Appears in *Proceedings of the Allerton Conference on Communication, Control and Computing*, 2003

Lightweight Flow Setup for Wirespeed Resource Reservation

Fred Kuhns, Jonathan Turner[1]
Computer Science and Engineering Department
Washington University in St. Louis
fredk@arl.wustl.edu, Jon.Turner@wustl.edu

Samphel Norden
Lucent Technologies
norden@lucent.com

Abstract

Lightweight flow setup (LFS) is a proposed reservation mechanism for use in the Internet that requires no complex signaling protocols and is simple enough to be implemented in the datapath of high performance routers. It can provide significant benefits even when partially deployed and incorporates mechanisms to support usage accounting, enabling network service providers to offer it as a value-added service, providing an incentive for commercial deployment. This paper provides a detailed description of the LFS protocol and an evaluation of its performance.

Index terms – resource reservation, flow setup

1. Introduction

Resource reservation is widely recognized as an essential requirement for applications such as interactive voice and video, which require a guaranteed minimum bandwidth and low delay, in order to provide a consistent high quality service to the end user. Because of the inherently unregulated nature of datagram traffic, it is not possible to make any strong performance guarantees in a network, based on the datagram service alone. The increasingly critical nature of networked applications and the rising expectations of network users make it important for applications to have a way of obtaining effective performance guarantees.

The conventional way to obtain performance guarantees in networks is to use a signaling protocol to reserve capacity from end-to-end for particular application data flows. Substantial efforts have been made toward developing and standardizing suitable signaling protocols for the Internet [ZH93,BR97], but no widespread deployment has taken place, and at this point, expectations for deployment remain low. There seem to be several reasons for the failure of signaling protocols to gain traction in the Internet to date.

- *Complexity*. Signaling protocols are widely perceived as complex, requiring substantial investment in developing and maintaining signaling software and substantial processing resources in routers.
- *Need for universal deployment*. Signaling protocols require software changes in all network routers and attached hosts. In the absence of widespread industry consensus on proceeding with deployment, it is difficult to get this done.
- *Insufficient business motivation*. Network providers have little motivation to support signaling protocols, since there is no compelling near term business reason to do so. This is partly because of the chicken-and-egg problem caused by the need for universal deployment and partly because Internet signaling protocols have not been designed with due consideration for usage regulation and accounting.

---

[1] This work supported by the Defense Advanced Research Projects Agency, Contract #N660001-01-1-8930.

These observations suggest that any new attempt to develop resource reservations for the Internet should be simple, incrementally deployable and should provide new opportunities for network operators to generate revenue. We propose a Lightweight Flow Setup service (LFS) to augment the Internet's datagram service. The proposed service requires no elaborate signaling protocol and is simple enough to be implemented largely within the router data path. It can be incrementally deployed and provides network operators with new business opportunities, giving them an effective motivation for deployment.

The proposed LFS protocol is far simpler than conventional protocols, such as RSVP [ZH93, BR97]. LFS focuses on one-way, unicast reservations. While this makes it less general than RSVP (which is designed to support general multicast applications), we believe these restrictions can allow LFS to overcome the obstacles that have prevented wide-spread deployment of RSVP. Reference [AL97] describes another approach to resource reservation for the Internet. Their method aggregates information about reserved flows and relies on hosts to constrain themselves to the rate allocated by the network, which the receiver estimates based on its observation of the incoming data rate. LFS bears some resemblance to fast reservation protocols developed in the early nineties to make ATM networks more suitable for bursty data traffic by enabling end-systems to make dynamic bandwidth reservations on *pre-established* virtual circuits [BO90,TU92]. This work was effectively subsumed by the subsequent development of the ABR explicit rate control protocol [JA96,JK96]. There have also been proposals that attempt to perform on-the-fly virtual circuit setup for data traffic [HJ98, BI98], but these proposals have never been put into practice, in part because of the fading interest in ATM technology in recent years. The proposed lightweight flow setup service can be viewed as a re-casting of such proposals in the IP context. It exploits conventional IP forwarding and flow classification mechanisms for making routing decisions and associating packets with flow state. The proposed protocol is similar to *Yessir* [PA99] in its overall character and objectives, but is simplified to allow the core reservation mechanisms to be implemented directly within the router data path.

Section 2 of the paper provides an overview of the proposed LFS service. Section 3 discusses issues relating to partial deployment. Section 4 provides details of an IP version 4 implementation. Section 5 presents an analysis of the performance of LFS and provides numerical results, demonstrating that the protocol can perform well under expected operating conditions.

