Structured Overlay Networks for a New Generation of Internet Services

Amy Babay, Claudiu Danilov, John Lane, Michal Miskin-Amir,
Daniel Obenshain, John Schultz, Jonathan Stanton, Thomas Tantillo, Yair Amir

Johns Hopkins University — {babay, dano, tantillo, yairamir}@cs.jhu.edu
LTN Global Communications — {johnlane, michal, jschultz, jonathan, yairamir}@ltnglobal.com
Spread Concepts LLC — {michal, jschultz, jonathan, yairamir}@spreadconcepts.com
Boeing Research & Technology — {Claudiu.B.Danilov}@boeing.com

***Abstract*—The dramatic success and scaling of the Internet was made possible by the core principle of keeping it simple in the middle and smart at the edge (or the end-to-end principle). However, new applications bring new demands, and for many emerging applications, the Internet paradigm presents limitations.**

For applications in this new generation of Internet services, structured overlay networks offer a powerful framework for deploying specialized protocols that can provide new capabilities beyond what the Internet natively supports by leveraging global state and in-network processing. The structured overlay concept includes three principles: A resilient network architecture, a flexible overlay node software architecture that exploits global state and unlimited programmability, and flow-based processing.

We demonstrate the effectiveness of structured overlay networks in supporting today's demanding applications and propose forward-looking ideas for leveraging the framework to develop protocols that push the boundaries of what is possible in terms of performance and resilience.

I. Introduction

The dramatic success and scaling of the Internet over the past five decades was made possible by the core principle of keeping it simple in the middle and smart at the edge (or the *end-to-end principle*). The simplicity of the network core makes it easy to scale and to add new applications, as all applications can be treated in the same manner: the core of the network is only responsible for best-effort packet switching.

However, new applications bring new demands, and for many new and emerging applications, the Internet paradigm presents limitations. Considering two broad classes of applications, video transport and monitoring and control of global clouds, we demonstrate that breaking the end-to-end principle and placing resources and intelligence in the middle of the network can support such applications when the native Internet cannot.

For these applications and others in this new generation of Internet services, *structured overlay networks* offer a powerful framework for deploying specialized protocols that can provide new capabilities beyond what the Internet supports by leveraging global state and in-network processing. By deploying overlay nodes with general-purpose computing resources in the middle of the network, this approach breaks the end-to-end principle at the overlay level, requiring no changes in the underlying network (i.e. the Internet) and allowing it to keep the same scalable design that has made it so successful. The structured overlay concept includes three principles:

- A resilient network architecture that leverages multiple Internet Service Provider (ISP) backbone networks for increased availability, resiliency, and predictability of service by instantiating overlay nodes in strategic data centers.
- An overlay node software architecture that maintains global state that is shared between all overlay nodes and updated in a timely manner, and that exploits unlimited programmability provided by general-purpose computers. This allows implementing services that provide properties such as exacting timeliness and resilience guarantees. This software architecture can be easily extended to support new overlay protocols that address new application demands.
- Flow-based processing, where packets are processed according to context associated with the particular flow to which they belong and the service required by that flow, in contrast to stateless forwarding of packets based on their destination. Flow-based processing enables, for example, hop-by-hop recovery within the overlay and redundant dissemination with corresponding de-duplication in the middle of the network.

The structured overlay approach provides a cost-effective solution for addressing new application demands, compared with alternative approaches such as building specialized (non-IP) networks, creating private IP networks, and extending the Internet infrastructure to natively support the new demands.

Specialized networks were built in the past to support special needs of high-value applications. A good example is the cable TV infrastructure for video distribution into homes. This is a very expensive proposition, and, in fact, the ubiquity of the Internet and its expanded capacity and attractive cost renders these networks obsolete over time. Creating a private IP network eliminates contention with other applications on the Internet and therefore allows more predictable service. However, this approach has two limitations: it is expensive, and it is limited by the basic end-to-end principle underlying

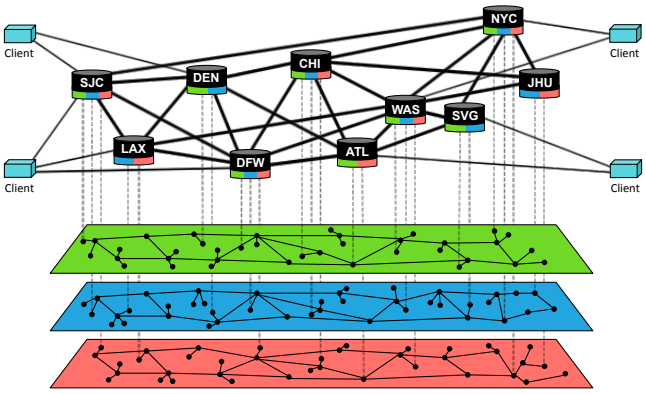

Fig. 1. Resilient Network Architecture

the IP service. Finally, extending the Internet infrastructure requires a long process of standardization and gradual adoption. Beyond that, extensions to the Internet protocol need to account for its scalability requirements and the many environments it must support, greatly limiting flexibility.

