The Case for Service Overlays

J. Brassil, R. McGeer, P. Sharma, P. Yalagandula
Hewlett-Packard Laboratories

B. L. Mark, S. Zhang
George Mason University

S. Schwab
Sparta, Inc.

Abstract— **The Internet was designed as a packet-switched network in the 1960's and 1970's, with the explicit intent of sacrificing quality-of-service guarantees for an individual application in order to optimize channel usage and provide optimal median service for all applications. This approach was successful, since the application mix of the Internet heretofore has been dominated by applications with low quality-of-service needs: primarily bulk data transfer and low-bandwidth text-based interactive applications. As the Internet absorbs other networks and applications with strong quality-of-service requirements (television, voice over IP) , this tradeoff changes. We are faced with the problem of introducing the quality-of-service guarantees of circuit-switching into packet-switched networks. Fundamental change to the lower layers of the Internet stack have proven infeasible; even strongly-motivated, well-designed modifications which made transition a first-class design consideration have had difficult introductions. ATM and IPv6 are two recent examples. One effective transition strategy for new networking techniques has been the use of overlays. In this paper, we introduce the concept of a Service Overlay network, to offer circuit-switched behavior on legacy IP networks, and establish the requirements on the underlying IP network to make this strategy effective.**[1]

I. Introduction

The fundamental architecture of packet routed networks has remained constant over the course of a generation, while the underlying scales and speeds of the networks have each changed by six orders of magnitude, and a dramatic change in the application mix of networks. In particular, packet-switched networks were originally designed for two broad classes of application: command-line access to remote systems (telnet), and (conceptually) asynchronous bulk data transfer (ftp, smtp, pop). These applications were characterized by bursty traffic, and minimal quality-of-service (QoS) requirements for either bandwidth or latency. For this application mix, a strategy that sought to optimize network utilization at the expense of guaranteed service for any individual application was the correct design choice, and this was reflected in the design of the original ARPANET that has become today's Internet. This architecture featured best-effort routing with no guarantees of delivery, latency, or available bandwidth to an application.

It goes without saying that for the original design points of the Internet, its architecture was a stunning success. However, the success of the Internet has made its protocols and infrastructure a convenient choice for many services far beyond its original use cases. The design choices of the Internet serve these applications less well than the telnet- and ftp-like applications that motivated its initial design. In particular, the new application mix emerging for the Internet includes services with needs for strong guarantees on latency and bandwidth. These applications include: 1) Internet telephony, or Voice over IP (VOIP); 2) Internet television and interactive digital television; 3) Shared visualization of sensor or computational data; 4) High-bandwidth interactive applications, e.g., telesurgery, video teleconferencing.

All of these applications require guaranteed bandwidth and latency, over time scales in the sub-second range. Since the original design of the Internet cannot accommodate bandwidth or latency guarantees without global overhaul, a number of workarounds have been devised, including the use of dedicated, application-specific lines and massive overprovisioning. A third strategy, end-point signaling and control, has been adopted spottily. It is a centerpiece of the approach discussed in this paper, and so we defer its discussion until later.

II. Limitations of Packet Networks for QoS

Use of dedicated, application-specific lines is a strategy adopted when the application is sufficiently high-value and sufficiently used to justify the expense. One simple example is the HP teleconferencing product, HALO, which connects a network of conferencing rooms worldwide [1]. HP guarantees a natural conferencing experience featuring 3 HD feeds from room to room. In order to guarantee low latency, no jitter, and adequate bandwidth, HP has built a worldwide, OC-level (> 45 Mb/s) network just for the HALO product. Further examples of special-purpose networks are those built for regular high-speed bulk data transfers such as large-scale financial data dumps and very high-data-rate scientific experiments.

