An Overview of Traffic Matrix Estimation Methods

Nina Taft
Intel Research Berkeley
Outline

- Problem Statement
- 1\textsuperscript{st} generation solutions
- 2\textsuperscript{nd} generation solutions
- 3\textsuperscript{rd} generation solutions
- Summary
Collaborators

- Alberto Medina, Kave Salamatian, Christophe Diot
- Konstantina Papagiannaki, Antonio Nucci
- Augustin Soule, Rene Cruz, Anukool Lakhina, Mark Crovella
What’s a traffic matrix?

- Usually contains average values
- Can define variety of matrices
  - Select timescale
  - Select node granularity: router, prefix, POP, etc.

<table>
<thead>
<tr>
<th></th>
<th>City A</th>
<th>City B</th>
<th>City C</th>
</tr>
</thead>
<tbody>
<tr>
<td>City A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>City B</td>
<td></td>
<td>25 Mbps</td>
<td></td>
</tr>
<tr>
<td>City C</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Who’s traffic matrix?

- Want to capture a description of the demand between all pairs of cities (or routers)
- Describes traffic demands for a single network (e.g., AS domain)
- Cities could be office sites, this could be the demand for a corporation’s network
  - Good for IT!!
Motivation

• Why do we need traffic matrices? They are the input to all other traffic engineering tasks
  • Routing, Load Balancing, OSPF weight assignment
  • Capacity Planning, Reliability Analysis

• Why can’t we measure them directly?
  • Current definition of flow granularity too fine
  • Today’s solution is centralized and leads to enormous communications overhead

• How well we can do using limited information?
Example Problem

How much traffic flows between origin-destination pairs?
A->D
A->C
B->C
B->D

SNMP byte counts per link
How much traffic flows between?
A->D: 4
A->C: 1
B->C: 3
B->D: 0
Example: Another Solution

How much traffic flows between?
A->D: 2
A->C: 3
B->C: 1
B->D: 2

Type of equations:
Link1 = X_{AD} + X_{BD}
Problem Statement

- **Given**
  - per-link traffic counts (from SNMP) and
  - Knowledge about paths used between pairs of nodes (using ISIS/OSPF weights to compute shortest paths)

- **Determine the amount of traffic volume between all origin-destination pairs in an IP network.**
  - OD pairs: can be PoPs or backbone routers
Notation: Problem Formulation

Routing matrix

\[
\begin{pmatrix}
\text{Link1} \\
\text{Link2} \\
\text{Link3} \\
. \\
\text{Link L}
\end{pmatrix}
= 
\begin{pmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & . & . \\
\end{pmatrix}
\begin{pmatrix}
\text{X}_{\text{AB}} \\
\text{X}_{\text{AC}} \\
\text{X}_{\text{AD}} \\
. \\
. \\
. \\
\end{pmatrix}
\]

Have linear system:

\[Y = AX\]
Problem Statement

- System: \( Y = AX \)
- We have \( Y \) from SNMP link measurements
- We know \( A \) from OSPF link weights (so we can compute shortest paths)

- **problem: find \( X \)**
- **issue:**
  - \# links \(<\ <\ \#\ OD\ pairs\)
  - \( \Rightarrow \) underconstrained system
  - \( \Rightarrow \) infinite \# of solutions
1st Generation Approaches

- Deterministic: LP [Medina, Taft, et. al. SIGCOMM02]
- Statistical
  - Method of Moments [Vardi, JASA 1996]
  - Maximum Likelihood [Cao, et. al. JASA 2000]
- Modeling assumption on OD flow (Poisson, Gaussian)
Modeling: MLE Framework

- Assume each OD pair is Gaussian
  \[ X \sim \text{Normal}(\lambda, \Sigma) \]
- since \( AX = Y \)
  then the link counts are:
  \[ Y \sim \text{Normal}(A\lambda, A\Sigma A') \]
- where \( \lambda \) is the vector of mean rates:
  \[ \lambda = (\lambda_1, ..., \lambda_c) \]
- Idea: populate the traffic matrix with the conditional mean \( E(X|Y) \)
- Use maximum likelihood to get the estimate
Modeling: MLE Framework

