Bridging the Network Reservation Gap Using Overlays

Angelos Stavrou
Dept. of Computer Science
Columbia University
angel@cs.columbia.edu

David Turner
Dept. of Computer Science
Drexel University
dmt36@drexel.edu

Angelos D. Keromytis
Dept. of Computer Science
Columbia University
angelos@cs.columbia.edu

Vassilis Prevelakis
Dept. of Computer Science
Drexel University
vp@drexel.edu

***Abstract*— We propose the concept of *Overlay-linked IntServ (OLIntServ),* a system architecture that combines network overlays with intra-domain QoS to provide assured communications over the Internet, while allowing ISPs to extend the reach of their currently under-utilized IntServ services. We describe our system prototype, and provide some preliminary experimental results on its efficacy.**

I. Introduction

The increasing reliance on the Internet for time- and mission-critical communications has brought to the forefront concerns about its availability and reliability. Whether due to natural traffic fluctuations or large-scale distributed denial of service attacks, the available bandwidth for relatively long-lived, time-sensitive communication streams can vary dramatically over the lifetime of such flows. To address such concerns, end-to-end network QoS reservation protocols were designed.

However, years of research on various QoS architectures for the Internet have resulted in sophisticated proposals that have not been broadly accepted commercially. In particular, Integrated Services (IntServ) and Differentiated Services (DiffServ) have long been supported by major router and operating system vendors, yet have only seen minimal use in practice. Without postulating as to the possible reasons behind the lack of enthusiasm on behalf of ISPs and users, we recognize the fact that an enterprise that wishes to have some QoS assurances today (and, we suspect, in the near future) has few options.

Network overlays has been shown to offer some statistical guarantees that the underlying network fabric does not. For example, overlays can improve the performance and robustness of unicast routing [1], [2] by providing alternate paths from a particular source to a particular destination along paths that proceeded through intermediate end-systems. However, such approaches require considerable application customization and coordination among the (distributed) overlay nodes.

We propose *BandExSOS*, a hybrid architecture that combines intra-ISP QoS reservations (based on RSVP) with a multi-path-routing overlay architecture based on our previous work on Secure Overlay Services (SOS) [3]. In our proposed approach, either (or both) communication end-points build secure RSVP tunnels [4] to the closest overlay node, which should reside in the same ISP (or is otherwise "reachable" by RSVP). End-to-end traffic is then routed through these overlay nodes. Inside the overlay, traffic is replicated and routed through several disjoint paths, seeking to improve resilience and to take advantage of traffic asymmetries.

Our experiments using Planetlab indicate that in many cases, our approach can result in a dependable decrease in end-to-end latency, and (perhaps most importantly) decrease the latency variance. Although preliminary, we believe our results demonstrate that our approach is a promising way of bridging the QoS reservation gap across the Internet.

II. Related Work

QoS provision and management has a wide-ranging literature. A lot of the early work was stimulated by the promise of ATM networks. The demand for these services was stimulated by multimedia traffic. The relevant promise was the control of multiplexing behavior in both end-points and network elements, with the idea that queuing disciplines such as Fair Queuing or its many variants could be used to allocate bandwidth resources and provide delay bounds.

Despite the ever-increasing use of time-sensitive protocols (*e.g.,* VoIP, audio on demand, *etc.*) bandwidth reservation has not been particularly successful. This is caused mainly by the fear that since these applications have modest bandwidth requirements the operation of a reservation and payment infrastructure would not be feasible economically. Recently, however, newer applications such as video on demand, telepresence, and Grid Computing, have bandwidth requirements that may constitute a significant portion of the available bandwidth. The overheads associated with reservation and billing are smaller (because we are dealing with fewer, more expensive reservations), while the benefits are greater because of the impact of the data flows on the infrastructure.