2. Overview

The proposed *Lightweight Flow Setup* (LFS) protocol supports *sender-initiated* bandwidth reservations for *one-way, unicast* flows. The limited objectives highlighted in italics were chosen intentionally to avoid the complexities of typical signaling protocols and to facilitate incremental deployment. There are certainly many other useful features that a flow setup mechanism might offer. The focus of LFS is on providing a minimal set of mechanisms that can serve a large fraction of application needs, while avoiding the obstacles to deployment that have stymied efforts to realize more comprehensive approaches. We believe that once a basic service is deployed and its value proven, the proposed mechanisms can serve as useful building blocks for more elaborate network services.

2.1. Basic Service Model

LFS uses a soft-state protocol that responds to the transmission of packets between endpoints, rather than explicit flow setup messages. In order for the application to specify the desired service level, some or all of the data packets must contain an embedded LFS reservation request. The arrival of the "first" packet of a flow at a router triggers the creation of per flow state at the router, the selection of a route to the desired destination and the reservation of bandwidth for the flow. Subsequent packets belonging to the flow follow the same path.

The *basic service* requires routers along the path to the destination to allocate the requested rate if and when sufficient bandwidth is available. Once bandwidth is allocated at a router, it is reserved for the flow until released (either explicitly or implicitly). If, when a reservation request arrives at a router, there is insufficient bandwidth to satisfy the request, a partial reservation (possibly zero) is made and traffic in excess of the actual allocation is forwarded on a *best-effort* basis. The reservation

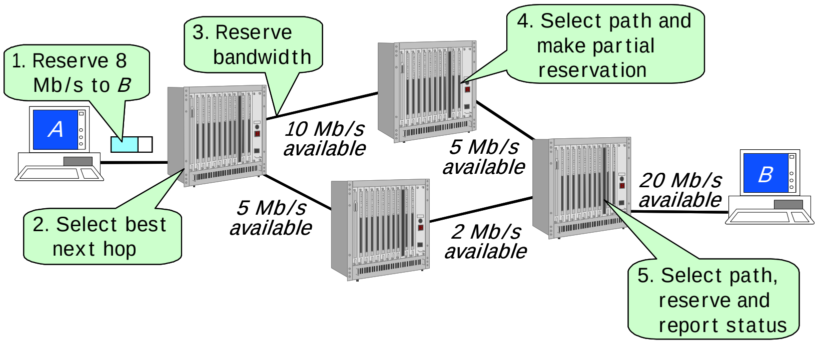

Figure 1: Example of LFS Operation

request is retained in a *pending reservation queue* at each router where the request fails, allowing it to complete the request as soon as resources become available (as other reserved flows terminate and release their resources). This allows forwarding of application traffic, while waiting for reserved resources to become available. Applications that cannot usefully operate in this mode may choose to abandon the reservation attempt.

Users may request a *status report* from the last LFS router on the path. The report indicates how much bandwidth was allocated on the path and includes the IP address of the reporting router. Each router on the path operates independently, making a purely local decision on the rate reservation. In well-engineered networks, the vast majority of reservation requests will be established end-to-end by the first packet containing a reservation request. Reservations that do not succeed initially will typically do so within a short time period, following the initial reservation request. LFS reservations are maintained using soft state, meaning that reservations must be periodically confirmed by the transmission of reservation requests. If not confirmed, reservations timeout and all associated resources are released. For most efficient operation, hosts should explicitly release reservations when they no longer require them, but this is not essential.

The basic service allows an end system to reserve a fixed bandwidth for a flow and maintain that reservation for as long as required. In fact, a flow can adjust its reservation up or down at any time, giving applications considerable flexibility. Of course, increases in reserved bandwidth may not be satisfied immediately, if any of the links on the flow's route are congested (once a route for a flow has been established, no re-routing is done).

The flow setup service requires no global coordination of reservations. Each router makes local decisions only, making the protocol simple to implement and allowing everything to be done in the router data path (possibly in hardware). Also, note that the service requires no explicit participation by the destination host. This makes it possible to usefully deploy the service, even in a network where only a fraction of hosts are LFS-enabled. For example, a web server that provides streaming video can use LFS to request a guaranteed bandwidth for its video transfers, allowing it to provide high quality video delivery to its users, without the long buffering delays now generally required. All the users of the service benefit from this, even if none of them has any built-in support for LFS.