While these special-purpose networks are highly effective and efficient, there is an obvious drawback: they are expensive, and thus are only used for very high value applications. The large body of lower-value, episodic applications (which require high dedicated bandwidth and quality of service for intermittent periods) cannot be serviced by this method. Massive overprovisioning is used where the underlying network bandwidth is far in excess of that required for the application. The classic rule-of-thumb is that today's Internet requires about 10 times the theoretical minimum bandwidth to run services with specific bandwidth and latency requirements. To see why, consider voice-over-IP (VOIP).

[1]This work was supported in part by the Defense Advanced Research Projects Agency under Contract N66001-05-9-8904 (Internet Control Plane).

Voice services are highly dependent on predictable, low latency. Satellite phone's chief user complaint was that round-trip times of 250 ms or so were very disruptive. Users treated the line as essentially half-duplex (one person speaks and then indicates the line is free with a verbal signal), rather than the normal full-duplex usage that one associates with phone conversations. Voice requires the arrival of a small packet every 100 ms. Since a packet is about 12 Kb, if the line speed is 20 Kb/s, a single data packet occupies the line for 600 ms, knocking out six small voice packets and leaving recognizable dead air. Even if a VOIP application manages, *on average*, 60 Kb/s, this is not good enough. VOIP does not need 60 Kb/s on average; it needs six Kb every 100 ms, *guaranteed*.

To see why overprovisioning is effective and its limits, consider the train of packets arriving at a destination for an application that requires one packet to arrive every k ms. The application is effective if, in every sequence of packets representing k ms, one arrives for the application. If there are r applications, each with a mean transmission rate equivalent to the chosen application, then the probability that any packet is from the designated application is $1/r$, or the probability that the packet is not from the designated application is $(r-1)/r$. If the bandwidth is t packets per millisecond, tk packets will arrive in k ms. Under the assumption of independent packet transmission events, the probability that the application successfully transmits a single packet in an interval of k ms is therefore

$$P_{\text{succ}} = 1 - [(r-1)/r]^{tk}. \tag{1}$$

An interesting second derivation of (1) is to note that tk is in units of packets, and is in particular the number of packets that one must see in order to see one packet from the application. The number tk is therefore the *minimum overprovisioning ratio* of the channel, which we denote by the parameter u. Further, we can write the expected bandwidth share of the application as $s = 1/r$, and denote the desired QoS parameter as q, the percentage of time that the application succeeds. With this change of variable, we can rewrite (1) as:

$$P_{\text{succ}}(u) = 1 - (1-s)^u \geq q, \tag{2}$$

where both s and q are between 0 and 1, and $u \geq 1$. A little algebra on (2) gives

$$u \geq \frac{\log(1-q)}{\log(1-s)}. \tag{3}$$

From the VOIP example, we can take $q = .99$ (a fairly modest QoS requirement), and, experimentally, we have $u \approx 10$. Solving (3) gives an experimental value of s as about 0.35; we conclude that a standard VOIP call is expecting about 35% of a channel.

In (3) q is fixed as a constant, given by a Service-Level Agreement or customary reliability expectation in the industry. This is generally expressed as a quantity of 9s, as in five 9s of service, which indicates 99.999% uptime, or $1 - q = 10^{-5}$. The mean bandwidth share s is determined experimentally by application load, and its inverse $r = 1/s$ can be thought of as the number of competing applications.

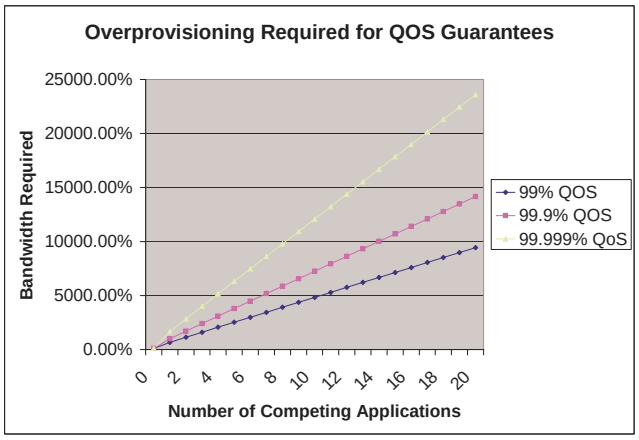

Fig. 1. Overprovisioned bandwidth required vs. number of competing applications.