In contrast, the structured overlay framework can be practically deployed using general-purpose computers, commodity data centers, and cost-effective access bandwidth provided by multiple ISPs. Moreover, there is no need for standardization: to the underlying network, an overlay looks like a normal user-level application.

We demonstrate the effectiveness of structured overlay networks in supporting today's demanding applications and propose forward-looking ideas for leveraging the power of the same overlay framework to develop innovative protocols to support emerging applications that push the boundaries of what is possible in terms of performance and resilience.

While we have been working on this vision for the past 17 years in both the research and commercial domains, this paper presents a holistic view of our structured overlay network concept for the first time.

The rest of the paper is organized as follows: Section II describes the key principles underlying the structured overlay framework, Section III describes basic applications using this framework, Section IV describes more advanced applications that require more sophisticated programmability and resources, Section V discusses emerging and future applications that require new capabilities and protocols within the structured overlay framework, Section VI puts the framework in the context of related work, and Section VII concludes the paper.

II. Structured Overlay Framework

Structured overlay networks create logical networks that run on top of the Internet. Three key principles support the powerful capabilities of the structured overlay: a resilient network architecture, software overlay routers with unlimited programmability, and flow-based processing. We describe each of these three principles below.

A. Resilient Network Architecture

The physical architecture supporting the structured overlay network enables its unlimited programmability, global state maintenance, fast reaction, and resilience. To support such capabilities, the physical architecture is constructed based on a resilient network architecture, illustrated in Figure 1.

The structured overlay network consists of overlay nodes connected to each other via overlay links (logical edges). Overlay nodes are physically instantiated as general-purpose computers residing in data centers, while the overlay links correspond to Internet paths between the overlay nodes. The use of general-purpose computers provides unlimited programmability, enabling a wide range of current and future applications with highly demanding requirements.

A key property of structured overlay networks is that they require only a few tens of well situated overlay nodes to provide excellent global coverage. This is because, in general, placing overlay nodes about 10ms apart on the Internet provides the desired performance and resilience qualities (discussed below), and about 150ms is sufficient to reach nearly any point on the globe from any other point. The limited number of nodes allows each overlay node to maintain global state concerning the condition of all other overlay nodes and the connections between them, allowing fast reactions to changes in the network, with the ability to route around problems at a sub-second scale. This is in contrast to the 40 seconds to minutes that BGP may take to converge during some network faults.

To make this sub-second rerouting possible, overlay networks exploit redundancy in the resilient network architecture. As shown in Figure 1, in such an architecture, each overlay node is connected to each other node through multiple redundant paths at the overlay level, and is connected to multiple underlying ISP backbones. This redundant architecture allows the overlay to change the underlying network path used for data transmission without relying on rerouting at the Internet level. This is accomplished by selecting a different overlay-level path or by choosing a different combination of ISPs to use for a given overlay link.

For overlay-level rerouting to be effective, disjointness in the overlay paths should reflect physical disjointness in the underlying networks: if different overlay paths overlap in the underlying network, a single problem in the underlying network can affect multiple overlay paths. To exploit physical disjointness available in the underlying networks, the overlay node locations and connections are selected strategically.

Overlay nodes are placed in well-provisioned data centers, as ISPs invest in such locations by laying independent fiber connections between them. The overlay topology can then be designed in accordance with the underlying network topology, based on available ISP backbone maps. Overlay links are designed to be short (on the order of 10ms) so that the Internet routing between overlay neighbors (i.e. overlay nodes connected by a direct overlay link) is relatively predictable. Short overlay links also enable improved performance and

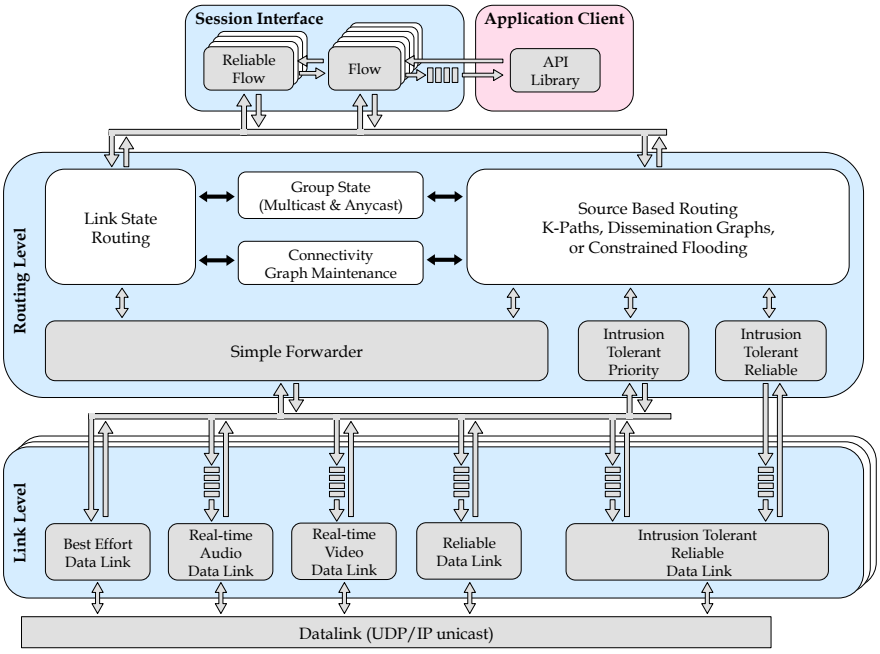

Fig. 2. Overlay Node Software Architecture

services by breaking the end-to-end principle at the overlay level and increasing the processing possibilities in the middle of the network, as discussed in Section III-A. Because short overlay links are preferred, it is not normally advised to build a continent- or global-sized overlay as a clique.

