Traffic Matrix Estimation of an IP Network

by

Usha Chengan

A Thesis
Submitted to the Faculty of Graduate Studies
In Partial Fulfillment of the Requirements
For the Degree of

Master of Science

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada

Supervisor: Attahiru Sule Alfa

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

Traffic Matrix Estimation of an IP Network

BY

Usha Chengan

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

OF

MASTER OF SCIENCE

Abstract

A point-to-point *traffic matrix* gives the volume of traffic between Origin-Destination (OD) pairs in an Internet Protocol (IP) network. This knowledge helps Internet Service Providers (ISPs) in performing various traffic engineering tasks such as routing protocols configuration, network planning and business planning. Measuring this traffic matrix directly is difficult and costly. However it is believed that it might be possible to obtain a reasonable estimate of this matrix from link loads. Hence, there is an on-going research effort to develop efficient and effective methods for inferring the traffic matrix from other readily available data such as link load measurements, routing and configuration data. There are different techniques existing in the literature for estimating traffic matrix from such easily measured data.

In this thesis, we selected three techniques which are known to perform very well most of the time, namely Tomogravity (TM), Entropy Maximization (EM) and Linear Programming (LP), and carried out a detailed comparative study among them. We found that for small networks Linear Programming performs better than Tomogravity and Entropy Maximization. On the other hand, for large networks, Tomogravity and Entropy Maximization perform equally well and much better than Linear Programming. Based on the comparative study, we proposed new directions for improving how to estimate traffic matrix using modified versions of the existing techniques. First, we improved the Linear Programming approach by incorporating additional network-specific information. We found that doing this improves the performance of Linear Programming further and also helps in estimating the traffic matrix of small networks precisely. Second, we developed

a hybrid model that combines Tomogravity, Entropy Maximization and Linear Programming. This model best estimates the traffic matrix compared to the other three methods. Third, we proposed a hierarchical model that helps to estimate the traffic matrix of large networks at coarse level with better accuracy. This hierarchical model can also be used to assess the quality of traffic estimation for real life networks for which the true matrix is not known.

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. I would like to express my gratitude to Bell Canada for providing real data of their network for this project.

I would like to express my special thanks to my supervisor Dr. A.S.Alfa for his support, patience, guidance, and inspiration all the time throughout this research and writing of this thesis.

I would like to acknowledge my thesis examiner Dr. Ekram Hossain and Dr. Ellen Liu for reviewing and commenting the thesis.

Many thanks go to the people of my research group, Dr.Luo, Al-Wasi, Subrata, Mostafizur, Gary, Sangho, Arash, Qiana, Shirley, Haitham, Eugene, Daglenia, and Irene for their technical discussions for this research. I would like to say special thanks to Mostafizur and Subrata who helped me in this research through their valuable suggestions and comments.

Especially, I would like to give special thanks to my husband, Rajesh, whose constant motivation and encouragement helped me to complete this work. I would also like to mention about our eleven week old little daughter Harshitha who came to life and added strength, confidence and occasional physical and mental challenges throughout my research work. It is an everlasting experience in my lifetime.

Also, I would like to express my sincerely gratitude to my parents for their encouragement and support all the time.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Traffic Matrix Overview

A traffic matrix provides, for any point i in the network to any point j of the network, the volume of traffic $T_{i,j}$ over a given time interval. $T_{i,j}$ is defined as the amount of traffic originating from i and destined to j. It is an element of a matrix T, known as the traffic matrix.

Let us consider a network that consists of 4 nodes as shown in Figure 1.1. The links are bi-directional. There is a traffic matrix T associated with this network. Let T be given as in Figure 1.2. An element $T_{i,j}$ of this matrix represents the amount of demand for IP traffic to go from node i to j, i,j = 1,2,3,4. This demand is then assigned along the appropriate paths based on some protocol e.g. Open Shortest Path First (OSPF). The assigned traffic results in flows along the links. These we call the link loads.

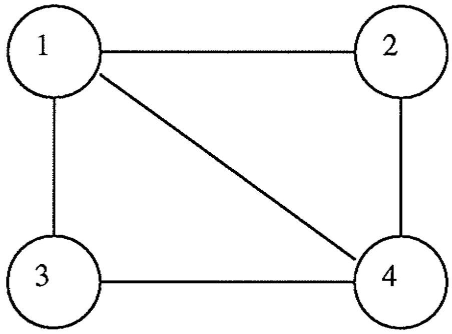

Figure 1.1: 4-node Network

Node	1	2	3	4
1	0	$T_{1,2}$	$T_{1,3}$	$T_{1,4}$
2	$T_{2,1}$	0	$T_{2,3}$	$T_{2,4}$
3	$T_{3,1}$	$T_{3,2}$	0	$T_{3,4}$
4	$T_{4,1}$	$T_{4,2}$	$T_{4,3}$	0

Figure 1.2: Traffic Matrix of 4-node Network

This traffic matrix has tremendous utility in wide variety of traffic engineering tasks such as load balancing, routing protocols configuration, network design, capacity planning and business planning. However, inputs required for direct measurement of traffic matrix is not available in large IP networks. On the other hand link load measurements, network routing and configuration data are readily available and traffic matrix can be estimated from these available data.

1.2 Thesis Contribution

The contribution of our thesis can be divided into two parts. In the first part, we conduct a detailed comparative evaluation of three most efficient techniques that have been proposed in the literature for traffic matrix estimation problem. The first technique that we study is the Tomogravity that combines gravity model with tomographic estimation step. The second technique is an information-theoretic approach known as Entropy Maximization. The third technique is based on the application of Linear Programming.

In the second part, we propose new directions for traffic matrix estimation based on the lessons learned from the comparative study. First, we improve the performance of Linear Programming by including additional constraints. Second, we develop a Hybrid Model that combines Tomogravity, Entropy Maximization and Linear Programming targeting better traffic matrix estimation. Third, we develop a Hierarchical model for estimating the traffic matrix of large network at coarse level with better accuracy.

1.3 Thesis Organization

The thesis is organized as follows. In Chapter 1, we briefly talk about traffic matrix and illustrate our contributions to this research. In Chapter 2, we describe the three estimation techniques namely Tomogravity, Entropy Maximization and Linear Programming that we consider in our comparative study. In Chapter 3, we present the experimental results of comparative analysis of the above said three techniques. In chapters 4, 5 and 6, we discuss our new directions for estimating traffic matrix namely - Improvements of Linear Programming by including additional constraints, Hybrid model and Hierarchical model respectively. In chapter 7, we illustrate the application of traffic matrix estimation techniques on Bell Canada IP Network. Chapter 8 concludes the paper.

Chapter 2

Literature Review

In this chapter, we briefly explain the three traffic matrix estimation techniques namely Tomogravity, Entropy Maximization and Linear Programming that we consider in our comparative study.

2.1 Tomogravity

Zhang et al. [1] developed "Tomogravity" for estimating traffic matrices in IP networks from link load measurements and routing configuration information. As the name "Tomogravity" indicates, it consists of two basic steps namely: a) a gravity modeling step and b) a tomographic estimation step.

Gravity Modeling

Social scientists commonly used gravity model, to model the movement of people, goods or information between geographic areas. In general, gravity model can be represented by the following equation:

$$X_{ij} = \frac{R_i A_j}{f_{ij}}$$

where,

X_{ij} - is the force from i to j

R_i - is the repulsive factors that are associated with i

A_j - is the attraction factors that are associated with j

f_{ij} - is the friction factor form i to j

In our context, we can interpret the above terms as follows:

X_{ij} - is the amount of traffic from node i to j

R_i - is the amount of traffic entering the network from node i

A_j - is the amount of traffic leaving the network via node j

f_{ij} - encodes the locality information for each source-destination pairs, usually a measure of travel or flow impedance.

The problem in finding the friction factor, f_{ij}, is that it leads to the problem of the same size as solving the traffic matrix. Hence, researchers have often approximated the friction factors to be a common constant, for example $f_{ij} = \sum_i R_i$ or $f_{ij} = \sum_j A_j$.

However, one cannot expect accurate results from gravity model. To overcome this drawback, the tomographic estimation step is applied to refine the initial results obtained from gravity modeling. This corrects the estimation errors caused due to the underlying assumption and thereby improves the overall accuracy.

Tomography

As a next step, to refine the solution obtained from gravity modeling, the tomographic method is used. This method is based on the following linear equation:

$$X = AT$$

where,

X - $(x_1, x_2 \ldots\ldots\ldots\ldots\ldots x_L)^T$, represents the L link load measurements.

T - $(t_1, t_2 \ldots\ldots\ldots\ldots\ldots t_M)^T$, represents the M traffic matrix elements.

A - represents the routing matrix.

More specifically, the tomographic step attempts to solve the following quadratic programming problem:

$$\text{Min} \quad \| T - T_g \|$$

$$\text{s.t. } \| AT - X \| \text{ is minimized}$$

where,

T - traffic matrix to be estimated by solving the above problem

T_g – initial traffic matrix obtained by using gravity modeling

$\| . \|$ - is the L2 norm of a vector

Zhang et al. also investigated weighted least-squares (or wlse) solutions as the objective function to minimize the above quadratic program. The authors investigated three different weighting schemes; the weight for each term of the traffic matrix is either i) constant, ii) linearly proportional to the terms in the gravity model traffic matrix or iii) proportional to the square root of the gravity model.

We replaced the gravity model by utility model [6] which is based on choice models to generate the initial solution and then refined it by using the tomographic estimation step. This is discussed in Appendix A.

2.2 Entropy Maximization

Zhang et al. [2] proposed a new approach to traffic matrix estimation using a regularization based on "entropy penalization". This information-theoretic approach chooses the traffic matrix consistent with the measured data that is information theoretically closest to a model in which the source/destination pairs are stochastically independent.

In information theory, the Discrete Shannon Entropy of a discrete random variable X taking values x_i is defined as,

$$H(X) = -\sum_i p(x_i) \log_2 p(x_i)$$

where, $p(x_i)$ is the probability that a random variable X assumes the value x_i. H(X) takes its maximum value when X is uniformly distributed.

Maximum Entropy Principle states that the unknown probability distribution can be obtained by listing all the constraints that has to be satisfied by the unknown probability distribution and searching for the probability distribution that maximizes the entropy subject to the listed set of constraints. That is, given some set of mathematical constraints C, the random variable X is estimated by maximizing the entropy H (X|C).

The conditional probability of one random variable Y with respect to another X is defined as:

$$H(Y \mid X) = -\sum_j p(x_i) \sum_i p(y_i \mid x_i) \log_2 p(y_i \mid x_i)$$

where $p(y_i \mid x_i)$ is the probability that Y = y_i conditional on X = x_i.

We can also define the Shannon information I (Y | X) = H (Y) – H (Y | X). Here, I (Y | X) represents the decrease in uncertainty about Y from measurement of X, or the information that we gain about Y from X. This is also referred to as mutual information and written as I (X, Y). The mutual information I (X, Y) is written as follows:

$$I(X,Y) = \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} = K(p_{x,y} \parallel p_x \times p_y)$$

where $K(f \parallel g) = \sum_i f_i \log(f_i / g_i)$ is the Kullback-Leibler divergence of f with respect to g, a well-known measure of distance between probability distributions.

The goal of Entropy Maximization is to minimize the mutual information which is obtained by the following optimization problem

$$Min \sum_l (T(l) - N \sum_{s,d} A(s,d;l) p(s,d))^2 + \lambda^2 I(S,D)$$

Here, *T(l)* is the traffic on link l. *A(s,d;l)* denotes the routing matrix i.e. *A(s,d;l)* gives the fraction of traffic from s to d which crosses link l. *N* is the total amount of traffic entering the network. *p(s)* is the fraction of traffic exiting from source s. *p(d)* is the fraction of traffic entering the destination d. *p(s,d)* is the fraction of traffic exiting from source s and entering the destination d. λ is the penalizing factor that depends on the level of noise. *S* is the set of origins and *D* is the set of destinations.

pdsco [4] and *maxent* [5] are the two different matlab packages available to solve the above optimization problem.

2.3 Linear Programming

Conway and Li [3] developed methods to estimate source-destination traffic proportions from aggregate link and source/sink traffic measurements that are measured over a finite number of disjoint time periods. In time period t, let

$\lambda_i^{(t)}$ be the total measured traffic entering the network via node i,

$\beta_i^{(t)}$ be the total measured traffic leaving the network via node i, and

$\gamma_k^{(t)}$ be the total measured traffic on link k.

The proportion of traffic from node i to node j, is $p_{ij}^{(t)}$, where $p_{ij}^{(t)} \geq 0$, $p_{ii}^{(t)} = 0$ and $\sum_{j=1}^{N} p_{ij}^{(t)} = 1$. The number of unknown proportions is N (N-1), where N is the total number of nodes in the network.

Suppose, we have the set of traffic measurements $\{\lambda_i^{(t)}, \beta_i^{(t)}, \gamma_k^{(t)};\ 1 \leq i \leq N, 1 \leq k \leq L\}$, then we can write the following sets of linear equations:

$$\beta_i^{(t)} = \sum_{j=1}^{N} \lambda_j^{(t)} p_{ji} : 1 \leq i \leq N$$

$$\gamma_k^{(t)} = \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i^{(t)} p_{ij} r_{ijk} : 1 \leq k \leq L$$

$$\sum_{j=1}^{N} p_{ij} = 1 : 1 \leq i \leq N$$

The parameter r_{ijk}, defines the routing of the traffic in the network. The value of r_{ijk} represents the proportion of traffic from node i to j that is routed via link k. The value of $r_{ijk} = 0$, if the traffic from node i to j does not use the link k. The above set of equations are solved to find p_{ij}'s.

However, one measurement period, generally does not provide sufficient number of independent equations to solve the total unknowns. So, multiple measurement time periods are used to construct the required number of independent equations. The above equations are arranged in the conventional matrix form AX = B and the following linear optimization problem is set up to solve for the unknowns:

LP Problem

Find a solution $[[x^{(1)}, \ldots\ldots x^{(N)}], [e^+, e^-]]$ which minimizes $1(e^+ + e^-)$ subject to the following constraints:

$$A[x^{(1)} \ldots\ldots\ldots\ldots x^{(N)}] = B + e^+ + e^-$$

$$1x^{(i)} = 1 \text{ for } 1 \leq i \leq N$$

$$X^{(i)} \geq 0^T$$

$$e^+ \geq 0^T$$

$$e^- \geq 0^T$$

where e^+ and e^- are positive and negative error terms, 1 and 0 and compatible vector of ones and zeros, $x^{(i)}$ is the column vector of unknowns corresponding to $\{ p_{ij} : 1 \le j \le N, j \ne i \}$

Conway and Li also developed another approximation method to reduce the computational complexity. But this approximation method does not perform well compared to the exact method.

2.4 IP Traffic Matrix Estimation Methods: Comparisons and Improvements

In [10], the authors have carried out a comparative study of three popular techniques namely Tomogravity, Entropy Maximization and Linear Programming. The experiments with gravity model were conducted using *wlse* and Entropy Maximization with *pdsco* [11] and *maxent* [12]. Based on the set of preliminary experiments, it was found that *wlse* and *pdsco* are better than maxent and Linear Programming in estimating traffic matrix. It was also found that *wlse* and *pdsco* does not differ much in average percentage error in estimating point to point traffic. On the other hand, it was found that change in amount of traffic in OD pair affects accuracy of calculation.

Based on the comparative study, it was concluded in [10] that the Tomogravity method best estimates the traffic matrix among the methods that were tested. Then, the authors incorporated some enhancements that improve the Tomogravity method. Specifically, it was found that knowing some point to point traffic reduces the error in estimating the traffic matrix using wlse method. However, this knowledge does not help to improve the performance of *pdsco*. Tomogravity method using wlse results in different amount of error for different networks with different traffic distributions for different weight assignments.

Chapter 3

Comparative Study of Tomogravity, Entropy Maximization and Linear Programming

3.1 Comparison Methodology

In reality, measuring the "real" traffic matrix is not economically feasible in large IP networks. Therefore, we do not have any real traffic matrix against which we can compare the results obtained using traffic matrix estimation techniques. This makes assessing the quality of traffic matrix estimation techniques difficult.

Hence, here for each test case we consider a network topology and generate synthetic traffic matrix T, where $T_{i,j}$ defines the traffic flow from Point of Presence (POP) i to POP j. The routing of the traffic is assumed to follow the shortest path, with equal weights assigned to all links. That is, if we have more than one shortest path between a given source-destination pair i-j, then the total traffic from i to j will be equally divided among the available routes. By this way we can define the entries of the routing matrix A. The traffic demands T will be routed on the network according to A, which determines the load on each link. The link load Y can be easily obtained using Y = AT. The set of data used as input to the three different traffic matrix estimation techniques are the routing matrix A, link load Y, production and attraction of each node in the network and we obtain the estimated traffic matrix $\hat{T}$ as output. We then compare the estimated traffic matrix $\hat{T}$ with the original traffic matrix T to assess the performance of the each method. We evaluate each method with respect to the average and maximum percentage errors yielded. For example, for a network with N nodes the average and maximum percentage error values is calculated as given below:

$$\text{Average \% error} = (\sum_{i=1}^{N}\sum_{j=1}^{N} abs(T_{i,j} - \hat{T}_{i,j}) / T_{i,j} * 100) / (N-1); i \neq j$$

$$\text{Maximum \% error} = \max(abs(T_{i,j} - \hat{T}_{i,j}) / T_{i,j} * 100); i, j = 1,2....N; i \neq j$$

3.1.1 Topologies

Medina et al. [6] compared the performance of Time-Varying Network Tomography [14], Linear Programming [13] and Bayesian Inference [15] techniques and concluded that Network Tomography is the best of the existing techniques. Also, Md. Mostafizur Rahman et al. [10] performed the comparative study of TM, EM and LP techniques and concluded that TM is the best among the three techniques that is used for estimating point to point traffic. Both the authors compared the performance of various traffic matrix estimation techniques considering a single 4-node and 14-node network. Our comparative study differs from the work done by above authors by considering 10 different models (model 1 - 10) for each 4-node and 14-node topology. Different network models for the same topology (4 - node or 14 - node) are considered in separate test cases (one test case for each network model) to observe the sensitivity of the estimation techniques to network model types. The reason for selecting two different topologies one representing small network and the other one representing large network is to observe the sensitivity of the estimation techniques to network size. Each node in 4(14)-node topology represents a POP, and the link represents the aggregated connectivity between the routers belonging to a pair of adjacent POPs. The 10 different 4-node and 14-node network models that we selected are given in Appendix B.

