A Routing Architecture for Scheduled Dynamic Circuit Services

Malathi Veeraraghavan
ECE Department
University of Virginia
mvee@virginia.edu

David Starobinski
ECE Department
Boston University
staro@bu.edu

ABSTRACT

A wide range of applications in the nascent field of cloud computing and in other fields require stable, low-delay wide-area connectivity together with capabilities of co-scheduling network and server resources. As a result, both research and commercial providers have recently started to deploy *scheduled dynamic circuit services (SDCS)* as a complement to their IP-routed and leased-line service offerings. Under such services, corporations, universities, or end-users dynamically place requests for a fixed-rate circuit lasting for a fixed duration, for either earliest possible usage or scheduled usage at some point in the future. The lack of adequate support for inter-domain routing and provisioning represents, however, a major hurdle to the full-scale deployment of SDCS. Indeed, existing protocols, such as BGP, do not cater to future dynamic allocation and scheduling of network resources. In this paper, we introduce a strawman protocol, called *scheduled circuit routing protocol (SCRP)*, geared towards inter-domain routing in SDCS-based architectures. The purpose of this protocol is to report available bandwidth information as a function of time across domains without revealing internal network topologies. We describe trade-offs associated with the design of such a protocol, more particularly the challenge of ensuring high performance while maintaining low implementation complexity, and present avenues for future research.

1. INTRODUCTION

A new type of communication service is now being offered by both research-and-education network providers and commercial providers. It is a *scheduled dynamic circuit service (SDCS)*, in which a user can dynamically request a fixed-rate circuit of fixed duration, either for earliest possible usage or usage at specific times in the future. By requiring users to specify durations, network schedulers know when ongoing circuits will terminate, and can thus schedule new circuit requests to start at specific future times.

There is currently no routing protocol standardized for SDCS. The research-and-education (REN) offering implements a protocol called InterDomain Controller Protocol (IDCP) [3], developed jointly by several backbone RENs (Internet2 and ESnet in the US, GEANT2 in Europe, and Canarie in Canada). In the IDCP architecture, a centralized scheduler is deployed in each domain. The goal of IDCP is to allow inter-domain schedulers to communicate with each other. While circuit scheduling and circuit provisioning aspects of IDCP have been developed, the routing ones have still not been addressed.

In this paper, we propose an approach to the design of a distributed routing protocol to support SDCS. By distributed, we mean that each domain can deploy one or more schedulers. We use the term "zones" to denote areas of responsibility for each scheduler. Small domains may consist of single zones while larger ones may have multiple zones. Schedulers in border zones, i.e., zones that contain border routers (those interconnected with routers in other domains), communicate via a new routing protocol called Scheduled Circuit Routing Protocol (SCRP). Like BGP, this protocol conveys reachability. But in addition, it carries time-denoted available bandwidth information to allow these schedulers to cooperatively select time ranges during which the requested bandwidth is available across the path.

The rest of this paper is organized as follows. We first describe the motivation for the development of SDCS. Next, we review current deployments, and briefly discuss related work. Thereafter, we introduce a strawman routing protocol for SDCS. As we describe the protocol, we highlight open research issues, including fundamental trade-offs resulting from the conflicting requirements of maximizing network resource utilization on the data plane while minimizing overhead on the control plane.

ACM ReArch 2010, November 30, 2010, Philadelphia, USA.

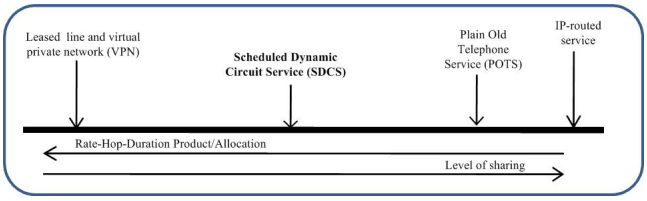

Figure 1: A spectrum of communication services

2. SCHEDULED DYNAMIC CIRCUIT SERVICES (SDCS)

This section answers the following big-picture questions. How do scheduled dynamic circuit services (SDCS) fit into the spectrum of traditional communication service offerings? What are the applications for such services, and where are these services deployed?

