Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Computer Communications 27 (2004) 387–399

computer communications

www.elsevier.com/locate/comcom

Benefits of traffic engineering using QoS routing schemes and network controls

Shekhar Srivastava, Balaji Krithikaivasan, Cory Beard*, Deep Medhi, Appie van de Liefvoort, Wesam Alanqar, Ananth Nagarajan

Division of Computer Science and Electrical Engineering, School of Computing and Engineering, University of Missouri-Kansas City, 550G Flarsheim Hall, 5100 Rockhill Road, Kansas City, MO 64110, USA

Received 31 July 2002; revised 15 July 2003; accepted 6 August 2003

Abstract

We demonstrate the benefits of traffic engineering by studying three realistic network models derived from an actual service provider network. We evaluate traffic engineering in the presence of QoS-based routing schemes compared with Destination-Based Routing, the default routing behavior for the Internet. We also simulate prioritization of important traffic flows by implementing priority in one or more of the path caching, path ordering, and actual route selection phases of the constraint-based routing framework. We observe that traffic engineering can provide 20–50% network capacity savings. We also observe that prioritization in more than one phase of constraint-based routing can provide even more significant benefits.

Keywords: Traffic engineering; Constraint-based routing; Quality of service routing

1. Introduction

Traffic engineering is becoming an increasingly important consideration for managing network performance. Also, becoming crucial is the ability for network service providers to provide high quality services to particular sets of their customers. These customers expect to be able to use the network and be provided reliable services, regardless of the network conditions or loading. These customers might be emergency workers, those that depend on teleconferencing, those that use IP telephony, etc. In many of today's networks, resources are overprovisioned, so customers do not experience congestion and are satisfied. In certain types of today's networks, however, such as regional or access networks, overprovisioning is not adequate to meet the demands that might occur. And it is difficult to predict if many of today's networks will continue to be overprovisioned in the future if network capacity upgrades do not keep pace with load growth.

The objectives of traffic engineering are twofold: to alleviate hot spots in the network by load balancing and to provide dependable service to certain classes of traffic. Traffic engineering objectives are achieved differently in packet based networks as compared to flow based networks. In packet based networks, load balancing can be achieved by carefully adjusting link weights as shown recently in Refs. [1,2]. Such mechanisms can be complemented with various Quality of Service (QoS)-based architectures like differentiated services to provide acceptable grade of service to particular classes on an aggregate basis.

However, if more fine grained control is required, such mechanisms are limited. For example, consider what might happen if one wished to provide reliable QoS to particular voice communication sessions over the Internet by emergency workers after a disaster. In such cases, networks would likely be heavily overloaded and all traffic would receive poor performance, regardless of how well link weights were defined. If, however, a flow-based approach were used, particular flows could be given special treatment to find the resources they needed and keep other flow-based or best effort traffic from using those resources. Such flows would reserve network resources using a mechanism such as

* Corresponding author. Tel.: +1-816-235-1550; fax: +1-816-235-1260.
E-mail address: beardc@umkc.edu (C. Beard).

0140-3664/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2003.08.003

integrated services or RSVP, then would be transported using Multi-Protocol Label Switching (MPLS), for example.

For flow based networks, both of the objectives of the traffic engineering can be accomplished in a unified manner. MPLS provides such an integrated framework. An important feature of MPLS is the ability to set up label switched paths for different services to reserve bandwidth, if and when needed. Furthermore, the possibility of doing constraint based routing in general, and for specific services if needed, is another attractive feature. The IETF literature (both RFCs and Internet drafts) has been deluged recently with various aspects and capabilities of MPLS [3], traffic engineering [4], and the use of various features to allow deployment of controls and architectures such as virtual private networks [5]. While most of these works describe the benefits in a qualitative manner (sometimes, from the point of view of 'good features' or 'best practices'), very few discuss the actual quantitative benefit. For example, there is a school of thought that believes that MPLS is not necessary; the current best-effort routing (alternatively called Destination-Based Routing (DBR)) is good enough *if* enough bandwidth is available in the network. Further, there is very limited work that discusses whether different controls that can be deployed in an MPLS environment for traffic engineering are actually beneficial from a network performance standpoint.

Given this debate, we have set out to study these tradeoffs. Our approach is simple. We consider three realistic traffic models with given topology where multiple services are provided and where we have a good idea of the traffic mix. Then we pose the following questions.

(1) What service levels could different classes of traffic receive depending on a combination of routing possibilities and various network controls that may be placed for traffic engineering?
(2) Would use of such mechanisms provide significant cost savings in terms of reduced network capacity requirements?
(3) Are conclusions dependent on the specific network being considered? Can general conclusions be made?
(4) What combinations of mechanisms seem to provide the best overall performance?

Our work is a case study of three network models to address these questions. Our team, consisting of members from both academia and industry, have worked together closely to create realistic simulation models derived from actual networks. Through this case study, we attempt to gain some insight into the tradeoffs. By no means do we address all the possible issues regarding deployment or capabilities of MPLS, such as signalling exchanges. Instead, we focus on the benefits that traffic engineering could provide.

We study the performance of different services in this example network, and work from the assumption that some classes of traffic require better grade-of-service (GoS) than others (i.e. lower blocking probabilities). We choose to perform the study from the viewpoint of blocking probabilities, but do not assume that the network would implement per flow state in both control and forwarding planes in a network. The scalability problems of such an approach are well documented. This work does, however, provide a basis from which performance can be assessed at the flow level, which is a very useful construct from which to understand network behavior and traffic engineering. By performing simulation at the flow level (instead of at the packet level), it is also possible to simulate the performance of thousands of flows in an efficient manner. Actual admittance of flows into the network could be implemented using endpoint admission control, network state in the control plane but not in the forwarding plane, RSVP, MPLS with resource reservations, aggregation mechanisms, etc. The goal of this work is not to assess the viability of those options, but rather to see the benefits of traffic engineering in general.

It is commonly known that in multi-service networks those classes which have higher bandwidth requirements per flow have higher blocking, because it is harder to find enough resources when flows require large bandwidth. We seek to eliminate that problem as well as give better GoS to those classes which need it the most. In the model studied here, we have four service classes and service class 1 (S1) is given priority because it is assumed that S1 traffic generates higher revenue and customers have high expectation for bandwidth to be available when needed.

As one important example, the mechanisms studied here could be applied to prioritize disaster management traffic in response to natural or man-made emergencies [6,7]. Recent terrorist events in the United States on September 11, 2001, have shown that telecommunication networks provide tremendous value to society in response to disasters. These events have also shown what is common with disaster response, however, that tremendous stress is placed on these networks from high loads and damaged facilities [8–10]. When resources are scarce, those users and user applications that are of higher value or importance should be given greater access to resources. During exceptional conditions like emergencies or disasters, users and user applications of greatest importance would be those which relieve danger to life and property. A mechanism for prioritizing user traffic would therefore be valuable in both normal and exceptional operating contexts.

Of particular interest are the performance improvements that can be realized by using traffic engineering. One type of performance comparison would be the amount of capacity needed to provide certain GoS levels whether traffic engineering (i.e. implementing mechanisms beyond OSPF and BGP) were used or not. A particular GoS level can always be provided given enough system capacity, so the key question is *how much* capacity is needed. We compare a network without traffic engineering to those which

implement various traffic engineering mechanisms and compare capacity requirements. We show for the models studied here that a network that does not use traffic engineering could need on the order of 20–50% more capacity to meet the same GoS requirements.