In Grid Computing in particular, efforts are already underway [5], to allow end-users to create end-to-end light paths (optical links that allow unstructured access to the fiber infrastructure) by combining individual segments very much as we described in the introduction. The current systems, however, are targeted towards the academic community and hence assume that end-users have the required expertise and have non-competitive usage strategies. Specifically under the "User Controlled Light Paths" framework [5], (a) end-users have to be known by the system in advance, (b) policy enforcement is not addressed, (c) there is no purchasing of bandwidth, since the network is considered a common resource. In a commercial environment, a similar system must deal with billing (*i.e.,* how the reserved bandwidth can be paid by the user) and must support bandwidth reservation in a scalable and secure manner.

Each reservation carries with it some overhead. This includes both protocol overhead, but also state that must be maintained by routers for each reservation. As the number of reservations increases so does the overhead. Unless there is some kind of aggregation of requests this overhead will ultimately define an upper bound on the number of reservations that can be accommodated by the existing infrastructure. The complexity of some of the proposed systems [6], [7] and the small scale of their test-beds casts grave doubts on their ability to scale to millions of users and thousands of network elements. Various techniques that attempt to improve scalability through aggregation are vulnerable to abuse. For example, Zhang *et al.* [8] describe request aggregation whereby multiple requests are merged into a single larger request for the total bandwidth asked for by the individual requests. This approach, however, may result in an upstream node declining the single request thus denying access to all the requests, even through some of the individual requests could have gone through [9].

III. ARCHITECTURE

Our architecture, shown in Figure 1 combines two different components: a distributed overlay network that is used for multi-path routing of traffic between any pair of overlay nodes, and a secure RSVP-based reservations system (BandEx) for building a tunnel from either communication end-point to the closest overlay node. In essence, the overlay provides the missing link between the IntServ reservations of the end-points. By exploiting redundant available capacity and by spreading the risk of unexpected traffic peaks across many links, our architecture should minimize the impact of such traffic peaks (or denial of service attacks) on any given link. Section IV contains some preliminary experimental evidence supporting our hypothesis. We should note that our results are consistent with independent work on the effect of multi-path routing on end-to-end latency [10] and availability [11].

In the remainder of this section we provide an overview of these two components. We refer the reader to our previous work describing these systems in more detail [12], [4].

A. SpreadSpectrum SOS

Our network overlay architecture extends the ideas of SOS [3]. SOS effectively implements functionality equivalent to that of a firewall "deep" enough in the network such that the access link to an end-host does not become congested as a result of a denial of service attack. In terms of network topology, this typically means the first or second-level router in the hosting Internet Service Provider's Point-of-Presence (POP). This distributed firewall may perform access control by using protocols such as IPsec or TLS, or by relying on authentication and authorization services from the system being protected. Traffic is then routed to a secret location, which can be the service provider itself or a node that is allowed to contact the service provider (called "secret forwarder" in SOS [3]), with all other traffic being filtered, as shown in Figure 2. The secret forwarder can vary over time, and is different for each site protected by the overlay; part of the functionality built in these indirection mechanisms concerns itself with maintaining and propagating this information to other indirection nodes. Otherwise, we assume that the identity of the protected server and all indirection nodes is publicly known or easily determined by an attacker.