3.1.2 Synthetic Traffic Matrices

We need to use synthetic traffic matrix since the real ones are not available. At the same time, the synthetic traffic matrix that we use must reflect the properties of the real traffic matrix. So, we consider five different types of synthetic traffic matrices that differ in the distribution used to generate their elements. We consider Constant, Poisson, Gaussian, Uniform and Bimodal traffic matrices. Constant traffic matrices are obtained by assigning a constant value of 300 Mbps, to all OD pair demands. Uniform traffic matrices are obtained by generating uniform random values in the interval [100; 500]. Poisson traffic matrices are generated from Poisson (λ), where λ is a continuous uniform random variable in the interval [100,500]. Normal traffic matrices are generated from $N(\mu = u; \sigma = 40)$ where u is a continuous uniform random variable in the interval [100,400]. Bimodal traffic matrices are generated by a mixture of two Gaussian random variables, the first Gaussian variable is $N(\mu_1 = 150; \sigma_1 = 20)$ with probability 0.8; and the second one is $N(\mu_2 = 400; \sigma_2 = 20)$ with probability 0.2. It is necessary to test any traffic matrix estimation technique against a variety of traffic distributions types to assess the dependency of the estimation technique on the distribution assumption made.

3.2 Comparing Tomogravity, Entropy Maximization and Linear Programming for different 4-Node Networks & traffic types

This section presents experimental results of the comparison between Tomogravity, Entropy Maximization and Linear Programming for 10 different 4-node networks and 5 various traffic types.

3.2.1 Constant Traffic

Here 10 different 4-node networks are considered with constant traffic types. The results of this analysis are presented below:

MODEL	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	35.01	43.24	7.33	221.90	226.85	27.37
2	46.00	46.66	11.43	146.55	190.41	37.67
3	0.00	0.00	0.00	0.00	0.00	0.00
4	36.43	36.68	31.85	126.95	155.39	134.00
5	31.22	36.19	29.31	165.40	170.33	117.38

6	8.40	3.00	4.43	25.20	9.19	9.85
7	43.46	45.17	32.88	227.15	230.13	128.98
8	28.05	7.93	4.44	93.25	29.80	12.83
9	38.87	45.09	32.21	205.35	231.52	149.54
10	38.08	37.79	32.25	140.95	168.24	150.71

Table 3.1: 4 Node Network - Constant traffic

From Table 3.1, we notice that LP performs better when compared to the other two methods for all models, except model 6. The above set of experiments also reveals that the performance of Tomogravity and Entropy Maximization depends on the network model. As we see in Table 3.1, EM performs better when compared to TM for model 6 & 8 and it's vice-versa for rest of the models. However, the performance of LP is independent on the network model for constant traffic type distribution.

3.2.2 Uniform Traffic

Here 10 different 4-node networks are considered with uniform traffic types. The results of this analysis are presented below:

MODEL	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	19.82	2.33	50.53	60.71	7.02	99.98
2	31.82	7.65	48.46	78.63	13.75	94.96
3	0.00	0.00	0.00	0.00	0.00	0.00
4	31.99	20.76	21.39	82.56	47.70	60.10

5	12.62	11.67	11.95	36.96	34.22	35.35
6	13.24	6.84	45.97	32.02	18.57	103.40
7	29.11	19.80	20.76	74.06	64.23	54.61
8	36.88	20.16	70.24	172.62	68.57	175.31
9	22.15	21.99	20.59	49.42	61.40	67.06
10	24.59	22.19	21.17	83.55	82.64	65.93

Table 3.2: 4 Node Network - Uniform traffic

In case of uniform traffic, EM achieves better results compared to TM & LP for most of the models (model 1 – 8). LP performs better compared to the other two methods only for model 9 & 10.

3.2.3 Poisson Traffic

Here 10 different 4-node networks are generated to be Poisson traffic types. The results of this analysis are presented below:

MODEL	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	17.25	7.24	5.48	43.11	18.54	9.51
2	20.94	11.04	8.94	43.26	30.57	23.10
3	0.00	0.00	0.00	0.00	0.00	0.00
4	23.02	15.20	15.41	68.24	34.88	38.30
5	17.01	12.54	11.44	68.24	40.75	33.91
6	4.80	14.61	10.37	13.43	30.50	29.76
7	26.75	15.87	12.37	68.34	43.77	33.95

8	24.51	16.57	12.41	78.25	37.98	34.87
9	17.95	26.55	25.81	43.56	58.41	60.60
10	31.97	30.53	25.11	115.28	104.48	57.69

Table 3.3: 4 Node Network - Poisson traffic

For 4-node networks with Poisson traffic, LP performs better compared to the other two methods for most of the models except model 4, 6 and 9. TM shows better results for model 6 & 9, and EM for model 4.

3.2.4 Normal Traffic

Here 10 different 4-node networks are considered with normal traffic type. The results of this analysis are presented in the below table. For normal type of traffic, LP performs better for all models except 2, 6, 7 and 8.

MODEL	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	32.00	16.13	6.07	125.88	74.11	17.26
2	33.28	17.68	42.96	118.09	64.56	118.85
3	0.00	0.00	0.00	0.00	0.00	0.00
4	29.63	24.83	24.30	99.25	74.52	66.56
5	17.13	16.23	15.16	51.39	54.16	55.37
6	27.46	21.44	41.20	74.10	63.33	98.97
7	28.29	24.51	25.49	96.28	70.46	70.44
8	26.97	22.75	25.20	63.99	61.71	65.14

9	32.42	28.49	21.76	69.99	79.97	81.34
10	31.32	31.41	21.55	126.14	145.40	80.18

Table 3.4: 4 Node Network - Normal traffic

3.2.5 Bimodal Traffic

Here 10 different 4-node networks are considered to be bimodal traffic. The results of this analysis are presented in Table 3.5. For bimodal type of traffic, LP performs better for most of the models.

MODEL	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	48.02	20.65	47.02	186.80	75.67	152.23
2	41.20	26.84	21.89	202.68	111.95	68.60
3	0.00	0.00	0.00	0.00	0.00	0.00
4	29.23	49.99	32.28	100.00	173.41	115.59
5	23.26	35.07	15.20	94.03	196.29	75.61
6	9.86	45.19	17.52	26.90	157.50	42.99
7	55.86	53.73	37.37	267.79	196.26	148.57
8	56.00	53.07	28.93	244.83	187.16	80.73
9	87.79	57.25	56.26	297.38	181.58	230.14
10	97.42	72.61	57.43	615.75	441.84	224.35

Table 3.5: 4 Node Network - Bimodal traffic

3.2.6 Summary

Based on the comparison of TM, EM and LP for different 4-node networks and traffic types, we conclude that LP performs better compared to the other two methods for most of the models and traffic types. As an exception, Entropy Maximization plays dominant role compared to LP only in case of uniform traffic.

3.3. Comparing Tomogravity, Entropy Maximization and Linear Programming for different 14-Node Networks & traffic types

This section presents experimental results of the comparison between Tomogravity, Entropy Maximization and Linear Programming for 10 different 14-node networks and 5 various traffic types.

3.3.1 Constant Traffic

Here 10 different 14-node networks are considered with constant traffic types. The results of this analysis are presented below:

Model	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	1.92	1.93	13.22	4.67	4.66	242.65
2	1.82	1.82	18.47	4.81	4.80	99.96
3	1.78	1.78	26.28	4.97	4.93	106.43

4	1.7	1.7	37.04	4.55	4.57	145.14
5	1.7	1.71	19.27	4.71	4.69	99.96
6	1.84	1.85	48.88	5.42	5.38	686.17
7	1.95	1.96	52.02	5.17	5.19	431.43
8	1.91	1.92	33.15	6.04	5.97	146.37
9	1.87	1.88	26.75	5.11	5.08	124.57
10	1.83	1.84	49.9	5.76	5.72	463.06

Table 3.6: 14 Node Network - Constant traffic

Considering the constant traffic, Tomogravity shows better results compared to the other two techniques for all the models.

3.3.2 Uniform Traffic

Here 10 different 14-node networks are considered with uniform traffic types. The results of this analysis are presented below:

MODEL	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	34.07	33.94	46.42	176.72	177.11	410.21
2	32.68	32.64	42.71	168.51	168.89	290.63
3	32.52	32.59	63.77	169.99	171.61	1268.65
4	30.19	29.96	76.08	179.19	181.31	1416.42
5	31.29	31.13	44.86	223.78	223.05	432.24
6	30.02	30.18	44.16	159.63	158.41	514.81
7	31.67	31.69	53.76	165.76	169.15	567.98
8	33.64	33.68	50	171.60	174.44	317.89

9	35.31	35.34	62.19	179.66	177.97	1349.56
10	32.28	32.36	83.2	202.06	200.26	943.57

Table 3.7: 14 Node Network - Uniform traffic

For uniform traffic, Tomogravity shows better results compared to the other two methods for most of the models and Entropy Maximization performs better for rest of the models. LP performs worse compared to the other two for all the models.

3.3.3 Poisson Traffic

Here 10 different 14-node networks are considered with Poisson traffic type. The results of this analysis are presented below:

Model	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	16.88	16.9	63.04	122.59	121.14	860.62
2	16.4	16.34	44.86	144.23	145.7	882.08
3	15.9	15.77	51.86	141.86	145.72	821.45
4	15.78	15.89	56.69	127.36	131.66	626.95
5	17.13	17.1	30.29	116.48	120.9	229.74
6	16.1	16.24	56.3	131.31	129.61	493.41
7	16.18	16.02	26.9	124.86	124.79	328.01
8	17.48	17.58	38.09	133.93	133.8	446.69
9	16.73	16.94	40.54	121.08	122.74	1482.66
10	17.07	16.99	49.49	101.49	102.16	1033.17

Table 3.8: 14 Node Network - Poisson traffic

As we see here, for Poisson traffic TM and EM performs better depending on the type of model. LP is worse for all the 14-node network models.

3.3.4 Normal Traffic

Here 10 different 14-node networks are considered with normal traffic types. The results of this analysis are presented below:

MODEL	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	19.06	19	61.19	194.42	192.86	1464.89
2	18.49	18.47	60.89	159.73	159.68	800.82
3	16.69	16.77	41.89	174.61	174.50	788.79
4	18.03	18.1	43.31	141.02	141.48	331.37
5	17.28	17.17	73.15	132.98	132.69	650.50
6	17.95	17.89	51.46	209.15	208.31	858.88
7	17.89	17.76	31.46	159.59	159.32	175.38
8	18.62	18.65	48.79	158.95	158.35	576.81
9	18.64	18.69	76.01	199.33	201.04	977.52
10	18.09	18.13	65.91	202.01	197.75	1198.95

Table 3.9: 14 Node Network - Normal traffic

As in Poisson traffic, TM and EM perform better depending on the type of model. LP is worse for all the 14-node network models.

3.3.5 Bimodal Traffic

Here 10 different 14-node networks are considered with bimodal traffic types. The results of this analysis are presented below:

MODEL	AVERAGE ERROR			MAXIMUM ERROR		
	TM	EM	LP	TM	EM	LP
1	63.21	64.8	55.9	1116.67	1079.63	1705.21
2	64.53	66.41	133.26	1082.10	1047.14	1582.42
3	63.93	64.31	107.32	582.96	550.09	2054.67
4	64.09	65.91	80.68	547.18	518.77	1242.30
5	62.72	63.63	85.93	627.69	602.82	1249.53
6	58.55	60.89	119.91	604.93	543.69	1269.28
7	61.09	63.1	104.81	999.94	931.35	7287.98
8	60.64	63.5	84.36	1245.37	1191.36	2424.07
9	61.83	64.16	79.87	1003.26	1014.16	1650.74
10	57.5	60.41	96.21	1468.37	1252.35	1093.10

Table 3.10: 14 Node Network - Bimodal traffic

TM shows better results compared to the other two for all the models expect model 1 for bimodal traffic.

3.3.6 Summary

For large networks, we observe that TM and EM perform equally well for different traffic types, whereas LP performs worse compared to the other two for all traffic types. The performance of the three methods for large networks is contrary to what happened with the smaller networks.

Chapter 4

New Directions for Traffic Matrix Estimation: *LP with Additional Constraints*

4.1 Improving Linear Programming by Incorporating Additional Network-Specific Information

From the comparative study in Chapter 3, we observe that LP performs better when compared to TM and EM for smaller networks, irrespective of the assumption made on the traffic type distribution. But this trend does not follow in case of large networks. On contrary, for large networks the performance of LP is worse compared to TM and EM for all traffic type distributions. The reason for this behaviour of LP is because only core backbone link constraints were included, which gives limited number of equations for solving unknown traffic matrix elements. For a network with n nodes, there are n (n-1) unknown traffic matrix elements. However, the number of links, m, among these nodes in a network ranges from O (n) to O (n^2); in general it is O (n). Hence, we need to estimate n (n-1) traffic matrix elements from m link load measurements. The difficulty here is the number of unknowns is usually far greater than the number of available equations. In case of small networks, we have sufficient number of independent equations to solve the unknowns. Hence we get better results. But when we deal with large networks, the number of available independent equations is usually very much less compared to the total number of unknowns, which gives less space for solving the unknown traffic matrix elements. Hence, we get worse results when we use LP for large networks.

To overcome this problem, we try to increase the total number of independent equations by including a) non-core backbone link constraints (section 4.2) and b) node flow conservation constraints (section 4.3). The following figure 4.1 illustrates the terminology used here.

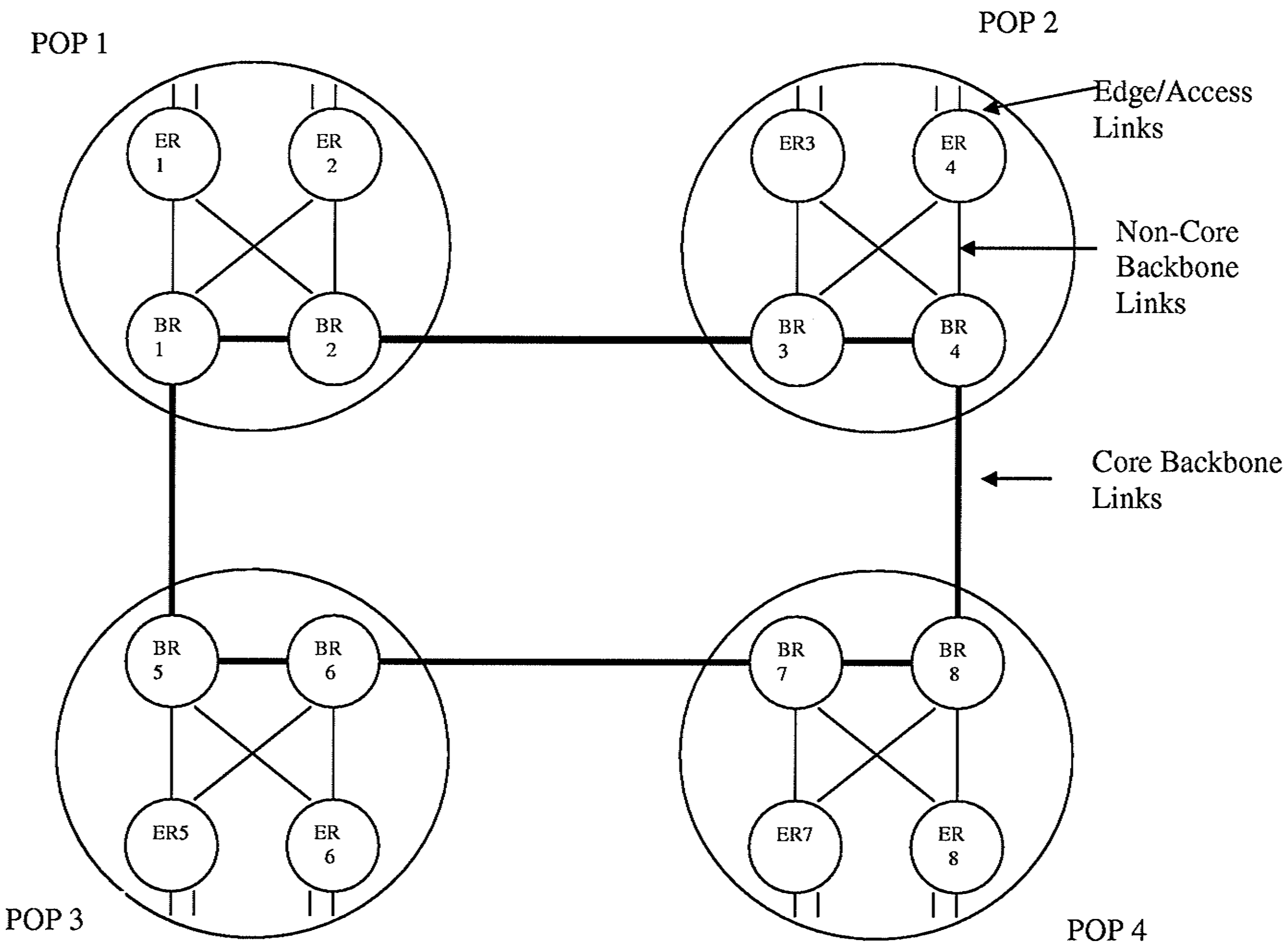

Figure 4.1: Simple Network Topology

Above figure shows four different POPs. The backbone routers that lie interior to the network are connected by core backbone links. The non-core backbone links connect the edge routers and the backbone routers. The edge/access links connect customers to the network.

4.2 Linear Programming with Non-Core Backbone Link Constraints (LP_with_NCLC)

Initially, only core backbone link constraints were included in the LP method for solving the traffic matrix elements. Since, LP shows poor performance due to the limited number of equations, we try to improve it by including non-core backbone link constraints as defined below:

$$\gamma_{k,i} = \sum_{j=1}^{N} \lambda_i p_{ij} r_{ijk} ;1 \leq k \leq L$$

where,

$\gamma_{k,i}$ - total measured traffic on non-core backbone link k in node i.

λ_i - total measured traffic entering the network via node i.

p_{ij} - proportion of traffic from node i to node j.

r_{ijk} - proportion of traffic originating at source node i and destined to node j that flows through non-core backbone link k.

