A Minimum Interference Hybrid Algorithm for MPLS Networks

Kavitha Banglore

A project submitted to the

Department of Computer Science

in partial fulfillment of the

requirements for the degree of

Master of Science

Major Professor
Dr. Xin Yuan

Committee Members
Dr. Dave Whalley Dr.Lois Hawkes

Abstract

This paper presents an online hybrid algorithm for dynamic routing of bandwidth guaranteed connections in Multi-Protocol Label Switching (MPLS) networks. Existing on-line algorithms for MPLS networks optimize one of the three objectives: minimizing the hop count, minimizing the maximum link load or minimizing the interference of the established connections. In this process, performance of the network with respect to the other objective is compromised. In this paper we study hybrid methods to optimize all three metrics at the same time. We develop a path selection heuristic based on a combined metric. The idea is to avoid loading links which are critical and which will cause the maximum link utilization to increase further. The heuristic allows us to give different weights to the three objectives. This paper also surveys the existing algorithms and compares their performance with the hybrid algorithm. We conclude that the hybrid algorithm performs well with respect to all three metrics as compared to algorithms which optimize only one metric.

1 Introduction

One of the major functions of Multi-Protocol Label Switching [6] networks is Traffic Engineering(TE). TE is concerned with performance optimization of operational networks. In general, it encompasses the application of technology and scientific principles to the measurement, modeling, characterization, and control of Internet traffic, and the application of such knowledge and techniques to achieve specific performance objectives. A major goal of Internet Traffic Engineering is to facilitate efficient and reliable network operations while simultaneously optimizing network resource utilization and traffic performance. TE has become an indispensable function in many large Autonomous Systems because of the high cost of network assets and the commercial and competitive nature of the Internet[2]. Traffic engineering maps traffic flows to the physical network topology and adaptively reconfigures the mapping to changing network conditions. It provides the ability to move traffic flows away from the shortest path calculated by the Interior Gateway Protocol (IGP)[4] and onto a less congested path. Traffic engineering allows an Internet Service Provider to fully exploit its network infrastructure.

In MPLS networks, TE can be static or dynamic. In static TE, the flows are known a priori. Off-line algorithms such as Integer Linear Programming (ILP) can be used to find the routes and set up the Label Switched Paths (LSPs) for the flows accordingly. In dynamic TE, flow requests arrive dynamically and online routing algorithms need to be used to find a good path for satisfying the request. Static algorithms work well in environments where the network behavior is predictable. However, in networks with constantly changing flows, static routing algorithms are inadequate. Hence most of the dominant algorithms today are dynamic routing algorithms which adjust to changing network situations by analyzing route update messages. These algorithms compute the route based on the current network circumstances.

Existing algorithms for dynamic TE include the Minhop algorithm[4], the Widest Shortest Path algorithm[5], the Shortest Widest Path algorithm[5], the Minimum Interference Routing Algorithm (MIRA)[1] and the Hybrid algorithm[3]. The Minhop algorithm tries to optimize on the path length by finding the path with minimum number of hops in the feasible network. This algorithm does not take into consideration factors like utilization and interference. The Widest Shortest Path algorithm finds the minhop path with the maximum residual bandwidth. This algorithm too does not take into consideration factors other than bandwidth. Among algorithms that try to minimize the link load, the Shortest Widest Path algorithm tries to avoid overloading bottleneck links by maximizing the residual capacity across the network. But this might result in taking unnecessarily long paths to avoid bottlenecks. The Hybrid algorithm tries to balance the two objectives of reducing the hop count and avoiding bottleneck links by assigning appropriate weights for them. MIRA tries to optimize on the interference factor by routing traffic through paths which do not interfere too much with the future LSP demands. By doing so, it accommodates more connections. This algorithm does not consider link utilization and may lead to congestion.

In this paper, we propose a new dynamic algorithm that tries to balance the three factors, hop count, link utilization and interference. Our scheme uses a metric which combines the cost of a link in terms of the utilization, with its contribution to the maximum link utilization and the interference caused by using that link. The algorithm then selects the routes as least cost paths using this metric. We will call this algorithm the Minimum Interference Hybrid (MIH) algorithm.

