Tomo-gravity

Yin Zhang Matthew Roughan

Nick Duffield Albert Greenberg

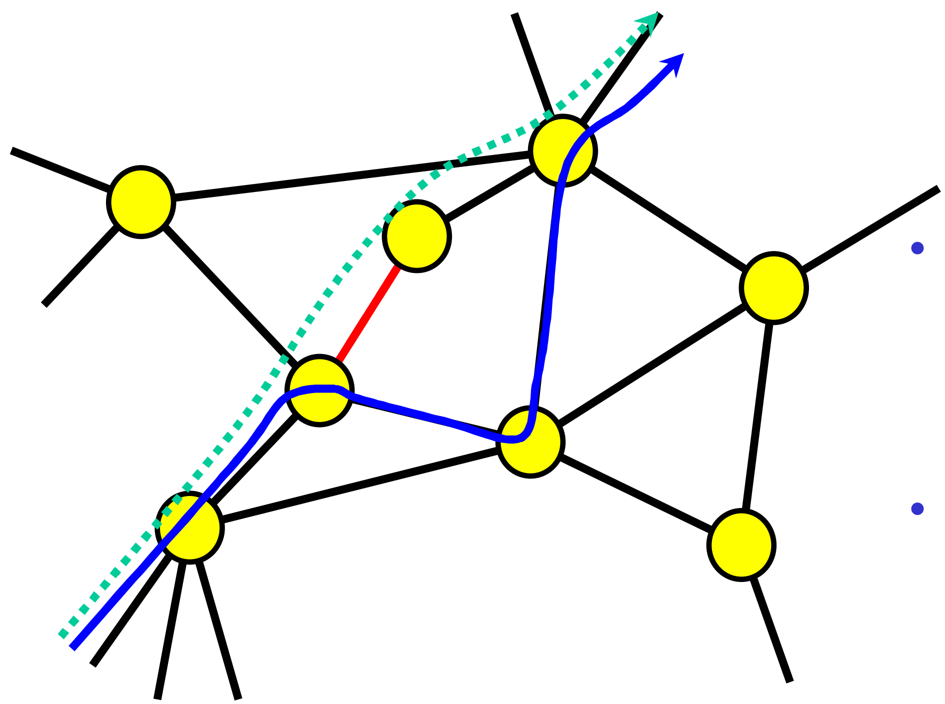
"A Northern NJ Research Lab"

{yzhang,roughan,duffield,albert}@research.att.com

ACM SIGMETRICS 2003

Network Engineering

Routes change under failures



- Reliability analysis
  - Predicting traffic under planned or unexpected router/link failures
- Traffic engineering
  - Optimizing OSPF weights to minimize congestion
- Capacity planning
  - Forecasting future capacity requirements

Can we do route optimization (or network engineering in general)?

- Feldmann et al. 2000
- Shaikh et al. 2002

- Fortz et al., 2002
- Roughan et al. 2003

Tomo-gravity

A3: "Well, we don't know the topology, we don't know the traffic matrix, the routers don't automatically adapt the routes to the traffic, and we don't know how to optimize the routing configuration. But, other than that, we're all set!"