Fig. 1, shows the overprovisioning required in terms of application bandwidth for various service level agreements in the presence of competing applications. The important thing to note about the graph is the dramatic growth in overprovisioning required even when the number of applications competing for the channel is relatively small and the QoS requirements fairly modest. This implies that overprovisioning is a highly-limited strategy for assuring QoS: it relies on either very high mean bandwidth share (very nearly a dedicated channel) or on very little channel usage by the application that requires QoS. Indeed, we can demonstrate this by computing the sensitivity of overprovisioning to the mean bandwidth share:

$$\frac{du}{ds} = \frac{\log(1-q)}{(1-s)[\log(1-s)]^2}, \tag{4}$$

which is singular at $s = 0$, is uniformly negative for $0 < s < 1$, and implies strong sensitivity at low bandwidth share and low sensitivity at high bandwidth share.

The preceding discussion indicates that existing strategies to extend packet-switched networks to high quality-of-service applications are strongly limited. In the next section, we turn to a signaling-based strategy that combines the strengths of packet- and circuit-switched networks, and its implementation in an overlay.

III. Service Signaling and Overlays

The discussion in the preceding section above implicitly assumed that the mean bandwidth share and overprovisioning were independent variables. In fact, of course, they are not: a channel provisioned for a high QoS application is likely to experience low contention. The maximum router queue depth d is given by

$$d = r - 1 = (1-s)/s. \tag{5}$$

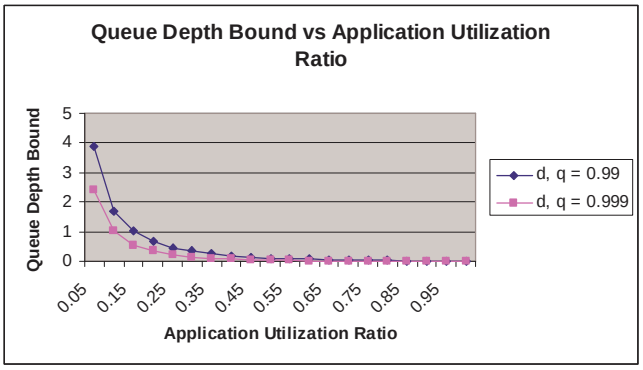

Fig. 2. Queue depth bound vs. utilization ratio.

The network core, which must support a wide mix of applications at any time, is highly overprovisioned: queue depth on core routers is typically less than 1, corresponding to a mean bandwidth share $s > 0.5$. The VOIP data given above confirms this: a 10:1 overprovisioning ratio corresponds to effective mean bandwidth share of about 35% end-to-end, which in turn is equivalent to a *maximum* router queue depth of close to 2 over the entire end-to-end path.

As can be seen from the analysis above, when the mean bandwidth share is higher than 0.5, a relatively low overprovisioning ratio is required. The identification of application mean bandwidth share with the inverse of the router queue depth gives us a strategy for offering guaranteed QoS over a packet-switched network: the *maximum* router queue depth seen by an application is given as a function of the overprovisioning ratio u. We further define the *utilization ratio* as the inverse of the overprovisioning ratio: $p \triangleq 1/u$. Using (5), we can rewrite (3) as

$$d \leq \frac{(1-q)^p}{1-(1-q)^p}, \tag{6}$$

which gives us an upper bound on the maximum queue depth seen by the application. Achieving the bound (6) does not require that the *actual* queue depth fit the bound, merely that the queue depth as seen by the application fits the bound. Note that since the expression $(1-q)^p$ lies between 0 and 1, the right-hand side of (6) is always bounded below by 0; thus, for any parameters q and p the apparent queue depth is achievable. A plot of the maximum queue depth d as a function of the utilization ratio p is shown in Fig. 2.

