CAIP: A Restoration Routing Architecture for DiffServ Aware MPLS Traffic Engineering

Fahad Rafique Dogar, Zartash Afzal Uzmi and Shahab Munir Baqai
Computer Science and Engineering
Lahore University of Management Sciences, Pakistan
{fahad,zartash,baqai}@lums.edu.pk

***Abstract*— We propose an architecture for MPLS restoration routing of DiffServ traffic. This architecture, namely Per Class Aggregate Information With Preemption (CAIP), facilitates provisioning of two key QoS features for multimedia traffic: prioritized guaranteed bandwidth and fast restoration in the event of an element failure. The CAIP architecture is scalable and requires propagating only per-class aggregate link usage information; such information can be readily piggybacked on link state routing packets using traffic engineering extensions to link state routing protocols [1]. CAIP results in improved bandwidth sharing compared to Simple Aggregate Information Scenario (SAIS), resulting in fewer LSP rejected requests and a greater amount of active bandwidth placed on the network. On average, CAIP rejects 881 LSP requests compared to 1010 rejected LSP requests in SAIS for a typical ISP network. Similarly, CAIP is able to place 220 units of bandwidth compared to 180 units of bandwidth placed in SAIS, thus showing an average improvement of about 22%. CAIP allows precise computation of preemptable bandwidth for an arbitrary set of prioritization requirements put forth by the service providers. We present a case study of service provider requirements and computation of preemptable bandwidth for those requirements. CAIP can be integrated with those restoration routing schemes that make use of propagating aggregate link usage information. Furthermore, existing preemption schemes can be used with CAIP in order to decide the actual LSPs which need to be preempted.**

***Index Terms*— Restoration Routing, DiffServ, MPLS-TE, Preemption, Bandwidth Sharing**

I. Introduction

The increasing demand for multimedia applications such as VoIP and video streaming has created new challenges for network service providers. The requirements for delay, throughput and packet loss for such applications are inherently different from those of data traffic. The IP service, which is characterized as best effort and performs satisfactorily with data traffic, fails to provide the desired Quality of Service (QoS) guarantees for multimedia applications. There is a need for providing such guarantees by differential treatment of the traffic for these multimedia applications.

Differentiated Services (DiffServ) architecture allows traffic differentiation on the basis of traffic class [2]. Packets are marked onto predetermined DiffServ classes at the edges. This marking determines the Per Hop Behavior (PHB) received by the packet at all the subsequent hops. The PHB is achieved through a combination of scheduling and queue management schemes and ensures that varying requirements of different traffic types are considered. While DiffServ specifies traffic differentiation to achieve higher service quality, the efficacy of DiffServ architecture is limited by the traditional shortest path routing employed by IP. This shortcoming can be overcome through the traffic engineering (TE) capabilities offered by Multi-protocol Label Switching (MPLS). MPLS-TE enables establishing explicit routes using constraint based routing of bandwidth guaranteed label switched paths (LSP) [3]. Significant research effort has recently been directed towards integrating DiffServ and MPLS [4], [5].

Bandwidth guarantees provided in MPLS are affected by failures of nodes and links in the network. Therefore, *restoration routing* mechanisms have been proposed which enhance the QoS guarantees provided to an LSP [6]–[8]. Restoration routing requires establishing backup paths which are node and link disjoint from the primary path. When a network element fails, traffic is redirected onto these backup paths until re-optimization of the primary path takes place. The re-optimization process requires a few seconds and it is assumed that further failures would not occur in the network during this short period [6], [9]. This assumption allows bandwidth sharing along those backup paths which would not be activated simultaneously, resulting in improved network utilization. Therefore, bandwidth sharing is used as the single most important criterion while evaluating the performance of a restoration routing scheme.

In this paper, we propose a restoration routing architecture for DiffServ aware MPLS traffic engineering. Our architecture, Per Class Aggregate Information With Preemption (CAIP), relies on propagating per-class aggregate link usage information. We show that propagating per-class aggregate link usage information results in improved bandwidth sharing compared to the case where aggregate link usage information is propagated. Moreover, CAIP allows precise calculation of preemptable backup bandwidth based on the requirements of the network service provider. We show the calculation of the preemptable bandwidth for predetermined rules as a case study. CAIP can also be integrated with existing restoration routing schemes which are based on propagating aggregate link usage information. Furthermore, existing preemption schemes [10], [11] can be used with CAIP in order to decide the actual LSPs which are to be preempted.