SDCS versus traditional communication services.

We organize services on a horizontal line as shown in Fig. 1 [19]. The measure that increases as one moves along the line from right-to-left is the *per-allocation rate-hop-duration product*. For example, each "allocation" in an IP-routed service across an Ethernet network is a maximum of 1500 bytes on just one hop (link). At the other extreme, even a relatively low-rate T1 six-month lease is an allocation of almost 3 TB on multiple hops (links on the end-to-end path of the T1 line). For Plain Old Telephone Service (POTS), using an average holding time of 3 minutes, the allocation is about 1.4 MB on multiple hops.

From these three examples, we see that there are significant gaps in the service offerings from a rate-hop-duration product perspective. In addition to these gaps, the only service at the high-end, leased-line service, is highly restrictive in that the two endpoints of a leased-line circuit have to be identified at the time of purchase. In contrast, scheduled dynamic circuit services (SDCS) customers first connect into SDCS-provider networks via *SDCS-access links*, and then, on an as-needed basis, request circuits with fixed rate and duration to any other SDCS customer. As Metcalfe noted, the value of a network service grows with the number of endpoints to which any single endpoint can connect [16]. Thus, SDCS addresses both the rate-hop-duration product gap and connectivity issues.

Having noted above how SDCS differs from leased-line service, consider next how it differs from standard circuit-switched (e.g., POTS) or virtual circuit-switched services (e.g., ATM and IntServ). POTS and the like only allow immediate-usage requests: the network either admits or rejects the call. There is no earliest-start time or scheduled future allocation by the network. Such a mode of operation, in which calls are simply blocked if resources are unavailable and have to retry at a later time, is economically viable for ordinary users if the number of simultaneous circuits supported on a link is large. In other words, the per-circuit rate is small relative to link capacity. In this scenario, network utilization can be high enough to keep costs low even while offering users a low call blocking probability.

On the other hand, a major goal of SDCS is to offer users a higher rate-hop-duration product/allocation, which could mean high per-circuit rates. Applying Erlang based teletraffic theory, it can be shown that operating a link at high utilization in an accept-or-reject mode comes with a concomitant cost of high call blocking probability if the number of circuits is small. By including network schedulers that can offer users an earliest-start time rather than simply rejecting the call if resources are unavailable, it becomes feasible to substantially increase utilization. Thus, previous work by the authors [23] shows that there exist scenarios where, over reasonable time horizon, SDCS mechanisms can achieve 95% utilization with a call-blocking probability of only 1%, while with an accept-or-reject scheme the call blocking probability can be as high as 23% with a utilization of only 80% when the per-circuit rate is one-tenth of link capacity.

Current deployments and applications using SDCS.

In the US, Internet2 and DOE's ESnet have deployed separate SDCS networks complementing their IP-routed networks. Scientists use SDCS to (i) *transfer files* of tera-to-peta-byte sizes, and (ii) *co-schedule network resources* with one or more of high-performance computing, storage, visualization resources, and high-end scientific instruments. Both ESnet and Internet2 deploy external software schedulers, which process requests for circuits, and then communicate with their switches to configure the (virtual-)circuits at the scheduled start time. Since a centralized scheduler handles all requests, there is no need for an intra-domain routing protocol. However, the need for a full-fledged inter-domain routing protocol to support SDCS has been identified as a problem by this community [3].

Commercial providers, such as AT&T and Verizon, are also offering their customers a scheduled dynamic circuit service, though not under this name. Verizon's service is referred to as *bandwidth on demand (BOD)* [4]. AT&T's Optical Mesh Service (OMS) is also an SDCS [2]. Example applications listed on the OMS web site [2] include disaster recovery, video on demand, FTP file transfers/bulk transfers, backbone network capacity adjustment and bandwidth on demand. Neither Verizon or AT&T currently support interdomain SDCS.

