

Load Balancing Routing with Bandwidth-Delay Guarantees

Kartik Gopalan Tzi-cker Chiueh Yow-Jian Lin
Florida State University Stony Brook University Telcordia Technologies*
kartik@cs.fsu.edu chiueh@cs.sunysb.edu yjlin@research.telcordia.com

Abstract

Current generation of network carriers compete intensely to satisfy the diverse wide-area connectivity requirements of customers. At the same time, the carriers inherently wish to maximize the usage efficiency of their network infrastructure. Much of the research in network resource management has been devoted to providing bandwidth guarantees and preventing network congestion. However, the rapid growth in number and diversity of real-time network applications has made it imperative to consider the impact of end-to-end delay requirements of traffic on network resource provisioning. We present an efficient network resource provisioning algorithm, called Link Criticality Based Routing (LCBR), which relies on the guiding theme that *load balancing leads to higher resource utilization efficiency*. LCBR applies a simple but very effective notion of link criticality to achieve network-wide load balance while simultaneously meeting the QoS requirements of bandwidth and end-to-end delay. In addition, LCBR can simultaneously provision both primary and backup routes to support fast recovery from node or link failures. This paper reviews the state-of-the-art in network resource provisioning with QoS guarantees, introduces the LCBR algorithm, and identifies future research challenges.

1 Introduction

In recent years, network carriers have witnessed a surge in the demand from their large organizational customers for dedicated wide-area connectivity with network services tailored to customers' performance requirements. Such customer-specific customization of network services is exemplified by the growth of content distribution services, real-time financial transaction networks, E-government, E-commerce and supply-chain management. An emerging example of such a customized network service offering is the wide-area Virtual Private Network (VPN) with Quality of Service (QoS) guarantees (or QVPN). The customization trend is being coupled with a rapid growth in the amount of real-time network traffic that is serviced by carriers' net-

work infrastructure. An additional force that influences resource allocation decisions is the carriers' inherent desire to maximize their revenue base by accommodating the requirements of as many customers as possible. These three competing forces have created an urgent need for network resource provisioning techniques that maximize the utilization efficiency of the network infrastructure.

Multi-Protocol Label Switching (MPLS) [13] has recently gained popularity as a technology for managing network resources and providing performance guarantees. In line with the philosophy of "*route at the edge and switch in the core*", MPLS allows aggregated traffic to be switched through long-term traffic tunnels, also known as Label Switched Paths (LSPs). Aggregation is performed at the network edges and the LSP's route may traverse a number of traditional ATM, frame relay, or custom-built MPLS switches in the network interior. MPLS can provide different forms of QoS guarantees to QVPN traffic by mapping each QVPN to a unique LSP. For instance, a long-distance VoIP trunk that carries real-time voice traffic may be mapped to a LSP that guarantees long-term bandwidth and end-to-end delay bound, in addition to protection against network failures along the QVPN path.

Determining the route taken by a QVPN forms the most critical component of the network resource management process since it involves resource provisioning decisions at the scale of the entire network. Traditional hop-by-hop dynamic routing [12] forwards packets along the shortest-path based purely on destination address in the packet header. Such an approach is sufficient for best-effort traffic but makes inefficient use of network resources as it forwards packets along already congested shortest paths while longer uncongested paths may never be utilized. On the other hand, QoS routing [2, 9] attempts to find a feasible route for a traffic flow with QoS constraints. Such schemes could consider either single QoS dimension [9] or multiple dimensions [2]. The main objective of QoS routing is to find a feasible route that satisfies the given QoS constraints rather than to optimize for network resource usage.

Explicit routing techniques, on the other hand, provide significant benefits over the traditional hop-by-hop and dynamic QoS routing schemes. Explicit MPLS-based routing provides the opportunity to intelligently tailor the route

*Research performed while at Stony Brook University

taken by each QVPN such that different parts of the network remain equally loaded. Such an approach avoids the creation of bottleneck links and maintains a high network resource utilization efficiency.

