Survey traffic matrix for optimizing network performance

Tran Cong Hung, Nguyen Khoi, Nguyen Xuan Phi, *Member, IEEE*

***Abstract*--Traffic matrix has many applications in different areas, that plays an important role in administrating computer networks. With traffic matrix as input, we can calculate to solve problems of our computer networks such as bandwidth utilization, load balancing, improving quality of service… So that, this paper will analyze estimation techniques and applications of traffic matrix into our computer networks.**

***Index Terms*--Traffic matrix, load balancing, shortest path first, bandwidth utilization, quality of services**.

I. INTRODUCTION

With rapid growth of the internet and the accompanying traffic, network traffic measurement plays an ever critical role in how network service providers and operators manage and plan network operations. For instance, the rise of data centers and emergence of cloud computing are making this measurement more complex, where content or service providers employ load balancing to dynamically adapt to user demands. Understanding the flow of traffic in such networks will help in improving the operations, management and security of today's IP networks as well as emerging services.

Traffic matrix (TM) – which represents the flow of data from each ingress point to each egress point through a network (we call that source-destination(SD) pair) – is an important piece of information needed to plan, manage and understand any networks. Unfortunately, direct measurements require expensive additional infrastructure support it can be prohibitively to instrument the entire IP network to collect such data. Many methods have been introduced to obtain the traffic matrix by estimation techniques that give us most accurate results compare to practical traffic matrix.

We can form a system $Y = AX$ where Y is link counts, A is routing matrix, X is traffic matrix. In that system, we know Y from SNMP data, we know A from routing policies, all we have to do is to solve the system to find X.

Manuscript received September 11th, 2013. Accepted: October 25th, 2013.

Tran Cong Hung, Assoc.Prof.Ph.D. is with the Posts & Telecommunications Institute of Technology, Vietnam (email: conghung@ptithcm.edu.vn).

Nguyen Khoi, Eng. is with the Posts & Telecommunications Institute of Technology, Vietnam (email: nguyenkhoi0909@gmail.com)

Nguyen Xuan Phi, M.Eng. is with the Posts & Telecommunications Institute of Technology, Vietnam (email: phinguyenxuan@agribank.com.vn)

This paper is divided into 5 sections: section 1 introduces about traffic matrix and its applications, section 2 presents the related works, section 3 introduces about estimation techniques and routing problems, section 4 is our experimental results and evaluation, section 5 is our conclusion and future work.

II. RELATED WORKS

Many researches have been done about TM in order to estimate TM more precisely. Many techniques have been introduced in [1], [2], [3], [4] and the results are applied to routing as in [5], [6], [7], [8], [9], [10], [11], [12] to optimizing network performance. However, traffic demands change all the time, we need to find techniques satisfying calculating time and preciseness.

III. ESTIMATION TECHNIQUES AND ROUTING PROBLEMS

A. Linear Programming (LP)[1]

Because the traffic matrix estimation problem imposes a set of linear relationships described by the system $Y = AX$, the basic problem can be easily formulated using a LP model and standard techniques can be used to solve it. Knowing that the link count Y_l has to be the sum of all the traffic demands that use link l, the LP model is defined as the optimization of an objective function:

$$max \ = \sum_{j=1}^{m} w_j \, X_j$$

where w_j is a weight for SD pair j. The objective function is subject to link constraints:

$$\sum_{j=1}^{c} A_{lj} X_j \ \le \ Y_l \ \ l \ = \ 1, \dots, r$$

and flow conservation constraints:

$$\sum_{l=(i,j)} Y_l a_{lk} \ - \sum_{l=(j,i)} Y_l \, a_{lk} \ = \begin{cases} X_k & \text{if } j = \text{src of } k \\ -X_k & \text{if } i = \text{dst of } k \\ 0 & \text{otherwise} \end{cases}$$

and positivity constraints $X_j \ge 0 \ \forall_j$

If a function that is the linear combination of all the demands is used, that means trying to maximize the load carried on the network, it will lead to solutions in which short SD will be assigned very large values of bandwidth while distant SD pairs will often be assigned zero flow. Although such solutions are feasible, these are not the aimed ones.