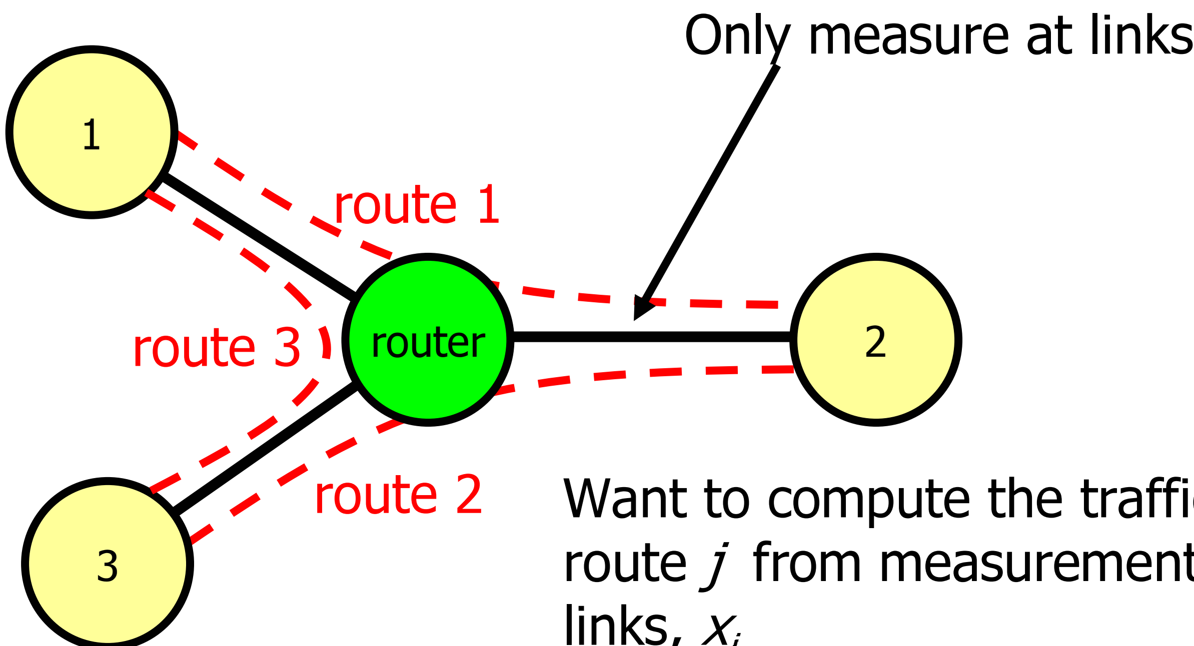
Central Problem: No Traffic Matrix

- For large IP networks, don't have good traffic matrix
  - Widely available SNMP measurements provide only link loads
    - Even this data is not perfect (glitches, loss, …)
- *As a result*, IP network engineering is more art than science
  - Yet, need accurate, automated, scientific tools for reliability analysis, capacity planning, traffic engineering

Tomo-gravity Solution

- Tomo-gravity infers traffic matrices from widely available measurements of link loads
  - Accurate: especially accurate for large elements
  - Robust: copes easily with data glitches, loss
  - Flexible: extends easily to incorporate more detailed measurements, where available
  - Fast: for example, solves AT&T's IP backbone network in a few seconds
- In daily use for AT&T IP network engineering
  - Reliability analysis, capacity planning, and traffic engineering

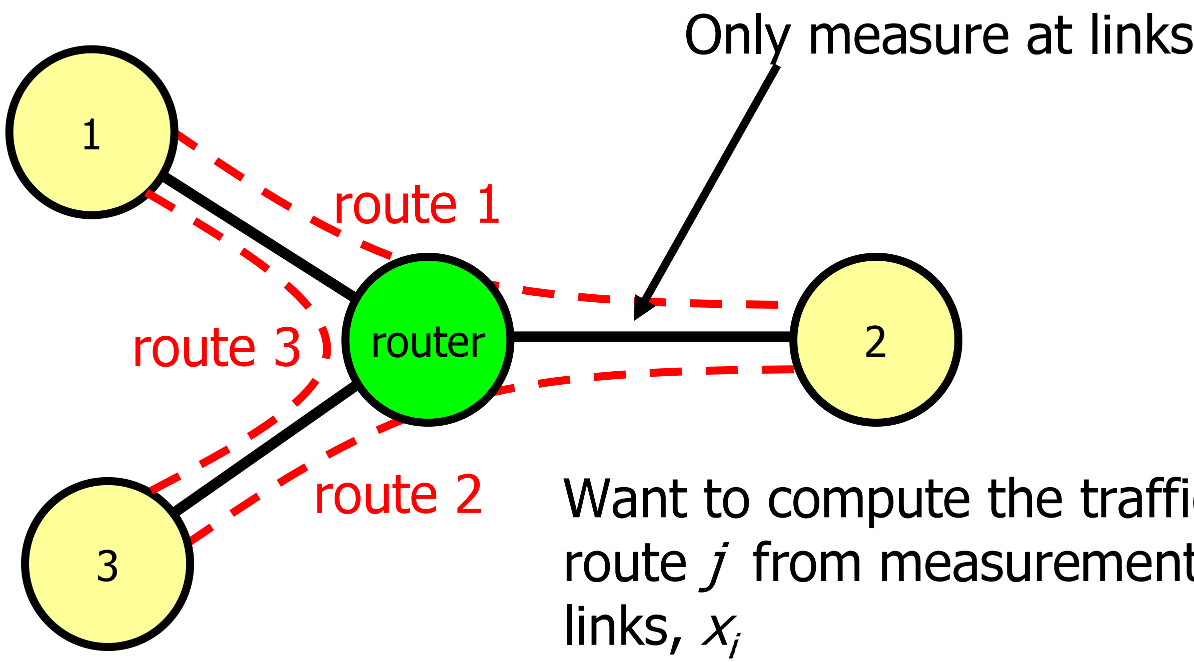
The Problem



Want to compute the traffic y_j along route j from measurements on the links, x_i

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

The Problem



Want to compute the traffic y_j along route j from measurements on the links, x_i