Scope for growth.

SDCS will initially continue its draw on applications that are currently served by leased lines. For example,

telepresence, telehealth, telesurgery, video-conferencing and distance-learning applications are often deployed on high-cost leased lines. The above applications all typically have some co-scheduling needs, making them well suited to SDCS. Moderate-sized businesses that cannot afford the high costs of leased line service will now be able to deploy these applications using the lower-cost SDCS, thus leading to economic growth. Furthermore, cloud computing providers, such as Amazon Web Services, recently started offering customers the opportunity to reserve capacity in advance on their infrastructure [1]. SDCS provide means to couple allocation of resources on the servers with allocation of network resources between a business and its cloud provider.

Related work.

A great deal of research on SDCS has been conducted, often under the headings of *advance-reservation* and *book-ahead* services [12, 21, 13, 20, 17, 14, 6, 15, 5, 18, 8, 11, 9, 10] However, no previous work proposes routing protocols to support SDCS. Instead the focus is primarily on scheduling algorithms. While the problem of route selection during circuit scheduling has been studied, that of reporting available bandwidth information as a function of time without revealing network topologies has not been addressed.

QoS routing solutions were developed for RSVP signaled IP networks. An excellent review paper [7] surveys several schemes. For example, QoS extensions to BGP were proposed in [22]. The tradeoff between routing optimality and routing message overhead was considered. Some ideas from this work can definitely be reused in designing routing protocols for scheduled dynamic circuit service. However, the dimension of time, with schedulers maintaining some type of available bandwidth information for the reservation window, adds a significant degree of complexity.

3. ROUTING ARCHITECTURE

3.1 Design goals

A routing protocol for scheduled dynamic circuit services should satisfy several requirements:

- It should support a variety of configurations, e.g., one or multiple schedulers per domain.
- It should support high rate-duration product circuits.
- It should support both earliest-start time requests and user-specified start time requests.
- It should support reservation for one or multiple *time slots* over a certain *reservation window*, where a time slot is defined to be the minimum time period for which a reservation can be made and a reservation window is the complete list of time slots available.
- It should achieve high network utilization on the data plane, e.g., strive to minimize delay for earliest-start time requests and blocking probability for user-specified start time requests.
- It should minimize control-plane overhead, i.e., the amount of control information stored, processed, and communicated by network nodes.

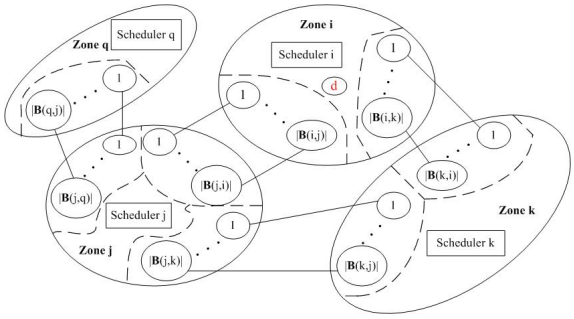

Figure 2: An example multi-zone topology

3.2 Scheduled Circuit Routing Protocol (SCRP): A Strawman

We present an initial approach to the design of SCRP. The description is by no means complete. Several open issues for future research are specifically identified in the following subsections. Our goal with this description is, thus, to provide a strawman to emphasize the issues that need to be addressed. For example, available bandwidth information needs to be aggregated across time slots before dissemination. If each time slot is one minute long, and the advance reservation window is one week long, then the total number of time slots exceeds 10000. Sending information for each of these time slots for all reachable destination subnets would lead to unacceptable message transmission and processing overhead.

SCRP architecture.

Fig. 2 illustrates the architecture, showing four *zones*, where a zone consists of a set of routers and a single scheduler. Internal routers and hosts within zones are omitted for clarity except for a single representative destination d, which will be referred to later. A zone could be an entire administrative domain or a group of routers within a domain. For instance, in the current REN deployment there is one scheduler per domain. At the other extreme, a scheduler could be built into each router controller. In this case, a zone could consist of a single router.

