Hybrid Routing for Scalable IP/MPLS Traffic Engineering

Huan Pham, Bill Lavery
School of Information Technology
James Cook University
Qld 4811, Australia
{huan, bill}@cs.jcu.edu.au

***Abstract*—This paper presents performance data for a new Hybrid Routing Method (HRM) for IP network traffic engineering. In this method, the major traffic streams between some source/destination pairs is routed via MPLS constraint-based routing, while the remaining traffic is routed via conventional IGP routing. When Internet traffic is heavy-tailed distributed, consisting of a few "elephant" and many "mice" flows, our traffic engineering results indicate that our HRM normally needs just a small number of tunnels to achieve a network performance that is comparable to that of fully meshed MPLS network. The method has been used to implement a HRM traffic engineering tool. The tool enables network operators to visualize and manage traffic to avoid congestion, as well as to decide where to place MPLS routers and tunnels.**

Keywords—Hybrid Routing, MPLS, IP, Traffic Engineering

I. INTRODUCTION

Traffic engineering for IP networks is the process of mapping traffic demand into the network topology, and realizing such mapping via routing protocols, so that a predefined performance objective is achieved. From the network point of view, the objective is an evenly loaded network with minimum congestion, i.e. there should not be unnecessarily over-utilized links while others are under-utilized.

Current IGP (Interior Gateway Protocol) routing protocols provide little capability for solving the above traffic engineering problem. These routing protocols (such as Intermediate System to Intermediate System - IS-IS, Open Shortest Path First - OSPF) are all destination-based Shortest Path First (SPF) protocols. At each router, routes are calculated in a distributed way, based on the link metric and the network topology, but not based on the network load status. The only way to change routes is by changing the metric associated with each link. Explicit routes and constraint based routing are therefore impossible. A number of researchers [1, 2] have reported optimizing the link metrics for the purpose of traffic engineering, but this approach only provides limited gain. In many cases, without constraint based routing and explicit routes capability, congestion caused by unbalanced load is unavoidable [3, 4].

This traffic engineering problem can be effectively addressed by the evolving Multi Protocol Label Switching (MPLS) technology [4, 5, 6, 7]. With MPLS, routes are calculated at source routers, called Ingress Routers, which take into account not only the network topology but also traffic oriented constraint (such as bandwidth, delay, hop count) and administrative constraints (i.e. some links or nodes are preferred for certain traffic demands). The network operator therefore has a greater control over how traffic is routed and traffic engineering can be more effective.

However, MPLS fully meshed networks have a scalability limitation. For a network of N routers, the number of Label Switch Paths (LSPs) that have to be set up and distributed (to routers along the LSPs) for all sources/destinations is in the order of N^2. This does not scale well for a large network. On the other hand, IGP has a much better scalability, because routes are calculated in a distributed fashion at each router, and the routing table size is in the order of N only. This suggests that a hybrid of IP and MPLS routing may provide a reasonable compromise, providing better traffic engineering than does plain IP routing, but better scalability than does fully meshed MPLS.

In the next section, we discuss the problems associated with existing hybrid routing schemes, and introduce our new Hybrid Routing Method. Section III presents the architecture of a HRM traffic engineering tool, and section IV explains the principle of the tunnel placement module, which is the core of our traffic engineering tool. Traffic engineering results for the AT&T Internet backbone are discussed in Section V. Finally Section VI summarizes the main contributions of the paper as well as our future research.

II. HYBRID ROUTING METHOD

In this new Hybrid Routing Method (HRM), some traffic is routed via MPLS constraint-based routing while the remaining traffic is routed via plain IP routing. This hybrid approach enables network operators to gradually upgrade their facilities to be MPLS enabled, rather than having to upgrade all at once; the later is the MPLS fully meshed model. The traffic engineering problem for this hybrid method reduces to determining where and how to set up LSPs, and then allowing the remaining traffic to be routed via plain IP routing.

Commercial tools such as WANDL IP/MPLS View [8] and OPNET Service Provider Guru [9] address similar problems. However, these tools are not accessible for public, and their algorithms for optimizing MPLS tunnels can not be evaluated.

