Available online at www.sciencedirect.com

ScienceDirect

Computer Communications 29 (2006) 3718–3732

computer communications

www.elsevier.com/locate/comcom

Path selection with preemption and re-routing control for multi-protocol label switching networks

Chun Hau Lau *, Boon-Hee Soong, S.K. Bose

School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798, Singapore

Received 17 January 2006; received in revised form 8 June 2006; accepted 8 June 2006
Available online 30 June 2006

Abstract

Multi Protocol Label Switching (MPLS) networks enhance the services of conventional best-effort IP networks by providing end-to-end Quality of Service (QoS) guaranteed Label Switched Paths (LSP) between customer sites. The LSP has to be set up in advance before carrying the traffic. Contention for network resources may happen if many LSPs try to use a common network link with limited bandwidth. In this paper, we investigate the problem of providing services to high priority LSPs whereby existing LSPs with lower priority may be preempted. The consequent interruption of the services of preempted LSPs would detrimentally affect users' perception on the QoS provided. Therefore, the preemption strategies may incorporate additional re-routing mechanisms to provide alternative paths for the LSPs which are to-be-preempted so that their services remain unaffected. A newly arrived high priority LSP in an MPLS network may find M possible paths between its source and destination. It may select the shortest path which may trigger preemption or choose a longer path which however utilizes more resources. We begin by formulating preemption strategies with global re-routing. Our investigations include the effects of routing of high priority LSPs on the shortest path and its alternative paths. We show that by persistently routing the high priority LSP on the shortest path, more preempted LSPs can be re-routed which would reduce the negative effects of preemption. However, as excessive re-routing may degrade the network performance as well, a re-routing control strategy is proposed to constrain the length of these re-routed paths. Finally, a decentralized preemption strategy with local re-routing is also presented to approximate the performance of the proposed strategy with significantly lower control overheads. Simulations show that with this approach, high priority LSPs can gain better access to network resources while simultaneously ensuring that, as compared to the existing preemption strategies, the network throughput and the ongoing connection services are not adversely affected.

Keywords: Preemption; Resource allocation; Re-routing

1. Introduction

Multi Protocol Label Switching (MPLS) networks provide a connection-oriented mechanism to support end-to-end Quality of Service (QoS) for end users. Label Edge Router (LER) that receives incoming packets to a MPLS network will attach a short label to them. Instead of using the packet's destination address, a Label Switched Router (LSR) within an MPLS network will use this short label to look up the next hop and replaces the original label with a new one. This forwarding mechanism reduces processing time and ensures Per-Hop Behavior (PHB) of the packets that belong to the same traffic class [1]. Prior to the actual data transmission, a Label Switched Path (LSP) has to be set up to secure sufficient network resources for the new request. Each LSP can carry different priority levels and is set up to satisfy specific QoS requirements such as bandwidth, delay, jitter, and loss probability. The common signaling protocols used for LSP set up are RSVP-TE [2] and CR-LDP [3]. One of the major functionalities of MPLS networks is its support for explicit routing or Traffic Engineering (TE). In explicit routing, LSP can be routed on paths other than the shortest path to achieve network optimization objective such as load balancing [4].

---

* Corresponding author. Tel.: +65 81273482.
E-mail addresses: PH171349@ntu.edu.sg (C.H. Lau), ebhsoong@ntu.edu.sg (B.-H. Soong), eskbose@ntu.edu.sg (S.K. Bose).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.06.005

In addition to explicit routing, MPLS-Traffic Engineering (MPLS-TE) [4] also proposed preemption mechanism for MPLS networks. Two priorities are defined in which *setup priority* is used to determine whether a LSP can acquire the network resources (e.g. trigger preemption) whereas *holding priority* is used to determine the relative importance of existing LSPs. Thus, a new LPS can use its setup priority to preempt the existing LSPs of lower holding priorities. MPLS networks can support eight priority levels with values ranging from 0 to 7, with 0 as the highest priority. Even though a LSP may be assigned different set up and holding priorities, it is required that the holding priority must be higher or equal to the setup priority [4]. This is to ensure that a new LSP setup successfully will not be preempted immediately afterwards by a subsequent LSP of the same setup priority.

Although MPLS-TE enables a LSP to be set up on a relatively favorable path, it is possible that some of the links on this favorable path do not have sufficient idle bandwidth for the new LSP. In that case, the network may either preempt a number of existing LSPs on the congested links to route the new LSP on this path or route it on an alternative path. However, preemption is highly disruptive in nature because the services of preempted LSPs are forcibly terminated in order to accommodate a higher priority LSP. Loss of network throughput and revenue will occur, especially if the services provided are connection-oriented, as in video conferencing, VoIP, or video streaming. Although various admission control policies such as [5][6] can reduce the problem of preemption due to oversubscription, preemption may still arise from other reasons such as node failure or link failure. In those cases, the LSPs affected may have to be re-routed on alternative paths which could in turn be congested. In [7], a dynamic look-ahead network resource reservation is used to inform the service provider of future LSPs so that sufficient bandwidth can be provided. This strategy can effectively reduce the number of preemptions but cannot entirely eliminate it. Routing algorithms such as MIRA [8], LIOA [9], and [10] that aim at minimizing blocking ratio by reducing future interference on critical network links will simultaneously minimize preemption probability. However, without preemption, all LSPs will be blocked regardless of its priority level if the network is congested. F. Blanchy *et al.* [11] proposed a routing algorithm with preemption that minimizes the re-routing of preempted LSPs. A score function is used in the algorithm to select the path that has the least number of LSPs to be re-routed if preemption is triggered.

We observe that preemption is a practical problem which can arise from events such as oversubscription, node failure and link failure. Without loss of generality, this paper investigates the problem of preemption and re-routing in oversubscribed networks. The strategies formulated can be readily applied to the cases of node failure and link failure by simply considering the affected LSPs as new requests. The objective of the proposed strategies is to provide better resources for higher priority LSPs and minimize the loss of throughput due to preemption. In order to prevent service interruption, we propose that alternative paths are secured prior to the actual preemption. We then seek to constrain the possible explosion of re-routing events due to preemption by using a re-routing control strategy that limits the length of the alternative paths used for re-routing. An earlier version of this work has been presented in [12]. It shows that preemption with controlled re-routing can achieve better network performance. This paper differs from the earlier version by performing thorough studies on the effects of the various parameters used in the strategies and its network performance. Furthermore, a decentralized approach is proposed for easier deployment and control overhead minimization.

The rest of the paper is organized as follows: Section 2 presents the related work on preemption problems. Section 3 describes the network model used in the formulation. Section 4 presents the preemption strategies with global re-routing. The performance of the proposed strategy is presented in Section 5. Section 6 formulates the decentralized preemption strategy with local re-routing and Section 7 evaluates the performance of this decentralized strategy. Finally, Section 8 concludes the paper.

2. Related work

MPLS with LSP protection [13,14] is a mechanism to protect LSP in the event of failure. A disjoint backup path is routed side-by-side with the primary path and it is used to carry the traffic if failure occurs on the primary path. Two methods of providing LSP protection exist, i.e. *global* re-route and *local* re-route as shown in Fig. 1 and 2. In global re-route, the alternative path is established between the source and the destination. On the other hand, local re-route constructs the alternative path between the two end nodes of the failed link.

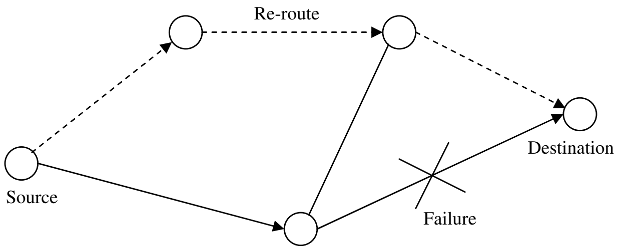

Fig. 1. Global re-route.

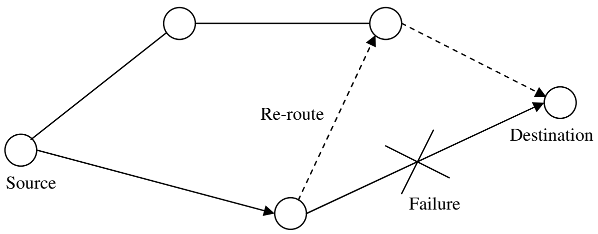

Fig. 2. Local re-route.

Although LSP protection mechanism can be used in preemption problem by setting up backup paths for low priority LSPs, it is an expensive operation because excessive resources are required to establish the backup paths. Therefore, it remains an operation reserved primarily for high priority LSPs which carry critical data. It is shown in [15–17] that local re-route for LSP protection can significantly reduce the control overhead involved and expedite the re-routing process. This local re-route approach is explored later in the formulation of decentralized preemption and re-routing strategy.

Earlier work on preemption problem is presented by Garay et al. [18] in which a systematic approach is proposed to terminate existing connections on a link if contention for resource arises. The authors have proven that the preemption problem to minimize bandwidth preempted is NP-complete. The same proof can be generalized into other preemption problems that use a different single or multiple preemption criteria. The service time based preemption approaches are presented in [19,20]. These preemption algorithms use connection service time or its expected value [20] to derive the throughput that can be possibly accrued from the existing connections and terminates the connection that has the least throughput. Therefore, the loss of throughput due to preemption is minimized. Heuristics algorithms presented in [21,22] propose preemption criteria that include bandwidth, priority level and number of connections preempted. Different orderings of criteria are used in [21] to preempt existing connections. In [22], a score function that evaluates existing LSPs based on the three criteria is developed. The LSP with the lowest score can have its bandwidth reduced or preempted in order to accommodate the new request. An evaluation of the preemption algorithm in [22] is presented in [23]. The study of preemption problem with random selection and QoS issues is presented in [24].