Another type of performance comparison is the relative benefit of using different types of traffic engineering mechanisms, especially when some traffic classes require much lower blocking than others. We use the framework presented in Ref. [11] and consider QoS routing to be a subset of constraint-based routing [12]. QoS routing only deals with the dynamic load conditions of the network whereas constraint-based routing also considers constraints imposed by network operators. We divide the constraint-based routing [12] into three phases. Therefore, we consider the first two phases as QoS routing and the third phase as Network Controls. It was concluded there that QoS routing need to be complemented by network controls to ensure maximum benefit. Moreover, authors in Ref. [13] provided results for various kinds of mechanisms for the network control phase of the routing framework from a case study using a single network model. The mechanisms studied include Trunk Reservation (TR), Service Class based Trunk Reservation (SCTR), and Service Class based Multi-Link Trunk Reservation (SMTR). It was observed that these mechanisms can provide significant benefit to high-priority traffic classes when applied to the network control phase of constraint-based routing.

In contrast to Ref. [13], where priority was only implemented in the network control phase of the routing process, this work shows how the implementation of priority in all three phases can provide even better performance. In other words, in addition to using variants of TR to provide prioritized grade of service, priority is also deployed by using preferences in the numbers of cached paths and the routing mechanisms that are used (e.g. Dynamic Random Routing (DRR), Maximum Available Capacity without Crankback, etc.). All three phases are combined into a unified framework that effectively provides priority through the three phases. To the best of our knowledge, no work exists which presents such a unified routing framework to provide effective prioritized service.

We also evaluate the benefits for three additional network models beyond the one previously investigated, to demonstrate that these results have general application beyond just a few specific network models. The approach we use is to study the impact of implementing priority in any one phase, and then investigate a series of combinations where priority is applied in two or three phases. While not claiming to have found optimal solutions, we show that some combinations can be identified that provide substantial performance improvements. So in summary, this work extends [11,13] by considering three additional network models, but also breaks new ground by providing a unified approach to prioritized network resource allocation using all three phases of the routing framework.

The rest of the paper is organized as follows. First, we review the three-phase framework for constraint-based routing and discuss mechanisms for providing priority service to particular traffic classes. Next, the different scenarios considered in our study are introduced, such as DBR and different QoS routing schemes (with path caching). In Section 3, we discuss the network topology and data, and the performance metrics. The results are presented in Section 4 by considering a normally loaded network model and viewing results with respect to blocking performance and network capacity requirements. Then the other two models are investigated and differences in results are discussed. We close with a summary of our observations in Section 5.

2. Priority mechanisms

A network that provides differentiated or guaranteed QoS needs to provide differing services and performance to different service classes. Such a requirement calls for more sophistication in the network as compared to simply finding paths with the fewest numbers of hops. Assuming that all links block independently, using these hop-count metrics will give priority to flows which traverse fewer links. As a flow requests a longer and longer path, the chance of getting accepted decreases, whereas for a QoS network higher priority needs to come from the class of the service. A flow of lower priority having a one hop path should have lower chances of getting accepted as compared to a flow of higher priority having a multi-link path. Hence, not only does a QoS-aware network need to recognize the class of a flow, it also needs to safeguard resources for a high priority class.

Acceptance or rejection of a flow belonging to a particular service class depends on the resources offered by the network during three phases based on the constraint based routing framework discussed in Refs. [11,14,15]. In the first phase, a set of shortest paths are computed based on simple hop-count and cached [16] for each service class by every source to all possible destinations on the network; this is referred to as the Preliminary Path Caching (PPC) phase. The network caches as many paths as is deemed suitable for a specific service class. In the second phase, the cached paths are ordered from most acceptable to least acceptable path (e.g. in terms of residual bandwidth) using a specific routing scheme [17]; this is called the Updated Path Ordering (UPO) phase. The network orders the paths depending upon the class of service and criteria for finding the best path through the network. In the third phase, a specific route is selected from the ordered paths to try to accommodate a newly arrived flow and is named the Actual Route Selection (ARS) phase. The specific route chosen depends on the class of the flow and resources that are reserved or restricted for availability for that class using TR or one of its variants. As mentioned

earlier, QoS routing forms the first two phases, namely PPC and UPO, of the constraint based routing framework. In other words, QoS routing along with the ARS phase is referred to as a constraint based routing framework.

We see that priority as seen by a service class is affected by the network's choice in the above mentioned three phases. The resources dedicated by the network for a specific service class in the PPC phase changes the number of routes available for a specific service class for ordering and hence the reachable links. Similarly for the other phases, performance of a service class depends on the choices made by the network. Thus, priority can effectively be provided to a service class in a network by choosing a point in a three-dimensional space spanned by the following three dimensions, namely number of cached paths (PPC phase), type of routing scheme (UPO phase) and degree of control (ARS phase). The choice made by the network for the three phases for a specific service class translates into performance as seen by the service class. These three dimensions can be seen as characterizing the overall prioritization mechanism being applied to a service class. More elaboration on the use of prioritization for each phase is as follows.

2.1. Priority in the PPC phase (number of cached paths)

Path Caching has been shown to have considerable impact on the performance of a network as shown in Ref. [18]. It has been shown that more paths help in load balancing and overall network performance. However, it has also been observed that an excessive number of stored cached paths leads to an overloaded network with inferior performance, because it adds longer paths to the cache. These longer paths use more network resources. We use the priority for a service class to determine the number of cached paths stored for that particular service class.

2.2. Priority in the UPO phase (choice of routing schemes)

The UPO phase uses routing schemes that attempt to find a path that satisfies one or more constraints of interest and that is optimal with respect to some scalar metric. The constraints can include residual bandwidth, delay, jitter, administrative policies, etc. While a feasible path can be selected using a simple hop-count based algorithm, additional constraints can be considered to improve the resource utilization by doing some measure of load balancing. The routing schemes that we used in this work use residual bandwidth or in other words, available bandwidth as an additional constraint. The residual bandwidth of a path is defined as the minimum amount bandwidth available on any of the links in the path.

Priority in the UPO phase is implemented by using different routing schemes for different service classes, for example allowing crankback for a priority class but not for other classes. The various routing schemes considered are as follows.

- Dynamic Random Routing (DRR)—This scheme is a simple and an efficient routing scheme based on Dynamic Alternate Routing [19,20]. It is also referred to as Cached Sticky Random Adaptive Routing (CaSRAR) [11]. There is no regular UPO phase. For the ARS phase, it maintains a direct path (if one exists) and a preferred alternate path. The flow tries the direct path first and if there is not enough bandwidth on it, the alternate path is tried next. When a flow gets blocked on the preferred alternate path, it is blocked and cleared. For future flows, it randomly selects a new alternate path from the cached paths.
- Maximum Available Capacity Routing with Periodic updates and No Crankback (MACRPNC)—The paths are sorted from most available bandwidth to least available bandwidth. This is done periodically with the period being the routing update interval. The direct path is chosen as the first option irrespective of the bandwidth availability on it. If blocked on the direct path, a second choice from the set of cached paths with most available capacity is made. If that path is also blocked, then the flow is blocked.
- Maximum Available Capacity Routing with Periodic updates and Crankback (MACRPC)—This is the same as MACRPNC, except that after a path is blocked the source can crankback and keep trying paths in the cache. This routing scheme has a configurable parameter for the number of crankbacks that are allowed.
- Maximum Available Capacity Routing with Instantaneous Computation (MACRIC)—This routing scheme operates the same as MACRPC except that when a new flow arrives, the entire network is freshly scouted to find a path with the most available bandwidth (hence, the term instantaneous computation). This process is repeated for every new flow arriving into the network. This routing scheme is utopian because it is impractical to provide completely updated state information for each newly arriving flow. This mechanism is of theoretical interest since it serves as a benchmark.
- Destination Based Routing (DBR)—This replicates the default routing in today's Internet which does not implement priority but can be used in comparison with the other routing schemes. Because the Internet does routing based on destination, no alternate routing is used. The way DBR is implemented here is to find a path with the shortest path based on hop-count and either that path has free resources or the flow is blocked.

