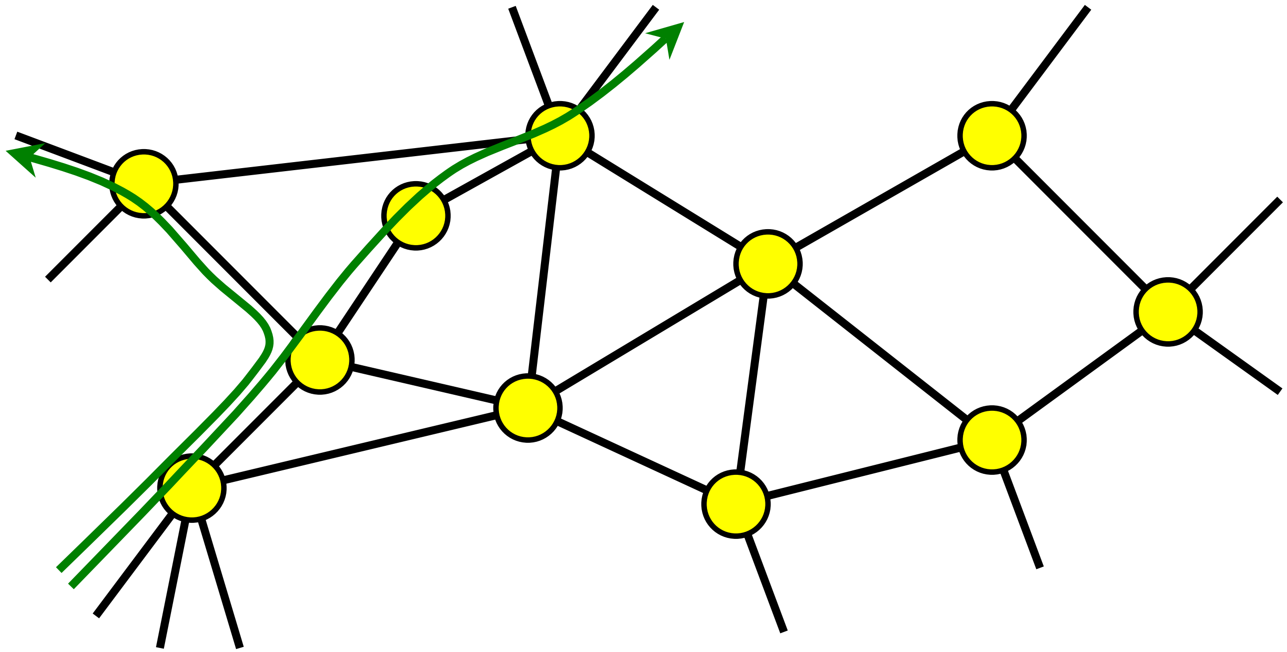
# Information, Gravity, and Traffic Matrices

