A Survey on the Design, Applications, and Enhancements of Application-Layer Overlay Networks

JINU KURIAN and KAMIL SARAC

University of Texas at Dallas

This article presents a survey of recent advancements in application-layer overlay networks. Some of the most important applications that have been proposed for overlays include multicast, QoS support, denial-of-service (DoS) defense, and resilient routing. We look at some of the important approaches proposed for these applications and compare the advantages and disadvantages of these approaches. We also examine some of the enhancements that have been proposed in overlay topology design, enhanced routing performance, failure resistance, and the issues related to coexistence of overlay and native layers in the Internet. We conclude the article with a comment on the purist vs pluralist argument of overlay networks that has received much debate recently. Finally, we propose a new deployment model for service overlays that seeks to interpose between these two approaches.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.3 [**Computer-Communication Networks**]: Network Operations

General Terms: Design, Economics, Performance, Reliability, Security

Additional Key Words and Phrases: Overlay networks, service overlay networks, performance, enhancements, deployment model

ACM Reference Format:

Kurian, J. and Sarac, K. 2010. A survey on the design, applications, and enhancements of application-layer overlay networks. ACM Comput. Surv. 43, 1, Article 5 (November 2010), 34 pages.
DOI = 10.1145/1824795.1824800 http://doi.acm.org/10.1145/1824795.1824800

1. INTRODUCTION

Over the last few years, overlay networks have garnered much interest in the research and industrial community. This interest has been sparked primarily due to several distinct advantages offered by overlay networks for the testing and deployment of novel and possibly disruptive applications in the Internet.

Some of the proposed applications for overlay networks include multicast [Chu et al. 2000]; content delivery networks [Yu et al. 1999; Krishnamurthy et al. 2001; Su et al. 2006]; quality of service [Duan et al. 2003; Li and Mohapatra 2004b; Subramanian et al. 2004]; enhanced routing performance [Andersen et al. 2001; Anderson et al. 1999; Akamai a]; anonymity [Dingledine et al. 2004; Abe 1999;

Authors' address: J. Kurian (contact author; email: jxk032000@utdallas.edu), The University of Texas at Dallas, Department of Computer Science, 800 West Campbell Road, Richardson, TX 75080-1407.

DOI 10.1145/1824795.1824800 http://doi.acm.org/10.1145/1824795.1824800

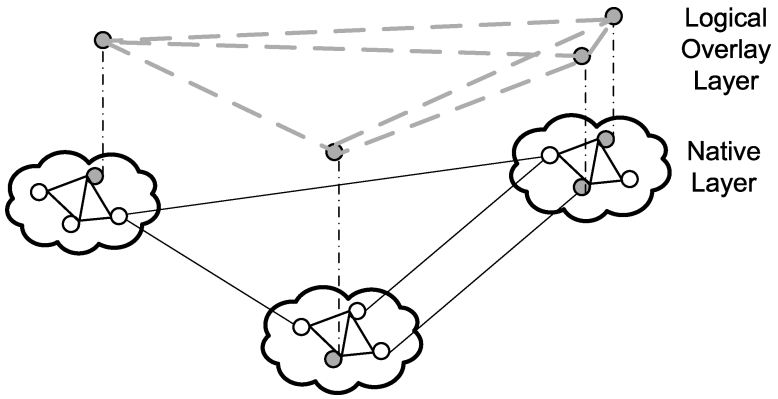

Fig. 1. Overlay model.

Anonymizer]; anycast [Freedman et al. 2006; Ballani and Francis 2005], IPv6 deployment [6bone]; testbeds [Chun et al. 2003; Touch et al. 2005]; denial of service (DoS) defense [Keromytis et al. 2002; Andersen 2003]; VoIP [Amir et al. 2005; Baset and Schulzrinne 2006]; reliable email [Agarwal et al. 2005]; distributed systems management [Liang et al. 2005]; and reliable name service lookup [Park et al. 2004]. Similarly, much work has gone into enhancing some of the important requirements associated with overlay networks like topology design, failure resistance, routing performance, Byzantine resilience, and native layer interaction.