The new algorithm was compared with other algorithms on a test network. As it tries to optimize three metrics, its performance was found to be good with respect to all three metrics as compared to other algorithms which performed well for the metric which they optimize and performed poorly for all other metrics. For example, when comparing the maximum link utilizations in the network for the different algorithms, the performance of our algorithm lies in between the Minhop algorithm that optimizes only the hop count metric and the Shortest Widest path algorithm that optimizes the link utilization metric. Hence this algorithm achieves a balance between the hop count, link utilization and interference factors.

This paper is organized as follows. Section 2 describes the current state of the art. Section 3 gives an overview of the MIH algorithm, describes the link metric and details the algorithm. Section 4 presents the results of simulation of the algorithm and its comparison with other algorithms. Section 5 concludes the paper.

2 Related Work

This section surveys the existing routing algorithms for traffic engineering in MPLS networks. As seen before, some of these algorithms focus on minimizing the hop count, some on minimizing the maximum link load and some on minimizing request rejections. All of these problems can be solved using Linear Programming(LP) solutions. For example, the problem of minimizing the maximum link utilization can be formulated as an LP as follows.

Let (V, E) be the network, where V is the set of nodes and E is the set of links. Let C_{ij} be the capacity of link $(i, j) \in E$. Let K be the set of traffic demands. Let S_k, t_k and d_k be the source node, destination node and the amount of traffic to be routed respectively, where $k \in K$. For each $(i, j) \in E$ and $k \in K$, let X_{ij}^k represent the fraction of demand k routed through (i, j). A demand maybe split over multiple paths with each path satisfying some fraction of the demand. Let α represent the maximum utilization in the network.

$$min \quad \alpha$$

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

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

$$\sum d_k X_{ij}^k \leq C_{ij}\alpha, \quad (i, j) \in E \tag{3}$$

$$0 \leq x_{ij}^k \leq 1, \quad \alpha \geq 0 \tag{4}$$

The objective function says that the variable to be minimized is the maximum of link utilization. Equations (1) and (2) are flow conservation constraints. Equation (1) says that traffic flowing into a node should be equal to traffic flowing out of a node except for the source and destination nodes. Equation (2) says that net flow out of the source node is 1 which is the total bandwidth scaled by d_k. Equation (3) states that for any link in the network, the total load should not exceed maximum utilization times the capacity of that link. Equation (4) states the non-negativity constraints.

The solution to this LP produces the optimal routes if the traffic demand can be split between multiple routes. But this is undesirable from the operational perspective. The LP can be changed to indicate the constraint that the load should not be split amongst multiple paths. To do this, equation (4) should be changed to state that x_{ij}^k should be either 1 or 0. In this case the problem becomes an Integer programming problem (ILP). This problem is NP hard, so no efficient algorithms exist which can produce exact solutions. But heuristic solutions exist for the problem, that provide sufficiently good solutions. As with the utilization problem, heuristics exist for other problems too. The following sections examine how these problems have been solved using such heuristics.

Minimizing Path Length

The simplest routing algorithm is the Minhop algorithm which finds paths with the minimum number of hops from the source to the destination. Though this algorithm is capable of finding the shortest path that can support the requested bandwidth, it has some drawbacks. Shortest paths of multiple traffic streams may converge causing overloading of specific paths whereas slightly longer paths might go underutilized. It may block future requests.

The Widest Shortest Path algorithm uses the hop count as the metric and chooses the path with the minimum hop count that can satisfy the request. If there are more than one such paths, the algorithm chooses the one with the maximum bandwidth. This algorithm optimizes the hop count and considers the bandwidth, but ignores the interference factor.

Minimizing Load

The Shortest Widest Path algorithm uses bandwidth as the metric and chooses the path with the maximum bottleneck bandwidth. The bottleneck bandwidth of a path is the minimum residual bandwidth among all the links in the path. If more than one path have the same maximum bottleneck bandwidth, it chooses the one with the minimum number of hops. The disadvantage of this algorithm is that it might choose lengthy paths in favor of less important shorter paths. So this algorithm optimizes only the link load metric and ignores other factors.