An example of LFS operation is shown in Figure 1. The numbers in the comment boxes indicate the sequence of the different events. Here, the sending host (*A*) starts sending a stream of data to the receiver (*B*) with embedded reservation requests. When the first router receives the first packet containing a reservation request, it selects an outgoing link to forward the request on and reserves the required bandwidth on the outgoing link. The selection of the outgoing link can be made using the standard datagram forwarding mechanisms, although better performance can be obtained using a separate forwarding table that selects paths based on available bandwidth. At the second router in the path, the outgoing link to the destination lacks sufficient bandwidth to fully accommodate the request. In this situation, the router makes a partial reservation and forwards packets along the path based on the partial reservation. Packets in excess of the reserved rate are forwarded on a best effort basis. The last router on the path selects a path to the destination, makes the required reservation and generates a

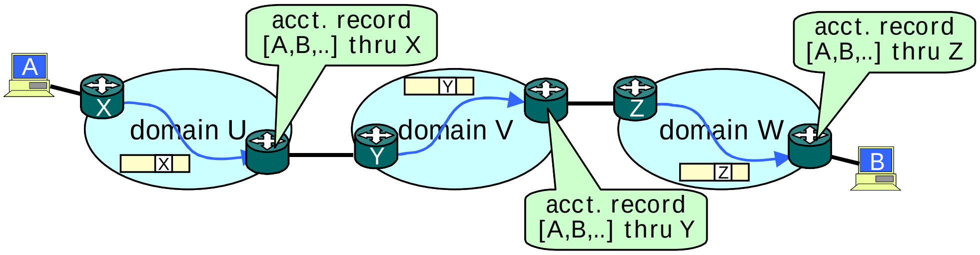

Figure 2. Use of Trace Field

status packet, indicating that 5 Mb/s of the requested 8 Mb/s has been reserved. The status packet is sent back to the originating host, *A*. Subsequent packets belonging to the flow (the flow is identified by the five-tuple consisting of the source and destination addresses, port numbers and the protocol field) are forwarded along the selected path. When more bandwidth becomes available on the link joining the second and third routers on the path, this bandwidth is allocated to the flow. The new allocated value is reported in subsequent status packets.

2.2. Soft Reservations

The mechanisms needed to support bandwidth reservations can be usefully extended to provide improved congestion control for traditional Internet applications that are capable of adjusting their bandwidth use to network load. LFS provides this capability through *soft rate reservations*. The resource reservations discussed in the preceding section, are referred to as *firm rate reservations* to distinguish them from soft rate reservations. Soft reservations are qualitatively different from firm reservations, in that the network may adjust a flow's soft reservation in response to changing traffic conditions, while a firm reservation, once established can only be released when the user is done with it. Flows may have both a firm rate reservation and a soft rate reservation. In this case, the network may not reduce a flow's soft rate below its firm rate. Flows that do not request a firm rate reservation are treated as having a firm rate reservation of 64 Kb/s (this is the smallest firm rate reservation supported by the protocol). Flows that do not request a soft rate reservation are treated as having a soft rate equal to their firm rate.

When the network gets a packet requesting a soft rate larger than a flow's firm rate, it attempts to allocate the requested bandwidth at each hop along the path. If there is contention for bandwidth at a link, the router allocates the bandwidth among all flows requesting a soft rate larger than their firm rate. As with firm rate reservations, applications may request status reports from the far end access router. This enables the application to know what end-to-end rate the network is currently providing. To monitor changes in its rate allocation, an application should continue to send reservation requests, with a frequency of at least one per network round-trip time. Each router forwards packets in accordance with each flow's current rate allocation. If a host sends packets at a higher rate than has been allocated by some intermediate router, that router will queue the excess packets in the presence of link congestion and will preferentially discard those packets, should it run out of buffer space.

2.3. Status Reporting

As outlined previously, all routers along a reserved flow's path must create and maintain per flow state for the duration of the reservation. If a router is not able to allocate the requested rate, then a field in the request is updated to indicate the actual allocation. As the request propagates, each transit router will in turn, allocate the requested rate or update the request to indicate the minimum allocation along the flow's path.

When a packet carrying the reservation request arrives at the last LFS router in the path, this router updates a status record for the flow. If a user wishes to be informed of the reservation status, a special flag in the reservation request is set and this router sends an end-to-end reservation status report directly to the sending user. When a reservation is released, the corresponding record is closed and optionally stored for later use.

To enable usage monitoring and accounting, each administrative domain, along the path taken by an LFS flow must record some basic information about each LFS flow that passes through the domain. Each reservation request includes a *trace field*, to facilitate such usage accounting. Each domain is free to use the trace field as it chooses. The typical use is to tag the packet with the identity of the ingress router interface at which the flow entered the domain. This information can then be recorded by the last router within the boundary domain, in a domain-specific accounting record. This is illustrated in Figure 2. Note that if usage monitoring is done at every router along the path, the trace field is not really needed. It is provided simply to reduce the amount of usage information that must be recorded.