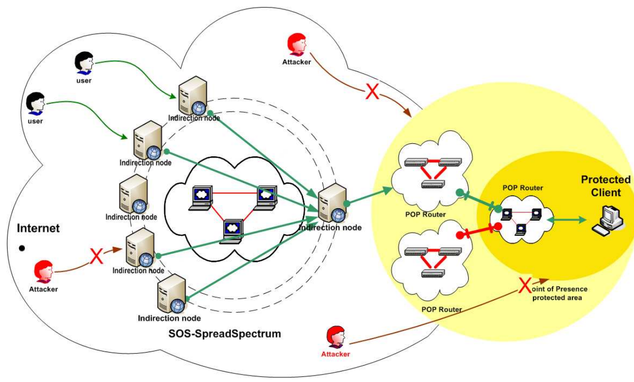

Fig. 2. **The Secure Overlay Services anti-DDoS system**

Our approach with multi-path SOS is straightforward: instead of picking one (possibly random) path through the overlay, spread the packets from the overlay ingress node (or from the end host, if no network reservations are possible) across all indirection nodes in a pseudo-random manner. This new communication mechanism also protects the client-server connection establishment and guarantees uninterrupted connectivity to the target server throughout the client's session. The admitted packets are internally forwarded to the overlay egress point (the node to which the remote peer has created an RSVP tunnel to), or to a random overlay node that is authorized to forward traffic to the remote end host. Only authorized clients are allowed to use the overlay and contact the hosting servers and these clients are provisioned in advance (*e.g.,* at registration time) with the appropriate authentication material, such as an RSA public/private key pair and a public-key certificate. BandExSOS may work in conjunction with filtering routers close to the hosting infrastructure, to allow only traffic from the overlay's egress points (identified through the RSVP tunnel) to reach end hosts. All other traffic is filtered out or at a minimum rate-limited.

B. BandEx

Having provided an effective mechanism for protecting the data traffic as it transits the network core, we are left with the problem of providing an equivalent level of protection for the "last-hop". In other words, we need a mechanism that will safeguard data streams between the connection end-points and the BandExSOS core.

We propose a mechanism for secure bandwidth reservation by allowing a content provider to pay for a customer's bandwidth. Under our scheme, the content provider (Figure 4) pays for the bandwidth to the customer, by sending the customer

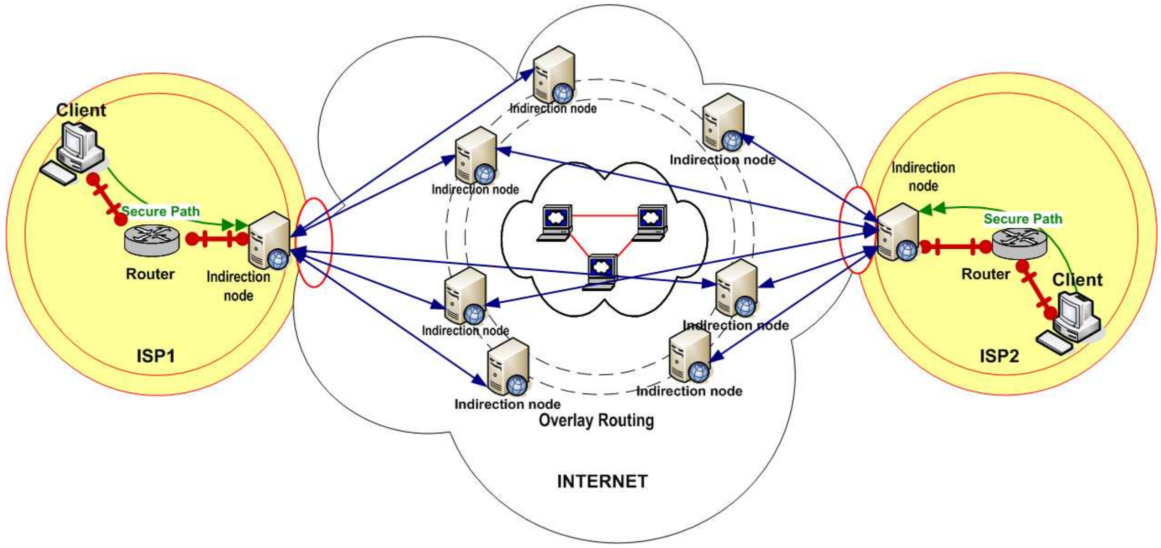

Fig. 1. **The BandExSOS architecture**

an electronic check for the reservation. The customer then initiates a secure-RSVP operation issuing checks guaranteed by the provider's check to the ISP's nodes. Eventually the path reaches the overlay cloud. The overlay is considered by RSVP as one hop, so the path eventually exits the overlay and resource reservation is resumed until the content provider's network is reached, completing the transaction.

This architecture assumes that the ISP (or ISPs) have a business relationship with the content provider. If this is not the case, then a credit institution such as a bank can act as the link between the two parties. In this case, the credit institution will issue a "spending authorization" credential to the content provider thus completing the chain of trust.