Before continuing with our main discussion, we first answer the question: *What are overlay networks and why are they required?*

An overlay network is a virtual network that is built on top of another. It is usually built (directly by end users or a third-party overlay service provider (OSP)) to provide an application or service not easily provided by traditional methods to interested end users. In general, overlay architectures consist of two layers: (1) the overlay layer comprised of the overlay nodes and virtual links; and (2) the native layer over which the overlay network is built. In the Internet today, the native layer is the IP layer which provides a best-effort delivery service between remote systems. The overlay layer is comprised of a selection of the native layer's nodes, logically interconnected in any desired manner. Figure 1 shows an example of an overlay network. In this article, we concentrate on overlay networks built for the purpose of providing a specific application, as described above. For a detailed overview of peer-to-peer overlay networks, the interested reader is referred to Lua et al. [2004].

The amount of work that is related to overlay networks can be staggering to a novice reader. Yet, to date, there has been no concerted effort to survey the disparate applications and the enhancements of generic overlay architectures. In this article, we aim to provide the reader with a comprehensive overview of the more significant applications for overlay networks, the issues that arise with each of these applications and some existing solutions to these issues. We also present recently proposed two main views, called the purist [Ratnasamy et al. 2005] and pluralist argument [Peterson et al. 2004], on the long-term impact of overlay networks in the Internet. While the purist view sees overlay networks as testbeds for experimentation with novel network architectures, the pluralist view considers overlays as an integral part of the future Internet in providing value-added network services to end users.

The main motivation behind the deployment of overlay networks is to counter many limitations of the current Internet architecture that have become obvious [Ratnasamy

et al. 2005] in recent years. Some of the major concerns include the inherent lack of security [Keromytis et al. 2002; Andersen 2003; Anderson et al. 2003; Mirkovic et al. 2002; Moore et al. 2001]; QoS guarantees [Subramanian et al. 2004; Duan et al. 2003]; mobility support [Snoeren et al. 2001]; multicast support [Chu et al. 2000; Almeroth 2000]; end-to-end service guarantees [Duan et al. 2003; Blumenthal and Clark 2001]; and the presence of unwanted and spurious traffic [Shin et al. 2006; Xu and Zhang 2005]. While the calls for change and the solutions proposed have been numerous, the acceptance and deployment of these solutions have not kept pace. Many researchers have voiced their concerns about this perceived "ossification" of the Internet [Turner and Taylor 2005; Peterson et al. 2004] which prevents even necessary changes in the infrastructure from taking place. In this context, overlay networks have emerged as a viable solution to such ossification by providing third-party service providers and users with a means to address some of the aforementioned issues at a smaller scale and without requiring universal change or coordination for the deployment of new services.

The rest of this article is organized as follows. Section 2 presents some of the more important overlay applications that have been proposed in the literature. Section 3 describes generalized overlay models for these applications and the enhancements that have been suggested based on these generalized models. Section 4 discusses the impact of the presence of overlay networks in the Internet. Section 5 discusses the purist vs pluralist view of overlays and proposes a new model of overlay deployment. Finally, Section 6 concludes the article.

2. APPLICATIONS OF OVERLAY NETWORKS

The proposed applications for overlay networks have been numerous and largely disparate. We consider some of the more important applications here.

2.1. Content Delivery Networks

Content Delivery Networks (CDNs) were initially deployed in the 1990s to overcome the rampant congestion that was plaguing the Internet at that time. CDNs are overlay nodes deployed across the Internet that dynamically cache content and services to deliver them to end users. Some of the most popular CDNs include Akamai [Akamai b]; Netli [Netli]; Accelia [Accelia]; EdgeStream [Edgestream]; Globule [Globule] and CoDeeN [Codeen].

The basic service offered by CDNs is data replication. Replication involves creating copies of a site's content (replicas) and placing it in carefully chosen locations across the Internet. To provide the best user experience, the user request is optimally redirected to the location that can best serve the user request. When combined with redirection, replication allows for reduced network latency and overhead and improves availability of the replicated site through the inherent redundancy in the system.