An optimal preemption algorithm is proposed in [25] with the same criteria as in [21]. The algorithm will first preempt existing connections from the lowest priority. A combination of connections is selected for preemption at the priority level that exceeds the bandwidth requirement. This preemption algorithm has exponential complexity. In [26], a backward connection preemption algorithm is introduced. It utilizes MPLS-TE framework to collate information about the existing LSPs and network links. A preemption decision can be then made so as to minimize the number of LSPs preempted. The application of military precedence on the MPLS technology is also proposed in [27].

Since preemption will prematurely tear down existing LSPs, end users are likely to experience service interruption and loss of data. Traditionally, preempted LSPs are rerouted after successfully setting up the new LSP that triggered preemption [4]. However, LSPs that cannot get sufficient bandwidth in the re-routing process will be terminated. The notion of *soft preemption*, proposed in [28], is used to denote the process of re-routing the to-be-preempted LSPs before termination. This mechanism enables existing LSPs to set up alternative paths before preemption so that ongoing services are not interrupted. The advantage of soft preemption is twofold: high priority LSPs can acquire sufficient bandwidth on the favorable path and the services of preempted LSPs are preserved as far as possible. Preemption with re-routing or soft preemption has been explored in [29] in which re-routable LSPs are favored for preemption because services on the LSPs are not interrupted. Although routing algorithms such as [11,30] are designed specifically to minimize re-routing in preemption, no conflict of interests exist between the routing algorithms and the preemption approach in [29]. While the routing algorithms [11,30] search for the path that triggers less preemption and thus minimizes re-routing, strategies in [29] will select the re-routable LSPs for preemption in order to minimize service interruption. A combination of both strategies will likely enhance the service continuation of lower priority LSPs.

In this paper, we investigate preemption problem in relation to the routing of high priority LSPs. A control strategy for the re-routing process is devised to prevent the avalanche of re-routing events. Decentralized routing algorithm that incorporates preemption and local re-routing is presented.

3. Network model

The network is represented by the graph, $G = (V, E)$ where V is the set of all vertices and E is the set of all edges. The total number of vertices is $N = |V|$ and the network has $N(N-1)/2$ node pairs. The set of all node pairs is represented by R, each indexed by r with source s and destination t. The edge that links two vertices $i, j \in V$, is denoted by $E_{ij} \in E$, with its bandwidth capacity given by C_{ij}. Traffic engineering extensions of the resource reservation protocol (RSVP-TE) [1] specifies that each LSP can be assigned with setup priority and holding priority. Setup priority specifies the importance of a new LSP during setup whereas holding priority specifies the relative importance of an existing LSP to hold on to the resources. To keep the integrity of the priority level, holding priority must be higher or equal to the setup priority. Without loss of generality, we assume that both the holding priority and setup priority carry the same value. MPLS can support up to eight priority levels ranging from numerical value 0 to 7, with 0 represents the highest priority. We denote the total number of priority levels supported as P with each priority level as p, where $p = (0, 1, 2, \ldots, P-1)$.

The new LSP, l_{new} that requests network resources from the MPLS network will notify its bandwidth requirement b_{new}, priority level p_{new}, and source-destination node pair r_{new}. The following notation represents the new LSP, $l_{new} = (b_{new}, p_{new}, r_{new})$. For a given node pair r, there are M numbers of possible paths through the network, each indexed by m. With the functionality of MPLS-TE [4], the network can explicitly choose the favorable path and the path chosen is not necessarily the shortest path. We order M in accordance to the ascending order of number of hops. Therefore, the first path in M is always the shortest path. The pair (r, m) identifies the mth path for the node pair r. The residual bandwidth on edge E_{ij} is represented by R_{ij}, $R_{ij} = C_{ij} - B_{ij}$, where B_{ij} is the sum of bandwidth used by the existing LSPs on E_{ij}. The network is able to admit LSP l_{new} on route (r_{new}, m) if the following condition is satisfied.

$$R_{ij} \geqslant b_{new}, \text{ for all } E_{ij} \in (r_{new}, m). \tag{1}$$

If condition (1) is violated, LSP l_{new} can choose to exhaust all the options available in M or trigger preemption to select the favorable path. This reveals a tradeoff between the selection of a favorable path but at the expense of preemption and the selection of the less favorable path. Let Q_{ij} denotes the preemptable bandwidth on edge E_{ij}, which is the sum of bandwidth from existing LSPs with priority level lower than the new LSP. For K existing LSPs,

$$Q_{ij} = \sum_{k=1}^{K} b_k \forall, \text{ for all } p_k > p_{new}. \tag{2}$$

The path (r_{new}, m) can be used to route LSP l_{new} by implementing preemption only if all the edges satisfy the following condition,

$$R_{ij} + Q_{ij} \geqslant b_{new}, \text{ for all } E_{ij} \in (r_{new}, m). \tag{3}$$

Preemption algorithm is used to determine the combination of LSPs to be preempted in order to route the new request. For the purpose of minimizing the interruption of service, soft preemption [28,29] is used to re-route the LSPs to be preempted. Alternative paths are established and the ongoing traffic is switched to the alternative paths before the existing LSPs are preempted. Although end users may notice a short period of delay caused by the execution of soft preemption, it does not affect the throughput and services provided. However, since it may not be possible to reroute all the preempted LSPs through soft preemption, a loss of network throughput will occur on LSPs that cannot execute soft preemption successfully.

Given that there are M possible paths that can be used to route the new LSP l_{new}, we can choose the path that will not trigger preemption or path that triggers preemption on one or more edges. The objective of the strategy is to maximize the number of LSPs completed and improve network throughput. Since the M possible paths are ordered with respect to the increasing number of hops, the new LSP will find that it is using more network resources as it proceeds through the path search. In order to conserve network resources, the new LSP will choose the shortest path if it has sufficient bandwidth. However, if the shortest path is not available, the LSP can trigger preemption to acquire the shortest path or choose a longer path but risk blocking more future LSPs. This consideration is particularly important to high priority LSPs because only these LSPs can obtain enough preemptable bandwidth from a congested link, and if these are routed on longer paths, then they cannot be preempted by other LSPs. The next section investigates the effects of these issues on network performance. Simulation results show that network throughput is improved while connection service disruption is reduced if high priority LSPs are constantly routed on the shortest path even though preemptions are triggered occasionally.

4. Preemption and re-routing strategy

Two variations of strategies are investigated in this section. The first strategy lets the new LSP search for all M possible paths before preemption is triggered. The second strategy explores the effects of constantly selecting the shortest path for the high priority LSP even if it is at the expense of preemption. Given M possible paths of the node pair r, we denote the hop-count difference between route $(r, m+1)$ and (r, m) as $\Delta h_{m+1,m}$.

$$\Delta h_{m+1,m} = H(r, m+1) - H(r, m). \tag{4}$$

The function H is used to compute hop counts of the pair (r, m). As the M possible paths are ordered in the ascending order of hop count, $\Delta h_{m+u,m} \geqslant 0$ for integer $u > 0$. Since preemption is only triggered when (1) is violated, it provides us the clue that, at this point, the network load is possibly within the medium to high range. The study in [31] shows that a routing algorithm that limits the hop count (i.e. shortest path) performs better than a load balancing routing algorithm in networks which are highly loaded or overloaded. This insight proves useful in the design of our preemption strategy.

4.1. Search all and preempt (SEP)

In this strategy, the new LSP with node pair r searches all the M possible routes in order to find the path that satisfy (1), i.e. all the edges on the path have sufficient bandwidth to admit the new LSP. The search is stopped immediately if a path is identified. Further search is not necessary as the subsequent paths will be of equal or higher length. However, if none of the M possible paths can admit the new LSP, preemption will be triggered at one of the routes to acquire the needed resources. A search will be carried out to find the route with the least number of edges that trigger preemption so that service interruption is minimized. If two or more paths are identified, the path is chosen arbitrarily. The rationale for SEP is to avoid preemption so that the services of ongoing LSPs are not interrupted. The new LSP will be rejected if none of the route can satisfy (3).

4.2. Limit hop count and preempt (LIP)

This strategy limits the search of possible paths and allows preemption to attain the shortest path. This is more applicable to high priority LSPs because low priority LSPs will find it harder to acquire preemptable bandwidth and thus have lower successful probability of preemption. By constantly selecting the shortest path for the high priority LSPs, these LSPs are less likely to interfere with future requests. A high priority LSP (e.g. priority 0) that is routed on a longer path not only blocks future requests, it cannot be preempted even by the other high priority LSPs. This strategy limits the path search by a threshold value, $\Delta h_{m,1} \leqslant \beta$, e.g. by assigning $\beta = 1$, the initial search space only covers paths that have at most one hop more than the shortest path. The search is stopped if a path with sufficient residual bandwidth is successfully found; otherwise the path with the least number of edges that trigger preemption will be selected. If none of the paths is feasible, the search space is expanded by one hop at a time until the search space covers all the M possible paths. This search space extension is necessary to preserve the integrity of priority levels. By adjusting the threshold β in accordance to different priority levels, the strategy enables high priority LSPs to route on the shortest path and lower priority LSPs on relatively longer paths.

4.3. Re-routing control

On the edge $E_{ij} \in (r_{new}, m)$ that triggers preemption, we propose that soft preemption is executed in order to minimize service interruption. Unlike link fault or node failure that triggers MPLS protection, preemption is more closely related to MPLS management that oversees LSP competition. No immediate tearing down of LSPs is necessary. Therefore, we can allow a grace period whereby existing LSPs are re-routed. In this re-routing process, the edge on which the preemption is triggered will send a re-routing signal to the source node of the to-be-preempted LSP one at a time. The source node is responsible to find an alternative path which does not interfere with the path used by the new LSP so that no competition of resources happens on all $E_{ij} \in (r_{new}, m)$. Global re-routing approach is used as the source node could find the shortest possible alternative path to destination.