The LFS protocol includes additional, optional status reporting mechanisms. These support reporting to routers within any single domain, reporting across the public network portion of the path and end-to-end reporting for private networks at the ends of a path. The use of these reporting mechanisms is optional and they incur overhead only when used. Details appear in Section 4.

3. Partial Deployment

LFS can be used strictly by a single network provider, or end-to-end through multiple intermediate administrative domains. As indicated earlier, the design of LFS makes it useful even when only partially deployed. We believe that significant benefits can be derived from deployment of LFS by just a few large ISPs and key information service providers. Using LFS, an information provider can obtain reservations that reach the local network of any customer of the large ISPs that the information provider connects to. Because bandwidth is typically plentiful within local networks, such a reservation delivers most of the benefit, without requiring the upgrade of large numbers of local networks and hosts. Because the large information providers reach lots of users, there is a potential to have a large impact on many users, through the actions of a small number of organizations and at a relatively low cost.

To enable the use of LFS when the protocol is partially deployed, we need some rules for how to use the protocol when only a subset of the network domains on the path from a sender to a receiver implements the protocol. The case of a path with an arbitrary mixture of LFS and non-LFS networks is difficult to handle in a general way, if we require reservations in each of the LFS segments on the path. There are two fundamental issues. First, a router within one LFS domain has no general way to know if there is another LFS domain along the path to the destination, making it difficult to implement the status reporting functions. Second, a non-LFS domain may route packets in the flow along different paths leading to the potential for “orphan reservations” in downstream LFS domains. To avoid these complexities, we do not attempt to setup reservations in all LFS domains along the path. Rather, we maintain a reservation only on the initial portion of the path that lies entirely within LFS domains. The router at the boundary between an LFS domain and a non-LFS domain effectively terminates the LFS reservation and is responsible for the reporting and accounting functions of the “last hop” router.

A strict interpretation of our partial deployment rule implies that the local network containing the sending host must be an LFS domain. Since bandwidth is typically plentiful within LANs and since LANs typically have just a single access point to each of their ISPs, we can reasonably relax the rule slightly, to allow use of LFS in situations where the sender’s LAN is a non-LFS domain. An ISP supporting LFS can act on LFS reservation requests received from a LAN, without encountering the complexities that make it difficult to handle the general partial deployment scenario. Also note that users whose ISPs do not support LFS can potentially get some of the benefits by using an IP tunnel to forward packets to another ISP that does support LFS.

4. Detailed Protocol Description for IPV4

This section provides a detailed description of the LFS protocol, suitable for operation in the IPv4 context. LFS reservation requests are embedded within application packets using an IP option. The format is illustrated in Figure 3 and the fields are described in detail, below.

- *Option Code* (8 bits). Identifies the LFS option,

- *Length* (8 bits). Specifies the option length in bytes.
- *Operation* (2 bits). Specifies a specific LFS operation. The operations are *Firm Rate Request*, *Soft Rate Request* and *Release***.** The *Release* operation causes a router to release all stored state for the flow. The reservation request operations are described below.
- *Flags* (6 bits). Four status request flags are defined. The *Host Status Request Flag* requests that a status report be sent to the host that sent the packet (identified by the source IP address). The report is sent by the router that terminates the LFS portion of the path. The *Client Network Status Request Flag* may be used by a LAN or enterprise network to obtain the status of the flow. The report is sent by the router that terminates the LFS section of the path, to the *Client Network Status Report Recipient*, whose IP address appears in the optional list of status report recipients. The *Public Network Status Request Flag* may be used by the first public LFS network on the path to request the status of the flow. The report is sent by the last public network router in the LFS segment of the path, to the *Public Network Status Report Recipient*, whose IP address appears in the optional list of status report recipients. The recipient of the public network status report forwards a copy of the status report along the path of the flow, so that all public networks along the path can obtain the status information. This report is removed by the last public network router in the LFS section of the path. The *Intra-domain Status Request Flag* may be used by the first router within a domain to request the status of a flow. The report is sent by the last router on the path within the domain, to the *Intra-Domain Status Report Recipient*, whose IP address appears in the optional list of status report recipients.