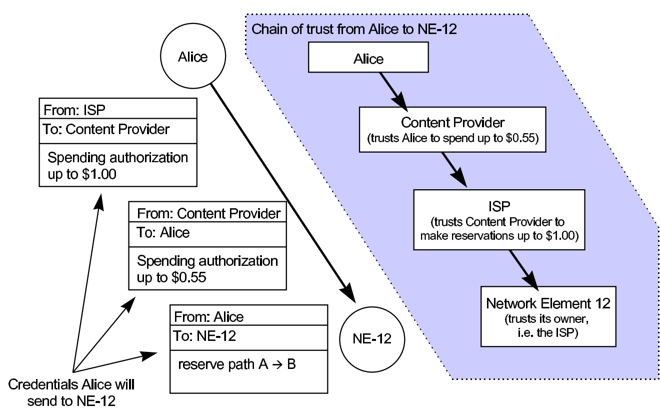

Fig. 3. **Trust relationship for secure reservations. Other trust relationships are possible, and are orthogonal to the overlay.**

Consider the case of Alice who wishes to watch a movie provided by a Content Provider (CP). When Alice selects her movie, she will be provided with the appropriate bandwidth purchasing credentials and an RSVPtransaction will be initiated to create a path between Alice's machine and the CP's network.

RSVP messages are composed of objects that specify important parameters for the reservation exchange. Two of these objects, RSVP's FLOWSPEC and POLICY_DATA, are relevant to our discussion. A FLOWSPEC contains the requested QoS parameters and the POLICY_DATA object contains information regarding authorization policies for the request. These objects are both checked before a reservation is made to ensure that the request is possible. RSVP uses the FLOWSPEC in admission control to check whether router actually supports and has adequate resources for the desired QoS. Additionally, policy control checks whether the reservation is authorized using the information contained within the POLICY_DATA object and most likely, a local policy. Both objects were designed to be completely opaque to the RSVP specification. That is, RSVP was not designed for a specific QoS or policy model in mind so that it could be extended easily for future QoS and policy control services. RFC 2210 specifies an implementation of IETF Integrated Services with RSVP which is probably the most common form of the FLOWSPEC in current implementations. Our implementation uses the POLICY_DATA object to convey policy information for the BandExSOS architecture. RFC 2750 describes the POLICY_DATA object as being composed of any number of policy elements. The information within these elements is application defined and is not dictated by RSVP.

Assuming that network element 12 (NE-12) belonging to ISP is in the reservation path, the RSVP message will include the credentials shown in Figure 3. NE-12 will receive the request and will be able to determine *solely on the basis of the information provided in the message* whether to grant or deny the reservation request. Thus, each network element can reserve resources on its own without communicating with other computers of the ISP's network. This reduces the overhead of the transaction and speeds the decision.

KeyNote Microchecks BandExSOS uses KeyNote [13] credentials for bandwidth reservation. These credentials describe the conditions under which a user is allowed to perform a transaction and the fact that a Content Provider is authorized to participate in a particular transaction.

Initially, the CP encodes the details of the desired bandwidth into a *credit authorization* that is sent to the customer, along

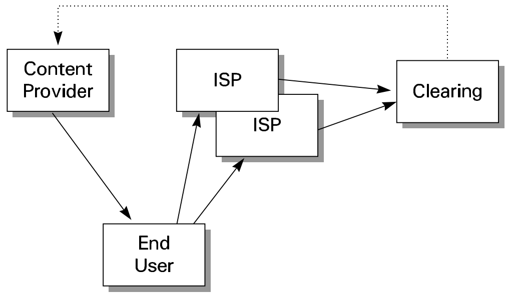

Fig. 4. **Microbilling architecture diagram.**

with additional credentials that authorize the CP to utilize the bandwidth under the same conditions as those enclosed in the credit authorization. The customer includes these credentials in the RSVP messages along with a microcheck for the desired bandwidth. The credit authorizations and microchecks are encoded as KeyNote credentials that authorize payment for a specific transaction. The user creates a KeyNote credential signed with her public key and sends it, along with her credential from the CP, to the first network element of the ISP. This credential is effectively a check signed by the user (the Authorizer) and payable to the ISP (the Licensee). The conditions under which this check is valid match the credit authorization sent to the user by the CP. Part of the credit authorization is a nonce, which maps payments to specific transactions, and prevents double-depositing by the ISP.