As the network has no information whether a LSP can be re-routed successfully, the re-routing process is started from the LSP with the highest priority among the to-be-preempted LSPs until the grace period expires. In order to limit the network resource consumed by the re-routed LSP, a new threshold value α is introduced so that the hop count of the alternative path (r,m_q) does not exceed that of the original path (r,m) by more than α hops, where m_q is the alternative path.

$$H(r, m_q) - H(r, m) \leqslant \alpha, (r, m_q) \neq (r, m). \tag{5}$$

4.4. Congestion based preemption algorithm

After the re-routing process, the amount of residual bandwidth on the preemption links will be higher as some of the existing LSPs are re-routed. If $R_{ij} \geqslant b_{new}$, no tearing down of LSP is required, otherwise we will have to tear down the appropriate combination of LSPs to free up sufficient bandwidth. Since preemption mainly occurs when the network load is high, we propose that the LSP on the most congested link or the one that consumes more network resources should be terminated. This will help to ease the congestion level of the network and increase the probability of accepting future LSP without triggering preemption. The congestion level of LSP l with (r, m) is defined as

$$\lambda_{l,(r,m)} = 1 - \frac{\arg\ \min_{E_{ij} \in (r,m)} R_{ij}}{C_{ij}}. \tag{6}$$

The score function that is used to evaluate the LSPs for preemption is given by (7). It considers the hop-count difference between the currently used mth path and the shortest path, and also the congestion level of the LSP.

$$S(l) = w_1 \cdot \Delta h_{m,1} + w_2 \cdot \lambda_{l,(r,m)} \tag{7}$$

w_1 and w_2 are the associated weights. However, in order to satisfy QoS requirements, only LSPs with priority lower than the new LSP are preemptable. On the edge $E_{ij} \in (r_{new}, m)$ that triggers preemption, let B_{ij}^{p} denotes the total bandwidth of the existing LSPs at priority level p. All the LSPs at the lowest priority $P-1$ will be preempted if $B_{ij}^{P-1} < b_{new} - R_{ij}$. This process will continue to the LSPs with the next priority level until $B_{ij}^{p} > b_{new} - R_{ij}$, where only a number of existing LSPs at priority p will be preempted. Thereafter, the score function (7) is used to determine the combination of LSPs to be terminated. The algorithm will preempt with the descending order of the score function until bandwidth requirement is satisfied. This essentially means that the LSP that uses relatively more resources than the shortest path and occupies the most congested link will be preempted. Hence, the network resource consumption and the congestion level are minimized. Future LSPs may have better access to the network without the need to trigger preemption.

5. Performance evaluation

This section presents the performance of the proposed preemption strategy i.e. SEP and LIP against existing approaches. The effects of the threshold value, α and β are thoroughly investigated. Simulations are carried out on the same network topology as used in [8] and are shown in Fig. 3. The network topology consists of 15 nodes and 28 links where all the nodes can act as both source and destination. All the links are bidirectional with bandwidth capacity of 500 U. The network supports four priority levels from 0 to 3.

New LSP arrives at the network with randomly chosen source-destination pair. The bandwidth request and priority level are uniformly distributed with $U(10, 50)$ and $U(0, 3)$, respectively. LSPs arrive according to the Poisson arrival process and the service time is exponentially distributed with mean of 800 s. We investigate the network performance by varying the traffic arrival rate from 0.02 to 0.3 LSP/s.

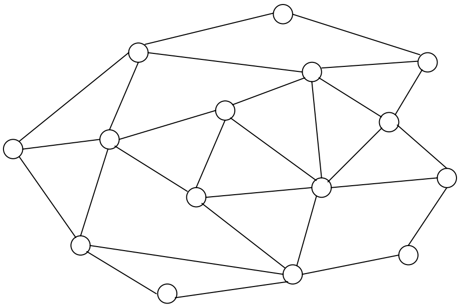

Fig. 3. NFS-NET network topology.

We first look at the performance of SEP without re-routing against the preemption algorithm proposed in [22], hereby named as PREM. Like SEP, we allow PREM to search all the M possible paths before triggering preemption. In the score function (7), the value of the first term is an integer $0 \leqslant \Delta h_{m,1} \leqslant \Delta h_{M,1}$ and the second term is a fraction between 0 and 1. Simulations show that the best performance is achieved when $w_1 = 1$ and $w_2 = 10$, in which both criteria exert relatively similar impact on the overall function. Fig. 4 shows the normalized throughput of SEP against PREM and the benchmark of no preemption. Both SEP and PREM perform worse than the network without preemption because usually more than one LSP are terminated on a preemption link in order to admit a new request. Therefore, constant execution of preemption will degrade the performance of the network. However, without preemption, all the LSPs are treated equally regardless of its priority and thus a high priority LSP may find itself being rejected due to network congestion. SEP achieves higher throughput than PREM because it is able to ease the congestion level of the network by terminating the LSP that utilizes more resources and occupies the most congested link. This reflects the fact that preemption is in proportion to the network load and that higher performance can be achieved by easing the load.

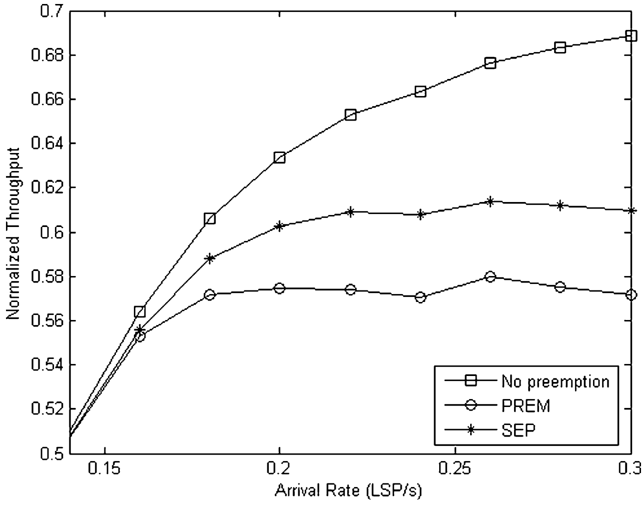

Fig. 4. Network throughput for no preemption, PREM and SEP without re-routing.

The effects of the re-routing threshold α on the network throughput and re-routing probability at the arrival rate of 0.3 LSP/s (high load) are presented in Fig. 5. Re-routing probability is defined as the ratio of the number of LSPs re-routed to the total number of LSPs selected for preemption. As α increases, the length of the alternative path used for re-routing increases proportionately. The preemption strategy [29] represents the case where α is unlimited. We notice that network throughput peaks when $\alpha = 3$ and it decreases gradually with higher α. However, re-routing probability continues to rise with respect to α. This shows that excessive re-routing of preempted LSPs will not improve the network performance; in fact it will degrade performance if control overhead associated with the re-routing process is taken into consideration. The throughput achieved at $\alpha = 3$ in Fig. 5 is higher than the throughput of SEP in Fig. 4 because re-routing strategy allows LSP services to continue uninterrupted.

The threshold β for LIP, is assigned as indicated below.

- *Case A*: ($\beta = 0$, prio 0), ($\beta = 1$, prio 1), ($\beta = 2$, prio 2).
- *Case B*: ($\beta = 1$, prio 0), ($\beta = 2$, prio 1), ($\beta = 3$, prio 2).
- *Case C*: ($\beta = 2$, prio 0), ($\beta = 3$, prio 1), ($\beta = 4$, prio 2).

In Fig. 6, the throughput performances of LIP (Case A, Case B, Case C) show significant improvement over SEP with $\alpha = 3$. This shows that strictly controlling the length of paths chosen by high priority LSPs will improve the overall network performance. When high priority LSPs constantly use shorter paths, network resource consumption and its interference on future requests are both minimized and, hence, more LSPs can be admitted overall. The improvement achieved on throughput is about 15% higher as compared to non-rerouting preemption strategy such as PREM [22]. This suggests that preemption strategy with re-routing capability, i.e. soft preemption can reduce the disruptive nature of preemption. LIP even performs marginally better than “No Preemption” at lower arrival rates where most of the preempted LSPs can anyway be

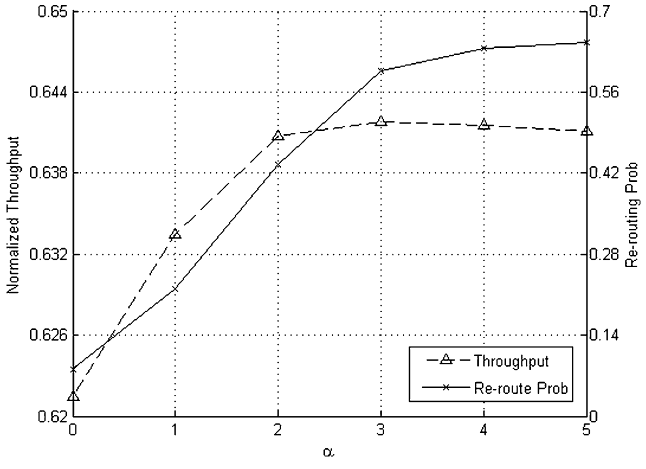

Fig. 5. The effects of α on network throughput and re-routing probability.

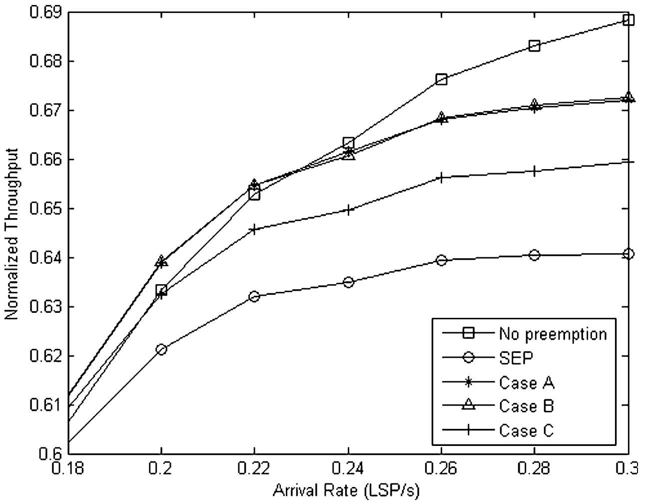

Fig. 6. Network throughput of LIP (Case A, Case B, Case C), SEP and no preemption.