Earlier research efforts at traffic engineering have primarily addressed either bandwidth guarantees [9, 10, 14] or congestion management [1]. However, with the increasing number of network applications that generate time-sensitive network traffic, the delay dimension of the QoS requirements is having an increasingly significant impact on traffic engineering decisions. Solutions, such as [5, 11], address the delay dimension of QoS requirements, but do not focus on the impact of the delay dimension on traffic engineering considerations. Earlier solutions either deal with fixed propagation delays and fixed link costs or consider finite values of discrete link delays and delay-dependent costs. Approaches such as [6, 15] consider automated traffic engineering using offline and online techniques. Alternatively, one can even over-provision resources in order to support a given delay requirement for aggregate backbone traffic [7]. Additionally, fault-tolerance along QVPN routes, in conjunction with delay and bandwidth guarantees, has not been explicitly considered in earlier traffic engineering approaches.

In this paper, we address the problem of selecting primary and backup routes for QVPNs that have both long-term bandwidth and end-to-end delay bound requirements. We propose an algorithm called *link criticality based routing* (LCBR) to select primary and backup routes that satisfy each QVPN's QoS requirements while improving the long-term resource usage efficiency of the network. *The guiding theme of LCBR is that load balancing leads to higher resource usage efficiency.* By ensuring that loads on different links of the network are as balanced as possible, LCBR can prevent critical network resources from being exhausted early and becoming a bottleneck for the entire network.

The first unique aspect of LCBR is that it quantifies network-wide load balance using a simple notion of link criticality that is based on measured traffic profile and ingress-egress node information. The second unique aspect of LCBR is that it incorporates the impact of end-to-end delay partitioning on the network-level route selection decisions. Thirdly, the framework of LCBR allows us to provision backup routes for fault-tolerance in conjunction with bandwidth and delay guarantees. Finally, LCBR accounts for a new QVPN's future impact on the network load balance during the online route selection phase itself, i.e. before the QVPN becomes operational. It is difficult to perform such impact analysis in advance using route selection schemes that rely upon some form of shortest path algorithm.

2 Model of Network Operations

We consider a network of routers and links that are under the administrative control of a carrier. A subset of routers are known to be ingress and egress points for network traffic. The administrative control over network resources could either be exercised in a centralized manner by a single entity with knowledge of global network state or in a distributed manner with the aid of a distributed link state protocol such as OSPF [12]. The work proposed in this paper fits in both the models of network resource management. QVPNs that carry customer's aggregate traffic are long-lived virtual connections possibly lasting several days or even months at a stretch. Additionally, QVPN requests for aggregate traffic flows arrive relatively infrequently when compared to the typical lifetime of TCP/IP or UDP sessions. The resource allocation mechanisms operate in an *online* environment with no apriori knowledge of which QVPN requests might arrive in future. Furthermore, re-routing of QVPNs that have already been set up is expensive in terms of service disruptions and re-allocation costs.

3 Network-wide Load Balance

The principal intuition behind LCBR is to select routes that best balance the loads across different parts of the network and keep the critical network links available for future QVPN requests. To illustrate the concept, consider the following simple example. Given the network topology in Figure 1, we need to select a route for a QVPN F_1 between nodes S_1 and D_1 . There are two candidate routes: (S_1, A, B, D_1) and (S_1, C, D, D_1) . Which of these two routes is better from the perspective of long-term network usage efficiency? Suppose we expect future QVPN requests between S_2 and D_2 as well, but do not know the exact QoS requirements of these QVPNs. Then the better route to select for QVPN F_1 would be (S_1, C, D, D_1) because it leaves the resources along the link (A, B) free for future QVPN requests between S_2 and D_2 . This example illustrates that route selection process should, as far as possible, avoid overloading those physical links that are of critical importance to a large number of source-destination pairs. By ensuring that different parts of the network are evenly loaded, one can ensure that no single link becomes a bottleneck resource.

But how exactly can one quantify the criticality of a network link and its impact of network load balance without having a precise knowledge of future QVPN request distribution? This forms the fundamental challenge in designing an effective online route selection algorithm.