SCRP is a routing protocol that runs between zone schedulers. Each scheduler i has a set of neighboring schedulers, $\mathbf{N}(i)$. Two schedulers i and j are neighbors

if their respective zones have links between their border routers. $\mathbf{B}(i,j)$ is a set of border routers in zone i that connects to one or more routers in the set $\mathbf{B}(j,i)$ in zone j. $\mathbf{D}_i$ represents the set of destinations (endpoints) within zone i.

For simplicity, we make the following assumptions in the description of the protocol: (i) the time-slot duration is the same in all zones, (ii) the number of time slots in the reservation window, K, is the same in all zones, and (iii) the minimum circuit rate is the same in all zones. Relaxing these assumptions is an important area of future work in the design of SCRP. For instance, an important open issue is in determining the right level of granularity for the length of time slots. A too fine granularity may result in high protocol overhead while a too coarse one may result in performance degradation.

In such a zone-based architecture, consider what the schedulers would need to report in a simple reachability based routing protocol. For each destination that can be reached by a scheduler, it would need to report the "best" border router within its zone for use by its neighbor. For example, ESnet and Internet2 are connected at five border "crossings," i.e., at Los Angeles, Seattle, Chicago, New York and Washington DC. Therefore the ESnet scheduler should advise the Internet2 scheduler on the "best" border router to use for each destination that it can reach. Add to this the dimension of time-range denoted available bandwidth, and that serves as the basic starting point for SCRP.

Per-link information maintained by a scheduler for all links within its zone.

The first question to be addressed is what data structures need to be maintained at the schedulers. In our strawman protocol, we consider *time ranges* each of which represent a contiguous set of time slots during which the available bandwidth remains unchanged. The total number of time ranges is a variable $K' < K$. We denote $\mathbf{T}_{lk} = (t_{lk}^{start}, t_{lk}^{end})$ as the k-th time range for link l, where $1 \leq k \leq K'$ and t_{lk}^{start} and t_{lk}^{end} respectively denote the times at which a time range starts and ends. We denote A_{lk} as the available bandwidth on link l during the k-th time range, τ to be the length of a time slot, and t_ρ to be the starting time instant of the reservation window. Then, for each link l in its zone, a scheduler holds the following information at the starting time instant of the reservation window $(t_\rho, t_\rho + K\tau)$:

$$\mathbf{V}_l(t_\rho) \triangleq \{(A_{l1}, \mathbf{T}_{l1}), (A_{l2}, \mathbf{T}_{l2}), \ldots, (A_{lK'}, \mathbf{T}_{lK'})\}. \quad (1)$$

Fig. 3 illustrates this metric. For example, 16 bandwidth units is the available bandwidth on link 1 during the time range $\mathbf{T}_{12}$.

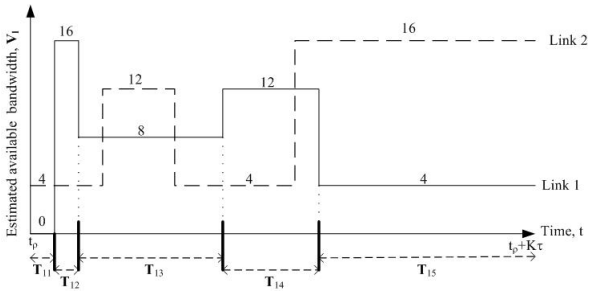

Figure 3: An example of available bandwidth for two links

Join operations executed by scheduler i to determine available bandwidth on paths from its border routers to destinations in its own zone.

Given $\mathbf{V}_l(t_\rho)$, one can readily compute a similar metric $\mathbf{W}_\mathbf{p}(t_\rho)$ for each path $\mathbf{p} \in \mathbf{P}$ where $\mathbf{P}$ is the set of paths from each border router within its zone to each of its own destinations. There could be multiple paths between any given border router and given destination. The complete set of paths could be very large, and therefore some technique needs to be developed to limit the number of paths considered or determine the "best" ones. This is yet another important area for future work.