There are several important concerns to be addressed in the operation of CDNs, including the metrics to be optimized, when and where to replicate content, what content to replicate, ensuring consistency among replicated sites, and redirecting the user request to the appropriate location [Yu et al. 1999]. We briefly discuss replica consistency and user redirection below. We refer the reader to Yu et al. [1999] and Krishnamurthy et al. [2001] for a thorough discussion of other relevant issues.

Figure 2 shows a typical CDN framework. Origin servers are the servers that the CDN serves. Replica servers or surrogate servers replicate the content of the origin servers across them. At a lower level are intermediate proxy servers or redirectors through which content is served from the replica server to the clients. Typically, a

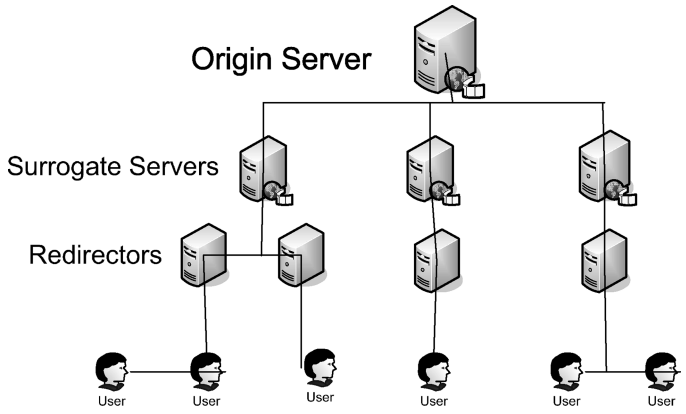

Fig. 2. CDN overview.

client's request to an origin server is redirected to a replica server which transfers the requested data through the proxy server to the end user.

To maintain consistency or freshness of replicated content, several techniques are commonly used, including:

— *Pull based*: Pull-based schemes place the task of ensuring consisteny on the replicas. The replicas periodically refresh their content from the origin server and update any changes made. There are several techniques employed by pull-based schemes to decide when the content needs to be refreshed. The versioning scheme used by Akamai [Akamai b] is one technique. The version of the replicated document is encoded in the document name. When a new version is requested, it creates a "cache miss" in the replica. This triggers the request for a newer version from the origin server. Other pull-based techniques include periodic polling [Gwertzman and Seltzer 1996]; TTL-based approaches [Moghul 2000; Breslau et al. 1999]; and TTR-based approaches [Urgaonkar et al. 2001].

— *Push-based*: Push-based schemes place the task of ensuring consistency on the server. The server is required to maintain state about its proxies, and pushes updated content to these proxies as required [Yu et al. 1999].

— *Hybrid schemes*: These schemes aim to take advantage of both pull-based and push-based schemes. Lease-based schemes [Duvvuri et al. 2003; Ninan et al. 2001] are examples of hybrid updating schemes.

Redirection techniques aim to direct the client's request to the replica server best suited to serve the request. The actual selection strategy used can be quite complex and usually tries to optimize various parameters, including latency, cost for transfer, load on the replica servers, freshness of content, and so on. Two of the most common techniques for redirection include DNS redirection and URL rewriting.

DNS redirection involves the authoritative DNS server of the client resolving the domain name of the origin server to one of the replica servers of the CDN. To achieve load-balancing, the replies are usually of very low TTL (having a low TTL allows the client to make a new DNS request, which may be directed to an alternate replica server). DNS redirecting can be full or partial. In a full redirection, the client is always redirected to the replica server that serves the entire website content from its cache or forwards the request to the webserver. In selective or partial redirection, the URL of the selected content is rewritten so that it is redirected to the CDN server while other