The steep decline in the queue depth bound with the application utilization ratio indicates that packets from a QoS-sensitive application must see a channel with effectively zero contention: in other words, they must be forwarded immediately. The TIA-1039 explicit rate in-band signaling protocol [2] permits an application to signal its required bandwidth, and routers to grant the request and explicitly manage queues to deliver it. Indeed, the Anagran flow routing implementation (cf. [3]) explicitly assures that average queue depth seen by all applications requiring guaranteed QoS is nearly zero, trivially assuring (6) once a flow's bandwidth has been assured. The preceding discussion gives an alternate strategy, however. If traffic can be managed wherever contention is expected (typically, at the edges of a network), then (3), or its alternative depth form (6) can be assured. The solution is to install *overlay traffic shapers* at each choke-point of the network (a choke-point is precisely defined as any point where the equivalent inequalities (3), (6) cannot be guaranteed by an unmanaged packet network). Together, the collection of these overlay traffic shapers form a *service overlay network*, or SOla.

IV. Functions and Architecture of a SOla node

The SOla concept is based on flow routing and the maintenance of per-flow state information at SOla nodes. As such, it shares some of the characteristics of earlier architectures aimed at providing QoS such as ATM (Asynchronous Transfer Mode) [4], MPLS (MultiProtocol Label Switching) [5], and RSVP (ReSource reserVation Protocol) [6]. As discussed in [3], the concept of flow routing avoids the scalability problems of these earlier protocols. In particular, flow routing employs inband signaling and does not require out-of-band connection setup procedures, which are problematic for the short flows that predominate today's Internet. The flow routing concept lies at the core of the CHART project under DARPA's Internet Control Plane program [7], [8]. Further, the SOla runs as an overlay on an existing TCP/IP network and so is very easy to deploy and adopt.

A SOla node is a conceptually simple device: it sits between an unconstrained overprovisioned network and a number of edge devices contending for a bandwidth-constrained channel. In contrast to conventional routers, the SOla node maintains per-flow state and processes traffic in terms of flows (cf. [3]). Generally speaking, a *flow* is a stream of packets transmitted along a network path having a common set of header values. The highest granularity of flow that we shall consider is identified by the following five parameters in the TCP/IP header: source and destination IP addresses, source and destination port numbers, and protocol number. For example, legacy UDP and TCP connections are flows. Flow routing, in conjunction with a flow-aware control plane enables tight QoS provisioning, which is not possible in today's packet-based Internet. The TIA-1039 protocol [2] is a flow-based inband signaling protocol, which forms the basis of such a control plane. We shall use the term *QoS-flow* to refer to a TIA-1039 flow or an aggregate of such flows, to distinguish them from legacy TCP and UDP flows.

The SOla node takes in TIA-1039 based QoS-flow requests, allocates bandwidth among the competing flows, and forwards packets into the constrained channel in accordance with the allocated bandwidth. Packets transmitted to the SOla node in excess of the negotiated rates are discarded. A set of SOla nodes introduced as overlay nodes in a network create a virtual network topology of overlay links. We claim that a SOla network can be designed to provide QoS. In particular, such a SOla network could support flows requiring

an available rate, as well as flows requiring a guaranteed rate. Furthermore, QoS provisioning in terms of requirements on latency and packet loss rate could also be supported.

A. Available Rate and Guaranteed Rate Flows

The TIA-1039 signaling protocol allows QoS-flows to request an available rate (AR) and a guaranteed rate (GR) via inband signaling [9], [2]. Other QoS-related parameters are supported by TIA-1039, but we shall focus on the AR and GR parameters. According to the TIA-1039 protocol, the first packet in a QoS-flow contains an additional header (hence, *inband signaling*) called a *QoS structure* which contains, among other fields, an AR field and a GR field (hence, *explicit rate signaling*). A packet containing a QoS structure is called a *QoS packet*. The AR and GR values are initialized by the sender and are modified by each TIA-1039 compliant node on the network path taken by the packet.