To determine whether he can expect to be paid (and therefore whether to accept the payment), the ISP passes the action description (the attributes and values in the offer) and the user's key along with the ISP's policy (that identifies the CP's key), the user credential, the credit authorization credential (signed by the CP), and the microchecks credential (signed by the user) to his local KeyNote compliance checker. If the compliance checker authorizes the transaction, the ISP is guaranteed that the CP will allow payment. The correct linkage among the ISP's policy, the CP key, the user key, and the transaction details follow from KeyNote's semantics [13]. If the transaction is approved, the ISP can configure the appropriate routers such that the user's traffic is treated according to the offer, and store a copy of the microcheck along with the user credential and associated offer details for later settlement and payment.

Periodically, the ISP will 'deposit' the microchecks (and associated transaction details) he has collected to the Clearing and Settlement Center (CSC). The CSC may or may not be run by the same company as the CP, but it must have the proper authorization to transmit billing and payment records to the CP for the customers. The CSC receives payment records from the various ISPs; these records consist of the offer, and the KeyNote microcheck and credential from the user sent in response to the offer. To verify that a microcheck is good, the CSC goes through a similar procedure as the ISP did when accepting the microcheck. If the KeyNote compliance checker approves, the check is accepted. Using her public key as an index, the user's account is debited for the transaction amount. Similarly, the ISP's account is credited for the same amount.

BandExSOS Operation Having seen the overall system architecture, let us look at a particular example. *Alice* is a user who wants to watch a movie from *Martha's* site. Alice will need to reserve bandwidth for that particular link with *Nick's* ISP. Alice contacts Martha and obtains a fresh *credit authorization* credential, which allows her to issue KeyNote microchecks. The CA credential shown below (most of the hex digits from the keys have been removed for brevity) allows Alice to write checks for up to 0.55 US Dollars, and she can do so until March 24th, 2007.

```
Keynote-Version: 2
Local-Constants:
      ALICE_KEY = "rsa-base64:MCgCIQ...
      MARTHA_KEY = "rsa-base64:MIGJAo..."
Authorizer: MARTHA_KEY
Licensees: ALICE_KEY
Conditions: app_domain == "Band-X" &&
      currency == "USD" && &amount < 0.56
      && date < "20070324" -> "true";
Signature: "sig-rsa-sha1-base64:QU6SZ..."
```

Alice contacts her ISP and receives an offer credential that contains the cost and parameters of the reservation. For example, 50Mbps on a connection from Dublin to NYC for 44 cents:

```
Keynote-Version: 2
Local-Constants:
      ISP_KEY = "rsa-base64:7231f..."
      ROUTE_KEY = "rsa-base64:33a41..."
Authorizer: ISP_KEY
Licensees: ROUTE_KEY
Conditions: app_domain == "Band-X" &&
      currency == "USD" &&
      &bandwidth <= "50Mbps" &&
      link_name == "Dublin-NYC" &&
      &amount >= 0.44
      && date < "20071120 -> "true";
Signature: "sig-rsa-sha1-base64:ab1XXA..."
```

In practice an Offer Credential includes QoS attributes, such as bandwidth, using the Intserv FLOWSPEC notation defined in RFC 2210. With the offer credential on hand, Alice then writes a check for the appropriate amount:

```
Keynote-Version: 2
Local-Constants:
      ALICE_KEY = "rsa-base64:Mcg..."
      ISP_KEY = "rsa-base64:7231f..."
Authorizer: ALICE_KEY
Licensees: ISP_KEY
Conditions: app_domain == "BAND-X" &&
      currency == "USD" && amount == "0.44"
      && nonce == "eb2c3dfc8e9a" &&
      date == "20071120" -> "true";
Signature: "sig-rsa-sha1-base64:Qsd..."
```