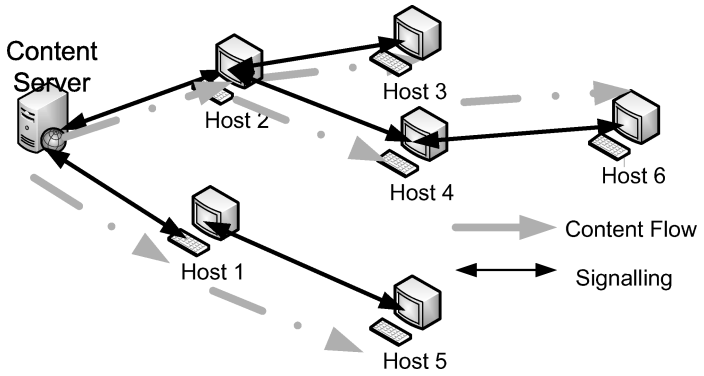

Fig. 3. Multicast overlays (adapted from ESM).

content is directly served by the origin server. (For, e.g., Akamai "akamaizes" URLs into "ARLs" via a series of DNS translations [Su et al. 2006]).

URL rewriting [Krishnamurthy et al. 2001] techniques avoid the need for a DNS lookup entirely by rewriting the URLS of replicated content with the IP address of one of the replicated servers. Content with high bandwidth requirements is usually selected for rewriting and pushed to replicated servers by the origin server. Other techniques also exist for user redirection; hybrid schemes that combine DNS redirection and URL rewriting, HTTP-level redirection and Layer 4-7 switching [Venu 2002] to name a few.

CDNs account for the bulk of overlay traffic in the Internet today, and have generated the most industrial and economic interest in overlay networks. The Akamai CDN, for example, spans 69 countries with over 15,000 servers and serves many high profile sites like yahoo.com and cnn.com as their clients.

2.2. Overlay Multicast

IP multicast was one of the first value-added services proposed in the Internet. Highly efficient in multireceiver applications, it received a lot of research interest for its development. IP multicast was, however, plagued with many problems which have prevented its widespread deployment. Today, IP multicast has been deployed primarily in the intradomain scale without much interdomain connectivity. The MBone overlay [Macedonia and Brutzman 1994; Almeroth 2000] was one of the first attempts to interconnect disjoint multicast zones to each other. IP tunneling was used for transporting data over multicast-unaware routers. MBone provided the required connectivity but was plagued with problems of unreliability, heavy loss, low throughput, and difficult management. Application-layer multicast was proposed as a viable alternative to IP multicast. In application-layer multicast, a multicast tree is built at the application layer between participating group members. Data delivery is provided through unicast tunneling mechanisms on a hop-by-hop basis between the group participants. Unlike IP multicast where data is replicated by the routers, in overlay multicast, data is replicated by overlay nodes. The operation of multicast overlays is shown in Figure 3.

Narada: Narada [Chu et al. 2000] or End System multicast (ESM) was one of the first multicast overlays proposed. In ESM, overlay nodes initially create an overlay mesh between all group members. Once the mesh is created, a shortest-path tree to the source is built on top of the mesh. This tree-building approach used in ESM is

called the **mesh-first** approach.[1] Although not as efficient as IP multicast, ESM offers some distinct deployment advantages over it. ESM allows the deployment of multicast services without manifestable changes in the network and maintains the networks' stateless nature. Since unicast tunneling is used between overlay nodes, interdomain connectivity is also not an issue. Although unicast is the primary mode of transport, ESM is more efficient than native unicast for multi-receiver applications. In Narada, the source has to transmit only a single packet to each child overlay node in the multicast tree, which was then replicated by the overlay nodes for data delivery. This reduces the overhead on the sender and the network and makes for more efficient data delivery.

