Network Tomography and Internet Traffic Matrices

Matthew Roughan
School of Mathematical Sciences
University of Adelaide

<matthew.roughan@adelaide.edu.au>

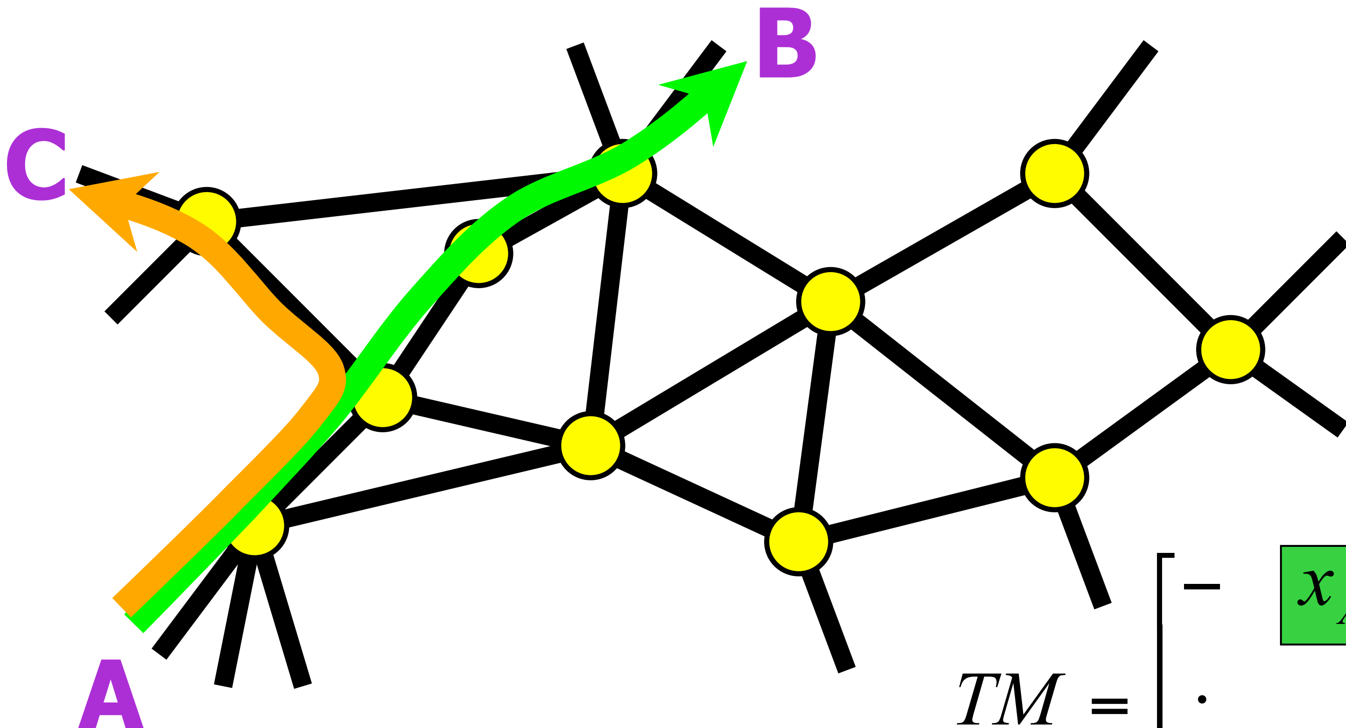
Credits

- David Donoho – Stanford
- Nick Duffield – AT&T Labs-Research
- Albert Greenberg – AT&T Labs-Research
- Carsten Lund – AT&T Labs-Research
- Quynh Nguyen – AT&T Labs
- Yin Zhang – AT&T Labs-Research

Problem

Have link traffic measurements

Want to know demands from source to destination

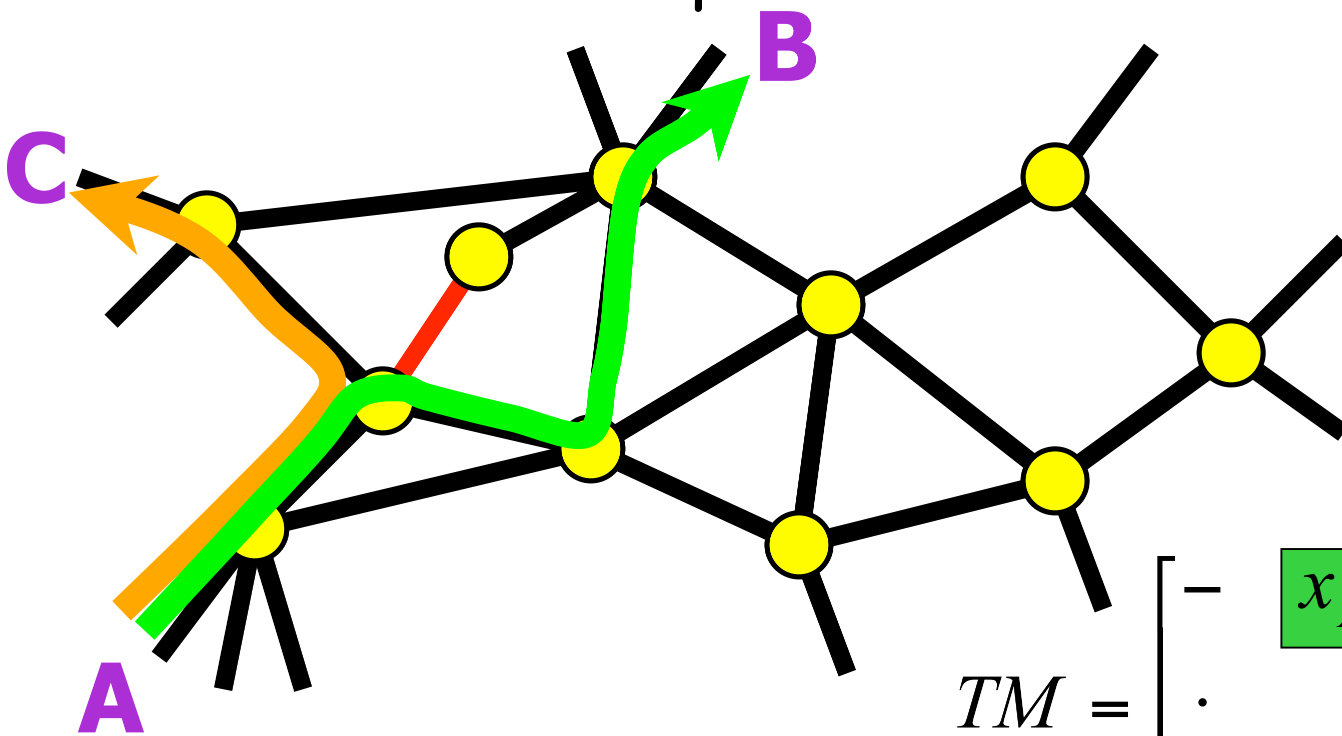


$$TM = \begin{bmatrix} - & \boxed{x_{A,B}} & \boxed{x_{A,C}} & \cdots \\ \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdots \end{bmatrix}$$

Example App: reliability analysis

Under a link failure, routes change

want to predict new link loads



$$TM = \begin{bmatrix} - & \boxed{x_{A,B}} & \boxed{x_{A,C}} & \cdots \\ \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdots \end{bmatrix}$$

Network Engineering

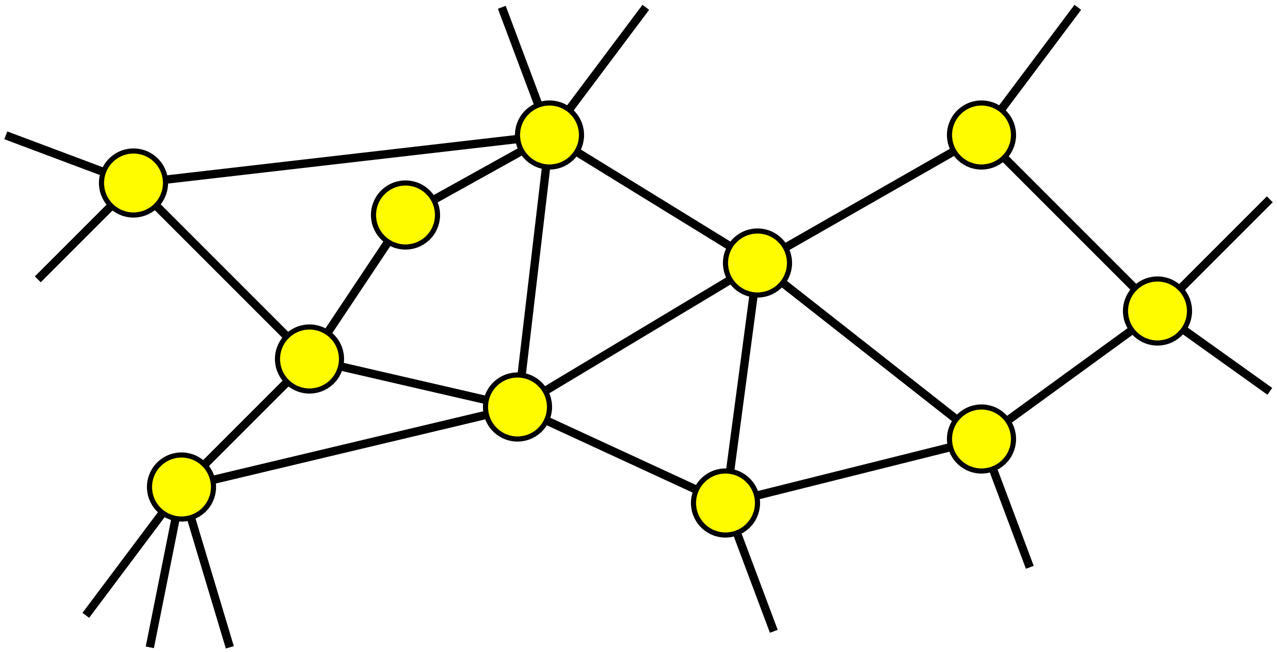
- What you want to do
  - a) Reliability analysis
  - b) Traffic engineering
  - c) Capacity planning
- What do you need to know
  - Network and routing
  - Prediction and optimization techniques
  - ? Traffic matrix