The available bandwidth on path $\mathbf{p}$ at time t_ρ is given by:

$$\mathbf{W}_\mathbf{p}(t_\rho) \triangleq \{(A_{\mathbf{p}1}, \mathbf{T}_{\mathbf{p}1}), \ldots, (A_{\mathbf{p}K''}, \mathbf{T}_{\mathbf{p}K''})\}, \quad \text{where } K'' < K. \quad (2)$$

We use a join operator $\oplus$ to combine the $V_l(t_\rho)$ vectors defined in Eq. (1) for all links on a path [22]. This operator computes the intersection of the time ranges reported by each link, and for each such intersection returns the minimum bandwidth:

$$\mathbf{W}_\mathbf{p}(t_\rho) = \mathbf{V}_1(t_\rho) \oplus \mathbf{V}_2(t_\rho) \oplus \cdots \oplus \mathbf{V}_{|\mathbf{p}|}(t_\rho). \quad (3)$$

As an example, consider a path composed of link 1 and link 2 whose $\mathbf{V}$ vectors are as shown in Fig. 3. Then the first time range of the intersection would be the same length as $\mathbf{T}_{11}$ and the available bandwidth is 0. Then, the second time range for the intersection would be the same length as $\mathbf{T}_{12}$ and the available bandwidth is 4, and so on.

Computation of routing updates to neighbor schedulers about own destinations.

In a routing update to a neighboring scheduler, for each of its own destinations, a scheduler should recommend the "best" border router to use for each time range, and provide the available bandwidth on the path from that border router to the destination. The available bandwidth information is required to allow the neighboring scheduler to select its "best" choice from

routing updates it receives from all its neighbors.

Consider a routing update sent by scheduler i to each of its neighbors $j \in \mathbf{N}(i)$. The question is how to construct a routing update vector $\mathbf{U}_{ij}(d, t_\rho)$ by combining time the different vectors $\mathbf{W}(t_\rho)$ from the different border routers to destination d in its zone. The routing update $\mathbf{U}_{ij}(d, t_\rho)$ sent from scheduler i to scheduler j about destination d at time t_ρ is as follows:

$$\mathbf{U}_{ij}(d, t_\rho) \triangleq \{(A_{ij1}, \mathbf{T}_{ij1}, B_{ij1}), (A_{ij2}, \mathbf{T}_{ij2}, B_{ij2}), \ldots, (A_{ijK'''}, \mathbf{T}_{ijK'''}, B_{ijK'''})\} \quad (4)$$

where $j \in N(i), d \in D_i$, $B_{ijk} \in \mathbf{B}(i, j)$ and $K''' < K$. The structure of the vector $\mathbf{U}$ is similar to that of the vectors $\mathbf{V}$ and $\mathbf{W}$, with the exception of an additional element in each time range identifying the appropriate border router. Designing an algorithm to determine these routing updates forms a key research challenge. For certain topologies, we may be able to prove their optimality or at least come up with some performance guarantees.

As an example of the routing update computation problem, consider the available bandwidth graphs shown in Fig. 3. For the following discussion, we relabel the graphs as follows. Instead of representing available bandwidth for two links, assume that these same two graphs represent available bandwidth on two paths, each from a different border router in set $\mathbf{B}(i, j)$ but to the same destination d within zone i. We denote these border routers B_1 and B_2. Let **p1** and **p2** be the paths from border routers B_1 and B_2, respectively, to destination d. In other words, the graphs shown for Link 1 and Link 2 in Fig. 3 are relabeled as $\mathbf{W_{p1}}(t_\rho)$ and $\mathbf{W_{p2}}(t_\rho)$, respectively.