The rest of the paper is organized as follows: In Section II, we explain the background material related to restoration routing. In Section III, we present the details of our proposed

architecture, CAIP. A case study is presented in Section IV. Finally, we give our conclusions in Section V.

II. Restoration Routing

Restoration routing provides protection against failures of links and nodes. Without any loss of generality, we only consider link protection in this paper. Switching from the primary path to a backup path must occur at a node that is upstream the point of failure along the primary path. There are two levels of restoration routing based on how further upstream along the primary path is the node that switches the primary traffic onto a backup path. In *path restoration*, a single backup path, which originates from the head end of the primary LSP, provides protection. In *local restoration*, each LSP passing through a possible failure point is protected by a backup path which originates from the node immediately upstream to this failure point. Although local restoration provides faster restoration, it requires higher bandwidth reservations compared to path restoration. This resource burden can be reduced by allowing both *inter-demand* and *intra-demand* sharing. In inter-demand sharing, backup paths protecting different primary LSPs can share bandwidth whereas in intra-demand sharing backup paths protecting different failure points of the same primary LSP can share bandwidth. The optimal calculation of primary and backup paths is made difficult due to the distributed and online nature of LSP requests. A centralized path computation server maintaining complete information about the network state can make optimal bandwidth sharing decisions compared to a distributed approach which would require significant routing protocol overhead in order to make a similar decision. The online nature of LSP requests implies that no a priori information about future requests is assumed. The most widely referenced online algorithm for local restoration relies on propagating aggregate link usage information [6]. We refer to this information scenario as the simple aggregate information scenario. In this scenario, the propagated information for every link (i, j) includes:

- $\boldsymbol{F_{ij}}$: Total bandwidth reserved on link (i, j) for primary LSPs
- $\boldsymbol{G_{ij}}$: Total bandwidth reserved on link (i, j) for backup LSPs
- $\boldsymbol{R_{ij}}$: Residual bandwidth on link (i, j)

Using the above information, the worst case sharable bandwidth on a link (u, v) for protecting a primary LSP passing through link (i, j) is given by $S_{uv}^{ij} = \max(0, G_{uv} - F_{ij})$. Usually, a conservative approach is taken and it is assumed that all the bandwidth reserved on link (i, j) for primary LSPs is backed up on link (u, v). It follows that the remaining backup bandwidth on link (u, v) is sharable for any backup path which is protecting a primary LSP passing through link (i, j). All links (u, v) where $S_{uv}^{ij} + R_{uv}$ is greater than the new LSP bandwidth request can be used by the backup path protecting the primary LSP passing through link (i, j). Therefore, the use of sharable bandwidth improves the chances of successful placement of an LSP request.

III. CAIP: Per-class Aggregate Information with Preemption

In a DiffServ architecture, there is a traffic class associated with each LSP request. We propose the separate book keeping of bandwidth reserved for active and backup paths for every traffic class. Subsequently, the aggregate link usage information should be propagated on a per class basis. Therefore, for every link (i, j), the following information is propagated:

- $\boldsymbol{F_{ij}^c}$: Bandwidth reserved on (i, j) for primary LSPs belonging to class $c, \forall c$.
- $\boldsymbol{G_{ij}^c}$: Bandwidth reserved on (i, j) for backup LSPs belonging to class $c, \forall c$.
- $\boldsymbol{R_{ij}}$: Residual bandwidth on (i, j)

Since, the typical number of classes used in DiffServ is between 5-8 [2], propagating per class aggregate link usage information does not cause scalability problems. This information can be propagated through modifications in the proposed protocol extensions for DiffServ-TE [12].

The above scheme allows dynamic allocation of bandwidth for active and backup paths as well as dynamic allocation of bandwidth for use by different traffic classes.[1] More importantly, the proposed dissemination of information results in improved bandwidth sharing since more information is available while making the calculation of sharable bandwidth. The fine grained information about each traffic class can be used to calculate the sharable bandwidth on a per class basis. The sum of the sharable bandwidth of individual classes is at least as much as the sharable bandwidth calculated on an aggregate basis.