Outline

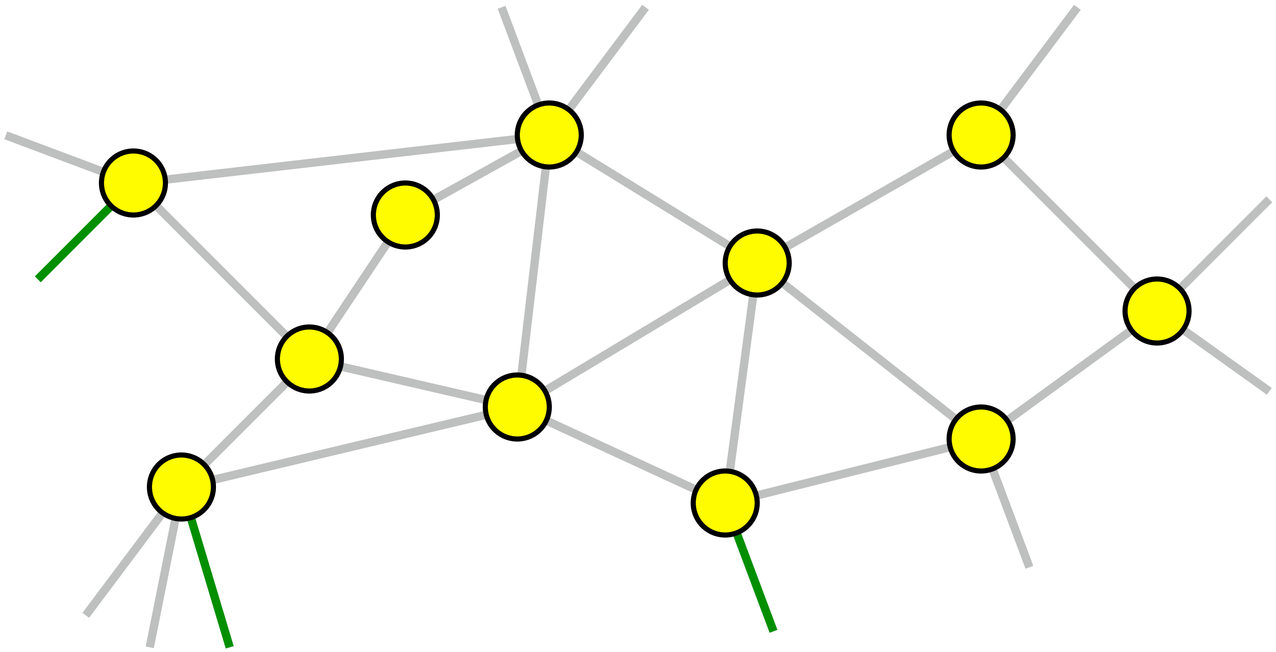
- Part I: What do we have to work with - data sources
  - SNMP traffic data
  - Netflow, packet traces
  - Topology, routing and configuration
- Part II:Algorithms
  - Gravity models
  - Tomography
  - Combination and information theory
- Part III: Applications
  - Network Reliability analysis
  - Capacity planning
  - Routing optimization (and traffic engineering in general)

Part I: Data Sources

Traffic Data

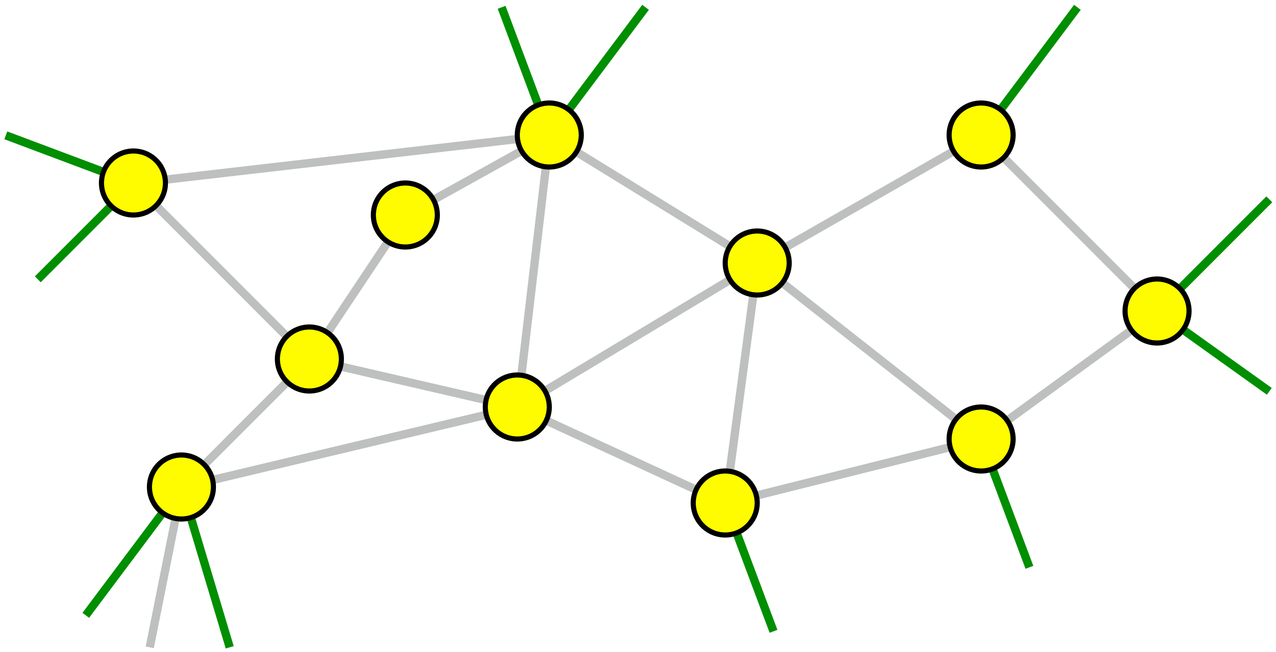


Data Availability - packet traces



Packet traces limited availability – like a high zoom snap shot
• special equipment needed (O&M expensive even if box is cheap)
• lower speed interfaces (only recently OC192)
• huge amount of data generated

Data Availability - flow level data

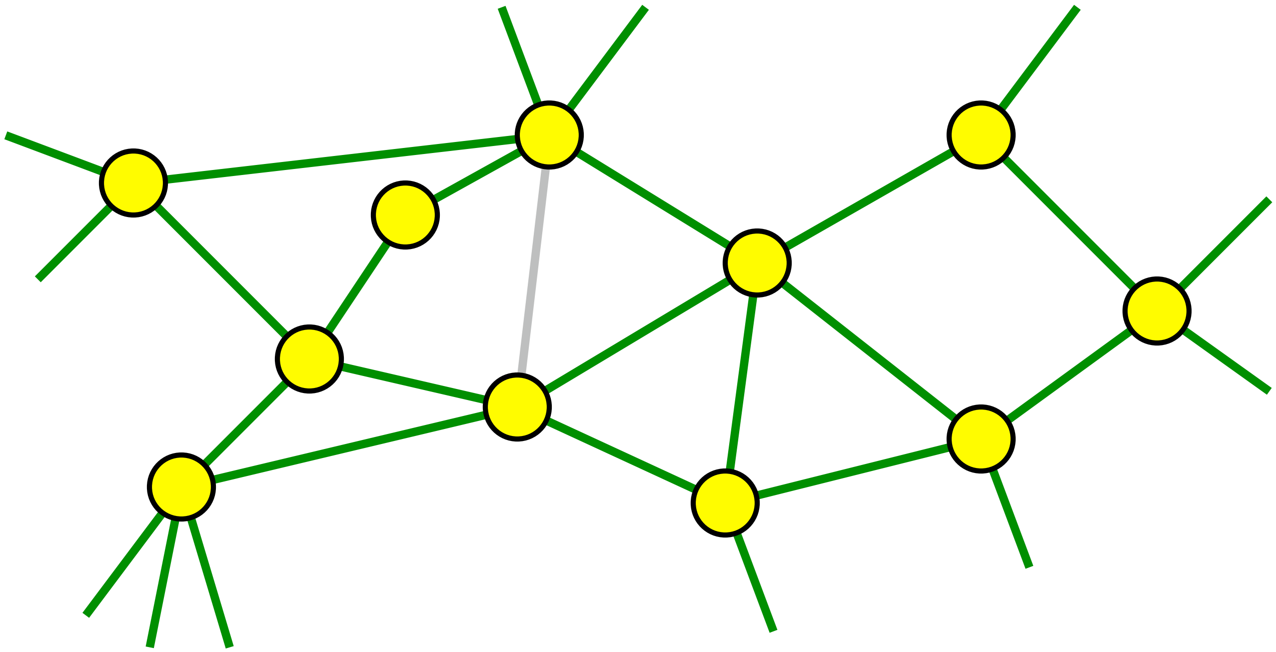


Flow level data not available everywhere – like a home movie of the network
• historically poor vendor support (from some vendors)
• large volume of data (1:100 compared to traffic)
• feature interaction/performance impact

Netflow Measurements

- Detailed IP flow measurements
  - Flow defined by
    - Source, Destination IP,
    - Source, Destination Port,
    - Protocol,
    - Time
  - Statistics about flows
    - Bytes, Packets, Start time, End time, etc.
  - Enough information to get traffic matrix
- Semi-standard router feature
  - Cisco, Juniper, etc.
  - not always well supported
  - potential performance impact on router
- Huge amount of data (500GB/day)

Data Availability - SNMP



SNMP traffic data – like a time lapse panorama
- MIB II (including IfInOctets/IfOutOctets) is available almost everywhere
- manageable volume of data (but poor quality)
- no significant impact on router performance

SNMP

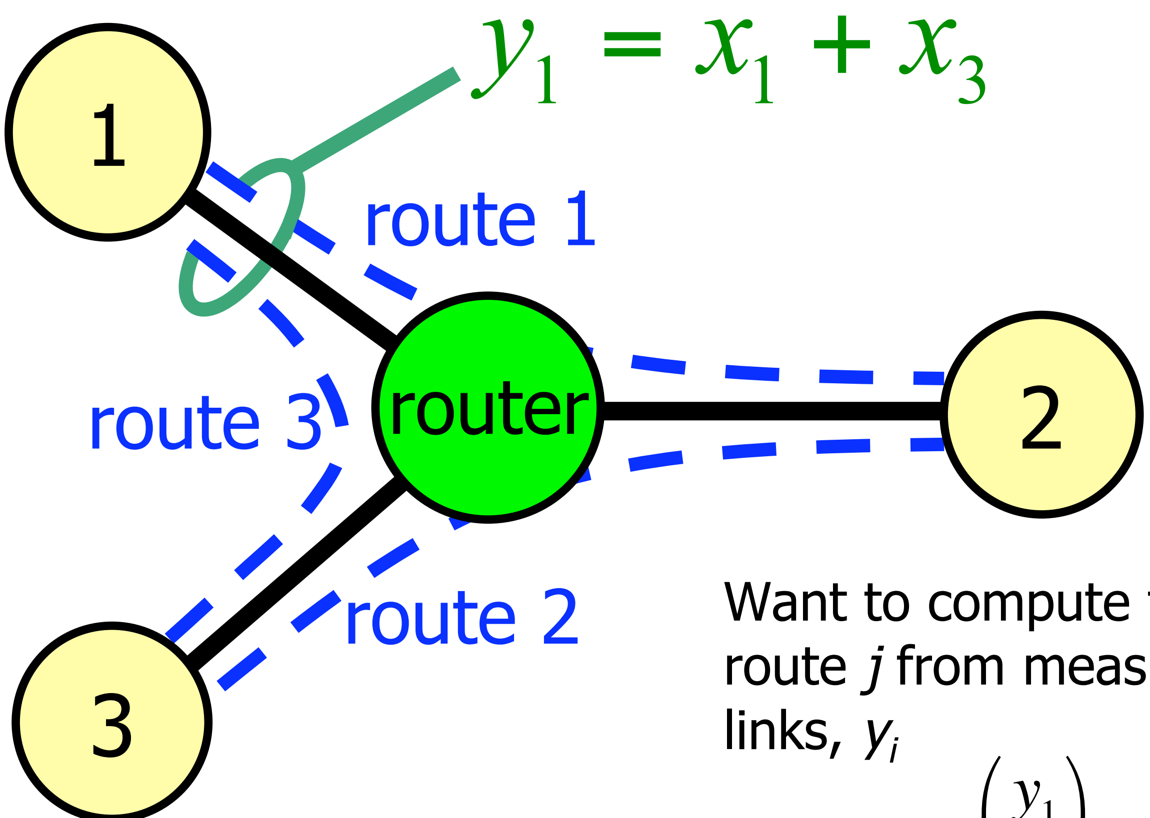
- Pro
  - Comparatively simple
  - Relatively low volume
  - It is used already (lots of historical data)
- Con
  - Data quality – an issue with any data source
    - Ambiguous
    - Missing data
    - Irregular sampling
  - Octets counters only tell you link utilizations
    - Hard to get a traffic matrix
    - Can't tell what type of traffic
    - Can't easily detect DoS, or other unusual events
  - Coarse time scale (>1 minute typically; 5 min in our case)