From Fig. 3, we see that path **p1** to the destination has no available bandwidth (shown as 0) until t_{11}^{end}; however after this point in time, it has a large available bandwidth (16 units). On the other hand, path **p2** does have 4 bandwidth units available immediately. For calls that require 4 bandwidth units or less, and that arrive at scheduler j destined to d soon after the routing update sent at t_ρ is received, it would be useful for scheduler j to have this availability information. Fig. 3 also shows that the time range for which 4 units are available on path **p2** is longer than the time range for which B_1 offers a path with 16 bandwidth units. Which should be reported?

One possible answer is both. This would allow for calls with bandwidth requirements higher than 4 to be accommodated with a path to border router B_1 after time instant t_{11}^{end}, while at the same time, it provides the 4-or-less bandwidth unit calls an earlier start time by being routed to border router B_2. It may be acceptable to have overlapping time ranges with different border routers reported as the "best" option because of the available bandwidth consideration. Thus, an important research topic is to come up with efficient solutions for border router selection with acceptable memory needs and message overhead. The next section proposes a possible approach.

Path switching.

Significant performance gains may be achievable if one allows *path switching*, i.e., switching between different paths throughout the duration of a connection. This new concept, in the context of SDCS, was introduced in [8, 10]. We explain the concept using Fig. 3. Assume a call requests 12 bandwidth units for a start time t within time range $\mathbf{T}_{14}$, but its duration is such that it will not be completed before the bandwidth on the path from border router B_1 drops to 4 bandwidth units (because of a prior reservation). With path switching, the circuit can be switched to the path through border router B_2 at t_{14}^{end} since this path has 12 units after this time instant.

As circuits will be held for moderate durations, path switching may be feasible to implement. For example, if circuit durations are 30 minutes, and circuit provisioning, based on our prior studies [24], is on the order of 1 second, it is acceptable to signal a change of circuit to applications and to re-provision the circuit on a new path.

Per-neighbor distance and routing tables computation at scheduler j.

Upon receiving routing updates for destination d from its neighboring scheduler i, scheduler j updates its distance table for destination d corresponding to this neighbor. Scheduler j creates a routing table for all reported destinations to serve calls originating in its own zone. This computation creates a vector listing a best border router toward which to route a new call for different sets of time ranges, with corresponding available bandwidth information for all destinations. The best border router is selected from across the best one (or best ones in the case of overlap) reported by each neighbor.

Finally, scheduler j has to create separate routing updates to send to its neighbors for all destinations that it holds in its routing table (not just its own destinations). As these updates are customized for each neighbor as described in the previous step, scheduler j must hold these routing updates in a routing table, so that if there are significant changes between the periodic routing update instants, it can send a triggered update. Thus, unlike BGP, in which each router has just one routing table, in SCRP, a scheduler maintains as many routing tables as neighbors plus one for its own zone.

4. SUMMARY

In this paper, we describe a strawman protocol, called SCRP, to enable inter-domain routing for SDCS ser-

vices. The *key ideas* in this design are: (i) information is reported for different time ranges, where a time range compresses information pertaining to multiple contiguous time slots, (ii) different border routers are identified for different time ranges, possibly overlapping, for each destination, and (iii) centralized, distributed and hybrid designs for scheduler deployment within domains are supported. The *key challenges* are: (i) finding a subset of paths from border routers to destinations and executing the join operation efficiently (vector **W**), (ii) computing the routing update vectors **U**, and (iii) generalizing the solution to allow for different domains to choose their own time-slot duration, minimum circuit rate and reservation window size. The paper identifies several key architectural issues, such as determining appropriate time slot granularity and providing support for path switching. Properly addressing these issues is key for the successful deployment of SDCS services and opens interesting avenues for further research in this area.

5. REFERENCES

[1] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/.

[2] AT&T Optical Mesh Service. http://www.business.att.com/.

[3] DICE InterDomain Controller Protocol (IDCP). http://hpn.east.isi.edu/dice-idcp/.