ESM, while seminal in multicast overlays, has several issues that restrict its effectiveness. One of the more significant issues with ESM is that its scalability is detrimentally affected by the group management protocol used. The model proposed by ESM has been enhanced over the years to allow for much larger group sizes. We next consider one approach that allows for much better scalability. There have been other significant improvements also allowing for more efficient tree construction, higher data rates, better failure resistance, and for optimizing various parameters like delay, data loss, overlay deployment costs, link stress, and load balancing. Some of the significant improvements include ALMI [Pendarakis et al. 2001]; Scattercast [Chawathe 2003], Yoid [Francis]; NICE [Banerjee et al. 2002]; Overcast [Jannotti et al. 2000]); Bullet [Kostic et al. 2003]); and TAG [Kwon and Fahmy 2002].

NICE: NICE [Banerjee et al. 2002] attempts to build a very low overhead overlay network which can scale to a very large number of nodes. Significant in the NICE architecture is the use of a layered, hierarchical approach. By distributed management of the overlay and providing addressing in a hierarchical manner, the amount of information that is required to be maintained at each node is significantly reduced.

The hierarchial addressing used in NICE is as follows. Nodes are organized into layers, with nodes in each layer further organized into clusters. Each cluster has a cluster leader, which is the node with minimum distance to all other nodes in the cluster. The cluster leaders of each layer are the only nodes joined to the next higher layer. In this arrangement, Layer 0 is the lowest level of hierarchy and contains all the nodes. This means that the worst-case information to be maintained by a node is about $O(\log N)$ other nodes (as opposed to $O(N)$ for ESM). Data delivery also follows the hierarchical delivery path. A source-specific tree is built from the source of the message for this purpose.

TAG: The importance of topology-awareness in overlay construction is well studied. Topology-aware grouping [Kwon and Fahmy 2002] (TAG) was proposed to exploit underlying network topology information in order to build a more efficient tree. The tree construction in TAG uses network measurements to optimally place new nodes in the network. When a new member desires to join a multicast session, the source S calculates the shortest path between itself and the new node (using tools like traceroute and pathchar or from OSPF topology servers). It then uses a path-matching algorithm to find the overlap between its currently used paths and the shortest path to the new node. The purpose of the path-matching algorithm is to ensure that a new node can join the multicast tree using an existing path in the tree at the point of overlap closest to it.

Another advantage of TAG is that unlike most multicast overlays, which aim to optimize a single metric like delay, bandwidth, or loss rate, TAG can use both delay

[1]The mesh first approach is in contrast with the **tree-first** approach where the tree is constructed directly between the group participants [Abad et al. 2004].

	ALMI	Narada	TAG	NICE	Yoid	Overcast
Tree construction Strategy	Mesh first, centralized to build a MST based on inter-node probes	Mesh first source-specif trees based on RPF	Tree first, centralized to place nodes in a tree	Hierarchial trees based on splitting nodes into clusters	Tree first new nodes become child of an existing parent	Tree first, centralized, nodes decide position wrt a root node
Topology Aware	No	No	Pathchar and OSPF info used	No	No	No
Reliability	Yes	No	No	No	Yes	Yes
Optimizing parameter	Delay	Delay	Delay and bandwidth	Delay	Latency	Latency
Group Size	Large	Small	Large	Very Large	Large	Large

Fig. 4. Comparison of various multicast overlays; ALMI [Pendarakis et al. 2001]; Narada [Chu et al. 2000]; TAG [Kwon and Fahmy 2002]; NICE [Banerjee et al. 2002]; Yoid [Francis]; and Overcast [Jannotti et al. 2000].

and bandwidth (because of the measurements made prior to overlay construction) as joint considerations during overlay construction. To achieve efficiency and fault tolerance, the intermediate nodes and the root nodes participate in periodic probing. As with most probing-based overlays, this probing can be expensive if the overlay is very large.

The field of application-level multicast has received much attention over the years. Figure 4 provides a tabular comparison among various approaches. For a more thorough overview of multicast overlays, the interested reader is referred to Abad et al. [2004].

2.3. QoS Guarantees

Over the years, much effort has gone into providing end-to-end QoS at the network layer in the Internet. IntServ [Braden et al. 1994] and DiffServ [Carpenter and Nichols 2002; Blake et al. 1998] were proposed and established as QoS standards for the Internet. However, similar to IP multicast, network-layer QoS faced several deployment problems in a large scale. One of the first solutions to the lack of deployment of QoS was an overlay-based testbed, Qbone [Qbone]. Since then, researchers have proposed the construction of overlay networks which can provide QoS guarantees to applications using the overlay network.