The published proposal by Wang and Zhang [10, 11] is a hybrid approach originating from the finding that, it would be possible to achieve an optimal solution for traffic engineering purposes using an IGP routing protocol if it supported arbitrarily traffic splitting. The authors then proposed to solve the optimal routing problem. The outcome is the set of

0-7803-7802-4/03/$17.00 © 2003 IEEE

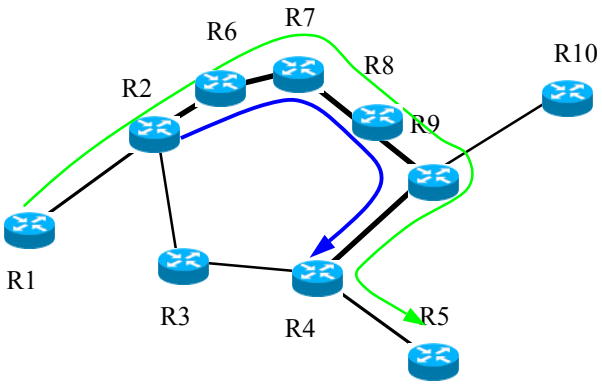

Figure 1: Local and End-to-End tunnels

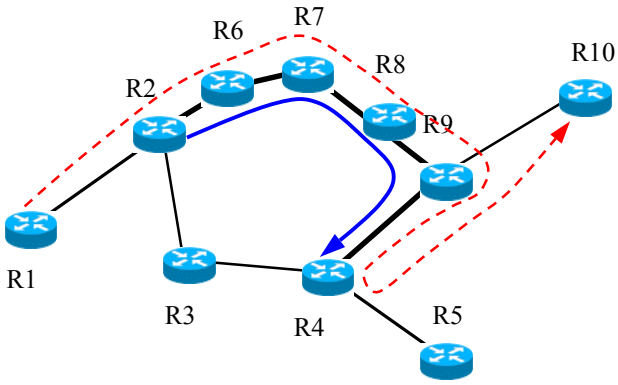

Figure 2: "Back flow" problem

weights and the traffic splitting ratio when there are multiple paths between any source/destination pair. Traffic is normally routed via shortest paths, and MPLS is only used for arbitrary splitting when it is needed.

However, this approach depends heavily on adjusting IGP metrics for traffic engineering, and so is problematic in that, when the traffic requirements change, the optimal set of weights and the ratio for traffic splitting change accordingly. Note that changing the whole set of weights in plain IP routing is generally undesirable as it often results in large convergence time, during which routing loops and packet losses may occur. In addition, this approach does not guarantee that the outcome set of weights are all integer and does require an arbitrary traffic splitting ratio among parallel paths. In practice, traffic splitting is usually done by mapping the paths with an integer number of "hash buckets", and the maximum number of buckets is often limited (16 for Cisco routers) [12, 13]. That means a truly arbitrary splitting ratio is not realizable in practice.

Therefore, our HRM has been designed so as to not rely on changing IGP metrics for traffic engineering. Rather, we actively use MPLS tunnels to avoid the potential congestion links that would otherwise be caused by plain IP routing. Depending on where one places the MPLS tunnels, the hybrid traffic engineering can be further classified as "local" or "end-to-end" tunnel approaches.

The local tunnel approach places MPLS tunnels, or Label Switch Paths (LSPs), locally around any overly congested links. In an example, given in Figure 1, traffic from R1 to R5 would normally overload the links R2-R3 and R3-R4 if only IGP routing was supported. A local tunnel is therefore set up between R2 and R4 to steer some of the traffic to go via a longer but wider path R2-R6-R7-R8-R9-R4.

The other option is to create an end-to-end tunnel between the major sources and destinations that contribute to the heavily loaded links, i.e. the tunnel R1-R2-R6-R7-R8-R9-R4-R5 between R1 and R5.

The two options both have advantages and disadvantages. The local tunnel approach has a quicker response to network failures, as it takes less time to inform the ingress router of intermediate node or link problems, and to switch packets to back-up tunnels. However, the local tunnels may result in a sub-optimal "back flow" problem as shown in Figure 2. Assume that there are two demands between R1-R5 and R1-R10, plain IP routing would normally cause links R2-R3 and R3-R4 to be overloaded. The local approach creates a tunnel to get around the most heavily loaded links, i.e. between R2 and R4. The traffic from R1 to R10 then has to take a round tour to R4 and back to R10 via R9.

