

Integrated Subset Split for Balancing Network Utilization and Quality of Routing

S. V. Kasmir Raja, and P. Herbert Raj

Abstract—The overlay approach has been widely used by many service providers for Traffic Engineering (TE) in large Internet backbones. In the overlay approach, logical connections are set up between edge nodes to form a full mesh virtual network on top of the physical topology. IP routing is then run over the virtual network. Traffic engineering objectives are achieved through carefully routing logical connections over the physical links. Although the overlay approach has been implemented in many operational networks, it has a number of well-known scaling issues. This paper proposes a new approach to achieve traffic engineering without full-mesh overlaying with the help of integrated approach and equal subset split method. Traffic engineering needs to determine the optimal routing of traffic over the existing network infrastructure by efficiently allocating resource in order to optimize traffic performance on an IP network.

Even though constraint-based routing [1] of Multi-Protocol Label Switching (MPLS) is developed to address this need, since it is not widely tested or debugged, Internet Service Providers (ISPs) resort to TE methods under Open Shortest Path First (OSPF), which is the most commonly used intra-domain routing protocol. Determining OSPF link weights for optimal network performance is an NP-hard problem. As it is not possible to solve this problem, we present a subset split method to improve the efficiency and performance by minimizing the maximum link utilization in the network via a small number of link weight modifications. The results of this method are compared against results of MPLS architecture [9] and other heuristic methods.

Keywords—Constraint based routing, Link Utilization, Subset split method and Traffic Engineering.

I. INTRODUCTION

IP routing typically uses shortest-path computation with some simple metrics such as hop-count or delay. Although the simplicity of this approach allows IP routing to scale to very large networks, it does not make the best use of network resources [2]. In large Internet backbones, service providers typically have to explicitly manage the traffic flows in order to optimize the use of network resources. This process is often referred to as traffic engineering.

The goal of traffic engineering is to optimize the performance of operational networks [7]. Common objectives of traffic engineering include balancing traffic distribution across the network and avoiding congestion hot spots. In this

paper, we consider a new approach that accomplishes traffic engineering objectives to overcome the splitting problem without full mesh overlaying. We present a formal analysis of the equal subset split and propose a systematic method for deriving the link metrics that convert a set of optimal routes for traffic demands to shortest-path with respect to the link weights.

We provide different approaches in Section II. In Section III we study the equal subset split approach. We describe the LP formulations and evaluate this method in Section IV. In Section V, we apply this method to the integrated approach. Then we draw conclusions.

II. BACKGROUND

Traffic engineering [2] has drawn much attention in recent years. Two important components of traffic engineering are traffic estimation and routing. A good understanding of the interplay between these two inter-related components will make significant contribution to network management and performance. A routing specifies how to route the traffic between each origin-destination pair across a network.

A. Integrated Approach

The Overlay approach [2] has been widely used by many service providers for traffic engineering in large Internet backbones. With this approach service providers establish logical connections between the edge nodes of a backbone, and then Overlay these logical connections onto the physical topology. These logical connections between edge nodes essentially form a full-mesh virtual network atop of the physical topology. While the Overlay approach has been widely implemented in current Internet backbones, it suffers the so-called “N-Square” problem. Second while IP routing runs over such a fully meshed virtual network, each edge node has to establish routing peering with (N-1) other nodes.

Yuji Wang, et al proposed a new approach called Integrated approach [3] that accomplishes traffic engineering objectives without full mesh overlaying. Instead of overlaying IP routing over the logical virtual network, the new approach runs shortest-path IP routing natively over the physical topology. It is theoretically proved that for any given traffic demands it is possible to select a set of link weights such that the shortest paths based on the selected link weights produce the same traffic distribution as that of the overlay approach with the assumption that traffic between the same

Manuscript submitted on March, 31, 2007.

S. V. Kasmir Raja is with SRM University, Chennai, Tamil Nadu, India (phone : 91-0431-2490731, 91-94433-49073; e-mail : svkr@yahoo.com).

P. Herbert Raj is with Dept of CSE, Alagappa University, Karaikudi, Tamil Nadu, India (e-mail : herbi_raj@hotmail.com).

source – destination pair can be split across multiple equal cost shortest paths, if exists.

Before we present the theoretic results, let us first illustrate with a simple example how the integrated approach works. Fig. 1 shows a simple network topology, link capacities, and traffic demands. Each link has a capacity of 5 units and each demand needs bandwidth of 4 units. Although link capacities, link weights and traffic demands are unidirectional in IP networks, we assume they are bidirectional here for simplicity.