Minimizing Load and Path Length

The Hybrid algorithm tries to balance the two objectives of avoiding bottleneck links and choosing the shortest path. It uses a metric which combines the cost of a path in terms of link utilization with its contribution to the maximum of link utilization[3]. The algorithm computes the shortest path based on this cost metric. In the metric the the link utilization after the new demand is added is combined with the increase in the maximum of link utilization due to selecting this link.

For each link, let f_{ij} be the current load and C_{ij} be its capacity. Let α be the maximum link utilization in the network. Let d_k be the amount of traffic to be routed. Then the metric is defined as,

$\alpha_{ij} = \frac{f_{ij}+d_k}{C_{ij}} + T \times max[0, \frac{f_{ij}+d_k}{C_{ij}} - \alpha]$

Here T is a tunable parameter. It allows us to give different weights to the two factors. This combination produces an optimal route in terms of minimizing utilization at the same time keeping track of the path length. This algorithm does not take into account the interference factor.

Minimizing Interference

The goal of the Minimum Interference Routing Algorithm (MIRA) is to accommodate as many requests as possible. The idea is to use those paths which do not interfere too much with future LSP demands between other source destination pairs[1]. This algorithm assumes that there is some knowledge about the potential ingress-egress pairs. Knowledge about potential ingress-egress pairs helps in routing new traffic along paths which are not critical to future requests, thus reducing the number of request rejections. The following is the terminology used to explain this algorithm.

- *Interference:* Interference of a path can be defined as the decrease in the value of max-flow of an ingress-egress pair due to routing that LSP between some other ingress-egress pair.
- *Minimum Interference Path:* Minimum Interference path is the route between an ingress-egress pair that maximizes the minimum max-flow between all other ingress-egress pairs.
- *Critical Links:* They are links which when used in a new route between an ingress-egress pair, decrease the max-flow between one or more other ingress-egress pairs.

The idea behind the algorithm is to choose a path for a new LSP such that it involves the least possible number of critical links. It calculates the shortest path by taking criticality as the link cost. The drawback of this algorithm is that in trying to accommodate the maximum number of requests, it might end up choosing long and highly loaded paths in favor of more critical shorter paths adding to the overall load of the network. It ignores the hop count and load of the chosen path.

3 A Minimum Interference Hybrid Algorithm

All the previous algorithms concentrate on optimizing one performance metric in favor of another. The Hybrid algorithm which seemed to be balancing link utilizations with path length, does not take into consideration specific ingress-egress pairs and the effect of the path selection on them. The Minimum Interference algorithm does the opposite by choosing the path which least interferes with other ingress-egress pairs, but it ignores the utilization factor. Our evaluation shows that these algorithms perform poorly with respect to factors which they do not optimize. The need therefore is for a scheme which will balance all these factors. This section presents the new scheme that tries to achieve a balance between the three different factors of path length, maximum utilization and interference.

ILP Formulation

The problem of achieving a balance between the three factors can be formulated as an ILP. Let $G(N, L, B)$ be the given network, with N being the set of nodes with cardinality n, L being the set of links with cardinality m and B being their set of bandwidths of the m links. Let P be the set of distinguished ingress-egress node pairs. Let p denote the cardinality of this set. Let M represent the node-arc incidence matrix. In this matrix each column has exactly two non zero entries. For example in the column corresponding to arc (i, j) the entry for node i is 1 and that for node j is -1. All other entries are 0. Let x^{sd} be a m vector corresponding to a pair $(s, d) \in P$. Let θ_{sd} represent a scalar that is maximum flow that can be sent between s and d. Let R be an m-vector of residual capacities of arcs. R is initialized to B. Let e^{sd} represent an n vector with a $+1$ in position d and -1 in position s. The current demand is assumed to be between nodes a and b for D units of bandwidth. The problem can be formulated as below.