The "end to end" tunnel approach overcomes this "back flow" problem because each tunnel is created to serve only one traffic demand causing no side effect on the IGP routing for other source/destinations. This gives more flexibility for the tunnel placement process. For this reason, we chose to implement this end-to-end approach for our HRM, even though this may imply additional MPLS enabled routers.

One may ague that in the given example, local approach can still create two separate tunnels between R2-R4 and R2-R9 to overcome the “back flow” problems. This is a valid solution, that can be also viewed as an “end to end” solution if we only apply traffic engineering for a subset {R2.. R9} of the network. In other words, our “end to end” notion can be relaxed, which may apply to a subset of routers and aggregated traffic between that subset.

For large networks, where response time to network failure along long tunnels may be of concern, it may be better that this "relaxed end-to-end" HRM approach is used for its sub-areas, such as the backbone network of the Internet where the traffic engineering is often most needed.

III. Architecture for HRM Traffic Engineering Tool

For the best performance of the HRM implementation, it is necessary to have a traffic engineering tool that is responsible for modeling OSPF traffic routing as well as the tunnel placement optimization. The architecture of our proposal is described in Figure 3.

From the bottom of the diagram upward, the traffic demand module is regularly updated via traffic projection and/or measurement. Given the topology database and traffic demands as inputs, a plain IP traffic load on each link is calculated. Based on the plain IP traffic load assignment, potential congested links are then identified. This, together with the topology and traffic database, will then serve as the inputs for the MPLS tunnel optimization module.

In the next part, we will discuss the tunnel optimization module, as this is the most important part of the traffic engineering tool.

IV. MPLS TUNNEL OPTIMIZATION

In order to decide where and how to place MPLS tunnels, an optimization objective for overall network performance needs to be defined. Without losing generality our tool uses the objective function of [14], which is based on the average end-to-end packet delay (or total end-to-end packet delay), taking into account the self-similar characteristics of the Internet traffic. This objective function also allows for the link utilization to be greater than 100%, at a cost of a very high penalty. The main reason is to discourage any traffic engineering solution that ends up with overloaded links. Of course, different objective functions, which might better represent network performance, can also be utilized for our HRM approach.

For any given network, the tunnel placement problem then reduces to setting up a limited number of tunnels between some source/destination routers such that the network performance objective function improves the most. For our objective function, this equates [14] to optimizing the total packet delay, Φ, summed over all links. That is, the aim is to minimise:

$$\Phi = \sum_{a \in A} \Phi_a \tag{1}$$

$$\Phi_a = \begin{cases} \rho_a + \dfrac{\rho_a^3}{(1-\rho_a)^3}, \text{for } \rho_a < 0.9 \\ 24301\rho_a - 21141, \text{for } \rho_a \geq 0.9 \end{cases} \tag{2}$$

where A is the set of arcs (or unidirectional links), $\rho_a = f_a/C_a$ is the link utilization, C_a and f_a are the link capacity and traffic load on the link a.

The tunnel placement problem without arbitrary traffic splitting is known to be a NP-hard problem [15]. There is no algorithm that can guarantee the solution to such problems within a reasonable time limit.

In this paper, we propose a greedy algorithm, which attempts to identify the best traffic demand candidates one at a time, to be routed via a LSP tunnel, and then implements the new LSP tunnel that improves the objective function the most. It is observed that, the most influential terms in the objective function (1) are the links with highest utilizations. Therefore, the best traffic demand candidates for constraint-based routing should be found among those traffic flows that contribute to the load on the heaviest utilized links.