SON: The service overlay network (SON) [Duan et al. 2003] approach aims to provide QoS guarantees in the interdomain scale through an overlay network. QoS is provided for the overlay network by purchasing bandwidth from ISPs via bilateral SLAs with certain QoS guarantees. The overlay nodes are logically interconnected through these bandwidth-guaranteed connections to provide end-to-end QoS guarantees. The use of SON allows the deployment of QoS-sensitive applications in the network without the overhead required for network-layer QoS. SON also simplifies QoS provisioning for ISPs by allowing QoS provisioning for a larger granularity of individual SONs, as opposed to individual flows.

	SON	OverQoS	QRON
Strategy	Inter-connected QoS provisioned links	Forward Error Corrction and ARQ to provide delivery guarentees	Bandwith provisioned links with QoS aware routing
Routing	NA	Depends on the overlay over which it is deployed	MSDP and PBSP
Topology	Depends on AS-level topology	Depends on overlay over which it is deployed	Hierarchial with nodes deployed globally
Underlay provisioning required ?	Yes	No	Yes

Fig. 5. Comparison of various QoS overlays, SON [Duan et al. 2003], OverQoS [Subramanian et al. 2004], QRON [Li and Mohapatra 2004b].

The significant challenge in the SON architecture is for the OSP to purchase bandwidth in an efficient manner from ISPs. Since this purchase of bandwidth is a capital-intensive affair, the bandwidth requirements need to be carefully calculated to minimize capital expense. On the other hand, the SON must be provided adequate bandwidth to support the QoS requirements of the services that it aims to support. Provisioning also needs to be made for possibly fluctuating requirements without excessive penalty. Duan et al. [2003] model the bandwidth provisioning requirements as optimization problems for static and dynamic bandwidth requirements and suggest approximate solutions to these problems. Other issues which have been studied in the context of SON include the design of an efficient SON topology [Vieira and Liebeherr 2004a] and reconfiguring the service overlay to optimize the cost of using the overlay [Fan and Ammar 2006].

One of the unique attributes of the SON approach is its dependence on ISPs to deploy the overlay. This is contrary to the general wisdom of complete independence from ISPs in overlay operation and deployment. We call this overlay deployment model the *SON model* of deployment. The SON model is contrary to the deployment model, wherein the overlay is completely independent of the ISPs (we call this model the *P2P model*) in its operation and deployment. As we discuss later, the involvement of ISPs can provide significant advantages to overlay-based applications. The SON deployment model has been adapted by various authors in creating service architectures for various applications. The Service Oriented Internet (SOI) architecture [Chandrashekar et al. 2003] is one instantiation of a SON overlay as a working infrastructure to provide VoIP. Another example, which we discuss next, is QRON which can be considered as an implementation based on the SON model.

QRON: In QRON, Li and Mohapatra [2004b] propose the creation of a bandwidth provisioned overlay network (OSN) similar to SON. The overlay network consists of overlay brokers (nodes) with bandwidth-guaranteed connections between them. As

with SON, QRON can provide a QoS-guaranteed overlay path to the application. QRON additionally tackles some of the architectural and functional aspects of a practical SON overlay.

One relevant issue with a practical SON overlay is its scalability to a large number of nodes. In QRON, the solution is to organize the overlay using a hierarchical clustering and naming scheme. Overlay nodes are clustered into several levels, with level-1 being the highest-level cluster consisting of level-2 clusters, and so on. The clustering is done such that nodes within the same AS and those that are physically close to each other are clustered together. Additionally, if the overlay nodes/clusters have multiple overlay links between each other, they are clustered together. This clustering approach ensures that nodes are clustered to have low latency between each other and a high degree of connectivity. The naming scheme used follows directly from the clustering scheme. A i-level cluster has an i-tuple for its name, for example, a 3-level cluster has a x.y.z tuple as its name. The clustering scheme reduces the overhead of propagating reachability and addressing information. Local information is broadcast within the cluster only. Reachability between clusters is then provided by organizing the gateway nodes of each cluster into an overlay mesh.