Yin Zhang, Matthew Roughan, Albert Greenberg, Nick Duffield, David Donoho

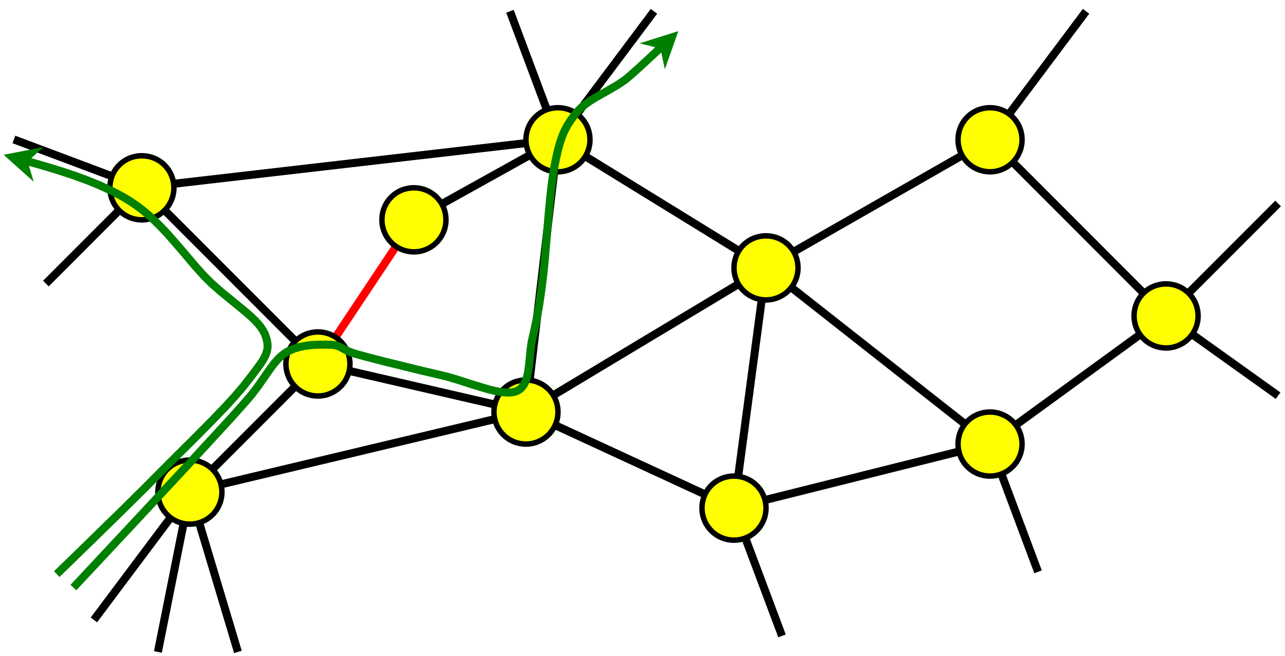
# Problem

Have link traffic measurements
Want to know demands from source to destination



# Example App: reliability analysis

Under a link failure, routes change
want to find an invariant

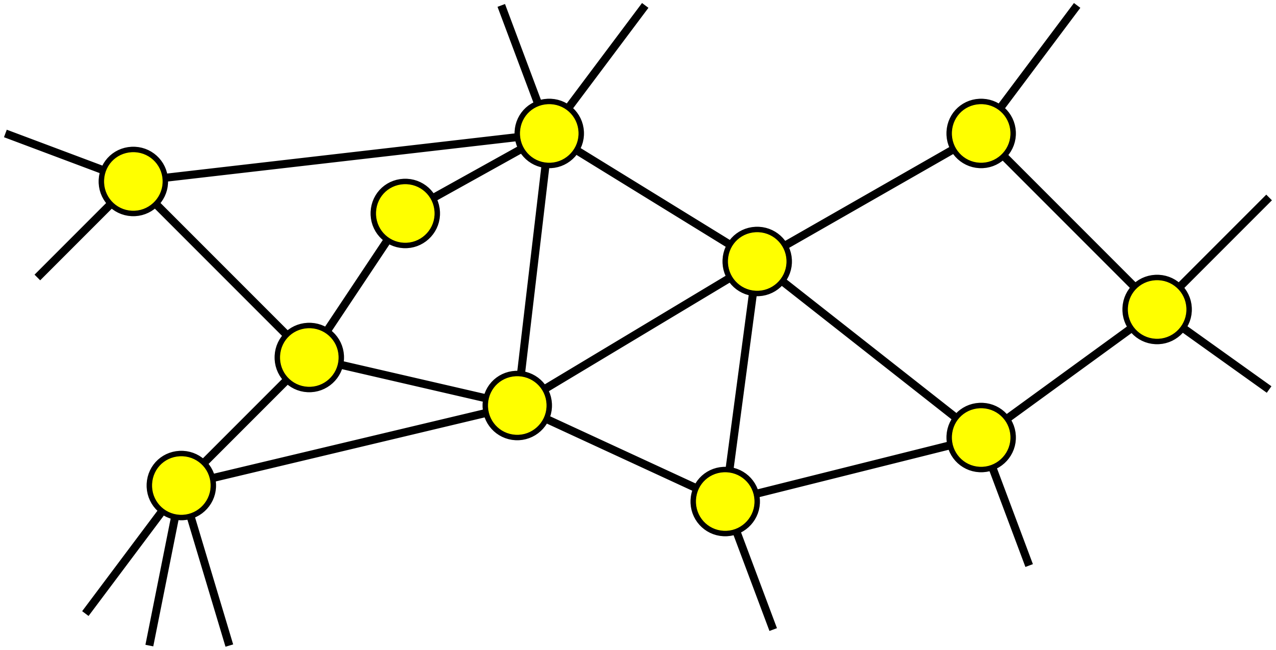


# Outline

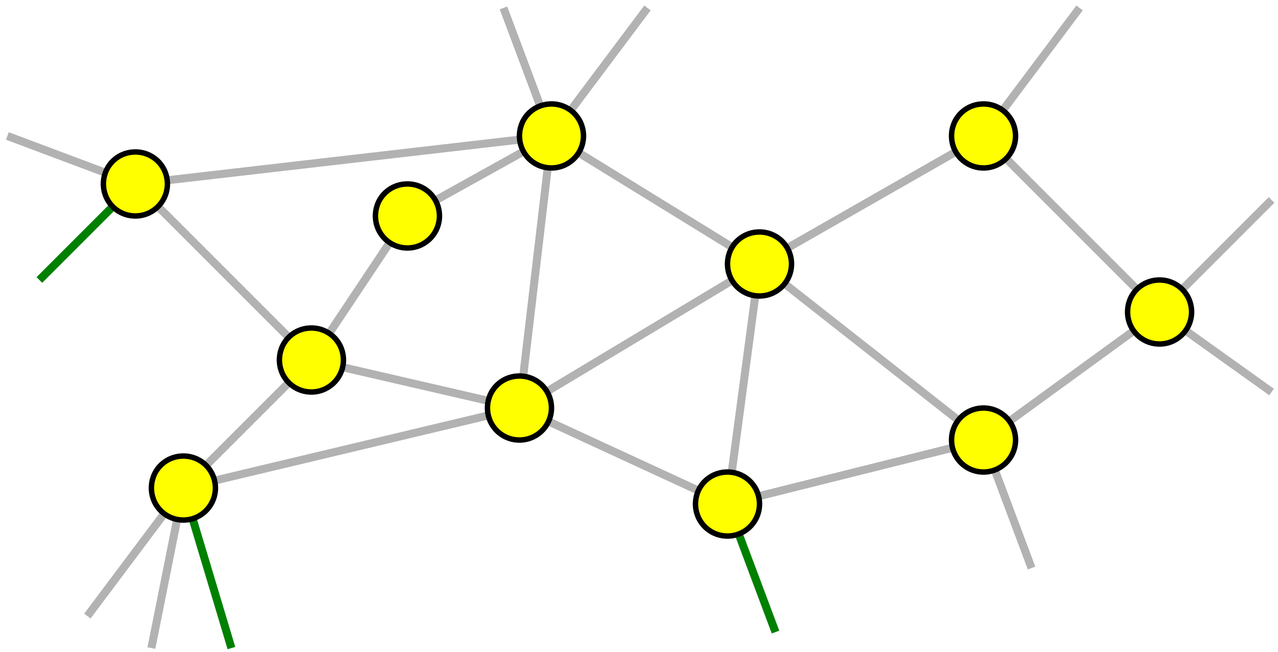
- Part I: What do we have to work with – data sources
  - SNMP traffic data
  - Netflow, packet traces
  - Topology, routing and configuration
- Part II:Algorithms
  - Gravity models
  - Tomography
  - Combination and information theor
- Part III: Applications
  - Network Reliability analysis
  - Capacity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

# Part I: Data Sources

# Traffic Data



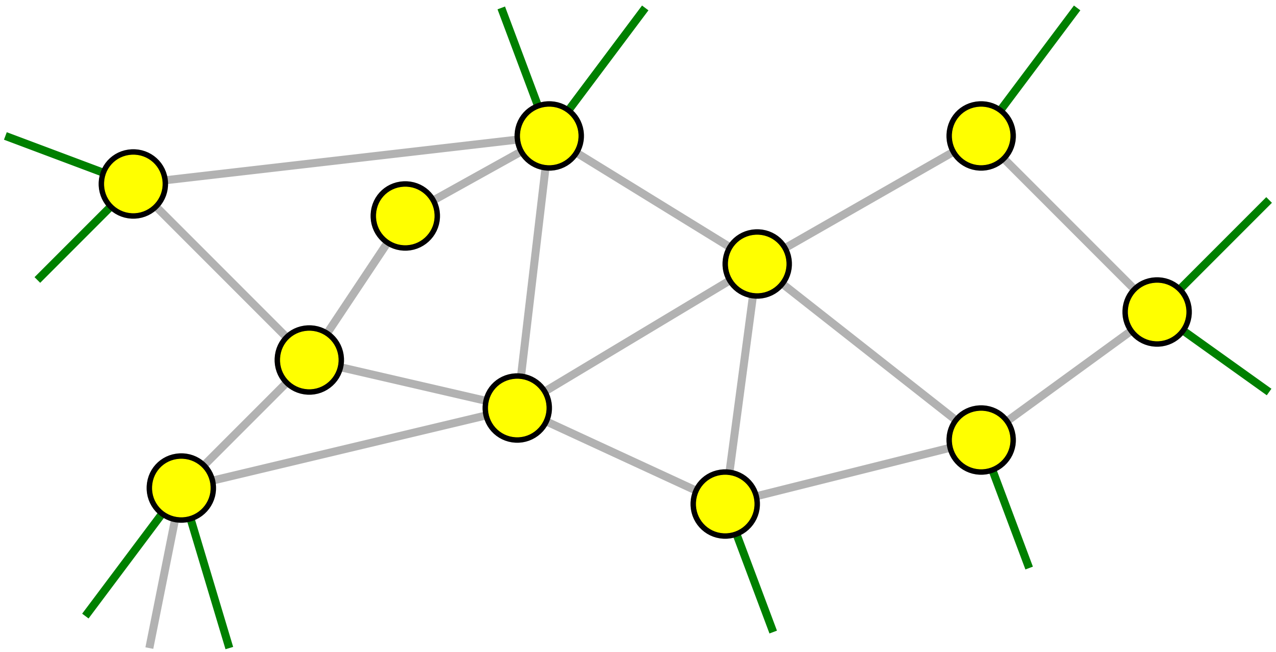
# Data Availability - packet traces



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 OC48 available, only just OC192)
- huge amount of data generated