$$max \quad -T_1\mu + T_2 \sum_{(s,d)\in P\setminus(a,b)} \alpha_{sd}\theta_{sd} - T_3 \sum_{i=1}^{m} x_{ab}[i]$$

$$Mx^{sd} = \theta_{sd}\, e^{sd}\ \forall(s, d) \in P \setminus (a, b) \tag{5}$$

$$Mx^{ab} = -D\, e^{ab} \tag{6}$$

$$x^{sd} + x^{ab} \leq R\mu\ \forall(s, d) \in P \setminus (a, b) \tag{7}$$

$$x^{sd} \geq 0\ \forall(s, d) \in P \tag{8}$$

$$x^{ab} \in \{0, d\}^m \tag{9}$$

Here μ is the maximum link utilization in the network. The first term in the objective function works towards minimizing the maximum link utilization. The second term maximizes the minimum max-flow between all the ingress-egress pairs in P and the third term minimizes the hop count of the path. Equation(5) is the maximum flow problem for each of the ingress-egress pairs in P except (a, b). Equation(6) states that D units of bandwidth need to be routed between a and b. Equation(7) of this formulation makes sure that, link utilization not only does not exceed the capacity of the link but also does not exceed the maximum link utilization μ for any link in the network. Equations (8) and (9) ensure the non negativity and single path restrictions. T_1,T_2 and T_3 are constants which allow us to manipulate the weight given to the three different objectives.

The problem presented above is NP-hard. But as with minimizing the maximum utilization problem, we can find efficient heuristic solutions to this problem. This can be done by changing this problem to a shortest path problem with appropriate weights for the links in the network.

Shortest Path Solution

In trying to balance the interference factor and the load factor, we need a link metric which will combine together these factors by assigning them appropriate weights. The interference of a path is measured in terms of the criticality of links in the path. Critical links in a network can be found by solving the Max-Flow Min-Cut problem for each ingress-egress pair in the network. This problem tries to find those links that are absolutely essential for routing a flow from the source to the destination. So the set of critical links between a source-destination pair is the set of min-cuts found for the pair. The criticality of a link is then computed as the number of ingress-egress pairs for which it is critical. Hence in the new link metric, the criticality of a link can be used for introducing the interference factor.

The idea behind including a utilization factor is to minimize the maximum of link utilizations. The load factor can involve two components. One is the load on a particular link and the second is the contribution of the link to the increase in the maximum utilization. The second factor takes care of the main objective, that is to minimize the maximum of link utilizations. The first factor controls the path length. By taking this factor into consideration, a shortest path kind of algorithm will choose paths whose additive link utilization is minimum.

Combining these ideas, the link cost metric can be fashioned in the following manner:

For each link, let f_{ij} be its current load, C_{ij} be its total capacity and α be the current maximum link utilization. Let g_{ij} be its criticality and β be the maximum criticality in the network. We initially set f_{ij}, α, g_{ij} and β to zero and update them every time a demand is routed. Let d_k be the latest demand.

Then the link cost metric is defined as follows:

$$\theta_{ij} = \frac{f_{ij}+d_k}{C_{ij}} + T_1 \times \max[0, \frac{f_{ij}+d_k}{C_{ij}} - \alpha] + T_2 \times \frac{g_{ij}}{\beta}$$

Here T_1 and T_2 are tunable parameters.

The algorithm selects the routes as least cost routes based on this link cost metric. In the metric the first term is the link utilization after the new demand is added, the second term is the increase in the maximum of link utilization due to selecting this link and the third term is the normalized criticality of the link. The second term produces optimal routes in terms of minimizing the maximum utilization.

Input: A graph $G(V, E)$, C the set of residual link capacities, P the set of distinguished ingress-egress pairs in the network, an ingress node a and an egress node b between which a flow of D units have to be routed.

Output: A route between nodes a and b having a capacity of D units of bandwidth.

Algorithm:

(1) Find maximum flow values between all $(s, d) \in P \setminus (a, b)$.

(2) Find the set of critical links for each pair.