L – total non-core backbone links in the network.

For example, the total traffic flow on non-core backbone link 1 (ER1 → BR1) in POP 1 of Figure 4.1 is calculated as follows:

$$\gamma_{1,1} = \lambda_1 * p_{11} * r_{111} + \lambda_1 * p_{12} * r_{121} + \lambda_1 * p_{13} * r_{131} + \lambda_1 * p_{14} * r_{141}$$

Assuming shortest path routing, the routing parameter $r_{111}, r_{121}, r_{131}, r_{141}$ takes values 0,0,1 and 0,5 respectively.

We tested this new approach for both small and large networks. Following section present the results of experiments performed.

4.2.1 4 Node Network (Constant traffic type)

Here, we consider 10 different 4-node networks with constant traffic type distribution. The table below shows the average and maximum error obtained when we estimated the traffic matrix with LP and LP_with_NCLC methods.

MODEL	AVERAGE ERROR		MAXIMUM ERROR	
	LP	LP_with_NCLC	LP	LP_with_NCLC
1	7.33	0.00	27.37	0.00
2	11.43	0.00	37.67	0.00
3	0.00	0.00	0.00	0.00
4	31.85	28.32	134.00	99.24

5	29.31	1.21	117.38	3.42
6	5.14	2.85	11.44	6.35
7	31.42	28.57	135.15	104.22
8	4.91	0.00	12.52	0.00
9	31.96	30.64	153.98	141.35
10	31.96	30.27	153.98	140.03

Table 4.1: Comparing LP and LP_with_NCLC for 4 Node Network (Constant traffic)

It is interesting to note from the above results that by including additional non-core backbone link constraints, LP_with_NCLC yields better performance for all the models compared to LP. Also, it gives 0% error for some models.

4.2.2 4 Node Network - Different traffic types

In the above section, we compared LP and LP_with_NCLC for 10 different 4-node network models assuming same constant traffic for all of them. To observe the behaviour of LP_with_NCLC for other traffic type distributions (Poisson, normal, uniform and bimodal), we assumed different traffic types for different models and performed the testing. The results of this experiment are presented below:

MODEL	AVERAGE ERROR		MAXIMUM ERROR	
	LP	LP_with_NCLC	LP	LP_with_NCLC
1 (Uniform)	50.53	0.00	99.98	0.00
2 (Poisson)	8.94	0.00	23.1	0.00

3 (Normal)	0.00	0.00	0	0.00
4 (Bimodal)	32.38	19.20	115.59	79.10
5 (Constant)	29.31	1.21	117.38	3.42
6 (Uniform)	45.97	42.50	103.4	105.40
7 (Poisson)	12.37	10.50	33.95	32.50
8 (Normal)	25.20	0.00	65.14	0.00
9 (Bimodal)	56.26	5.30	230.14	17.00
10 (Bimodal)	21.17	10.90	65.93	32.70

Table 4.2: Comparing LP and LP_with_NCLC for 4 Node Network (Different traffic)

From the above table, we conclude that for small networks LP_with_NCLC performs better when compared to the basic LP irrespective of the network model and traffic type. Also, it yields 0% error for all traffic type distributions and not only for constant traffic. This adds the benefit of using this method for estimating exact traffic matrix of any small network without worrying about the nature of traffic distribution type.

4.2.3 14 Node Network

Since real IP networks are large in size, one of our main concerns is how to check whether increasing the total number of equations would help to improve traffic matrix estimation for large network models. Hence, we compare LP and LP_with_NCLC for 14-node networks with different traffic types and the results are presented below:

a) Constant Traffic

Here, we consider 10 different 14-node networks with constant traffic. The results of this analysis are presented below:

MODEL	AVERAGE ERROR		MAXIMUM ERROR	
	LP	LP_with_NCLC	LP	LP_with_NCLC
1	13.22	8.21	242.65	54.02
2	18.47	12.75	99.96	99.25
3	26.28	21.19	106.43	162.12
4	37.04	19.01	145.14	100.00
5	19.27	15.14	99.96	98.73
6	48.88	14.45	686.17	78.37
7	52.02	10.63	431.43	90.21
8	33.15	27.34	146.37	476.89
9	26.75	22.09	124.57	167.62
10	49.9	34.98	463.06	297.83

Table 4.3: Comparing LP and LP_with_NCLC for 14 Node Network (Constant traffic)

b) Uniform Traffic

Here, we consider 10 different 14-node networks with uniform traffic. The results of this analysis are presented below:

MODEL	AVERAGE ERROR		MAXIMUM ERROR	
	LP	LP_with_NCLC	LP	LP_with_NCLC
1	111.20	40.28	654.08	321.39
2	79.15	28.18	593.61	359.75
3	59.35	32.16	781.89	269.51
4	93.72	34.99	1289.47	152.67
5	51.31	32.19	422.42	140.36
6	68.75	32.92	537.02	122.41
7	71.58	24.66	762.94	291.29
8	44.15	28.87	750.51	391.80
9	49.05	75.69	1847.42	449.22
10	67.06	44.52	2134.87	574.15

Table 4.4: Comparing LP and LP_with_NCLC for 14 Node Network (Uniform traffic)

c) Poisson Traffic

Here, we consider 10 different 14-node networks with Poisson traffic. The results of this analysis are presented below:

MODEL	AVERAGE ERROR		MAXIMUM ERROR	
	LP	LP_with_NCLC	LP	LP_with_NCLC
1	63.04	25.73	860.62	388.99
2	44.86	18.32	882.08	359.75
3	51.86	26.55	821.45	269.51
4	56.69	25.33	626.95	152.67
5	30.29	19.08	229.74	140.36
6	56.3	18.63	493.41	122.41
7	26.9	24.99	328.01	291.29

8	38.09	36.66	446.69	391.80
9	40.54	24.57	1482.66	449.22
10	49.49	31.10	1033.17	574.15

Table 4.5: Comparing LP and LP_with_NCLC 14 Node Network

(Poisson traffic)

d) Normal Traffic

Here, we consider 10 different 14-node networks with normal traffic. The results of this analysis are presented below:

MODEL	AVERAGE ERROR		MAXIMUM ERROR	
	LP	LP_with_NCLC	LP	LP_with_NCLC
1	61.19	35.18	1464.89	388.99
2	60.89	28.18	800.82	359.75
3	41.89	32.16	788.79	269.51
4	43.31	34.99	331.37	152.67
5	73.15	32.19	650.50	140.36
6	51.46	32.92	858.88	122.41
7	31.46	24.66	175.38	291.29
8	48.79	28.87	576.81	391.80
9	76.01	75.69	977.52	449.22
10	65.91	44.52	1198.95	574.15

Table 4.6: Comparing LP and LP_with_NCLC for 14 Node Network

(Normal traffic)

e) Bimodal Traffic

Here, we consider 10 different 14-node networks with bimodal traffic. The results of this analysis are presented below:

MODEL	AVERAGE ERROR		MAXIMUM ERROR	
	LP	LP_with_NCLC	LP	LP_with_NCLC
1	55.9	51.12	1705.21	1136.99
2	133.26	28.18	1582.42	359.75
3	107.32	32.16	2054.67	269.51
4	80.68	34.99	1242.30	152.67
5	85.93	32.19	1249.53	140.36
6	119.91	32.92	1269.28	122.41
7	104.81	24.66	7287.98	291.29
8	84.36	28.87	2424.07	391.80
9	79.87	75.69	1650.74	449.22
10	96.21	44.52	1093.10	574.15

Table 4.7: Comparing LP and LP_with_NCLC for 14 Node Network (Bimodal traffic)

4.2.4 Summary

As we see from Tables 4.3 to 4.7, LP_with_NCLC provides better performance compared to basic LP for various large network models and traffic types. These experiments clearly indicate that the reason for poor performance of Linear Programming is due to insufficient number of equations available for solving the total unknown traffic matrix elements. Hence, we focussed on improving the performance of Linear Programming further by including node flow conservation constraints in addition to non-core backbone links constraints. This is explained in next section.

4.3 Linear Programming with Node Flow Conservation Constraints (LP_with_NFCC)

The node flow conservation set of constraints algebraically states that the sum of the flow through arcs directed towards a node equals the sum of the flow through arcs directed away from that node. This can be represented as below for node j:

$$\sum_i x_{ji} - \sum_i x_{ij} = 0, \forall j$$

The above set of constraint holds good only for transhipment node. We can use the similar convention to write the flow conservation equation at the source or sink node:

$$\sum_i x_{ji} - \sum_i x_{ij} = b_j, \forall j$$

b_j is a positive constant for source node and negative constant for sink node. We include these flow conservation constraints in LP_with_NCLC method to test whether it helps in improving the accuracy further. The results for the 14-node topology are given below.

4.3.1 14 - Node Network

a) Constant Traffic

Here, we consider 10 different 14-node networks with constant traffic. The results of this analysis are presented below:

Model	AVERAGE ERROR			MAXIMUM ERROR		
	LP	LP_with_NCLC	LP_with_NFCC	LP	LP_with_NCLC	LP_with_NFCC
1	13.22	8.21	7.55	242.65	54.02	55.86
2	18.47	12.75	10.07	99.96	99.25	99.61
3	26.28	21.19	12.18	106.43	162.12	142.91
4	37.04	19.01	14.35	145.14	100.00	109.96
5	19.27	15.14	11.43	99.96	98.73	87.12
6	48.88	14.45	11.99	686.17	78.37	81.75
7	52.02	10.63	7.89	431.43	90.21	87.01
8	33.15	27.34	17.69	146.37	476.89	215.77
9	26.75	22.09	15.68	124.57	167.62	150.42
10	49.9	34.98	20.16	463.06	297.83	209.31

Table 4.8: Comparing LP, LP_with_NCLC & LP_with_NFCC (Constant traffic)

b) Poisson Traffic

Here, we consider 10 different 14-node networks with Poisson traffic. The results of this analysis are presented below:

Model	AVERAGE ERROR			MAXIMUM ERROR		
	LP	LP_with_NCLC	LP_with_NFCC	LP	LP_with_NCLC	LP_with_NFCC
1	63.04	25.73	22.37	860.62	388.99	228.14
2	44.86	18.32	19.45	882.08	359.75	376.19
3	51.86	26.55	26.88	821.45	269.51	185.07
4	56.69	25.33	26.02	626.95	152.67	162.28
5	30.29	19.08	18.01	229.74	140.36	114.81
6	56.3	18.63	18.71	493.41	122.41	129.04
7	26.9	24.99	26.71	328.01	291.29	366.90
8	38.09	36.66	26.52	446.69	391.80	247.13
9	40.54	24.57	23.66	1482.66	449.22	420.23
10	49.49	31.10	31.32	1033.17	574.15	561.63

Table 4.9: Comparing LP, LP_with_NCLC & LP_with_NFCC (Poisson traffic)

c) Normal Traffic

Here, we consider 10 different 14-node networks with normal traffic. The results of this analysis are presented below:

Model	AVERAGE ERROR			MAXIMUM ERROR		
	LP	LP_with_ NCLC	LP_with _NFCC	LP	LP_with_NCLC	LP_with_NFCC
1	61.19	35.18	30.23	1464.89	388.99	556.45
2	60.89	28.18	23.67	800.82	359.75	302.67
3	41.89	32.16	34.30	788.79	269.51	298.87
4	43.31	34.99	26.99	331.37	152.67	379.98
5	73.15	32.19	30.80	650.50	140.36	170.49
6	51.46	32.92	30.78	858.88	122.41	396.71
7	31.46	24.66	21.86	175.38	291.29	100.88
8	48.79	28.87	28.49	576.81	391.80	1221.75
9	76.01	75.69	37.50	977.52	449.22	288.70
10	65.91	44.52	39.21	1198.95	574.15	182.36

Table 4.10: Comparing LP, LP_with_NCLC & LP_with_NFCC

(Normal traffic)

d) Bimodal Traffic

Here, we consider 10 different 14-node networks with bimodal traffic. The results of this analysis are presented below:

Model	AVERAGE ERROR			MAXIMUM ERROR		
	LP	LP_with_ NCLC	LP_with _NFCC	LP	LP_with_NCLC	LP_with_NFCC
1	55.90	51.12	50.52	1705.21	1136.99	728.33

2	133.26	28.18	42.15	1582.42	359.75	1035.36
3	107.32	32.16	45.25	2054.67	269.51	704.60
4	80.68	34.99	48.65	1242.30	152.67	450.95
5	85.93	32.19	55.91	1249.53	140.36	492.76
6	119.91	32.92	50.49	1269.28	122.41	610.64
7	104.81	24.66	90.92	7287.98	291.29	5336.70
8	84.36	28.87	56.88	2424.07	391.80	1472.25
9	79.87	75.69	59.92	1650.74	449.22	930.94
10	96.21	44.52	44.05	1093.10	574.15	759.86

Table 4.11: Comparing LP, LP_with_NCLC & LP_with_NFCC (Bimodal traffic)

e) Uniform Traffic

Here, we consider 10 different 14-node networks with uniform traffic. The results of this analysis are presented below:

Model	AVERAGE ERROR			MAXIMUM ERROR		
	LP	LP_with_NCLC	LP_with_NFCC	LP	LP_with_NCLC	LP_with_NFCC
1	111.20	40.28	34.62	654.08	321.39	297.81
2	79.15	28.18	33.25	593.61	359.75	332.64
3	59.35	32.16	34.30	781.89	269.51	305.10
4	93.72	34.99	28.94	1289.47	152.67	265.26
5	51.31	32.19	45.23	422.42	140.36	396.75
6	68.75	32.92	33.52	537.02	122.41	305.65

7	71.58	24.66	29.67	762.94	291.29	281.03
8	44.15	28.87	34.33	750.51	391.80	333.72
9	49.05	75.69	51.55	1847.42	449.22	1385.40
10	67.06	44.52	46.75	2134.87	574.15	1443.41

Table 4.12: *Comparing LP, LP_with_NCLC & LP_with_NFCC*

(Uniform traffic)

4.3.2 Summary

For constant and normal traffic, adding node flow conservation constraints improves the performance further compared to LP_with_NCLC for all models. However, it does not follow the same trend for other traffic types. For Poisson, bimodal and uniform traffic types the performance of LP_with_NFCC depends on the network model. This gives the choice for the network operators to choose between LP_with_NCLC and LP_with_NFCC estimation techniques depending on the traffic distribution type in their network.

Chapter 5

New Directions for Traffic Matrix Estimation: *Hybrid Model*

5.1 Introduction

We develop a hybrid model that combines Tomogravity, Entropy Maximization and Linear Programming methods targeting for better traffic matrix estimation. We formulate a quadratic programming (QP) problem, to estimate the proportional values [a_t, a_e, a_l] that minimize the difference between estimated and actual traffic matrix elements. We use a_t, a_e and a_l to denote the optimal estimate that is obtained for the proportion of values to be selected from Tomogravity, Entropy Maximization and Linear Programming methods respectively. We denote total number of models and source-destination pairs by M and S, respectively. We use $T^t_{m,s}, T^e_{m,s}, T^l_{m,s}$ to denote the traffic matrix elements obtained by using Tomogravity, Entropy Maximization and Linear Programming methods corresponding to $\{1 \le m \le M, 1 \le s \le S\}$. We use T (m,s) to denote the actual traffic matrix elements. The formulated quadratic programming problem is given below:

QP Problem

To find [a_t, a_e, a_l] such that we minimize Z subject to the constraints given below:

Minimize Z,

$$Z = \sum_{m=1}^{M} \sum_{s=1}^{S} (a_t T_{m,s}^{t} + a_e T_{m,s}^{e} + a_l T_{m,s}^{l} - T(m,s))^2 \qquad (5.1)$$

Subject to:

$$\begin{aligned} &a_t + a_e + a_l = 1 \\ &0 \le a_t \le 1 \\ &0 \le a_e \le 1 \\ &0 \le a_l \le 1 \end{aligned} \qquad (5.2)$$

Our basic idea is to estimate the optimal proportional values a_t, a_e, a_l for 4-node network using hybrid model. On obtaining these values, we apply the same proportional values to 14-node network and compare the estimated traffic matrix with original traffic matrix.

5.2 4-Node Network Model

5.2.1 Same Type of Traffic (case 1)

Here, we consider 10 different 4-node networks with same constant traffic. Applying the QP problem, we obtain the following proportional values.

$$a_t = 0.1194$$

$$a_e = 0.0 \tag{5.3}$$

$$a_l = 0.8806$$

Table 5.1 shows the comparison of hybrid model with the other three methods. As we notice, hybrid model that combines the three techniques using the proportional values shown in equation 5.3 performs better compared to the individual three for some models.

MODEL	AVERAGE ERROR				MAXIMUM ERROR			
	TM	EM	LP	HYBRID (case 1)	TM	EM	LP	HYBRID (case 1)
1	35.01	43.24	7.33	8.91	221.9	226.85	27.37	50.59
2	46	46.66	11.43	11.91	146.55	190.41	37.67	50.67
3	0	0	0	0	0	0	0	0
4	36.43	36.68	31.85	30.91	126.95	155.39	134	133.16
5	31.22	36.19	29.31	28.41	165.4	170.33	117.38	123.12
6	8.4	3	5.14	3.52	25.2	9.19	11.44	8.67

7	43.46	45.17	31.42	29.98	227.15	230.13	135.15	146.13
8	28.05	7.93	4.91	2.91	93.25	29.81	12.52	9.03
9	38.87	45.09	31.96	32.49	205.35	231.53	153.98	160.11
10	38.08	37.79	31.96	32.12	140.95	168.24	153.98	152.42

Table 5.1: Hybrid Model - 4 node network (case 1)

5.2.2 Different Types of Traffic (case 2)

In section 5.2.1, we tested the hybrid model assuming the same type of traffic for all the models and obtained the values of a_t, a_e, a_l *(shown in equation 5.3)*. Though it shows better performance compared to TM, EM and LP for most of the models, we continued the testing to verify whether assuming different traffic types for different models will improve the performance further. To check this, here we consider 10 different 4-node networks with different traffic types and obtained the following proportional values:

$$a_t = 0.1054$$

$$a_e = 0.0 \qquad (5.4)$$

$$a_l = 0.8946$$

Comparing the proportional values in equations 5.3 and 5.4, we can see that there is not much difference among them. We can understand that hybrid model selects larger proportion from the method that performs better for most of the models and smaller proportion from rest of the methods. For 4-node networks, since LP performs better compared to TM & EM, the value of a_l is larger (approx. 90%) compared to a_e and a_t (approx 10%).