  In addition to the status request flags, there is a *Congestion Indication Flag*, which can be set by any router along the path to signal that it is congested and that the requested reservation is unlikely to be satisfied promptly. The value of the congestion indication flag is included in the status reports, allowing the requesting host to abandon the reservation request if it chooses to do so.
- *Requested Rate* (8 bits) and *Allocated Rate* (8 bits) fields. Rates are expressed using a simple floating point representation with a 4 bit mantissa and a 4 bit exponent. Specifically, if the value of the first four bits is m and the value of the last four bits is x, then the rate defined by the field is $m \cdot 2^x \times 64$ Kb/s. This allows reservation rates ranging from 64 Kb/s to over 4 Gb/s. Successive rates differ by no more than a factor of 1.0625. The *Requested Rate* field contains the desired rate while the *Allocated Rate* field contains the minimum rate allocated along the path.
- *Trace Field* (24 bits). This field is used to enable each domain to determine where a flow entered the domain. This enables it to monitor and account for usage. Each domain is free to define its own specific use of the trace field.
- *Status Report Target Recipients* (0-12 bytes). This field is an optional list of up to three status report recipients. Each is associated with one of the status request flags and is present if and only if the corresponding status request flag is present. The order in which they appear (if present) is *Client Network Status Recipient, Public Network Status Recipient, Intra-domain Status Recipient.*

When a router receives an IP packet with an embedded LFS reservation request it attempts to allocate the requested bandwidth. Both firm and soft rate requests specify the desired rate in the *Requested Rate* field. If a router cannot allocate the requested bandwidth, it indicates this by updating the *Allocated Rate* field to the minimum of the field's current value and the amount that the router reserved. The initial value of the *Allocated Rate* field should equal the value of the *Requested Rate* field. If the requested rate can be allocated, then the *Allocated Rate* field is not altered. When an LFS *Release* request is received at an LFS router, the corresponding flow state is removed and any allocated bandwidth is freed.

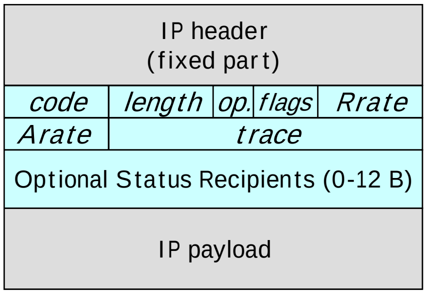

Figure 3. Reservation Request Format

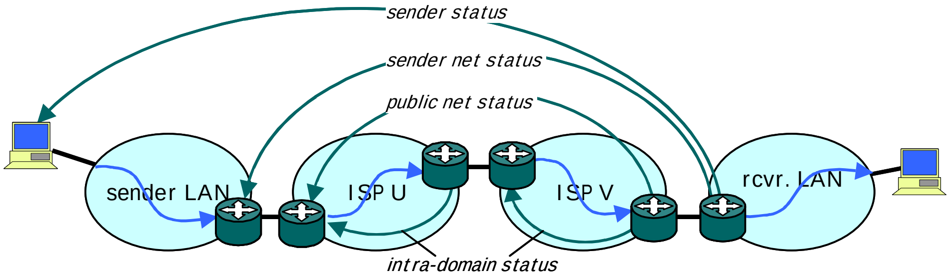

Figure 4. LFS Status Reporting

Figure 4 illustrates the use of LFS status reports. As indicated earlier, the last LFS router in the path sends the *Host Status Report* to the host that sent the packet originally (identified by the source address field in the packet). This router also sends the *Client Network Status Report* to the router identified in the optional *Client Network Status Recipient* field. This field is typically used to identify the gateway router connecting a client network (LAN) to a wide area network. This allows the client network to monitor the reservation status in order to verify that the organization is receiving its expected level of service. The public network status report is sent by the last LFS-capable router in the public network portion of the path. The report is sent to the designated *Public Network Status Recipient* (typically, the first public network router on the path), which forwards it back along the path, to inform all the public networks on the path of the status of the flow. The *Intra-domain Status Report* allows individual domains to verify the status within their domains and is sent by the last LFS router in the domain to the designated *Intra-domain Status Recipient* (typically, the first router in a domain). The public network and intra-domain status report features are provided primarily to allow network operators to deal with exceptional situations. The trace field enables flows to be monitored for accounting purposes and this is expected to be sufficient for normal network operation.