The state-of-the-art in addressing this problem from a traffic engineering perspective is the Minimum Interfer-

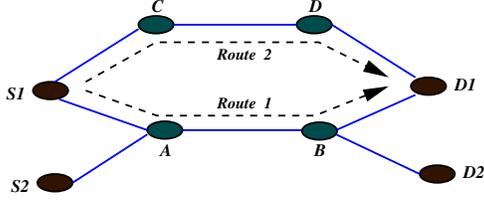


Figure 1: A simple problem of route selection. Selecting the route (S_1, C, D, D_1) leaves the link (A, B) free for future QVPNs between S_2 and D_2 .

ence Routing Algorithm (MIRA) [10]. MIRA identifies the critical network links by using the knowledge of possible ingress-egress pairs in the network. The criticality of a link in MIRA depends upon the number of mincuts between the different ingress-egress pairs that the link belongs to. MIRA’s notion of link criticality is useful to avoid links that impact the maxflows of a large number of ingress-egress pairs, but proves to be insufficient in identifying critical links that may not be part of any mincuts. Profile-Based Routing (PBR) [14] is another approach that attempts to remedy this problem by using measured profile of past traffic to solve a multi-commodity network flow problem and to set up advanced reservations for each traffic class. The advance reservations guide and limit the actual flow reservations during online provisioning.

Both MIRA and PBR are designed to address QVPNs that require long-term bandwidth reservations, but do not address QVPNs that require end-to-end delay guarantees as well. Furthermore, both are based on variations of Dijkstra’s shortest path algorithm which has inherent limitations in simultaneously tackling the twin requirements of bandwidth and delay, as will be explained in Section 4. LCBR, on the other hand, is designed to address both bandwidth and delay requirements of aggregate QVPNs. The notion of link criticality and network-wide cost metrics in LCBR are designed ground up to capture the extent of current and expected future load balance in the network. Let us now look at the details of link criticality and network cost metrics in LCBR.

3.1 Link Criticality

The criticality of a link in LCBR is measured by the notion of future *expected load* on each link, which indicates how critically different source-destination pairs in the network need a link for carrying their traffic. A link that is expected to carry sizeable amount of traffic between different source-destination nodes would be more critical than one that is expected to carry less. More formally, assume that a total of x network routes are possible between a source-destination pair (s, d) and y of these routes pass through a link l . Then the criticality $\phi_l(s, d)$ of the link l with respect to source-

destination pair (s, d) is defined as the fraction y/x of routes between s and d that pass through link l .

Assume that we have knowledge of the expected bandwidth demand $B(s, d)$ between each source-destination pair (s, d) based on measured daily traffic profiles or service-level agreements. The total expected load ϕ_l on link l is defined as the fractional sum $\sum_{(s,d)} \phi_l(s, d)B(s, d)$ of expected demands on the link l from all possible source-destination pairs in the network. Since only a small subset of all the nodes in the network are typically possible sources or destinations for QVPN traffic, computing ϕ_l does *not* involve an exhaustive computation of all the n^2 possible values of $\phi_l(s, d)$, where n is the number of nodes in the network. Furthermore, link criticality $\phi_l(s, d)$ is largely static since it is completely determined by topology of the network and changes only when the topology changes. Similarly, $B(s, d)$ changes relatively infrequently, such as on a daily basis. Thus the values of ϕ_l can be periodically pre-computed offline and kept ready for use in the online route selection phase described next.

3.2 Quantifying Network-wide Load Balance

Let C_l be the total bandwidth capacity of a link and R_l be its residual (unreserved) capacity at any time. Since the goal of route selection is to maintain a high level of load balance in the network, it is important to get an objective measure of the load balance at any instant. Towards this end, we begin by defining the dynamic cost of each link $cost(l) = \phi_l/R_l$ as the expected load per unit of available capacity at link l . Thus, a link l with small residual capacity R_l or large expected load ϕ_l is considered more expensive for use in QVPN routes. The extent of load balance in a network G is measured by a metric $cost(G) = \sum_{l \in G} \left(cost(l) - \frac{\phi_l}{C_l} \right)^2$, which represents the squared magnitude of the distance vector between the actual link costs and the minimum link costs in the network G . The term ϕ_l/C_l represents the minimum value of link cost when the residual capacity is maximum at $R_l = C_l$. A smaller value of $cost(G)$ represents a higher degree of load balance and vice-versa. Ideally, we would like the cost of the network to be as close to the idle-state operating point $(\phi_1/C_1, \phi_2/C_2)$. Squared sum captures the impact of both magnitude of individual link costs as well as the variations among them.