Connecting each overlay node on multiple ISPs provides additional redundancy and resilience. Multihoming in this way allows the overlay to route around problems affecting a single provider and allows most traffic to avoid BGP routing by traversing only *on-net* links (i.e. overlay links that use the same provider at both endpoints), which generally results in better performance (although any combination of the available providers may be used, if desired).

B. Overlay Node Software Architecture

The overlay software runs on each overlay node as a normal user-level program. Here, we describe our vision for the overlay node software architecture that can support demanding applications today and can be extended to provide new services to meet the needs of future applications. Key properties of the software architecture include the use of general computing resources, a two-level (client-daemon) hierarchy, state sharing between overlay nodes, and a flexible and extensible design that facilitates adding and recombining different overlay protocols. These key properties are supported by the resilient network architecture described above.

Because overlay nodes are physically instantiated in general purpose computers, the overlay software is able to leverage general purpose computing resources. For example, the overlay software can make use of the physical computer's ample memory to store sent messages for later retransmissions or to track received messages to allow de-duplication of retransmitted or redundantly transmitted messages. Similarly, the arbitrary processing possible in a general-purpose computer allows for sophisticated network protocols and more advanced features like cryptographic processing.

The overlay software is designed as a two-level hierarchy using a client-daemon architecture. Overlay nodes act as both servers and routers: as servers, they accept and serve client connections, while as routers they perform network functions such as forwarding packets destined for other overlay nodes. To receive service from the overlay, a client simply connects to an overlay node (a client may run on the same physical machine as the overlay node software or on a remote machine).

The overlay node software architecture we envision is shown in Figure 2. This architecture consists of three levels: the session interface, the routing level, and the link level. The *session interface* is responsible for managing client connections, with each client connection treated as a separate flow. Each client specifies the particular overlay services that should be used for its flow. The *routing level* makes decisions about how to forward incoming packets based on the routing service specified for the flow (Link State or Source Based), the current state of the network (obtained via the Connectivity Graph Maintenance component), and the packet's source and destination or destinations (with multicast group membership maintained by the Group State component). The *link level* transmits the packet on the relevant overlay link or links determined by the routing level, using the link level protocol specified for the flow (e.g. Best Effort, Real-time Audio, etc.).

A key feature of the software architecture is its support for state sharing among the overlay nodes. In Figure 2, the

Connectivity Graph Maintenance and *Group State* components represent two types of shared state. The Connectivity Graph Maintenance component enables fast rerouting in response to changes in network conditions by allowing the overlay nodes to share information about their connections to neighboring overlay nodes. This information can include the current loss and latency characteristics of the overlay links. The Group State component enables multicast and anycast capabilities that are generally not available on the Internet: all of the overlay nodes share information about whether they have clients interested in a particular multicast group, making it possible to disseminate multicast messages to all relevant nodes or to select the best target for a given anycast message (as anycast messages are delivered to exactly one member of the relevant group). The two-level hierarchy makes this state sharing practical by allowing each overlay node to track only which of its own connected clients are members of a particular group and which other overlay nodes are relevant to that group; an overlay node does not need to maintain any information about clients connected to the other overlay nodes.

Another key feature of the software architecture is its flexible design that allows many different routing-level and link-level protocols to coexist and facilitates adding new protocols at both levels. The architecture shown in Figure 2 includes two classes of routing protocols: Link State Routing and Source Based Routing. In Link State Routing, an overlay node determines how to forward incoming packets based on their destination and its current knowledge of the network state, similar to link-state routing on the Internet. In Source Based Routing, the overlay node introducing a message into the network stamps it with the path it should traverse to reach its destination. This can be implemented via a unified source-based routing mechanism in which each packet is stamped with a bitmask indicating exactly the set of overlay links it should traverse (where each bit in the bitmask represents an overlay link). This mechanism enables routing schemes that are not possible on the Internet, including the use of multiple node-disjoint paths, arbitrary subgraphs of the overlay topology (*dissemination graphs*), or constrained flooding on the overlay topology [1], [2].

Figure 2 also shows several link-level protocols that offer a range of timeliness, reliability, and resilience guarantees. Client applications can select the combination of routing and link protocols that best supports their particular demands, and new protocols can be easily added to support new applications. In Sections III - V we discuss example applications and protocols to support them. A single overlay node can serve many clients (with the clients potentially using different combinations of protocols), and multiple overlays can even be run in parallel (with each overlay potentially using a different variant of the overlay software).