Recall from Section II that the sharable bandwidth for simple aggregate information scenario (SAIS) is:

$$S_{SAIS} = \max(0, G_{uv} - F_{ij}) \tag{1}$$

We define the sharable bandwidth on link (u, v) for protecting a primary LSP passing through link (i, j) in CAIP as:

$$S_{CAIP} = \sum_{c=1}^{n} \max(0, G_{uv}^c - F_{ij}^c) \tag{2}$$

$$\geq \max\left(0, \sum_{c=1}^{n} (G_{uv}^c - F_{ij}^c)\right) \tag{3}$$

$$= \max\left(0, \sum_{c=1}^{n} G_{uv}^c - \sum_{c=1}^{n} F_{ij}^c\right) \tag{4}$$

$$= \max(0, G_{uv} - F_{ij}) \tag{5}$$

Thus, $S_{CAIP} \geq S_{SAIS}$ with equality if and only if there does not exist some class c such that $G_{uv}^c - F_{ij}^c < 0$. That is, the bandwidth sharing in CAIP is at least as much as the bandwidth sharing in the simple aggregate information scenario. This improvement is further illustrated through the following example: Consider two flows traversing a link (i, j), and belonging to traffic classes c_1 and c_2, respectively. On link

[1] With minor modifications, this scheme can be adapted to cater to statically assigned bandwidth pools for different traffic classes.

(i, j), the active bandwidth reserved for c_1 and c_2 are 10 and 15 units respectively. On link (u, v), the backup bandwidth reserved for c_1 is 20 units while no backup bandwidth is reserved for c_2. For this scenario:

$$S_{SAIS} = \max(0, 20 - 25) = 0 \quad (6)$$

$$S_{CAIP} = \max(0, 20 - 10) + \max(0, 0 - 15) = 10 \quad (7)$$

The sharable bandwidth on link (u, v) in CAIP is 15, whereas the sharable bandwidth on link (u, v) in SAIS is 0.

Apart from bandwidth sharing, another advantage of CAIP is its ability to provide extra information in the process of preemption. In DiffServ aware MPLS-TE architecture, higher priority LSPs can preempt LSPs belonging to a lower priority class [13], [14]. In existing preemption schemes [10], [11], the preemption decision is taken by the node which needs to preempt bandwidth. Without any aggregate information about active and backup bandwidth, path calculation is independent of the preemption and does not take into account the preemptable bandwidth options on a given link. With CAIP, precise calculation of preemptable bandwidth can be made during path computation. Depending on the network service provider's criterion, backup and/or active bandwidth of lower priority traffic classes can be considered as the preeemptable bandwidth. This calculation can improve the chances of successful placement of higher priority LSPs by including links where adequate bandwidth can be made available through preemption. Note that the computation of preemptable bandwidth does not indicate which LSPs should be preempted. The actual preemption of LSPs require mechanisms such as those given in [10] and [11]. CAIP allows integration with existing preemption mechanisms given in [10] and [11] for the actual preemption of bandwidth.

IV. A Case Study

We now present a case study in order to elucidate the use of our proposed architecture and to show the calculation of preemptable bandwidth for user defined rules.

A. Problem Definition

We consider a network of n nodes and m unidirectional links. LSP requests arrive one by one at the ingress node, and the routing algorithm has no a priori knowledge of future requests. An LSP request is characterized by the LSP ingress node, LSP egress node, the associated bandwidth demand and the traffic class of the LSP request. The traffic class number starts from 1 and increases for higher priority classes. In order to serve an LSP request, a bandwidth guaranteed primary path must be setup along with locally restorable backup paths that protect against the failure of links along the primary path. If the routing algorithm is able to find sufficient bandwidth in the network for the requisite primary and backup paths, the paths are setup, and the LSP request is accepted; otherwise bandwidth preemption is considered for all links in the network. To this end, bandwidth reserved for backup paths belonging to lower traffic classes is considered as preemptable. Note that for this case study, we do not include bandwidth reserved for primary paths as the preeemptable bandwidth. However, depending on the network service provider's criterion, such bandwidth can be included in the calculations. Our goal is to provide backup paths for maximum number of LSP requests as long as the bandwidth reserved for these backup paths is not needed by higher priority request. Therefore, no higher priority request should be rejected because of provisioning of backup path to a lower priority request.