5. Performance of LFS

In the LFS protocol, each router forwards reservation requests and independently allocates the requested bandwidth, on its outgoing link *when it is available*. This raises the question of how long a sending host may have to wait for a reservation to be granted. For uniform reservation rates, the performance of LFS at a single link can be modeled by an *M/M/m* queue [KL75]. Such a queue can be analyzed using a birth-death process in which the transition rate from state k to state k+1 is λ, and the transition rate from state k to state k–1 is $m \min\{k,m\}$, where $1/\lambda$ is the average time between arrivals of new LFS sessions, $1/\mu$ is the average duration of a session, after its reservation is accepted, and m is the number of sessions that can share a link without exceeding the bandwidth of the link. The probability p_k that exactly k sessions either have a reservation or are waiting for their reservation to be accepted is

$$p_k = \begin{cases} p_0 (m\rho)^k / k! & k \le m \\ p_0 m^m \rho^k / m! & k \ge m \end{cases}$$

$$p_0 = \left[\frac{(m\rho)^m}{m!(1-\rho)} + \sum_{k=0}^{m-1} (m\rho)^k / k! \right]^{-1}$$

where $\rho = \lambda / m\mu$ [KL75]. A new reservation request arriving at a link in state $k>m$ must wait for $k-m+1$ sessions to complete before it is served. The waiting time is the sum of $k-m+1$ exponential random variables, all with a mean of $1/m\mu$. That is, it follows an *Erlang* distribution. Consequently, the distribution of the random variable d, representing the reservation delay, is given by

$$f_d(t) = \sum_{k=m}^{\infty} p_0 \frac{m^m \rho^k}{m!} \frac{m\mu (m\mu t)^{k-m}}{(k-m)!} e^{-m\mu t}$$

A straightforward calculation shows that

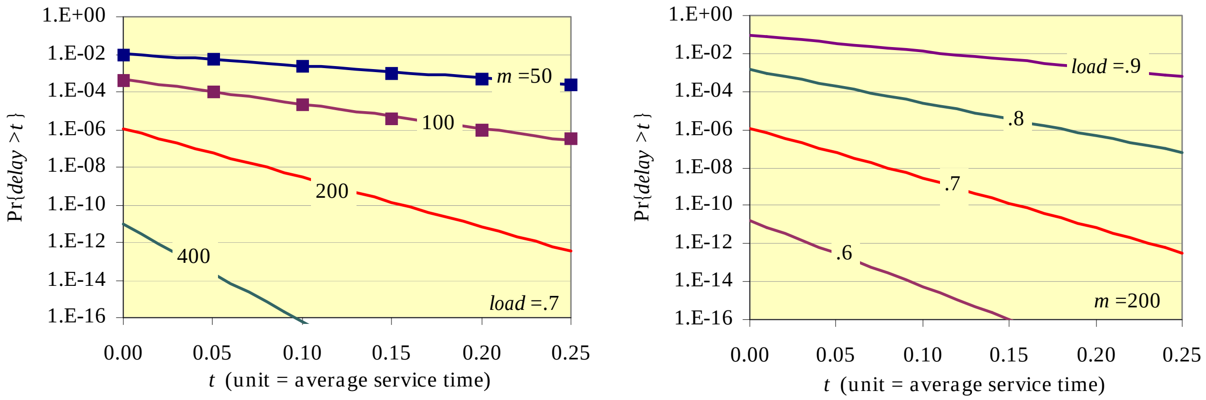

Figure 5. Tail of delay distribution for single LFS link.

$$f_d(t) = (p_0(m\rho)^m / m!)m\mu e^{-(1-\rho)m\mu t}$$

We can obtain the tail of the delay distribution by integrating this expression, giving.

$$\Pr\{d > t\} = \frac{p_0(m\rho)^m}{m!(1-\rho)} e^{-(1-\rho)m\mu t}$$

This analysis is conservative, in that it assumes that all sessions wait until their reservation is granted before beginning to transmit. This causes the analysis to over-estimate the delay. The analysis is for exponentially distributed session durations. When *m* is large (50 or more), we expect this to have little impact on the numerical value of the results. This expectation is confirmed below, by comparison of the analytical results with simulation results for Pareto-distributed session durations.

The tail of the delay distribution is plotted in Figure 5. The chart on the left shows the delay distribution for different values of *m* while the offered load is fixed at ρ=.7. Note that an OC-48 link can accommodate 240 reservations of 10 Mb/s each. We expect reservations of this magnitude to be near the high end of the range for typical applications, meaning that backbone links can accommodate hundreds or even thousands of typical reservations. Note that even for *m*=50, only about 1% of reservations requests experience any delay at all, when the offered load is 0.7. For *m*=200, roughly three out of one billion reservations experience a delay greater than 10% of the average session duration. For the *m*=50 and *m*=100 curves, simulation results are shown along with the analytical results (the simulation data is indicated by the solid squares). The simulation results are for Pareto-distributed session durations with a shape parameter of 1.5, giving infinite variance. The distribution of the session duration has no discernible impact on the delay distribution, as expected.