4 Link Criticality Based Routing

We now consider the problem of selecting a route between source s and destination d along which a QVPN F_N can be set up with QoS guarantees. We are interested in two forms of QoS guarantees for F_N : (a) the end-to-end delay encountered by packets of F_N should be smaller than D_N , and (b) the long-term bandwidth requirement ρ_N of

F_N must be satisfied at each link along the route. The route must be selected so as to maintain network-wide load balance, which in turn translates to minimizing the network cost metric $cost(G)$. We first examine why the direct application of shortest-path algorithm is not feasible. Next we address the problem of primary route selection followed by primary-backup route selection.

4.1 Why not Dijkstra’s algorithm?

Since each link’s contribution to the network cost-metric can be uniquely identified by the term $(cost(l) - \phi_l/R_l)^2$, it may be tempting to apply the Dijkstra’s shortest path algorithm in order to select the route which minimizes $cost(G)$. However, the simultaneous presence of delay and bandwidth constraints on the link prevent us from applying the shortest path algorithm directly. Pure bandwidth constraints can be handled using the approach proposed in [9] by eliminating links with smaller bandwidth than the required bandwidth and applying shortest path algorithm on the residual graph. However, presence of end-to-end delay constraints further complicates the problem due to several reasons.

First, each link l is capable of supporting a range of delay budgets for F_N depending upon the amount of bandwidth reserved for F_N at link l . Specifically, the larger is the amount of reserved bandwidth, the smaller is the queuing delay experienced by F_N ’s packets at the link. This raises a number of possibilities for partitioning the end-to-end delay budget of a QVPN among the individual links of a route. However, in order to perform the delay partitioning, we need to know which links constitute the route. Since the classical shortest path algorithm incrementally builds the route one link at a time, it cannot solve the end-to-end delay partitioning problem.

Secondly, the process of reserving link bandwidth along a route increases the load on the constituent links and alters the network-wide load balance. For instance, links that were lightly loaded before assigning resources to a new QVPN may become heavily loaded after the QVPN becomes active. Consequently, in order to maintain network load balance, we need to look ahead and account for the future impact of the route selection decision. Again, the incremental route construction approach of shortest path algorithm cannot account for the future impact of selected route on network load balance. An alternative solution could be to examine every possible route between source and destination explicitly, perform end-to-end delay partitioning and select the route that best maintains the load balance. However, this solution suffers from the drawback that number of routes between any source-destination pair would grow exponentially with network size and connectivity.

Input: 1. New QVPN specification (s, d, D_N, ρ_N)
 2. Each link’s expected load ϕ_l , residual capacity R_l and total capacity C_l .
 3. Set S of k shortest path routes between s and d .
 Output : Primary route X_N

For all links l do
 $cost(l) = \frac{\phi_l}{R_l}$

$cost_{min} = \infty$;
 For each route $r \in S$ that satisfies (D_N, ρ_N) do
 Partition the D_N among links of route r .
 Recompute resulting residual link capacities R'_l .
 Recompute the link costs $cost(l) = \frac{\phi_l}{R'_l}$.
 Recompute the network-wide cost metric $cost(G)$.
 If $cost(G) < cost_{min}$ then
 $cost_{min} = cost(G)$; $X_N = r$.

If $(cost_{min} > \alpha)$ then
 Reject F_N
 else
 Select route X_N as primary route for F_N

Figure 2: The P-LCBR algorithm selects the primary route for a QVPN F_N between source s and destination d . F_N requires an end-to-end delay bound of D_N and bandwidth of ρ_N .

4.2 The Primary LCBR Algorithm

In practice, shorter routes typically tend to utilize fewer network resources in comparison to longer routes and hence it is *very likely* that the route which best minimizes the network-wide cost metric is one among a set of k -shortest candidate routes. P-LCBR algorithm applies this insight to narrow down the set of candidate routes that can minimize $cost(G)$ to those having fewer links. P-LCBR performs route selection in two phases - offline and online. In the offline phase, performed once for the entire network, P-LCBR pre-computes the set of k -shortest candidate routes between each source and destination and computes the expected load ϕ_l for each link based on the computed candidate routes. A set of k -shortest candidate routes can be pre-computed using well known algorithms such as [4].