$$\mathbf{x} = \mathbf{A}^T \mathbf{y}$$

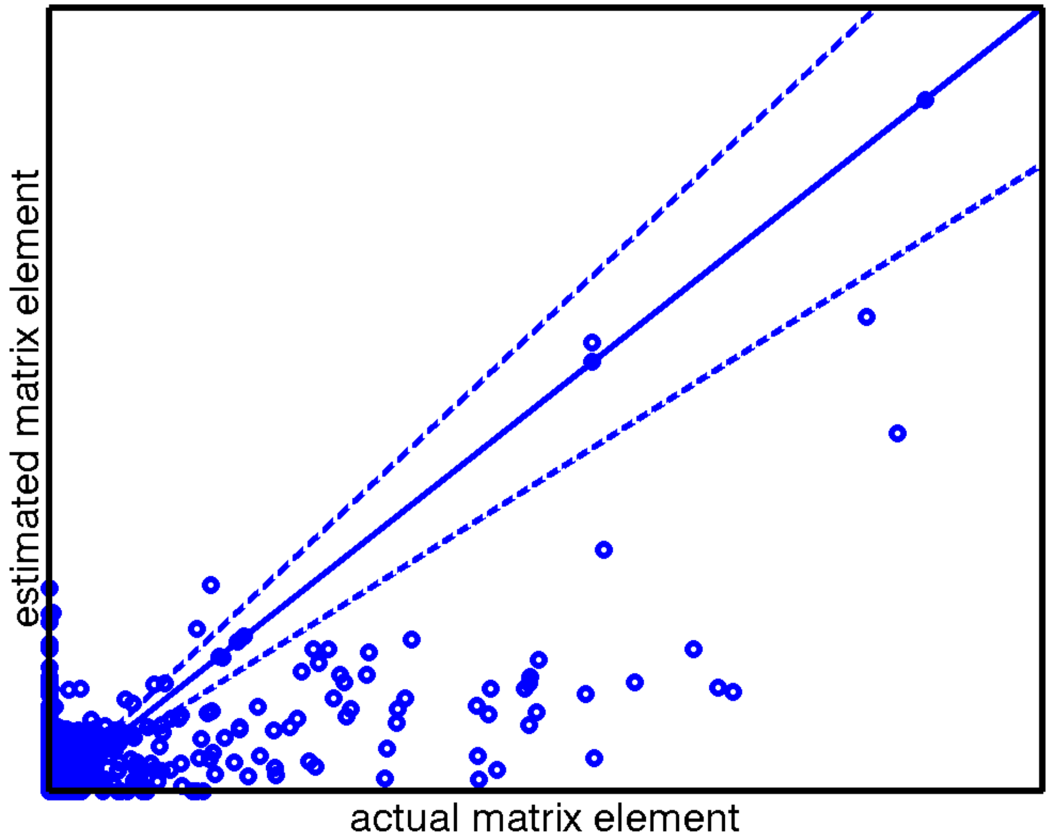
Approaches

- Existing solutions
  - Naïve (Singular Value Decomposition)
  - Gravity Modeling
  - Generalized Gravity Modeling
  - Tomographic Approach
- New solution
  - Tomo-gravity

How to Validate?

- Simulate and compare
  - Problems
    - How to generate realistic traffic matrices
    - Danger of generating exactly what you put in
- Measure and compare
  - Problems:
    - Hard to get Netflow (detailed direct measurements) along whole edge of network
      - If we had this, then we wouldn't need SNMP approach
    - Actually pretty hard to match up data
      - Is the problem in your data: SNMP, Netflow, routing, …
- Our method
  - Novel method for using partial, incomplete Netflow data

Naïve Approach



In real networks the problem is highly under-constrained

Simple Gravity Model

- Motivated by Newton's Law of Gravitation
- Assume traffic between sites is proportional to traffic at each site