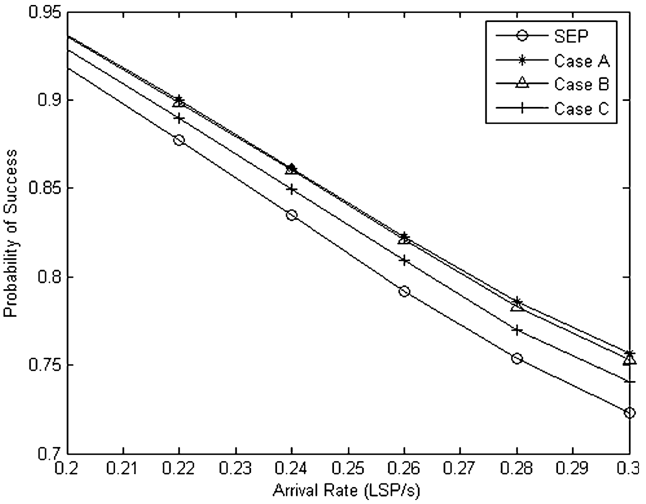

Fig. 8. Probability of success for LIP (Case A, Case B, Case C) and SEP.

re-routed. This is also confirmed by the high re-routing probabilities observed for LIP in Fig. 7 for low arrival rates, i.e. between 0.1 and 0.2 LSP/s. However, SEP cannot perform better than “No Preemption” in terms of throughput at low arrival rate because the high priority LSPs may be routed on the longer paths and thus interfere with future requests.

Fig. 8 illustrates the Probability of Success for the different LIP cases and for SEP. Probability of Success is defined as the ratio of the total number of LSPs completed successfully to the total arrival. Interestingly, Case A and Case B perform similarly in terms of throughput and success probability although Case A clearly has higher re-routing probability. The reason is that by tightly controlling the path length of high priority LSPs with smaller values of β, more preemptions are triggered thus leading to a higher re-routing probability. These extra preemption events do not contribute to the network performance. It merely indicates the underlying frequent reorganization of LSPs, in which high priority LSPs are constantly trying to acquire the shortest path. In view of this, if we take re-routing overheads into consideration, then LIP Case B would be a better strategy as fewer LSP re-routings are required.

In Fig. 9, we show the average path length vs priority levels as obtained by the three cases of LIP, the SEP and the “No Preemption” strategies for LSPs that completed successfully at 0.3 LSP/s. The results indicate that LIP can reduce the path length significantly for all the four priority levels. Without preemption, the average path length across different priority level is almost the same. For LIP, the average path length increases from priority 0 to priority 1 but decreases abruptly thereafter (the drop happens at priority 3 in Case A). This is due to the fact that at high arrival rates (0.3 LSP/s), more resources are acquired by high priority LSPs causing low priority LSPs to be re-routed on longer path which are then subject to intensive pre-

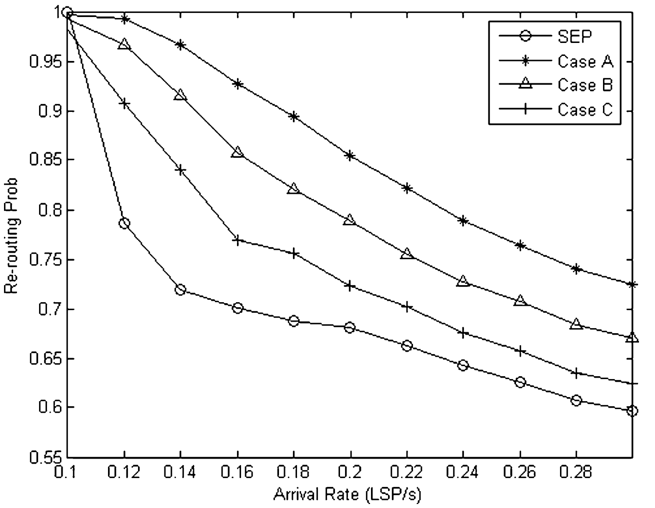

Fig. 7. Re-routing probability of LIP (Case A, Case B, Case C) and SEP.

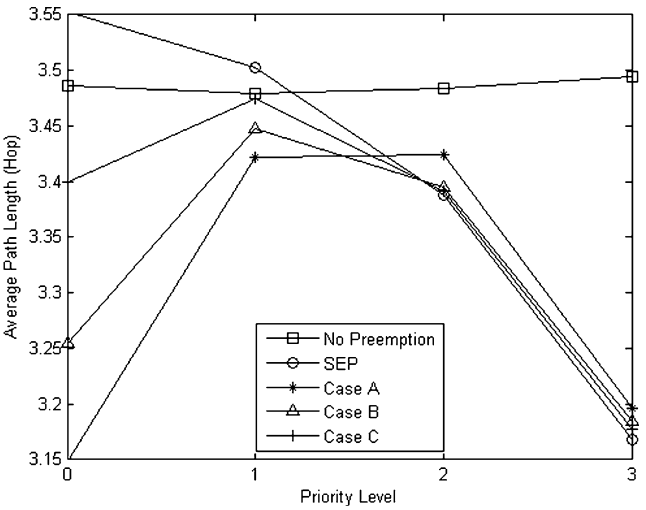

Fig. 9. Average path length of LIP (Case A, Case B, Case C), SEP and no preemption.

emption. Hence, only those low priority LSPs which get routed on the shortest path tend to be completed successfully. This is confirmed by the average path length of priority 3, which has about the same value as for priority 0. The average path length for SEP is higher than No Preemption for priority 0 and priority 1. Since preemption allows more high priority LSPs to be admitted and the way SEP searches for all possible paths before triggering preemption contribute to the higher path length observed. Therefore, LIP will be better at satisfying delay sensitive applications. The choice for the threshold value β is highly dependent on the applications supported and the network performance to be achieved. If the application is highly delay sensitive, assigning $\beta = 0$ will ensure that high priority LSPs are routed on the shortest path. However, with all the overheads taken into consideration, LIP Case B seems to be able to provide the benefits of higher throughput and resource consumption minimization. It is a balance between the strict control of Case A and good overall performance.

For LIP Case B, Fig. 10 shows the distribution of the successfully completed LSPs as per their different priority levels. Since there are four priority levels, a network that does not support preemption will treat the LSP equally and the distribution will be 25% for each of the priority levels. Accordingly, in Fig. 10, we note that the distributions are indeed close to 25% for all the priorities at low arrival rates, i.e. below 0.14 LSP/s. This is because, in this region, most of the LSPs preempted can be re-routed. We can confirm this from Fig. 7 which shows that the re-routing probabilities for LIP Case B in this region is indeed better that 90%. However, with higher traffic loads, the competition for resources increases proportionately and only LSPs with high priority are successful. The increases in the distributions for priorities 0 and 1 in Fig. 10 at higher load coincide with the degradation of performance observed for priorities 2 and 3. We note that the distribution of priority 2 rises slightly between 0.18 and 0.22 LSP/s before falling gradually and going below the 25% benchmark when the LSP arrival rate exceeds 0.28 LSP/s. This indicates that LSPs priority 2 are able to acquire resources from priority 3 without being excessively preempted by high priority LSPs when the network is not too highly congested. For operating conditions where the arrival rate is higher than 0.3 LSP/s, the network is significantly biased towards high priority LSPs; in this case, the service obtained by the lower priority LSPs of priorities 2 and 3 (especially priority 3) is very poor and decreases further with increasing load. Therefore, our preemption strategies are best suited for network with medium to high traffic load, beyond which low priority LSPs will be heavily penalized.

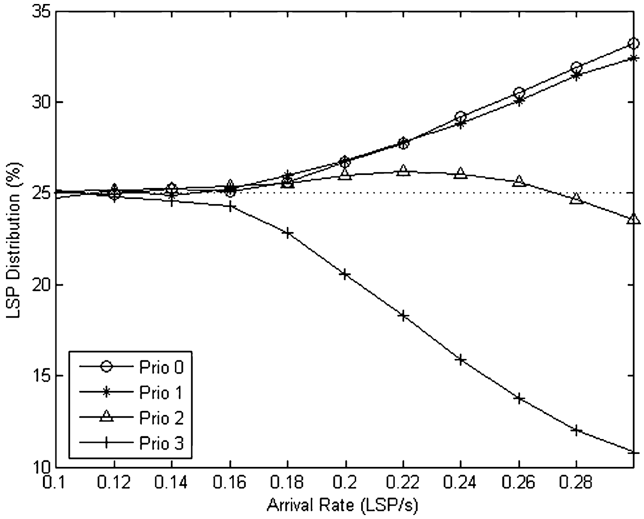

Fig. 10. Distribution of LSPs completed successfully for LIP Case B.

6. Decentralized preemption strategy

The preemption strategies presented in Section 4 have some drawbacks in real implementations. The first problem is its global re-routing approach, in which a grace period must be allocated to re-route the preempted LSPs one-by-one. The re-routing process involves sending *re-route request* signal to the source node, finding an alternative path, establishing and switching traffic to the alternative path, and terminating original LSPs. The overall cost will be k times of the re-routing cost if there are k existing LSPs to be re-routed. The overhead cost is compounded by the scenario where multiple links are involved in the re-routing process. Apart from the re-routing cost, the new LSP has to tolerate a greater set up delay as well. Secondly, the preemption algorithm requires that every link keep tracks of the paths taken by the LSPs so that the hop count and its congestion level can be properly evaluated. These challenges point to the requirements of a more decentralized approach, such that preemption and re-routing is managed by the link locally. This section attempts to reformulate the solutions by incorporating local re-routing and simple preemption algorithms which nevertheless provide results comparable to the earlier strategies. Local re-route will significantly reduce the overhead cost and delay incurred as alternative paths are only needed to be set up to bypass the preemption link.