The online phase of P-LCBR is illustrated in Figure 2 and executes upon the arrival of each new QVPN request. The algorithm first computes $cost(l) = \phi_l/R_l$ for each link l in the network using the pre-computed value ϕ_l and current residual capacity R_l . For each pre-computed candidate route r between s and d , P-LCBR performs the following sequence of three operations. (1) It checks if the QoS requirements (D_N, ρ_N) of F_N can be satisfied by the available resources along route r . (2) If there are sufficient resources, then P-LCBR partitions the end-to-end delay D_N among the links of r . Existing algorithms to partition end-to-end delay are described and compared in [8]. (3) After partitioning D_N among links of r , P-LCBR recomputes the

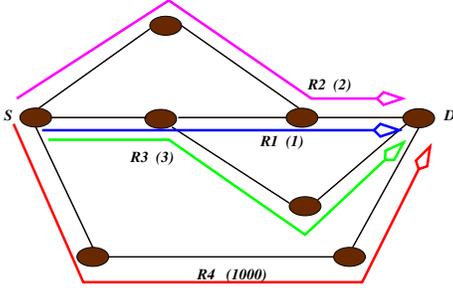


Figure 3: Disjoint Route Pair Selection Problem.

per-link remaining capacity R_l and the projected network-wide cost $cost(G)$ that would result if route r is assigned to F_N .

The route set up request for F_N is rejected if either (a) no route r has sufficient resources to satisfy F_N 's QoS requirements or (b) the minimum projected value of $cost(G)$ for any route r is greater than a pre-defined cost threshold α . Latter case indicates that admitting F_N would take the network to a highly critical state that may not be conducive for admitting future QVPN requests. If these two checks do not reject the QVPN F_N , then P-LCBR assigns F_N to a route r that yields the minimum value of $cost(G)$.

4.3 Primary-Backup LCBR Algorithm

Now let us consider the problem of finding both a primary route X_N and a backup route Y_N for a new QVPN request F_N . In addition to same bandwidth and delay guarantees as the primary route, the backup route provides the guarantee that if at most one network element (a link or a node) along primary route X_N fails, then the F_N 's traffic would be successfully diverted to the backup route Y_N . The goal of PB-LCBR algorithm is to select a disjoint route pair that minimizes $cost(G)$. There are two approaches: a greedy approach that involves less pre-computation, but might result in inefficient resource allocation and a non-greedy approach that achieves more efficient resource allocation at a slightly higher pre-computation cost.

The Greedy PB-LCBR Algorithm A simple greedy approach to selecting primary and backup routes is as follows. First select a primary route X_N that yields the smallest value of $cost(G)$ using the algorithm in Figure 2. Then derive a reduced network graph G' from G by removing all the network elements in the route X_N . Finally, use the algorithm similar to P-LCBR to select a backup route Y_N from the reduced graph G' that yields the smallest value of $cost(G)$. This greedy PB-LCBR algorithm outlined above is simple but leads to inefficient and skewed solutions. For instance, in Figure 3, ideally $R2$ should be the primary and $R3$ the backup, for a total cost of 5. However, the greedy

approach would select $R1$ as the primary, which leaves $R4$ as the only choice for disjoint backup route, for a total cost of 1001. The main reason for this inefficiency is the fact the greedy approach decouples the selection of primary and backup routes into two separate phases.

The Non-Greedy PB-LCBR Algorithm The non-greedy PB-LCBR algorithm overcomes the problem associated with the greedy approach by examining both the candidate primary and backup routes. The algorithm is similar in structure to P-LCBR algorithm consisting of an offline phase and an online phase. However, there are two important variations that deserve mention.

First, the offline phase of PB-LCBR pre-computes a set of $k = k_1 \times k_2$ candidate primary-backup route pairs for every source-destination pair in the network (as opposed to just k candidate primary routes in the case of P-LCBR). This pre-computation can be performed by first discovering k_1 -shortest candidate primary routes from the network graph G . Next for each candidate primary route X , we discover k_2 candidate backup routes from the residual network graph that excludes the links and nodes along route X .