Table 5.2 shows the comparison of hybrid model (case 2) with the other three methods. Similar to case 1, the hybrid model performs better for some models and LP for few other models.

MODEL	AVERAGE ERROR				MAXIMUM ERROR			
	TM	EM	LP	HYBRID (case 2)	TM	EM	LP	HYBRID (case 2)
1 (Uniform)	19.82	2.33	50.53	43.77	60.71	7.02	99.98	90.79
2 (Poisson)	20.94	11.04	9	9.26	43.26	30.57	23.74	20.55
3 (Normal)	0	0	0	0	0	0	0	0
4 (Bimodal)	25.83	20.23	21.16	17.86	88.49	53.46	50.03	43.53
5 (Constant)	31.22	36.19	29.31	28.52	165.4	170.33	117.38	122.44
6 (Uniform)	10.5	5.57	4.14	4.81	33.27	15.48	9.28	8.86
7 (Poisson)	28.15	30.7	26.66	23.35	94.49	99.17	78.27	70.48
8 (Normal)	42.49	39.68	19.89	22.01	203.76	180.91	72.64	86.46
9 (Bimodal)	36.03	22.94	15.83	17.24	90.86	42.96	41.99	43.01
10 (Bimodal)	97.42	72.61	51.07	55.45	615.75	441.84	205.65	248.87

Table 5.2: Hybrid Model - 4 node network (case 2)

Our basic idea in testing the hybrid model for two different cases, one assuming the same type of traffic for all models and the other assuming different types of traffic for different models is to select the proportional values from either one which outperforms the other. But, from the analysis performed in the above sections we notice that hybrid model in case 1 performs better for some models and hybrid model in case 2 for rest of them. Since, we did not see any significant difference between the proportional values obtained in case 1 (*equation 5.3*) and case 2 (*equation 5.4*), we continue testing the performance of hybrid model for large networks by taking the values from equation 5.3.

5.3 14-Node Network Model

Here, we combine TM, EM and LP for 14-node network with the proportional values obtained for 4-node network (*equation 5.3*). But as we see in Table 5.3, the error of the hybrid model *(HYBRID - Eq 5.3 in table 5.3)* is high compared to TM and EM. This is because, for smaller networks LP performs better for most of the cases and hence hybrid model chooses higher proportion of value from LP with a_l as 0.8806 and lower proportion from TM with a_t as 0.1194. When we directly apply these values for a 14-node network, the result of hybrid model is worse since LP does not perform well for large networks. This shows that we cannot apply the proportional values a_t, a_e and a_l obtained from smaller networks to large networks directly.

So, we tested the hybrid model for different 14-node networks and obtained the following values:

$$a_t = 0.403$$

$$a_e = 0.592 \tag{5.5}$$

$$a_l = 0.005$$

From the above result, we see that hybrid model chooses very small proportional value from LP since it performs worse compared to the other two for larger networks. In this case the result of hybrid model *(HYBRID - Eq 5.5 in table 5.3)* is better compared to TM, EM and LP.

AVERAGE ERROR					MAXIMUM ERROR				
TM	EM	LP	HYBRID (Eq. 5.3)	HYBRID (Eq. 5.5)	TM	EM	LP	HYBRID (Eq. 5.3)	HYBRID (Eq 5.5)
21.83	21.76	36.43	33.54	21.63	102.20	98.95	410.21	368.48	100.30

Table 5.3: Hybrid Model - 14 node network

5.4 Combining TM, EM and LP for 14-Node Network Models

5.4.1 Poisson Traffic

Here, we obtained the proportional values by combining 10 different 14-node networks assuming Poisson traffic types. The table below shows the comparison of hybrid model with the other 3 methods. As we see, for the Poisson traffic types hybrid model provides better performance compared to the other 3 methods irrespective of the model type.

Model	AVERAGE ERROR				MAXIMUM ERROR			
	TM	EM	LP	HYBRID	TM	EM	LP	HYBRID
1	16.88	16.90	22.37	16.85	122.59	121.14	228.14	119.81
2	16.40	16.34	19.45	15.72	144.23	145.70	376.19	134.00
3	15.90	15.77	26.88	15.46	141.86	145.72	185.07	128.76
4	15.78	15.89	26.02	15.61	127.36	131.66	162.28	131.45
5	17.13	17.10	18.01	15.55	116.48	120.90	114.81	95.65
6	16.10	16.24	18.71	14.98	131.31	129.61	129.04	125.24
7	16.18	16.02	26.71	15.20	124.86	124.79	366.90	121.70
8	17.48	17.58	26.52	16.92	133.93	133.80	247.13	121.47
9	16.73	16.94	23.66	15.60	121.08	122.74	420.23	124.29
10	17.07	16.99	31.32	17.29	101.49	102.16	561.63	180.60

Table 5.4: Hybrid Model - 14 Node Network (Poisson Traffic)

5.4.2 Constant Traffic

Here, we obtained the proportional values by combining 10 different 14-node networks assuming constant traffic types. The table below shows the comparison of the hybrid model with the other 3 methods. As we see, the hybrid model provides better performance compared to the other 3 methods irrespective of the model type.

Model	AVERAGE ERROR				MAXIMUM ERROR			
	TM	EM	LP	HYBRID	TM	EM	LP	HYBRID
1	1.92	1.93	7.55	1.90	4.67	4.66	55.86	4.57
2	1.82	1.82	10.07	1.78	4.81	4.80	99.61	4.66
3	1.78	1.78	12.18	1.77	4.97	4.93	142.91	4.35
4	1.70	1.70	14.35	1.69	4.55	4.57	109.96	4.53
5	1.70	1.71	11.43	1.69	4.71	4.69	87.12	4.68
6	1.84	1.85	11.99	1.81	5.42	5.38	81.75	5.03
7	1.95	1.96	7.89	1.93	5.17	5.19	87.01	5.02
8	1.91	1.92	17.69	1.90	6.04	5.97	215.77	5.88
9	1.87	1.88	15.68	1.83	5.11	5.08	150.42	5.00
10	1.83	1.84	20.16	1.82	5.76	5.72	209.31	5.72

Table 5.5: Hybrid Model - 14 Node Network (Constant Traffic)

5.4.3 Normal Traffic

Here, we obtained the proportional values by combining 10 different 14-node networks assuming normal traffic types. The table below shows the comparison of the hybrid model with the other 3 methods. As we see, the hybrid model provides better performance compared to the other 3 methods irrespective of the model type.

Model	AVERAGE ERROR				MAXIMUM ERROR			
	TM	EM	LP	HYBRID	TM	EM	LP	HYBRID
1	19.06	19.00	30.23	19.20	194.42	192.86	556.45	68.45
2	18.49	18.47	23.67	17.65	159.73	159.68	302.67	64.81
3	16.69	16.77	34.30	16.34	174.61	174.50	298.87	54.09
4	18.03	18.10	26.99	17.07	141.02	141.48	379.98	66.99
5	17.28	17.17	30.80	16.72	132.98	132.69	170.49	63.55
6	17.95	17.89	30.78	17.76	209.15	208.31	396.71	68.48
7	17.89	17.76	21.86	16.69	159.59	159.32	100.88	52.58
8	18.62	18.65	28.49	17.65	158.95	158.35	1221.75	69.38
9	18.64	18.69	37.50	18.57	199.33	201.04	288.70	54.70
10	18.09	18.13	39.21	17.93	202.01	197.75	182.36	102.92

Table 5.6: Hybrid Model - 14 Node Network (Normal Traffic)

5.4.4 Uniform Traffic

Here, we obtained the proportional values by combining 10 different 14-node networks assuming uniform traffic types. The table below shows the comparison of the hybrid model with the other 3 methods. As we see, the hybrid model provides better performance compared to the other 3 methods irrespective of the model type.

Model	AVERAGE ERROR				MAXIMUM ERROR			
	TM	EM	LP	HYBRID	TM	EM	LP	HYBRID
1	34.07	33.94	46.42	21.75	176.72	177.11	410.21	99.96
2	32.68	32.64	42.71	22.27	168.51	168.89	290.63	89.15
3	32.52	32.59	63.77	21.76	169.99	171.61	1268.65	89.39
4	30.19	29.96	76.08	21.04	179.19	181.31	1416.42	92.77
5	31.29	31.13	44.86	20.26	223.78	223.05	432.24	83.02
6	30.02	30.18	44.16	20.41	159.63	158.41	514.81	102.37
7	31.67	31.69	53.76	20.90	165.76	169.15	567.98	97.69
8	33.64	33.68	50.00	22.66	171.60	174.44	317.89	87.03
9	35.31	35.34	62.19	22.27	179.66	177.97	1349.56	77.72
10	32.28	32.36	83.20	22.19	202.06	200.26	943.57	75.72

Table 5.7: Hybrid Model - 14 Node Network (Uniform Traffic)

5.4.5 Bimodal Traffic

Here, we obtained the proportional values by combining 10 different 14-node networks assuming bimodal traffic types. The table below shows the comparison of the hybrid model with the other 3 methods.

Model	AVERAGE ERROR				MAXIMUM ERROR			
	TM	EM	LP	HYBRID	TM	EM	LP	HYBRID
1	63.21	64.80	51.12	51.01	1116.67	1079.63	1136.99	328.82
2	64.53	66.41	28.18	28.1	1082.10	1047.14	359.75	268.80
3	63.93	64.31	32.16	32.01	582.96	550.09	269.51	206.08
4	64.09	65.91	34.99	34.0	547.18	518.77	152.67	159.67
5	62.72	63.63	32.19	32.1	627.69	602.82	140.36	172.34
6	58.55	60.89	32.92	32.9	604.93	543.69	122.41	121.86
7	61.09	63.10	24.66	24.32	999.94	931.35	291.29	208.67
8	60.64	63.50	28.87	28.77	1245.37	1191.36	391.80	320.83
9	61.83	64.16	75.69	52.11	1003.26	1014.16	449.22	331.66
10	57.50	60.41	44.52	43.3	1468.37	1252.35	574.15	304.79

Table 5.8: Hybrid Model - 14 Node Network (Bimodal Traffic)

5.5 Summary

A hybrid model that combines the three traffic matrix estimation techniques provides better performance compared to TM, EM and LP irrespective of the network model. Also, the hybrid model does not show any dependency on the nature of traffic distribution.

Chapter 6

Using Hierarchical Models for Assessing the Quality of Traffic Matrix Estimation for Real Life Networks

6.1. Introduction

ISPs frequently face the need to expand or modify their existing network to meet the requirements of new traffic demands. Such modification or expansion may involve adding or removing links and switches, rerouting existing traffic and changing network topology. The knowledge of traffic matrix is necessary for ISPs to perform such modifications to their existing network. This knowledge is also important to perform a wide variety of traffic engineering tasks such as load balancing, capacity and business planning. However, inputs required for direct measurement of traffic matrix is not available in large IP networks. On the other hand link load measurements, network routing and configuration data are readily available and traffic matrix can be estimated from these available data.

For a network with n nodes and m links, there are $c = n\ (n-1)$ unknown traffic matrix elements. Hence, we need to estimate c traffic matrix elements from m link load measurements. The difficulty here is the number of unknowns is far greater than the number of available equations which leads to an under-determined problem.

There are several existing methods to estimate traffic matrix from link load data. Among them, we selected three techniques namely Tomogravity, Entropy Maximization and Linear Programming, and performed comparative study among them. The experimental results of this comparative study presented in Chapter 3 were based on testing over hypothetical data for which "real" traffic matrix was known.

However in reality, measuring the "real" traffic matrix is not economically feasible in large IP networks. Therefore, we do not have any real traffic matrix against which we can compare the results obtained using traffic matrix estimation techniques. This makes assessing the quality of traffic matrix estimation techniques for real life large networks difficult. To overcome this problem, we develop a Hierarchical model which helps us to check the quality of the results obtained using any traffic matrix estimation techniques.

Basically Hierarchical model consists of three steps: In the first step, we reduce any given large network into small network by grouping nodes. In the second step, we apply Linear Programming method to estimate the traffic matrix of any large network at coarse level. On extensive analysis of LP technique, we found that LP with additional constraints is able to estimate traffic matrix of small networks with very less error (for some network models it gives 0% error). Also, this optimization technique performs better compared to Tomogravity and Entropy Maximization for small networks. One other thing we have established is that as the network size gets smaller, the performance of LP for estimating traffic matrix improves. As a result of this step, we are able to estimate the traffic matrix of any large network at coarse level with much less error. In the third step, we estimate the traffic matrix of original large network using any of the existing traffic matrix estimation techniques. Then, we use the information obtained at coarse level about the network in previous step to check how good the results that are generated by any traffic matrix estimation techniques.

6.2. Hierarchical Model

6.2.1 Reducing a large network into a small network by grouping

We reduce any large network into a small network by grouping a set of nodes into a super-node. The link connecting the two different super-nodes represents the corridor between the nodes belonging to a pair of adjacent groups. Partitioning a large network into different groups is done by drawing a screen-line. This partitioning has to be carried out carefully in order to ensure that the resulting grouping is a reasonable representation. We explain the idea of grouping with a 4-node example. We will use "node" and "super-node" interchangeably when there is no possibility of confusion, otherwise we will distinguish them. We use the 4 node network of Figure 6.1 as an example.

Case 1: Grouping nodes {(1, 3) and (2, 4)}

Let us consider a network that consists of 4 nodes as shown in Figure 6.1. The links are bi-directional. Here, we shrink the 4-node graph into a 2-super-node graph by grouping nodes 1 and 3 as super-node A and nodes 2 and 4 as super-node B. The dotted line represents the screen-line which divides the network into two groups. The reduced 2-super-node graph is shown in Figure 6.2.

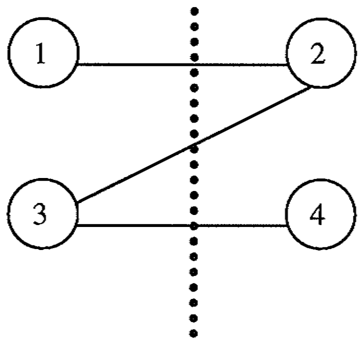

Figure 6.1:
4 - Node Graph

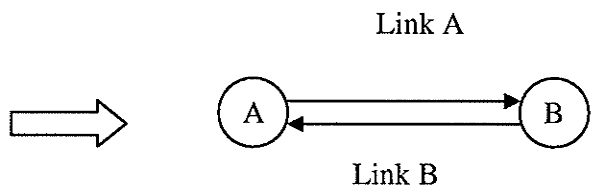

Figure 6.2:
2 - Super-Node Graph

Corridor A represents aggregated link which consists of links $L_{1\rightarrow 2}$, $L_{3\rightarrow 2}$ and $L_{3\rightarrow 4}$. Similarly, corridor B represents aggregated link which consists of links $L_{2\rightarrow 1}$, $L_{2\rightarrow 3}$ and $L_{4\rightarrow 3}$.

The link load measurements of corridor A and B are derived from Table 6.1.

Links	Link Load (Mbps)
1→2	15
2→1	22
2→3	20
3→2	21
3→4	17
4→3	13

Table 6.1: Link load measurements Of 4-node network

Links	Link Load (Mbps)
Link A	53
Link B	55

Table 6.2: Link load measurements of 2-super-node network

In the 2-super-node network, we have two unknown traffic matrix elements. They are traffic from super-node A to B and from super-node B to A. The link load measurement of corridor A (53 Mbps) tells us that the total traffic from super-node A to super-node B is 53 Mbps. Similarly, traffic on corridor B (55 Mbps) tells us that the total traffic from super-node B to super-node A is 55 Mbps. This gives us the following new constraints:

Sum of traffic (N1 → N2 + N1 → N4 + N3 → N2 + N3 → N4) = 53 Mbps

Sum of traffic (N2 → N1 + N2 → N3 + N4 → N1 + N4 → N3) = 55 Mbps → (1)

But, these estimated traffic matrix elements of 2-super-node network do not represent true values. The reason for the link load measurement in Table 6.2 not representing the actual traffic values is explained below:

Corridor A, which is a combination of links 1 → 2, 3 → 2 and 3 → 4 not only carry the traffic from super-node A to B, but also includes the traffic that is distributed within nodes in A and B and also those from the nodes in B to A. Due to this, we cannot certainly say that the total load on corridor A represents only the traffic from super-node A to super-node B. As stated above, out of 53 Mbps which is carried on corridor A only part of it is designated traffic from super-node A to super-node B. The remaining amount of traffic represents the traffic distributed within nodes in super-node A and B and also from super-node B to A.

Due to this problem, we cannot use the above traffic matrix elements obtained from the reduced 2- super-node network to evaluate the estimated traffic matrix of the 4-node network.

Case 2: Grouping nodes {(1, 2) and (3, 4)}

Here, we consider an alternative way of grouping 4-node network. The same 4 - node network is reduced into a 2- super-node network by grouping nodes 1 and 2 as super-node A and nodes 3 and 4 as super-node B. The grouping is shown in the following figures:

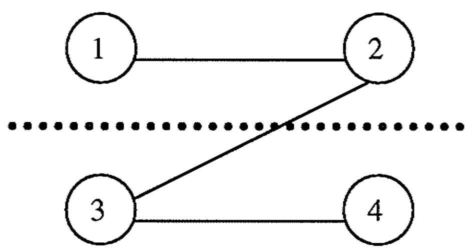

Figure 6.3:
4 - Node Graph

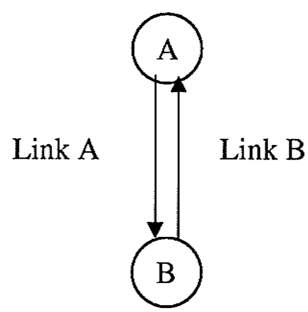

Figure 6.4:
2 - Super-Node Graph

Corridor A represents link $L_{2 \to 3}$ and Corridor B represents link $L_{3 \to 2}$. The link load measurements of corridor A and B can be derived from Table 6.3.