[4] Verizon Bandwidth on Demand (BOD). http://www22.verizon.com/.

[5] M. Andrews, A. Fernandez, A. Goel, and L. Zhang. Source routing and scheduling in packet networks. *Journal of the ACM (JACM)*, **52**(4):582–601, July 2005.

[6] L.-O. Burchard. On the performance of computer networks with advance reservation mechanisms. In *Proc. of the 11th International Conference on Networks*, Sydney, Australia, Sep. 2003.

[7] S. Chen and K. Nahrstedt. An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions. *IEEE Network*, 12(6):64–79, 1998.

[8] R. Cohen, N. Fazlollahi, and D. Starobinski. Graded Channel Reservation with Path Switching in Ultra High Capacity Networks. In *Proc. IEEE Gridnets'06*, October 2006. San-Jose, CA.

[9] R. Cohen, N. Fazlollahi, and D. Starobinski. Competitive Advance Reservation with Bounded Path Dispersion. In *Proc. of the IEEE High-Speed Networks Workshop*, April 2008.

[10] R. Cohen, N. Fazlollahi, and D. Starobinski. Path Switching and Grading Algorithms for Advanced Channel Reservation Architectures. *IEEE/ACM Transactions on Networking*, 17(5):1684–1695, October 2009.

[11] N. Fazlollahi, R. Cohen, and D. Starobinski. On the Capacity Limits of Advanced Channel Reservation Architectures. In *Proc. of the IEEE High-Speed Networks Workshop*, May 2007.

[12] D. Ferrari, A. Gupta, and G. Ventre. Distributed advance reservation of real-time connections. *IMultimedia Systems'97*, **5**(3):187–198, May 1997.

[13] A. G. Greenberg, R. Srikant, and W. Whitt. Resource sharing for book-ahead and instantaneous-request calls. *IEEE/ACM Trans. Netw.*, 7(1):10–22, 1999.

[14] R. Guérin and A. Orda. Networks with advance reservations: The routing perspective. In *Proc. of IEEE INFOCOM*, Tel Aviv, Israel, 2000.

[15] H. Lee, M. Veeraraghavan, H. Li, and E. P. Chong. Lambda Scheduling Algorithm for File Transfers on High-speed Optical Circuit. In *IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2004)*, April 2004. Chicago, USA.

[16] B. Metcalfe. Metcalfe's law: A network becomes more valuable as it reaches more users. *Infoworld*, Oct. 1995.

[17] O. Schelén and S. Pink. Resource sharing in advance reservation agents. *Journal of High Speed Networks*, 7(3-4):213–228, 1998.

[18] A. Schill, S. Kuhn, and F. Breiter. Resource Reservation in Advance in Heterogeneous Networks with Partial ATM Infrastructures. In *Proceedings of INFOCOM'97*, April 1997. Kobe, Japan.

[19] M. Veeraraghavan, M. Karol, and G. Clapp. Optical dynamic circuit services. *accepted for publication in IEEE Communications Magazine*, 2010.

[20] J. T. Virtamo. A model of reservation systems. *IEEE Transactions on Communications*, 40:109–118, 1992.

[21] L. C. Wolf and R. Steinmetz. Concepts for resource reservation in advance. *Multimedia Tools Appl.*, 4(3):255–278, 1997.

[22] L. Xiao, K.-S. Lui, J. Wang, and K. Nahrstedt. QoS Extension to BGP. In *IEEE International Conference on Network Protocols*, pages 100–109, Los Alamitos, CA, USA, 2002.

[23] X. Zhu and M. Veeraraghavan. Analysis and design of book-ahead bandwidth-sharing mechanisms. *IEEE Transactions on Communications*, 56(12):2156–2165, Dec. 2008.

[24] X. Zhu, X. Zheng, and M. Veeraraghavan. Experiences in implementing an experimental wide-area GMPLS networks. *IEEE Journal on Selected Areas in Communication*, 25(3):82–92, April 2007.