B. Statistical Approaches

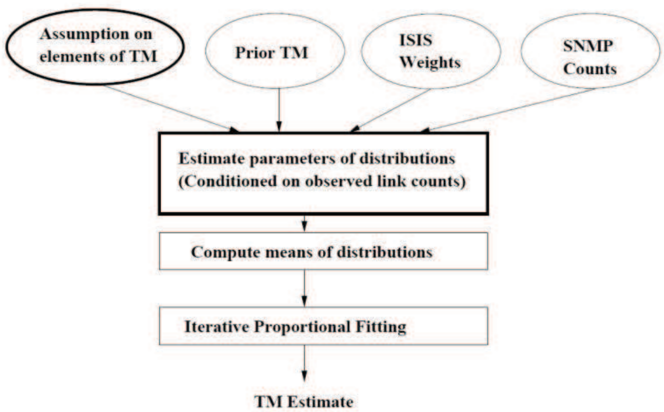

Fig. 1. General diagram for statistical approach [1], [2]

There are four general inputs to the statistical approaches. Although the assumptions made on the traffic demands are not actually an input, they may be seen as influencing the specific statistical strategy to use. Statistical methods usually need a prior TM to get started. This important input may come from an outdated version of the TM, or an initial estimate obtained by some other mechanism. The ISIS weights are used to compute shortest paths which in turn generate the *A* matrix. The final input, SNMP data, gives the observed links counts *Y*. These inputs are used to impose constraints on the estimated TM.

Given the inputs, the first and main step of the estimation procedure is to estimate all the parameters of the distributions assumed for the TM components. This typically involves estimating Λ where $\Lambda = \{\lambda_1, \ldots, \lambda_m\}$, denotes the vector of mean rates (i.e., each λ_j denotes the mean rate of SD pair X_j). Once the parameters are obtained, the next step is to compute the conditional mean value for the distribution associated with each component of the TM. A final adjustment step is usually applied to the result from the previous step corresponds to an iterative proportional fitting algorithm (IPF). The IPF algorithm proceeds to adjust the values of the estimated traffic matrix such that the error with respect to the row and column sums is minimized.

C. Gravity Modeling [2], [3]

Gravity models, taking their name from Newton's law of gravitation, are commonly used by social scientists to model the movement of people, goods or information between geographic areas. In Newton's law of gravitation the force is proportional to the product of the masses of the two objects divided by the distance squared. Similarly, in gravity models for cities, the relative strength of the interaction between two cities might be modeled as proportional to the product of the populations. A general formulation of a gravity model is given by the following equation:

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

where X_{ij} is the matrix element representing the force from i to j; R_i represents the repulsive factors that are associated with "leaving" from i; A_j represents the attractive factors that are associated with "going" to j; and f_{ij} is a friction factor from i to j.

In our context, we can naturally interpret X_{ij} as the traffic volume that enters the network at location i and exits at location j, the repulsion factor R_i as the traffic volume entering the network at location i, and the attractive factor A_j as the traffic volume exiting at location j. The friction matrix (f_{ij}) encodes the locality information specific to different SD pairs. The inference friction factors is an equivalent problem of the same size as the inference of the TM itself. Accordingly, it is necessary to approximate the actual friction matrix using models with fewer parameters. A common constant for the friction factors, which is arguably the simplest among all possible approximation schemes, will be assumed. The resulting gravity model simply states that the traffic exchanged between locations is proportional to the volumes entering and exiting at those locations. One possible explanation for this is that geographic locality is not a major factor in today's Internet, as compared to ISP routing policies