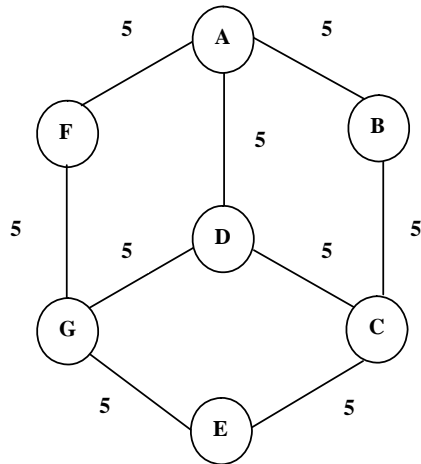
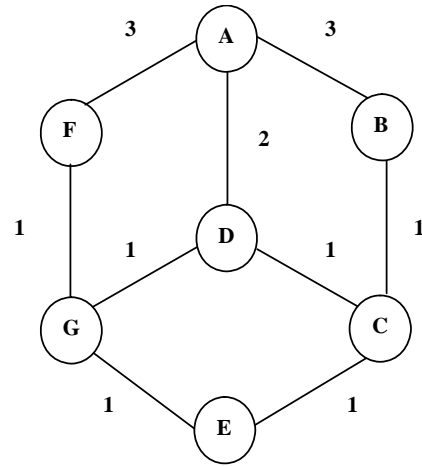


Fig. 1 Network Topology, Capacity and Traffic Demands

To meet the traffic engineering objectives, we need to place the demands over the links in a way that the traffic distribution is balanced and there is no congestion or hot spot in the network. The optimal routes can be calculated using a linear programming formulation [8].

Since the network here is rather small, this process of traffic engineering can be done manually. The optimal routes for achieving balanced traffic distribution are as follows. Demand A to B uses path AB, and demand A to F uses AF. Demand B to F has two paths. Half of the demand goes over BCDGF and the other half over BCEGF. Demand A to E also has two paths. Half traverses path ADCE and the other half traverses ADGE. This optimal routing result in a 4-unit load on every link – the traffic distribution is balanced and the link utilization is 80% uniformly for the entire network. To implement the optimal routes with the overlay approach we simply set up six logical connections: AB, AF, BCDGF, BCEGF, ADCE and ADGE, and run IP routing over them. BCDGF and BCEGF appear as equal-cost paths, so routing protocols such as OSPF will perform load sharing over them. We simply calculate and set the appropriate link weights on the links, and the shortest-path routing will calculate the paths by itself. Fig. 2 shows the link weights under which the shortest paths match the optimal routes exactly.



A->B : 4 A-B
A->F : 4 A-F
B->F : 2 B-C-D-G-F
B->F : 2 B-C-E-G-F
A->E : 2 A-D-G-E
A->E : 2 A-D-C-E

Fig. 2 Optimal link weights and optimal routes

This Integrated approach has a number of advantages. First, it retains the simplicity of IP routing and requires little changes to the basic Internet architecture. Once the weights are calculated and set, the shortest-path routing protocol such as OSPF [4] can calculate the paths in the normal way, and packets are forwarded along the shortest paths.

Second it eliminates the “N-Square” problem all together and reduces managing overheads in setting up logical connections.

B. OSPF Optimized Multipath

Another related work on achieving better traffic distribution without full-mesh overlaying is to use equal-cost load balancing in the OSPF routing protocol [5]. The effectiveness of this approach largely relies on how many equal cost shortest paths exists between each source and destination pair. In a related effort, OSPF link weights and equal traffic load sharing is combined to improve performance [6]. The analysis shows that the link weight optimization problem under equal load sharing is NP-hard.

In addition, a load search heuristic algorithm is proposed which achieves a performance quite close to the optimal routing only on a specific example. There are some issues in the equal cost splitting method. The solutions are flow specific – need destination specific solutions. Moreover it needs modification in IP routing.

III. OVERVIEW OF SUB-SET SPLIT

A. Subset split method

One approach to overcome the “splitting problem” is to approximate optimal link load. The current routing tables have thousands of routing prefixes. Instead of routing each prefix on all equal cost paths, we can selectively assign next hops to (each) prefix i.e., remove some equal cost next hops assigned to prefixes.

Let us first illustrate with a simple example how the subset split works. Fig. 3 (a) shows a simple network topology, link capacities and the prefixes of the hops in the network.