Second variation in PB-LCBR concerns the online route selection phase. The set of pre-computed route pairs (X, Y) computed during the offline phase are supplied as input to the online phase. The only difference from P-LCBR is that for each candidate route pair (X, Y) (1) both primary route X and backup route Y are checked to ensure that sufficient resources are available to satisfy F_N 's QoS requirements and (2) end-to-end delay constraint D_N is partitioned along both X and Y . Finally, just as in P-LCBR, candidate route pair that yields the minimum value of $cost(G)$ is selected as the primary-backup route.

It should be noted that failures occur infrequently in typical networks and all solutions need to contend with the issue that resources reserved along the backup route remain unutilized till a failure occurs along some primary route. This inefficiency can be addressed using the *backup resource aggregation* [3, 8] approach in which QVPNs, whose primary routes are completely independent, share their reservations on the common links along their backup routes.

5 Effectiveness of Load Balancing

In this section, we briefly compare the performance of three load balancing routing algorithms (a detailed evaluation is presented in [8]). The first algorithm applies the load balancing criteria of the Widest Shortest Path (WSP) algorithm [9] to the delay-bandwidth constrained route selection problem. The WSP variant examines each candidate route (route pair) in the order of increasing length, and among all the feasible candidates with minimum length, the candi-

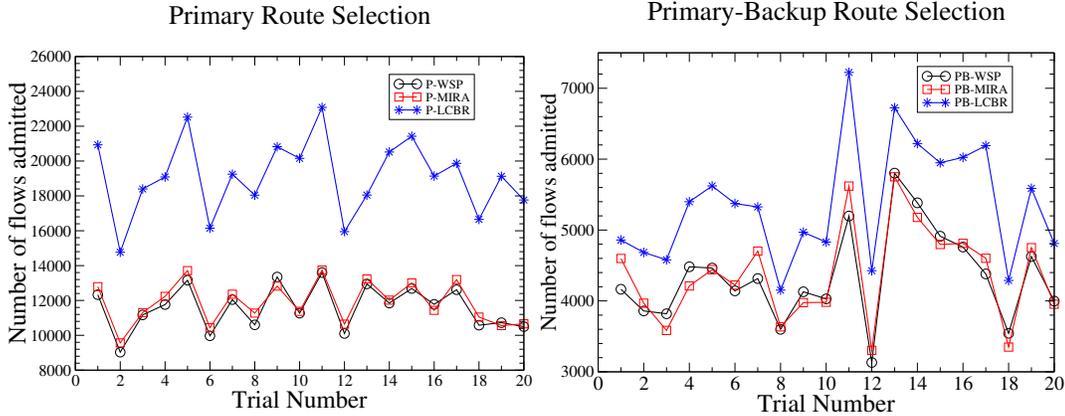


Figure 4: Number of QVPNs admitted for primary and primary-backup route selection.

date having maximum residual bottleneck link capacity is selected. The second variant similarly applies the route selection criteria from the Minimum Interference Routing Algorithm (MIRA) [10] to the delay-bandwidth constrained route selection. In all the algorithms, the intra-path delay partitioning is performed using the Load-based Slack Sharing algorithm [8]. Simulations are performed using the AT&T CERFnet nationwide backbone topology, with link capacities between 45 Mbps to 200 Mbps. QVPNs request a QoS of 100Kbps average rate and 50ms end-to-end delay.

Figure 4 provides a snapshot comparison of the three load balancing routing algorithms. In all the simulated scenarios, LCBR consistently admits more QVPNs than WSP and MIRA. LCBR performs better since it bases routing decisions on a network-wide load balancing criteria. Specifically, LCBR algorithm maintains a lower value of $cost(G)$ for a longer duration compared to WSP and MIRA algorithms. On the other hand, WSP does not consider the ingress-egress profile information, and performs only limited load balancing by selecting the widest shortest route among all candidates. In general, MIRA and WSP perform similarly on the AT&T CERFnet topology. MIRA does take into account the ingress-egress information in determining link criticality. In situations where MIRA does perform worse, the links along the mincut between ingress-egress pairs do not faithfully represent the most critical links in the network, thus defeating the link metric based on mincuts.