1. Simple Gravity Model

In this very simple gravity model, M. Ericsson, M. Resende, and P. Pardalos aim to estimate the amount of traffic between edge links. Denote the edge links by l_1, l_2, ... they estimate the volume of traffic $T(l_i, l_j)$ that enters the network at edge link l_i and exits at edge link l_j. Let $T^{in}_{link}(l_i)$ denote the total traffic that enters the network via edge link l_i, and $T^{out}_{link}(l_j)$ denotes the corresponding quantity for traffic that exits the network via edge link l_i. The gravity model can then be computed by either of

$$T(l_i, l_j) = T^{in}_{link}(l_i)\frac{T^{out}_{link}(l_j)}{\sum_k T^{out}_{link}(l_k)} \text{ or}$$

$$T(l_i, l_j) = \frac{T^{in}_{link}(l_i)}{\sum_k T^{in}_{link}(l_k)} T^{out}_{link}(l_j)$$

The first equation states that the traffic matrix elements $T(l_i, l_j)$ are the product of the traffic entering the network via edge link l_i and the proportion of the total traffic leaving the network via edge link l_j, while the second is reversed and is identical under traffic conservation – that is, the assumption that the interior network is neither a source, nor sink of traffic.

2. Generalized Gravity Model

M. Ericsson, M. Resende, and P. Pardalos develop the equations for a gravity model under the following additional assumptions, which reflect dominant Internet and ISP routing policies:

Transit peer (peering link to peering link) traffic: They assume that the volume of traffic that transits across the backbone from one peer network to another is negligible.

Outbound (access link to peering link) traffic: They apply the proportionality assumption underlying gravity modeling on a peer-by-peer basis: that is, the traffic exiting to a specific peer comes from each access link in proportion to the traffic on that access link. They assume that all of the traffic from a single access link to the given peer exits the network on the same peering link (determined by the IGP and BGP routing configuration).

Inbound (peering link to access link) traffic: A network operator has little control over the injection of traffic into its network from peer networks. Accordingly, they assume that the traffic entering from a given peering link is split amongst the access links in proportion to their outbound traffic.

Internal (access link to access link) traffic: They assume that the fraction of internal traffic from a given access link a_i to a second access link a_j is proportional to the total traffic entering the network at a_i, and compute the traffic between the links by normalization.

D. Tomography [2], [3], [4]

Network tomography is the problem of determining the end-to-end traffic matrix from link loads. The link traffic is the sum of the traffic matrix elements that are routed across that link, and so M. Ericsson, M. Resende, and P. Pardalos see the problem as follows: they have a set of observables $Y = (y_1, y_2, \dots y_L)^T$, the traffic (as measured in packets or bytes) that traverses the L links of the network during some period, written as a column vector $X = (x_1, x_2, \dots, x_m)^T$. According to system $Y = AX$ they have a matrix A[L,m] = {aij} called routing matrix which defined as:

$$a_{ij} = \begin{cases} f_{ij}, & \text{if traficc for } j \text{ traverse link } i \\ 0, & \text{otherwise} \end{cases}$$

They need to solve the inverse problem to obtain x. For general topologies and routing there are typically many more unknowns than constraints, and so Y = AX is highly under-constrained and does not have a unique solution. Their approach is not to incorporate additional constraints, but rather to use the gravity model to obtain an initial estimate of the solution, which needs to be refined to satisfy the constraints. It is important to reduce the size of the problem to make computation of the solution more manageable.

E. Tomogravity [3], [4]

Tomogravity is the combination of gravity model and tomography to exploit strong points of both gravity model and tomography.

Step 1: calculate vector Traffic matrix $X_g = (x_{g1}, x_{g2}, \dots, x_{gm})^T$ from general gravity model.

Step 2: solve system Y = AX by tomography technique to find X_0 subject to $\|X_0 - X_g\|$ min (least square solution)

$$\|X_0 - X_g\| = \sqrt{(x_{01} - x_{g1})^2 + (x_{02} - x_{g2})^2 + \cdots + (x_{0m} - x_{gm})^2} = \sqrt{\sum_{j=1}^{m} (x_{0j} - x_{gj})^2}$$