The preemptable bandwidth on any link (u, v) for a primary LSP belonging to class k and passing through link (i, j) is:

$$P_{uv} = \sum_{c=1}^{k-1} \min(G_{uv}^c, F_{ij}^c) \quad (8)$$

In other words, all the backup bandwidth belonging to lower priority classes which is non-sharable is considered as preemptable. The calculated preemtable bandwidth is now considered as available bandwidth for that link and another effort is made to calculate primary and backup paths. If sufficient bandwidth is available in the network, LSP request is accepted; otherwise the LSP request is rejected. Note that we only consider preemption if primary and backup paths are not available in the first place. While setting up the backup paths, exactly P_{uv} amount of bandwidth is preempted. To this end, we use the algorithm that minimizes the number of LSPs that are actually preempted.

B. Simulation Scenarios

For the simulation scenarios we consider three traffic classes: premium class, medium priority class and best effort class. The routing behavior received by a request depends on its traffic class and the applicable simulation scenario. We formulate three simulation scenarios making use of the preemption capabilities present in CAIP: Backup Paths with Preemption (BWP), Backup Paths without Preemption (BNP) and No Backup Path with No Preemption (NBNP). The names of the scenarios reflect the applicable backup provisioning and preemption rules for medium priority and best effort classes. We always provide a backup path to the premium priority request. Furthermore, this backup path cannot be preempted. In BWP, backup paths are also provided to requests belonging to medium priority class and best effort class. However, these backup paths can be preempted by requests belonging to a higher priority class. Therefore, a premium class request can preempt backup paths belonging to medium and best effort classes. Similarly, a medium class request can only preempt backup paths protecting best effort traffic while there is no option of preemption for best effort requests. In BNP, backup paths are also provided to medium priority and best effort requests and these paths are not preemptable. In the last scenario, NBNP, no backup path is provided to medium priority and best effort requests and hence there is no option for preemption. The only scenario with preemption, BWP, represents the benefit of preemption which is not available in the other two scenarios since they represent two extremes: In BNP, backup paths are provided but they cannot be preempted

later if a higher priority request arrives. On the other hand, in NBNP, in order to accept future higher priority requests no backup path is provided. Note that the preemption capability, as illustrated by BWP, highlights the benefits of CAIP.

C. Simulation Experiments

In this section, we describe the simulation experiments[2] that were used to compare the three simulation scenarios: BWP, BNP and NBNP. Moreover, we present results that depict the improvement in network efficiency due to improved bandwidth sharing in CAIP compared to SAIS. We conduct a set of experiments and compare the total number of rejected LSP requests and the total bandwidth placed under various scenarios. These statistics depict a realistic representation of a network service provider's goal of accepting maximum number of LSP requests, keeping in view the applicable rules for backup provisioning and bandwidth preemption. Moreover, it is important to consider the bandwidth associated with the accepted LSP requests since it is possible for a less efficient scheme to accept a large number of small bandwidth requests compared to a more efficient scheme which accepts fewer requests which comprise a greater amount of bandwidth. Therefore, in order to provide a better comparison, we present statistics for the amount of bandwidth placed on the network in addition to the number of rejected requests.

In our simulations, we use the local restoration scheme proposed by Kodialam et al. in [6]; the algorithms proposed therein for the computation of primary and backup paths are integrated with CAIP. The simulation experiments are conducted on a homogeneous network topology which is adapted from the network used in [9]. It represents the Delaunay triangulation for the twenty largest metros in continental Unites States [9]. All links in the network are uni-directional having a capacity of 12 units. Each node in the network may be an LSP ingress or egress. Therefore, there are 380 possible ingress-egress pairs in the network. LSP requests arrive one by one, and are characterized by an ingress, an egress, a traffic class, and the associated bandwidth request. The LSP ingress and egress nodes are chosen randomly from amongst all ingress-egress pairs. Similarly, the traffic class of the request is also randomly chosen. The bandwidth demand for an LSP request is uniformly distributed between 0.1 and 0.6 units, and the call holding time for each LSP request is infinite.