The chart on the right of Figure 5 shows the delay distribution for different values of the load, with *m* fixed at 200. For loads of up to 0.8, the delay remains quite small and even at a load of 0.9, less than 1% of reservation requests are delayed by more than about 12% of the average session duration.

A reservation that passes through several hops experiences a delay of $\leq t$, if and only if the delay at each link on the path is $\leq t$. If the delays at each stage are independent, then

$$\Pr\{d_h > t\} = 1 - (1 - \Pr\{d_1 > t\})^h = 1 - \left(\frac{p_0(m\rho)^m}{m!(1-\rho)} e^{-(1-\rho)m\mu t} \right)^h$$

where d_h is the random variable for the delay experienced by an LFS session that spans *h* hops. Note, that even when the delays are not independent,

$$\Pr\{d_h > t\} \leq h \Pr\{d_1 > t\} = h \frac{p_0(m\rho)^m}{m!(1-\rho)} e^{-(1-\rho)m\mu t}$$

Figure 6 shows the delay distribution for several different values of the hop count. For the *h*=16 case, the upper bound is also shown. For this case, the independence assumption makes a negligible difference in the delay distribution.

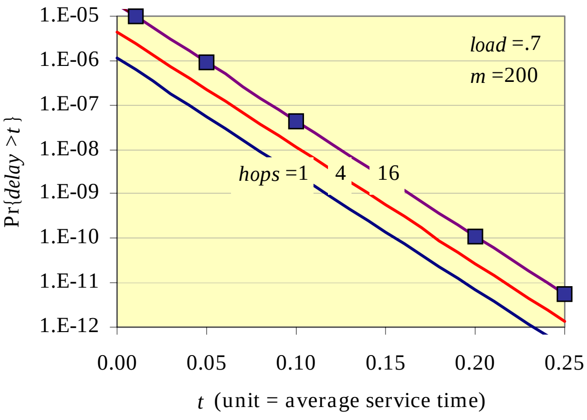

Figure 6. Delay Distribution for Multi-hop Paths

The above results show that LFS can perform well when the traffic load does not exceed certain limits. In a well-engineered network, one can expect traffic to stay within such limits, as a general rule. However, it's also important to understand how LFS behaves under more exceptional conditions. It's easy to see that under heavy load, the queue of pending requests can grow very long, meaning that few if any requests will get their requested reservation in an acceptable amount of time. Under overload conditions, LFS actually performs better with no queue at all. This is demonstrated by Figure 7. The chart at the left shows two sets of curves. The first set shows how much of the input traffic received at a link is granted a reservation within time 0.1/μ, when there is an unbounded queue for pending requests. We see that at heavy loads, none of the input traffic receives its reservation within the given time period. The second set of curves shows how much of the input traffic is granted a reservation when there is no queue for pending requests (requests that arrive when there is no bandwidth available are simply discarded). We see in this case, that even under overload conditions, most requests are satisfied.

The chart on the right in Figure 7 shows the same information, but expressed as the fraction of requests that are rejected (for the infinite buffer case, a request is considered rejected if the reservation is not satisfied within a time period of .1/μ. This chart quantifies the benefit provided by the buffer at lower loads. Small, finite buffers can provide intermediate performance, allowing one to trade-off performance under normal conditions against robustness during overload.

6. Closing Remarks

The LFS protocol was designed to provide a basic resource reservation mechanism for use in the Internet, to support applications that require consistent, high quality of service. It was designed with three objectives in mind.

- *Simplicity*. The protocol is simple enough to be implemented in the datapath of modern routers, and inexpensive enough so that implementation costs are not a serious impediment to service deployment.
- *Beneficial when partially deployed*. LFS does not require participation of the receiving end system, allowing it to be usefully deployed in asymmetric applications, such as web sites that

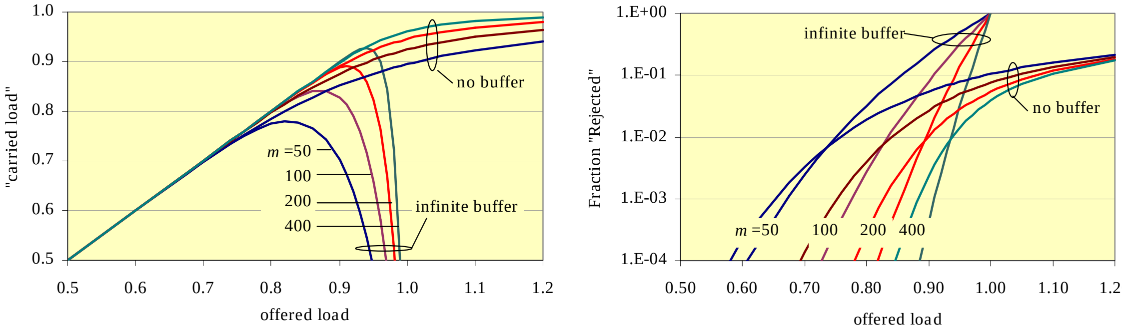

Figure 7. Comparison of LFS with and without Buffer for Pending Requests

transmit streaming media. If deployed by even a single large, national ISP, it can provide significant benefits to a substantial user population.