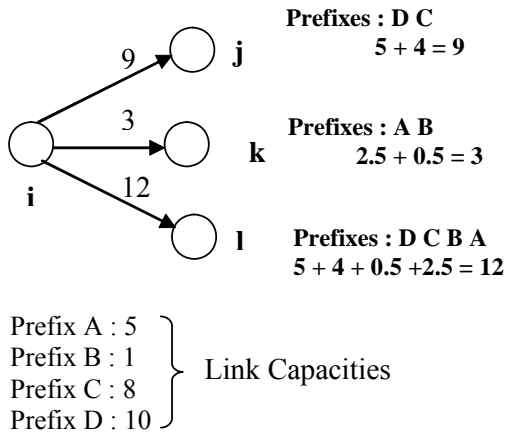


Fig. 3 (a) Subset split method – Simplified network

The prefixes of the hops are listed below.

Prefix A : Hops k,l
Prefix B : Hops k,l
Prefix C : Hops j,l
Prefix D : Hops j,l

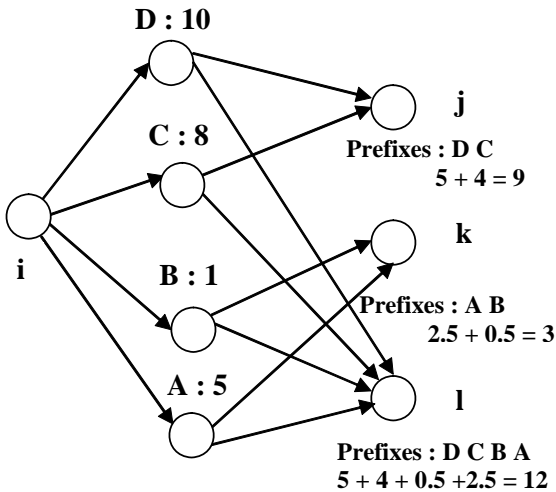


Fig. 3 (b) Subset split method – Complete flow from source “i” to destinations ‘j’, ‘k’ and ‘l’

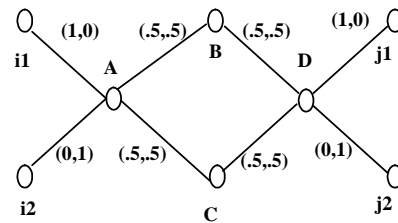
Fig. 3(b) shows the complete flow from the source node “i” to the three different destination nodes namely “j”, “k” and “l” of the network topology illustrated in Fig. 3 (a).

For example we can assign D and C as prefix to j. Since the total capacity of the link $i \rightarrow j$ is 9, we can assign 5 units of D and 4 units of C i.e., we can equally divide the prefix’s capacities D and C to achieve the capacity 9 and assign next hops to each prefix. We can selectively assign the next hops based on the link capacities.

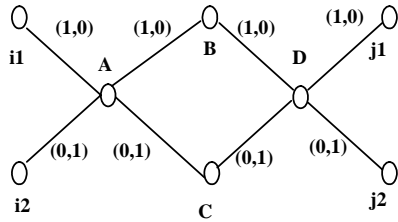
This subset split approach can be applied for selectively assigning next hops based on the link capacities of the prefixes for different demands of the network.

B. Routing and Performance Metrics

A routing $f_{ij}(l)$ specifies the fraction of traffic for the Origin-Destination (OD) pair $i \rightarrow j$ on link l . When the demand for the OD pair $i \rightarrow j$ is d_{ij} , the traffic on link l is $d_{ij}f_{ij}(l)$. Throughout the paper, we denote a routing as f , a link as l and the capacity of a link l as $C_{ij}(l)$.



(a)



(b)

Fig. 4 An Example

In Fig. 4, we present two example routings for illustration. The vector $(f_{i1j1}(l), f_{i2j2}(l))$ on each link l specifies the routing. For example, in Fig. 4(a), (1,0) on link $(i1,A)$ specifies that 100% of the traffic for OD pair $i1 \rightarrow j1$ travels link $(i1,A)$; while no traffic of $i2 \rightarrow j2$ on $(i1,A)$. The vector $(.5, .5)$ on link (A,B) specifies 50% of the traffic of $i1 \rightarrow j1$ (as well as $i2 \rightarrow j2$) travels link (A,B) . In Fig. 4(a), there are two paths for $i1 \rightarrow j1$, $i1ABDj1$ and $i1ACDj1$. There is only one path for $i1 \rightarrow j1$ in figure 4(b), $i1ABDj1$.

Similarly OD pair $i2 \rightarrow j2$ has two paths in Fig. 4(a), while there is only one path in Fig. 4(b).