The computation of primary and backup routes for an LSP request depends on the simulation scenario. In case of BWP, for each LSP request, if it is possible to route the requisite primary and locally restorable backup paths without preemption, then the LSP request is immediately accepted; otherwise preemption is considered and another attempt is made to place the request, failure of which results in the rejection of LSP request. In case of BNP, for each LSP request, if it is possible to route the requisite primary and locally restorable backup paths, then LSP request is accepted and the associated bandwidth reservations are made on the network; otherwise the LSP request is rejected. In the last scenario, NBNP, both primary and locally restorable backup paths are provided for the premium class requests. On the other hand, for middle priority and best effort requests, only primary paths are provided. Using the above rules, we conducted 100 experiments with randomly selected ingress-egress pairs. We present the average of the total number of rejected LSPs and the total bandwidth associated with the accepted LSPs in these hundred experiments.

[2]The methodology to conduct simulation experiments of this paper is similar to the one used in our earlier work on restoration routing [15] [16].

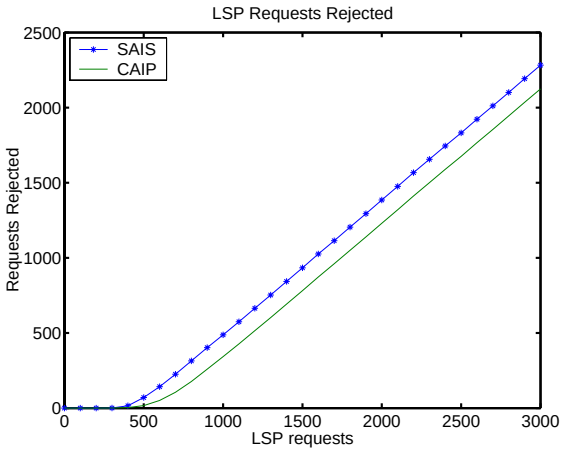


Fig. 1. Information Scenarios: LSP Requests Rejected

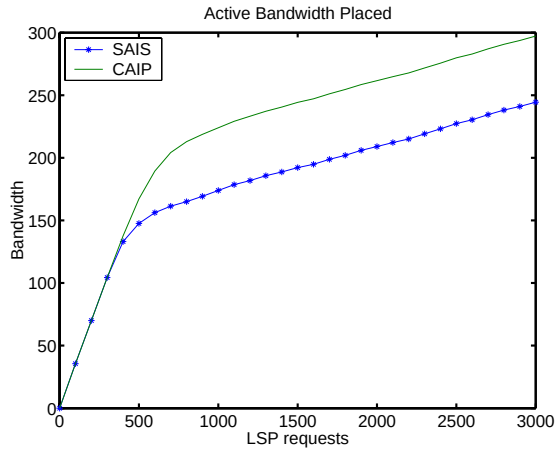
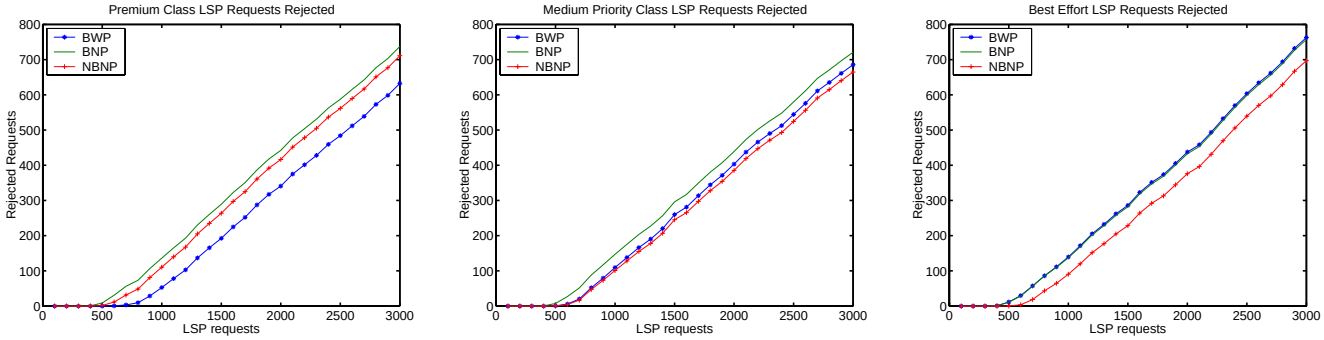