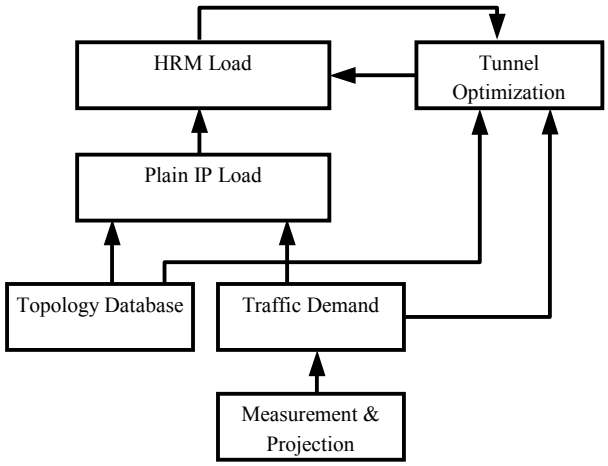

Figure 3: Traffic Engineering Tool Architecture

In addition, previous research [16, 17] has shown that the Internet traffic consists of a few "elephant" flows and many "mice" flows. Moving those "mice" flows from a heavily loaded links by MPLS tunnels would not significantly improve the overall network performance, while costing tunnel set-up and maintenance. Therefore, our algorithm only searches among the largest flows contributing to heavily loaded links, as the candidates for potential tunnel placement.

The greedy algorithm is described as follows. We first calculate traffic flows and the resulting link loads routed by the SPF routing protocol, using the Dijkstra algorithm. We now have available the SPF routed flows on each link, as well as the sources and destinations of the flows $f(s,t,l_a)$.

Then, the links are considered in the order of decreasing link utilization. For each link candidate, flows from different source/destination routers are sorted according to their sizes. We again consider only a limited number of largest size flow candidates, as they are the dominant contributors to the link congestion; typically we use only the largest ten flows per link.

For each candidate flow, the effectiveness of constraint-based routing of that flow is then evaluated. The traffic demand from the corresponding source/destination pair is removed from the set of SPF traffic loads on all links, which will then be routed via a new LSP or parallel LSPs. The process of routing a traffic demand via single path or parallel paths constraint-based routing is described in the sections IV.*A*, and IV.*B* below.

If the best constraint-based routing solution for the 10 largest flows on the link results in improvement then it is chosen as the next actual MPLS tunnel implementation, and we repeat the process for the new heaviest link. If no improvement results, no tunnel is implemented and the search is done for the largest size flows on the next heaviest link. The whole process is repeated until some maximum number of LSPs, L, is reached, or no further improvement is made.

The algorithm is formally described as follows:

1. Model HRM traffic routing, initially with no tunnels, to determine plain IP flows on the links $f(s,t,l_a)$.

2. Sort the links according to the link utilization, in decreasing order. Select the highest utilization link as the candidate link.
3. Sort the plain IP traffic flows on the candidate link according to the flow size in decreasing order.
4. For each of the ten largest plain IP flows, try constraint-based routing the candidate traffic demand, using single LSP or multiple LSPs (as described in the sections IV.*A*, and IV.*B*).
5. If at least one of the solutions in step 4 results in improvement then the solution with the best objective function improvement per number of required LSPs, *N*, i.e. $\frac{\Phi_{old} - \Phi_{new}}{L}$, is chosen as the next MPLS tunnel(s) to be implemented. Update HRM traffic. Next candidate link is the new highest utilization link.

   Else (i.e. all solutions in 4 result in no improvement) the next candidate link is the next highest utilization link.
6. If the maximum allowed number of LSPs has been reached, or the next candidate link is the lowest utilization link (i.e. no improvement can be made), then stop; otherwise go back to step 3.

Although this greedy algorithm does not guarantee an optimal set of LSPs (subject to a limited number of LSPs and non arbitrary splitting ratios), the effectiveness of this algorithm can be evaluated. Here effectiveness means the difference between the performance objective function found by our algorithm, and that found by the "optimal" solution; the optimal solution allows for unlimited LSPs and arbitrary splitting and is solved using Flow Deviation (FD) algorithm described in [18].

A. Single LSP constraint-based routing

In this case, only a single LSP is permitted to route a candidate traffic demand *d*. We first remove this demand from set of plain IP routing demands. Then our objective is to find an LSP for this demand, to minimise the overall delay increase on every link on the path. Therefore the new LSP is actually the shortest path from the source to the destination; given the link "weight" of each link is the increase in delay on that link after loading the new traffic demand *d*:

$$w_a = \frac{\Phi_a(C_a, (f_a + d)) - \Phi_a(C_a, f_a)}{d} \qquad (3)$$

This problem is solved using the Dijkstra algorithm.