The overlay software interface looks like a normal application to the underlying network and like a powerful network (with additional services) to the applications that use it. Applications can either connect to the overlay via an API similar to the Unix sockets interface or use seamless packet interception techniques that allow unmodified applications to take advantage of overlay services. Clients are identified by the IP address of the overlay node to which they connect and a virtual port, mimicking the IP address plus port addressing scheme of the Internet. Anycast and multicast are implemented similarly as part of the IP space, just like in IP.

The architecture in Figure 2 is inspired by our experience with the Spines open-source overlay messaging framework [3] and its derivatives, which realize a similar architecture and implement many of the protocols described.

C. Flow-Based Processing

In contrast to the Internet's stateless packet switching, the structured overlay networks of our vision employ flow-based processing. Flows may be point-to-point (unicast or anycast) or point-to-multipoint (multicast). From a client's perspective, a flow consists of a source, one or more destinations, and the overlay services selected for that flow. A client can select different overlay services (e.g. routing and link protocols) for each application data flow.

The overlay node's access to ample memory and processing resources allows it to maintain the flow-based state needed to support basic services like reliability, as well as more advanced services like authentication. Within the overlay, application data flows may be aggregated based on their source and destination overlay nodes or the services they select, with state maintenance and processing performed on the aggregate flows.

D. Cost and Deployment Considerations

The primary cost of the structured overlay concept is the need to deploy and manage processing resources (in the form of general-purpose computers) in the data centers hosting overlay nodes.

Surprisingly, the latency costs of structured overlay networks are small: since overlay node locations are carefully selected, as discussed in Section II-A, the latency overhead of using a multi-hop indirect overlay path rather than the direct Internet path is small. Furthermore, the computational costs to traverse up and down the network stack at overlay nodes on today's commodity computers amount to less than 1ms additional latency per intermediate overlay node on the path (while the propagation delay to cross a continent is on the order of 35-40ms).

However, depending on the traffic load, a single computer may not be able to provide the necessary processing at line speed. To deal with this issue, additional processing resources can be deployed as clusters of computers running in the data centers. Each computer in a cluster can act as a node in one or several overlays, serving a subset of the total traffic. Beyond the costs of management and processing, the structured overlay approach incurs additional costs for hosting in the data centers and for bandwidth from multiple Internet service providers.

III. Basic Applications

A. Broadcast-Quality Video Transport

As a concrete example of how the basic structured overlay concept enables new services that are not well supported

by the native Internet, we first consider broadcast-quality video transport. This service requires reliably transmitting continuous video streams to multiple endpoints.

Delivering the streams to multiple endpoints efficiently requires a multicast capability that is not practically available on the Internet, but is possible at the overlay level (exploiting the shared state and two-level hierarchy discussed in Section II-B).

Moreover, continuous transmission requires a service with higher availability and quality of service than the Internet can natively provide. Waiting tens of seconds to minutes for Internet routing to converge after a failure is not acceptable. Video transport can take advantage of the overlay's sub-second rerouting to provide the needed availability. To provide smooth reliable delivery of video packets, a hop-by-hop overlay recovery protocol (*Reliable Data Link* in Figure 2) can be used [4].

The hop-by-hop recovery protocol takes advantage of the fact that overlay links are generally short, as discussed in Section II-A. The resilient network architecture used to construct the overlay essentially replaces a high-latency end-to-end network path with a series of low latency overlay links. By adding automatic repeat request (ARQ) mechanisms to each overlay link, the overlay can localize and recover losses much faster and with lower overhead than an end-to-end approach. To provide smoother packet delivery, intermediate nodes are permitted to forward packets out of order; the final destination is responsible for buffering received packets until they can be delivered in order. This hop-by-hop recovery and out-of-order forwarding on the overlay can significantly reduce the latency and jitter of reliable communication, as it speeds and smoothes final packet delivery.

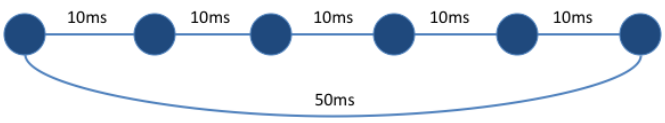

Fig. 3. 50ms network path vs. five 10ms overlay links

As an illustrative example, consider a symmetric network path that spans a continent with a one-way latency of 50ms, as illustrated in Figure 3. With ARQ protocols, reactively recovering a lost packet takes at least one round trip for the receiver to request and receive a retransmission. Therefore, a packet recovered end-to-end has at least 100ms of additional latency for a total minimum latency of 150ms. If that network path can be replaced with a series of five 10ms latency overlay links using hop-by-hop recovery, then a recovered packet has only at least 20ms additional latency for a total minimum latency of 70ms. Using this hop-by-hop recovery approach, the overlay is able to provide a reliable service with better timeliness characteristics and a smoother delivery pattern than an end-to-end service.

B. Resilient Monitoring and Control

As another example, we consider monitoring and control of global clouds. Like video transport, cloud *monitoring* requires transmitting continuous data streams to multiple destinations, although in this case timely delivery of the latest data is more important than completely reliable transmission. Cloud *control* requires reliably transmitting commands that may change the state of the cloud to one or more destinations.