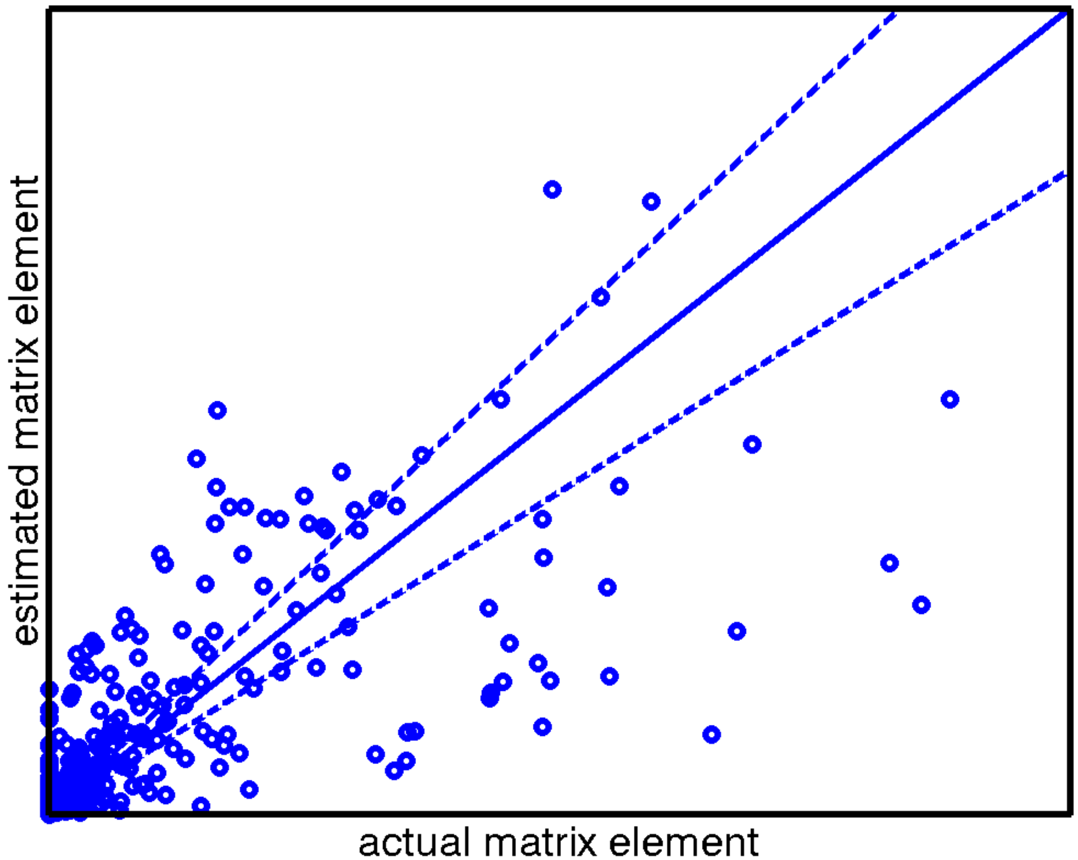
$$y_1 \propto x_1 \, x_2$$

$$y_2 \propto x_2 \, x_3$$

$$y_3 \propto x_1 \, x_3$$

- Assume there is no systematic difference between traffic in different locations
  - Only the total volume matters
  - Could include a distance term, but locality of information is not so important in the Internet as in other networks

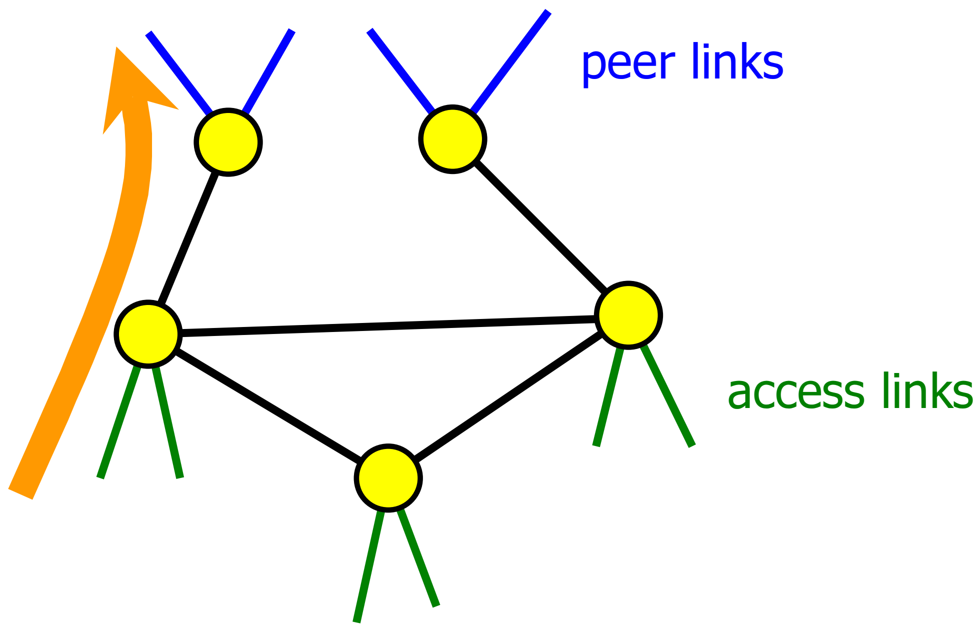
Simple Gravity Model



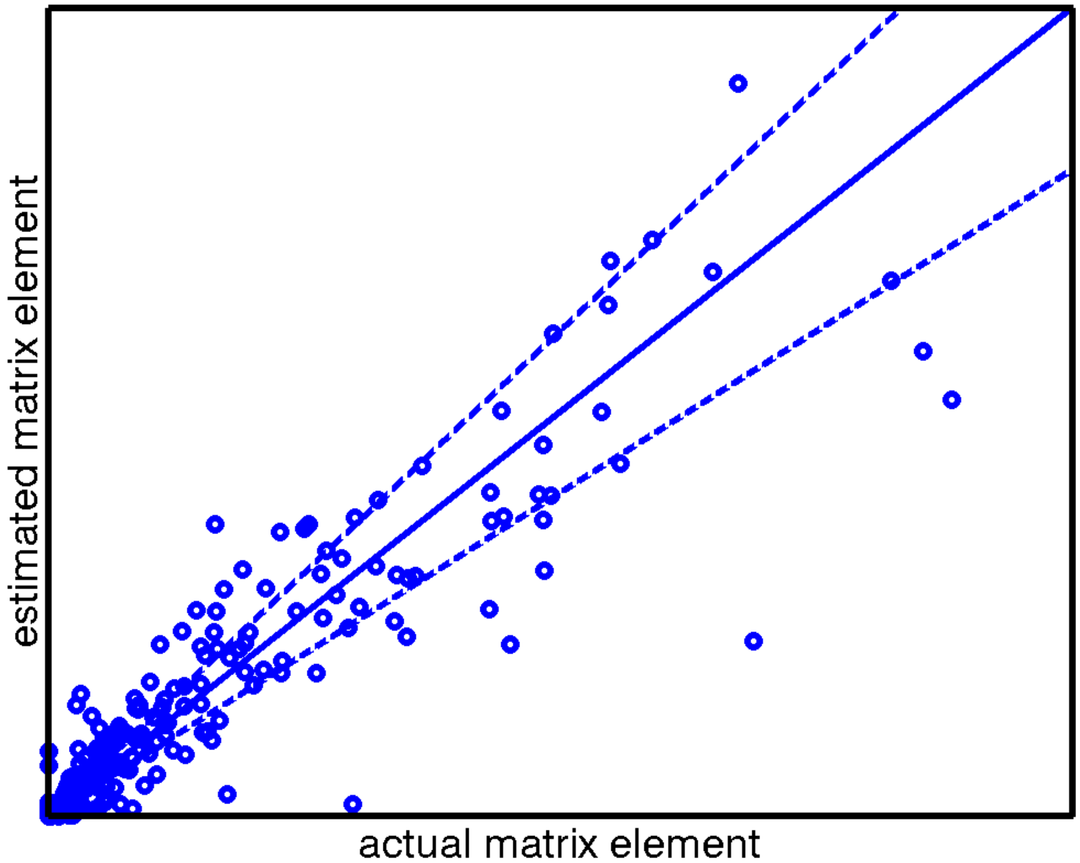
Better than naïve, but still not very accurate