A routing \mathbf{f} is defined as:

$$\forall i, \forall j \neq i : \sum_{e \in \text{OUT}(i)} f_{ij}(e) - \sum_{e \in \text{IN}(i)} f_{ij}(e) = 1;$$

$$\forall k, \forall i \neq k, \forall j \neq k, i : \sum_{e \in \text{OUT}(k)} f_{ij}(e) - \sum_{e \in \text{IN}(k)} f_{ij}(e) = 0;$$

$$\forall \text{ edge } e, \forall i, j \neq i : f_{ij}(e) \geq 0;$$

In the above, $\text{IN}(i)$ and $\text{OUT}(i)$ denote the sets of edges “into” and “out of” node i respectively.

For a given routing \mathbf{f} and a given traffic demand D [10], the maximum link utilization measures the goodness of the routing, i.e., the lower the maximum link utilization, the better the routing:

$$\max_i \frac{\sum_{i,j} d_{ij} f_{ij}(I)}{C_{ij}(I)}$$

IV. OPTIMIZING LINK UTILIZATION OF THE NETWORK

A. Basic Assumptions

In this section, we first discuss the basic assumptions. We then introduce the mathematical notation, and present a linear programming formulation. In the above example, we consider balanced traffic distribution as the overall objective for traffic engineering. For some other applications, the optimization objectives may be different, such as minimum congestion or maximum throughput. We do not restrict ourselves to any specific objectives. Our results are generic and can be applied to all of these objectives.

We model the IP network as a set of nodes connected by links with fixed bandwidth capacities. We assume that the point to point traffic demands between nodes are given, each with a fixed bandwidth requirement.

B. LP Formulation

We next introduce the concept of linear programming (LP) [11] formulation. We model the IP network as a set of nodes connected by links with varying bandwidth capacities.

Let digraph $G = (V, E)$ represent the IP network, where V is the set of nodes and E is the set of links. Please note that the links and their capacities are directional, i.e., link (i, j) is considered different from (j, i) , each with its own capacity. C_{ij} represents the capacity of the link $(i, j) \in E$. Let K be the set of origin destination flows.

For $k \in K$, d_k be the demand, s_k be the source and t_k be the destination node. X_{ij}^k be the fraction of flow k going over $(i, j) \in E$. Let α be the maximum link utilization. Our aim is to minimize the maximum link utilization.

Min α

Subject to.

$$\sum_{j: (i,j) \in E} X_{ij}^k - \sum_{j: (j,i) \in E} X_{ji}^k = 0, \quad i \neq s_k, t_k, k \in K \quad (1)$$

$$\sum_{j: (i,j) \in E} X_{ij}^k - \sum_{j: (j,i) \in E} X_{ji}^k = 1, \quad i \neq s_k, k \in K \quad (2)$$

$$\sum_{j: (i,j) \in E} X_{ij}^k - \sum_{j: (j,i) \in E} X_{ji}^k = -1, \quad i \neq t_k, k \in K \quad (3)$$

$$\sum_{k \in K} d_k X_{ij}^k \leq C_{ij} \alpha, \quad (i, j) \in E \quad (4)$$

$$X_{ij}^k \geq 0 \quad (5)$$

The objective function is to minimize the maximum link utilization. The first set of constraints (1), (2) and (3) are flow conservation constraints.

The constraint states (1) that the traffic flowing into a node has to equal traffic flowing out of the node for any node other than the source node and the destination node for each demand, (2) the net flow out of the source node is 1, which is the total required bandwidth after scaled by d_k and (3) the net flow out of the destination node is -1. Next constraint (4) is the link capacity utilization constraint. It says that the total amount of bandwidth consumed by all the demands routed on a link should not exceed the maximum utilization rate times the total capacity of the link. The last constraint (5) restricts the X_{ij}^k variables to be greater than or equal to 0.

We can solve the above LPF problem with the classic Simplex method [12]. The optimal solution of (LPF) gives a route or a set of routes (splitting) for each demand. In case that a demand has to be splitting, it also gives the proportions according to which the traffic between the source and the destination nodes should be distributed across the multiple routes.

Bertsekas and Gallager indicate in their textbook [13] that the optimization objectives of minimax link utilization and minimum average delay are essentially equivalent. So we introduce the linear programming formulation for minimum delay also.

B. Minimum Delay Routing

As we stated above the network is represented by a graph $G=(V, E)$. The traffic matrix is given by $r_s(d)$ i.e., the traffic entering s destined for d . Then,

$$r = \sum_{s,d \in V} r_s(d)$$

Let f_{ij}^{sd} be the expected traffic (bps) on link (i, j) for source / destination pair s, d and f_{ik} be the expected traffic (bps) on link (i, k) . Then,

$$f_{ik} = \sum_{s,d} f_{ij}^{sd} \quad (i, k) \in E$$

The delay of message from s to d is T_{sd} and T be the delay of random message. Then the LP formulation is,

$$D_T(\{f_{ik}\}) \equiv E[T] = r^{-1} \sum_{s,d \in V} r_s(d) E[T_{sd}]$$

Minimize $D_T(\{f_{ik}\})$

Subject to.