Although the service required for monitoring is somewhat different than that needed for control, the flexible overlay software architecture can support both simultaneously. While a completely reliable link-level protocol is needed for control messages (e.g. the Reliable Data Link), a protocol that guarantees timeliness in all cases may be more appropriate for monitoring messages. In both cases, the overlay is able to provide better overall performance than the native Internet by using protocols that leverage processing in the middle of the network.

Both cloud monitoring and control use overlay-level multicast to disseminate information to multiple destinations, and this capability greatly simplifies the monitoring architecture: rather than needing to connect each of many endpoints being monitored to each of several destinations that need to receive the monitoring streams, each endpoint simply connects to the overlay, joining or sending to the relevant multicast groups. Only receivers need to join the multicast group (any client can send to the group), and the overlay is able to construct the most efficient multicast tree to route messages to all overlay nodes that have clients in the group. Common destinations for multicast monitoring data include displays, logging processes, and realtime analysis engines (e.g. that use machine learning to predict problems based on patterns). The overlay provides mesh connectivity for all destinations without requiring each endpoint to create multiple connections.

IV. Advanced Concepts and Applications

As applications become more advanced and demanding, the need for computation and resources inside the network increases. Here, we discuss more advanced applications and show how the same structured overlay vision can support these applications by introducing new protocols and processing capabilities within the same software architecture.

A. Live Broadcast-Quality Video Transport

Live broadcast-quality video transport represents a significantly more demanding application than the video transport discussed in Section III-A. It has the same availability, reliability, and multicast demands, but introduces a much more stringent timeliness requirement. For example, in interviews with remote studios, timely delivery within about 200ms is critical to support natural interaction between the participants and provide the illusion that they are in the same place.

To support such a service on a continent scale, an overlay link protocol can be tailored to deliver packets within the required latency bound. The NM-Strikes protocol (illustrated in Figure 4) is a real-time protocol that, while not guaranteeing complete reliability, guarantees complete timeliness [5]. Because of the burstiness of loss on the Internet, the challenge is to bypass the window of correlation for loss within the allotted time. The base of the protocol is to send each packet

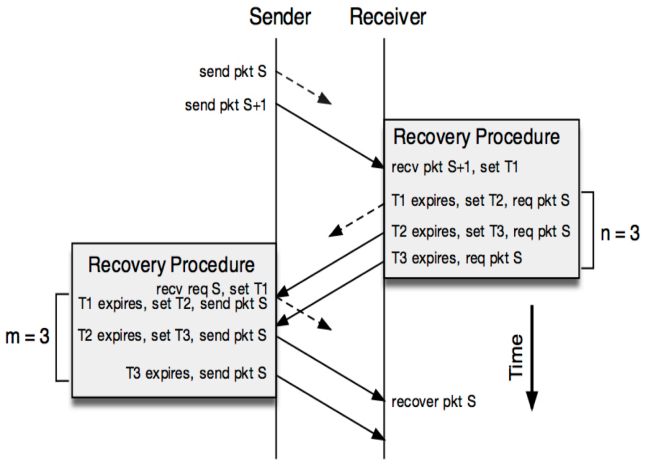

Fig. 4. NM-Strikes protocol for live video transport

with a sequence number. When the receiver detects an out of order packet, it schedules N retransmission requests for each missing packet. These requests are scheduled at different times in the future to reduce the probability that all of the requests are affected by the same correlated loss event on the network. The requests should be spaced out as much as possible, but not so much that the deadline is not met. The sender, upon receipt of the first request for a retransmission, will schedule M retransmissions, also spaced to avoid correlated loss. Ideally, the timers will be set such that even the M^{th} (final) response to the N^{th} request will still reach the destination on time. On the scale of a continent with a 40ms propagation delay, the 200ms latency bound allows about 160ms for the protocol to recover lost packets. A receiver that receives a requested packet can cancel any remaining scheduled requests for that packet.

To provide ordered delivery, the final destination must build a buffer that will rearrange the packets that were recovered into the correct order. If a recovered packet arrives after later packets were already delivered, it is discarded.

The overall cost of the NM-Strikes protocol (on the sender to receiver side) is $1 + Mp$, where p is the loss rate, since in the worst case, each missed packet will be retransmitted M times. This protocol is fairly complex, requiring memory and relatively sophisticated processing on the flow. State must be maintained for each flow to know when packets should be retransmitted and requested. While such a capability is not available in routers and switches, it is relatively easy to implement in a general-purpose computer.

B. Intrusion-Tolerant Monitoring and Control

The resilient monitoring and control application presented in Section III-B uses the structured overlay approach to achieve resilience to failures, compromises, or attacks on the underlying network by making use of the redundancy in the overlay's resilient network architecture and its sub-second rerouting capability. However, monitoring and control of high-value infrastructure may require a stronger service that can also withstand attacks on the overlay itself, including compromises of overlay nodes. Such applications can be supported by employing more advanced processing and state maintenance in the overlay.

Because the number of overlay nodes is small, each overlay node can know the identities of all valid overlay nodes in the system, and can use cryptography to authenticate messages and ensure that they originate from authorized overlay nodes. However, authentication is not sufficient to protect against an attacker that compromises an overlay node, thus gaining access to that node's credentials.