By using time varying constraints, such as available bandwidth, frequent updates on the status of the links are necessary. One of the major factors affecting the performance is the periodicity of these updates. If the period between successive updates is too long, the bandwidth availability information is no longer valid. So, the flows can get blocked even if the best path (according to the last update) is chosen. However, too frequent updates increase

the network overhead and might introduce oscillations in the network. We do not address this issue in depth in this work. For further details, see Ref. [18].

2.3. Priority in the ARS phase (activation of control)

A variety of controls can be implemented to limit access to network resources for certain types of flows. These seek to provide better GoS to higher priority service classes as follows.

2.3.1. No Control

No Control (NC) is the absence of any control in the ARS phase. Here, if the routing scheme finds a path with bandwidth sufficient for the flow, the flow is accepted, otherwise it is blocked.

2.3.2. Trunk reservation

TR [21] is a simple call admission control scheme that favors direct traffic (i.e. one hop paths). When the available bandwidth on a link falls below a particular threshold value, alternately routed flows will be blocked even if there is enough bandwidth to accommodate them. In essence, once the threshold value is reached, only the direct traffic has access to the link. This prevents a problem where alternate routing uses more total bandwidth (i.e. the sum of bandwidths on all links on a path) than direct routing and can ultimately decrease the efficiency of bandwidth usage. A TR threshold is set as a percentage of link bandwidth.

2.3.3. Service Class based Trunk Reservation

In the previous case, all flows belonging to direct traffic are treated equally in accessing the TR area. In the SCTR approach, only flows belonging to a GoS stringent service class have access to the link once the threshold is reached. This means the non-GoS stringent class flows will be blocked similar to alternately routed flows, even if they are direct flows.

2.3.4. Service Class based Multi-Link Trunk Reservation

SMTR extends the SCTR approach by allowing both direct and overflow traffic of GoS stringent service classes to access the TR area. This means alternately routed flows of a GoS stringent class still can access a link in addition to the direct flows of a GoS stringent class when the available bandwidth falls below the threshold.

3. Simulation environment and network setup

To conduct our study, we have used Multi-Service Dynamic Routing Simulator (MuSDyR) [22]. There is no packet level detail in this simulator which allows us to simulate thousands of simultaneous flows in an efficient manner. This allowed simulation times to be sufficiently long to produce low variance in the results over multiple simulation runs with carefully chosen seeds. The flows are assumed to follow Poisson arrival processes with exponential holding times, their mean rates depending on the traffic classes.

3.1. Network topology

The primary network, which we call Network I comprises of 15 nodes (labeled A–O) connected by 58 links. This network was derived from an actual service provider network. Due to the space constraints and the nature of the topology, we do not provide a graphical picture of the topology. Instead, we have enumerated the links in Table 1. The second network we study, which we call Network II, is a reduced capacity version derived from Network I. Network III, a more densely connected network was created by removing three nodes and 21 links from Network. The links we eliminated from Network are the ones which do not carry direct traffic and the eliminated nodes do not generate any traffic. This created three networks to be studied, all asymmetric to varying degrees and with varying load levels. The motivation for using three networks was to determine the general applicability of our approach.

3.2. Traffic models

The simulator provides many service models from which service classes can be constructed. All the models require specification of the Erlang load to be generated and the flow duration. The models differ in the parameters that characterize their bandwidth requirement. Some of the implemented models include Fixed Rate (FR), Uniform Fixed Rate (UFR) and Variable Rate (VR) On–Off model which can be understood as:

- FR model: In this model, the bandwidth requirement of each flow is equal to the provided bandwidth (input parameter).

Table 1
Network topology model

Link	Cap (Mbps)	Link	Cap (Mbps)	Link	Cap (Mbps)	Link	Cap (Mbps)
A–B	933	A–C	933	A–E	1866	A–G	1866
A–H	1866	A–I	933	A–J	933	A–K	1866
A–M	1866	A–O	1866	B–C	1866	B–F	1866
B–G	933	B–H	1866	B–I	1866	B–L	933
C–E	933	C–F	933	C–G	2799	C–I	3732
C–J	1866	C–K	933	C–M	933	C–N	2799
C–O	933	D–E	933	D–H	1866	D–I	933
D–J	933	D–M	933	E–G	2799	E–I	2799
E–K	1866	E–N	2799	F–G	933	F–H	1866
F–J	933	G–H	933	G–K	933	G–M	2799
G–N	933	H–I	1866	H–J	933	H–K	933
H–L	1866	I–K	1866	I–M	3732	I–N	933
J–K	933	J–L	1866	J–O	933	K–M	933
K–N	2799	L–M	1866	L–O	1866	M–N	3732
M–O	933	N–O	1866				

- UFR model: In this model, the effective bandwidth requirement of each flow is sampled uniformly in the interval $(L \times \mathrm{BW}, U \times \mathrm{BW})$ where BW is the provided bandwidth (input parameter) and $0 \leq L < U \leq 1$.
- VR model (VR): In this model, the effective bandwidth requirement of each flow is computed from five input parameters, namely the Sustained Flow Rate (SFR), the Peak Flow Rate (PFR), the Mean Active Burst Period (A), the Flow Loss Ratio (FLR) and the Buffer Size (B) based on the fluid-flow model given in Ref. [23].

3.3. Service classes

The traffic for the network model studied here is comprised of four service classes each having different loads between each pair of nodes for a given class. Each service class has its own routing table and makes its decision based on the status of the paths and the GoS requirement. The four service classes, namely S1, S2, S3 and S4, are explained below:

- Service Class 1 (S1): The S1 Service class is considered to be the high priority traffic class. It is constructed using the UFR traffic model. The S1 service class multiplexes multiple sources with different but fixed bandwidth requirements between a nodepair. The service class derives its characteristics from the behavior of the sources. For a nodepair (i,j), let there be N_{ij} sources multiplexed to the S1 service class having bandwidths bw_{ij}^k. Then the maximum bandwidth of the S1 service class between the nodepair (i,j) is $\mathrm{BW}_{ij} = \sum_{k=1}^{N_{ij}} \mathrm{bw}_{ij}^k$. The Erlang load of the flows for the S1 service class between the nodepair (i,j) is $\rho_{ij} = N_{ij}$. Since the bandwidth of the service class is derived by averaging over flows, the bandwidth of an arriving flow is determined from a uniformly distributed sample between 75 and 100% of BW_{ij}. The sampling is done every time a new flow arrives for a nodepair. The average flow duration is 300 s.
- Service Class 2 (S2): The S2 service class is implemented to have a small guaranteed bit rate, plus unspecified requirements above that. S2 is constructed from the FR traffic model. It comes as a request with a minimum bit rate which is allocated to the connection and a VR part, for which there are no guarantees and, hence, no reservation of resources. Every active nodepair generates S2 traffic with an FR part that has an inter-arrival time of 10 s and flow duration of 180 s.
- Service Class 3 (S3): The S3 service class is constructed from the VR traffic model. For a nodepair (i,j), let there be N_{ij} sources multiplexed to the S3 service class having PFRs pfr_{ij}^k and SFRs sfr_{ij}^k. Then the peak and SFRs for the S3 service class between the nodepair (i,j) are $\mathrm{SFR}_{ij} = \sum_{k=1}^{N_{ij}} \mathrm{sfr}_{ij}^k$ and $\mathrm{PFR}_{ij} = \sum_{k=1}^{N_{ij}} \mathrm{pfr}_{ij}^k$. Other parameters were chosen as Active Burst Length of 1 s, Buffer Size of PFR_{ij} and Cell Loss Ratio of 0.1%. The Erlang load of the flows for the S1 service class between the nodepair (i,j) is $\rho_{ij} = N_{ij}$. The same bandwidth is used for all the flows for a nodepair. The average flow duration is 600 s.
- Service Class 4 (S4): The S4 service class is implemented the same as the S3 service class with Cell Loss Ratio as 0.01% and flow duration of 180 s.