If we compare the SEP and LIP strategies, they represent the extreme cases of searching for all possible paths and limiting the search substantially. This motivates us to design a decentralized algorithm that strikes a balance between the two. Before we proceed with the explanation of the algorithm, a useful feature exhibited by the local-rerouting approach is detailed below.

6.1. Network links segregation

We segregate the network links into four categories which will be important for the routing algorithm.

(a) *Admissible Link* – this link is ready to accept the new LSP with its residual bandwidth $R_{ij} \geqslant b_{new}$.

(b) *Preemptable Link* – this link can only accept the new LSP by triggering preemption, $R_{ij} + Q_{ij} \geqslant b_{new}$. In addition, all the preempted LSPs can be locally re-routed such that the network throughput is unaffected.
(c) *Disruptive Link* – this link differs from the preemptable link in that not all preempted LSPs can be locally re-routed. Some of the LSPs have to be terminated which will affect the network throughput.
(d) *Infeasible Link* – this link cannot accommodate the new LSP as $R_{ij} + Q_{ij} < b_{new}$.

By defining the links in these four categories, the routing algorithm can search for a path that consists of purely admissible links or with a mix of preemptable and disruptive links. From the viewpoint of network throughput, only admissible links and preemptable links are favored because existing LSPs are not terminated. However, preemptable link is associated with mandatory local re-routing and thus causes higher overhead cost. Its usage should be minimized to reduce the overall network load.

Fig. 11 shows how local re-routing may in fact reduce the overall traffic load. All the links in the topology have bandwidth equal to 50 U. LSP 1 with bandwidth 10 is originally routed on the path *GF*. LSP 2 with bandwidth 50 and source-destination (*A*, *E*) arrives later and finds that there are two paths with purely admissible links available, i.e. *ABCDE* and *AGCFE*. By choosing either one of the paths, the total bandwidth consumption of LSP 1 and LSP 2 is 210 U. However, if preemptable link *GF* is chosen by re-routing LSP 1, the total bandwidth consumption is 170 U. This choice will therefore provide benefits of overall network load minimization and better future admission success. Following this strategy, routing algorithm may be designed in a way such that the path that minimizes overall network load is selected. Unlike traditional preemption approaches [18–22] in which preemption is triggered only if path with pure admissible links are not available, we seek to proactively manage the network resources by taking the advantage of preemption. Although this active management may introduce more re-routing events, simulation results show that the network achieves better performance.

We denote the two end nodes of the link that triggers preemption as v_s and v_t, and x_{ij} as the network flow from node *i* to node *j*. A maximum flow problem [32] can be constructed to find out the amount of bandwidth that can be re-routed to bypass the link that triggers preemption.

Problem 1: Maximum flow for local re-route
Maximize z
Subject to

$$\sum_{j\in V} x_{ij} - \sum_{j\in V} x_{ji} = \begin{cases} z \text{ for } i = v_s & (8) \\ 0 \text{ for all } i \in V - \{v_s, v_t\} & (9) \\ -z \text{ for } i = v_t & (10) \end{cases}$$

$$0 \leqslant x_{ij} \leqslant R_{ij} \text{ for all } E_{ij} \in E \tag{11}$$

$$z \leqslant b_{new} - R_{v_s v_t} \tag{12}$$

Constraints (8) and (10) define the maximum flows emanating from v_s and end at v_t. Constraint (9) is for flow conservation. Constraint (11) specifies that the flow must be positive and smaller than the residual bandwidth. Constraint (12) is used to limit the maximum flow to the amount of bandwidth that needs to be preempted. By solving the above problem, a set of links with its associated flows that can be used to locally re-route the preempted LSPs will be obtained. For a preemptable link, the maximum flow is $z = b_{new} - R_{v_s v_t}$. Given the solution of Problem 1, the net extra traffic load created due to the local re-routing is given by

$$y_{v_s v_t} = \sum_{i\in V} \sum_{j\in V} x_{ij} - z. \tag{13}$$

Since disruptive link is not able to re-route all the LSPs preempted, the solution will give the set of flows with the maximum re-routable bandwidth $z < b_{new} - R_{v_s v_t}$. The amount of traffic load created due to the local re-route is also given by (13). The total bandwidth that has to be terminated is

$$u_{v_s v_t} = R_{v_s v_t} - z \tag{14}$$

With the set of flows given by the solution in Problem 1, the network will know how to divert the traffic of ongoing LSPs so that sufficient residual bandwidth can be reserved for the new LSP. In order to divert the traffic, the MPLS network can set up tunnels [2] on the links with positive flows. This process remains a local activity without the

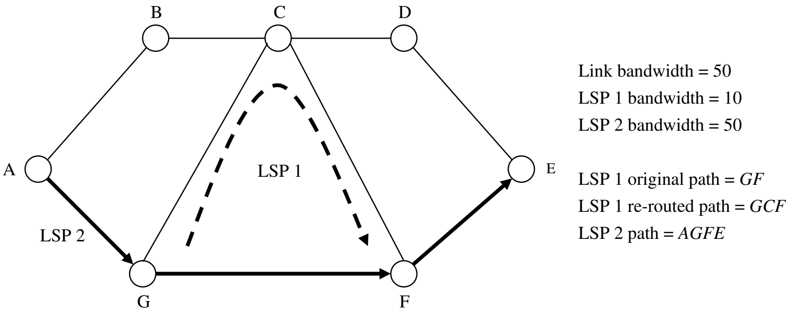

Fig. 11. Reduction of overall traffic load by local-rerouting.

need for source node participation. In MPLS architecture, the node v_s only needs to attach an extra label to the arriving packets before dispatching them to the next hop. This label helps the packets to route through the tunnel and is removed at v_t. Upon the completion of service of the LSPs using the tunnels, it will be torn down accordingly. The whole process is independent of source node and end users.

6.2. Routing and preemption algorithms

Based on the solutions given by Problem 1, network links can approximate the maximum bandwidth that can be locally re-routed. We then seek to capture the attributes of the links with respect to the four categories defined above by using the following variable, θ_{ij}.

$$\theta_{ij} = \begin{cases} b_{new} \text{ if } E_{ij} \text{ is admissible} \\ b_{new} + y_{ij} \text{ if } E_{ij} \text{ is preemptable} \\ b_{new} + y_{ij} + \gamma \cdot u_{ij} \text{ if } E_{ij} \text{ is disruptive} \\ \infty \text{ otherwise} \end{cases} \tag{15}$$

Eq. (15) is specifically designed to guide the routing algorithm in choosing the route favorable for the new request. The scaling factor $\gamma > 0$, is used to magnify the bandwidth that needs to be preempted. As the termination of on-going LSPs will interrupt network services, the use of disruptive links is made to become not as favorable as the admissible links.

Problem 2: Routing to minimize network load
Minimize$\sum_{(i,j)\in V}\theta_{ij}x_{ij}$
Subject to

$$\sum_{j\in V} x_{ij} - \sum_{j\in V} x_{ji} = \begin{cases} 1 \text{ for } i = s \\ 0 \text{ for all } i \in V - \{s,t\} \\ -1 \text{ for } i = t \end{cases} \tag{16}$$

$$x_{ij} \geqslant 0 \text{ for all } E_{ij} \in E. \tag{17}$$

The source and destination of the new LSP are represented by s and t respectively. Constraint (16) is used for flow conservation. If we set γ to a high value, the algorithm will first look for paths with purely admissible links before considering preemptable links and disruptive links. In fact, by minimizing the variable θ_{ij}, the path will include preemptable links only if the total network load including local re-route is lower than the path of purely admissible links. This approach resembles that of SEP in which longer path is favorable than the path that triggers preemption. With large γ value, disruptive links will be selected only under the condition that its absence will result in a disconnected graph from the source to destination.

In contrast, if we set γ to a low value, the routing algorithm will favor a shorter path even if preemption will be triggered, thus it resembles that of LIP. This presents the service provide with a unique factor to control the extent of preemption and the length of network path used. Furthermore, γ also serves the purpose of differentiating the various disruptive links. Given a specific γ value, disruptive links that terminate less number of existing LSPs are more favorable than those that terminate more LSPs. As a result, we minimize service interruption concurrently.

In order to make the decentralized approach more scalable, we replace the centralized preemption function (7) with function (18) to select a combination of connections from K existing LSPs on the edge E_{ij}. Preemption is triggered with the ascending order of (18)

$$S(k) = (b_{new} - R_{ij} - b_k)^2. \tag{18}$$

This changes only affect the existing LSPs at the priority level which $B^p_{ij} \geqslant b_{new} - R_{ij}$. Eq. (18) is designed such that the LSP that has the closest bandwidth in comparison to the bandwidth that needs to be preempted is selected. Therefore, the number of LSPs and bandwidth preempted will be reduced. Under this preemption strategy, the link only needs to keep track of the residual bandwidth R_{ij}, the LSPs information i.e. bandwidth and priority, and the maximum re-routable bandwidth. This significantly reduces the amount of information needed in Section 4 and minimizes the overall signaling overhead.