Fig. 2. Information Scenarios: Active Bandwidth Placed

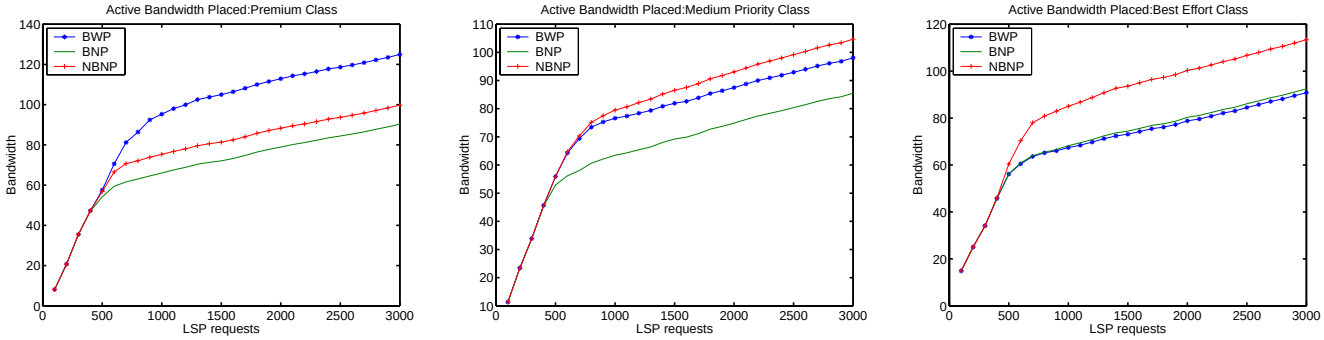
D. Simulation Results

Figure 1 shows the number of LSPs rejected in CAIP in comparison with SAIS. Efficient bandwidth sharing in CAIP results in better network utilization and hence fewer number of rejected LSP requests. Moreover, the bandwidth associated with the accepted LSP requests is also greater in CAIP compared to SAIS and is illustrated in Figure 2.On average, CAIP rejects 881 LSP requests compared to 1010 rejected LSP requests in SAIS. Similarly, CAIP is able to place 220 units



(a) (b) (c)

Fig. 3. Simulation Scenarios: LSP Requests Rejected



(a) (b) (c)

Fig. 4. Simulation Scenarios: Active Bandwidth Placed

of bandwidth compared to 180 units of bandwidth placed in SAIS, thus showing an average improvement of about 22%.

Figure 3 shows the number of LSPs rejected in the three simulation scenarios. When preemption is used as in BWP, there is a greater amount of bandwidth available for the premium class traffic and consequently fewer requests belonging to this class are rejected. Therefore, preemption of backup paths belonging to medium priority and best effort classes allows us to accommodate a greater number of premium class requests. Moreover, the bandwidth placed for premium class in BWP is also greater than the bandwidth placed for the same class in the other two scenarios. Also, note that NBNP rejects fewer premium class requests compared to BNP, since no backup paths are provided to medium priority and best effort requests. Therefore, unlike BNP where backup paths are provided to these classes, more bandwidth is available for use by premium class requests in NBNP. Figure 3(a) shows the number of requests rejected while Figure 4(a) represents the bandwidth placed on the network for premium class traffic in the three simulation scenarios. Similarly, Figure 3(b) shows the number of LSPs rejected in the three simulation scenarios for the medium priority class traffic.

In BWP, bandwidth belonging to the best effort class traffic is available for preemption by medium priority requests. In NBNP, medium priority requests do not receive a backup path and, therefore, greater number of such requests is accepted. For the medium priority class, the greatest number of rejected requests is under BNP because a backup path is required while preemption is not available. This is also supported by the amount of active bandwidth belonging to medium class traffic which is shown in Figure 4(b).

