the predicted paths was observed



- Hops tends to predict more paths as being the preferred paths
  - 4-20% node-pairs are partial, only 47-81% full
- Weights: 84-99% full, 1-3% partial

Conclusions

- A novel constraint-based approach to approximate intra-domain link weights
- The inferred weights characterize intra-domain routing better than hop count and latency based metrics
  - Good predictive power
- Future work
  - Investigate the “realism” of our weights
    - Predict backup paths
  - Understand intra-domain traffic engineering policies
  - Study link weight changes and link failures