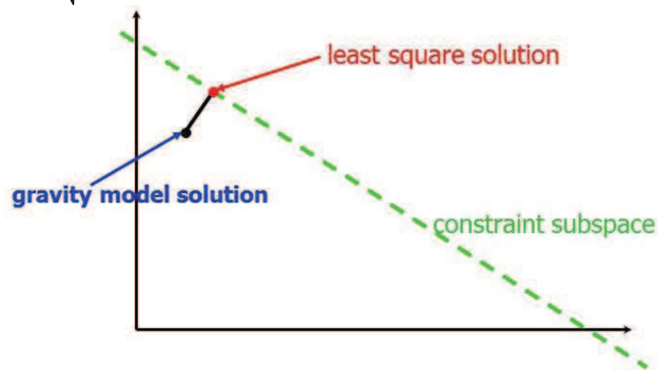

Fig. 2. An illustration of the least-square solution [2], [3]

To minimize distance of X_0 to X_g, singular value decomposition can be used to solve the quadratic program subject to the tomographic constraints. But the result may contain negative values so that negative values will be replaced with zero and then perform IPF to obtain non-negative solution that satisfies the constraints.

F. Routing

<u>*Destination-Based vs. Source/Flow-Based Routing*</u>

Two fundamentally different routing concepts exist, which strongly influence the optimization procedure and the achievable results: destination-based routing and source- or flow-based routing. Conventional routing protocols such as OSPF, EIGRP, or IS-IS, follow the next-hop destination-based routing paradigm. Within each router the forwarding decision for an IP packet is based solely on the destination address specified in the packet header. A router looks up the prefix of the destination address in its routing table, determines the outgoing interface, and sends the packet to the appropriate neighbor. No information about the source or any other context of the packet is taken into account. As a consequence, this routing procedure is simple and quite efficient. However, it imposes limitations on routing optimization, as illustrated in Fig. 2. Whenever two traffic flows with the same destination cross each other's way they are merged and sent out over the same interface. This might cause traffic overload on some links, while other links are still only lightly utilized.

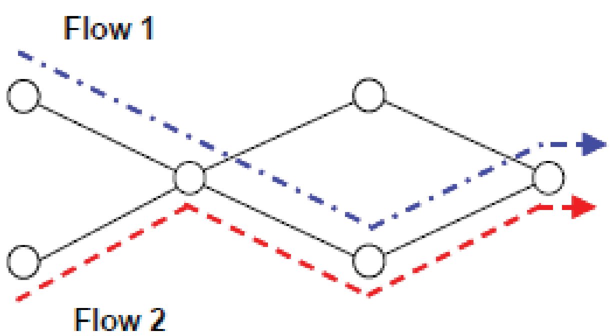

Fig. 3. Limitations of destination-based routing [5], [6]

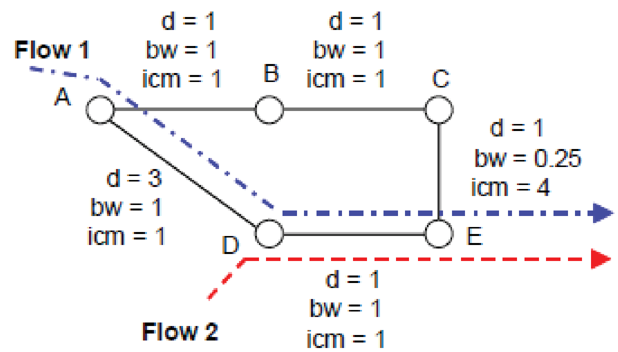

Fig. 4. Multiple-metric routing [5], [6]

Single-Metric vs. Multiple-Metric Routing

In the case of destination-based routing protocols a router determines an outgoing interface based on metric values, which quantitatively describe the distance to a destination node. Most commonly, single additive metrics are assigned to every link, and a shortest-path algorithm is used to determine the preferred path from each node to every other node in the network ("single-metric routing"). While link metrics often have physically relevant meanings such as "propagation delay" or "cost", they can also be used in a generic way purely for the sake of routing optimization. By setting appropriate link metric values, one can implicitly influence and, thus, optimize the routing scheme. In addition to single-metric protocols, routing schemes exist, which allow more than one metric taken into account when computing the length of a path towards a destination node ("multiple-metric routing"). One example is Cisco's routing protocol EIGRP, which incorporates four metric types. However, only two of them are used by default: one additive metric ("delay") and one concave metric ("bandwidth"). The distance to a destination node is now computed by the normalized metric formula.