Topology and configuration

- Router configurations
  - Based on downloaded router configurations, every 24 hours
    - Links/interfaces
    - Location (to and from)
    - Function (peering, customer, backbone, …)
    - OSPF weights and areas
    - BGP configurations
  - Routing
    - Forwarding tables
    - BGP (table dumps and route monitor)
    - OSPF table dumps
- Routing simulations
  - Simulate IGP and BGP to get routing matrices

Part II: Algorithms

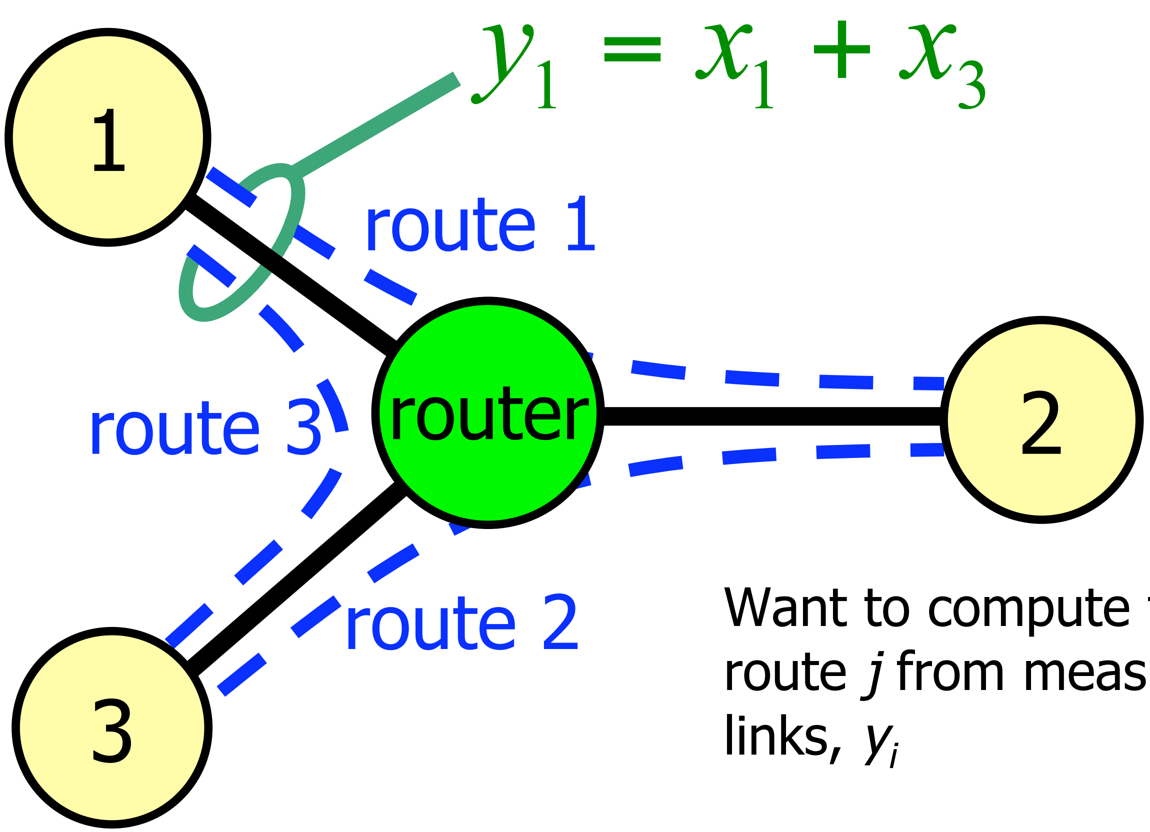
The problem



Want to compute the traffic x_j along route j from measurements on the links, y_i

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

The problem



Want to compute the traffic x_j along route j from measurements on the links, y_i