We have discussed the approach used to compute the Erlang load and bandwidth of a flow. Now, we define 'effective load' of a flow between nodepair (i,j) of service class s as $\rho_{ij}^s \times \mathrm{BW}_{ij}^s$ and that of service class s is computed as $\sum_{(i,j)} \rho_{ij}^s \times \mathrm{BW}_{ij}^s$, where $s = \mathrm{S1}, \mathrm{S2}, \mathrm{S3}, \mathrm{S4}$.

3.4. Traffic matrix

The traffic profile consists of 37 source–destination pairs, all of which have direct links between them. This is significant because no traffic needs to use multi-link paths, except when alternate routing is required when the direct link is full. It is also significant because in the topology there are 58 links, so 21 links have no direct traffic specified and are idle unless they are used for alternate routing.

Due to the voluminous nature of the traffic matrix, we only give an idea of the matrix rather than the complete matrix itself. Among the 37 active nodepairs, 36 of them have S1 and S3 traffic, six of them have S4 traffic and all of them have S2 traffic. The Erlang load and the bandwidth requirement of a traffic class across all the nodepairs are not necessarily the same. The ratio of the overall effective load of the traffic classes is given by 4.69 (S1) : 4.58 (S2) : 90.71 (S3) : 0.02 (S4), as derived from an actual service provider network. We would like to add that the traffic is not representative of the network performance at all times. We believe that the most important aspect of the traffic is the distribution of effective load throughout the topology, not necessarily the actual loading levels of the network as a whole. The load is distributed asymmetrically which is always true of real-world networks.

We refer to the traffic matrix that we discussed above as 5% S1. From this traffic matrix, we have generated three other traffic matrices namely 10% S1, 15% S1 and 20% S1 with increasing fraction of S1 effective load to the overall network effective load. Note that the major portion of network traffic comes from S3 flows. Since we have considered traffic class S1 as our high priority class, these traffic matrix variations allow us to see how various mechanisms perform when S1 becomes a larger fraction of the overall network effective load. However, in order to make sure that over all network effective load is kept fixed, we move the effective load appropriately from traffic class S3 to S1 in the case of 10% S1, 15% S1 and 20% S1. This approach is aimed at bringing out the interplay between the fraction of high priority traffic

(to the overall traffic) and the effectiveness of the constraint based routing mechanisms.

Let p denote the fraction of S1 effective load (i.e. $p = 5$ in the case of 5% S1 traffic matrix). Let ρ and bw represent the Erlang load and the bandwidth, respectively, with subscripts denoting the service class (S1, S3, etc.) for the 5% S1 traffic matrix. In order to obtain the traffic matrices 10% S1, 15% S1 and 20% S1, the Erlang load of the S1 and S3 traffic classes for each nodepair are generated from the corresponding nodepairs in the 5% S1 traffic matrix using the following method. Note that ρ^{new} represents the Erlang load obtained for the new traffic matrix.

$$\alpha = \frac{p}{5} - 1 \tag{1a}$$

$$\rho_{s1}^{\text{new}} = \rho_{s1}(1 + \alpha) \tag{1b}$$

$$\rho_{s3}^{\text{new}} = \frac{\rho_{s3}\text{bw}_{s3} - \alpha\rho_{s1}\text{bw}_{s1}}{\text{bw}_{s3}} \tag{1c}$$

3.5. Performance metric

The primary performance parameter we seek to observe is the Bandwidth Denial Ratio (BDR) which considers the overall ratio of total admitted bandwidth to total requested bandwidth. It seeks to capture the effect of having flows of widely differing bandwidths all attempting to access the network. This is in contrast to naive blocking (the classical call blocking) which is an overly simplistic view of blocking performance in a multi-class environment. Both of these are discussed as follows.

In order to describe the relevance of the BDR, we assume a network with M classes of service. Let, B_i be the blocking encountered, λ_i the arrival rate, and bw_i the bandwidth of flows for the ith class. Here, B_i is defined as:

$$B_i = \Pr(\text{Arrival is blocked from Class } i) \tag{2}$$

For such a network, various performance measures can be explained as:

- Naive Blocking (NB) stands for the traditional blocking definition which gives the average blocking level across all classes, computed as follows.

$$\text{NB} = \frac{\sum_{\forall i} B_i}{M} \tag{3}$$

  By averaging directly over the classes, NB loses the information about the composition of blocked calls. It is still meaningful as precisely the probability that a call gets blocked if there is a single class ($M = 1$) or in a case where there are multiple classes and they have the same arrival rates (λ_is are equal). Since it does not account for differences in λ_is and bw_is, the utility of the metric is fairly restricted in a multi-service, multi-class network.

- Weighted Blocking (WB) reflects the fraction of requests that have been blocked irrespective of the bandwidth of the connection. This weights the overall blocking computation by the relative amounts of Erlang load generated by each class.

$$\text{WB} = \frac{\sum_{\forall i} \lambda_i B_i}{\sum_{\forall i} \lambda_i} \tag{4}$$

  This metric is the conditional probability of a flow getting blocked given a flow of class i arrives at the network. The metric weights the blocking encountered by a flow to the fraction of calls of that flow. Hence, it gives a good measure of the probability of a randomly chosen call getting blocked if all bw_is are equal. Since it does not account for the bandwidth information of the lost flow, it looses its credibility in the case of multi-service networks with each class having a different bandwidth requirement. Since we know that blocking encountered by a class also depends on the bandwidth of the flows, WB has restricted utility in such a multi-service environment, except that it may make sense from user perspective.

- BDR reflects the fraction of bandwidth units that are blocked by the network. BDR is computed as follows:

$$\text{BDR} = \frac{\sum_{\forall i} \lambda_i B_i \text{bw}_i}{\sum_{\forall i} \lambda_i \text{bw}_i} \tag{5}$$

  Another way of seeing the metric can be as follows, let N be the set of flows that arrive to a network and let the bandwidth requirement of flow j be given by w_j. Let N_{D} be the set of flows which were denied service to the network. BDR is then given by

$$\text{BDR} = \frac{\sum_{j \subset N_{\text{D}}} w_j}{\sum_{j \subset N} w_j} \tag{6}$$

  BDR gives the probability of a unit of bandwidth being denied conditioned on being requested. BDR not only accounts for the relative Erlang load of a class, but also captures the differing bandwidth requirement of classes. The metric makes sense from a service provider perspective since it tells the fraction of bandwidth demand that was rejected by the network. It can be seen in direct relationship with the effective services provided to the users assuming they (services) were requested. In other words, a flow blocked with higher bandwidth is a bigger loss than one of lower bandwidth getting lost. BDR effectively captures the different losses incurred in blocking of different service classes.

3.6. Notation

The following notation is used.