The number of LSPs rejected in the three simulation scenarios for the best effort class traffic is depicted in Figure 3(c). In NBNP, since no backup path is required for LSP requests, therefore, greater number of such requests is accepted. However, for the other two scenarios, backup paths are required. In BWP, no preemptable bandwidth is available since best effort class is the lowest priority class while in BNP there is no option of preemption. Therefore, as Figure 4(c) depicts, NBNP not only accepts a greater number of best effort requests but the associated bandwidth with these requests is also the greatest.

V. Conclusions

In this paper, we proposed a restoration routing architecture for DiffServ aware MPLS traffic engineering. This architecture, namely CAIP, provides prioritized guaranteed bandwidth along with fast restoration in the event of an element failure. The CAIP architecture is scalable and requires propagating only per-class aggregate link usage information; such information can be readily piggybacked on link state

routing packets using traffic engineering extensions to link state routing protocols.

CAIP allows precise computation of preemptable bandwidth for an arbitrary set of prioritization requirements put forth by the service providers. We presented a case study of service provider requirements and computation of preemptable bandwidth for those requirements. Towards this end, we considered three simulation scenarios, BWP, BNP and NBNP and compared their results for placing three different traffic classes. Each of these scenarios represents a particular preemption and backup provisioning option that can be used with our architecture.

CAIP can also be integrated with existing restoration routing schemes which are based on propagating aggregate link usage information. Furthermore, existing preemption schemes can be used with CAIP in order to decide the actual LSPs which are to be preempted.

ACKNOWLEDGEMENTS

This work was supported by a research grant from Cisco Systems, San Jose, CA.

REFERENCES

[1] D. Katz, K. Kompella, and D. Yeung, "RFC 3630: Traffic engineering (TE) extensions to OSPF version 2."

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "RFC 2475: An architecture for differentiated services," December 1998.

[3] E. Rosen, A. Viswanathan, and R. Callon, "RFC 3031: Multiprotocol label switching architecure," January 2001.

[4] F. L. Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen, "RFC 3270: Multi-protocol Label Switching (MPLS) support for differentiated services," May 2002.

[5] F. L. Faucheur and W. Lai, "RFC 3564: Requirements for support of differentiated services-aware MPLS traffic engineering," July 2003.

[6] M. S. Kodialam and T. V. Lakshman, "Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information," in *Proceedings of Infocom*, April 2001, pp. 376–385.

[7] ——, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in *Proceedings of Infocom*, March 2000, pp. 902–911.

[8] J. Marzo, E. Calle, C. Scoglio, and T. Anjali, "Adding QoS protection in order to enhance MPLS QoS routing," in *Proceedings of ICC*, May 2003, pp. 1973–1977.

[9] S. Norden, M. M. Buddhikot, M. Waldvogel, and S. Suri, "Routing bandwidth guaranteed paths with restoration in label switched networks," in *Proceedings of ICNP*, November 2001, pp. 71–79.

[10] J. D. Oliveira, C. Scoglio, I. Akyildiz, G. Uhl, and J. Smith, "A new topology-aware LSP preemption policy for DiffServ-MPLS networks," in *Proceedings of Networks 2002*, May 2002.

[11] J. D. Oliveira, C. Scoglio, I. Akyildiz, and G. Uhl, "A new preemption policy for DiffServ-aware traffic engineering to minimize rerouting," in *Proceedings of Infocom*, June 2002, pp. 695–704.

[12] F. L. Faucheur (Editor), "Internet Draft: Protocol extensions for support of differentiated-service-aware MPLS traffic engineering," March 2004.

[13] W. Lai (Editor) and D. McDysan (Editor), "RFC 3386: Network hierarchy and multilayer survivability," November 2002.

[14] J. D. Oliveira, J. P. Vasseur, L. Chen, and C. Scoglio, "Internet Draft: LSP preemption policies for MPLS traffic engineering," June 2004.

[15] F. Aslam, S. Raza, F. R. Dogar, I. U. Ahmad, and Z. A. Uzmi, "NPP: A facility based computation framework for restoration routing using aggregate link usage information," in *Proceedings of QoS-IP*, February 2005, pp. 150–163.

[16] S. Raza, F. Aslam, and Z. A. Uzmi, "Online routing of bandwidth guaranteed paths with local restoration using optimized aggregate usage information," in *Proceedings of ICC*, May 2005.