$$M = \frac{1}{\min_i(bw_i)} + \sum_i d_i = \max_i(icm_i) + \sum_i d_i$$

Parameter bw_i denotes the bandwidth of a link i, while d_i refers to its delay value. Thus, a router takes the sum of all delay values towards the destination node and adds a bandwidth component, which is the inverse of the smallest bandwidth along the path ("bottleneck"). From all possible path options it selects the one with smallest path metric M.

For further considerations we will refer to the bandwidth component as "inverse capacity metric" (*icm)* and take the maximum along the path instead of the reciprocal value of the bandwidth minimum. Fig. 3 illustrates the concept of bandwidth-delay sensitive routing. If only the delay metrics *d* were taken into consideration, flow 1 would take the upper path along nodes B-C-E. However, link C-E has a smaller normalized bandwidth of 0.25 and, therefore, contributes to M with an inverse capacity metric of 4. Thus, the cost value associated with path A-B-C-E is 7 (delay sum of 3 plus bandwidth component of 4), while path A-D-E has only an overall metric of 5. Therefore, router A would choose router D as its next-hop neighbor.

Routing optimization based on the multiple-metric concept has some advantages over the pure shortest-path approach, as can be demonstrated on the network scenario in Fig. 4.

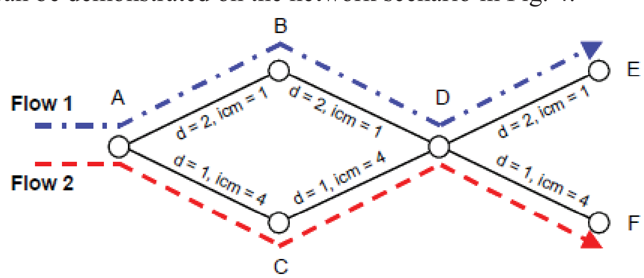

Fig. 5. Fish-pattern routing with multiple metrics [5], [6]

Assume we have two traffic flows with different destinations, whose paths have several nodes in common. Let A be the first node where the two flows come together and D be last common node on their way. While shortest path routing would merge the flows at node A and send both of them either over B or over C, multiple-metric routing protocols can achieve the flow pattern given in the Fig. 4. For flow 1, the chosen path has a total metric of 7, while the link metrics along the route via C would sum up to 8. For flow 2 the situation is different. The total metrics of the upper and the lower path are 9 and 7, respectively. The trick is to use the inverse capacity metric to make one path option appear more costly for one traffic flow, while for the other flow a larger *icm* value has no extra effect (since it experiences already high *icm* values on other links along the path, which the two flows do not share).

Equal-Cost Multi-Path (ECMP) [5]

Another possible feature of routing protocols, which influences the optimization process, is load sharing. In destination-based routing protocols this capability is often implemented in form of the "equal-cost multi-path" concept. Whenever a router can reach a destination node via several paths with equal metric sums, it splits up the traffic evenly across all corresponding outgoing interfaces.

G. Optimizing network - reducing link utilization

Link utilization [7], [8], [9], [10], [11]

The network congestion ratio, which refers to the maximum value of all link utilization rates in the network, is denoted as *r*. Rate of each link utilization is defined as:

$$r_{ij} = \frac{l_{ij}}{c_{ij}} \; where \; i,j \in V$$

where V is set of nodes in the network and c_{ij} is capacity of link (*i,j*), l_{ij} is traffic traversing on link (*i,j*).

Minimizing *r* means that admissible traffic is maximized. The admissible traffic volume is accepted up to the current traffic volume multiplied by 1/*r*. Minimizing *r* with routing control is the objective of link utilization.

Reducing link utilization [9], [10], [11], [12]

ISPs have SLA (Service Level Agreement) that guarantees bandwidth for leased lines they provide, it is also quality of service that ISPs have to ensure. When a network administrator distributes link load on links he has to make sure that there is not bottle neck in his network that means he can guarantee SLA.