B. Parallel LSPs constraint-based routing

If truly arbitrarily splitting was supported, the optimal parallel LSPs placement for one traffic demand candidate would be just a special case of optimal routing, where there is only one traffic demand between the single source and destination under the consideration. The traffic demands from other sources/destinations can be considered as fixed residual loads. Therefore this problem of finding optimal LSPs for a single traffic demand could be solved optimally, again using the FD algorithm.

However, in practice traffic splitting among multiple LSPs can not be arbitrary. The splitting ratio is proportional to the number of hash buckets associated with each LSP in the set of parallel LSPs. This can be thought of as equivalent to a number of evenly load balancing tunnels, some of which follow the same route and are merged to become a bigger pipe. This problem can be solved by constraint based routing a number of equal size LSPs (the number of them is less than or equal to 16 for Cisco routers); this is similar to the single LSP scenario presented before, plus merging of the LSPs if they follow the same route.

Taking into account the fact that different parallel path constraint-based routing solutions may use different number of LSPs we use the average delay improvement rate per LSP as the evaluation criterion. This means the solution that maximizes $\frac{\Phi_{old} - \Phi_{new}}{L}$ will be chosen for tunnel placement, where Φ_{old} and Φ_{new} are the total delay before and after constraint-based routing this demand candidate, and *L* is the number of parallel LSPs used for constraint based routing this demand.

C. Application of End-to-End approach to a subset of routers

As discussed earlier, "end-to-end" tunnels respond less quickly to network failure compared to the "local" approach. Therefore, for a large network, we may want to relax the "end to end" definition to only a small subset of routers, for example backbone or core routers. These routers are identified, together with the accumulated traffic requirement between each source/destination router pair, including transit traffic from other "exterior" routers, entering at the source router and departing at the destination router. This is achieved via measurement, or by extracting data from the global traffic demand database. Solving the tunnel placement for the subset of routers is then one case of the end to end tunnel placement problem, and implementation is straight.

V. PERFORMANCE

We have implemented the proposed HRM traffic engineering tool. To evaluate its effectiveness, the traffic engineering tool is applied to the AT&T Internet backbone and all routers are assumed to be MPLS capable. The topology database is obtained from the Rocket Fuel program [19, 20]. For simplicity, we only consider backbone routers. In addition, routers belonging to the same cities are merged as one. The simplified topology consists of 107 routers and 140 links. The real link capacities and traffic matrix are not available. Without losing generality, we assign each link a capacity of 1000 Kbps. The traffic demand between any source/destination pair is generated randomly, as follows:

$$D_{i,j} = a \times e^b \qquad (4)$$

where *a* is a random number between [0,1] and b is a random number between [0, *ln(MaxDemand)*]. We then adjust the *MaxDemand* parameter to get about 150% maximum

utilization to simulate an overloaded network. For this traffic model, demand between any source/destination pair is random between [0, *MaxDemand*], and mostly small, but can be very high for a small number of source/destination pairs. This is to reflect traffic demand characteristics observed in practice, ie the Internet traffic from a source router mostly heads to few destination routers, while demands for the other destinations are very low.

The performance of our HRM is shown in the Figures 5 and 6. Figure 5 shows the network objective function versus the number of LSPs established. The horizontal line is the theoretical optimal average packet delay, calculated using the FD algorithm; the other line is our actual average packet delay objective function. Initially, with zero LSP, ie plain IP routing, the average packet delay is 2.2 sec. which is about 100 times larger than the 22.6ms of the optimally routed networks.

The figure indicates that, about 60 LSPs are needed to bring average packet delay down to about 23 ms, which is within 2% to the optimal value of 22.6ms.

Another measure of network performance would be maximum utilization across all links. Although not specifically addressed as the objective function, our algorithm also effectively reduces the maximum utilization, since the search for new tunnel implementation always first tries the heaviest utilized link. Figure 6 presents the maximum link utilization as a function of the number of LSPs. The maximum link utilization is reduced from 167.5% for the network using plain IP routing to 84% for the network using 60 LSPs; note that this utilization is the same as that obtained with optimal routing.

These results indicate that only 60 LSPs is needed for the network of 107 routers and 140 links to get to a performance that is very close to the optimal solution. In contrast, for a fully meshed MPLS routing, the number of LSPs required is of the order of 107^2 or about 10 000. An earlier paper [21] reported comparable results for arbitrary networks that are not as asymetric as the AT&T Internet backbone.