The intrusion-tolerant overlay messaging service ensures that compromised overlay nodes cannot prevent messages sent by correct overlay nodes from reaching their destination (provided that some correct path through the overlay still exists) [1]. This is done using redundant dissemination schemes to ensure that messages reach correct overlay nodes and fairness schemes to ensure that correct nodes will forward messages from all sources fairly, even if compromised nodes launch resource consumption attacks.

The redundant dissemination schemes use the source-based routing capability described in Section II-B. By using k node-disjoint paths, a source can protect against up to $k - 1$ compromised nodes anywhere in the network (since each compromised node can disrupt at most one of the k paths). Alternatively, a source can use constrained flooding, which sends each packet on all links in the overlay topology. Constrained flooding ensures that messages are successfully delivered as long as at least one path of correct nodes exists between the source and destination.

The fair scheduling schemes use flow-based state to store messages and enforce fairness in message forwarding. In Figure 2, two scheduling schemes are shown: Intrusion Tolerant Priority and Intrusion Tolerant Reliable. *Priority* messaging matches the needs of monitoring messages that require timely service (while being as reliable as possible, given current network conditions), while *Reliable* messaging matches the needs of control messages that must be delivered with complete end-to-end reliability (while being as timely as possible). Both Priority and Reliable messaging use fair buffer allocation and round-robin scheduling to ensure that a compromised source cannot consume the resources of other sources to prevent their messages from being forwarded.

Priority messaging maintains storage per source and treats each active source in a round-robin manner when selecting the next message to forward on a given outgoing link. Sources assign priorities to their messages, and if a node's storage for a particular source fills, additional messages from that source will cause the oldest lowest priority message for that source to be dropped to permit timely delivery of the highest priority messages.

Reliable messaging maintains storage per source-destination flow (so a compromised destination cannot block a source) and treats each active flow in a round-robin manner. When a node's storage for a particular flow fills, it stops accepting new messages for that flow, creating backpressure (potentially all the way back to the source).

V. Emerging and Future Applications

Current research uses the same structured overlay concept, including the resilient network architecture, extensible overlay node software architecture, and flow-based processing, to develop innovative capabilities to support emerging and future applications. Such applications require highly demanding combinations of timeliness, reliability, and resilience.

A. Real-Time Remote Manipulation

Remote manipulation of physical objects for applications such as remote robotic surgery or remote robotic ultrasound requires both meeting strict timeliness requirements and providing high reliability. The level of interactivity required for such applications, which may involve both visual and haptic feedback, is considerably more demanding than even the timeliness required for live TV (Section IV-A).

For interaction to feel natural in these applications, the roundtrip latency (i.e. the time between initiating an action, affecting the remote object, and receiving the feedback from the remote object) must be no more than about 130ms, translating to a one-way latency requirement of 65ms. On the scale of a continent, where propagation delay may be around 40ms, this leaves only 20-25ms of flexibility for buffering or recovery of lost packets, in contrast to the 160ms available when the goal is to meet the 200ms one-way latency requirement of live TV.

This strict deadline reduces the effectiveness of recovery protocols like the NM-Strikes protocol discussed in Section IV-A. Therefore, to provide a highly reliable service, a new approach is needed. Ongoing research aims to solve this problem by combining a predecessor of the NM-Strikes protocol that only allows one request and one retransmission per lost packet [6], [7] with a redundant dissemination scheme that uses specialized *dissemination graphs* [2]. Dissemination-graph-based routing uses the overlay's source-based routing capability to send messages over an arbitrary subgraph of the overlay topology. In contrast to disjoint paths, which add redundancy uniformly throughout the network, dissemination graphs can be tailored based on current network conditions to add targeted redundancy in problematic areas of the network.

B. Monitoring and Control of Critical Infrastructure

Due to the importance of critical infrastructure systems, their monitoring and control systems must be resilient to sophisticated attacks and compromises (similarly to the intrusion-tolerant monitoring and control discussed in Section IV-B). However, certain critical infrastructure control systems, such as SCADA for the power grid, require strict timeliness, on the order of 100-200ms for a control command to be delivered and executed in response to received monitoring data.

For the control system to withstand compromises, this 100-200ms can include the time to execute an intrusion-tolerant agreement protocol to ensure that the correct control command is issued. Because such protocols typically include multiple rounds of authenticated message exchanges, this combination of requirements is particularly challenging to support and is likely to require new protocols within the structured overlay framework. In particular, the cryptography required to support intrusion tolerance today becomes a barrier to timely message delivery as the size of the system grows, and critical infrastructure systems may monitor many devices in the field. The specific requirements of critical infrastructure systems and techniques to support them are the subject of current research.

C. Compound Flows

The unlimited programmability enabled through the use of general-purpose computers as overlay nodes opens up new possibilities for sophisticated in-network processing and transformation of flows. This has the potential to be useful for a wide range of applications; an initial use being developed today is for video transcoding in the cloud. As an example, a video stream of a live sports event is sent from the stadium as a broadcast-quality MPEG transport stream on the overlay and delivered to several sports network destinations that carry it through the cable networks to the home. One of the destinations of the transport stream can be a transcoding facility in the cloud that transcodes the signal to different formats and quality levels and transports it to CDNs and social media sites for delivery to mobile devices.