6.3. Multiple preemptions on single path

By solving Problem 1 and Problem 2, the network is able to find an appropriate path. However, the path chosen by the routing algorithm may consist of more than one preemptable links and disruptive links. Consequently, the different set of flows given by the solution of Problem 1 on different links may interfere. In order to resolve this problem, multi-commodity problem [32] can be used by assuming that each of the links that triggers preemption as a single commodity. Let $G' = (V',E')$ denotes the subgraph that eliminates all the edges of the path used by the new LSP, $E' = E' \cup E_{ij} \notin \{E_{sv_1}, E_{v_1v_2}, \ldots, E_{v_nt}\}$. The multi-commodity problem is defined as follows,

Problem 3: Multiple re-routes
Maximize$\sum_{1\leqslant a\leqslant A}z^{(a)}$
Subject to

$$\sum_{j\in V'} x^{(a)}_{ij} - \sum_{j\in V'} x^{(a)}_{ji} = \begin{cases} z^{(a)} \text{ for } i = v^{(a)}_s, a = 1,2,\ldots,A \\ 0 \text{ for all } i \in V' - \{v^{(a)}_s, v^{(a)}_t\}, a = 1,2,\ldots,A \\ -z^{(a)} \text{ for } i = v^{(a)}_t, a = 1,2,\ldots,A \end{cases} \tag{19}$$

$$\sum_{1\leqslant a\leqslant A} x^{(a)}_{ij} \leqslant R_{ij} \text{ for all } E_{ij} \in E' \tag{20}$$

$$x^{(a)}_{ij} \geqslant 0 \text{ for all } E_{ij} \in E', a = 1,2,\ldots,A \tag{21}$$

$$z^{(a)} \leqslant b_{new} - R^{(a)}_{v_sv_t}, a = 1,2,\ldots,A \tag{22}$$

Each commodity is denoted by a, which represents the preemptable link or disruptive link. For example, if there are two preemptable links $A = 2$, then the set of flows $x^{(1)}_{ij}$ is used to re-route the LSPs on the first preemptable link and $x^{(2)}_{ij}$ for the second link. Constraint (20) ensures that the total flows on a single link will not exceed the residual bandwidth. After obtaining the solution for Problem 3, the network will know the amount of bandwidth to be re-routed, how to re-route and trigger preemption accordingly before admitting the new LSP.

However, most of the exact algorithms for the solving multi-commodity flow problem have long running time (even though polynomial) [33]. Thus, we propose that sequential preemption is used, in which multiple links that trigger preemption will do so in successive manners. Every preemption link will take turn to execute Problem 1 again to resolve the links that can be used for re-routing. LSPs re-routing and termination (if necessary) will be executed before the next preemption link starts the preemption process. The advantage of implementing sequential preemption is that interference of network resources is less likely to happen. LSPs that are terminated in the previous links may free up more resources for subsequent links to re-route existing connections. The flow diagram of the decentralized strategy is illustrated in Fig. 12.

6.4. System complexities

For the SEP and LIP strategies, the network needs to search for multiple possible paths. This can be resolved by using k-shortest paths (KSP) routing algorithm. The time complexity for finding the k paths is given by $O(k * N^2 \log N)$ [34], where N is the number of nodes. Given a link that triggers preemption, if there are n numbers of existing LSPs with lower priority levels than the new LSP, KSP has to be run n times in order to try to re-route the n existing LSPs. After re-routing, the network may have to terminate a combination of existing LSPs on different links to free up sufficient network resources. For n_r numbers of remaining LSPs with lower priority levels, the network needs at most $O(n_r^2)$ in time complexity to attain the desired combination. Hence, the overall time complexity for SEP and LIP is approximated by $O[k * N^2 \log N + n_p(n * k * N^2 \log N + n_r^2)]$, where n_p is the number of links that triggers preemption. Since $n_r \leqslant n$, the time complexity is primarily dominated by KSP.

In the decentralized algorithm, the maximum flow problem (Problem 1) can be solved by using Edmonds–Karp (EK) algorithm [35] with the complexity of $O(N|E|^2)$, where $|E|$ is the number of links. Problem 2 can be solved by Dijkstra's algorithm with the complexity of $O(|E| + N \log N)$. In Problem 3, we propose that sequential preemption is used in which the maximum flow problem is solved in successive manners on every links that trigger preemption. The time complexity for Problem 3 is thus given by $O(n_p * N|E|^2)$. Similarly, the network needs $O(n_r^2)$ on each link to select the combination of LSPs for termination if necessary. The overall time complexity for the decentralized algorithm is approximated by $O[(|E| + N \log N) + n_T(N|E|^2) + n_p(N|E|^2 + n_r^2)]$, where n_T is the total number of links that run Problem 1 to check if the link is a preemptable link or disruptive link. This overall complexity is highly dominated by the EK algorithm.

Generally, for a fully connected network with $N \geqslant 3$, the number of links $|E| \geqslant N$. Thus the computation time for EK algorithm will grow faster than KSP. However, for a reasonably sparse graph such as the NSF-NET in Fig. 3, both SEP (or LIP) and the decentralized algorithm will have comparable computation time. Furthermore, the decentralized algorithm gains added advantage through its simplicity and scalability. No excessive per LSP information is needed. The network links only need to keep the pri-

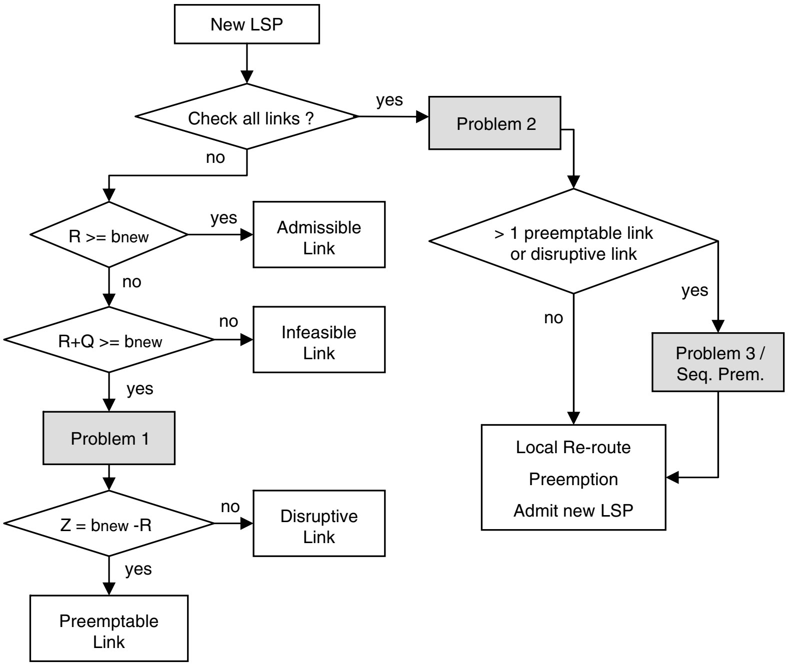

Fig. 12. The flow diagram of the decentralized strategy.

ority level and bandwidth information. As re-routing is handled by links locally, no per LSP global re-routing is required which could minimize new LSP setup delay. Network service provider is able to use the single scaling factor, γ to make a decision on the tradeoff between selecting shorter path but triggering higher numbers of preemption or vice versa.

7. Performance comparisons

This section compares the performance of decentralized preemption strategy against LIP and SEP. We have used the same network topology and simulation parameters here as in Section 5. LIP in this section refers to Case B as indicated in Section 5. For the decentralized preemption strategy (DE), Problems 1, 2 and 3 cited above are solved by using the CPLEX optimizer [36]. Nevertheless, the results obtained will be the same if Edmonds–Karp algorithm and Dijkstra's algorithm are implemented. We solve Problem 3 by using both the original formulation as well as sequential preemption for the purpose of performance comparisons. The results based on sequential preemption are used for all the DE performances reported below. We set the scaling factor, γ to 5 and C in the simulations, where C is the link capacity. At $\gamma = 5$ (DE, $\gamma = 5$), the algorithms will choose shorter path and more preemptions will be triggered. Conversely, at $\gamma = C$(DE, $\gamma = C$), the network will tend to search for the path that has high numbers of admissible links, for which the path could be longer.

Fig. 13 presents throughput performance of different preemption strategies at increasing traffic load. The preemption strategy PREM [22] has low throughputs because it does not incorporate re-routing. Fig. 13 also indicates that decentralized strategy (DE, $\gamma = C$) has similar performance to SEP. This is due to the fact that the strategy is trying to including as many admissible links and preemptable links as possible before considering disruptive links. If the shorter paths are congested, the strategy will select longer path so as to avoid preemption. Thus, the results are similar to SEP as expected. On the other hand, we notice that significant throughput improvement is shown by (DE, $\gamma = 5$). By lowering γ, the strategy may favor a shorter path that includes preemptable links and disruptive links than a longer path of purely admissible links. As such, the strategy resembles LIP. This result also confirms that better throughput can be attained by limiting the path length, which in effect minimizes total network resources utilized.

In our simulations, we observe that slight degradation of throughput is observed for $\gamma < 5$. As the network triggers more preemption, some of the existing LSPs may not be re-routed successfully. These LSPs may be terminated which would lead to lower throughput. The performance difference between (DE, $\gamma = 5$) and LIP is primarily the effect of global re-routing. As the LSPs are re-routed from its source to destination in global re-routing, they have better opportunities at finding shorter paths, thus minimizing overall network resource consumption.