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

Underconstrained linear inverse problem

$$y = Ax$$

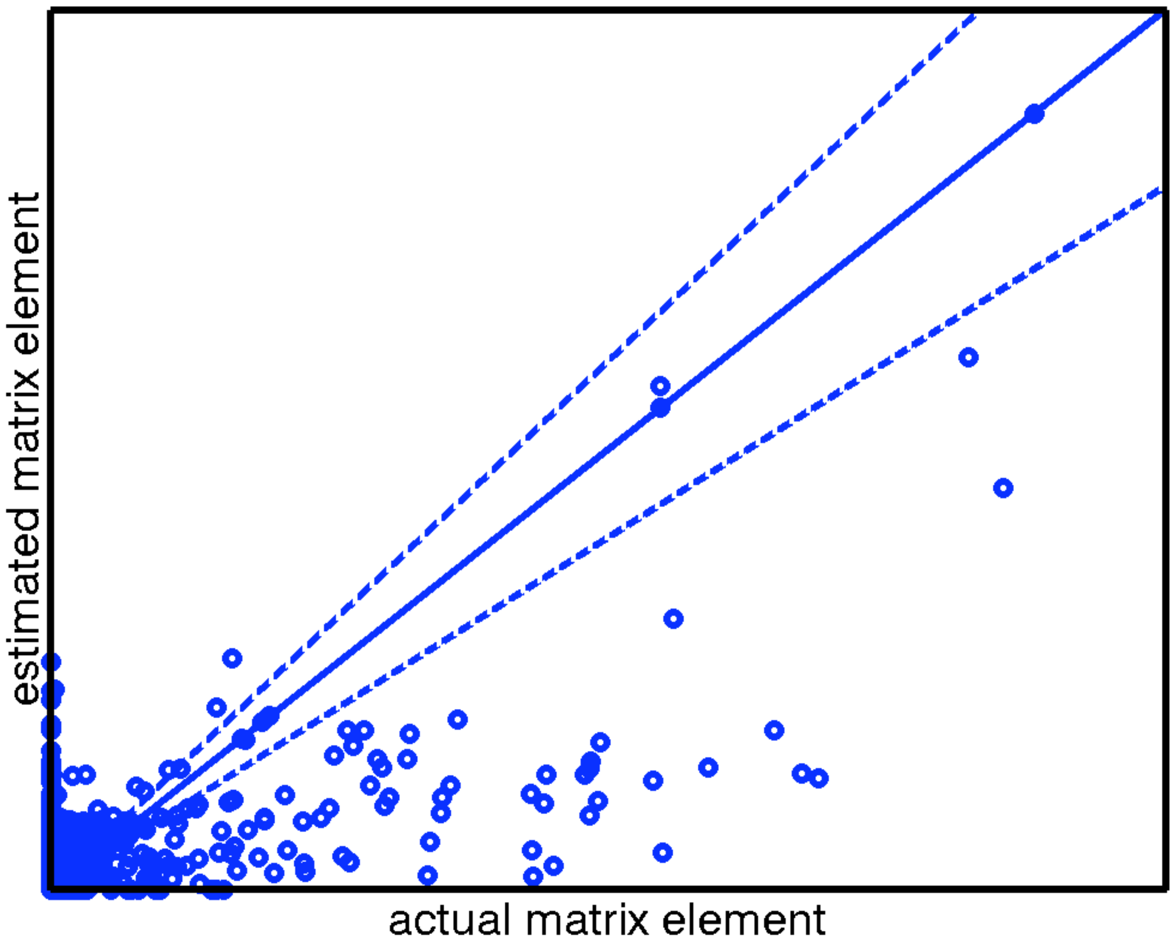
Traffic matrix

Link measurements

Routing matrix

Many more unknowns than measurements

Naive approach



Gravity Model

- Assume traffic between sites is proportional to traffic at each site

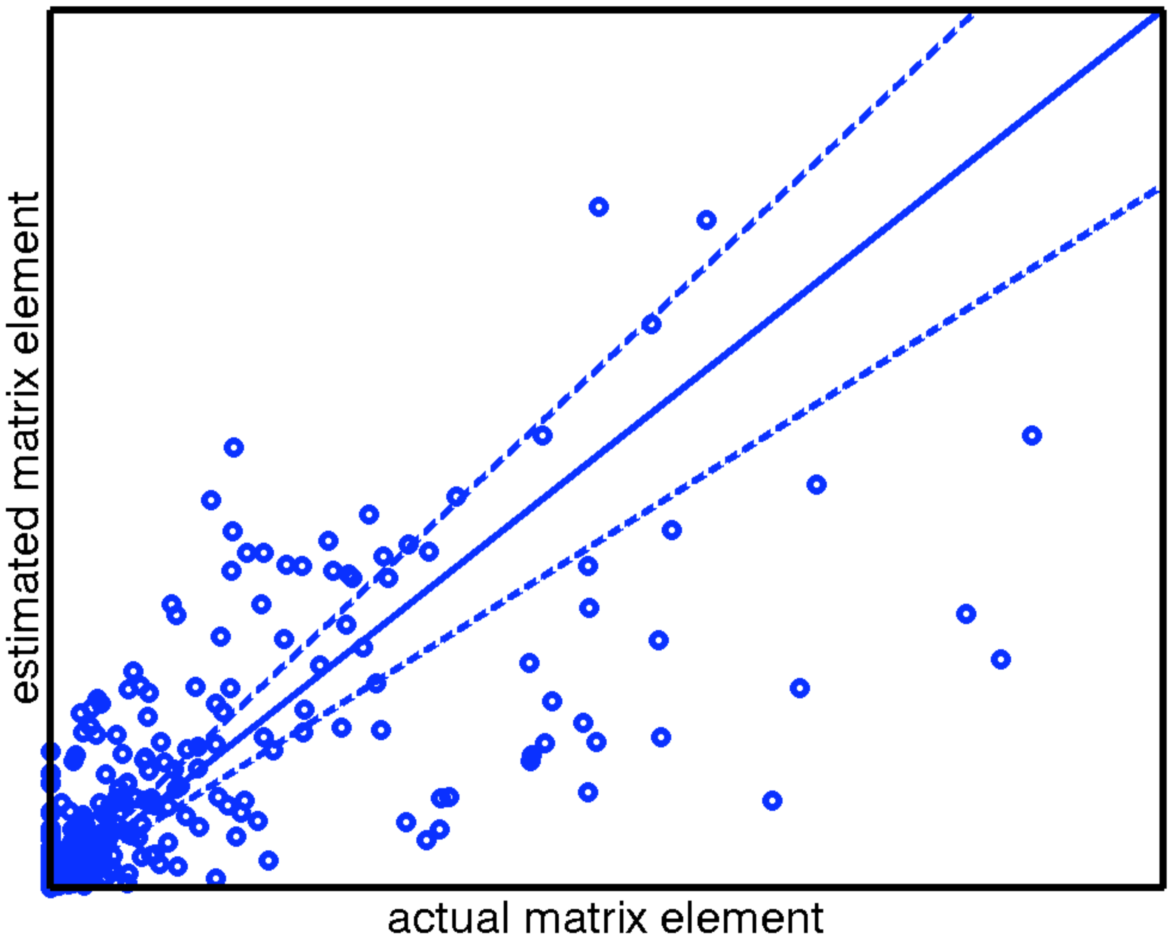
$$x_1 \propto y_1\, y_2$$

$$x_2 \propto y_2\, y_3$$

$$x_3 \propto y_1\, y_3$$

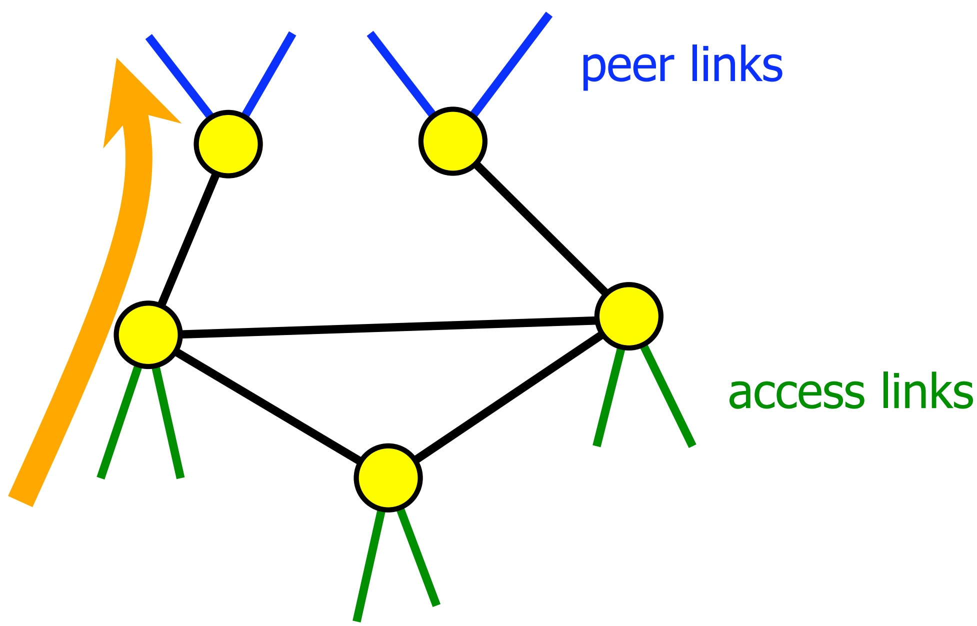
- Assumes there is no systematic difference between traffic in LA and NY
  - Only the total volume matters
  - Could include a distance term, but locality of information is not as important in the Internet as in other networks

Simple gravity model



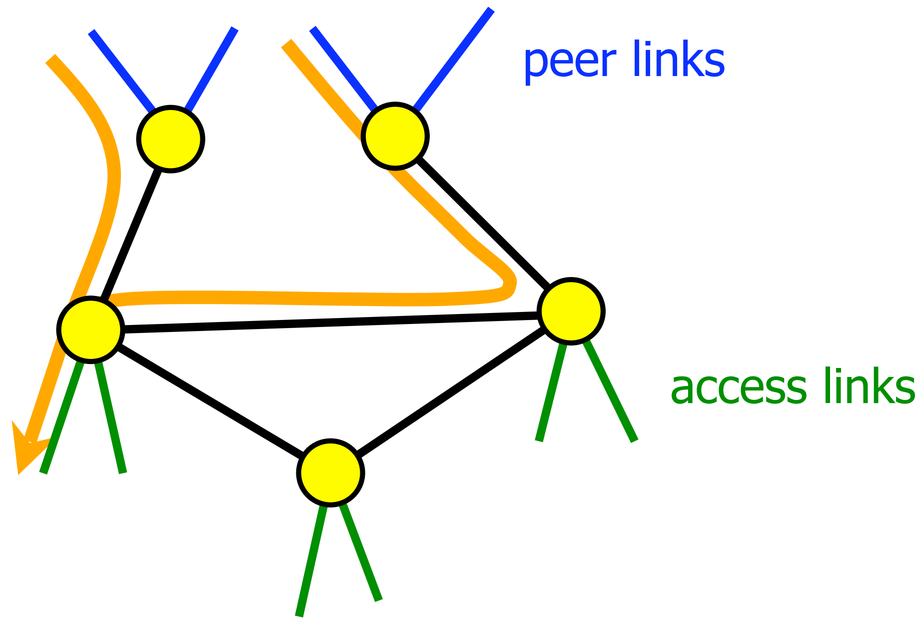
Generalized gravity model

- Internet routing is asymmetric
- A provider can control exit points for traffic going to peer networks

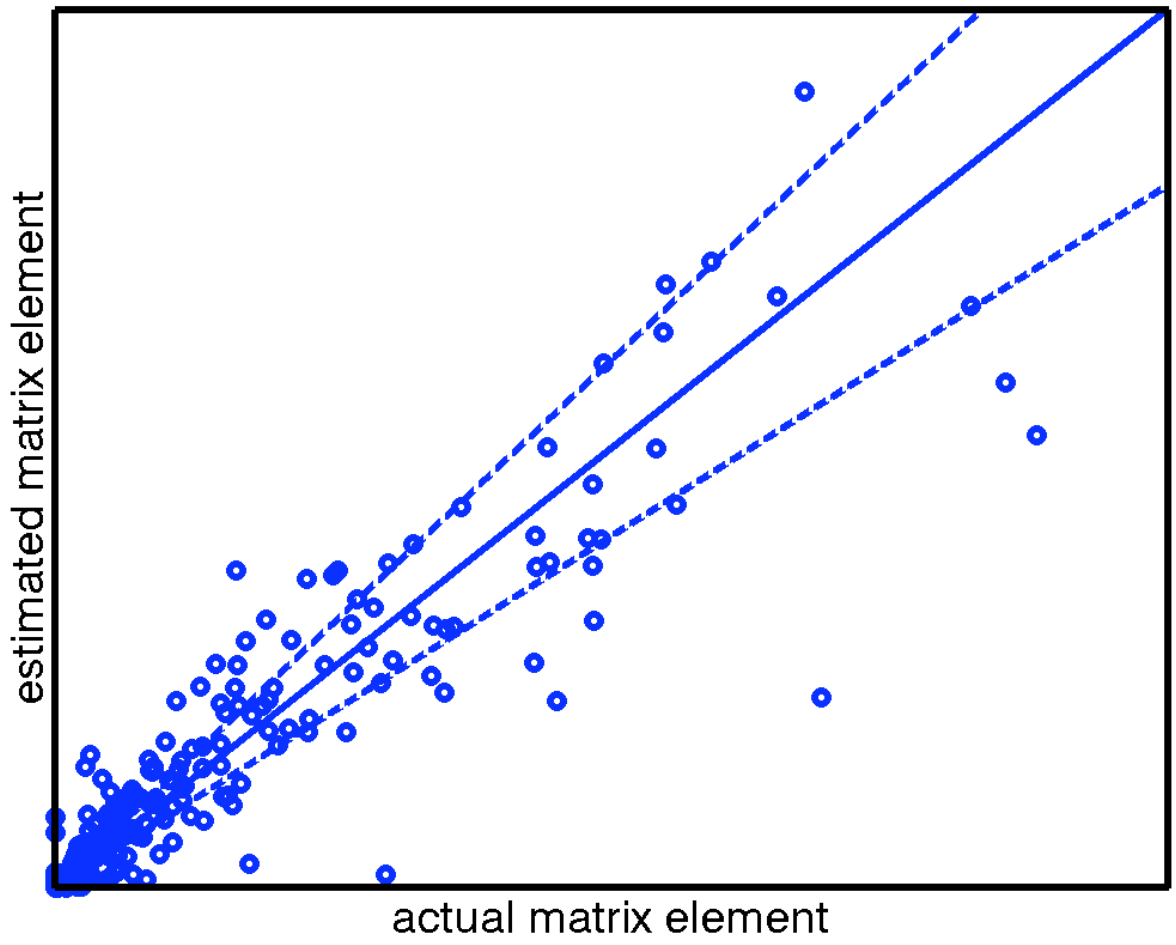


Generalized gravity model

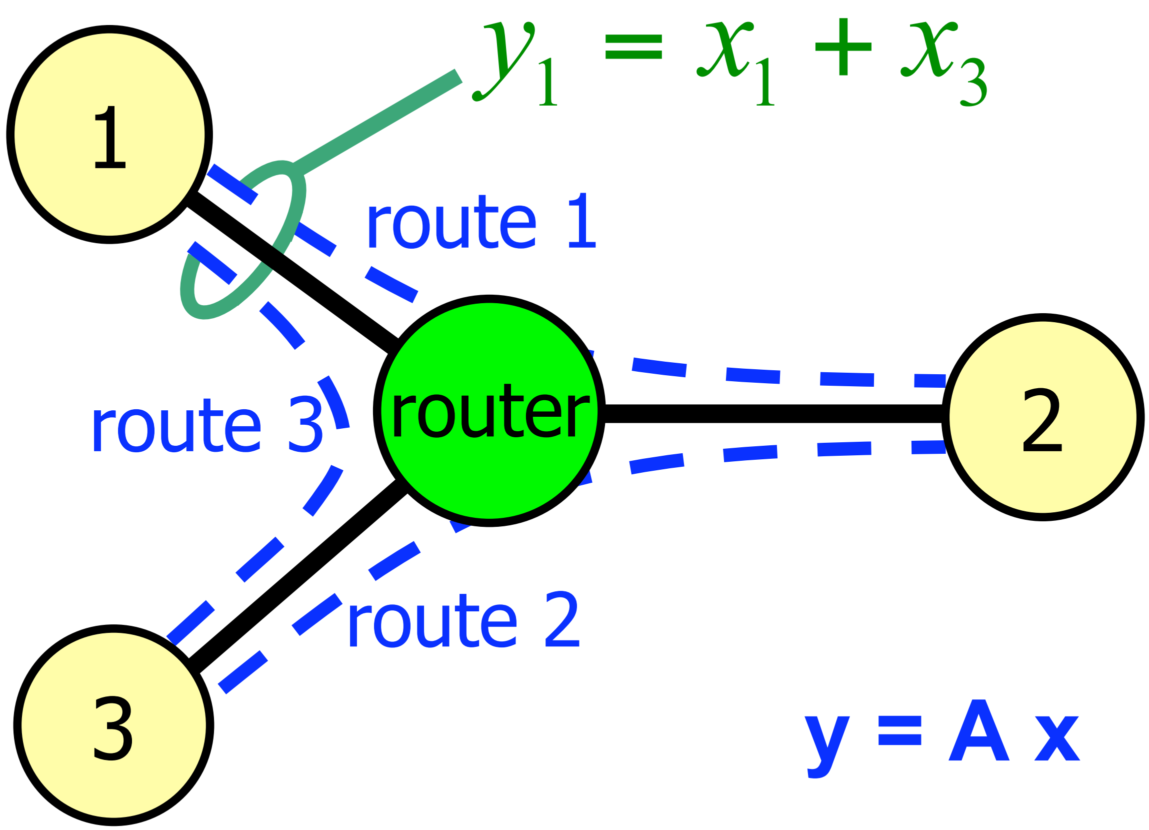
- Internet routing is asymmetric
- A provider can control exit points for traffic going to peer networks
- Have much less control over where traffic enters