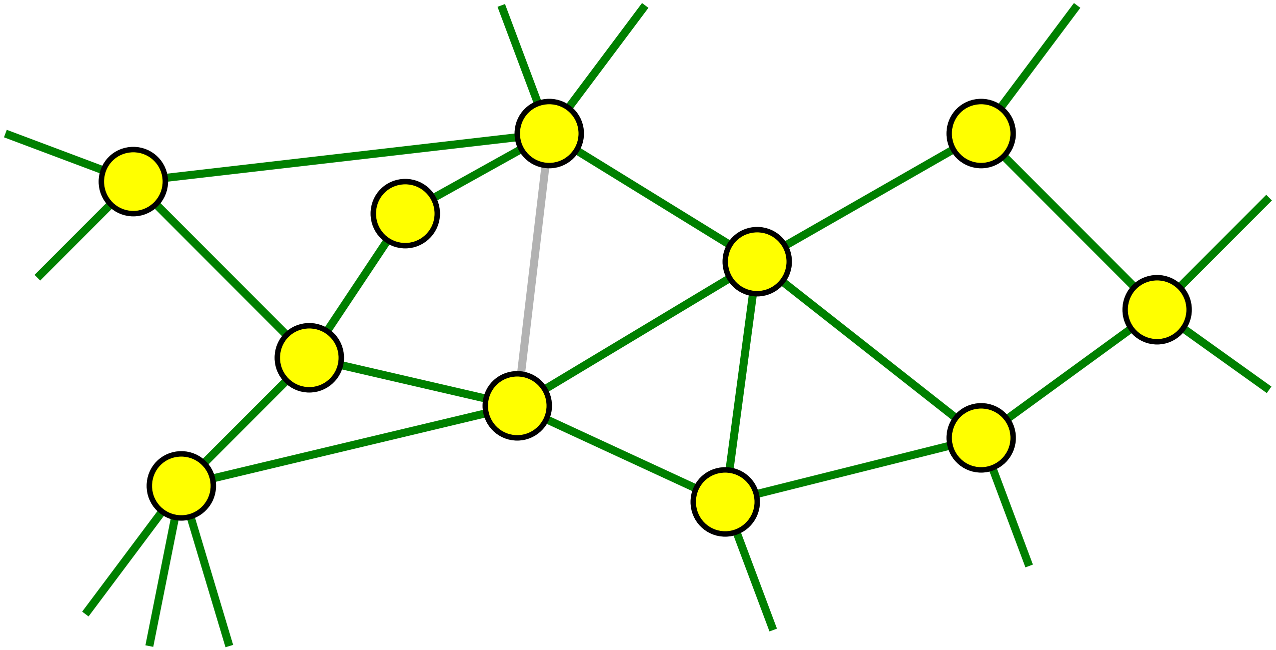
# Data Availability - flow level data



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

# Data Availability - SNMP

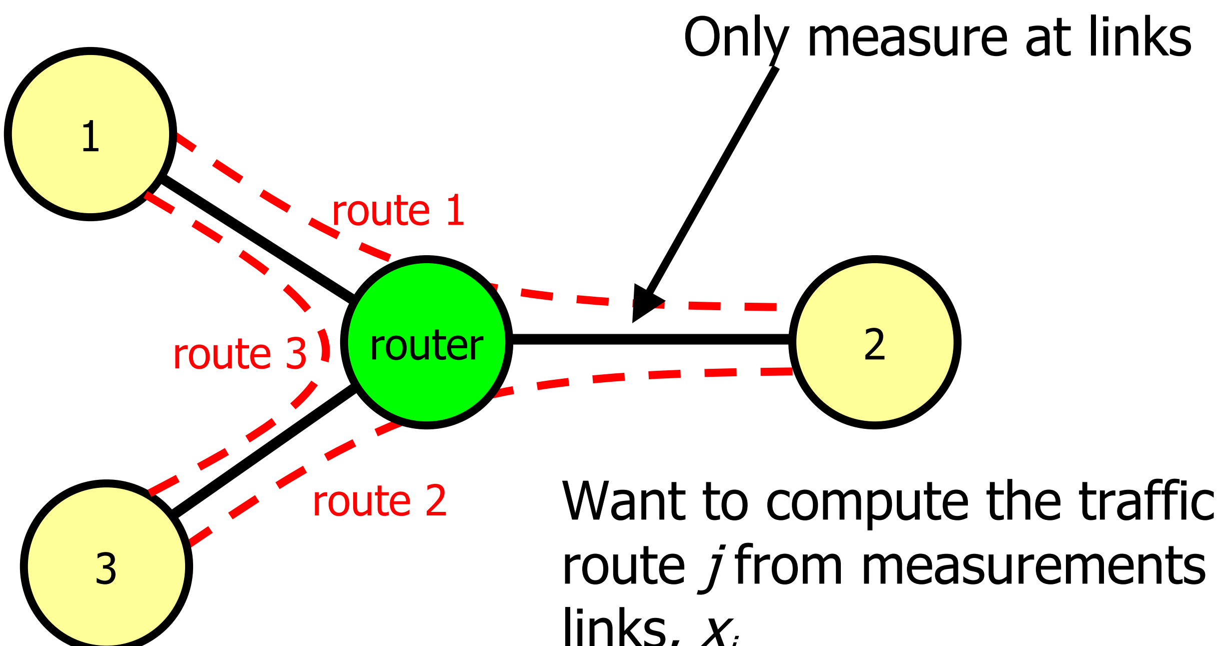


SNMP traffic data – like a time lapse panorama

- MIB II (including IfInOctets/IfOutOctets) is available almost everywhere
- manageable volume of data (but poor quality)
- no significant impact on router performance