Reliability and timeliness guarantees must be met throughout the entire compound flow, including its transformation. Network conditions and failures may lead to rerouting that can include the selection of a transcoding facility at a different location.

VI. Related Work

Pioneering overlay network systems include X-Bone [8], which facilitates instantiating overlays over IP networks, and RON [9], which provides robust routing around Internet path failures. Other overlay approaches that improve on the performance and quality of service of the Internet include OverQoS [10], which offers statistical loss and bandwidth guarantees, using a combination of forward error correction (FEC) and packet retransmissions, and other work using redundant dissemination schemes, such as multiple disjoint paths [11], [12] or sets of potentially overlapping paths [13].

Multicast is a necessary capability for many of the applications we discuss. Initial efforts to provide multicast services focused on the IP level, resulting in the basic IP-multicast service [14]. While IP-multicast is scalable in the number of users per group, it is not scalable in the number of groups. Moreover, it uses a single addressing scheme for the entire Internet and was disabled by commercial ISPs, although it is used today in private networks to implement an IPTV service. The MBone [15], [16] provided a means of connecting individual multicast-enabled networks to create a multicast service over the Internet. However, because the MBone relied on IP-multicast with its single global addressing scheme, it was ultimately not practical.

Overlay approaches that provide application-layer multicast have generally been more successful and include the NICE

protcol [17], in which peers are arranged hierarchically such that every peer receives data from its parent or siblings and forwards the data to its children and siblings, and its extension to multiple parallel overlays to distribute traffic in video content distribution applications [18]. Content Delivery Networks (CDNs) such as Akamai [19] represent an alternative approach, caching video and other content as files stored at many widely distributed proxy servers. The content can then be distributed to a large number of end users with high availability and performance. However, a caching approach (as opposed to the flow-based approach of structured overlays) does not support real-time guarantees for highly interactive emerging applications.

The problem of constructing networks that are resilient to attacks and even compromises has been investigated from several perspectives. One of the first solutions in this space was the work of Radia Perlman, which used public-key authenticated flooding of link-state updates and separate buffers per node to ensure correct routing in the presence of compromised nodes in a single physical network [20]. SCION [21] secures and protects Internet routing by organizing Autonomous Systems (ASes) into Isolation Domains (ISDs) and protecting communication between any pair of ISDs from interference by external ISDs. While the work of Perlman and SCION both provide solutions, they have significant barriers to deployment, requiring changes to IP or cooperation with ISPs. In contrast, structured overlay networks can be deployed over the existing Internet infrastructure without any coordination with ISPs.

ODSBR combines shortest path routing and disguised probing techniques to localize faults and provide routing resilient to compromised nodes [22]. This approach could be implemented within a structured overlay framework to provide an alternative intrusion-tolerant messaging service that presents a different trade-off between timeliness and cost compared with the approach in Section IV-B.

While peer-to-peer (P2P) overlay networks (surveyed in [23]) build logical networks on top of the Internet, they differ considerably from the structured overlay networks we describe. Many P2P overlays target file-sharing applications, with the goal of providing efficient lookup of resources in a dynamic environment where overlay peers can join and leave relatively frequently, and thus solve a different set of problems than we consider. However, some P2P systems more relevant to the applications we discuss target applications like live media streaming. In general, P2P overlays aim to scale to a large number of peers in self-organizing, server-less architectures. In contrast, the structured overlays we describe are based on a small set of well provisioned overlay nodes that act as servers connected in a carefully designed overlay topology that changes infrequently. A large number of clients can connect to the servers of the structured overlay (with each client typically connecting to the closest overlay server). The investment associated with structured overlays, if feasible, supports better performance and resilience.

MPLS [24] provides a protected virtual circuit capability with multiple label switched paths over a single provider IP network. This provides bandwidth allocation and prioritization to traffic classes (i.e. flows). MPLS enables IP multicast routing to work within the MPLS virtual network between all the sites that participate in that private network. Normally, an enterprise will contract with an ISP to provide an MPLS service between all of that enterprise's sites. In such a case, the enterprise is able to use IP multicast between its sites, and is able to define prioritized flows originating at and delivered to its sites. The MPLS routers only provide packet forwarding and are not able to support higher-level services such as hop-by-hop reliability, packet de-duplication, or message authentication, that require significant processing and state maintenance in the router.

Software Defined Networking (SDN) represents an alternative and complementary approach to improving network capabilities with enhanced programmability (see [25] for an overview). In contrast to overlay networks that provide unlimited programmability to support complex flow-based processing (e.g. for recovery protocols, authentication, and fair forwarding) and even more advanced flow transformations (e.g. for video transcoding), SDN has, so far, largely focused on the separation of the control and data planes and control plane innovations that simplify network management.

MPLS and SDN both enhance services at the network level, in contrast to structured overlays, which operate at the overlay level. As such technologies continue to evolve to support new capabilities at the network level (e.g. Segment Routing [26]), it will be interesting to see how they can interact with structured overlays to support emerging and future Internet applications. For example, as network-level enhancements become more powerful, some of the capabilities provided by structured overlays today may be able to migrate to the network level to reduce cost and improve performance. Future applications may be able to combine all these technologies in innovative ways to construct even more advanced services with new capabilities.