Generalized gravity model



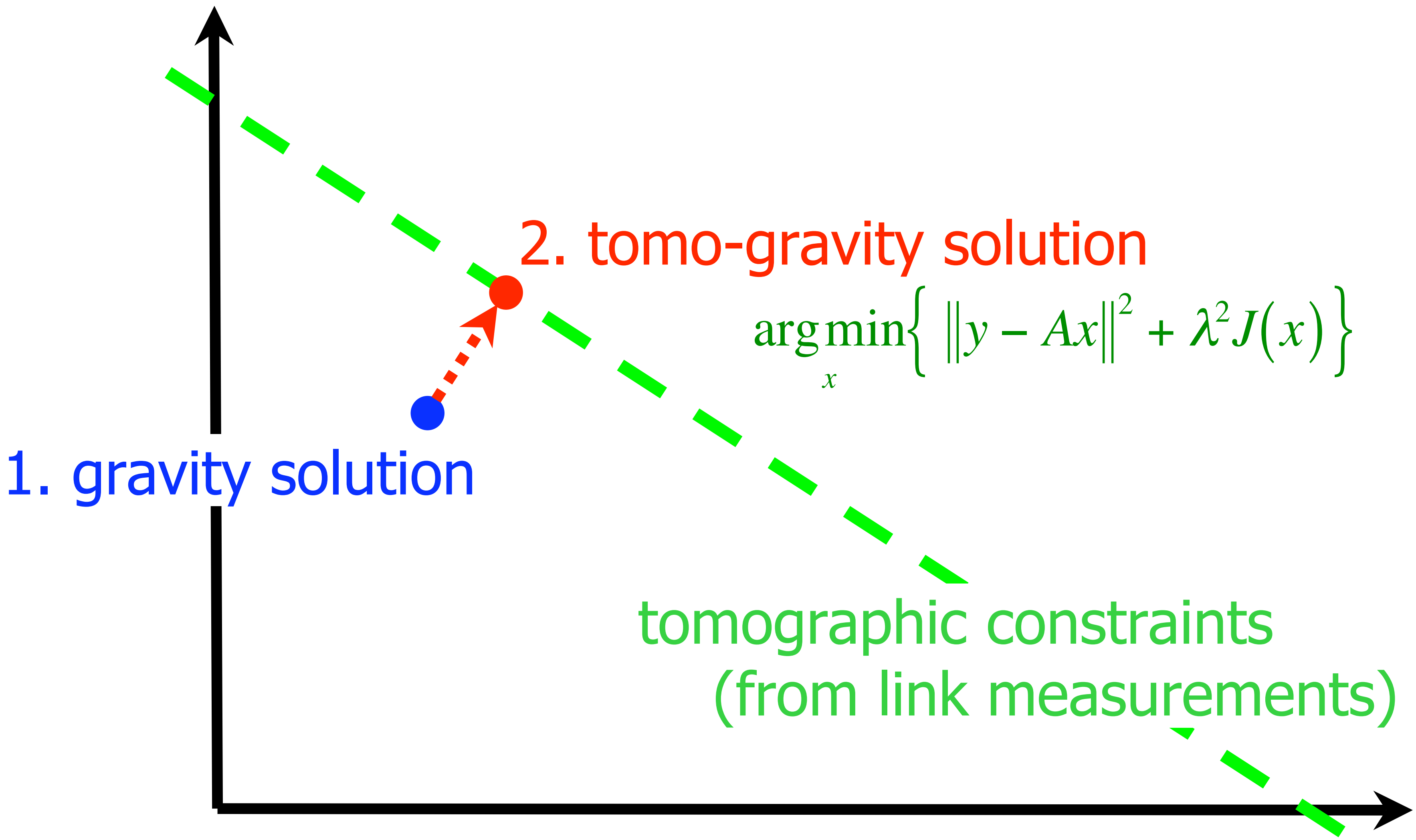
Tomographic approach



Direct Tomographic approach

- Under-constrained problem
- Find additional constraints
- Use a model to do so
  - Typical approach is to use higher order statistics of the traffic to find additional constraints
- Disadvantage
  - Complex algorithm – doesn't scale (~1000 nodes, 10000 routes)
  - Reliance on higher order stats is not robust given the problems in SNMP data
  - Model may not be correct -> result in problems
  - Inconsistency between model and solution

Combining gravity model and tomography

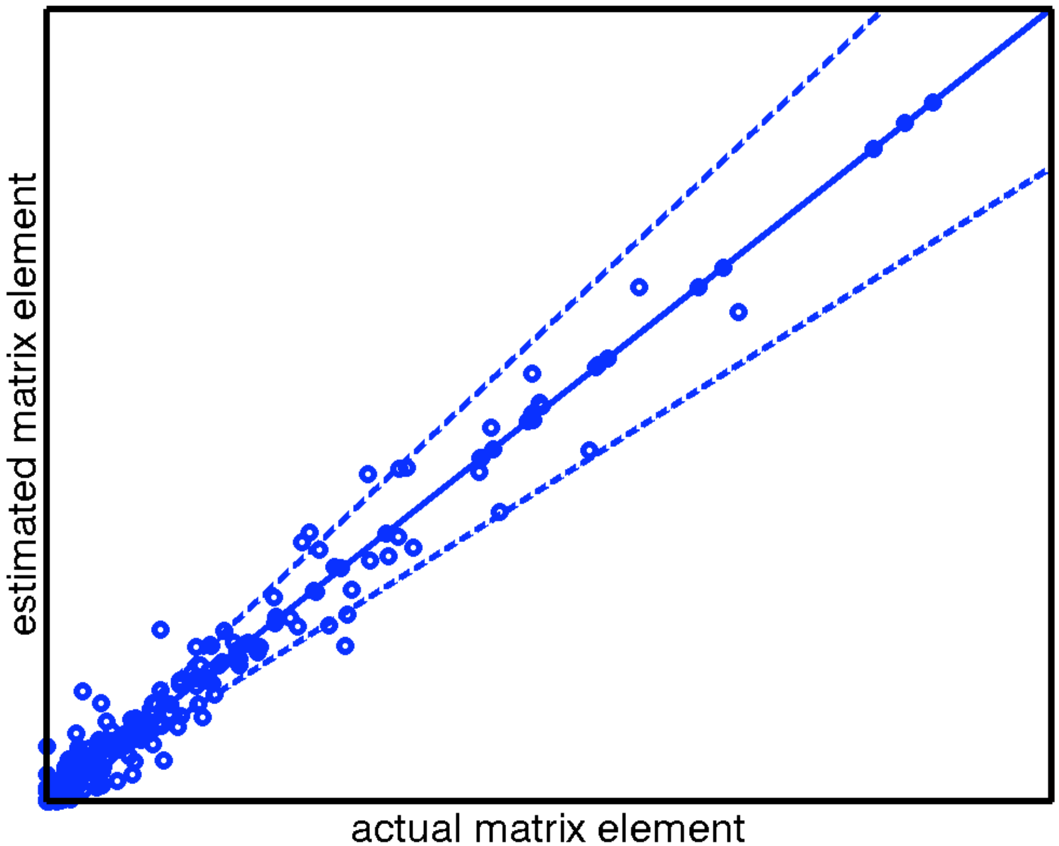


Regularization approach

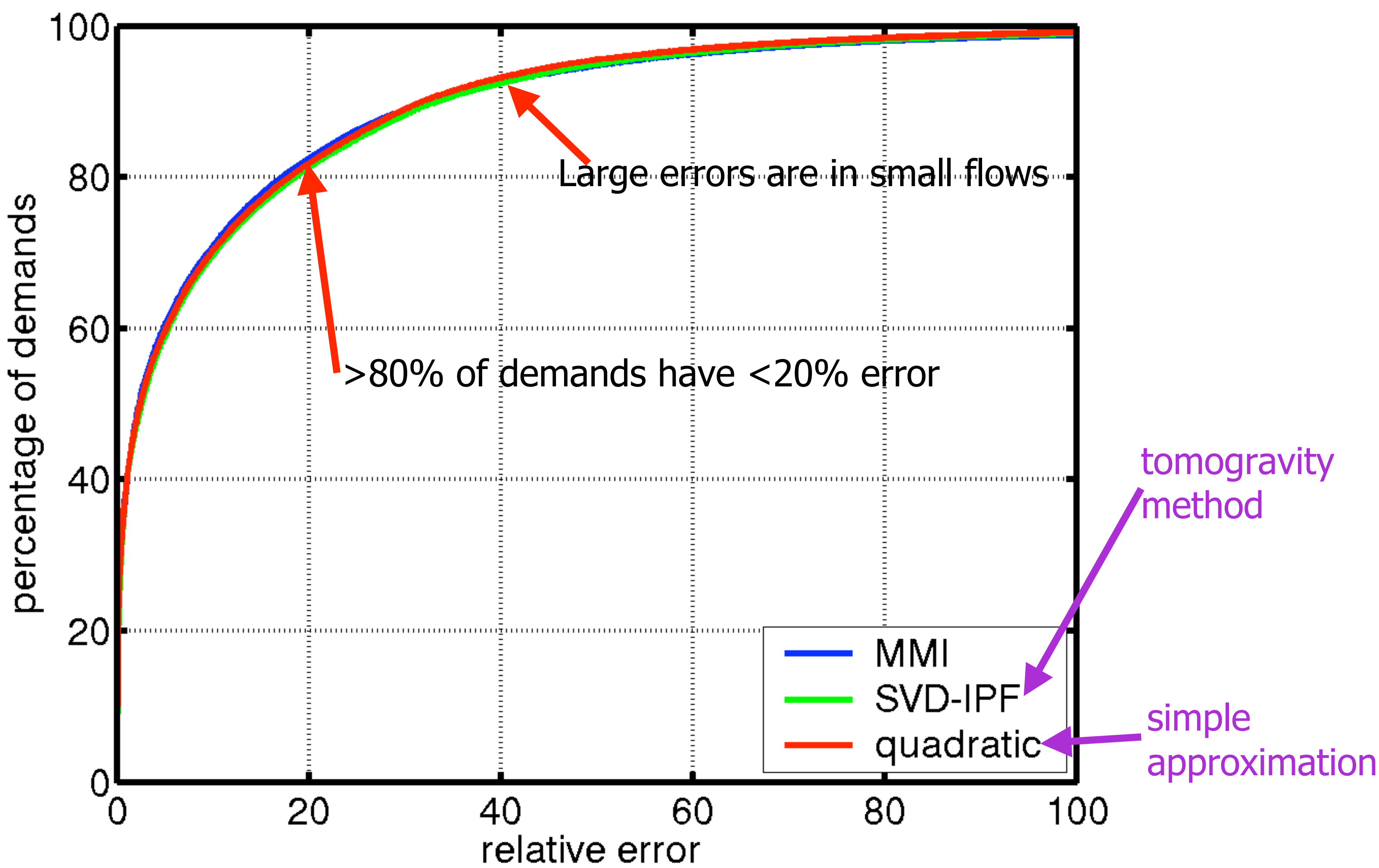
- Minimum Mutual Information:
  - minimize the mutual information between source and destination
- No information
  - The minimum is independence of source and destination
    - P(S,D) = p(S) p(D)
    - P(D|S) = P(D)
    - actually this corresponds to the gravity model
  - Add tomographic constraints:
    - Including additional information as constraints
    - Natural algorithm is one that minimizes the Kullback-Liebler information number of the P(S,D) with respect to P(S) P(D)
      - Max relative entropy (relative to independence)