In common network topology, OSPF is usually used, traffic flows traverse on shortest paths to their destinations. The shortest paths are determined by weights, the paths have the least value of weight is the shortest paths. This also makes the shortest path become most heavy traffic path, it impacts our network performance.

Link utilization can reduced by balancing on shortest paths. One of solutions is using value k_m^{pq}, that is rate of distributing traffic demands of SD pair (*p,q*) through node *m* and the constraint is

$$k_m^{pq} \geq 0 \; và \sum k_m^{pq} = 1 \; where \; p,q,m \in V$$

When there are values of k_m^{pq}, link utilization can be calculated on each link with the input is traffic matrix. To find set of k_m^{pq} a linear problem must be solved with constraints of link capacity and traffic demands from traffic matrix (estimated by above techniques). However, this is a hard to solve because number of SD pairs is much more than number of nodes of network which leads to set of k_m^{pq} has too many elements.

One of solutions is to use Two Phase Load Balanced Routing. First, traffic demands from source are balanced over intermediate nodes then traffic demands will be carried on shortest paths to destinations.

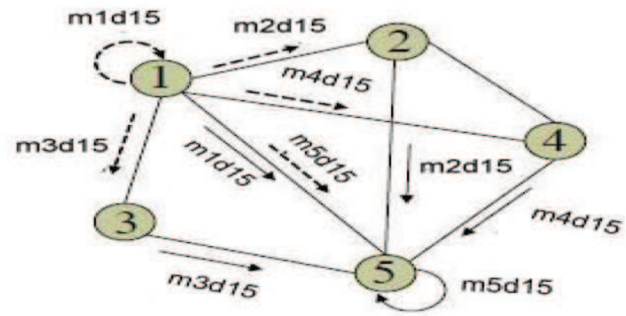

Fig. 6. Two Phase Load Balanced Routing [12]

In Fig. 5 traffic demand from node 1 to node 5 is balanced over intermediate nodes 1, 2, 3, 4, 5 at phase one (dash arrow). At phase two, these traffic flow are carried to node 5 on shortest paths (continuous arrow).

So, with traffic matrix as input, network performance can be improved, that means optimizing network.

IV. EXPERIMENTAL RESULTS AND EVALUATION

In this section we have tested our solution on different network topologies. To test the result we use Matlab[12] and link counts are randomly generated. We simulated the situation of very heavy traffic network and we see that link utilization is better reduced for smaller topology.