A TIA-1039 node on the path, determines the AR value that can be supported for the flow on the outgoing link and overwrites the AR field in the QoS structure, if necessary. Similarly, the TIA-1039 node determines the GR value that can be supported on the local outgoing link and overwrites the GR field in the QoS structure, if necessary. The QoS structure eventually reaches the receiver, which then forwards it back to the sender. The sender then transmits at the rate $\mathrm{AR}+\mathrm{GR}$. The sender inserts a QoS structure into the packet stream of a flow once every 128 packets.

A TIA-1039 node determines the AR on an outgoing link by determining the per-flow available bandwidth on the link, which fluctuates depending on the load on the link. By contrast, the GR value is not affected by the prevailing traffic load. Rather, the GR value is determined as a function of the outgoing link capacity, and the GR values that have already been assigned to other flows. An admission control function ensures that the sum of the GR values of all flows assigned to a given outgoing link does not exceed the link capacity. A modified TCP driver [9], [7], [10], which we refer to as TCP-ER, has been developed which initiates TIA-1039 signaling and controls the TCP window size based on explicit rate information carried by QoS response packets.

B. Support of Available Rate

The drawback of overlay networks is the flip side of their strength: if overlay networks do not modify the underlying network, neither can they control it. Rather, the strength of the overlay is in shaping traffic through a network with known, predictable behavior. The difficulty lies when there is uncontrolled cross-traffic in the underlying network; traffic that does not transit the overlay shaper.

In order to cope with this, we propose a bandwidth probe control (BPC) mechanism to determine the available bandwidth of an overlay link with cross-traffic. We draw a distinction here between BPC and probing techniques that attempt to detect the available bandwidth of an underlying network path through the transmission of packet-trains or packet-pairs. Examples of bandwidth probing techniques include Pathload [11], pathChirp [12], and Spruce [13]. The basic idea of bandwidth probing is to inject a sequence of packets at the ingress of a given path and then to deduce the available bandwidth by observing the statistics of the delay jitter introduced into the sequence received at the egress of the path. Such bandwidth probing techniques cannot be used to realize SOlas for the following reasons:

1) Studies have shown (cf. [13]) that the probing techniques can suffer from severe inaccuracies in estimating the available bandwidth on a path. Our own experimentation with several of the popular techniques on PlanetLab [14] has led us to the conclusion that probing techniques tend to severely underestimate, and in some cases even overestimate, the available bandwidth on a path, depending on the nature of the cross-traffic.
2) Cross-traffic is typically such that the path available bandwidth is a highly nonstationary and time-varying random process. Probing techniques are based on statistics derived from multiple sets of probes injected into the path and hence tend to converge rather slowly.
3) TCP cross-traffic attempts to use up all of the available bandwidth on a path. If a given path is relatively "clean," i.e., low latency and low packet loss rate, the available bandwidth estimated by probing techniques will be nearly zero.

In contrast to estimating the bandwidth left over by cross-traffic on a path, the bandwidth probe control approach applies local congestion control between the overlay nodes at the ingress and egress of the path to determine the share of available bandwidth that can be used by the associated overlay link. Consider the case where the cross-traffic consists entirely of TCP flows and the overlay link traffic consists entirely of available rate (AR) flows. The mechanism determines the "fair" share of the path bandwidth that should be allocated to the overlay link. We shall assume a simplistic notion of fairness in terms of TCP flow fairness. Alternative forms of fairness, e.g., utility flow fairness, could easily be incorporated into the approach. In the case of TCP flow fairness, if there are m cross-traffic TCP flows on the bottleneck link of the path and n TCP-ER flows on the overlay link, the aggregate available bandwidth share for the overlay link should be

$$C_{\mathrm{ov}} = \frac{nC_{\mathrm{bot}}}{m+n},$$

where C_{bot} denotes the capacity of the path's bottleneck link.