Validation

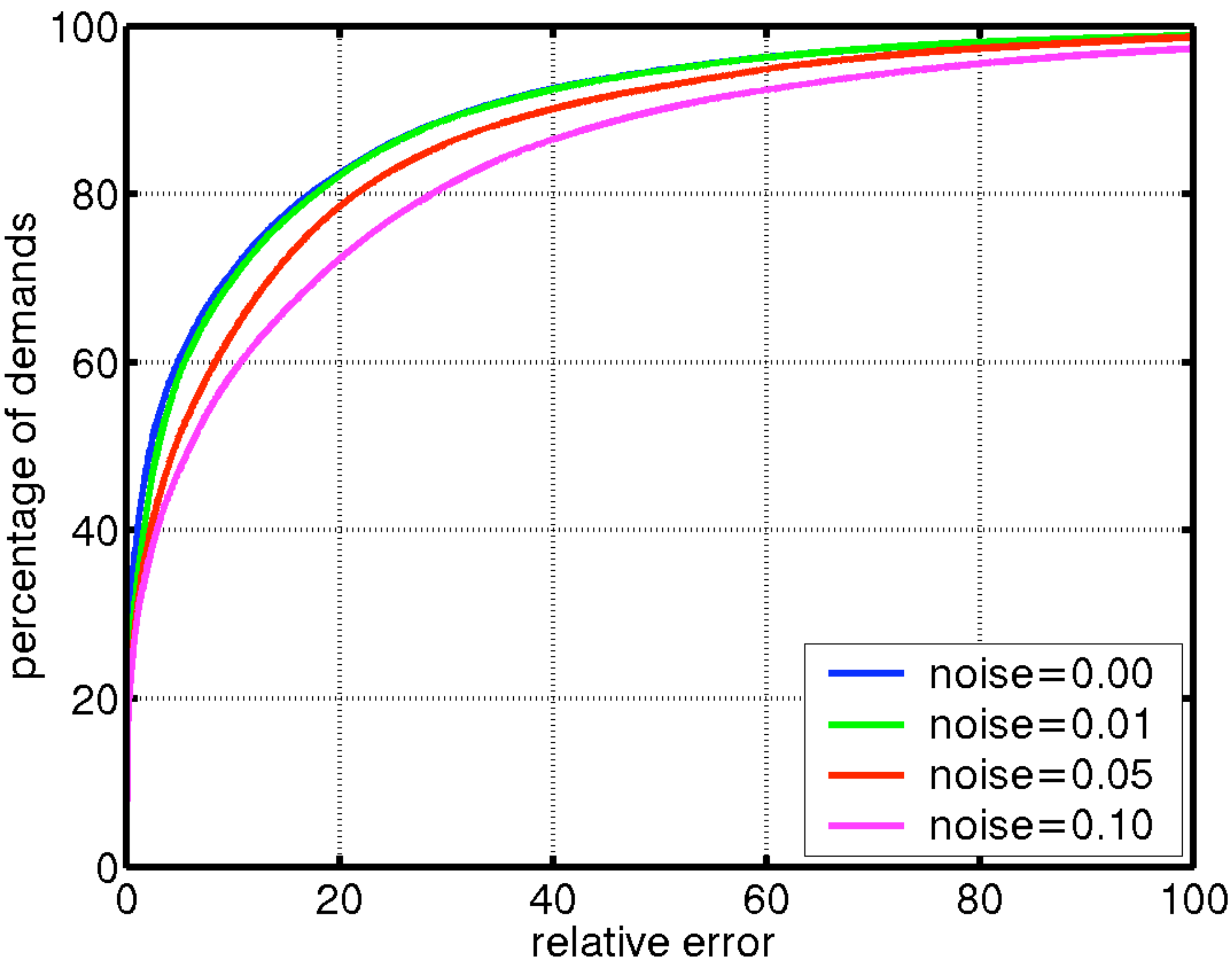
- Results good: ±20% bounds for larger flows
- Observables even better



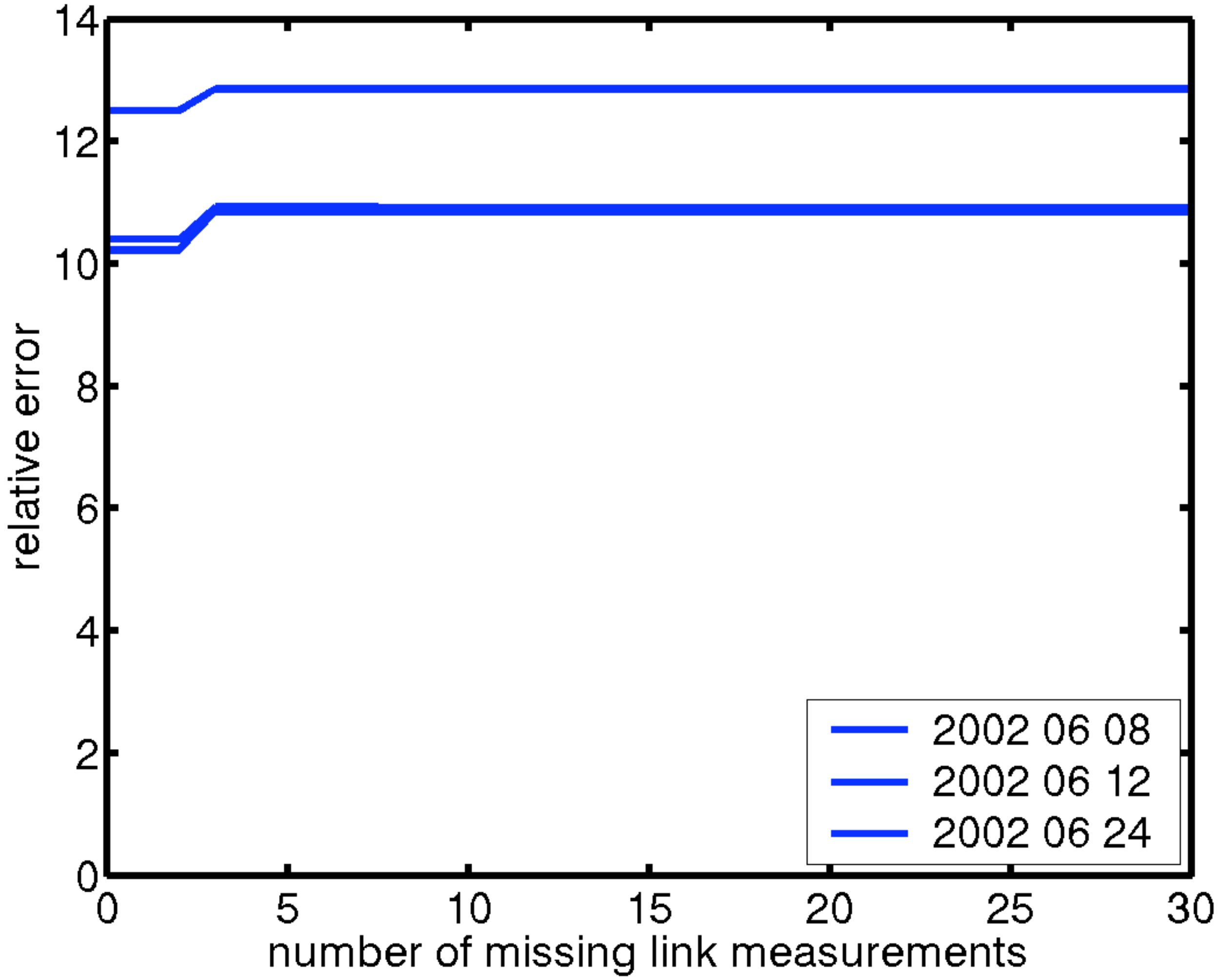
More results



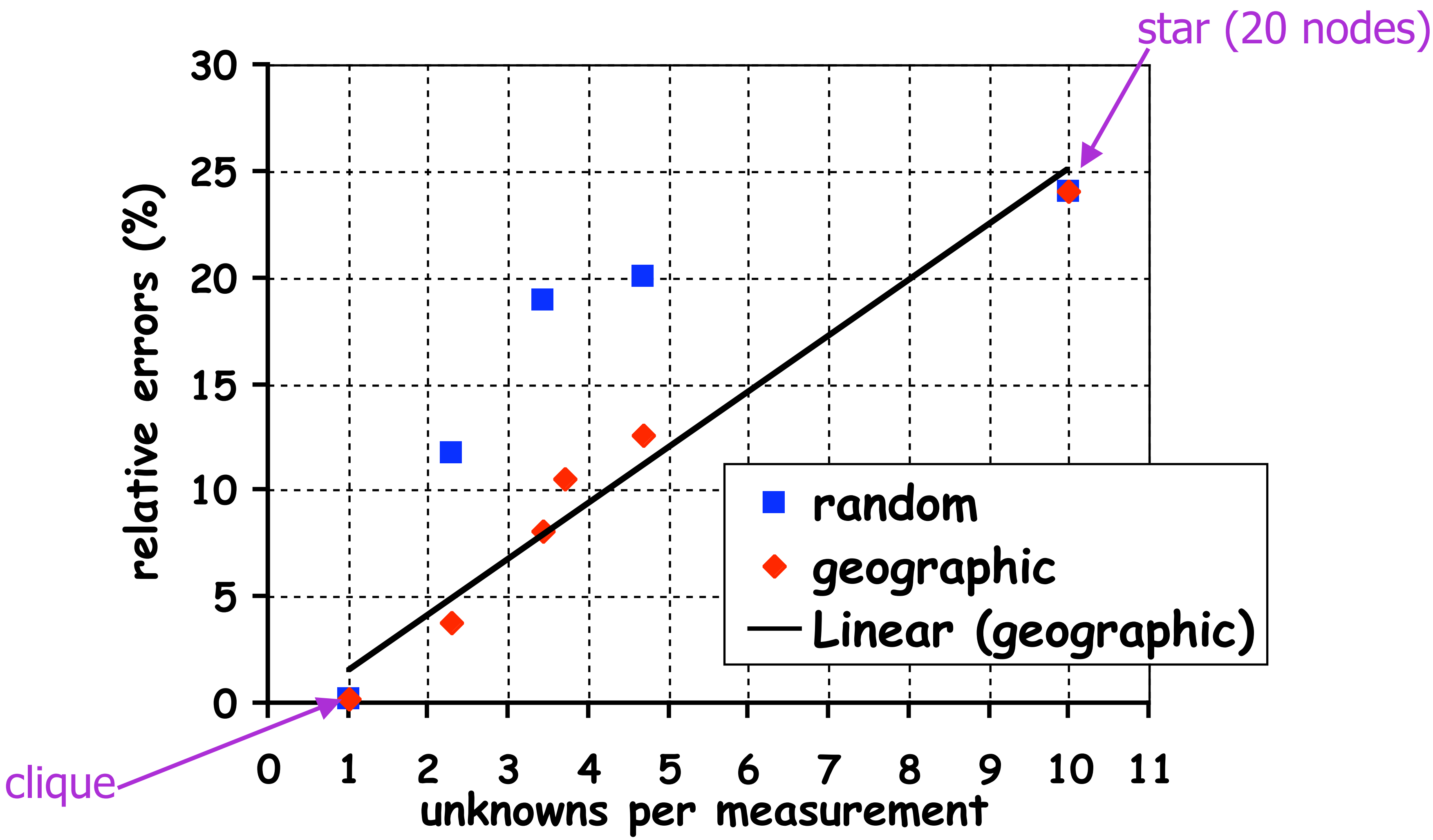
Robustness (input errors)