- Assume each OD pair is Gaussian
  \[ X \sim \text{Normal}(\lambda, \Sigma) \]
- since \( AX = Y \)
  \[ Y \sim \text{Normal}(A\lambda, A\Sigma A') \]
- where \( \lambda \) is the vector of mean rates:
  \[ \lambda = (\lambda_1, ..., \lambda_c) \]
- Typically need to model the relationship between the variance and mean
  \[ \Sigma = \phi \text{diag}(\lambda_1^b, ..., \lambda_c^b) \]
- The parameters we need to estimate are:
  \[ \text{Let } \theta = (\lambda, \phi) \]
MLE approach (cont’d)

• We want conditional distribution: $X/Y$

• To estimate $\theta$, we need to maximize the log-likelihood equation:

\[
\ell(\theta | y_1, \ldots, y_K) =
\]

\[
-\frac{K}{2} \log |A\Sigma A'| - \frac{1}{2} \sum_{k=1}^{K} (y_k - A\lambda)'(A\Sigma A')^{-1}(y_k - A\lambda)
\]

• We populate the TM with: $X = E[X / \theta, Y]$
Why does this work?

- Make use of correlations across links
Performance (1st gen)

- Comparative Survey:
  - LP method errors: 100-200% (not a viable method)
  - Other method errors: 20%-60%. (not good enough)
  - MLE Method typical error range: 10% - 40%. (decent, but needs some improvement)
- High sensitivity to quality of prior information
- Outliers can be really bad, over 100%
The fundamental problem is that of an under-constrained system.

MLE estimation methods (e.g., EM) require a "starting point" (initial condition/prior/etc)

1st gen methods are sensitive to the starting point because can get stuck in local minima

Can we find "intelligent starting points" based on network properties? A model? More data?
Network Components

BR: backbone router
ER: edge router

POP 1
ER
BR
ER
BR

POP 2
BR
ER
BR
ER
BR
ER

POP 3
BR
ER
BR
ER

Access & peering links
Alternate Problem Statement

- Let $R_i$ be total amount of traffic leaving POP i

- Traffic $\text{POP}(i\rightarrow j) = R_i \alpha_{ij}$

- What is $\alpha_{ij}$?
  - the proportion of total emitted traffic headed to each egress POP
  - called the “fanout”

- Problem: estimate the fanouts $\alpha_{ij}$
Gravity model

• Link-to-link gravity model:

\[ X_g(L_i, L_j) = X^{\text{in}}(L_i) \frac{X^{\text{out}}(L_j)}{\sum_k X^{\text{out}}(L_k)} \]

• Use this to build up router-to-router gravity model

• Solve min \( ||X - X_g|| \) s.t. \( ||AX - Y|| \) is minimized
  • Use a least squares type solution
Performance (2nd gen)

- Validation data: obtained using Netflow from AT&T U.S. backbone for one month
- Use largest flows constituting 75% of total traffic load
- 5th and 95th percentile of relative errors are within +/- 23%
- Worst case: 90%
3rd generation models

- 2nd generation methods used SNMP link counts plus smart priors.
- Can we further reduce errors?
- Carriers set a 10% average error rate as general target.
- How can we get more information/measurements?
Idea: Use Partial Flow Measurements

- Advances in flow monitoring technologies are making it more feasible to use them. BUT, the communications overhead is still very large.
- Idea: can we use flow monitors occasionally?
- What can we do if we can get an exact copy of a traffic matrix periodically?
- Calibrate smart models
Linear Estimation

- We have an inherently linear system: \( Y = A X \)
- Want an estimate of a random sequence \( \{X(t)\} \) that we cannot directly observe, from another sequence \( \{Y(t)\} \) that we can observe.
- Estimation theory tells us that the best MMSE estimator is \( E(X/Y) \).
- SNMP is known to be noisy, so observations should be described by \( Y = A X + V \) where \( V \) is a Gaussian noise term.
Knowledge about the data