BDR_{s1}	BDR of only the S1 service class across the network
BDR_{net}	BDR of all service classes across the network
K_{s1}	number of cached paths for the S1 service class
K_{osvc}	number of cached paths for all other service classes
RS_{s1}	routing scheme used for the S1 service class
RS_{osvc}	routing scheme used for other service classes

3.7. Experiment setup

The Experiments were devised to provide priority to flows of the S1 service class by varying the control parameters in PPC, UPO and ARS phases according to the following guidelines.

- For the PPC phase: K_{s1} as 4, 6 and 8 and K_{osvc} varying from 4 to K_{s1}. In general, priority is given to the S1 service class by giving more cached paths to choose from.
- For the UPO phase: RS_{s1} and RS_{osvc} chosen from DRR, MACRIC, MACRPC and MACRPNC.
- For the ARS phase: changes from NC, TR, SCTR and SMTR.

The setup allows us to study the behavior of the overall network and S1 service class under possible variations which can be exercised to provide desirable performance to the S1 service class. Of particular importance is the impact on the overall network performance by implementing mechanisms to improve the GoS of the S1 service class. To test the robustness of the results, we have constructed four traffic matrices as explained in Section 3.4. We chose to move the load from S3 class (since it is the predominant class) to the S1 service class. Hence the following terms should be understood in this perspective:

- 5% S1—S1:S3::4.69:90.71 ($p = 5$)
- 10% S1—S1:S3::9.21:86.28 ($p = 10$)
- 15% S1—S1:S3::13.51:82.02 ($p = 15$)
- 20% S1—S1:S3::17.84:77.74 ($p = 20$)

We then apply these to three network models. For Network I, all of the control schemes employed were experimented with $p/4$ and $p/2$ reservation for the S1 service class where p is the fraction of traffic for that traffic class.

4. Results and discussion

We ran extensive simulations exploring the possible options for the PPC, UPO and ARS phases and present results for interesting, significant and insightful cases only. We view the results from the variation in K_{s1} and K_{osvc} and from the perspective of varying capacity to evaluate the benefits and cost of various options for PPC, UPO and ARS phases for the S1 and other service classes.

In general, the load on the network would be characterized as normal loading. We have scaled the network capacity to make a network level BDR of approximately 0.10. This would correspond to a case where a network was performing under stress and is in need of performance improvement, either by adding new capacity to the network or implementing additional controls. The goal of this work is to see how those controls would perform and whether they would obviate the need for new capacity.

We first simulated this network without any path caching, routing schemes, or network controls using DBR. This performance is what we would expect with today's Internet without any constraint-based routing, MPLS, etc. These results can be used as a base case against which other routing mechanisms can be compared. In the case of 5% S1, blocking for DBR for the S1 class is 0.056 and blocking for the overall network is 0.177. In the case of 20% S1, blocking for S1 is 0.057 and blocking for the overall network is 0.161.

Note that in this paper we have used hop-count as the metric for DBR. Recently, determining optimal link weights for networks with DBR (in particular, OSPF networks [1,2]) has received significant attention. Our interest here has been to see if the results would be significantly different if we were to use optimal link weights. Recall that our network is a loss-network model; thus, for the given demand, no feasible flow exists that can carry *all* the traffic through the network. Thus, first we consider a feasible network for this purpose by doubling the network bandwidth. Secondly, we have aggregated the traffic for multiple services into one traffic demand volume (per demand pair) by summing the product of load and the bandwidth per session for each service class. We have then determined the optimal link weight by using the duality-based approach described in Ref. [1]. We have found that for the topology and demand scenario we have considered (Network I), the optimal link weights for all links remain at 1 (i.e. hop-count metric) except for two links. In other words, for all practical purposes, the optimal weight is essentially equivalent to the hop-count metric. Thus, we can infer that simulation results would have minimal difference with optimal link weights compared to hop-count metrics.

4.1. Basic results

Now in examining constraint based routing, first we check our assertions from the previous sections and observe the behavior of the network by changing only a single mechanism, either the routing scheme or number of cached paths. For a case where all classes use the DRR routing

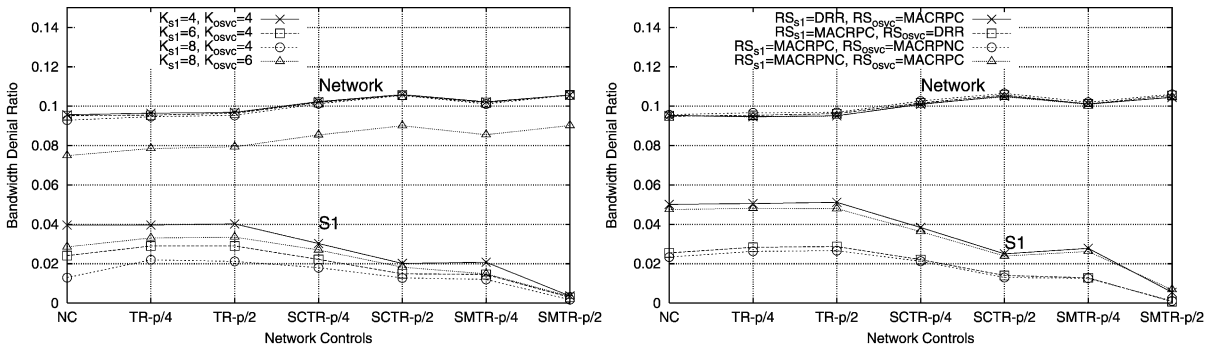

Fig. 1. BDR of S1 service class and overall network for 5% S1 with (left) $RS_{s1} = RS_{osvc} = DRR$ and (right) $K_{s1} = K_{osvc} = 4$ with changing network controls.

scheme, we present results in Fig. 1 on the left plot that show the BDR of the S1 service class (on the lower set of curves) and the overall network (on the upper set of curves) for 5% S1. Each curve shows how the BDR changes for different network controls specified along the x-axis from NC to SMTR with a reserved area of 2.5% of the capacity. The proportion of S1 traffic is $p = 5\%$, and the TR, SCTR, and SMTR controls create reserved areas with proportions $p/4$ and $p/2$ of link capacity. Various curves are produced for different combinations of K_{s1} and K_{osvc}.

Note first of all that all of the curves, even with no controls (NC) and only DRR is being used, demonstrate performance much better than DBR where blocking was 0.056 for S1 and 0.177 for the overall network. We also observe significantly different results when we change the balance between K_{s1} and K_{osvc}. The worst performance for S1 occurs when $K_{s1} = K_{osvc} = 4$ (i.e. no priority in numbers of cached paths). The BDR for the S1 service class improves significantly when the S1 class is allowed to use more cached paths than the other classes. For example consider the case using NC. Blocking can be half as much for S1 when $K_{s1} = 8$ and $K_{osvc} = 4$ than when $K_{s1} = K_{osvc} = 4$.

We also can significantly reduce S1 blocking by implementing TR, SCTR, and SMTR controls. In all cases, however, tradeoffs must be made in the network-level BDR; some mechanisms may cause a significant rise in the network BDR. Once a suitable control is chosen, this would constitute an implementation of priority in two phases.

In the right plot of Fig. 1, we compare routing schemes while keeping $K_{s1} = K_{osvc} = 4$. We get significantly different results by varying the routing schemes between the S1 service class and the other service classes. Note also that for some combinations of routing schemes, but not all of them, blocking for S1 can be much lower than 0.056, which would have been the result for DBR. In all cases, however, the overall network blocking was much less than 0.177.

The plots in Fig. 1 lend credence to using mechanisms beyond DBR. In Section 4.4 we examine this further by considering the amount of capacity savings that could result. The plots in Fig. 1 also support our approach of providing different priorities to different service classes by implementing different methods for numbers of cached paths (PPC phase), routing schemes (UPO phase) and network controls (ARS phase).