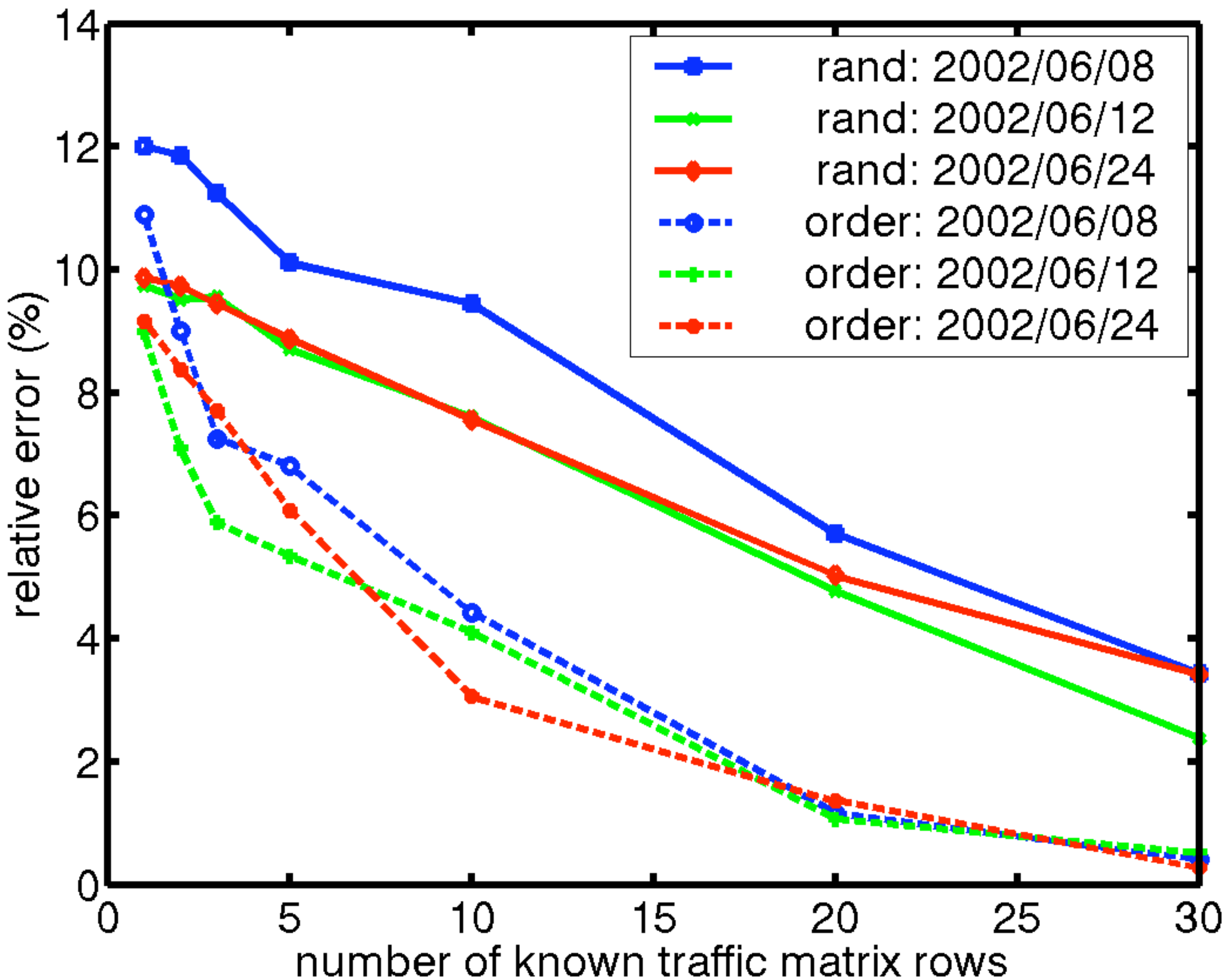
Robustness (missing data)



Dependence on Topology



Additional information - Netflow



Part III: Applications

Applications

- Capacity planning
  - Optimize network capacities to carry traffic given routing
  - Timescale – months
- Reliability Analysis
  - Test network has enough redundant capacity for failures
  - Time scale – days
- Traffic engineering
  - Optimize routing to carry given traffic
  - Time scale – *potentially* minutes

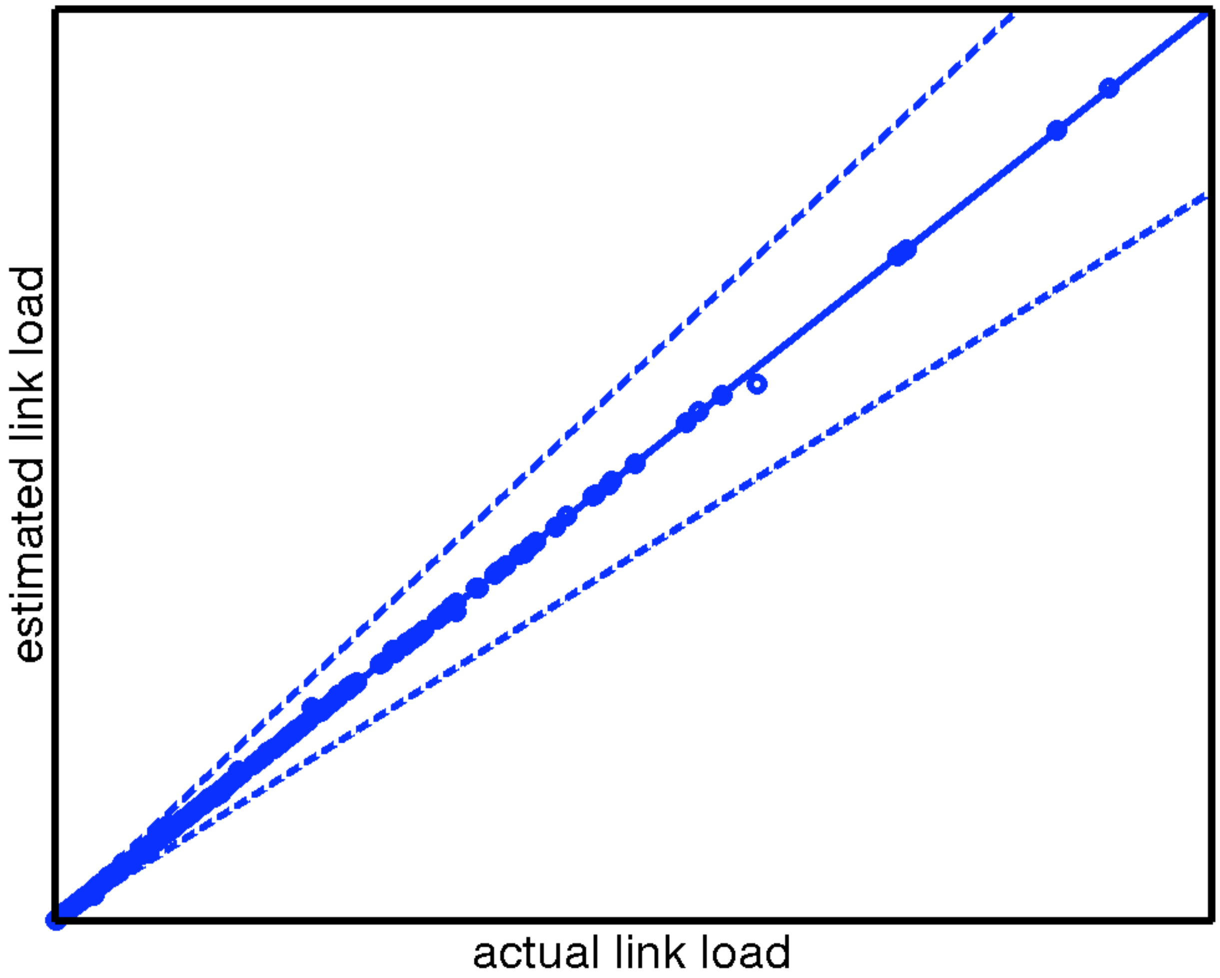
Capacity planning

- Plan network capacities
  - No sophisticated queueing (yet)
  - Optimization problem
- Used in AT&T backbone capacity planning
  - For more than well over a year
  - North American backbone
- Being extended to other networks

Network Reliability Analysis

- Consider the link loads in the network under failure scenarios
  - Traffic will be rerouted
  - What are the new link loads?
- Prototype used (> 1 year)
  - Currently being turned form a prototype into a production tool for the IP backbone
  - Allows "what if" type questions to be asked about link failures (and span, or router failures)
  - Allows comprehensive analysis of network risks
    - What is the link most under threat of overload under likely failure scenarios

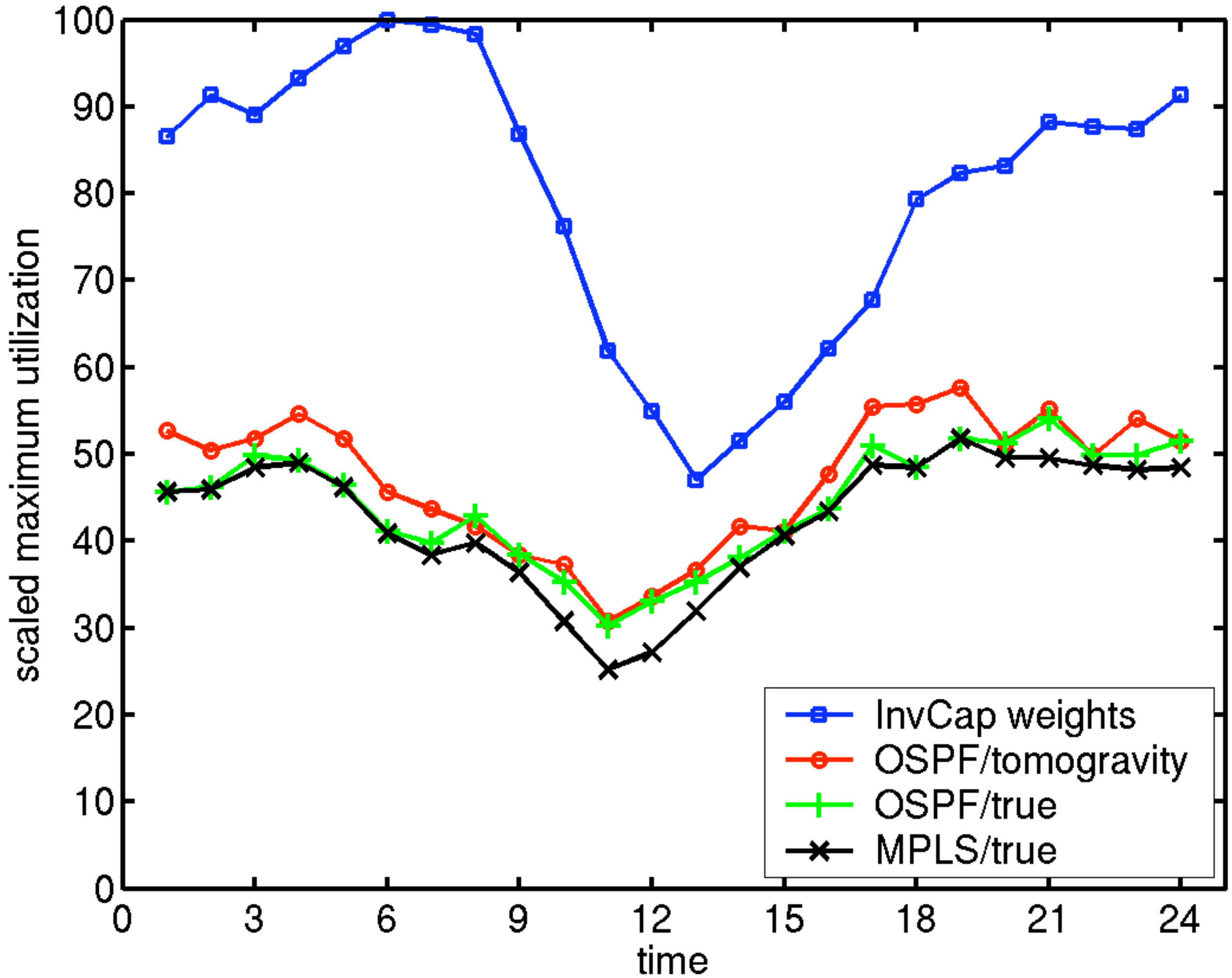
Example use: reliability analysis



Traffic engineering and routing optimization

- Choosing route parameters that use the network most efficiently
  - In simple cases, load balancing across parallel routes
- Methods
  - Shortest path IGP weight optimization
    - Thorup and Fortz showed could optimize OSPF weights
  - Multi-commodity flow optimization
    - Implementation using MPLS
    - Explicit route for each origin/destination pair