(3) Compute criticality g_{ij} of each link ij.

(4) Find maximum criticality β in the network.

(5) Compute utilization f_{ij} for each link ij.

(6) Find maximum utilization α in the network.

(7) Compute the weights as $\theta_{ij} = \frac{f_{ij}+D}{C_{ij}} + T_1 \times max[0, \frac{f_{ij}+D}{C_{ij}} - \alpha] + T_2 \times \frac{g_{ij}}{\beta}\ \forall (i, j) \in L$

(8) Eliminate links with residual bandwidth less that D.

(9) Using Dijkstra's algorithm find shortest path between a and b based on the above link weights.

(10) Route the demand of D units from a to b and update the residual capacities in C.

Figure 1: The MIH Algorithm

The third term produces optimal routes in terms of minimizing the interference. The first term refines the link metric by making sure that unnecessarily long paths are not chosen to optimize utilization and interference. The parameters T_1 and T_2 can be tuned according to how much weight needs to be given to the three different factors. For example if T_1 is 10 and T_2 is 5, then each unit increase in the maximum utilization is equivalent to ten times the increase in the utilization of the link and twice the increase in the criticality of the link.

4 Performance study

Figure 2 shows the test network on which the algorithm was simulated. The test network has 15 nodes and 28 bidirectional links. The capacity of the light links is 1200 units and the darker links is 4800 units. The bidirectional links have the same capacity in both the directions. There are assumed to be four distinguished ingress-egress pairs $(S1, D1), (S2, D2), (S3, D3)$ and $(S4, D4)$ as indicated in the figure. The MIH algorithm was simulated and its performance was compared to the other algorithms in terms of changes in the maximum link utilizations, residual maximum flow and path length. During the simulations, source-destination pairs are randomly chosen from the four distinguished ingress-egress

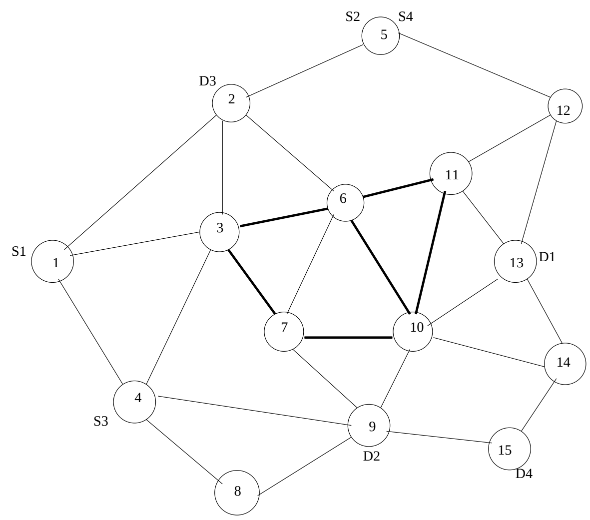

Figure 2: The test network

pairs and LSP bandwidth demands are uniformly distributed between 1 and 4 units.

We use the following legends in the graphs.

- MIH(T1, T2): Minimum Interference Hybrid algorithm with weights T1 and T2 described in the previous section.
- MIRA: Minimum Interference Routing Algorithm.
- WSP: Widest Shortest Path algorithm.
- SWP: Shortest Widest Path algorithm.
- Minhop: minimum hop algorithm.
- Hybrid(T): Hybrid algorithm with weight T as described in Section 2.

The selection of parameters $T1$ and $T2$

This section explains how parameters $T1$ and $T2$ can be fine tuned to obtain good performance. The parameter $T1$ adds weight to the utilization factor and $T2$ to the interference factor. Assigning a value of 0 to T1 will turn the algorithm into something similar to the MIRA. Similarly assigning a value of 0 to $T2$ will change it into the Hybrid algorithm. Giving more weight to $T1$ will improve the

T1	Max Utilization	Available Flow	Total Usage
0	0.6600	1836	19695
2	0.6575	1838	19701
4	0.6525	1841	19720
6	0.6483	1841	19721
8	0.6467	1841	19712
10	0.6400	1844	19724
12	0.6375	1844	19732