Network throughput is not the sole performance parameters for preemption strategies. For the sake of completeness, we introduce *service disruption* rate [29] in Fig. 14. It is defined as the ratio of LSPs terminated to the total number of LSPs uninterrupted throughout its lifetime. Given that a LSP is admitted, this measurement gives us the idea how probable the LSP will be terminated due to preemption. This measurement is important especially if revenue is only generated when a LSP is completed successfully. The result shows that PREM terminates close to 35% of existing LSPs at high traffic load due to no re-routing. We notice that network throughput performance is reflected in the service disruption rate. Lower service disruption rate corresponds to higher throughput and vice versa. Since only low priority LSPs can be preempted, lower service disruption rate also means higher completion rate for low priority LSPs. Hence, the strategies designed not only provide access

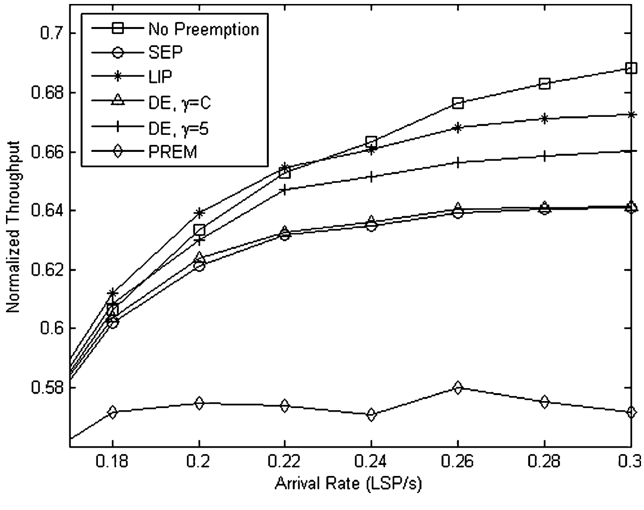

Fig. 13. Throughput comparison of various preemption strategies.

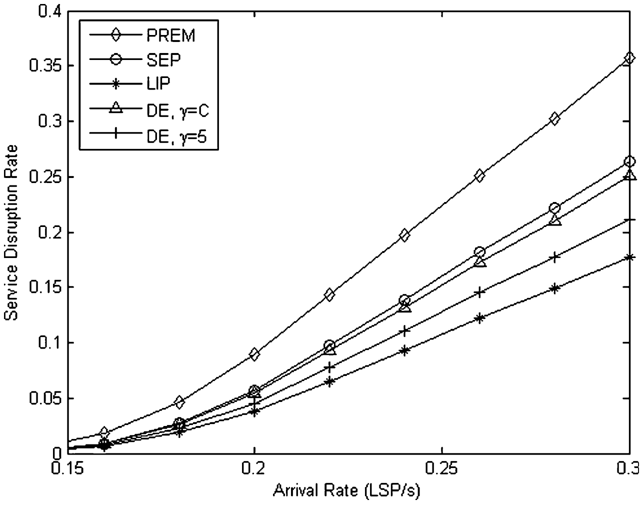

Fig. 14. Service disruption rate of SEP, LIP and decentralized strategies.

to high priority LSPs but also minimize the degradation of services on low priority LSPs.

Figs. 15 and 16 show the preemption probability and re-routing probability respectively. The preemption probability represents the probability that a new LSP arrival will trigger preemption whereas re-routing probability is a measurement of the ratio of LSPs re-routed to the total number of LSPs selected for preemption. By limiting the path length, both LIP and (DE, $\gamma = 5$) show higher preemption probability as the network will trigger preemption (if necessary) to route the new LSP on a shorter path. However, the network performance is not highly penalized as a high percentage of the LSPs preempted can be re-routed successfully. As LSPs are constantly routed on the shorter paths, the overall traffic load imposed on the network is reduced such that more LSPs can be re-routed. This observation explains the performance of SEP and (DE, $\gamma = C$) where the strategies trigger lower number of preemptions but a higher percentage of LSPs cannot be re-routed. Although LIP and (DE, $\gamma = 5$) are relatively more active in managing the network resources through preemption, we believe that the higher control overhead ensued is acceptable as the network achieves better performance. Furthermore, the γ value can be fine tuned to find a better operating point that fits the network performance objective.

Fig. 17 shows the average path length for different strategies at the arrival rates of 0.3 LSP/second. From the average path length of LSPs with priority 0 (highest priority), we notice the extent of path searching process. Decentralized strategies with different γ values find a balance between SEP and LIP. Although (DE, $\gamma = C$) and SEP show similar performance in network throughput, we notice that (DE, $\gamma = C$) has a shorter path length. This is mainly contributed by preemptable links in which the network can choose the shorter path if the overall traffic load offered is reduced. A check on the LSPs distribution shows that all the schemes have similar performance as illustrated in Fig. 10.

Given the three preemption strategies proposed, we find out that LIP provides the best performance but that it comes with extensive network resource management as the network triggers preemption and re-routing events greedily. This may introduce very large amounts of control overhead. The decentralized strategy presents us with a unique scaling factor, γ which we can use to adjust the tradeoff between better performance and higher preemption. Furthermore, the decentralized strategy is simpler and more scalable as compared to SEP and LIP as no per-LSP information is needed. The network link only needs to keep track of its residual bandwidth and some simple LSP information such as priority and bandwidth. The primary reason that LIP performs better than the decentralized strategy is the effectiveness of global re-routing. Global re-route is able to find a better alternative path because it is source routed, in which the alternative path may be shorter than the original path. Furthermore, LSPs originating from different source-destination pairs have a better chance to be re-routed through global re-routing. In comparison, local re-route will constantly introduce

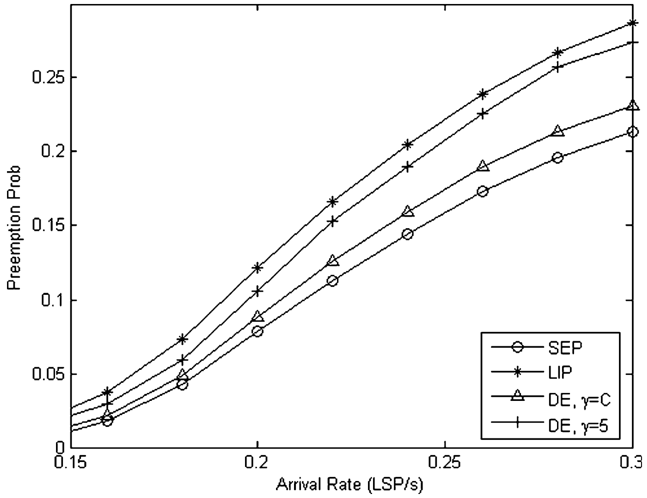

Fig. 15. Preemption probability of SEP, LIP and decentralized strategies.

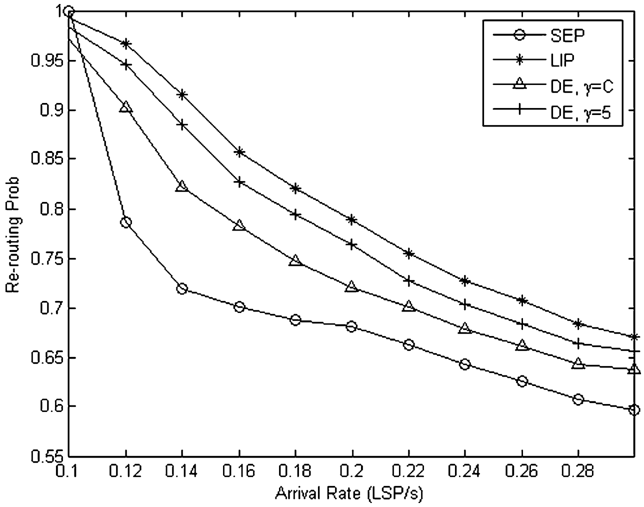

Fig. 16. Re-routing probability of SEP, LIP and decentralized strategies.

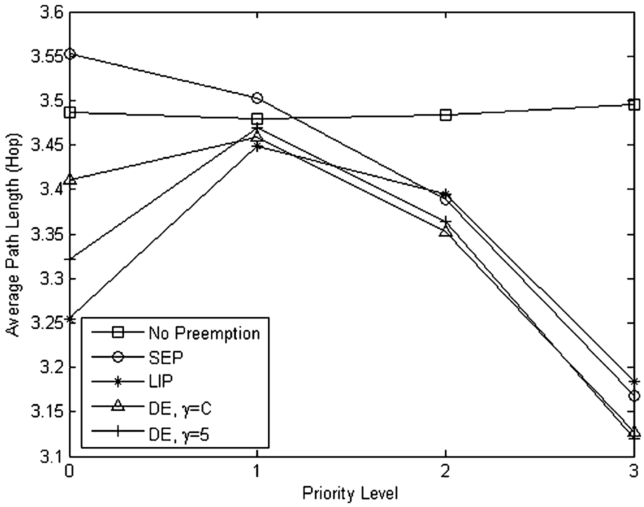

Fig. 17. Average path length of SEP, LIP and decentralized strategies.

more hops than the original path as the preempted LSPs can only take the routes that bypass the preemption link.

The Problem 3 defined in Section 6 introduces extra computational complexities to the decentralized strategy. However, in most of the cases, fewer than 15% of the LSPs admitted use more than one preemptable link or disruptive link. Therefore, the results obtained through sequential preemption are comparable to the original formulation. By using sequential preemption, LSPs terminated on the previous links may free up more resources to re-route LSPs on the subsequent links. This implication narrows the performance gap between the exact solution based on Problem 3 and sequential preemption. A check on the network throughput shows that no more than 2% gain is realized through the exact solutions. Similarly, service disruption rate is reduced by no more than 3%.

8. Concluding remarks

In this paper, three preemption schemes with re-routing mechanisms are presented. The SEP scheme allows LSPs to search for all possible paths before triggering preemption to acquire network resources. The second scheme, LIP, seeks to limit the search space so that new high priority LSPs are assigned with shorter paths. In both schemes, global re-routing mechanisms are used to route the to-be-preempted LSPs on alternative paths so that network throughput will not be adversely affected. Our studies show that networks achieve better throughput by implementing LIP. This is due to the fact that by routing the high priority LSPs on shorter paths, the overall network resource consumption is minimized and thus more future LSPs can be admitted. SEP shows poorer performance because the high priority LSPs that search all the possible paths may at times use relatively longer paths. For ease of implementation, a decentralized preemption scheme with local re-routing is formulated which gives comparable results to SEP and LIP. The decentralized scheme uses routing algorithm to find the path that consumes minimum network resources. It features a unique scaling factor that can be used to adjust the extent of path length for LSPs, thus gives us the flexibility to control the network at a desired operating condition. In order to minimize the loss of throughput, links that can locally re-route the existing LSPs are selected ahead of the links that have to terminate most of the existing LSPs. The results indicate that by more actively managing the network resources, this can thereby satisfy the requirements of higher priority LSPs and achieve better overall performance at the same time.