Comparison of route optimizations



Conclusion

- Properties
  - Fast (a few seconds for 50 nodes)
  - Scales (to hundreds of nodes)
  - Robust (to errors and missing data)
  - Average errors ~11%, bounds 20% for large flows
- Tomo-gravity implemented
  - AT&T's IP backbone (AS 7018)
  - Hourly traffic matrices for > 1 year
  - Being extended to other networks

http://www.maths.adelaide.edu.au/staff/applied/~roughan/

Additional slides

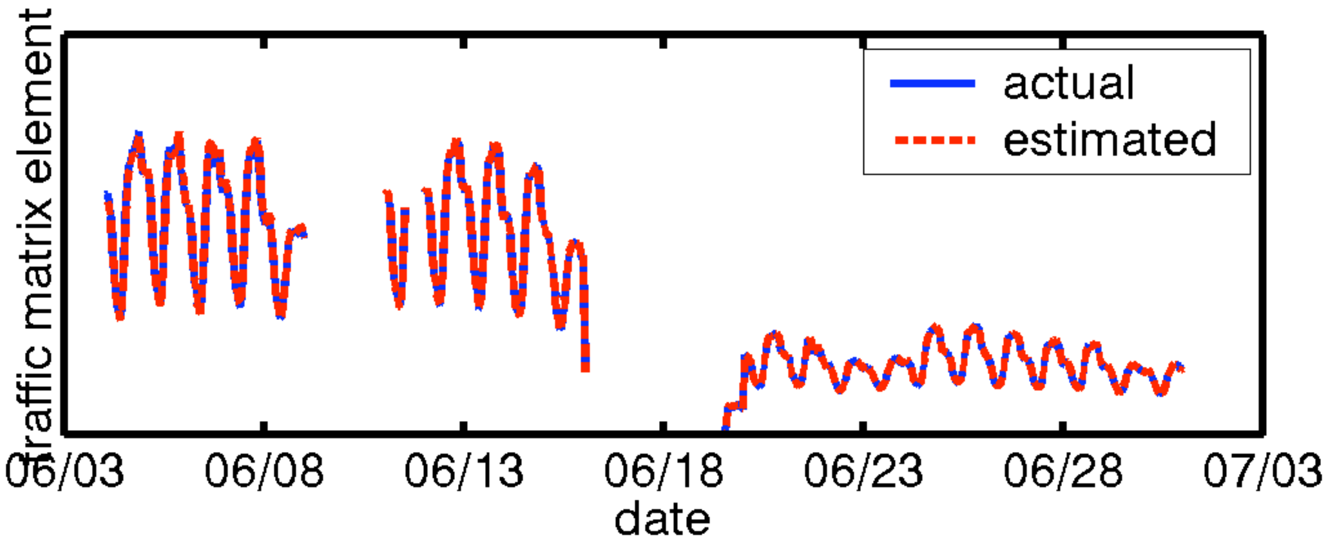
Validation

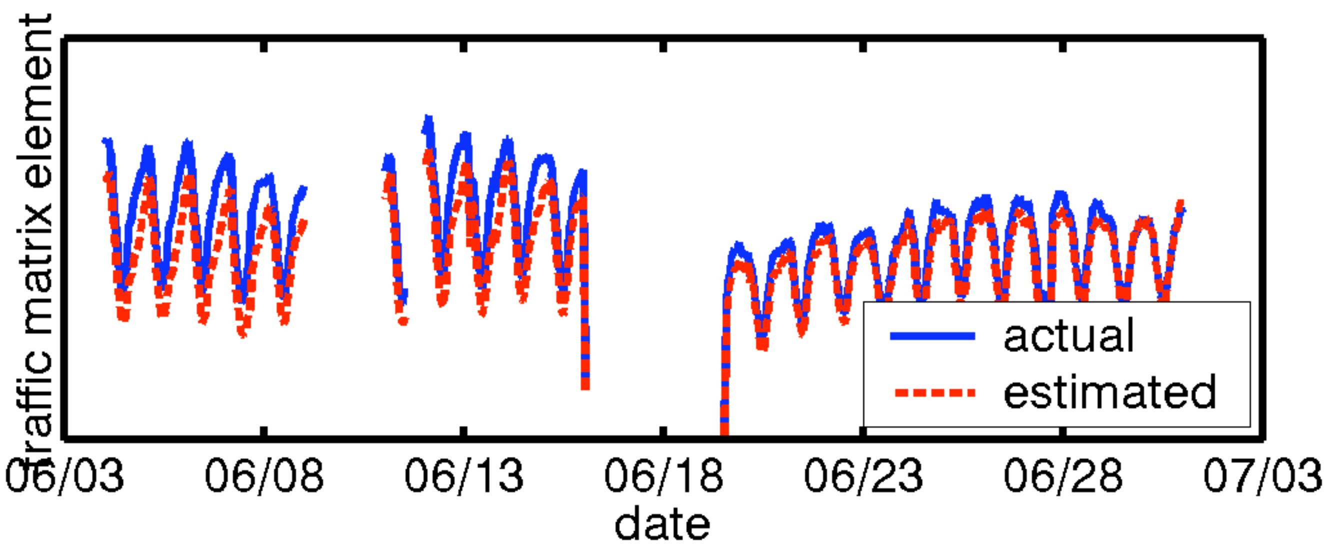
- Look at a real network
  - Get SNMP from links
  - Get Netflow to generate a traffic matrix
  - Compare algorithm results with “ground truth”
  - Problems:
    - Hard to get Netflow along whole edge of network
      - If we had this, then we wouldn’t need SNMP approach
    - Actually pretty hard to match up data
      - Is the problem in your data: SNMP, Netflow, routing, …
- Simulation
  - Simulate and compare
  - Problems
    - How to generate realistic traffic matrices
    - How to generate realistic network
    - How to generate realistic routing
    - Danger of generating exactly what you put in

Our method

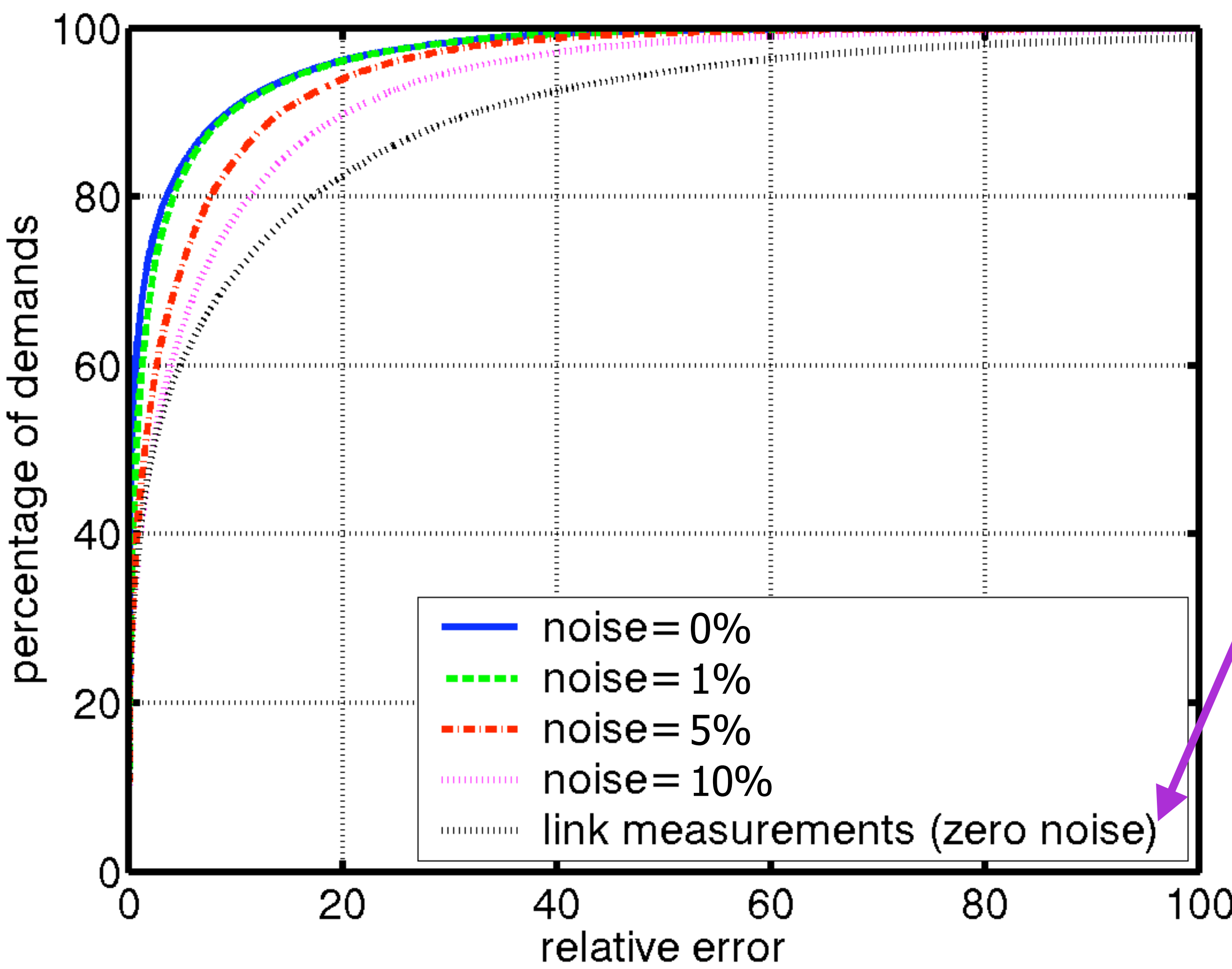
- We have netflow around part of the edge (currently)
- We can generate a partial traffic matrix (hourly)
  - Won't match traffic measured from SNMP on links
- Can use the routing and partial traffic matrix to simulate the SNMP measurements you would get
- Then solve inverse problem
- Advantage
  - Realistic network, routing, and traffic
  - Comparison is direct, we know errors are due to algorithm not errors in the data

Estimates over time





Local traffic matrix (George Varghese)



for reference
previous case