Generalized Gravity Model

- Internet routing is asymmetric
  - Hot potato routing: use the closest exit point
- Generalized gravity model
  - For outbound traffic, assumes proportionality on per-peer basis (as opposed to per-router)



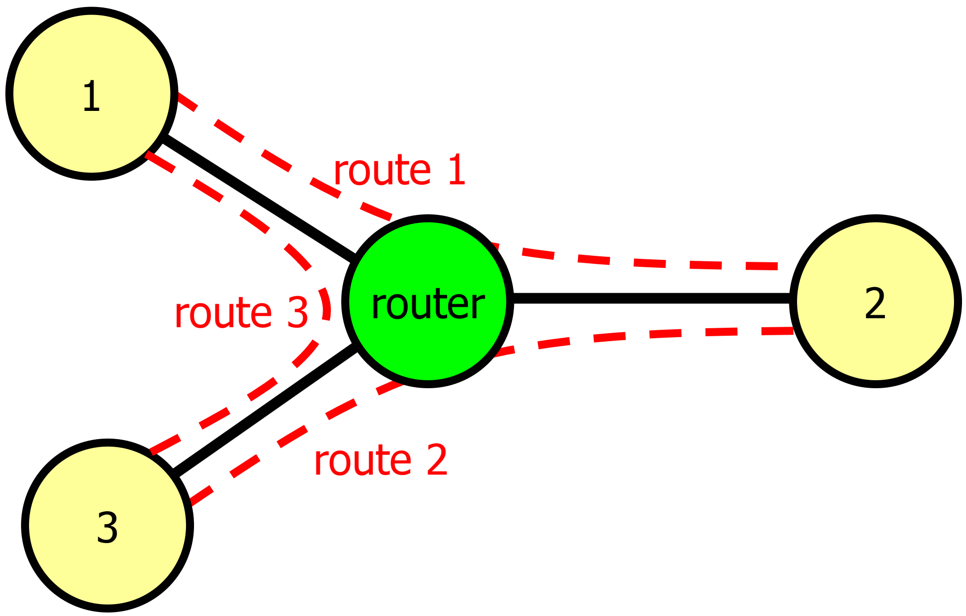
Generalized Gravity Model



Fairly accurate given that no link constraint is used

Tomographic Approach

- Apply the link constraints



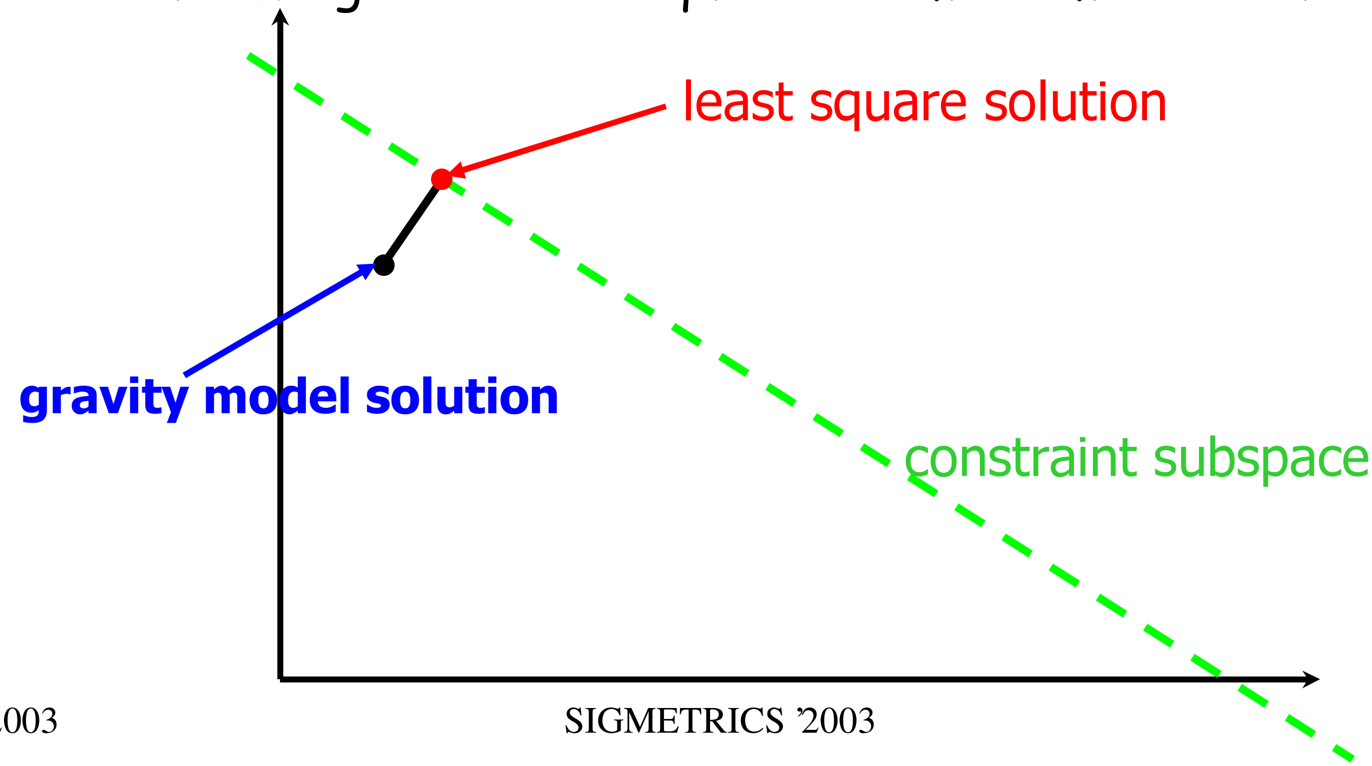
$$x = A^T y$$

Tomographic Approach

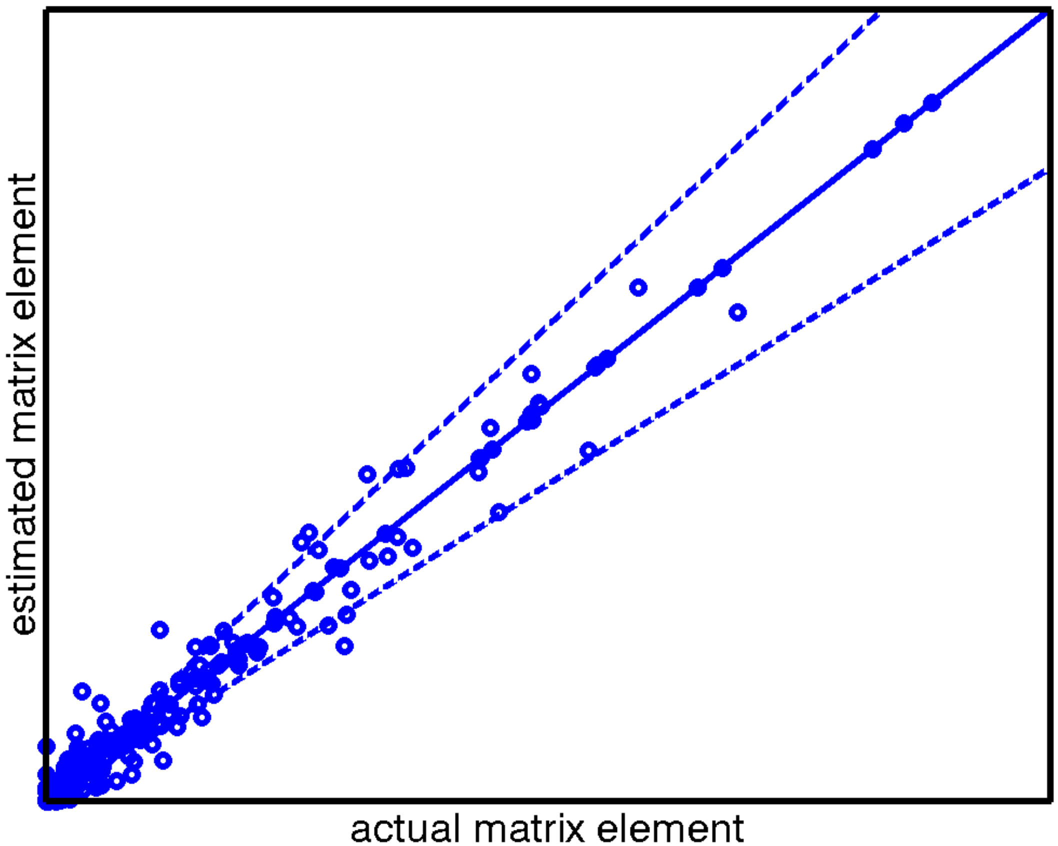
- Under-constrained linear inverse problem
- Find additional constraints based on models
  - Typical approach: use higher order statistics
- Disadvantages
  - Complex algorithm – doesn't scale
    - Large networks have 1000+ nodes, 10000+ routes
  - Reliance on higher order statistics is not robust given the problems in SNMP data
    - Artifacts, Missing data
    - Violations of model assumptions (e.g. non-stationarity)
    - Relatively low sampling frequency: 1 sample every 5 min
    - Unevenly spaced sample points
  - Not very accurate at least on simulated TM