4.2. Changing traffic matrices between 5% S1 and 20% S1

In the following sections, we make observations by comparing the plots for 5% S1 and 20% S1. For example, see Fig. 2 where the left plot is for 5% S1 and the right is for 20% S1. We see that as the fraction of traffic for the S1 service class increases (and S3 decreases), the composition of flows in the network is altered. In the traffic model, the average flow bandwidth of the S1 service class is much lower than that of S3, and hence as we move load from S3 to the S1 service class, we bring down the average bandwidth

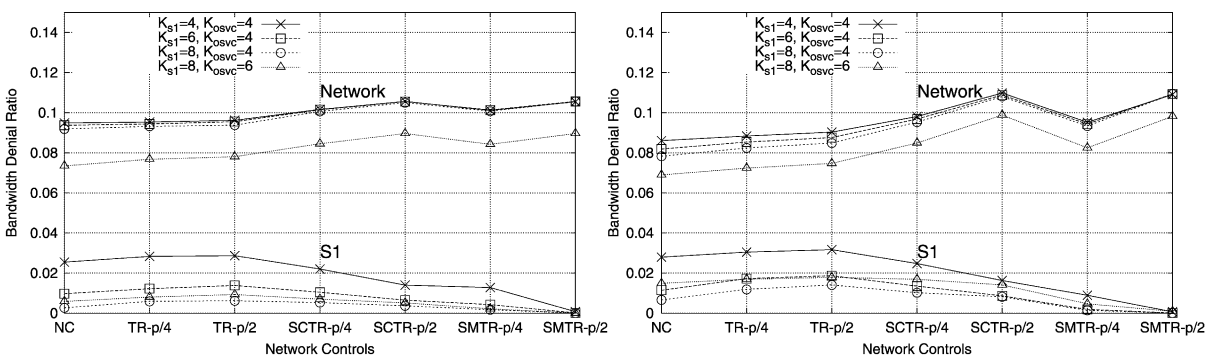

Fig. 2. BDR of S1 service class and overall network with $RS_{s1} = MACRPC$ and $RS_{osvc} = DRR$ for (left) 5% S1 and (right) 20% S1 with changing network controls.

per flow. Hence, the overall network performance, even for NCs, improves as we move from 5% S1 to 20% S1 in all the observed cases. Another check of intuition would be that as the fraction for the S1 service class increases, its impact on the network level BDR in all the cases is more pronounced. When service-specific controls (i.e. SCTR and SMTR) become active and alleviate the BDR of the S1 service class, the network pays a price in terms of the overall BDR. As the traffic matrix changes from 5% S1 to 20% S1, a heavier price is paid, since the price for rescuing a bigger fraction of S1 traffic is higher in all the cases.

4.3. Interactions between routing schemes, K_{s1}, and K_{osvc}

So far we have seen that we can provide improved performance to priority traffic by implementing priority in two phases at a time, namely in the ARS phase using TR, SMTR, etc. and either the PPC or UPO phase. Now we seek to integrate all three phases.

Routing schemes discussed in the Section 2.2 can be split into two categories based on their whether or not they allow crankback. Routing schemes like MACRPC accept a flow if there is any path which can accommodate the flow, restricted by the number of cached paths and the number of crankbacks allowed. While attempting to route a flow across longer paths, these schemes do well for the flows at hand but put a higher load on the overall network by moving flows over longer and longer paths. Hence, they perform well for lightly loaded networks whereas for networks with heavier loads, they have the tendency to move the network to higher loaded conditions. On the other hand, schemes like MACRPNC accept a flow only if it is allowed on the first try. Such a criteria reduces the possibility of getting the network heavily loaded since the flows will be blocked as soon as the network starts to get overloaded since chances of getting through a longer path are relatively small. Hence, these routing schemes keep the network in a more lightly loaded state by blocking the flows rather than moving them to longer paths. They perform worse in lightly loaded cases whereas for networks with higher loading they give much better performance. Routing schemes like DRR randomly move the load around (regardless of whether paths are shorter and longer) and give performance in between the two other discussed schemes.

In order to obtain better performance for the S1 service class, we will be more interested in cases when RS_{s1} = MACRPC or MACRIC and RS_{osvc} = MACRPNC and DRR. This will give the S1 class the chance to try to be admitted on many possible paths, where other service classes will have limited possibilities for paths that could be used.

4.3.1. RS_{s1} = MACRPC *and* RS_{osvc} = DRR

Returning to Fig. 2, we can see the BDR of the S1 service class and the overall network with RS_{s1} = MACRPC and RS_{osvc} = DRR for 5% S1 and 20% S1 with changing network controls. We observe that $K_{s1} = 8$ and $K_{osvc} = 6$ performs well, giving significantly lower overall network level BDR and fairly low BDR for the S1 service class. $K_{s1} = 4$ and $K_{osvc} = 4$ restricts the network too much and ends up performing poorly on both the network level and the S1 service class level. The S1 service class performance comparison for $K_{s1} = 6$ and $K_{osvc} = 4$ with $K_{s1} = 8$ and $K_{osvc} = 6$ is interesting as their relative performance levels flip between 5% S1 and 20% S1. In the case of 5% S1, both the BDR of the overall network and S1 decrease as we increase the number of cached paths for both S1 and other service classes. The small fraction of the S1 service class leads to minimal interference between classes. As the S1 fraction increases to 20% with $K_{s1} = 8$, S1 collides more with other service classes causing a higher BDR. Observe that a higher value of K_{osvc} allows more effective load of other service classes to get accepted and thereby forces the S1 service class to have a higher BDR. By allowing the S1 service class to use longer paths we allow inefficient use of links to ensure higher priority for the S1 service class. Such an approach might effectively be supplemented by increased controls like SMTR which allow better utilization of resources and also guarantee the required precedence.

One interesting observation is that TR degrades the performance of S1 as well as the overall network compared to NC; this is most noticeable in the case of 20% S1. This can be attributed to the fact that the load on the network is not uniformly distributed among the traffic pairs, whereas the TR is uniform over the network. Uniformly reserving capacity is harmful when lightly loaded parts of the network do not even need the reservation.

4.3.2. General conclusions

Other scenarios were considered like that in Fig. 2. In many ways the plots and system behavior were similar, and the following additional conclusions were made.

- Prioritization in the ARS phase has more impact compared to prioritization in the PPC phase. Increasing K_{s1} creates substantial improvement in S1 performance (with diminishing returns), since it provides a routing scheme like MACRPC with more paths to choose from. It is best to first increase K_{s1}, since it also may improve performance at the network level, than to use more extensive controls, since they hurt performance at the network level.
- Improvement at the network level can be affected substantially through the choice of K_{osvc}. An appreciable improvement can be seen in Fig. 2 as K_{osvc} was increased from 4 to 8. Also, performance at the network level was virtually the same for the same values of K_{osvc}, especially for 5% S1.
- Using a routing scheme like MACRIC does not improve the service class level performance much as compared to MACRPC. In other words, routing that uses instantaneously updated information (which has a heavy

signaling cost) is not necessarily advantageous over one where routing is based on updated information that is periodically disseminated.

4.4. Changing network capacity

In the Section 4, we evaluated the various options for PPC, UPO and ARS phases. From those results, we now examine them on the basis of varying network capacity. We intend to explore the relative changes encountered when these combinations are employed in comparatively lightly and heavily loaded networks.