# Part II: Algorithms

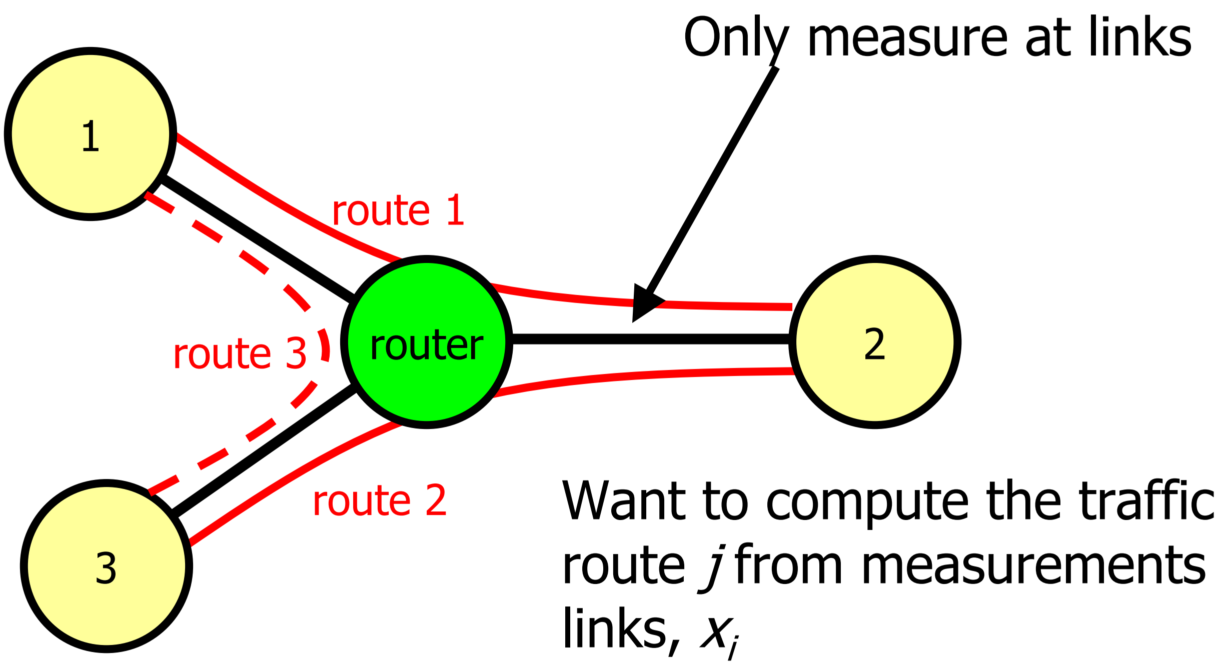
# The problem




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

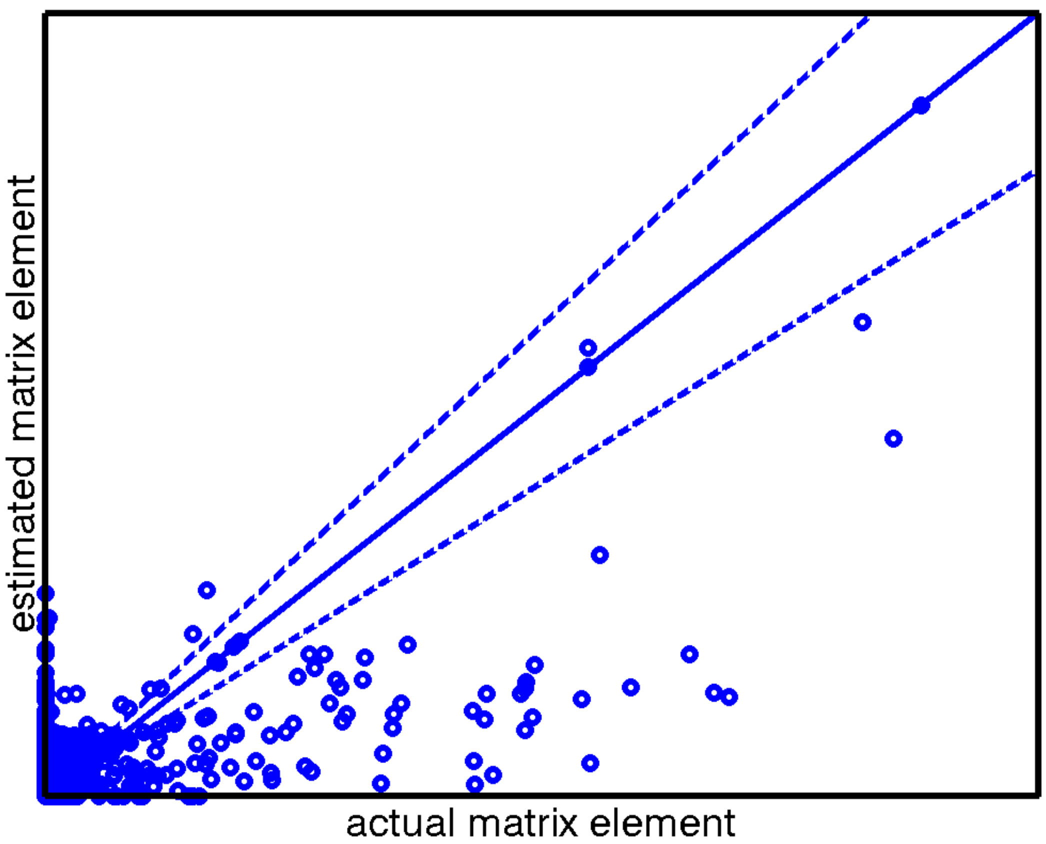



Want to compute the traffic $t_j$ along route $j$ from measurements on the links, $x_i$