VI. CONCLUSION

In this paper, we have presented performance data for a new Hybrid Routing Method, HRM, for traffic engineering using both IGP protocols (such as OSPF or IS-IS) and MPLS routing. The HRM routes a limited number, L, of traffic flows using MLPS, and routes the remainder using the conventional IGP routing protocol. Simulations suggest that keeping L small (~60 for a network of 107 routers) yields significant improvement in network performance, and that network performance using HRM can approach that of a fully meshed MPLS network. Importantly, our approach is scalable to large networks, because it requires much less LSPs (in the order of O(N) roughly) compared to $O(N^2)$ for fully meshed MPLS networks.

Our method identifies the traffic flows that most impact on network performance, and determines for those flows the LSPs that will most improve network performance. This approach allows for progressive implementation of MPLS into an existing plain IP network. The network operator can

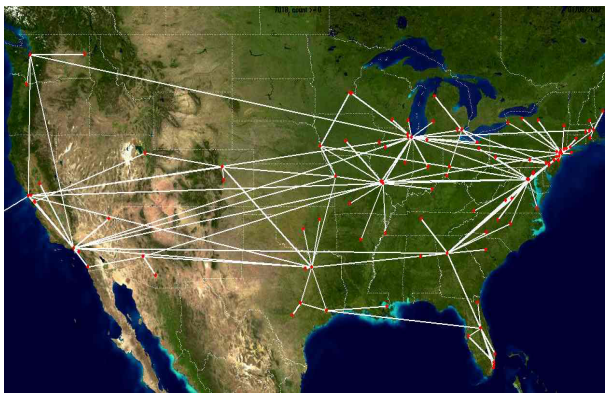

Figure 4: Network Diagram

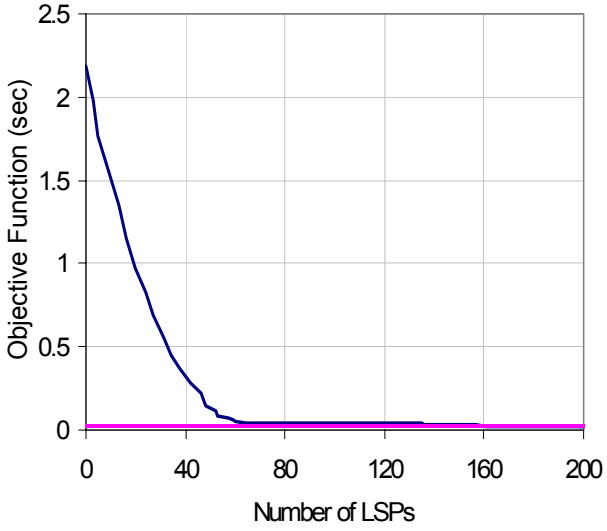

Figure 5: Objective Function vs Number of LSPs

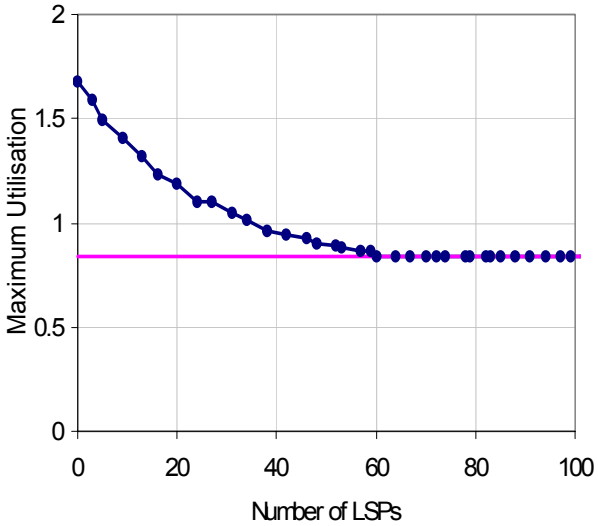

Figure 6: Maximum Utilisation vs Number of LSPs

identify the routers that might be first upgraded to MPLS operations as those which will have most impact on network performance.