- We know something about X: it has temporal and spatial correlations.
Temporal and Spatial Correlations

- Let $X_{t+1} = C X_t + W_t$ where

$$
\begin{bmatrix}
X_1(t+1) \\
X_2(t+1) \\
\vdots \\
X_N(t+1)
\end{bmatrix} =
\begin{bmatrix}
c_{11} & c_{12} \\
& c_{22} \\
& & \ddots \\
& & & c_{22}
\end{bmatrix}
\begin{bmatrix}
X_1(t) \\
X_2(t) \\
\vdots \\
X_N(t)
\end{bmatrix} + 
\begin{bmatrix}
w_1(t) \\
w_2(t) \\
\vdots \\
w_N(t)
\end{bmatrix}
$$

- diagonal terms capture temporal correlation, and off-diagonal terms capture spatial correlation

- And ‘w’ captures traffic fluctuations (noise)
Putting it all together

- We have a classical linear dynamic system

\[ X_{t+1} = C X_t + W_t \]
\[ Y_t = A X_t + V_t \]

Model for OD flows, capturing their evolution. Includes temporal and spatial dependencies.

Traffic fluctuations

Measurement noise

Observables as a linear function of unobservables
Kalman Filtering

- Does both estimation and prediction
- Produces best linear minimum variance estimator

Prediction:
\[ \hat{X}_{t+1|t} = C \hat{X}_t|t \]

Estimation:
\[ \hat{X}_{t+1|t+1} = \hat{X}_{t+1|t} + G_{t+1} [ Y_{t+1} - A \hat{X}_{t+1|t} ] \]

- estimate
- predictor
- gain
- error
Adapting / recalibrating

- How do we get the C matrix?
- Turn on flow monitors for 24 hours and measure it!
- Suppose it changes over time?
- Check for diversion and recalibrate when necessary
  - Check the errors throughout a day
  - If more than 10% of them are > $2 \times \text{Var}(err)$, then turn flow monitors on again for another 24 hour period
Validation Data

- Collected at Sprint for 3 weeks during summer 2003
- European backbone
  - 13 POPs (Point of Presence. One POP per city.)
  - 27 Routers
- Used Netflow everywhere (at all edges)
  - expensive full-on measurement solution
Performance: Error Metrics

- Relative L2 norm
  - Spatial error:
    \[
    ERR - SP(n) = \sqrt{\sum_{t=1}^{T} \left[ X(n,t) - \hat{X}(n,t) \right]^2} / \sqrt{\sum_{t=1}^{T} X^2(n,t)}
    \]
  - Temporal error:
    \[
    ERR - TI(t) = \sqrt{\sum_{n=1}^{N} \left[ X(n,t) - \hat{X}(n,t) \right]^2} / \sqrt{\sum_{n=1}^{N} X^2(n,t)}
    \]
Spatial Errors

Empirical CDF

$F(x)$

$x=L2$ norm, Spatial Error

Gravity
Kalman
Temporal Errors

Empirical CDF

F(x)

x=L2 norm, Temporal Error

- Gravity
- Kalman
Overhead Metric

- At what cost do we get lower errors? It costs us to get the data for model calibration.
- Let $D(l) = \# \text{days link ‘l’ used a flow monitor}$
- Then, $OH = \frac{\sum_{l=1}^{L} D(l)}{21 * L}$
  
  Units are ‘link-days’

- $OH=1$ for brute force full-on measurement
- $OH=0$ for gravity model (no direct flow measurements)
Errors vs. overhead

- Average Relative L2 Temporal Error (%)
- Measurement overhead metric (%)

Legend:
- Gravity
- Kalman
- FullMeasure
Summary

• Kalman can do much better in terms of both spatial and temporal errors
  • Due to spatio-temporal model

• Tradeoff: we can cut the errors in half, at the cost of 10% measurement overhead

• For 90% less measurements than a full-measurement approach, we can produce highly accurate estimates.

• We rid of the bias (mostly) using partial flow measurements
That’s it!

THANK YOU …
more questions?