Links	Link Load (Mbps)
1-->2	15
2-->1	22
2-->3	20
3-->2	21
3-->4	17
4-->3	13

Table 6.3: Link Load measurements of 4-node network

Links	Link Load (Mbps)
Link A	20
Link B	21

Table 6.4: Link Load measurements of 2-Super-node network

The link load measurement of corridor A (20 Mbps) tells us that the total traffic from super-node A to super-node B is 20 Mbps. Similarly, traffic on corridor B (21 Mbps) tells us that the total traffic from super-node B to super-node A is 21 Mbps. This gives us the following new constraints:

Sum of traffic (N1 $\rightarrow$ N3 + N1 $\rightarrow$ N4 + N2 $\rightarrow$ N3 + N2 $\rightarrow$ N4) = 20 Mbps

Sum of traffic (N3 $\rightarrow$ N1 + N4 $\rightarrow$ N1 + N3 $\rightarrow$ N2 + N4 $\rightarrow$ N2) = 21 Mbps $\rightarrow$ (3)

This represents true values and we can use these values to check the quality of the results generated for 4-node network using any traffic matrix estimation techniques. The key difference in this way of grouping is that we drew the screen-line against the links which carry only the traffic passing between the two divided groups. For example, here corridor A and B carries only the traffic passing between super-node A and B and does not include any traffic that is distributed within nodes in super-node A and B.

Hence, one way to overcome the above discussed problem of grouping is by properly choosing the screen-line which shrinks the large network into small network. The screen-line partitioning a network into different group of nodes should be selected such that traffic passing between any two groups does not cross the screen-line more than once. However, in a practical situation it is difficult to create such "perfect" groupings. In such cases, we create a set of good groupings such that the amount of traffic crossing the screen line more than once is kept to a minimum. Then, we use this set to test real life data and pick the results that is dominant for most groupings. We illustrate this with a 10-node network example in section 6.2.2.

6.2.2 Obtaining the traffic matrix of large network at coarse level

Let us assume a network that consists of 10 nodes as shown in Figure 6.5. The links are bi-directional. We group the 10-node network into three different models such that the amount of traffic crossing the screen line more than once is kept to a minimum. In model1, we shrink the 10-node network into 3-node network by grouping nodes 1, 2, 3 and 4 as super-node A, nodes 5, 6 and 7 as super-node B and nodes 8, 9 and 10 as super-node C. The reduced 3-super-node graph is shown in Figure 6.6.

Here, corridor A → B represents aggregated link which consists of links $L_{1 \to 5}$ and $L_{4 \to 6}$, corridor B → A represents aggregated link which consists of links $L_{5 \to 1}$ and $L_{6 \to 4}$, corridor B → C represents aggregated link which consists of links $L_{5 \to 8}$ and $L_{6 \to 9}$, corridor C → B represents aggregated link which consists of links $L_{8 \to 5}$ and $L_{9 \to 6}$.

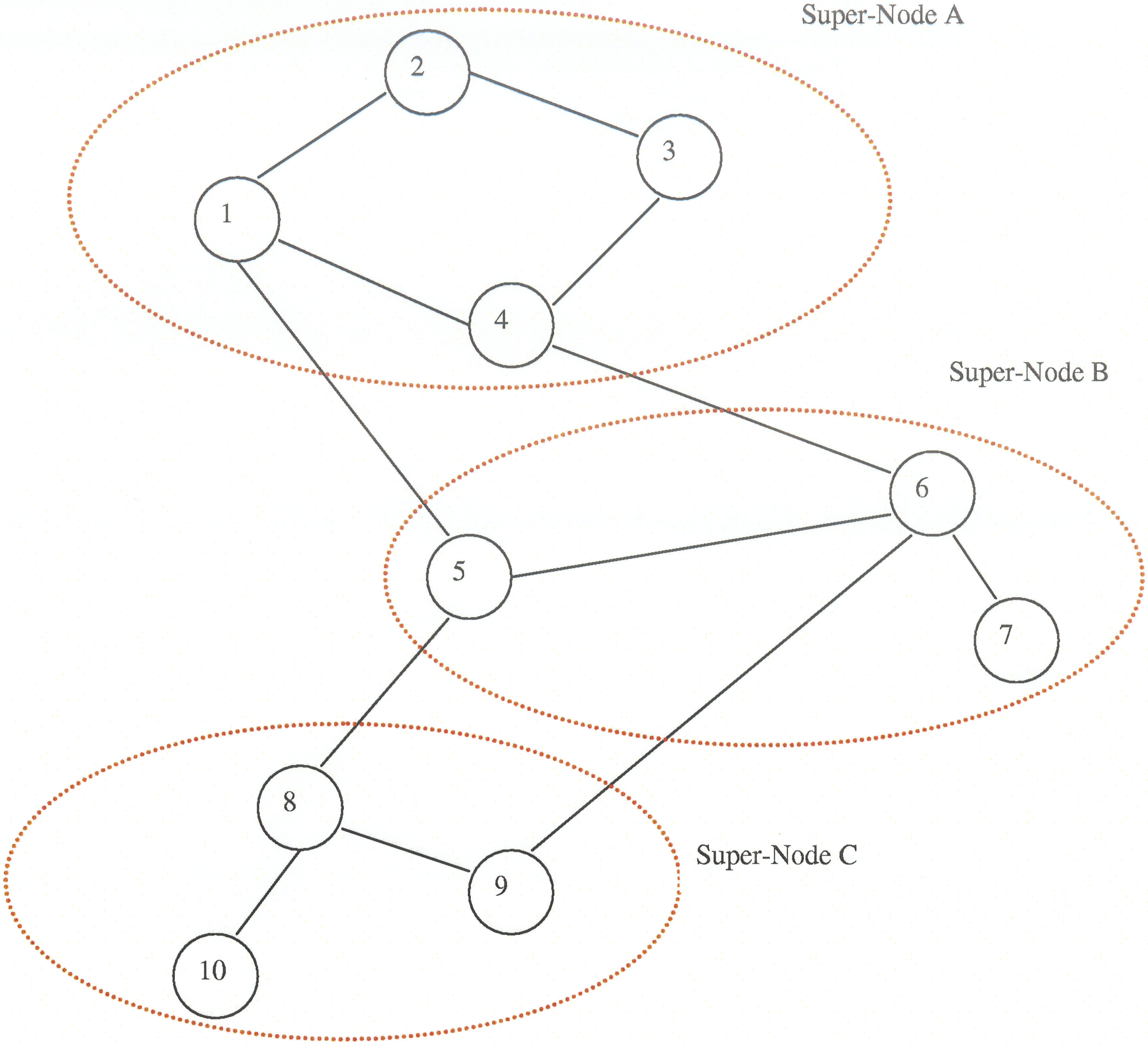

Figure 6.5: 10 – node Network

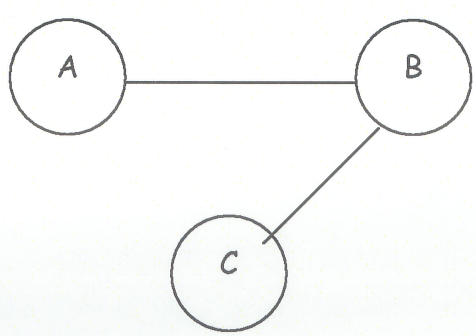

Figure 6.6: 3 – Super-node Network (Model 1)

In model2, we shrink the 10-node network into 3-super-node network by grouping nodes 1, 2, 3 and 5 as super-node A, nodes 4, 6 and 7 as super-node B and nodes 8, 9 and 10 as super-node C. The reduced 3-super-node graph is shown in Figure 6.7.

Here, corridor A → B represents aggregated link which consists of links $L_{1\rightarrow4}$, $L_{5\rightarrow6}$ and $L_{3\rightarrow4}$, corridor A → C represents link which consists of $L_{5\rightarrow8}$, corridor B → A represents aggregated link which consists of links $L_{4\rightarrow1}$, $L_{6\rightarrow5}$ and $L_{4\rightarrow3}$, corridor B → C represents link which consists of $L_{6\rightarrow9}$, corridor C → A represents link which consists of $L_{8\rightarrow5}$, corridor C → B represents link which consists of $L_{9\rightarrow6}$.

In model3, we shrink the 10-node network into 4-super-node network by grouping nodes 1, 2 and 3 as super-node A, nodes 4 and 5 as super-node B, nodes 6 and 7 as super-node C and nodes 8, 9 and 10 as super-node D. The reduced 4-super-node graph is shown in Figure 6.8.

Here, corridor A → B represents aggregated link which consists of links $L_{1\rightarrow5}$, $L_{1\rightarrow4}$ and $L_{3\rightarrow4}$, corridor B → A represents aggregated link which consists of links $L_{5\rightarrow1}$, $L_{4\rightarrow1}$ and $L_{4\rightarrow3}$, corridor B → C represents aggregated link which consists of links $L_{4\rightarrow6}$ and $L_{5\rightarrow6}$, corridor B → D represents link which consists of $L_{5\rightarrow8}$, corridor C → B represents aggregated link which consists of links $L_{6\rightarrow4}$ and $L_{6\rightarrow5}$, corridor C → D represents link which consists of $L_{6\rightarrow9}$, corridor D → B represents link which consists of $L_{8\rightarrow5}$, corridor D → C represents link which consists of $L_{9\rightarrow6}$.

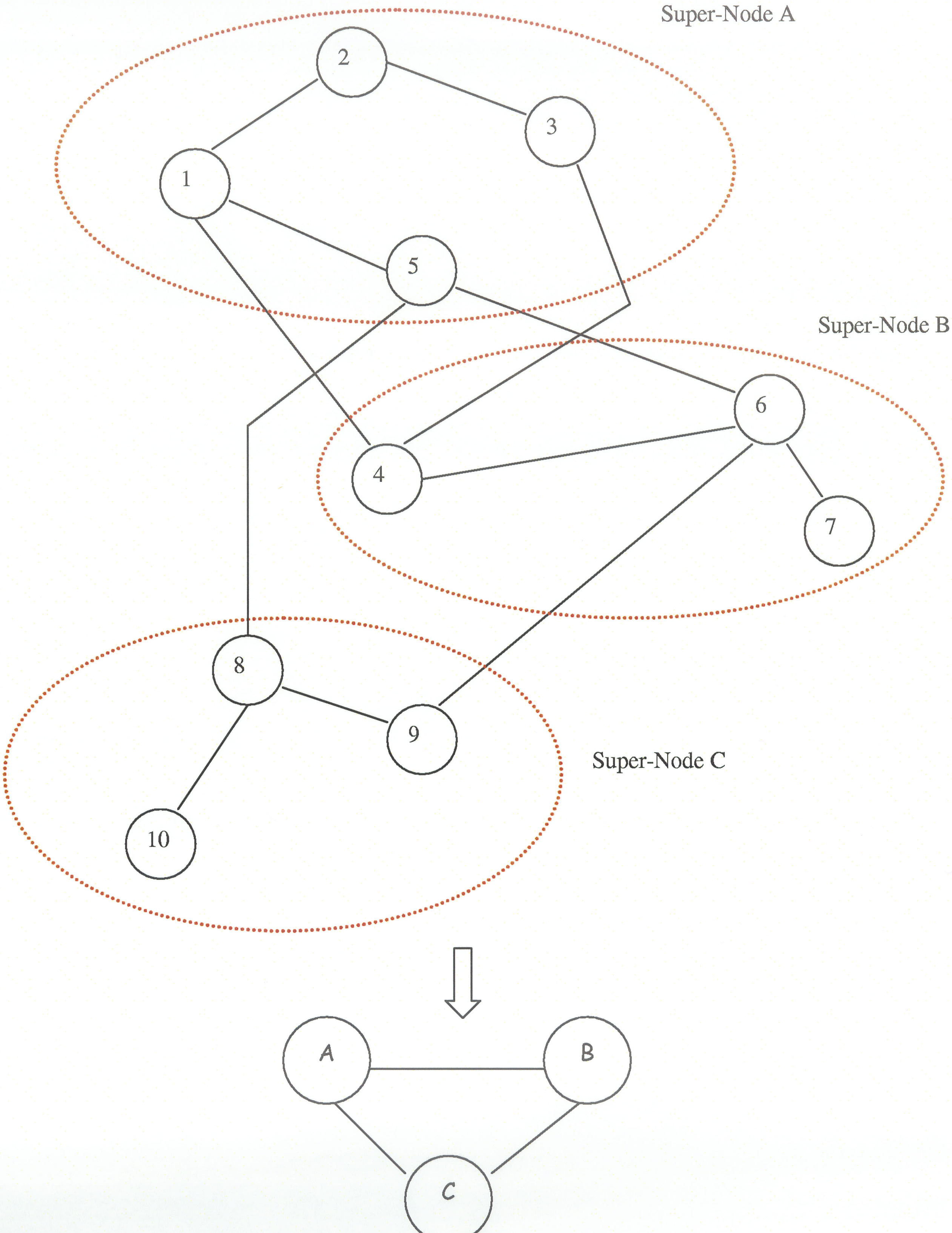

Figure 6.7: 3-Super-node Network (Model 2)

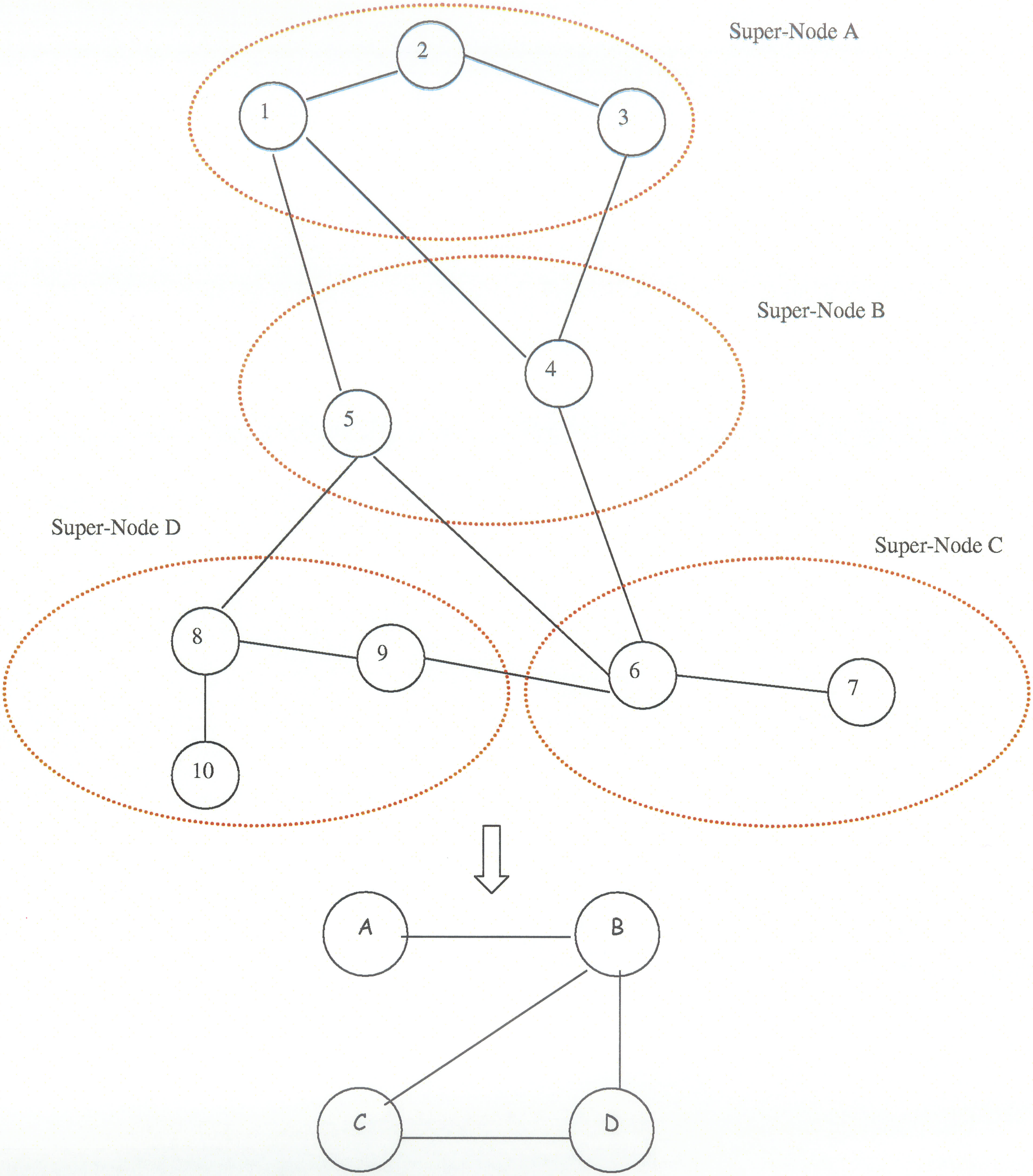

Figure 6.8: 4-Super-node Network (Model 3)

We estimate the traffic matrix of these three grouped network models at coarse level using Linear Programming. The result is presented in Tables 6.5 and 6.6:

Model1		Model2	
OD Pair	Estimated OD traffic	OD Pair	Estimated OD traffic
A→A	2174.00	A→A	592.34
A→B	2307.76	A→B	4146.36
A→C	2635.25	A→C	2485.30
B→A	2166.01	B→A	3763.46
B→B	1413.24	B→B	0.00
B→C	1651.76	B→C	1360.54
C→A	2269.99	C→A	2163.20
C→B	1609.01	C→B	1274.64
C→C	1197.00	C→C	1638.16

Table 6.5: Estimated OD traffic at coarse level of model 1 and 2

Model3	
OD Pair	Estimated OD traffic
A→A	111.08
A→B	1326.63
A→C	2165.20
A→D	1789.09
B→A	1490.09
B→B	633.21
B→C	737.49
B→D	696.21
C→A	1749.33
C→B	684.46
C→C	0.00
C→D	965.21
D→A	1620.50
D→B	542.70
D→C	879.32
D→D	2033.49

Table 6.6: Estimated OD traffic at coarse level of model 3

6.2.3 Assessing the quality of traffic matrix estimation

In the final step, we estimate the traffic matrix of large network using Tomogravity, Entropy Maximization and Linear Programming. Then, we use the set of groupings performed in the previous step to assess the quality of the estimation done by various techniques.

In Tables 6.7, 6.8 and 6.9, we estimate the performance of each technique by comparing their results with the estimated traffic matrix values at coarse level produced by grouping models 1, 2 and 3 respectively.