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

# Naive approach

In real networks the problem is highly under-constrained



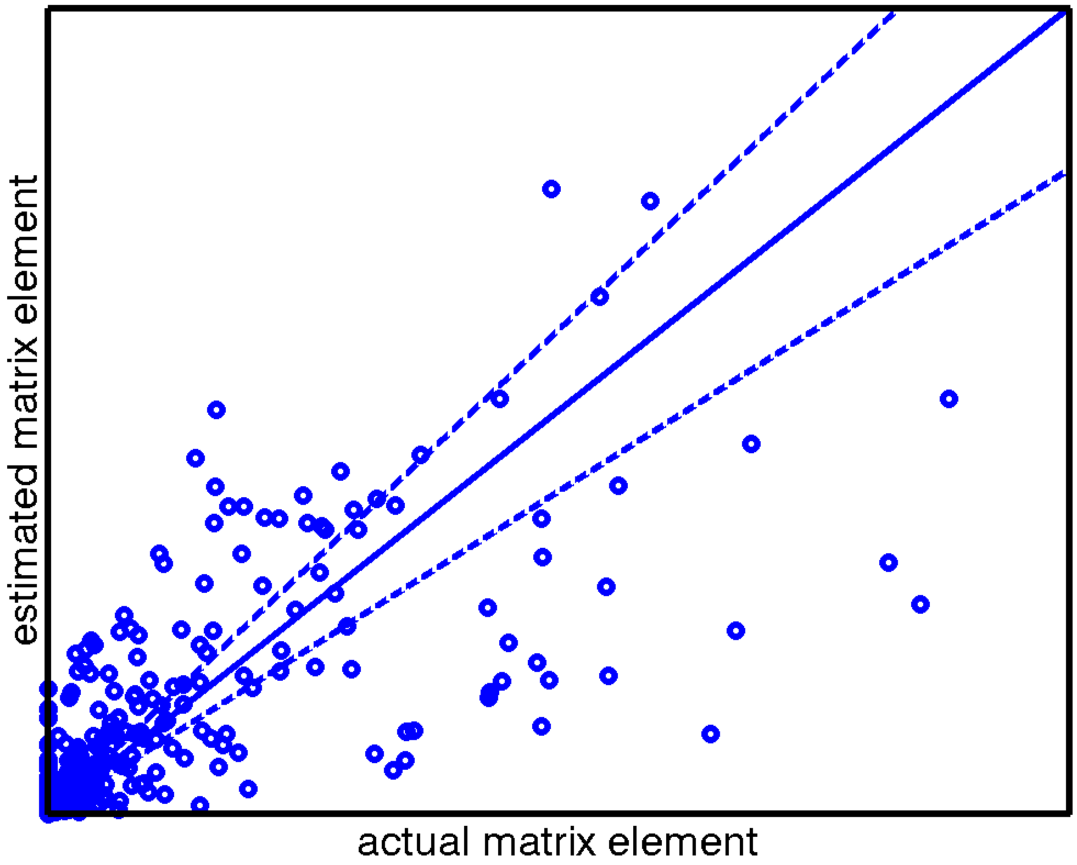
# Gravity Model

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

$$y_1 \propto x_1\, x_2$$
$$y_2 \propto x_2\, x_3$$
$$y_3 \propto x_1\, x_3$$

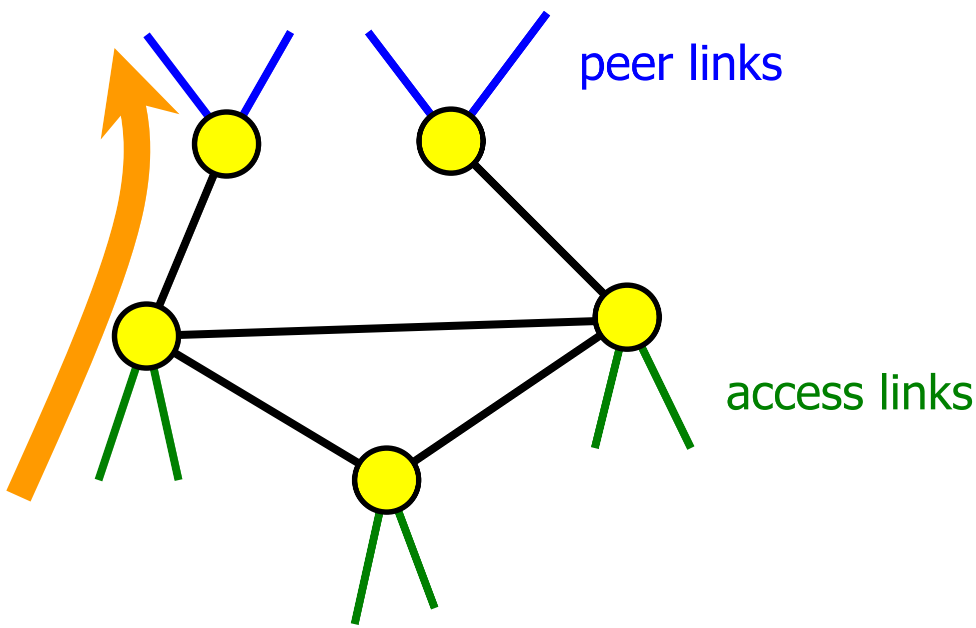
- Assumes there is no systematic difference between traffic in LA and NY
  - Only the total volume matters
  - Could include a distance term, but locality of information is not as important in the Internet as in other networks