We have implemented a preliminary bandwidth probe control scheme based on AIMD (Additive-Increase Multiplicative Decrease) congestion control that could be implemented on a SOla node. Fig. 3 shows the throughput received by a TCP-ER flow and a cross-traffic TCP flow under bandwidth probe control over a path of capacity 200 kbps when the packet loss rate is zero. The results were obtained on Emulab [15] using an implementation of bandwidth probe control on a software router based on Click [16], [17].

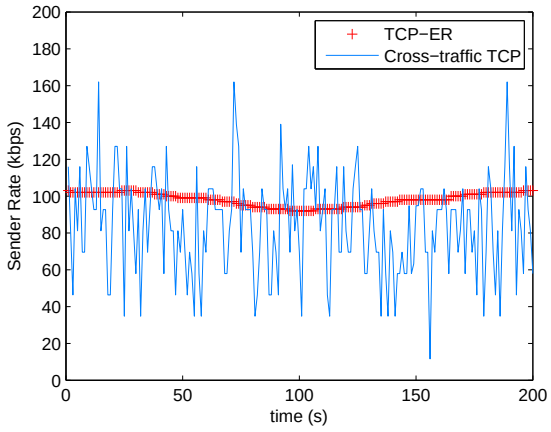

Fig. 3. Bandwidth probe control on 200 kbps capacity path with zero packet loss: Sender rates for TCP-ER and cross-traffic TCP.

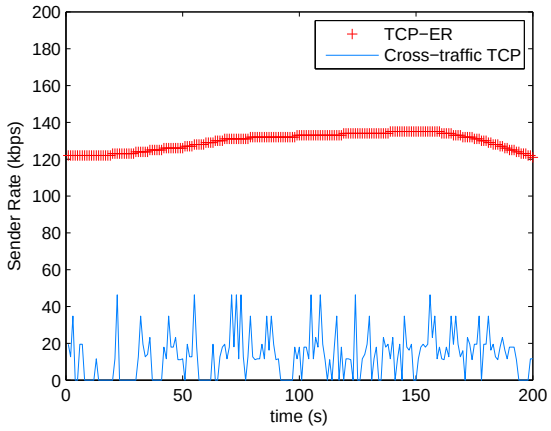

Fig. 4. Bandwidth probe control on 200 kbps capacity path with 10% packet loss rate: Sender rates for TCP-ER and cross-traffic TCP.

Observe that the TCP-ER flow on the overlay link receives a nearly constant throughput of about 100 kbps, while the cross-traffic TCP flow receives an average throughput of approximately 100 kbps. The legacy TCP flow follows the sawtooth pattern typical of TCP congestion control. Fig. 4 show the throughputs of the two flows when the packet loss rate on the bottleneck link is set to 10%. The legacy TCP flow achieves an average throughput of less than 20 kbps due to the packet losses, while the TCP-ER flow achieves a nearly constant throughput of 125 kbps. This demonstrates that it is possible to achieve nearly circuit-switched throughput performance for TCP-ER flows on an overlay link in the presence of TCP cross-traffic. The same mechanism is able to determine the available bandwidth accurately in the presence of UDP cross-traffic. We remark that the legacy TCP cross-traffic is controlled by conventional TCP congestion control mechanisms. On the other hand, UDP cross-traffic is uncontrolled.

C. Support of Guaranteed Rate

The presence of uncontrolled cross-traffic in the underlying network makes it difficult to provide guarantees of any sort in a conventional overlay network. In particular, the outgoing bandwidth to another service overlay point is not known to the shaper, and as a result bandwidth rate requests may be improperly granted (or denied).