OD PAIR	Estimated OD traffic	TM	% of Error	EM	% of Error	LP	% of Error
A→A	2174.002	1959.431	9.87	2192.428	0.85	2174	0.00
A→B	2307.758	2136.822	7.41	2419.823	4.86	3967.239	71.91
A→C	2635.247	2758.434	4.67	2523.177	4.25	975.7604	62.97
B→A	2166.011	2181.047	0.69	2231.207	3.01	742.1435	65.74
B→B	1413.238	1141.18	19.25	1120.56	20.71	1177.619	16.67
B→C	1651.756	1821.778	10.29	1763.823	6.78	3311.238	100.47
C→A	2269.992	2112.837	6.92	2204.793	2.87	3693.857	62.73
C→B	1609.011	1642.684	2.09	1674.206	4.05	185.1432	88.49
C→C	1197.002	1168.657	2.37	1082.054	9.60	1197	0.00

Table 6.7: Comparing TM, EM & LP technique results with grouping model 1

OD PAIR	Estimated OD traffic	TM	% of Error	EM	% of Error	LP	% of Error
A→A	592.34	1994.81	236.77	2135.12	260.46	2658.61	348.83
A→B	4146.36	2103.93	49.26	2482.36	40.13	2412.24	41.82
A→C	2485.30	2738.64	10.19	2534.57	1.98	2153.15	13.36
B→A	3763.46	2277.71	39.48	2173.69	42.24	1732.72	53.96

B→B	0.00	1042.03	100.00	1172.84	100.00	1257.43	100.00
B→C	1360.54	1841.57	35.36	1752.44	28.80	2133.85	56.84
C→A	2163.20	2234.47	3.29	2138.24	1.15	2127.67	1.64
C→B	1274.64	1521.06	19.33	1740.76	36.57	1751.33	37.40
C→C	1638.16	1168.66	28.66	1082.05	33.95	1197.00	26.93

Table 6.8: Comparing TM, EM & LP technique results with grouping model 2

OD PAIR	Estimated OD traffic	TM	% of Error	EM	% of Error	LP	% of Error
A→A	111.08	841.16	657.26	1106.41	896.05	1163.39	947.35
A→B	1326.63	1113.12	16.09	1043.63	21.33	1496.15	12.78
A→C	2165.20	1039.67	51.98	1320.24	39.02	2001.29	7.57
A→D	1789.09	2077.60	16.13	1894.14	5.87	731.17	59.13
B→A	1490.09	1158.79	22.23	1071.11	28.12	1009.68	32.24
B→B	633.21	304.91	51.85	382.82	39.54	16.12	97.45
B→C	737.49	743.39	0.80	835.25	13.26	864.63	17.24
B→D	696.21	1341.88	92.74	1269.46	82.34	1666.57	139.38
C→A	1749.33	1141.16	34.77	1133.11	35.23	708.43	59.50
C→B	684.46	712.74	4.13	684.76	0.04	32.20	95.30
C→C	0.00	363.54	100.00	386.70	100.00	769.11	100.00
C→D	965.21	1160.74	20.26	1123.40	16.39	1889.26	95.74
D→A	1620.50	1662.70	2.60	1632.79	0.76	2089.50	28.94
D→B	542.70	1021.90	88.30	1077.44	98.53	1642.54	202.66
D→C	879.32	1070.92	21.79	1168.76	32.92	146.97	83.29
D→D	2033.49	1168.66	42.53	1082.05	46.79	1197.00	41.14

Table 6.9: Comparing TM, EM & LP technique results with grouping model 3

As we see from Tables 6.7 to 6.9, the estimated traffic matrix results of TM, EM and LP are close to the results obtained using grouping model 1 compared to the other two models. Hence, we pick model 1 as reference to assess the quality of estimation technique results. For the example of 10-node network Tomogravity, Entropy Maximization and Linear Programming produce average errors of 7.1%, 6.3% and 52%, respectively.

6.3 Bell Canada IP Network

We estimate the traffic matrix of the Bell Network at coarse level by grouping the network into five different ways (model 1 to 5) as shown in Figure 6.9 to 6.13. We use Linear Programming method for estimating the traffic matrix at coarse level. The results of coarse estimation using different models are given in Table 6.10 to 6.14 (column LP_COARSE).

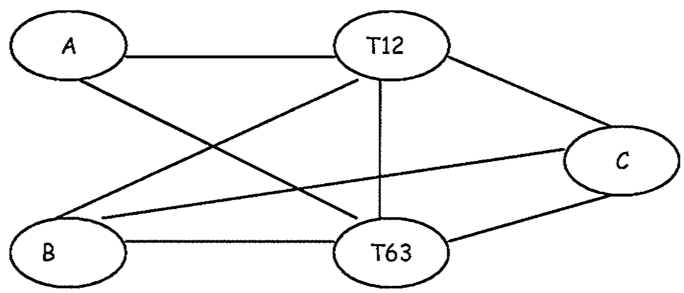

Figure 6.9 Bell Canada IP Network grouping model 1

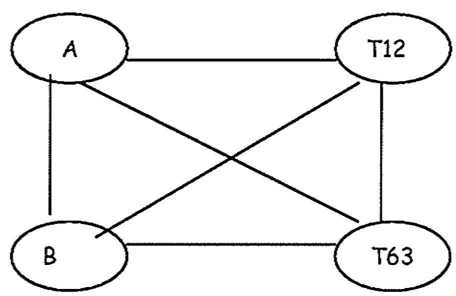

Figure 6.10 Bell Canada IP Network grouping model 2

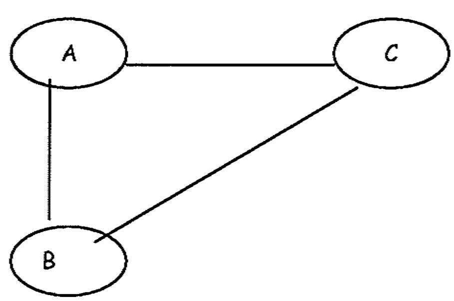

Figure 6.11 Bell Canada IP Network grouping model 3

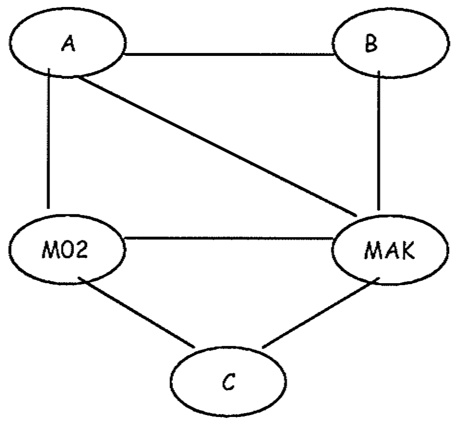

Figure 6.12 Bell Canada IP Network grouping model 4

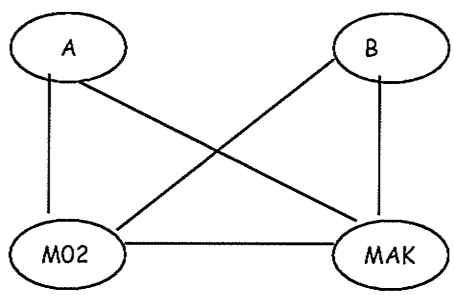

Figure 6.13 Bell Canada IP Network grouping model 5

As a next step, the traffic matrix of the Bell Network is estimated at granular level using Tomogravity, Entropy Maximization, Linear Programming and hybrid model. These results are shown in Table 6.10 to 6.14 TM_GRANULAR, EM_GRANULAR, LP_GRANULAR and HYBRID_GRANULAR columns respectively.

OD PAIR	LP_ COARSE	TM_ GRANULAR	EM_ GRANULAR	LP_ GRANULAR	HYBRID_ GRANULAR
A→A	0.00	1065.56	1761.43	0.00	963.78
A→T12	2845.74	2198.05	1832.89	1628.40	2025.69
A→B	1722.18	926.18	1510.91	878.81	1010.27
A→T63	2142.07	2258.70	2134.51	499.79	1887.04
A→C	1756.27	3366.44	2381.00	5459.26	3627.33
T12→A	1946.01	1986.44	1987.66	2450.27	2079.40
T12→T12	0.00	0.00	0.00	0.00	0.00
T12→B	1807.88	844.59	921.50	486.33	785.25
T12→T63	1685.99	1462.90	1369.52	116.71	1178.72
T12→C	1891.76	1101.55	1178.23	4278.35	1749.18
B→A	1916.05	1667.52	2099.34	483.25	1499.76
B→T12	2810.41	1223.19	1166.47	231.96	1015.87
B→B	336.06	846.08	955.62	1023.06	898.98
B→T63	2704.02	1151.92	1257.97	3411.47	1620.79
B→C	1219.62	2240.11	1560.66	3836.43	2450.65
T63→A	1383.19	1733.66	1876.06	1533.89	1716.48
T63→T12	1911.72	1750.05	1580.10	856.64	1544.18
T63→B	1485.88	656.21	760.93	642.01	670.11
T63→T63	0.00	0.00	0.00	0.00	0.00
T63→C	1839.23	1607.28	1841.07	3587.49	2040.71
C→A	4549.96	4266.16	2814.70	6601.02	4500.87
C→T12	1683.59	1406.38	1469.75	5008.24	2136.88
C→B	455.80	2147.38	1350.31	3298.66	2250.09
C→T63	1634.07	1446.44	1715.61	3973.40	1994.89
C→C	14370.43	8994.16	9029.30	3812.54	7963.47

Table 6.10: Traffic matrix estimation of Bell Network at coarse level using model 1

SD PAIR	LP_ COARSE	TM_ GRANULAR	EM_ GRANULAR	LP_ GRANULAR	HYBRID_ GRANULAR
A→A	4098.82	4505.34	6327.30	2385.12	4372.79
A→T12	6630.85	3421.24	2999.37	1860.36	3041.56
A→B	1219.62	5608.41	3957.30	9315.69	6085.67
A→T63	5503.13	3410.63	3392.47	3911.26	3507.83
T12→A	5906.74	2831.04	2909.17	2936.60	2864.65
T12→T12	0.00	0.00	0.00	0.00	0.00
T12→B	1424.91	1101.55	1178.23	4278.35	1749.18

T12→T63	0.00	1462.90	1369.52	116.71	1178.72
B→A	455.80	6413.55	4165.01	9899.69	6750.96
B→T12	2620.62	1406.38	1469.75	5008.24	2136.88
B→B	16954.42	8994.16	9029.30	3812.54	7963.47
B→T63	2663.01	1446.44	1715.61	3973.40	1994.89
T63→A	5141.67	2389.86	2637.00	2175.89	2386.59
T63→T12	0.00	1750.05	1580.10	856.64	1544.18
T63→B	1478.35	1607.28	1841.07	3587.49	2040.71
T63→T63	0.00	0.00	0.00	0.00	0.00

Table 6.11: Traffic matrix estimation of Bell Network at coarse level using model 2

OD PAIR	LP_ COARSE	TM_ GRANULAR	EM_ GRANULAR	LP_ GRANULAR	HYBRID_ GRANULAR
A→A	11087.76	12455.37	12542.19	7085.69	11395.29
A→B	5352.01	2426.98	3193.35	2007.15	2465.63
A→C	5978.17	6075.27	5400.29	13325.10	7417.22
B→A	7766.54	4042.63	4523.78	4126.68	4136.42
B→B	0.00	846.08	955.62	1023.06	898.98
B→C	1219.62	2240.11	1560.66	3836.43	2450.65
C→A	8358.56	7118.98	6000.05	15582.66	8632.64
C→B	455.80	1922.29	1226.10	2730.48	1972.53
C→C	13879.52	8994.16	9029.30	3812.54	7963.47

Table 6.12: Traffic matrix estimation of Bell Network at coarse level using model 3

OD PAIR	LP_ COARSE	TM_ GRANULAR	EM_ GRANULAR	LP_ GRANULAR	HYBRID_ GRANULAR
A→A	14508.08	12455.37	12542.19	7085.69	11395.29
A→B	4139.65	2426.98	3193.35	2007.15	2465.63
A→M02	600.25	708.52	884.34	3036.58	1202.26
A→MAK	581.04	499.82	419.64	2577.62	902.55
A→C	2588.91	3654.87	2906.75	7710.90	4346.36
B→A	6111.57	4042.63	4523.78	4126.68	4136.42
B→B	3299.71	846.08	955.62	1023.06	898.98
B→M02	1467.30	419.33	255.90	776.41	464.60
B→MAK	103.44	310.04	142.59	435.04	308.25

B→C	531.56	1093.23	909.37	486.68	942.49
M02→A	1237.70	871.59	1088.14	4444.23	1620.76
M02→B	1184.06	438.43	213.67	416.96	398.18
M02→M02	0.00	0.00	0.00	0.00	0.00
M02→MAK	0.86	304.66	265.54	0.00	237.47
M02→C	2642.33	1328.63	1382.69	203.76	1112.30
MAK→A	1462.51	618.11	541.90	2376.35	957.57
MAK→B	128.91	391.01	122.45	1020.16	473.87
MAK→M02	0.81	413.15	337.54	0.00	318.42
MAK→MAK	0.00	0.00	0.00	0.00	0.00
MAK→C	2595.05	1321.95	1416.18	790.77	1230.80
C→A	3873.55	5138.53	3848.63	7620.93	5428.58
C→B	534.26	1075.69	902.80	1468.43	1126.56
C→M02	2688.89	1389.61	1513.97	305.35	1192.66
C→MAK	2327.32	966.29	1313.67	0.00	828.61
C→C	1490.19	1146.66	1023.41	179.00	933.41

Table 6.13: Traffic matrix estimation of Bell Network at coarse level using model 4

OD PAIR	LP_ COARSE	TM_ GRANULAR	EM_ GRANULAR	LP_ GRANULAR	HYBRID_ GRANULAR
A→A	27232.36	22133.63	23290.06	17915.13	22976.86
A→B	4193.12	5386.50	4499.71	8551.84	4739.52
A→M02	1375.44	1355.88	1298.93	4451.90	1331.68
A→MAK	1130.62	946.36	669.00	3012.65	750.46
B→A	5571.39	6973.31	5385.30	10429.85	5801.48
B→B	1406.30	1146.66	1023.41	179.00	1048.31
B→M02	2207.59	1389.61	1513.97	305.35	1476.49
B→MAK	1728.91	966.29	1313.67	0.00	1221.37
M02→A	2032.37	1509.09	1406.18	4861.19	1451.92
M02→B	2439.69	1328.63	1382.69	203.76	1362.48
M02→M02	0.00	0.00	0.00	0.00	0.00
M02→MAK	592.89	304.66	265.54	0.00	273.46
MAK→A	1663.32	1171.25	753.35	3396.51	870.75
MAK→B	1789.50	1321.95	1416.18	790.77	1389.52
MAK→M02	734.46	413.15	337.54	0.00	353.89
MAK→MAK	0.00	0.00	0.00	0.00	0.00

Table 6.14: Traffic matrix estimation of Bell Network at coarse level using model 5

Following table shows the average error obtained by using five different models.

MODEL	AVERAGE ERROR			
	TM	EM	LP	HYBRID
1	49.3	38.2	108.8	53.4
2	31.0	34.6	67.6	30.0
3	35.8	28.0	91.7	39.2
4	57.8	37.2	144.2	67.5
5	26.0	26.0	92.5	25.0

Table 6.15: *Average Error produced by TM, EM, LP and Hybrid techniques with different grouping models*

Based on the average error produced by different estimation techniques with different groupings, model 5 seems to estimate close to the results obtained at granular level followed by model 2, 3, 1 and 4. Hence, with model 5 as reference we assess the quality of results generated by the estimation techniques. For Bell Canada IP Network, TM, EM, LP and Hybrid methods produce average errors of 26%, 26%, 92.5% and 25% respectively.

6.4 Summary

Motivated by the ability of LP method to estimate the traffic matrix very close to accuracy for small networks, we developed a hierarchical model. In this model, we reduce a large network into small by grouping and obtain the traffic matrix of large network at coarse level. Then with the help of the information obtained about the network at coarse level, we evaluate the performance of traffic matrix estimation techniques.

Chapter 7

Bell Canada IP Network

In this chapter, we discuss the Bell Canada IP Network. We explain how we derive various input data required for estimating traffic matrix from the SNMP link load measurements. The data used for illustration in this chapter does not represent the real network data because of the proprietory aspects of the data, and Non-Disclosure Agreements.

7.1 Sample Bell Network Data

We were provided with core and non-core backbone link load measurements of Bell Canada network. Core backbone links are those that connect the core routers in the network, whereas non-core backcone links connect edge routers and core routers together. In this section, we consider only one region 'Calgary' for illustration purpose. Table 7.1, shows sample core-link information for Calgary region in the network.

Source Device	Destination Device	Speed (Mbps)	Transmitted Traffic (%)	Received Traffic (%)
CR1_Calgary	CR2_Vancouver	1000	0.36	0.5
CR2_Calgary	CR1_Toronto	2488	0.4	0.22
CR1_Calgary	CR2_Calgary	622	0.6	0.2
CR2_Calgary	CR1_Calgary	622	0.1	0.08

Table 7.1: Core-link load measurements

As we see, the Calgary region consists of two core routers. CR1_Calgary has two core links one connecting to CR2_Vancouver and other connecting to CR2_Calgary. Similarly, CR2_Calgary has two core links one connecting to CR1_Toronto and other connecting to CR1_Calgary. Also, the table provides us information regarding speed of the link running between core routers. For example, the speed of the core-link between CR1_Calgary and CR2_Vancouver is 1000 Mbps. Transmitted traffic and received traffic provides us the percentage of traffic transmitted and received on each link. This information is useful in calculating the flow on each link which is explained in Section 7.2.

Table 7.2, shows sample non-core backbone link information for the same calgary region.

Source Device	Destination Device	Speed (Mbps)	Transmitted Traffic (%)	Received Traffic (%)
ER1_Calgary	CR1_Calgary	155	0.11	0.62
ER2_Calgary	CR1_Calgary	100	0.36	0.21
ER1_Calgary	CR2_Calgary	622	0.05	0.23
ER2_Calgary	CR2_Calgary	622	0.66	0.08

Table 7.2: Non-core-link load measurements

From the information provided in Table 7.2, we see that there are totally 2 edge routers present in Calgary region. Similar to core-link information, this table also provides us the speed, percentage of traffic transmitted and received on each non-core backbone link.