First consider $K_{s1} = 8$, $K_{osvc} = 4$, $RS_{s1} = MACRPC$ and $RS_{osvc} = MACRPNC$. All controls are exercised at $p/4$. We present results in Fig. 3 for the BDR of the S1 service class and overall network (this time on separate plots) for 5% S1 with varying capacity. The most obvious results from Fig. 3 are the difference in blocking at all capacity levels between DBR and all of the other mechanisms. At 80% of base capacity, all of the other mechanisms can produce virtually the same BDR for S1 as does DBR at 130% of base capacity. On the curves for the overall network, the same performance is provided at 100% of base capacity for all other mechanisms as at 130% for DBR. Such capacity savings allow us to conclude that 30–50% more capacity is needed to provide the same performance for DBR as is provided by using alternate routing combined with network controls.

We also observe that for the overall network level BDR, TR performs slightly better than NC for heavier load but not as well as NC as the network is lightly loaded. For the S1 service class active control helps improve the S1 performance at lower loads but the controls and NC merge with increasing capacity. Rightfully, as the network gets enough capacity, the class with lower average flow bandwidth manages to get lower BDR even for NC. The observation draws our attention towards the influence of various controls and prioritization mechanisms on the BDR of the S1 service class. These mechanisms not only ensure good performance at higher capacities (NC also does it) but also at the relatively heavily loaded conditions. Hence, to a large extent these mechanisms can be used to avoid or delay upgrades of the network.

We investigated other combinations of PPC, UPO and ARS phases. Those results showed similar results as Fig. 3, except that the types of network controls caused more or less significant differences between curves depending on the combination.

4.5. Overloaded networks—Networks II and III

In this section, we check and select a few combinations of routing schemes and cached paths on two other network models to see if the same conclusions we have reached in previous sections would also be applicable here and applicable in general. This model keeps the same network topology but has a higher load to the point the network would be considered overloaded. There are three ways in which a network can made to be overloaded, (1) by decreasing the capacity of each link proportionately, (2) by removing selective links, and (3) by doing both. The two cases which we found interesting were by decreasing the capacity proportionately and by doing both (removing selective links and decreasing capacity proportionately).

4.5.1. Network II—proportional decrease

In Section 4.4, we have shown the performance of the S1 service class and overall network for chosen combinations with varying network capacity in Network I. For this experiment, we take a snap-shot when the network capacity is decreased to 66% of its capacity in Network I proportionately on all links. Such a downsized network is referred to as Network II. Here, we consider the performance of two combinations of routing schemes. We present results in Fig. 4 for the BDR of the S1 service class and the overall network for 5% S1 with various network controls. We observe that TR does have a small positive impact on the network level BDR here as the network is heavily

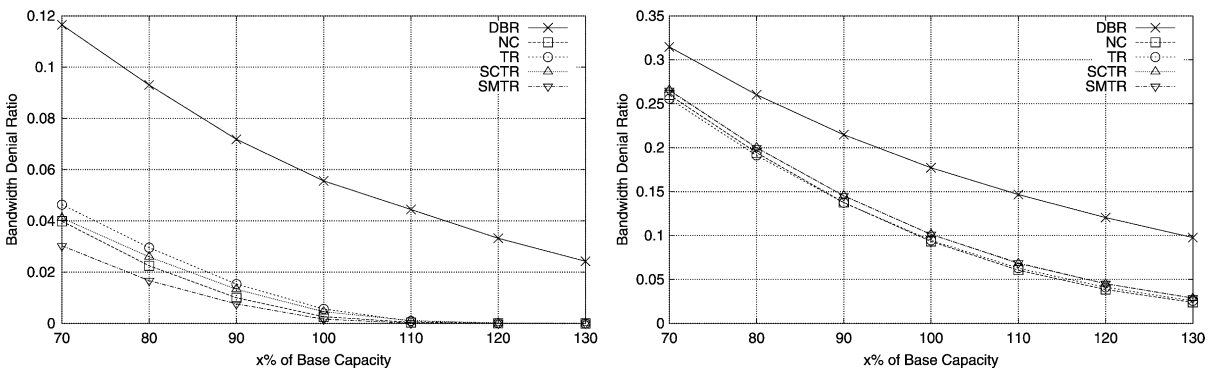

Fig. 3. BDR of (left) S1 service class and (right) overall network with $K_{s1} = 8$, $K_{osvc} = 4$, $RS_{s1} = MACRPC$ and $RS_{osvc} = MACRPNC$ for 5% S1 with varying capacity.

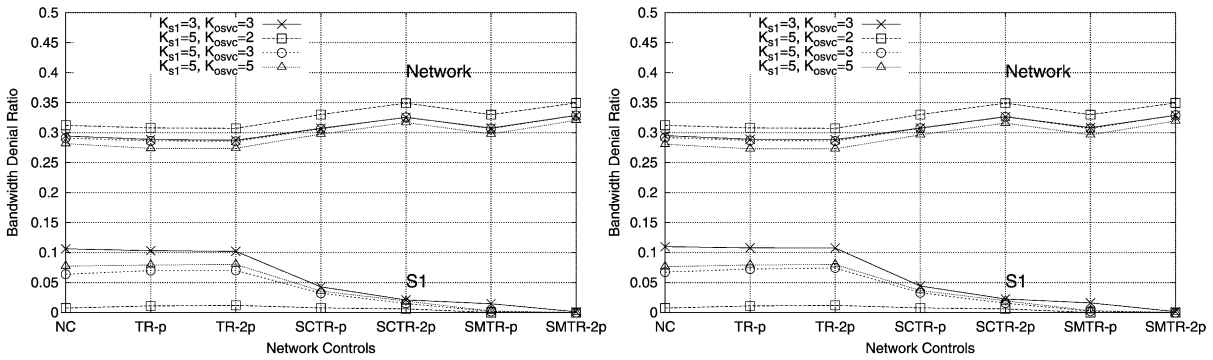

Fig. 4. BDR of S1 service class and overall network with (left) RS_{s1} = MACRPC, RS_{osvc} = DRR and (right) RS_{s1} = MACRPC, RS_{osvc} = MACRPNC for 5% S1 with increasing network controls.

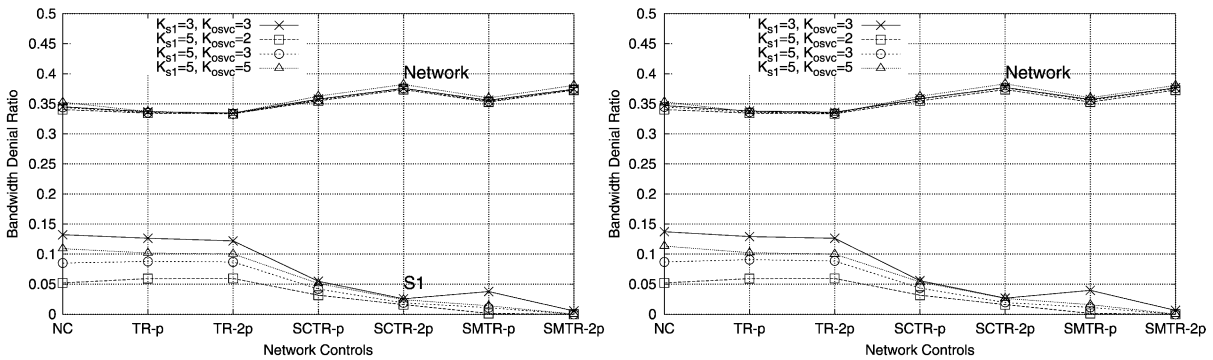

Fig. 5. BDR of S1 service class and overall network with (left) RS_{s1} = MACRPC, RS_{osvc} = DRR and (right) RS_{s1} = MACRPC, RS_{osvc} = MACRPNC for 5% S1 with increasing network controls.

loaded. In the first network model, TR made performance worse. We also observe that with the activation of S1-specific controls, S1 performance improves significantly whereas at the network level, a heavier price is paid as the BDR rises as much as 0.05. When compared with the normally loaded network in Network I, we observe that harsher reservation is warranted to ensure the same BDR for the S1 service class. Since the network level BDR is higher, more reservation (now at levels p and $2p$ instead of $p/4$ and $p/2$) is needed to bring the BDR of the S1 service class to the same level. Thus we see that deploying service-specific controls in a heavily loaded network causes more degradation in the overall network level performance.