Our results also indicate that it may often not be necessary to ever upgrade all routers in a network to MPLS capability. In highly asymmetric networks, a small number of MPLS routed traffic flows implemented with a small number of "core" routers can produce network performance approaching a fully MPLS enabled network.

Note that the previous published approach by Wang and Zhang [10, 11] relies heavily on adjusting the link weight metric and on MPLS arbitrary traffic splitting ratio assumption, both of which are not realistic. Our method, on the other hand, is not based on changing IGP metric, and does not assume arbitrary traffic splitting ratio, and therefore is more practical.

For further work, we will investigate a range of representative network topologies and traffic models, as well as other algorithms for optimizing the tunnel placement, to minimize the number of MPLS routers and to allow for the placement of backup tunnels in case of network failures. The tunnel placement module will also consider the re-optimization of the existing tunnels when the traffic demands vary dramatically from the existing values.

ACKNOWLEDGMENT

We would like to thank A/Prof. Greg Allen and Dr. Bruce Litow for their helpful discussions and comments. We also would like to thank the reviewers for their insightful comments and suggestions. This work has been supported by James Cook University through an International Post-Graduate Research Scholarship.

REFERENCES

[1] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights", in Proc. IEEE INFOCOM 2000, March 2000.

[2] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world", IEEE Journal on Selected Areas in Communications, Spring, 2002.

[3] B. Davie and Y. Rekhter, "MPLS technology and applications", Morgan Kaufmann Publishers, 2000.

[4] D. Awduche et al, "Requirements for traffic engineering over MPLS", IETF RFC2702.

[5] E. Rosen et al, " Multiprotocol label switching architecture", IETF RFC3031, http://ietf.org/rfc/rfc3031.txt, 2001.

[6] X. Xiao, A. Hannan, B. Bailey, and L. M. Ni, "Traffic engineering with MPLS in the Internet", IEEE Network magazine, March/April 2000.

[7] G. Swallow, "MPLS advantages for traffic engineering", IEEE Communication magazine, December 1999.

[8] WANDL IP/MPLS View software, http://www.wandl.com

[9] Opnet Service Provider Guru software, http:// www.opnet.com

[10] Y. Wang and L. Zhang, "A scalable and hybrid IP network traffic engineering approach", IETF Internet Draft, http://www.ietf.org / proceedings/ 02mar/ I-D/ draft-wang-te-hybrid-approach-00.txt, June 2001.

[11] Y. Wang, Z Wang and L. Zhang, "Internet traffic engineering without full mesh overlaying", Proceeding of INFOCOM, Alaska, April, 2001.

[12] Cisco, "MPLS traffic engineering and enhancements" http://www.cisco.com/ univercd/cc/ td/doc/product/ software/ios121/ 121newft/ 121t/ 121t3/traffeng.htm

[13] Cisco, "Troubleshooting load balancing over parallel links using Cisco express forwarding (CEF)" http://www.cisco.com/ warp/ public/ 105/ loadbal_cef.html

[14] H. Pham, B. Lavery, "On performance objective functions for optimising routed networks for best QoS", Proceedings of QoSIP, Italy, February, 2003.

[15] Y. Wang and Z. Wang, "Explicit routing algorithms for Internet traffic engineering", Proceedings of ICCCN'99, Boston, September 1999.

[16] W. Fang, L. Peterson, "Inter-AS traffic partterns and their implecation", proceedings of IEEE Globecom, Brazil, December 1999.

[17] K. Papagiannakietall, "On the feasibility of identifying elephants in Internet backbone traffic", Technical report, Sprint labs, 2001, http://ipmon.sprintlabs.com/pubs-trs/trs/TR01_ATL_110918.pdf

[18] A. Kershenbaum, "Telecommunications network design algorithms", McGraw-Hill, 1993

[19] N. Spring, R Mahajan, D. Wetherall, "Measuring ISP Topologies with Rocketfuel", proceedings of Sigcomm, Pittsburgh, August, 2002.

[20] Rocketfuel website, http://www.cs.washington.edu/research/networking/ rocketfel/

[21] H. Pham, B. Lavery "A New Scalable, Hybrid Approach for IP Traffic Engineering without Full Mesh Overlaying", proceedings of ICT2003, Tahiti, February, 2003.