7.2 Input Data Generation

The inputs required for estimating traffic matrix using the techniques discussed in previous chapters are network topology, production, attraction, link load and routing matrix. In this section, we explain how these inputs are generated for Bell Canada network with the data provided to us. We use the sample data shown in Section 7.1 for illustration purpose.

7.2.1 Network Topology

Topology gives us information regarding the arrangement of nodes in a network and how these nodes are connected. For example from Table 7.1, we know that calgary is connected to Vancouver and Toronto.Also from Table 7.2, we can obtain the topology in more detail showing the arrangement of edge routers and core routers within Calgary. Similarly with the complete network information we obtain the topology of Bell Canada network. Figure 7.1, shows the topology that is obtained with the information in Tables 7.1 and 7.2.

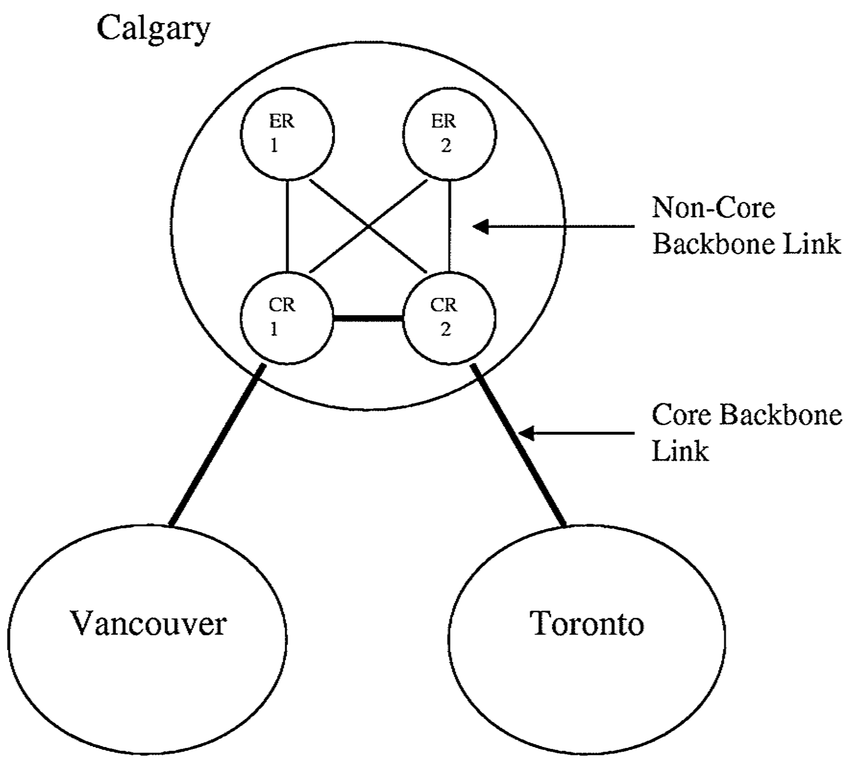

Figure 7.1: Topology showing calgary region

7.2.2 Production and Attraction

Production is the total amount of traffic generated by each region that enters into the network. For example, from Figure 7.2 we can say the production of Calgary is the sum of incoming traffic on edge links 1 to 6. Attraction, is the total amount of traffic destined to a particular region that leaves the network. For example, attraction of Calgary is the sum of outgoing traffic on edge links 7 to 12. Since we do not have the edge link load data of Bell network, we obtain the production and attraction from the non-core link information that is provided to us. The method of deriving the production and attraction is explained below by considering Calgary region as an example.

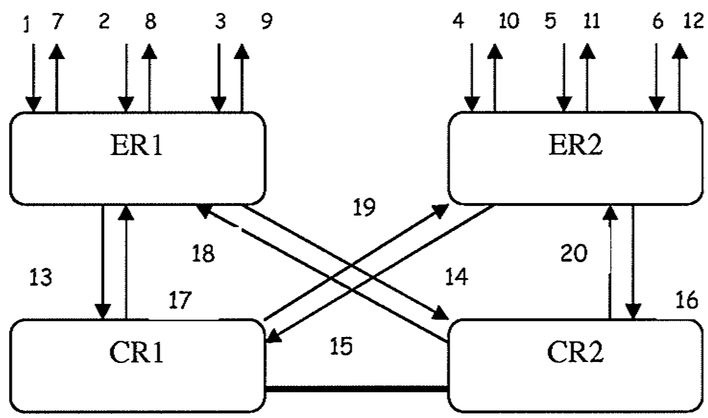

Figure 7.2: Calgary Region

As we know, the node flow conservation theory algebraically states that the sum of the flow through arcs directed towards a node equals the sum of the flow through arcs directed away from that node. Based on this, we can obtain the production and attraction of Calgary from non-core link information as shown below:

Production of Calgary = Sum of incoming traffic ($L_1 + L_2 + L_3 + L_4 + L_5 + L_6$)

= Sum of outgoing traffic ($L_{13} + L_{14} + L_{15} + L_{16}$)

Attraction of Calgary = Sum of outgoing traffic ($L_7 + L_8 + L_9 + L_{10} + L_{11} + L_{12}$)

= Sum of incoming traffic ($L_{17} + L_{18} + L_{19} + L_{20}$)

7.2.3 Link Load

Link load is the total measured traffic flow on each link. We calculate the link load of each link i by using the following formula:

$$linkload_i = (speed_i * \%oftransmit\ tedtraffic_i) + (speed_i * \%ofreceived\ traffic_i)$$

For example, we can calculate the load on link between CR1_Calgary and CR1_Vancouver from the data provided in Table 7.1 as follows:

$$\text{Linkload} = 1000 * 0.36 + 1000 * 0.5 = 360 + 500 = 860 \text{ Mbps}$$

This tells us that 360 Mbps of traffic flows from CR1_Calgary to CR1_Vancouver and 500 Mbps flow from CR1_Vancouver to CR1_Calgary. In total, 860 Mbps of traffic flows between CR1_Calgary and CR1_Vancouver.

7.2.4 Routing Matrix

Routing is a process of determining the path to be used for forwarding the network traffic from a particular source to a destination. This routing information which gives the hierarchy of link routing at each node is described in the routing matrix associated with a network control protocol. Here, we assume shortest path routing algorithm for determining the routing matrix information. Table 7.3, shows the routing matrix of an example 4-node network in Figure 7.3.

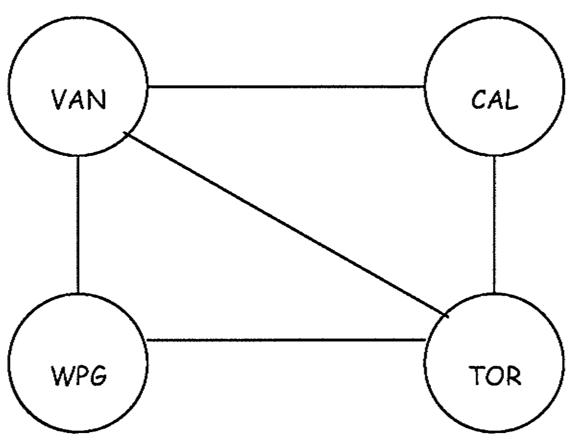

Figure 7.3: 4-node Network

	VAN ->WPG	VAN ->TOR	VAN ->CAL	WPG ->VAN	WPG ->TOR	WPG ->CAL	TOR ->VAN	TOR ->WPG	TOR ->CAL	CAL ->VAN	CAL ->WPG	CAL ->TOR
VAN->WPG	1	0	0	0	0	0	0	0	0	0	0.5	0
VAN->TOR	0	1	0	0	0	0	0	0	0	0	0	0
VAN->CAL	0	0	1	0	0	0.5	0	0	0	0	0	0
WPG->VAN	0	0	0	1	0	0.5	0	0	0	0	0	0
WPG->TOR	0	0	0	0	1	0.5	0	0	0	0	0	0
TOR->VAN	0	0	0	0	0	0	1	0	0	0	0	0
TOR->WPG	0	0	0	0	0	0	0	1	0	0	0.5	0
TOR->CAL	0	0	0	0	0	0.5	0	0	1	0	0	0
CAL->VAN	0	0	0	0	0	0	0	0	0	1	0.5	0
CAL->TOR	0	0	0	0	0	0	0	0	0	0	0.5	1

Table 7.3: Routing Matrix of 4-node Network

Each row in the above routing matrix represents directed link in the network and each column represents the source-destination pairs. In the above example 4-node network, we have totally 10 directed links and 12 OD pairs. For example, the first column tells us that 100% of traffic from Vancouver to Winnipeg is routed via the link VAN → WPG, since this is the only shortest available path. On the other hand, if there is more than one shortest path between an OD pair then the total traffic is divided equally among these available paths. For example, as we see in the above table the total traffic from Calgary to Winnipeg is equally divided between the two paths CAL → VAN → WPG and CAL → TOR → WPG.

7.3 Results

We estimated the traffic matrix of Bell Canada IP Network using Tomogravity, Entropy Maximization, Linear Programming and Hybrid model. We also checked the quality of results generated by these methods using the hierarchical modeling.

First, we reduced the Bell IP network that consists of 27 POPs and 92 directed links into 4 node network with 10 directed links. The 4-node network is shown in Figure 7.4. Secondly, we obtained the traffic matrix of the network at coarse level by applying LP on the 4-node network. The aggregated results obtained using LP are shown in Table 7.4.

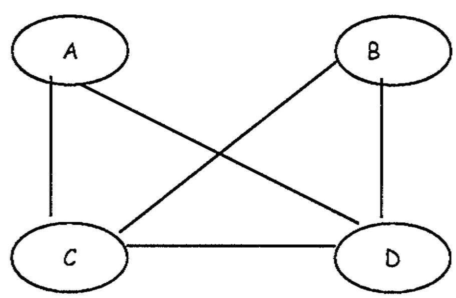

Figure 7.4: Reduced Bell IP Network

SD PAIR	LP_COARSE
A→A	27232.36
A→B	4193.12
A→C	1375.44
A→D	1130.62
B→A	5571.39
B→B	1406.30
B→C	2207.59
B→D	1728.91
C→A	2032.37

C→B	2439.69
C→C	0.00
C→D	592.89
D→A	1663.32
D→B	1789.50
D→C	734.46
D→D	0.00

Table 7.4: Estimated traffic matrix of Bell IP Network at coarse level

Finally, we checked the quality of the results that are obtained at granular level using Tomogravity, Entropy Maximization, Linear Programming and Hybrid model. This is done by comparing the results generated by these techniques with the aggregated results obtained by using reduced 4-node network. The comparison is shown in the below table.

SD PAIR	LP_COARSE	TM_GRANULAR	EM_GRANULAR	LP_GRANULAR	HYBRID_ GRANULAR
A→A	27232.36	22133.63	23290.06	17915.13	22976.86
A→B	4193.12	5386.50	4499.71	8551.84	4739.52
A→C	1375.44	1355.88	1298.93	4451.90	1331.68
A→D	1130.62	946.36	669.00	3012.65	750.46
B→A	5571.39	6973.31	5385.30	10429.85	5801.48
B→B	1406.30	1146.66	1023.41	179.00	1048.31
B→C	2207.59	1389.61	1513.97	305.35	1476.49
B→D	1728.91	966.29	1313.67	0.00	1221.37
C→A	2032.37	1509.09	1406.18	4861.19	1451.92
C→B	2439.69	1328.63	1382.69	203.76	1362.48
C→C	0.00	0.00	0.00	0.00	0
C→D	592.89	304.66	265.54	0.00	273.46
D→A	1663.32	1171.25	753.35	3396.51	870.75
D→B	1789.50	1321.95	1416.18	790.77	1389.52
D→C	734.46	413.15	337.54	0.00	353.89
D→D	0.00	0.00	0.00	0.00	0

Table 7.5: Comparison between traffic matrixes estimated at coarse and granular level

7.4 Summary

By comparing the estimated traffic matrix at granular level with aggregated data, we are able to assess the quality of results generated by different estimation techniques. As we see in Table 7.6, for the Bell IP network Tomogravity, Entropy Maximization, Linear Programming and Hybrid model produces an average error of 26%, 26%, 93% and 25% respectively.

SD PAIR	% of ERROR			
	TM	EM	LP	HYBRID
A→A	18.72	14.48	34.21	15.63
A→B	28.46	7.31	103.95	13.03
A→C	1.42	5.56	223.67	3.18
A→D	16.30	40.83	166.46	33.62
B→A	25.16	3.34	87.20	4.13
B→B	18.46	27.23	87.27	25.46
B→C	37.05	31.42	86.17	33.12
B→D	44.11	24.02	100.00	29.36
C→A	25.75	30.81	139.19	28.56
C→B	45.54	43.33	91.65	44.15
C→C	0.00	0.00	0.00	0.00
C→D	48.61	55.21	100.00	53.88
D→A	29.58	54.71	104.20	47.65
D→B	26.13	20.86	55.81	22.35
D→C	43.75	54.04	100.00	51.82
D→D	0.00	0.00	0.00	0.00

Table 7.6: % of error produced by TM, EM, LP and Hybrid Model

Chapter 8

Conclusion

In this chapter, we summarize the results obtained in this thesis.

8.1 Conclusions

In the first part of the thesis, we performed a comparative study of three most efficient traffic matrix estimation techniques namely Tomogravity, Entropy Maximization and Linear Programming. From the preliminary experiments performed, we noticed that LP performs better than TM and EM for small network models irrespective of the nature of traffic distributions. But in case of large networks, TM and EM performs equally well compared to LP. From this study, we observed that the performance of LP varies according to the network size, with its performance getting worse as the network size gets larger.

In the second part of the thesis, based on the lessons learned from the first part we proposed new directions for estimating traffic matrix. First, we identified the possible reasons for conflicting behaviour of LP between different network sizes and resolved them. One of the main reasons for such behaviour is insufficiency of total constraints available for solving the unknown traffic matrix elements. We overcame this drawback in our newly proposed optimization techniques namely LP_with_NCLC and LP_with_NFCC by adding extra non-core backbone link constraints and node flow conservation constraints. Both these optimization techniques show better performance compared to LP irrespective of the network size and traffic distributions. The main advantage of our new optimization technique is the ability to estimate exact traffic matrix for most of the small network models.

Second, we developed a Hybrid Model which combines Tomogravity, Entropy Maximization and Linear Programming. It uses quadratic programming to find optimal proportional values, such that the difference between actual and estimated traffic matrix elements using the three techniques is minimized. Based on the testing of Hybrid model for various 14 - node networks and traffic distributions, we found that hybrid model performs better than the other 3 techniques (TM, EM and LP). The behaviour of hybrid model does not show any dependency on the nature of traffic distribution and network model. Thus, this method offers network operators flexible option for estimating traffic matrix of their network that follows any type of traffic distribution.

Third, motivated by the key advantage of our new optimization technique, we developed a hierarchical model. This model helps us to assess the quality of results generated by any traffic matrix estimation technique, especially when we do not have the real matrix to compare with.

Finally, in our thesis we illustrated the application of our new traffic matrix estimation techniques on Bell Canada IP Network.

Appendix A

TomoUtility Model

A.1 Introduction

As explained in Chapter 2, Tomogravity consists of two steps namely: a gravity modeling step and a Tomographic estimation step. The gravity modeling step is used to obtain the initial solution by solving a gravity model using edge link load data. Then the initial solution is refined by using the tomographic estimation step. Here we replaced the gravity model by utility model [6] which is based on choice models to generate the initial solution and then refined it by using the tomographic estimation step.

A.2 Utility Model

The authors express the random utility of POP i choosing to send a packet to POP j as the sum of the deterministic component, V_j^i, and a random component, ε_j^i, as follows:

$$U_j^i = V_j^i + \varepsilon_j^i$$

The deterministic component V_j^i which quantifies the attractiveness of choice j for POP i is specified as follows:

$$V_j^i(X_j^i) = \sum_{m=1}^{M} \mu_m w_j^i(m) + \gamma_j$$

μ_m defines the relative importance of attribute m with respect to others, { w_j^i } defines a vector of attributes including both the attributes of alternative j and those of the decision-maker i and γ_j is a scaling term representing the amount of attractivity of POP j not captured by the attributes.

The random component that is used to capture some level of uncertainty in the decision process is modeled using a multinomial logit function. The probability of choosing a given alternative j from the set of choices C is given by the following equation:

$$P_C^i(j) = \frac{e^{V_j^i}}{\sum_{k \in C} e^{V_k^i}}$$

The authors model the traffic between a pair of POPs using the above probability function as follows:

$$X_{ij} = O_i \frac{e^{V_j^i}}{\sum_{k \in C} e^{V_k^i}}$$

where O_i represents the total outgoing bytes sent into the network by POP i.

In summary, the authors believe that POP fanouts behave according to a multinomial logit function and POP attributes influence the fanout via the exponent in the mlogit function.

A.3 Comparing Tomogravity and TomoUtility Model

The TomoUtility model that combines the Utility model with tomographic estimation step is tested on different 4-node networks and traffic types. Also, we have compared the tomoutility model with the Tomogravity model. The results are presented in the following sections.