4.5.2. Network III—link reduction and proportional decrease

Now consider the third network model, in which we removed links without any direct traffic and let the ones with direct traffic remain. This creates a network with more evenly distributed load since all links have direct traffic. Then, we decrease the network capacity to 66% of its capacity proportionately on all links. Such a downsized network suffers from high BDR. We present results in Fig. 5 for the BDR of the S1 service class and the overall network with RS_{s1} = MACRPC, RS_{osvc} = MACRPNC and RS_{s1} = MACRPC, RS_{osvc} = MACRIC for 5% S1 with increasing network controls. The performance results are quite similar for this network and for the second network. The only difference is with SMTR. Since classes have less options for finding alternate routes through idle links, multi-link TR has less effect on reducing BDR.

In examining these two additional networks, the main conclusion is that implementing priorities in numbers of cached paths, routing schemes, and network controls has similar benefit. So for the three networks studied we observed similar performance even with changing loading levels and load distribution. This allows us to view our conclusions as somewhat general in nature, rather than applicable only to a specific network.

5. Conclusion

In our work, we have compared various mechanisms of providing priority to a specific service class. The framework used for QoS routing for traffic engineering has three phases: PPC, UPO and ARS. The PPC phase characterizes the number of cached paths to be maintained at the node group for each service class. The UPO phase decides the degree of coordination between the node group routing entries and the state of links of the network. The ARS phase exercises the controls and decides the path to try for an

incoming flow. The choice made by the network during these three phases reflects the precedence given to the specific service class. We therefore see the priority provided to a service class as a point in this three-dimensional space and the values chosen (for caching, routing schemes and network controls) influence the performance as seen by a service class.

We started with a representative network and traffic, then conducted extensive simulations and attempted to get insights into the interdependence between the chosen values for the three phases. We also experimented with variations of traffic (different compositions of service classes) and three network models. We realized that doing too much along any one vector (phase) does not seem efficient and encounters diminishing returns. Keeping higher differences between number of cached paths for different service classes is effective for reasonable values of difference after which the performance of the overall network deteriorates with no gain at the service class level. Attempting to exercise harsher network controls (increasing the TR levels with respect to p) only leads to substantial degradation of the overall network level performance with minimal improvement in the performance of the service class. Along the same lines, using a routing scheme like MACRIC does not improve the service class level performance as much as compared to MACRPC. In other words, routing that uses instantaneously updated information (which has a heavy signaling cost) is not necessarily advantageous over one where routing is based on updated information that is periodically disseminated.

Such observation leads us to believe that efficient and effective prioritization can be achieved by giving soft (less harsh) priority at multiple phases to a service class. Such an approach not only causes lesser harm at the network level but also provides more effective priority to the service class. And finally, we can make the overall conclusion that the implementation of QoS routing schemes can provide substantial benefits when used in conjunction with network controls. We have not studied the network protocols that would be needed to implement these schemes, but rather have shown how TE mechanisms could provide substantial benefits, regardless of which candidate protocols are used.

References

[1] M. Pioro, A. Szentsi, J. Harmatos, A. Juttner, P. Gajowniczek, S. Kozdrowski, On open shortest path first related network optimisation problems, Performance Evaluation 48 (1–4) (2002) 201–223.

[2] B. Fortz, M. Thorup, Internet traffic engineering by optimizing OSPF weights, IEEE INFOCOM, March, 2000.

[3] E. Rosen, A. Viswanathan, R. Callon, Multiprotocol label switching architecture, Internet Engineering Task Force, RFC 3031, January 2001.

[4] Internet Engineering Task Force Traffic Engineering Working Group, http://www.ietf.org/html.charters/tewg-charter.html

[5] K. Muthukrishnan, A. Malis, A core MPLS IP VPN architecture, Work in progress, IETF draft-ietf-ppvpn-rfc2917bis-00.txt.

[6] C. Beard, V. Frost, Prioritized resource allocation for stressed networks, IEEE/ACM Transactions on Networking 6 (5) (2001) 618–633.

[7] C.C. Beard, V.S. Frost, Ticket servers for network traffic prioritization, Journal of Network and Systems Management 11(2) (2003).

[8] B. Brewin, Nation's networks see sharp volume spikes after attacks, Computerworld, September 17, 2001.

[9] E. Noam, Testing the communications network, The New York Times, September 24, 2001.

[10] The City of Oklahoma City, Alfred P. Murrah Federal Building Bombing: Final Report, Fire Protection Publications, Stillwater, Oklahoma, 1996.

[11] D. Medhi, Quality of service (QoS) routing computation with path caching: a framework and network performance, IEEE Communications Magazine 40 (12) 106–113 (2002).

[12] D. Awduche, et al., Requirements for Traffic Engineering Over MPLS, Internet Engineering Task Force, RFC 2702, September 1999.

[13] S. Srivastava, B. Krithikaivasan, V. Venkatachalam, C. Beard, D. Medhi, A. van de Liefvoort, W. Alanqar, A. Nagarajan, A case study on evaluating the benefits of MPLS Traffic Engineering through constraint-based routing and network controls, IEEE International Conference on Communications April (2002).

[14] R.J. Sivasankar, S. Ramam, S.P. Subramaniam, T. Srinivasa Rao, D. Medhi, Some studies on the impact of dynamic traffic in a QoS-based dynamic routing environment, IEEE International Conference on Communications June (2000).

[15] D. Medhi, I. Sukiman, Multi-service dynamic QoS routing schemes with call admission control: a comparative study, Journal of Network and Systems Management 8 (2) (2000) 157–190.

[16] M. Peyravian, A.D. Kshemkalyani, Network path caching: issues algorithms and a simulation study, Computer Communications 20 (8) (1997) 605–614.

[17] G. Apostolopoulos, R. Guerin, S. Kamat, S. Tripathi, Quality of service based routing: a performance perspective, Computer Communication Review 28 (4) 17–28 (1998).

[18] L.N. Rard, Impact of triggered update in a QoS-based dynamic routing environment with network path caching, Master's Thesis, Computer Networking, School of Interdisciplinary Computing and Engineering, University of Missouri-Kansas City.

[19] R.G. Gibbens, F.P. Kelly, P.B. Key, Dynamic alternate routing—modeling and behavior, Proceedings of ITC-12, Torino, Italy, 1988.

[20] G.R. Ash, Dynamic Routing in Telecommunications Network, McGraw-Hill, New York, 1998.

[21] F.P. Kelly, Routing and capacity allocation in networks with trunk reservation, Mathematics of Operations Research 15 (1990) 771–793.

[22] MuSDyR Manual, Computer Networking Technical Report, School of Interdisciplinary Computing Engineering, University of Missouri-Kansas City, 2000.

[23] R. Guerin, H. Ahmadi, M. Naghsineh, Equivalent capacity and its application to bandwidth allocation in high-speed networks, IEEE Journal on Selected Areas in Communications 9 (7) 966–981 (1991).