- *Strong business motivation.* LFS was designed to give network providers an incentive to move quickly toward deployment. The design includes mechanisms to enable the usage monitoring needed to support accounting for reservation-based services, enabling network providers to generate additional revenues, through their offering of the service.

LFS represents a significant departure from more complex signaling protocols such as RSVP. Limiting the protocol's objectives to sender-initiated, one-way, unicast flow reservation makes it possible to achieve the objectives listed above, while still providing the essential missing ingredient needed to effectively support high quality multimedia applications in the Internet.

References

[AL97] Almesberger, Werner, Tiziana Ferrari and Jean-Yves Le Boudec. "Scalable Resource Reservation for the Internet," INTERNET-DRAFT draft-almesberger-srp-01.txt, November 1997.

[BO90] Boyer, P. "A Congestion Control for the ATM." *International Teletraffic Congress Seminar on Broadband Technologies: Architectures, Applications and Performance,* 10/90.

[BI98] Bian, Q., K. Shiomoto and J. Turner. "Dynamic Flow Switching: A New Communication Service for ATM Networks," *Proceedings of Infocom*, 4/98.

[BE95] Bennett, J. and H. Zhang. "Worst-case Fair Weighted Fair Queueing," In *Proceedings of Infocom*, 1995.

[BR97] Braden, B., L. Zhang, S. Berson, S. Herzog and S Jamin. "Resource ReSerVation protocol (RSVP) – version 1 functional specification." RFC 2205, Internet Engineering Task Foirce, 10/97.

[CS92] Clark, D., S. Shenker and L. Zhang. "Supporting real-time applications in an integrated packet network: Architecture and mechanisms," *Proceedings of ACM SIGCOMM*, 8/92.

[HJ98] Hjálmtýsson, Gisli and K. K. Ramakrishnan. "UNITE - An Architecture for Lightweight Signalling in ATM Networks," in the *Proceedings of IEEE Infocom'98*, pp. 832-840, San Francisco, CA, March 1998.

[JA96] Jain, R. "Congestion Control and Traffic Mangement in ATM Networks: Recent Advances and a Survey," *Computer Networks and ISDN Systems*, 10/96.

[JK96] Jain, R., S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan. "ERICA Switch Algorithm: A Complete Description," ATM Forum/96-1172, 8/96.

[KL75] Kleinrock, Leonard. *Queueing Systems, Volume 1, Theory*, J. Wiley, 1975.

[KU02] Kuhns, Fred, John Dehart, Anshul Kantawala, Ralph Keller, John Lockwood, Prashanth Pappu, W. David Richard, David Taylor, Jyoti Parwatikar, Ed Spitznagel, Jon Turner and Ken Wong. "Design and Evaluation of a High Performance Dynamically Extensible Router." *Proceedings of the DARPA Active Networks Conference and Exposition*, 5/2002.

[PA99] Pan, P. and H. Schulzrinne, "YESSIR: A Simple Reservation Mechanism for the Internet," *Computer Communication Review*, Vol. 29, No. 2, April 1999.

[TU92] Turner, J. "Managing Bandwidth in ATM Networks with Bursty Traffic," *IEEE Network*, 9/92.

[WA97] Waldvogel, M., G. Varghese, J. Turner and B. Plattner. "Scalable High Speed IP Routing Lookups," In *Proceedings of SIGCOMM*, 1997.

[WO00] Wolf, Tilman and Jonathan Turner. "Design Issues for High Performance Active Routers," *Proceedings of the Zurich Conference on Digital Communication*, 2/00.

[ZH93] Zhang, L., S. Deering, D. Estrin, S. Shenker and D. Zappala. "RSVP: a new resource ReSerVation protocol," *IEEE Network*, 9/93.

[ZH91] Zhang, L. "Virtual Clock: a Net Traffic Control Algorithm for Packet Switched Networks," *ACM Trans. on Computer Systems*, 5/91.

[ZF94] Zhang., H., and D. Ferrari. "Rate-controlled service disciplines," *Journal of High Speed Networks*, 1994.