A.3.1 Constant Traffic

Here, we consider 10 different 4-node networks with constant traffic type distribution. The results of this analysis are presented below:

Model	AVERAGE ERROR		MAXIMUM ERROR	
	TM_with_ GRAVITY	TM_with_ UTILITY	TM_with_ GRAVITY	TM_with_ UTILITY
1	35.01	31.41	221.90	118.30
2	46.00	40.26	146.55	184.50
3	0.00	0.00	0.00	0.00
4	36.43	40.90	126.95	153.60
5	31.22	40.03	165.40	131.75
6	8.40	25.41	25.20	66.90
7	43.46	43.86	227.15	130.25
8	28.05	42.48	93.25	146.25
9	38.87	46.69	205.35	184.75

10	38.08	50.06	140.95	201.30

Table A.1: TomoUtility - 4 Node Network (Constant Traffic)

A.3.2 Uniform Traffic

Here, we consider 10 different 4-node networks with uniform traffic type distribution. The results of this analysis are presented below:

Model	AVERAGE ERROR		MAXIMUM ERROR	
	TM_with_ GRAVITY	TM_with_ UTILITY	TM_with_ GRAVITY	TM_with_ UTILITY
1	19.82	41.32	60.71	82.53
2	31.82	45.23	78.63	90.35
3	0.00	0.00	0.00	0.00
4	31.99	45.28	82.56	139.30
5	12.62	32.60	36.96	139.30
6	13.24	22.91	32.02	53.30
7	29.11	49.32	74.06	139.30
8	36.88	23.48	172.62	58.57
9	22.15	53.73	49.42	133.46
10	24.59	50.07	83.55	155.53

Table A.2: TomoUtility - 4 Node Network (Uniform Traffic)

A.3.3 Poisson Traffic

Here, we consider 10 different 4-node networks with Poisson traffic type distribution. The results of this analysis are presented below:

Model	AVERAGE ERROR		MAXIMUM ERROR	
	TM_with_ GRAVITY	TM_with_ UTILITY	TM_with_ GRAVITY	TM_with_ UTILITY
1	17.25	49.05	43.11	81.31
2	20.94	55.51	43.26	88.35
3	0.00	0.00	0.00	0.00
4	23.02	47.40	68.24	127.96
5	17.01	32.51	68.24	127.96
6	4.80	39.30	13.43	86.05
7	26.75	56.20	68.34	127.96
8	24.51	43.99	78.25	92.52
9	17.95	59.56	43.56	175.71
10	31.97	52.94	115.28	115.37

Table A.3: TomoUtility - 4 Node Network (Poisson Traffic)

A.3.4 Normal Traffic

Here, we consider 10 different 4-node networks with normal traffic type distribution. The results of this analysis are presented below:

Model	AVERAGE ERROR		MAXIMUM ERROR	
	TM_with_ GRAVITY	TM_with_ UTILITY	TM_with_ GRAVITY	TM_with_ UTILITY
1	32.00	35.24	125.88	160.81
2	33.28	41.06	118.09	151.78
3	0.00	0	0.00	0.00
4	29.63	38.56	99.25	183.04
5	17.13	15.28	51.39	59.39
6	27.46	37.45	74.10	121.50
7	28.29	46.44	96.28	220.02
8	26.97	58.07	63.99	184.24

9	32.42	45.74	69.99	198.41
10	31.32	30.35	126.14	160.97

Table A.4: TomoUtility - 4 Node Network (Normal Traffic)

A.3.5 Bimodal Traffic

Here, we consider 10 different 4-node networks with bimodal traffic type distribution. The results of this analysis are presented below:

Model	AVERAGE ERROR		MAXIMUM ERROR	
	TM_with_ GRAVITY	TM_with_ UTILITY	TM_with_ GRAVITY	TM_with_ UTILITY
1	48.02	52.76	186.80	187.02
2	41.20	100.71	202.68	295.26
3	0.00	0.00	0.00	0.00
4	29.23	150.24	100.00	454.67
5	23.26	62.01	94.03	201.95
6	9.86	104.15	26.90	241.91
7	55.86	107.99	267.79	501.11
8	56.00	159.07	244.83	531.36
9	87.79	116.46	297.38	453.13
10	97.42	97.36	615.75	278.57

Table A.5: TomoUtility - 4 Node Network (Bimodal Traffic)

A.4 Summary

Replacing gravity modeling step by utility model seems not to be a good choice. Using utility model performs worse compared to the gravity modeling for various 4-node networks and traffic types.

Appendix B

Network Models

B.1 4-Node Network Models

The 10 different 4-node network models that we selected for our testing purpose are given in Tables B.1 to B.10.

Node	N1	N2	N3	N4
N1	0	1	1	1
N2	1	0	1	0
N3	1	1	0	1
N4	1	0	1	0

Table B.1 4-Node Network - Model 1

Node	N1	N2	N3	N4
N1	0	1	0	1
N2	1	0	1	0
N3	0	1	0	1
N4	1	0	1	0

Table B.2 4-Node Network - Model 2

Node	N1	N2	N3	N4
N1	0	1	1	1
N2	1	0	1	1
N3	1	1	0	1
N4	1	1	1	0

Table B.3 4-Node Network - Model 3

Node	N1	N2	N3	N4
N1	0	1	0	0
N2	1	0	1	0
N3	0	1	0	1
N4	0	0	1	0

Table B.4 4-Node Network - Model 4

Node	N1	N2	N3	N4
N1	0	1	1	0
N2	1	0	1	0
N3	1	1	0	1
N4	0	0	1	0

Table B.5 4-Node Network - Model 5

Node	N1	N2	N3	N4
N1	0	1	1	0
N2	1	0	1	1
N3	1	1	0	1
N4	0	1	1	0

Table B.6 4-Node Network - Model 6

Node	N1	N2	N3	N4
N1	0	1	1	0
N2	1	0	0	0
N3	1	0	0	1
N4	0	0	1	0

Table B.7 4-Node Network - Model 7

Node	N1	N2	N3	N4
N1	0	1	1	0
N2	1	0	0	1
N3	1	0	0	1
N4	0	1	1	0

Table B.8 4-Node Network - Model 8

Node	N1	N2	N3	N4
N1	0	0	1	1
N2	0	0	1	0
N3	1	1	0	0
N4	1	0	0	0

Table B.9 4-Node Network - Model 9

Node	N1	N2	N3	N4
N1	0	0	0	1
N2	0	0	1	0
N3	0	1	0	1
N4	1	0	1	0

Table B.10 4-Node Network - Model 10

B.2 14-Node Network Models

The 10 different 14-node network models that we selected for our testing purpose are given in Tables B.11 to B.20.

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	0	0	0	1	0	0	0	0	0	0	1	0	0
N2	0	0	0	1	1	0	1	1	0	1	0	0	0	0
N3	0	0	0	1	0	0	0	0	0	0	0	0	0	1
N4	0	1	1	0	1	1	1	0	0	0	1	1	0	0
N5	1	1	0	1	0	1	0	0	0	0	0	0	0	0
N6	0	0	0	1	1	0	0	0	1	0	1	0	0	0
N7	0	1	0	1	0	0	0	0	1	1	0	0	0	0
N8	0	1	0	0	0	0	0	0	0	1	0	0	0	0
N9	0	0	0	0	0	1	1	0	0	1	0	0	1	0
N10	0	1	0	0	0	0	1	1	1	0	0	1	0	0
N11	0	0	0	1	0	1	0	0	0	0	0	0	0	1
N12	1	0	0	1	0	0	0	0	0	1	0	0	1	1
N13	0	0	0	0	0	0	0	0	1	0	0	1	0	0
N14	0	0	1	0	0	0	0	0	0	0	1	1	0	0

Table B.11 14-Node Network - Model 1

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	1	1	1	1	0	0	0	0	0	0	1	0	0
N2	0	0	1	1	1	0	1	1	0	1	0	0	0	0
N3	0	0	0	1	0	0	0	0	0	0	0	0	0	1
N4	0	1	1	0	1	1	1	0	0	0	1	1	0	0
N5	1	1	0	1	0	1	0	0	0	0	0	0	0	0
N6	0	0	0	1	1	0	0	0	1	0	1	0	0	0
N7	0	1	0	1	0	0	0	0	1	1	0	0	0	1
N8	0	1	0	0	0	0	0	0	0	1	0	0	0	0
N9	0	0	0	0	0	1	1	0	0	1	0	0	1	0
N10	0	1	1	0	0	0	1	1	1	0	0	1	0	0
N11	0	0	0	1	0	1	0	0	0	0	0	0	0	1
N12	1	0	0	1	0	0	0	0	0	1	0	0	1	1
N13	0	0	0	0	0	0	0	0	1	0	0	1	0	0
N14	0	1	1	0	0	0	0	0	0	0	1	1	0	0

Table B.12 14-Node Network - Model 2

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	0	0	1	1	0	0	0	0	0	0	1	0	0
N2	0	0	1	1	1	0	1	1	0	1	0	0	0	0
N3	0	1	0	1	0	0	0	0	0	0	0	0	0	1
N4	1	1	1	0	1	1	1	0	0	0	1	1	0	0
N5	1	1	0	1	0	1	0	0	0	0	0	0	0	0
N6	0	1	0	1	1	0	0	0	1	0	1	0	0	0
N7	0	1	0	1	0	0	0	0	1	1	0	0	0	1
N8	0	1	0	0	0	0	0	0	0	1	0	0	0	0
N9	0	0	0	0	0	1	1	0	0	1	0	0	1	0
N10	0	1	1	0	0	0	1	1	1	0	0	1	0	0
N11	0	0	0	1	0	1	0	0	0	0	0	0	0	1
N12	1	0	0	1	0	0	0	0	0	1	0	0	1	1
N13	0	1	0	0	0	0	0	0	1	0	0	1	0	0
N14	0	1	1	0	0	0	0	0	0	0	1	1	0	0

Table B.13 14-Node Network - Model 3

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	0	0	1	1	0	0	1	0	0	1	0	1	0
N2	0	0	1	1	1	0	1	1	0	1	0	0	0	0
N3	0	1	0	1	0	0	0	0	0	0	0	0	0	0
N4	1	1	1	0	1	1	1	0	0	0	1	1	0	0
N5	1	1	0	1	0	1	0	0	0	0	0	0	0	0
N6	0	0	0	1	1	0	0	0	1	0	1	0	0	0
N7	0	0	0	1	0	0	0	0	1	1	1	0	0	1

N8	0	0	0	0	0	0	0	0	0	1	1	0	0	0
N9	0	0	0	0	0	1	1	0	0	1	0	0	1	0
N10	0	1	1	0	0	0	1	1	1	0	0	1	0	0
N11	0	0	0	1	0	1	0	0	0	0	0	0	0	1
N12	1	0	0	1	0	0	0	0	0	1	0	0	1	1
N13	0	1	0	0	0	0	0	0	1	0	0	1	0	0
N14	0	1	1	0	0	0	0	0	0	0	1	1	0	0

Table B.14 14-Node Network - Model 4

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	0	0	1	0	0	0	1	0	0	0	0	1	1
N2	0	0	1	1	1	0	1	1	1	1	0	0	0	1
N3	0	1	0	1	0	0	0	0	0	0	0	0	0	1
N4	1	1	1	0	1	1	1	0	0	0	1	1	0	0
N5	0	1	0	1	0	1	0	0	0	0	0	0	0	0
N6	0	0	0	1	1	0	0	0	1	0	1	0	0	0
N7	0	0	0	1	0	1	0	0	1	1	1	0	0	1
N8	0	0	0	0	0	0	0	0	0	1	1	0	0	0
N9	0	0	0	0	0	1	1	0	0	1	0	0	1	0
N10	0	1	1	0	0	0	1	1	1	0	0	1	0	0
N11	0	0	0	0	0	1	0	0	1	0	0	0	0	1
N12	1	0	0	0	0	0	0	0	1	1	0	0	1	1
N13	0	1	0	0	0	0	0	0	1	0	0	1	0	0
N14	0	1	0	0	0	0	0	1	0	0	0	1	0	0

Table B.15 14-Node Network - Model 5

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	0	0	0	1	0	0	0	0	0	0	1	0	0
N2	0	0	0	1	1	0	1	1	0	1	0	0	0	0
N3	0	0	0	1	0	0	0	0	0	0	0	0	0	1
N4	0	1	1	0	1	1	1	0	0	0	1	1	0	0
N5	1	1	1	1	0	1	0	0	0	0	0	0	0	0
N6	1	0	1	1	1	0	0	0	1	0	1	0	0	0
N7	1	1	1	1	0	0	0	0	1	1	0	0	0	0
N8	1	1	1	0	0	0	0	0	0	1	0	0	0	0
N9	1	0	1	0	0	1	1	0	0	1	0	0	1	0
N10	1	1	1	0	0	0	1	1	1	0	0	1	0	0
N11	1	0	0	1	0	1	0	0	0	0	0	0	0	1
N12	1	0	0	1	0	0	0	0	0	1	0	0	1	1
N13	0	0	0	0	0	0	0	0	1	0	0	1	0	0
N14	0	0	1	0	0	0	0	0	0	0	1	1	0	0

Table B.16 14-Node Network - Model 6

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	0	0	0	1	0	0	0	0	0	0	1	0	0
N2	0	0	0	1	1	0	1	1	0	1	0	0	0	0
N3	0	0	0	1	0	0	0	0	0	0	0	0	0	1
N4	0	1	1	0	1	1	1	0	0	0	1	1	0	0
N5	1	1	0	1	0	1	0	0	0	0	0	0	0	0
N6	0	1	0	1	1	0	0	0	1	0	1	0	0	0
N7	0	1	0	1	0	0	0	0	1	1	0	0	0	0
N8	0	1	0	0	0	0	1	0	0	1	0	0	0	0
N9	0	1	0	0	0	1	1	0	0	1	0	0	1	0
N10	0	1	0	0	0	0	1	1	1	0	0	1	0	0
N11	0	1	0	1	0	1	1	0	0	0	0	0	0	1
N12	1	1	0	1	0	0	1	0	0	1	0	0	1	1
N13	0	1	0	0	0	0	1	0	1	0	0	1	0	0
N14	0	1	1	0	0	0	1	0	0	0	1	1	0	0

Table B.17 14-Node Network - Model 7

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	0	0	0	1	0	1	0	0	1	0	1	0	0
N2	0	0	0	1	1	0	1	1	0	1	0	1	0	0
N3	0	0	0	1	0	0	0	0	0	0	0	1	0	1
N4	0	1	1	0	1	1	1	0	0	0	1	0	0	0
N5	1	1	0	1	0	1	0	0	0	0	0	0	0	0
N6	0	0	0	1	1	0	0	0	1	0	1	0	0	0
N7	0	1	0	1	0	0	0	0	1	0	0	0	0	0
N8	0	1	0	0	0	0	0	0	0	0	0	1	0	0
N9	0	0	0	0	0	1	1	0	0	0	0	1	1	0
N10	0	1	0	0	0	0	1	1	1	0	0	0	0	0
N11	0	0	0	1	0	1	0	0	0	0	0	0	0	0
N12	1	0	0	1	0	0	0	0	0	1	0	0	1	0
N13	0	0	0	0	0	0	0	0	1	1	0	0	0	0
N14	0	0	1	0	0	0	1	0	0	0	1	0	0	0

Table B.18 14-Node Network - Model 8

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	1	0	0	1	1	0	0	1	1	0	1	1	0
N2	0	0	0	1	1	0	1	0	0	1	0	0	1	0
N3	0	0	0	1	0	0	0	0	0	0	0	0	1	1
N4	0	1	1	0	1	1	1	0	0	0	0	1	1	0
N5	1	1	0	1	0	1	0	0	0	0	0	0	0	0
N6	0	0	0	1	1	0	0	0	1	0	1	0	0	0
N7	0	1	0	1	0	0	0	0	1	0	0	0	0	0

N8	0	1	0	0	0	0	0	0	0	1	0	0	0	0
N9	0	0	0	0	0	1	1	0	0	0	0	0	0	0
N10	0	1	0	0	0	0	1	1	0	0	1	1	0	0
N11	0	0	0	1	0	1	0	0	0	0	0	0	0	1
N12	1	0	0	1	0	0	0	1	0	1	1	0	0	1
N13	0	0	0	0	0	0	0	0	1	1	0	1	0	0
N14	0	0	1	0	0	0	0	1	0	0	0	1	1	0

Table B.19 14-Node Network - Model 9

Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
N1	0	0	0	0	1	0	0	0	1	0	1	1	1	0
N2	0	0	0	1	1	0	1	1	0	1	0	0	0	0
N3	0	0	0	0	0	0	0	0	0	0	0	0	1	1
N4	0	1	1	0	1	1	1	1	0	0	1	1	0	0
N5	1	0	0	1	0	0	0	0	0	0	0	0	1	0
N6	0	1	1	0	1	0	0	0	1	0	1	0	0	0
N7	0	0	0	1	0	0	0	0	1	1	0	0	1	0
N8	0	0	1	0	0	0	0	0	0	0	0	0	0	0
N9	0	1	0	1	0	1	1	0	0	1	0	0	1	0
N10	0	0	0	0	0	0	0	1	1	0	0	1	0	0
N11	0	1	0	1	0	1	0	0	0	0	0	0	0	1
N12	1	0	0	1	1	0	1	1	0	1	0	0	1	1
N13	0	1	0	0	0	1	0	0	0	0	0	1	0	0
N14	0	0	0	1	1	0	0	0	0	0	1	1	0	0

Table B.20 14-Node Network - Model 10

References

[1]. Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. ACM SIGMETRICS, 31(1):206–217, 2003.

[2]. Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information theoretic approach to traffic matrix estimation. In SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pages 301–312, Karlsruhe, Germany, August 2003.

[3]. A. E. Conway and M. Li. Estimating the traffic distribution matrix of a packet-switched network using aggregate traffic measurements. Telecommunication Systems, 24(1):9–28, 2003.

[4]. M. A. Saunders and B. Kim. Pdsco: Primaldual interior method for separable convex objectives. http://www.stanford.edu/group/SOL/software/pdsco.

[5]. P. C. Hansen. Regularization tools (for matlab). http://www.imm.dtu.dk/pch/Regutools/index.html.

[6]. A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, pages 161–174, Pittsburg, USA, August 2002.

[7]. J. Cao, D. Davis, S. V. Wiel, and B. Yu. Time-varying network tomography. American Statistical Association, 95:1063–1075, 2000.

[8]. C. Tebaldi and M. West. Bayesian inference on network traffic using link count data. American Statistical Association, 93:557–576, 1998.

[9]. Y. Vardi. Network tomography: estimating source-destination traffic intensities from link data. American Statistical Association, 91:365–377, 1996.

[10]. Md. Mostafizur Rahman, Subrata Saha, Usha Chengan and Attahiru Sule Alfa: IP Traffic Matrix Estimation Methods: Comparisons and Improvements. ICC, 2006.

[11]. M. A. Saunders and B. Kim. Pdsco: Primaldual interior method for separable convex objectives. http://www.stanford.edu/group/SOL/software/pdsco.

[12]. P. C. Hansen. Regularization tools (for matlab). http://www.imm.dtu.dk/pch/Regutools/index.html.

[13]. O. Goldschmidt. ISP Backbone Traffic Inference Methods to Support Traffic Engineering . In Internet Statistics and Metrics Analysis (ISMA) Workshop, San Diego, CA, December 2000.

[14]. J. Cao, D. Davis, S. Vander Weil, and B. Yu. Time-Varying Network Tomography. J. of the American Statistical Association., 2000.

[15]. C. Tebaldi and M. West. Bayesian Inference of Network Traffic Using Link Count Data. J. of the American Statistical Association., pages 557–573, June 1998.