Flow constraints

The objective function is to minimize the delay of message from source to destination of a random message. The flow constraints are already stated in the LP formulation itself. The optimal solution gives a route or a set of routes that has only a minimum delay for the delivery of a random message from source to destination.

V. CONCLUSION

The new approach is proposed for achieving traffic engineering in the backbones. Instead of relying on the mapping of logical connections of physical links to manage traffic flows in the network, we run IP routing natively over the physical topology, and control the distribution of traffic flows through setting appropriate link weights for shortest path routing. For any set of optimal routes, shortest paths will be calculated with respect to a set of positive link weights. Instead of routing each prefix on all equal cost paths, selectively we can assign next hops to each prefix. This will minimize the maximum link utilization and improve the quality of routing to almost perfect.

As stated by Bertsekas and Gallager the essential optimization objectives, i.e., minimize the maximum link utilization and minimum average delay are achieved by this Integrated sub-set split approach. Besides these major objectives, there are other factors to consider, such as path dispersion and path variation [14]. Path dispersion is concerned with the number of paths. Path variation is concerned with how far the paths are from the shortest paths and variation of path lengths.

We studied the quality of the routing with no knowledge of traffic demands based on the path dispersion and the path variation. The Integrated equal sub-set split approach strikes a good balance between the conflicting objectives of minimizing the maximum link utilization and optimizing the quality of the routing with the help of the minimum delay LP formulation. Moreover the heuristics provide good performance.

ACKNOWLEDGMENT

We would like to acknowledge all the people who assisted us during our work at St. Joseph's College, Tiruchirapalli and **Ms. A. Padmapriya**, Lecturer, Department of Computer Science and Engineering, Alagappa University, Karaikudi, Tamil Nadu, India.

REFERENCES

- [1] A. Chakrabarti and G. Manimaran, Reliability constrained routing in QoS networks, IEEE/ACM Trans. on Networking, vol. 13, no. 3, pp. 662-675, June 2005.
- [2] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. "Requirements for traffic engineering over MPLS", Internet draft, IETF, April 1998, draft-awduche-mpls-traffic-eng-00.txt.
- [3] Yufei Wang, Zheng Wang and Leah Zhang "Internet Traffic Engineering without full mesh overlaying", Infocom, 2001.
- [4] J. Moy, "OSPF version 2", Internet rfc", IETF, May 1998, Technical report RFC 2328.
- [5] C. Villamizar, "OSPF optimized multipath", Internet draft, IETF, March 1998, draft-ietf-ospf-omp-00.txt.
- [6] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights", Proceedings of INFOCOM'2000, March 2000.
- [7] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic engineering over MPLS. IETF RFC 2702, September 1999.
- [8] Yufei Wang and Zheng Wang, "Explicit routing algorithms for internet traffic engineering", Proceedings of ICCCN'99, Sept 1999.
- [9] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture, IETF RFC 3031, Standards Track, January 2001.
- [10] A. Soule, A. Nucci, E. Leonardi, R. cruz and N. Taft, How to identify and estimate top largest OD flows in dynamic environment. In Proceedings of SIGMETRICS'04, pages 49-60. New York, June 2004.
- [11] CPLEX: Mathematical Problem Solver. <http://www.cplex.com>.
- [12] V. Chvatal, Linear Programming, W H Freeman and Company, 1983.
- [13] D.P Bertsekas and R.G.Gallager, Data Networks (2nd Edition), Printrice Hall, pp 434.
- [14] Yuxi Li, Janelle Harms and Robert Holte "A simple method for balancing network utilization and quality of routing", Proceedings of ICCCN'05. Sept. 2005.

S. V. Kasmir Raja received his Ph.D in Physics from the Department of Physics, Madras University, Chennai, Tamil Nadu, India. He retired as Professor and Head of the Department of Computer Science from St. Joseph's College, Tiruchirapalli. He is currently working as Dean of Research, SRM University, Chennai, Tamil Nadu, India.

P. Herbert Raj received his Master Degree in Computer Science from the St. Joseph's College, Tiruchirapalli, Tamil Nadu, India in 1996. He is currently pursuing Ph.D in the area of Quality of Service. His areas of interest include Traffic Engineering and Quality of Service. Currently, he is working as a Lecturer at Department of Computer Science and Engineering, Alagappa University, Karaikudi, Tamil Nadu, India.