The `nonce` is a random number that must be different for each check, guaranteeing that there will be no double-depositing of checks. Alice then sends the Offer Credential and the micro-check to Nick's router using RSVP. Nick receives these credentials, validates the microcheck to make sure that he will get paid, and configures the router appropriately. If the check is not good, Nick will say so, and refuse to make the reservation. Nick will verify that he will get paid, and will evaluate the Offer Credential and the microcheck using a simple policy such as:

```
Keynote-Version: 2
Local-Constants:
      NICK_KEY = "rsa-base64:7231f..."
      MARTHA_KEY = "rsa-base64:MIGJAo..."
Authorizer: POLICY
Licensees: MARTHA_KEY && NICK_KEY
Conditions:
      app_domain == "BAND-X" -> "true";
```

This policy says that anything that Nick's key *and* the Martha's key jointly authorize is allowed. Thus, Alice must submit a valid payment and a valid Offer Credential. Since the bandwidth was paid for, and a path can be found from POLICY to a user (Alice) that has delegated to Nick's key, which in turn has created an open-access Offer Credential, the operation is allowed.

If additional routers need to be configured in Nick's ISP, the first router forwards the necessary information to the next. Note that it is not necessary for the router itself to perform the signature verifications and policy validations: it can simply refer these operations to a Policy Decision Point (PDP), as is envisioned by the IntServ architecture.

Discussion The mechanism described above allows bandwidth reservation to be set up for the "last hop" in a way that is both efficient and manageable. We now address the two major problems that made bandwidth reservation a non-starter in the past, namely scalability and trust.

Scalability issues emerge as multiple reserved paths converge in the backbone and core portions of the network, burdening routers with large numbers of reservation entries.

In our case the scalability problem is not a concern as we are only interested in the last-hop, the protection of data streams in the core is handled by another mechanism. Thus, we use reservations only in the portion of the path that they are useful and dispense with them in the part where they cause problems.

The problem of *trust* is more complex, as allowing a customer to issue a reservation request (*e.g.,* using a reservation protocol such as RSVP), implies that we need some way to determine whether we *trust* the requests issued by the customer to our network elements. The problem is made worse by the fact that connections may span provider boundaries, thus the network elements of a remote provider may receive requests from a customer that has no previous relationship with the provider. Our system addresses this by limiting both the number of network elements that need to receive reservations and the domains that need to be crossed before reaching the core overlay network, and by allowing a (relatively) small number of entities (the content providers) to have trusted relationships with the ISP. For connections spanning national boundaries, our framework also supports the use of credit institutions (*e.g.,* banks) that can form a top level trust layer, linking ISP to content providers in remote jurisdictions. Note that the trust structure is independent of the overlay mechanism.

IV. Evaluation

To demonstrate the feasibility of our architecture, we implemented the BandExSOS prototype and deployed the indirection nodes in 80 PlanetLab nodes, while having the client and server reside in our local network. Our architecture spreads the packets across all indirection nodes, without performing any measurements or using any type of feedback from the network. Although such analysis has been shown to be beneficial [10], our results are encouraging even in the naive case.

Perhaps the most surprising aspect of our implementation is its size: excluding cryptographic libraries, the system consists of less than 4,000 lines of well commented C code. Although this is a prototype implementation and does not include management code and other facilities that would be required in a production system, we feel that the system is surprisingly lightweight and easy to comprehend.