The implementation of any one of the schemes is application specific. For application that needs strict control on path length and delay, LIP provides the best result but with the cost of higher control overhead. Although the decentralized strategy is not able to perform as well as LIP, it involves less control overhead and complexity. Therefore, decentralized strategy is more suitable for applications that do not have high requirements. For future work, we will consider the implementation of these schemes on MPLS testbed for performance evaluation. A centralized server is needed for SEP and LIP schemes in order to make the preemption and re-routing decisions whereas the decentralized scheme can be deployed on routers. Control overheads and network stability will be investigated.

References

[1] E. Rosen, A. Viswanathan, R. Callon, Multiprotocol label switching architecture, IETF RFC 3031 (2001).

[2] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RSVP-TE: extensions to RSVP for LSP tunnels, IETF RFC 3209 (2001).

[3] B. Jamoussi, L. Anderson, R. Callon, R. Dantu, L. Wu, P. Doolan, et al., Constraint-based LSP setup using LDP, IETF RFC 3212 (2002).

[4] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. Mcanus, Requirements for traffic engineering over MPLS, IETF RFC 2702 (1999).

[5] Z. Duan, Zhi-Li Zhang, Y.T. Hou, Lixin Gao, A core stateless bandwidth broker architecture for scalable support of guaranteed services, IEEE Trans. Parallel Distributed Syst. 15 (2) (2004).

[6] R. Hill, H.T. Kung, A diff-serv enhanced admission control scheme, IEEE Global Telecommun. Conf. (Globecom) (2001).

[7] I. Ahman, J. Kamruzzaman, S. Aswathanarayaniah, Dynamic look-ahead time in book-ahead reservation in Qos-enabled networks, in: Proc. of 12th IEEE Int'l Conf. on Networks, 2004.

[8] K. Kar, M. Kodialam, T.V. Lakshman, Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications, IEEE J. Selected Areas Commun. 18 (12) (2000).

[9] A.B. Bagula, M. Botha, A.E. Krzesinski, Online traffic engineering: the least interference optimization algorithm, IEEE Int. Conf. Commu. (2004).

[10] A. Capone, F. Martignon, Analysis of dynamic Qos routing algorithms for MPLS networks, IEEE Int. Conf. Commun. (2004).

[11] F. Blanchy, L. Mélon, G. Leduc, Routing in MPLS network featuring preemption mechanisms, IEEE Int. Conf. Telecommun. (2003).

[12] C.H. Lau, B.-H. Soong, S.K. Bose, A new preemption strategy with re-routing control for MPLS networks, IEEE Fifth International Conference on Information, Communications and Signal Processing (2005).

[13] J.L. Marzo, E. Calle, C. Scoglio, T. Anjali, QoS online routing and MPLS multilevel protection: a survey, IEEE Commun. Mag. (2003) 126–132.

[14] J.L. Marzo, E. Calle, C. Scoglio, T. Anjali, Adding QoS protection in order to enhance MPLS QoS routing, IEEE Int. Conf. Commun. (2003).

[15] L. Li, M.M. Buddhikot, C. Chekuri, K. Guo, Routing bandwidth guaranteed paths with local restoration in label switched networks, IEEE J. Sel. Area. Commun. 23 (2) (2005).

[16] S. Raza, F. Aslam, Z.A. Uzmi, Online routing of bandwidth guaranteed paths with local restoration using optimized aggregate usage information, IEEE Int. Conf. Commun. (2005).

[17] L. Melon, F. Blanchy, G. Leduc, Decentralized local backup LSP calculation with efficient bandwidth sharing, IEEE Int. Conf. Commun. (2003).

[18] J.A. Garay, I.S. Gopal, Call preemption in communication networks, Infocom (1992) 1043–1050.

[19] A. Bar-Noy, R. Canetti, S. Kutten, Y. Mansour, B. Schieber, Bandwidth allocation with preemption, SIAM J. Comput. 28 (5) (1999) 1806–1828.

[20] C.H. Lau, B.-H. Soong, S.K. Bose, SERA: service ratio based preemption for minimizing throughput loss in networks, IEEE Tencon (2005).

[21] M. Peyravian, A.D. Kshemkalyani, Decentralized network connection preemption algorithms, Comput. Netw. ISDN Syst. (1998).

[22] J.C. de Oliveira, C. Scoglio, I.F. Akyildiz, G. Uhl, New preemption policies for DiffServ-aware trraffic engineering to minimize rerouting in MPLS networks, IEEE/ACM Trans. Netw. 12 (4) (2004).

[23] R.C. Vieira, P.R. Guardieiro, A proposol and evaluation of a LSP preemption policy implemented with fuzzy logic and genetic algorithms in a Diffserv/MPLS test-bed, Proc. Int. Conf. Commun. Circ. Syst. (2005).

[24] V. Stanisic, M. Devetsikiotis, A dynamic study of providing quality of service using preemption policies with random selection, IEEE Int. Conf. Commun. 3 (2003) 11–15.

[25] Sung-eok Jeon, R.T. Abler, A.E. Goulart, The optimal connection preemption algorithm in a multi-class network, IEEE Int. Conf. Commun. (2002).

[26] Li. Lei, S. Sampalli, Backward connection preemption in multi-class QoS-aware networks, IEEE Int. Workshop Qual. Ser. (2004).

[27] V. Fineberg, Specification of the military precedence and preemption in the DS-TE networks, IEEE Mil. Commun. Conf. (2004).

[28] M.R. Meyer, D. Maddux, J.-P. Vasseur, C. Villamizar, A. Birjandi, MPLS traffic engineering soft preemption, Draft-ietf-mpls-soft-preemption-01 (2003).

[29] C.H. Lau, Boon-Hee Soong, S.K. Bose, Preemption with re-routing to minimize service disruption, IEEE ICCS (2004) 521–525.

[30] B. Szviatovszki, Á. Szentesi, A. Jüttner, Minimizing re-routing in MPLS networks with preemption-aware constraint-based routing, Comput. Commun. 25 (2002) 1076–1084.

[31] Q. Ma, P. Steenkiste, On path selection for traffic with bandwidth guarantees, IEEE Int. Conf. Netw. Protocols (1997) 191–202.

[32] R.K. Ahuja, T.L. Magnant, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, Englewood Cliffs, NJ, 1993.

[33] B. Awerbuch, T. Leighton, Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks, Proc. ACM Symp. Theory Comput. (1994) 487–496.

[34] J. Yen, Finding the K shortest loopless paths in a networks, Manage. Sci. 17 (11) (1971) 712–716.

[35] J. Edmonds, R.M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, J. ACM 19 (2) (1972) 248–264.

[36] <http://www.ilog.com/products/cplex/>.

Chun Hau Lau received his B.Eng in Electrical and Electronics Engineering from Universiti Teknologi Malaysia. He joined Singapore Technologies (ST) Electronics briefly as a Project Engineer before taking part in the Research and Training Program (RTP) initiated by the Singapore's Economic Development Board (EDB). He is currently working towards his Ph.D. degree in Nanyang Technological University, Singapore. His research interests include MPLS networks, routing protocols, SPN modelling and wireless sensor networks.

Dr. Boon-Hee Soong received his B.Eng. (Hons I) degree in Electrical and Electronic Engineering from University of Auckland, New Zealand and the Ph.D. degree from the University of Newcastle, Australia, in 1984 and 1990 respectively. He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University. From October 1999 to April 2000, he was a Visiting Research Fellow at the Department of Electrical and Electronic Engineering, Imperial College, UK under the Commonwealth Fellowship Award. He also served as a consultant for Mobile IP in a recent technical field trial of Next-Generation Wireless LAN initiated by IDA (InfoComm Development Authority, Singapore). He is holder to Tan Chin Tuan Fellowship award with an attachment at Duke University in June 2004. He has served as Technical Program Committee for several conferences Globecom and ISWCS. He has been teaching a number of subjects related to the field of Network Performance and wireless communications. His area of research interests includes Sensor Networks, Mobile Ad-hoc networks, mobility issues, optimization of wireless networks, routing algorithms, system theory, Quality of service issues in high-speed networks, and signal processing. He has published over 100 international journals and conferences. He is currently a Senior member of IEEE and a member of ACM.

Dr. S.K. Bose joined the Nanyang Technological University in Singapore in June 2003 after more than twenty years with the faculty of the Department of Electrical Engineering at the Indian Institute of Technology in Kanpur, India as Assistant Professor (1982-1991) and Professor (1991-2003). Early in his career, he was associated with the Corporate Research and Development Center of the General Electric Co. at Schenectady, New York where he worked on power line communications, optical fiber communications and mobile satellite communications. During his stay in I.I.T., Kanpur, Dr. Bose was a founding member of the ERNET Project team which designed, set up and ran the ERNET computer network interconnecting the major research and academic institutions in India between themselves and the Internet. He was also involved in consultancy work with the Indian Space Research Organization in the areas of mobile satellite networks and satellite based Internet access. He has been actively involved in the organization of several major conferences and has also been providing volunteer support to IEEE activities. He has published numerous books, journal and conference papers in the areas of microprocessors, computer networks and queuing systems. He also maintains a very popular web site on queueing systems at http://www.ntu.edu.sg/home/eskbose/qbook/qbook.html. Dr. Bose is a Senior Member of IEEE and a member of Sigma Xi and Eta Kappa Nu.