6 Issues and Challenges

Several open research issues remain to be tackled in the area of efficient network resource provisioning with QoS guarantees. First, the optimal solution to the load balancing routing problem in general is unknown. Hence there is a need to quantify the room for further improvement over state-of-the-art heuristic approaches. Second, LCBR

presently assumes that each QVPN is mapped to a single route (route pair). The overall utilization efficiency of the network can be further improved, and resource fragmentation can be alleviated, if it is possible to use multiple paths to support QVPN traffic. However, this raises the research question of how many paths should be used, how to partition a QVPN into multiple paths, and how to cumulatively guarantee QoS across multiple paths. Third, most current approaches account only for the maximum bandwidth requirements of the QVPNs, but not their actual loads. In practice, QVPNs rarely carry their full traffic loads. Therefore, another significant research challenge is to effectively incorporate statistical multiplexing in the load balancing routing framework. Finally, the development of network resource management concepts in the inter-domain context, where QVPN routes may straddle across multiple administrative domains, would be of considerable practical significance.

7 Conclusions

Modern network service providers' primary challenge is to deliver guaranteed performance while maximizing the return-of-investment in their network infrastructure. We have argued that the key to maximizing resource usage efficiency is to maintain a high degree of network-wide load balance in the route selection process. In this paper, we first summarized the state-of-the-art techniques for network resource provisioning with QoS guarantees. Next we introduced the Link Criticality Based Routing (LCBR) algorithm which selects primary and backup routes for aggregate traffic flows with end-to-end delay, long-term bandwidth and fault-tolerance guarantees. LCBR applies a simple yet effective notion of link criticality to maintain high degree of load balance and resource usage efficiency across the network. Finally we outlined some of the open research challenges in the area of network resource provisioning.

References

- [1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and principles of internet traffic engineering. RFC 3272, May 2002.
- [2] S. Chen and K. Nahrstedt. On finding multi-constrained paths. In *Proc. IEEE ICC'98*, June 1998.
- [3] K. Dovrolis and P. Ramanathan. Resource aggregation for fault tolerance in integrated services packet networks. *ACM Computer Communications Review*, 28(2):39–53, April 1998.
- [4] D. Eppstein. Finding the k shortest paths. In *Proc. of the 35th Annual Symposium on Foundations of Computer Science*, pages 154–155, November 1994.
- [5] F. Ergun, R. Sinha, and L. Zhang. QoS routing with performance dependent costs. In *Proc. of INFOCOM 2000, Tel Aviv, Israel*, March 2000.
- [6] A. Feldmann and J. Rexford. IP network configuration for intradomain traffic engineering. *IEEE Network Magazine*, 15(5):46–57, September 2003.
- [7] C. Fraleigh, F. Tobagi, and C. Diot. Provisioning IP backbone networks to support latency sensitive traffic. In *Proc. of INFOCOM'03, San Francisco, CA*, March 2003.
- [8] K. Gopalan. Efficient provisioning algorithms for network resource virtualization with QoS guarantees. Ph.D. Thesis, Computer Science Dept., Stony Brook University, Aug. 2003.
- [9] R. Guerin, A. Orda, and D. Williams. QoS routing mechanisms and OSPF extensions. In *Proc. of IEEE Globecom'97, Phoenix, AZ*, volume 3, pages 1903–1908, Nov. 1997.
- [10] M.S. Kodialam and T.V. Lakshman. Minimum interference routing with applications to MPLS traffic engineering. In *Proc. of INFOCOM'2000, Tel Aviv, Israel*, pages 884–893, March 2000.
- [11] D.H. Lorenz, A. Orda, D. Raz, and Y. Shavitt. Efficient QoS partition and routing of unicast and multicast. In *Proc. of IWQoS 2000, Pittsburgh, PA*, pages 75–83, June 2000.
- [12] J. Moy. OSPF Version 2. RFC 2328, April 1998.
- [13] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. RFC 3031, Jan. 2001.
- [14] S. Suri, M. Waldvogel, D. Bauer, and P.R. Warkhede. Profile-based routing and traffic engineering. *Computer Communications*, 26(4):351–365, 2003.
- [15] P. Trimintzios, P. Flegkas, and G. Pavlou. Policy-driven traffic engineering for intra-domain quality of service provisioning. In *Proc. of QoS/ICQT'02, Zurich, Switzerland*, October 2002.