Table 1: Tuning parameter T1

T2	Max Utilization	Available Flow	Total Usage
0	0.6467	1841	19712
2	0.6575	2366	20203
4	0.6567	2366	20200
6	0.6567	2366	20200
8	0.6575	2366	20208
10	0.6575	2366	20208
12	0.6575	2366	20208

Table 2: Tuning parameter T2

performance of the algorithm in terms of the utilization metric, but it might have the adverse effect on the available flow and vice-versa. To arrive at the optimal values for $T1$, the value of $T2$ is fixed and the performance is measured for different values of $T1$ for a fixed number of demands. The optimal value of $T2$ is obtained similarly.

Table 1 shows the maximum utilization values, available flow values and total usage values for varying values of $T1$, when $T2$ is given a fixed value of 0 and the performance is measured for 2000 demands. As seen from this table, the value of 8 seems to be a good balance between all three metrics.

Table 2 shows the maximum utilization values, available flow values and total usage values for varying values of $T2$, when $T1$ is given a fixed value of 8 and the performance is measured for 2000 demands. As seen from this table, the value of 4 seems to be a good balance between all three metrics. For all the subsequent performance studies, these are the values used for $T1$ and $T2$.

Maximum load

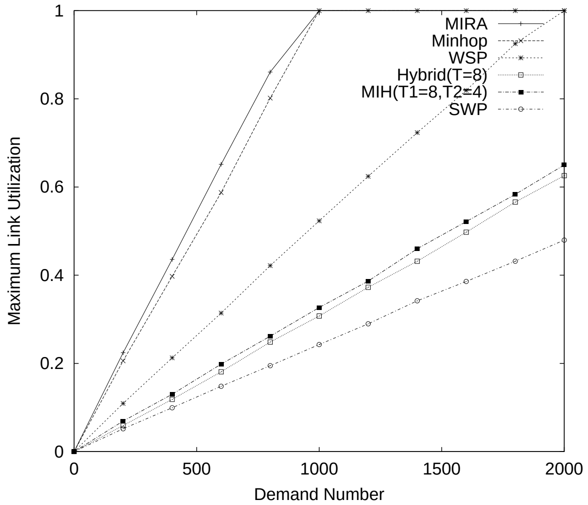

Figure 3: Comparisons of Maximum link utilizations

Figure 3 summarizes the change in the maximum link utilization for the different algorithms. As seen from Figure 3, the MIRA algorithm and the Minhop algorithm do not perform well in terms of link utilization. Their maximum link utilization reaches 1 after about 1000 demands. This is the result of unbalanced traffic in the network due to overloading only certain links in the network. The Widest Shortest Path algorithm performs better than these two algorithms. This is because among all the shortest paths, it chooses the one with the maximum residual bandwidth. The Hybrid algorithm which takes into consideration link utilization as well the path length performs better than these algorithms. The MIH algorithm performs similar to the Hybrid algorithm and is worse only than the Shortest Widest Path algorithm which optimizes only the link utilization metric.

Remaining max flow

Figure 4 summarizes the reduction in the available flow between ingress-egress pair $(S1, D1)$. As seen from the Figure 4, the minhop algorithm performs the worst. The available flow reduces rapidly and reaches 0 after about 3000 demands which means that after 3000 requests, no more paths can be established between $S1$ and $D1$. Widest Shortest Path algorithm initially performs better than the Hybrid algorithm, but as the number of demands increases its performance degrades. This could be