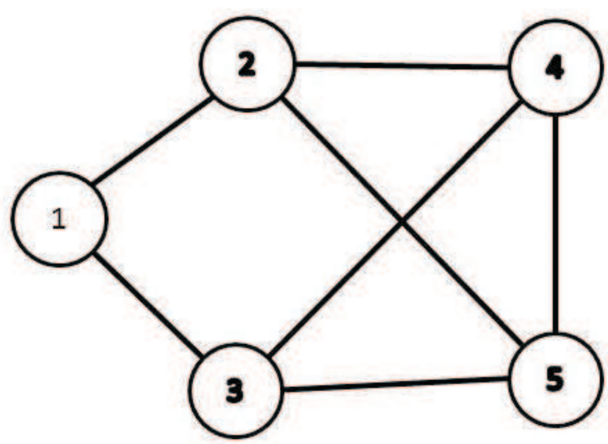

Fig. 7. Five-node topology

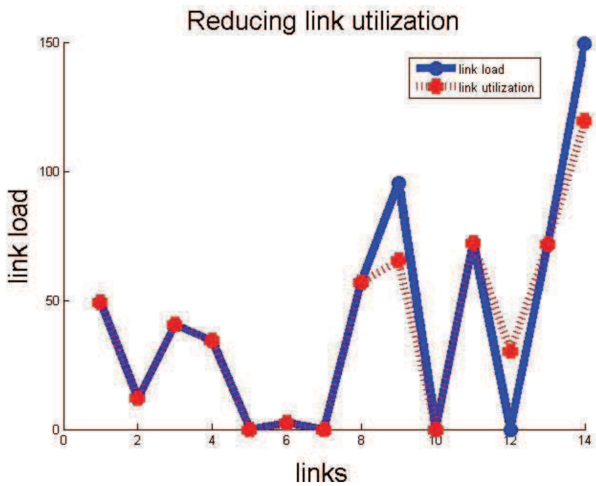

Fig. 8. Optimizing five-node topology

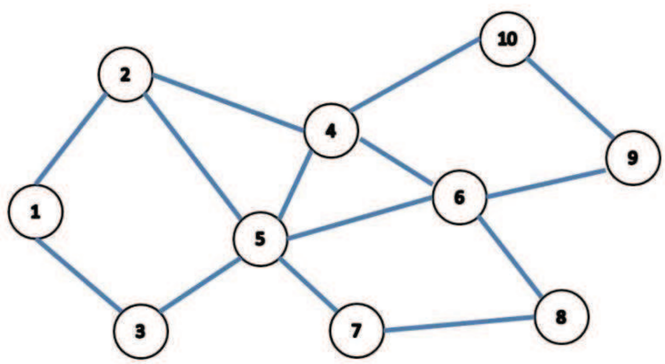

Fig. 9. Ten-node topology

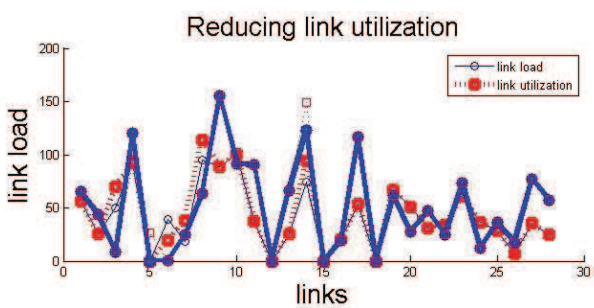

Fig. 10. Optimizing ten-node topology

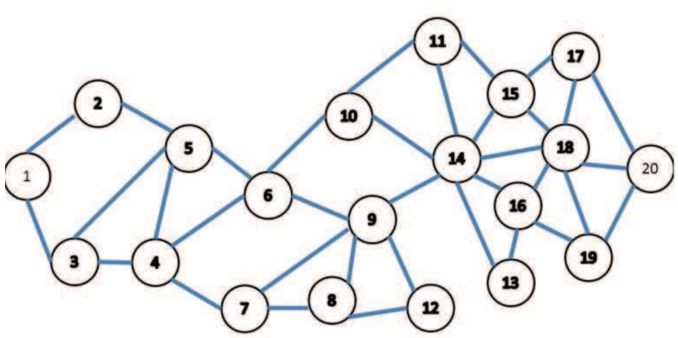

Fig. 11. Twenty-node topology

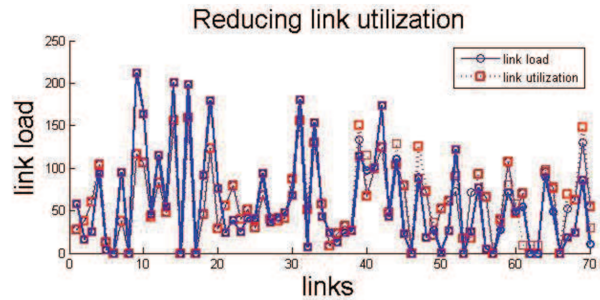

Fig. 12. Optimizing twenty-node topology

V. CONCLUSION

This paper suggest solution optimizing network performance with traffic matrix as input. When estimating traffic matrix we have to accept error from estimated traffic matrix to real one so to possibly get the most accurate traffic matrix we have to improve estimation techniques. Our experimental results show that the larger topology is the less link utilization reduced, so that we will also have to improve our solution such as balancing.

The results we have had with large networks, which have many links connecting nodes, show that we have used our network resource better. We see that links' capacity used better. When a link has much heavy traffic, the traffic is shared for other links and the network performance is raised. However, sometimes we faced situations that traffic could not be shared because other links could not get more traffic.