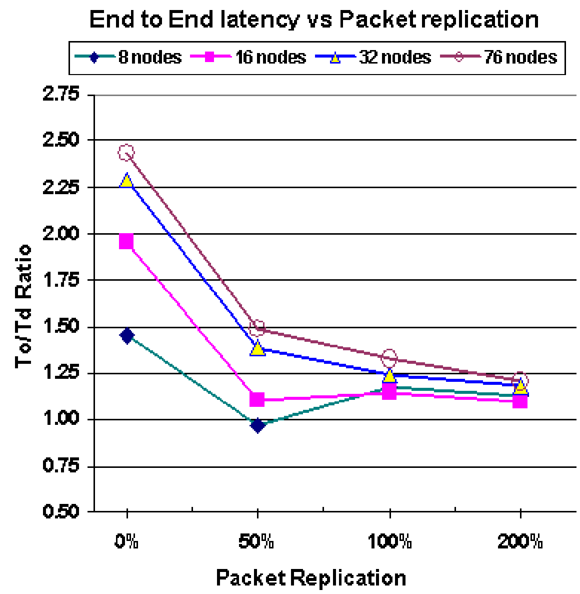

Fig. 5. **End-to-end average latency results for the index page and a collection of pages for www.cnn.com. The different points denote the change in the end-to-end latency through the overlay (T_o) when compared to the direct connection (T_d). Different lines represent different overlays sizes. Increasing the replication factor and overlay size, we get lower average latency results because of the multi-path effect on the transmitted packets.**

Looking at the end-to-end average latency results in Figure 5, we note that as we increase the replication factor (*i.e.,* the number of packet copies that are routed through different paths in the overlay), and for larger overlay networks, we get better average latency results. The worst-case scenario involves a 2.5 increase in latency, dropping to 1.5 with 50% packet replication (*i.e.,* probability of replicating a packet of 50%).

To measure the effectiveness of our system in the presence of highly variable traffic, we simulated network unreachability by disabling overlay nodes at random. In our experiment, the overlay ingress point kept spreading data across all overlay nodes, since it was unaware which of the overlay nodes were temporarily unreachable (no feedback). We then varied the portion of the overlay nodes we disabled and measured the

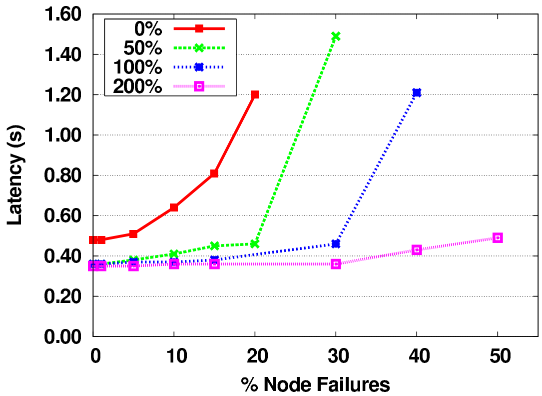

Fig. 6. **Impact of failures of the overlay network on end-to-end latency. Different curves represent varying levels of packet replication. With 200% packet replication, latency increases by less than 25% when up to 50% of paths fail.**

resulting increase in latency. The results are shown in Figure 6. When we do not use any replication, TCP connections perform relatively well when the losses are up to 9%-10% of the total packets transmitted. As we increase the packet replication factor, we achieve higher network resilience.

The dominant cost of the reservation component is that of authentication and authorization, *i.e.,* in the evaluation of the credentials to determine whether the request is consistent with the policy. While in the example shown earlier we show requests involving three parties, this is not always the case. For example, one or more credit institutions may be used to link the content provider with various ISPs. For this reason, network elements may receive requests containing chains of two or more credentials and hence expend more computational power in order to evaluate them. Table I shows how the addition of credentials affects the overall processing of requests (the numbers originated from tests run on a Dell PowerEdge 1550 and represent average times, in microseconds, over 100 trials). We show the time taken to process requests which include 3 to 7 credentials with the policy engine active (column 2) and with the policy engine replaced with a routine that always returns true (column 3). The difference between the two measurements (column 4) shows the cost of credential evaluation which is significant. The request processing time increases even if the policy engine is disabled, because the system must still parse and process the larger requests.