# Simple gravity model



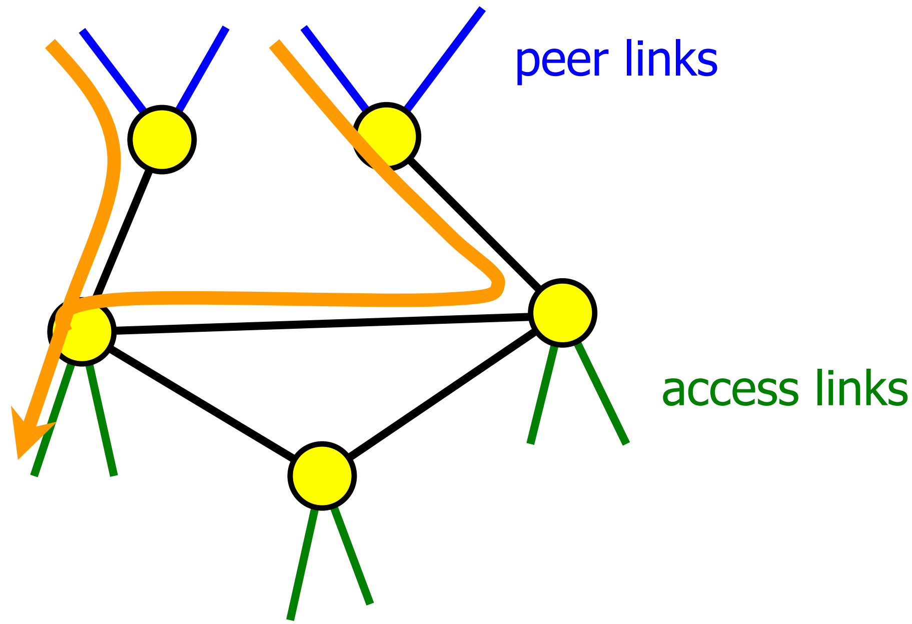
# Generalized gravity model

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

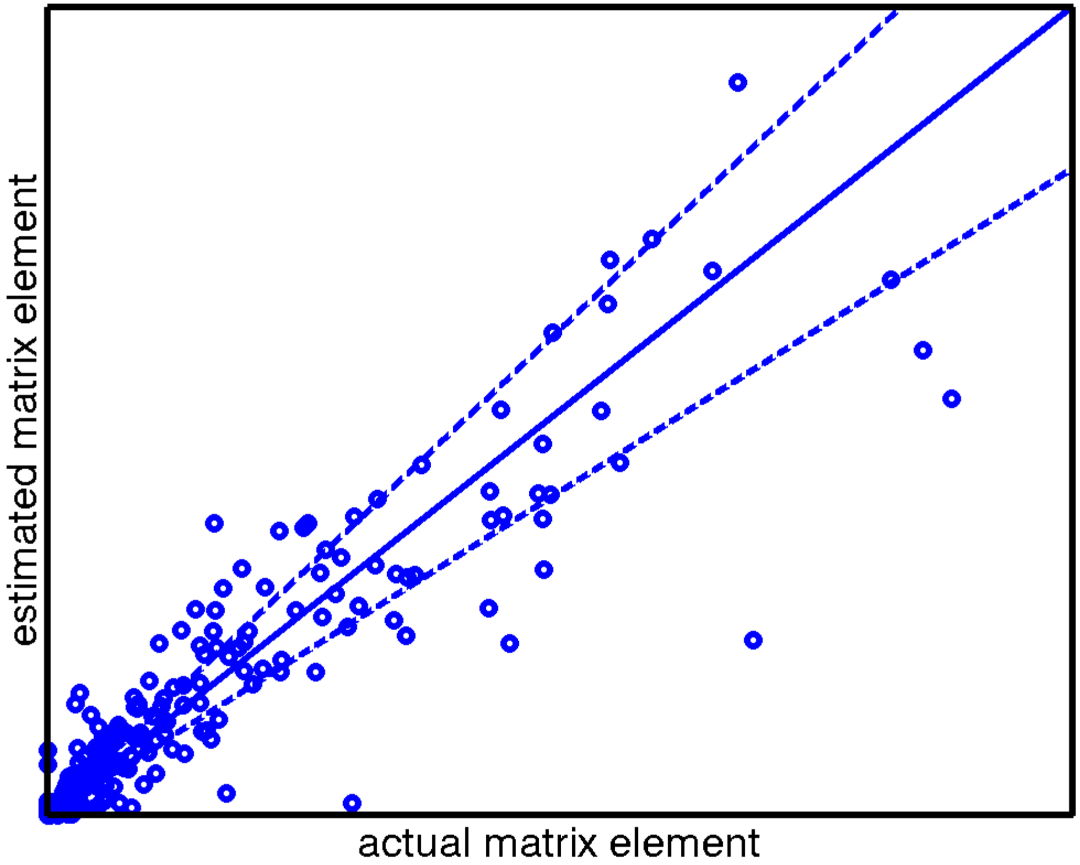


# Generalized gravity model

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

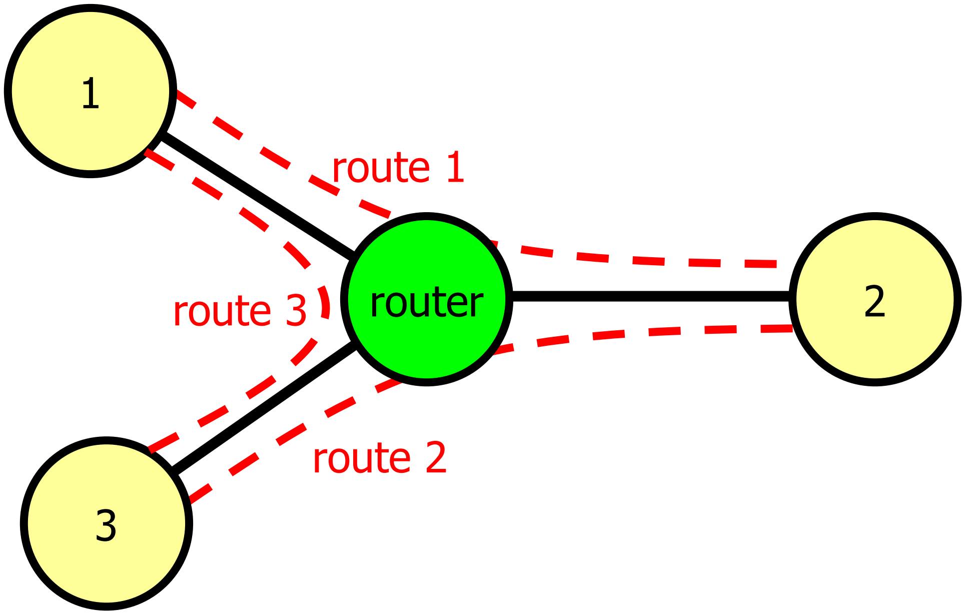


# Generalized gravity model



# Tomographic approach

- Solve the constraints




$$x = A^T y$$

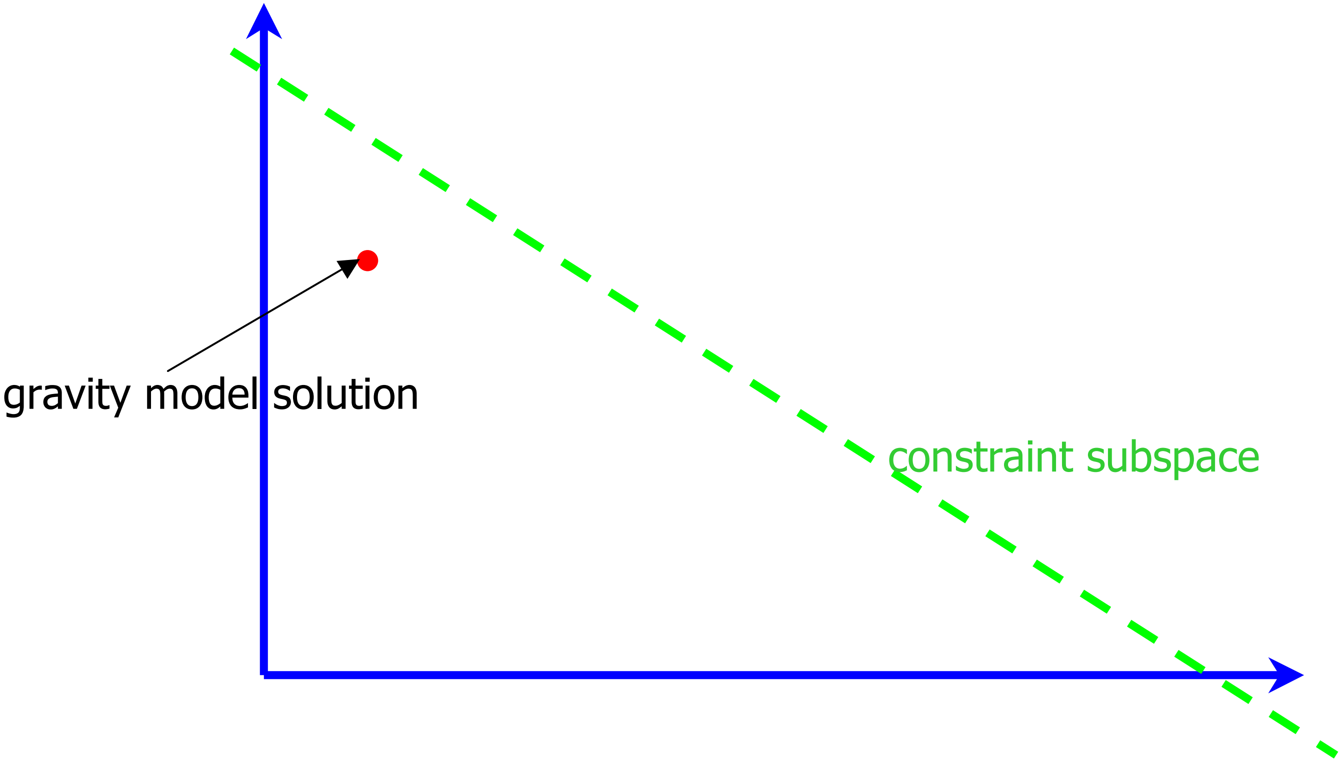
# Direct Tomographic approach

- 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 routers
    - Can reduce size of problem (by looking at the core)
      - Still orders more routers than PoPs
  - Model may not be correct -> result in problems