Our research aims to solve problem of heavy traffic for ISPs and we can collect better and more practical results if we have dataset of traffic from real network. We intend to combine method in [14] and [15] with our solution to minus errors in estimating traffic matrix and raise performance for extreme large network.

REFERENCE

[1] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot (2002), "Traffic matrix estimation: Existing techniques and new directions" *Proc. ACM SIGCOMM*.

[2] Yin Zhang, Matthew Roughan, Nick Duffield, Albert Greenberg (2003), "Fast Accurate Computation of Large Scale IP Traffic Matrices from Link Loads".

[3] M. Ericsson, M. Resende, and P. Pardalos (2002), "A genetic algorithm for the weight setting problem in OSPF routing" *Journal Combinatorial Optimization*, pp. 299-333.

[4] J. Cao, D. Davis, S. Vander Weil, and B. Yu (2000), "Time-varying network tomography" *Journal of the American Statistical Association*.

[5] M.G.C. Resende et al (2002), "A memetic algorithm for the weight setting problem in OSPF routing" *6th INFORMS Telecommunications Conference*.

[6] A. Riedl (2002), "A hybrid genetic algorithm for routing optimization in IP networks utilizing bandwidth and delay metrics" *Proceedings of IEEE Workshop on IP Operations and Management (IPOM)*.

[7] B. Fortz, M. Thorup (2000), "Internet traffic engineering by optimizing OSPF weights" *Proceeding of INFOCOM 2000*.

[8] Ritesh Dayama (2011), "Load balanced routing using OSPF" *International Journal on Computer Science and Engineering (IJCSE)*.

[9] Thomas Erlebach and Maurice Riiegg (2004), "Optimal bandwidth reservation in hose-model VPNs with multi-path routing" *IEEE INFOCOM*.

[10] M.Roughan, M. Thorup, and Y. Zhang (2003), "Traffic engineering with estimated traffic matrices" *Proceedings of the 1st ACM/USENIX Internet Measurement Conference (IMC '03)*.

[11] Eiji Oki and Ayako Iwaki (2009), "Fine two-phase routing with traffic matrix" *Computer Communications and Networks*.

[12] Marija Antic, Natasa Maksic, Petar Knezevic and Aleksandra Smiljanic (2010), "Two phase load balanced routing suing OSPF" *IEEE journal on selected areas in communications*.

[13] http://www.mathworks.com

[14] Vijiay Kumar Adhikari, Sourabh Jain, and Zhi-li Zhang (2011), “From traffic matrix to routing matrix: PoP level traffic characteristics for a Tier-1 ISP

[15] A. Omidvar and H. S. Shahhoseini (2011), “Intelligent IP traffic matrix estimation by neural network and genetic algorithm”

AUTHORS

TRAN CONG HUNG was born in Vietnam in 1961

He received the B.E in electronic and Telecommunication engineering with first class honors from HOCHIMINH university of technology in Vietnam, 1987.

He received the B.E in informatics and computer engineering from HOCHIMINH university of technology in Vietnam, 1995.

He received the master of engineering degree in telecommunications engineering course from postgraduate department Hanoi university of technology in Vietnam, 1998.

He received Ph.D. at Hanoi university of technology in Vietnam, 2004.

His main research areas are B – ISDN performance parameters and measuring methods, QoS in high speed networks, MPLS.

He is, currently, Associate Professor Ph.D. of Posts and Telecoms Institute of Technology in HOCHIMINH, Vietnam.

NGUYEN KHOI was born in Vietnam in 1987

Obtained BE in Information Technology from Post and Telecommunication Institute of Technology (PTIT), Vietnam, 2010.

Will receive Master in PTIT, 2013, major in Networking and Data Transmission.

He is, currently, Network Administrator at Metro Cash & Cary Vietnam, 2013.

NGUYEN XUAN PHI was born in Vietnam in 1980

Obtained BE in Information Technology from HOCHIMINH university of technology in Vietnam, 2006.

He received Master in PTIT, 2012, major in Networking and Data Transmission.