In the previous subsection, we discussed how a bandwidth probe control mechanism can determine the available bandwidth on a SOla for flows requiring an available rate (e.g., TCP-ER flows). A guaranteed rate flow rate can also be provided over a given SOla overlay link, assuming that the underlying network path supports priority scheduling, e.g., DiffServ [18]. Packets of GR-flows on a SOla are marked as high priority and all other packets, including cross-traffic TCP and UDP packets are assumed to be of lower priority. The non-SOla nodes on the underlying network path must give precedence to packets belonging to the GR-flows.

It is not hard to see that guaranteed flow rates can be provisioned on the overlay link under the following conditions:

1) the underlying path bottleneck capacity is known,
2) the underlying path does not change,
3) the overlay link in question does not share a common physical link with any other SOla overlay link.

Under these assumptions, the SOla overlay nodes can perform allocation and admission control for GR-flows traversing the SOla overlay link. Since all cross-traffic is assumed to have lower priority than the GR-flows, a circuit-like guaranteed service can be realized on a SOla.

The first assumption is not difficult to satisfy using existing tools. The path bottleneck capacity generally changes on a relatively slow time-scale and can be estimated with reasonable accuracy using a probing technique such as Pathrate [19] or CapProbe [20]. Note that whereas the path *available bandwidth* changes on a fast time-scale and is notoriously difficult to estimate via probing, as discussed above, path *capacity* is much easier to estimate with sufficient accuracy. Alternatively, the bottleneck path capacity can be determined more precisely from the underlying network using network management queries such as SNMP requests.

The second and third assumptions do not hold in general. Nevertheless, a protocol running only on the SOla nodes can be designed to deal with this issue. Each SOla node must keep track of the current GR allocations for its outgoing overlay links, as well as the GR allocations for any other overlay links that overlap with these outgoing overlay links. A simple way of achieving this is to maintain a global database of GR allocations in the network. Such a database could be implemented in a distributed manner to minimize the extra overhead incurred.

We remark that the network path corresponding to an overlay link can be fixed using MPLS (Multi-Protocol Label Switching) [5]. In this case, the overlay link corresponds to a label switched path. Establishing LSP-based tunnels between SOla nodes minimizes overlap among SOla overlay links.

D. Network Sensing

The knowledge of real-time status of overlay links is essential for providing QoS guarantees in an end-to-end fashion. Determining different network properties of overlay links such as latency, packet loss rate, and capacity in real-time for a system with large number of SOla nodes poses a huge scalability challenge. We propose to leverage S^3 [21], a distributed system that scales with large number of overlay links through measuring only a small number of links and accurately estimating the metrics on other links via scalable inference algorithms such as NetVigator [22] for latency estimation. Also S^3 follows service-oriented architecture principles and provides access to the measurement infrastructure as a web service with flexible interfaces to invoke one-time and continuous measurements. A key issue is the tradeoff between the overhead incurred by network sensing and the quality of the link metric estimates in terms of accuracy and responsiveness.

E. Adaptive Flow Routing

Network sensing and the bandwidth probe control mechanism enable a SOla node to assign flows to paths in order maximize throughput or to satisfy specific end-to-end QoS requirements on delay and/or packet loss rate. We refer to the assignment of flows to paths, taking into account real-time information on path quality and resource availability as *adaptive flow routing*. With adaptive flow routing, the distribution of flows on paths adapts to changes in path quality, as determined by networking sensing, and resource availability, as determined by bandwidth probe control. The goal of adaptive flow routing is to maximize network utilization via intelligent load balancing while providing QoS to GR-flows.

In order to realize adaptive flow routing, the SOla node must maintain multiple paths to a given destination along with information on path quality and resource availability for each path. The performance of an adaptive routing overlay is studied in [17], but in that study, all flows to a common destination were constrained to follow the same path at any given time. This type of adaptive routing is more precisely referred to as *adaptive alternative routing*. Adaptive flow routing may be viewed as a kind of multipath routing whereby multiple paths to the same destination are used *simultaneously*. Some preliminary results on this type of multipath flow routing are discussed in [23].