Packet Size	Average Time With BandEx	Average Time Without BandEx	Average Time Difference
2612	2793.14	248.96	2544.18
3548	3479.87	261.3	3218.57
5116	4753.54	304.14	4449.4
6356	5518.06	321.46	5196.6
7620	6721.52	337.72	6383.8

TABLE I

Request processing cost *vs.* **number of credentials used (μ-secs)**

Even though these operations are relatively expensive, the impact of the overhead is minimal since (a) such operations occur only when the bandwidth is initially allocated, (b) the cost is distributed among the network elements, so that if a new reservation request affects n network elements, the total time for the operation is that of the slowest element, and (c) the number of credentials in a chain depends on the parties involved (*i.e.,* the number of middlemen in the transaction), so it is unlikely to be higher than 4 to 5.

To avoid computational denial of service attacks, wherein attackers send bogus reservation requests to overlay nodes and routers, ISPs must allow only their users to use BandEx (*i.e.,* filter RSVP requests coming from outside the ISP perimeter). Misbehaving local users can be detected and quarantined through credential revocation or other mechanisms.

V. Conclusion

We proposed the concept of *Overlay-linked IntServ (OLIntServ),* a system architecture that combines network overlays with intra-domain QoS to provide assured communications over the Internet, while allowing ISPs to extend the reach of their currently under-utilized IntServ services. We describe our prototype system architecture that uses our SpreadSpectrum SOS overlay network [12] and the BandEx secure network reservations system [4]. By combining the two systems, it is possible to provide a secure, highly available, disruption-tolerant end-to-end path without requiring end-to-end (across multiple ISPs) availability of QoS reservation primitives such as DiffServ. Our preliminary experimental results show that this approach promises to help bridge the gap between inter- and intra-ISP network reservations. Our plans for future work include building and deploying a full system prototype, conducting further experiments using real-time traffic and delay-sensitive applications, and examining the impact of smart attacks against the system.

References

[1] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The End-to-End Effects of Internet Path Selection," in *Proceedings of ACM SIGCOMM*, September 1999.

[2] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient Overlay Networks," in *Proceedings of ACM SOSP*, October 2001.

[3] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure Overlay Services," in *Proceedings of ACM SIGCOMM*, August 2002, pp. 61–72.

[4] D. M. Turner, V. Prevelakis, and A. D. Keromytos, "The Bandwidth Exchange Architecture," in *Proceedings of the* 10^{th} *IEEE Symposium on Computers and Communications (ISCC)*, June 2005, pp. 939–944.

[5] H. Guy, "Everything you ever wanted to know before you use and/or deploy UCLP on your advanced network," in *Proceedings of the CANARIE's Advanced Networks Workshop*, November 2004.

[6] W.Jarrett, T.Michalareas, and L.Sacks, "Operational Support Issues for IP QoS Based Networks," in *Proceedings of the IEE Services over the Internet Colloquium*, June 1999.

[7] R. J. Edell, N. McKeown, and P. Varaiya, "Billing Users and Pricing for TCP," *IEEE JSAC*, vol. 13, no. 7, pp. 1162–1175, 1995.

[8] L. Zhang, S. Deering, and D. Estrin, "RSVP: A new resource ReSerVation protocol," *IEEE Network*, vol. 7, no. 5, September 1993.

[9] M. Talwar, "RSVP Killer Reservations, IETF Draft (draft-talwar-rsvp-kr-01.txt)," 1999. [Online]. Available: http://www.isi.edu/rsvp/DOCUMENTS/draft-talwar-rsvp-kr-01.txt

[10] A. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante, "Drafting Behind Akamai (Travelocity-Based Detouring)," in *Proceedings of ACM SIGCOMM*, September 2006, pp. 435–446.

[11] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. N. Rao, "Improving Web Availability for Clients with MONET," in *Proc. of NSDI*, May 2005.

[12] A. Stavrou and A. Keromytis, "Countering DoS Attacks With Stateless Multipath Overlays," in *Proc. of the* 12^{th} *ACM Conference on Computer and Communications Security (CCS)*, Nov. 2005, pp. 249–259.

[13] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, "The KeyNote Trust Management System Version 2," RFC 2704, Sept. 1999.