- Alternative: use the gravity model

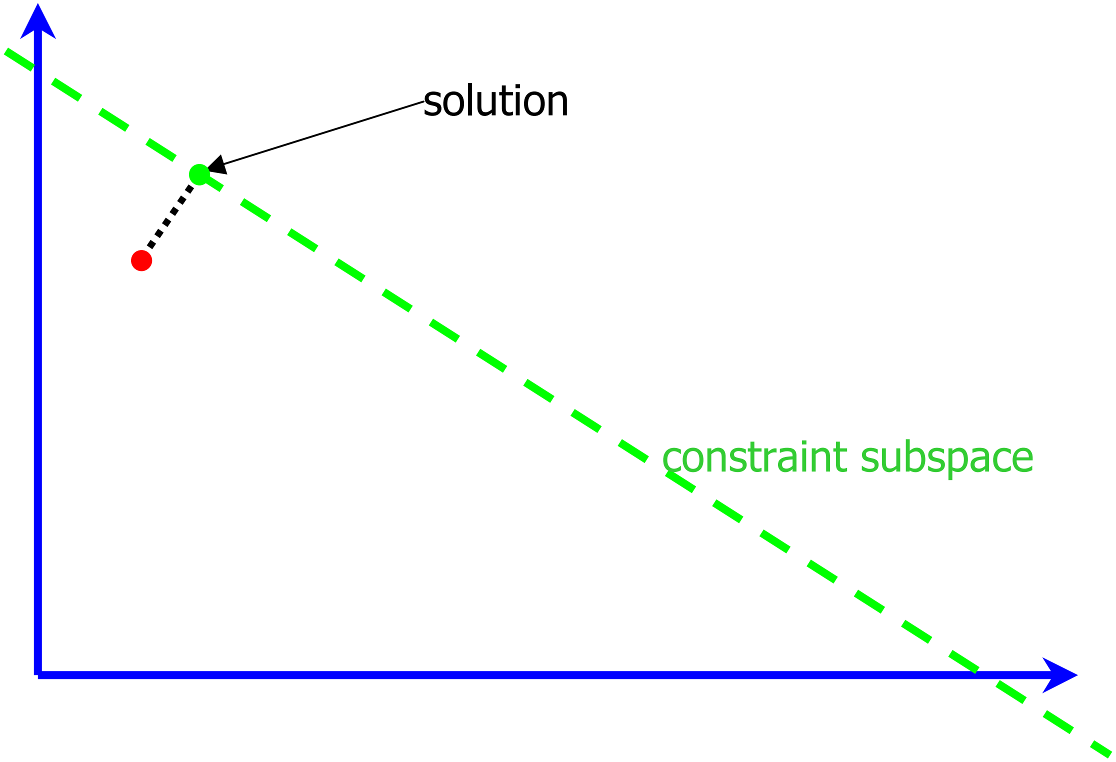
# Combining gravity model and tomography

- In general the aren't enough constraints
- Constraints give a subspace of possible solutions



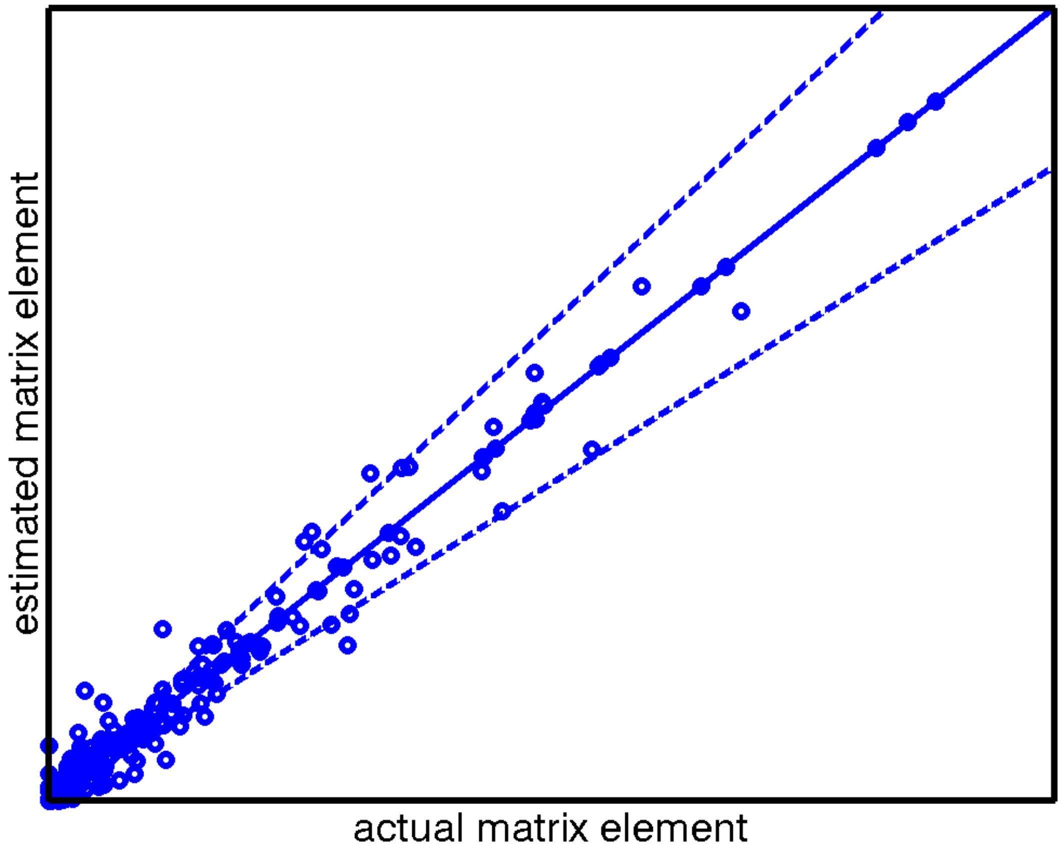
# Solution

- Find a solution which
  - Satisfies the constraint
  - Is close to the gravity model *(in some sense)*