V. Conclusion

We pointed out the major limitations of today's packet-switched networks for providing QoS and proposed a QoS architecture based on signaling mechanisms deployed on a service overlay network or SOla. The main ingredients of the SOla architecture can be summarized as follows: 1) Traffic shapers applied at network choke-points; 2) A bandwidth probe control mechanism; 3) Priority scheduling and a database of rate allocations for overlay links; 4) Network sensing of overlay link characteristics. We discussed the impact of a SOla architecture on network operations and proposed performance metrics and test scenarios to evaluate the effectiveness of the SOla concept.

References

[1] HALO. http://www.hp.com/halo.

[2] L. G. Roberts, "QoS Signaling for IP QoS Support, version 2," tech. rep., Telecommunication Industry Association (TIA), Apr. 2007.

[3] L. G. Roberts, "The Next Generation of IP - Flow Routing," in *Proc. SSGRR 2003S International Conference*, (L'Aquila Italy), Jul. 2003.

[4] ATM Forum, "ATM Traffic Management Specification Version 4.0 <af-tm-0056.000>," Apr. 1996.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services." IETF RFC 2475, Dec. 1998.

[6] B. Braden, L. Zhang, B. S., S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification." IETF RFC 2205, Sep. 1997.

[7] A. Bavier *et al.*, "Increasing TCP Throughput with an Enhanced Internet Control Plane," in *Proc. IEEE Milcom '06*, (Washington DC), Oct. 2006.

[8] R. McGeer *et al.*, "The CHART system: A high performance, fair transport architecture based on explicit rate signaling," in *Proc. ACM SIGOPS Operating Systems Review*, Jan. 2009.

[9] L. G. Roberts, "Major Improvements in TCP Performance over Satellite and Radio," in *Proc. IEEE Milcom'06*, (Washington DC), Oct. 2006.

[10] J. Brassil *et al.*, "The CHART System: A High-Performance, Fair Transport Architecture Based on Explicit-Rate Signaling," *ACM SIGOPS Operating Systems Review*, vol. 43, pp. 26–35, Jan. 2009.

[11] M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," *IEEE/ACM Trans. on Networking*, 2003.

[12] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths," in *Proc. Passive and Active Measurement Workshop*, 2003.

[13] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proc. Internet Measurements Conference*, (Florida), 2003.

[14] PlanetLab. http://www.planet-lab.org.

[15] B. White *et al.*, "An Integrated Experimental Environment for Distributed Systems and Networks," in *OSDI02*, (Boston, MA), pp. 255–270, USENIX Assoc., Dec. 2002.

[16] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click modular router," *ACM Trans. on Computer Systems*, vol. 19, pp. 263–297, Aug. 2000.

[17] B. L. Mark and S. Zhang, "A Multipath Flow Routing Approach for Increasing Throughput in the Internet," in *Proc. IEEE PacRim'07*, Aug. 2007.

[18] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services." IETF RFC 2475, Dec. 1998.

[19] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 963 – 977, Dec. 2004.

[20] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "CapProbe: A Simple and Accurate Capacity Estimation Technique," in *ACM SIGCOMM 2004*, (Portland, OR), 2004.

[21] P. Yalagandula, P. Sharma, S. Banerjee, and S.-J. Lee, "S^3: A Scalable Sensing Service for Monitoring Large Networked Systems," in *Proc. ACM SIGCOMM Workshop on Internet Network Management*, 2006.

[22] P. Sharma, Z. Xu, S. Banerjee, and S.-J. Lee, "Estimating network proximity and latency," *ACM Comput. Commun. Rev.*, vol. 36, no. 3, pp. 39–50, 2006.

[23] B. L. Mark, S. Zhang, R. McGeer, J. Brassil, P. Sharma, and P. Yalagandula, "Performance of an Adaptive Routing Overlay under Dynamic Link Impairments," in *IEEE Milcom'07*, Aug. 2007.