VII. Conclusion

We have presented a structured overlay approach to support present and future applications with requirements that cannot be met on the Internet. Our framework is based on three key principles: a resilient network architecture, software routers with unlimited programmability, and flow-based processing. Our structured overlay approach puts intelligence in the middle of the network by using software routers in overlay nodes, hosted in strategic data centers and served by several ISP backbone networks, to support a demanding new generation of Internet services.

Acknowledgment

This work was supported in part by NSF grant 1535887 and by DARPA grant N660001-1-2-4014. Its contents are solely the responsibility of the authors and do not represent the official view of the NSF or DARPA.

REFERENCES

[1] D. Obenshain, T. Tantillo, A. Babay, J. Schultz, A. Newell, M. E. Hoque, Y. Amir, and C. Nita-Rotaru, "Practical intrusion-tolerant networks," in *Proceedings of the 36th International Conference on Distributed Computing Systems (ICDCS)*, June 2016, pp. 45–56.

[2] A. Babay, E. Wagner, M. Dinitz, and Y. Amir, "Timely, reliable, and cost-effective internet transport service using dissemination graphs," in *Proceedings of the 37th International Conference on Distributed Computing Systems (ICDCS)*, June 2017.

[3] "The Spines Messaging System," www.spines.org, access: 2017-03-28.

[4] Y. Amir and C. Danilov, "Reliable communication in overlay networks," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. and Networks*, June 2003, pp. 511–520.

[5] Y. Amir, J. Stanton, J. Lane, and J. Schultz, "System and method for recovery of packets in overlay networks," U.S. Patent 8 437 267, May, 2013.

[6] Y. Amir, C. Danilov, S. Goose, D. Hedqvist, and A. Terzis, "1-800-OVERLAYS: Using overlay networks to improve VoIP quality," in *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, 2005, pp. 51–56. [Online]. Available: http://doi.acm.org/10.1145/1065983.1065997

[7] ——, "An overlay architecture for high-quality VoIP streams," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1250–1262, Dec 2006.

[8] J. Touch, "Dynamic internet overlay deployment and management using the x-bone," *Computer Networks*, vol. 36, no. 23, pp. 117 – 135, 2001, theme issue: Overlay Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128601001724

[9] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. Symp. Operating Syst. Principles*, 2001, pp. 131–145. [Online]. Available: doi.acm.org/10.1145/502034.502048

[10] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz, "OverQoS: An overlay based architecture for enhancing internet QoS," in *Proceedings of the 1st Symposium on Networked Systems Design and Implementation (NSDI)*, 2004, pp. 71–84.

[11] A. C. Snoeren, K. Conley, and D. K. Gifford, "Mesh-based content routing using XML," in *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP)*, 2001, pp. 160–173. [Online]. Available: http://doi.acm.org/10.1145/502034.502050

[12] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2003, pp. 91–100. [Online]. Available: http://doi.acm.org/10.1145/948205.948218

[13] K. Karenos, D. Pendarakis, V. Kalogeraki, H. Yang, and Z. Liu, "Overlay routing under geographically correlated failures in distributed event-based systems," in *On the Move to Meaningful Internet Systems*, 2010, pp. 764–784.

[14] S. Deering, "Host extensions for IP multicasting," RFC 1112, SRI Network Information Center, August 1989.

[15] M. R. Macedonia and D. P. Brutzman, "Mbone provides audio and video across the internet," *Computer*, vol. 27, no. 4, pp. 30–36, April 1994.

[16] V. Kumar, *MBone: Interactive Multimedia on the Internet*. Indianapolis, IN, USA: New Riders Publishing, 1996.

[17] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '02. New York, NY, USA: ACM, 2002, pp. 205–217. [Online]. Available: http://doi.acm.org/10.1145/633025.633045

[18] K. To and J. Y. Lee, "Parallel overlays for high data-rate multicast data transfer," *Computer Networks*, vol. 51, no. 1, pp. 31 – 42, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128606000892

[19] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: A platform for high-performance internet applications," *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 3, pp. 2–19, Aug. 2010. [Online]. Available: http://doi.acm.org/10.1145/1842733.1842736

[20] R. Perlman, "Network layer protocols with Byzantine robustness," Ph.D. dissertation, Massachusetts Institute of Technology, 1989.

[21] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. Andersen, "SCION: Scalability, control, and isolation on next-generation networks," in *IEEE Symp. Security and Privacy (SP)*, May 2011, pp. 212–227.

[22] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to Byzantine failures," in *Proc. 1st ACM Workshop on Wireless Security*, 2002, pp. 21–30.

[23] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys Tutorials*, vol. 7, no. 2, pp. 72–93, Second Quarter 2005.

[24] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," RFC 3031, January 2001.

[25] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 87–98, Apr. 2014. [Online]. Available: http://doi.acm.org/10.1145/2602204.2602219

[26] C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, and R. Shakir, "Segment routing architecture," Internet Engineering Task Force Internet-Draft, February 2017. [Online]. Available: https://tools.ietf.org/html/draft-ietf-spring-segment-routing-11