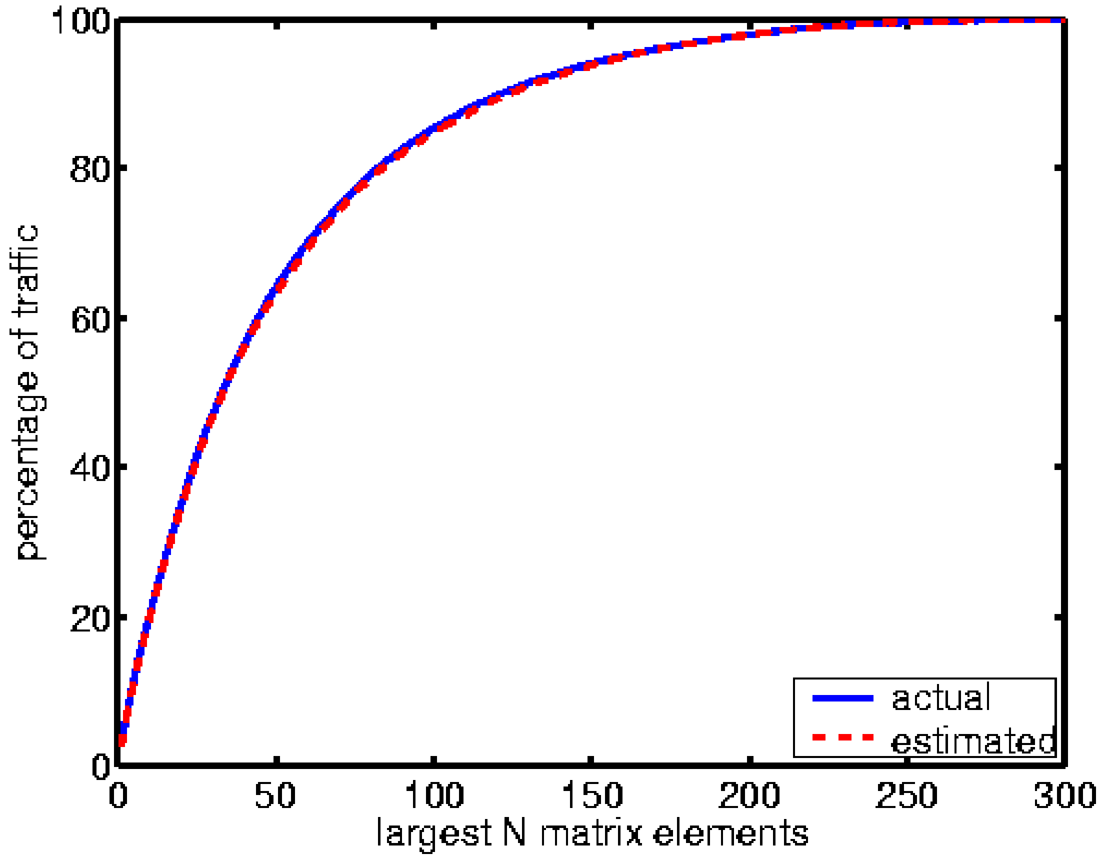


# Validation

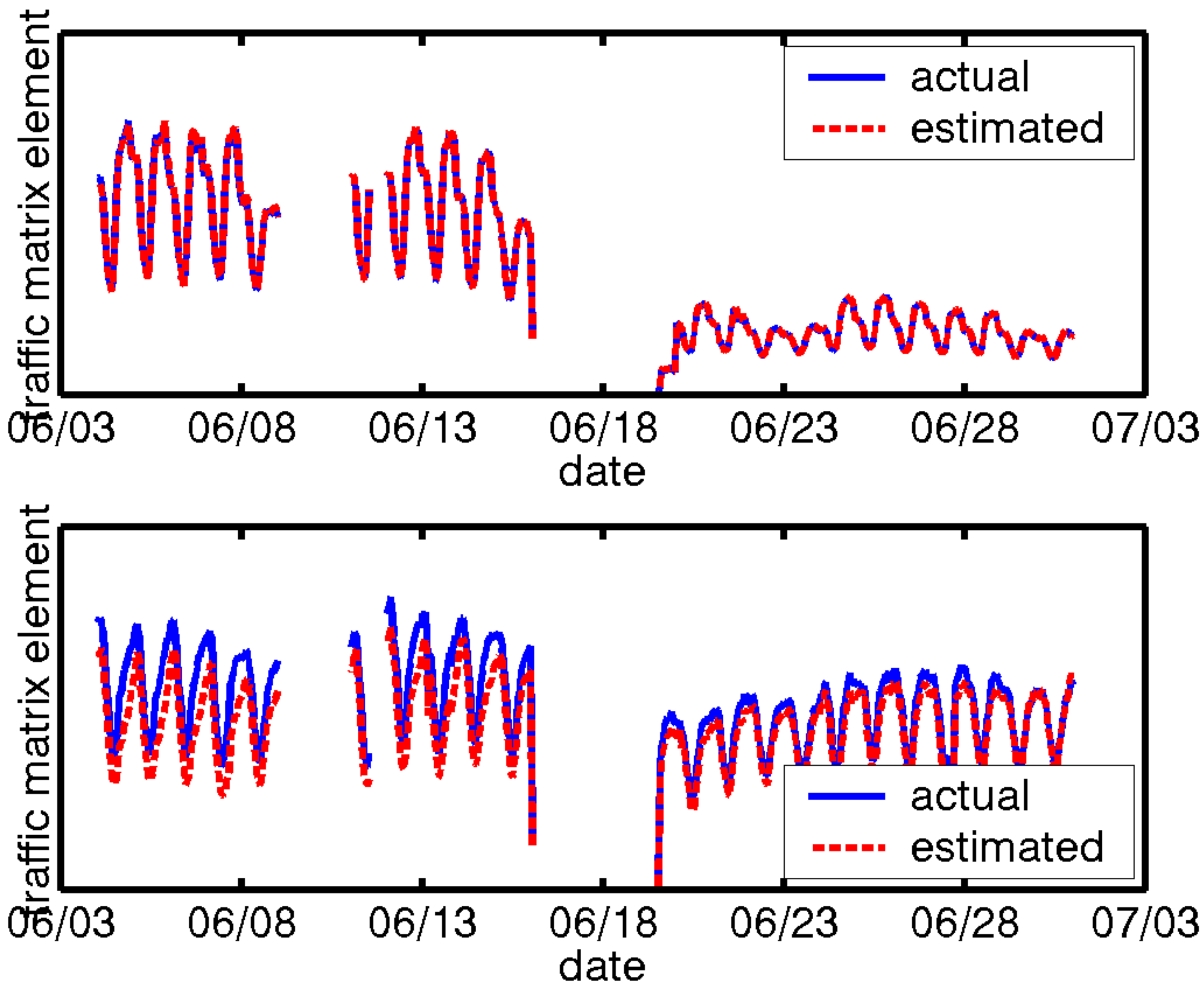
- Results good: ±20% bounds for larger flows
- Observables even better
- Robust
- Fast



# Distribution of flow sizes



# Estimates over time



# Information Theory

- natural relationship to information theory
  - Max entropy:
    - maximize uncertainty given a set of constraints
  - 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)
  - provides a natural distance for us in the previous algorithm
      - Quadratic distances are a linear approximation to the KL distance

# Insights

- Gravity model = independence of source and destination
  - Generalized gravity model = independence conditional on class of the source and destination
    - Can now rigorously derive this model
- There is a natural distance metric for this problem
- The solution can now be seen as showing "how far" we are from the gravity model in a probabilistic sense
- We can quantify the distance of the solution from any particular model – e.g. general vs simple gravity model
  - Provides a direct method for testing quality of priors, independent of algorithm used to get solution
  - For example, choice model prior used by SprintLab
- We know how to add in additional information rigourously
  - Isolated netflow
  - Local traffic matrices

# Part III: Applications

# Existing Applications

- Network Reliability Analysis
  - Consider the link loads in the network under failure
  - 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
  - Used in Planned Cable Intrusions (PCIs)
- Capacity planning
  - Results have been used in backbone capacity planning
    - Since Oct 2002 (in conjunction with other data)

# Routing optimization

- Used with OSPF optimization
  - Get within 6% of OSPF optimum using true TM
  - Get within 12% of absolute best (e.g. using MPLS)
- Has been used on a more limited basis, in connection with reliability analysis
  - OSPF weights computed by trial and error
  - Aim: prevent negative impact from failures
    - Concern in 2002 over three large links in a shared risk group

# Conclusion

- Nice algorithm
  - Connection with transport theory
  - Connection with information theory
- Practical applications
  - Network reliability
  - Capacity planning
  - Routing optimization
- To Do
  - Build better prior models
  - Study the traffic matrices themselves
  - Point-to-multipoint traffic matrices
  - Other applications
    - Anomaly detection

# Additional slides

# 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)

# 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

# Validation

# Some Approaches

- 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