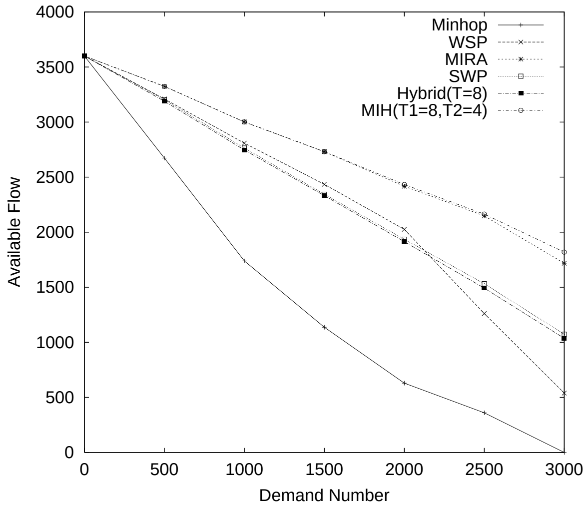

Figure 4: Available Flow

because the WSP finds the shortest path and then the widest among them. It does not check to see if certain important links are being overloaded, while other less important links with less capacity are under-utilized. The MIRA algorithm and the MIH algorithm both perform equally well initially. But as the number of demands increases, the MIH algorithm begins to perform better. More available flow means more bandwidth available for future demands. Hence MIH algorithm will result in less number of request rejections than all other algorithms.

Path Length

The metric that is used to measure the path length is the total bandwidth usage in the network which is the sum of the loads in all the links in the network. Figure 5 summarizes the total bandwidth usage in the network. As seen in figure 5, the Minhop algorithm and the Widest Shortest path algorithm show the best performance. The Hybrid algorithm also performs well since the metric it uses takes into account the path length. The Shortest Widest path algorithm as expected has the worst performance as it might favor long paths with more residual bandwidth over shorter paths. The performance of MIH algorithm lies in between these two extremes. This is because it tries to optimize on three factors as compared to one or 2 factors considered by the other algorithms.

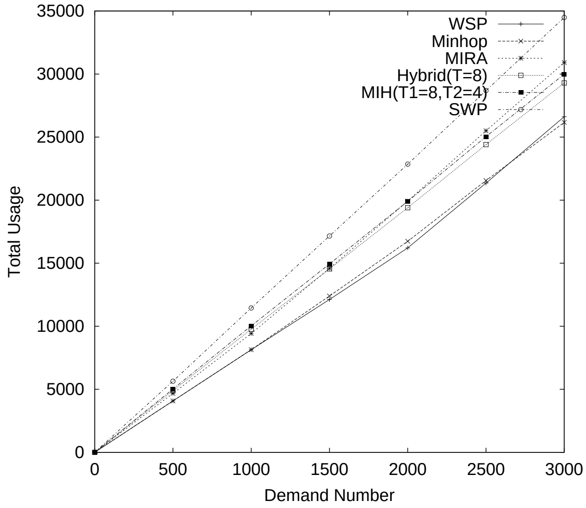

Figure 5: Total Usage

5 Conclusions

In this paper we surveyed the existing online route selection algorithms for MPLS networks. We introduced the MIH algorithm which combines different approaches to arrive on a link metric which will maintain a balance between the maximum link utilization, interference and hop count metrics. The idea behind the algorithm is to avoid loading links which will increase the maximum link utilization in the network or which are critical for satisfying future bandwidth demands between other ingress-egress pairs. We compared MIH with the existing dynamic routing algorithms for traffic engineering through simulation and we conclude that MIH achieves a balance among the three metrics.

References

[1] Murli Kodialam, T.V.Laksham, "Minimum Interference Routing with Applications to MPLS Traffic Engineering", June 2000.

[2] R.Guerin, D.Williams, A.Orda, "QoS Routing Mechanisms and OSPF Extensions", *Proceedings of globecom*, 1997.

[3] Zheng Wang, "Internet QOS Architectures and Mechanisms for Quality of Services, " Morgan Kauffman Publishers, 2002.

[4] Moy,J., "OSPF:Anatomy of an Internet Routing Protocol". New York:Addison-Wesley, 1998.

[5] Ma, Q., and P.Steenkiste. "Routing Traffic with Quality-Of-Service Guarantees in Integrated Service Networks". Proc. of NOSSDAV'98, 1998.

[6] Callon, R., A.Viswanathan, and E.Rosen. "Multiprotocol Label Switching Architecture". Internet draft (work in progress), IETF, August 1999.