Our Solution: Tomo-gravity

- "Tomo-gravity" = tomography + gravity modeling
- Exploit topological equivalence to reduce problem size
- Use least-squares method to get the solution, which
  - Satisfies the constraints
  - Is closest to the gravity model solution
  - Can use weighted least-squares to make more robust

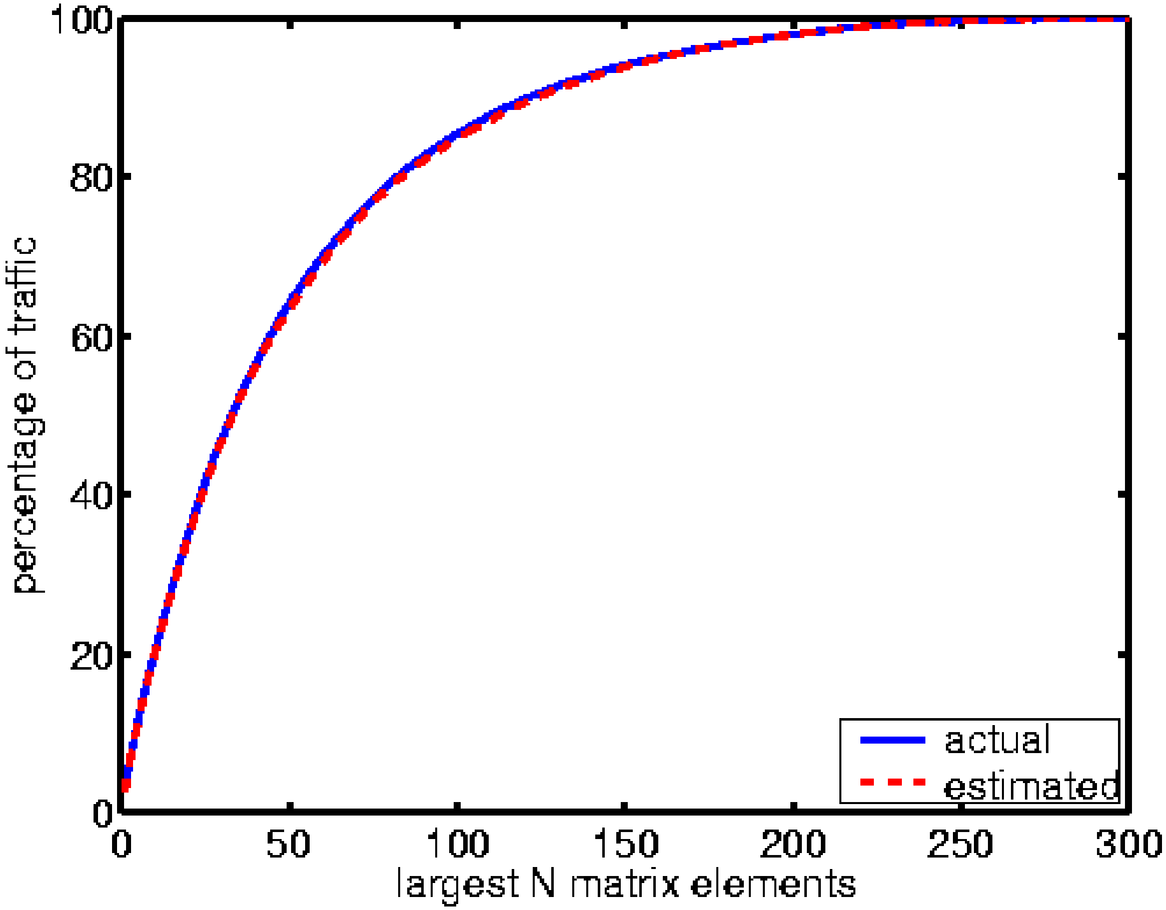


Tomo-gravity: Accuracy



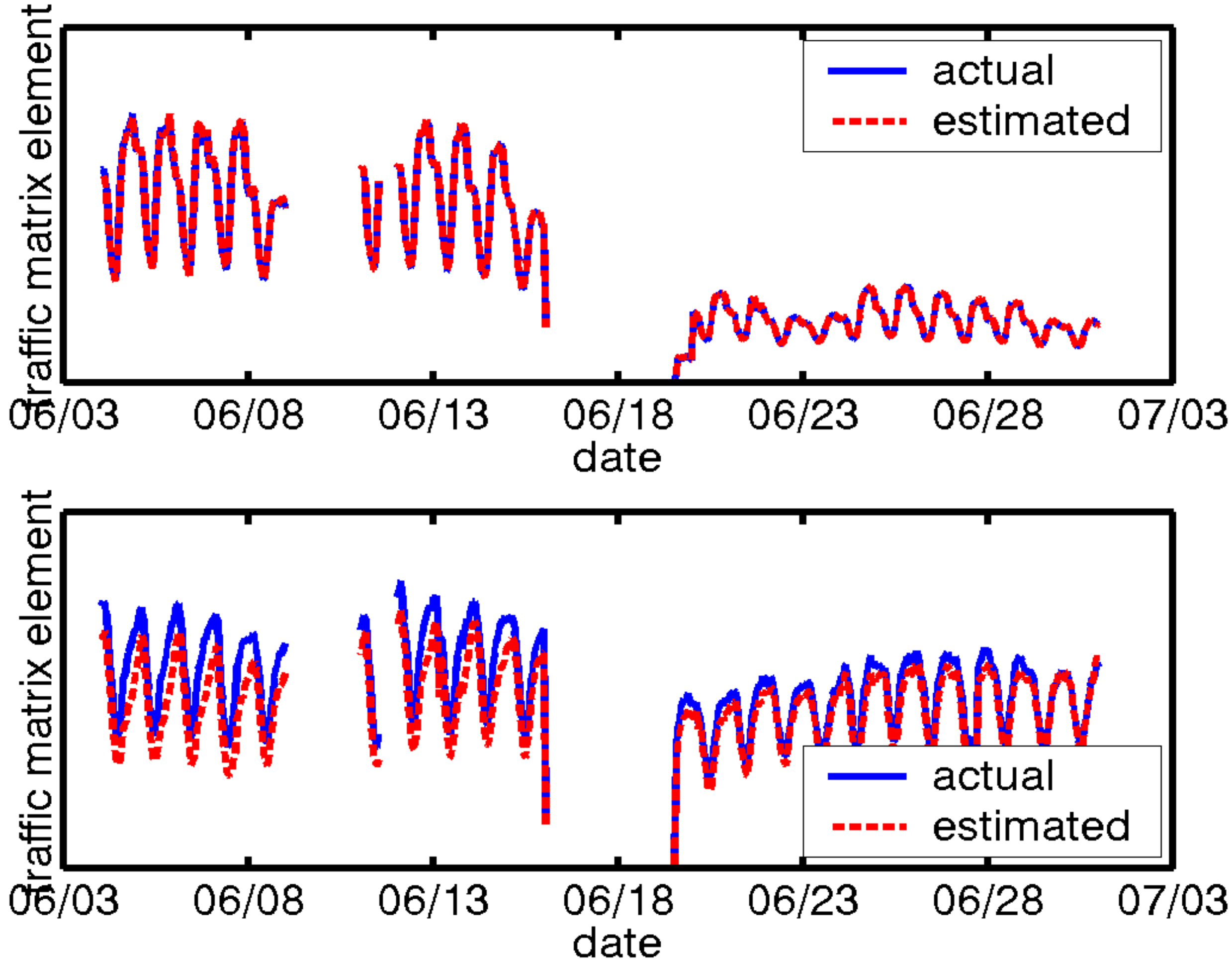
Accurate within 10-20% (esp. for large elements)

Distribution of Element Sizes



Estimated and actual distribution overlap

Estimates over Time



Consistent performance over time

Summary: Tomo-gravity Works

- Tomo-gravity takes the best of both tomography and gravity modeling
  - Simple, and quick
    - A few seconds for whole AT&T backbone
  - Satisfies link constraints
    - Gravity model solutions don't
  - Uses widely available SNMP data
    - Can work within the limitations of SNMP data
    - Only uses first order statistics → interpolation very effective
  - Limited scope for improvement
    - Incorporate additional constraints from other data sources: e.g., Netflow where available
- Operational experience very positive
  - In daily use for AT&T IP network engineering
  - Successfully prevented service disruption during simultaneous link failures

Future Work

- Understanding why the method works
  - Sigcomm 2003 paper provides solid foundation for tomo-gravity
- Building applications
  - Detect anomalies using traffic matrix time series

Thank you!

Backup Slide: Validation Method

- Use partial, incomplete Netflow data
  1. Measure partial traffic matrix $\mathbf{y_p}$
     - Netflow covers 70+% traffic
  2. Simulate link loads $\mathbf{x_p = A^T y_p}$
     - $\mathbf{x_p}$ won't match real SNMP link loads
  3. Solve $\mathbf{x_p = A^T y}$
  4. Compare $\mathbf{y}$ with $\mathbf{y_p}$
- Advantage
  - Realistic network, routing, and traffic
  - Comparison is direct, we know errors are due to algorithm not errors in the data
  - Can test robustness by adding noise to $\mathbf{x_p}$