Routing on top of the overlay is another important consideration. The authors propose two routing schemes: modified shortest distance path (MSDP) and proportional bandwidth shortest path (PBSP), which aim to select paths with the best available bandwidth in addition to the traditional shortest paths. Finally, the authors posit the presence of an overlay service layer (OSL) above the transport layer at all overlay nodes, which provides common functionalities like overlay routing, topology discovery, overlay link performance estimation, and resource allocation to the application layer.

Generally, QoS is a difficult application to provide without explicit ISP support [Crowcroft et al. 2003]. OverQoS [Subramanian et al. 2004] however provides a different perspective without such explicit support. In OverQoS, overlay nodes implement forward error correction (FEC) and automatic repeat request (ARQ) schemes to provide an upper bound on the loss experienced by traffic to provide a measure of QoS guarantees on top of overlay architectures. Figure 5 shows a tabular comparison between the different QoS approaches we have discussed in this section.

2.4. Improving the End-to-End Performance and Resiliency of Internet Routing

To improve scalability and to hide local policies, ISPs traditionally heavily filter and aggregate their BGP route announcements to peers. As a direct result of this, when faults occur, BGP fault recovery and routing convergence can be exceedingly slow [Balakrishnan et al. 1997; Labovitz et al. 2000; Paxson 1997]. Additionally, ISPs typically choose their interdomain paths to suit local policy requirements, rather than choosing the path with best performance [Savage et al. 1999]. This results in Internet paths that suffer from low resiliency and availability. Hence users are often provided with suboptimal performance in end-to-end latency, packet loss rate, and TCP throughput [Anderson et al. 1999].

The Detour [Anderson et al. 1999] project identified some of the significant problems mentioned above in the Internet. Their studies were conducted via a framework of geographically distributed overlay nodes. By routing through these overlay nodes, the authors identified the presence of superior alternate paths not generally used by unicast routing in the Internet. In fact, in 30 to 80% of the cases measured, the alternate path was significantly better than the default unicast route. To exploit this inherent path redundancy in the Internet, routing overlay networks have been proposed. A routing overlay can avoid slow BGP convergence by routing around the failure through its

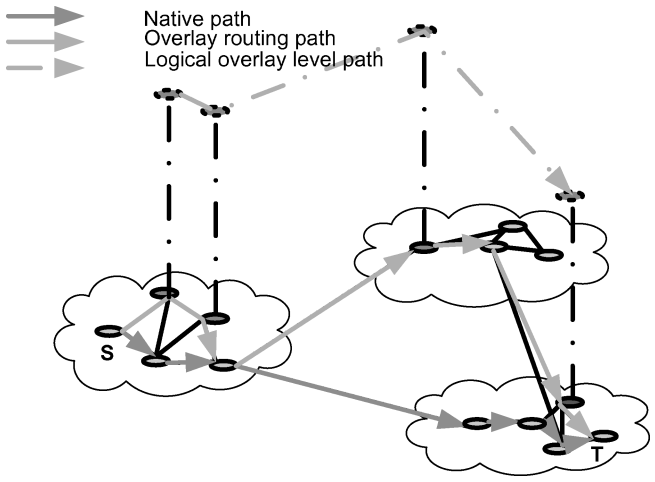

Fig. 6. Operation of routing overlays.

intermediate nodes. Through active probing between overlay nodes, overlay networks can also quickly detect and recover from failures in the network. Finally, by routing through paths that provide the best end-to-end performance irrespective of ISP policies, overlay routing can provide better performance than network-layer routing. This operation of routing overlays is shown in Figure 6. As can be seen, overlay nodes across multiple domains maintain overlay-level paths between them. A failure in the physical path comprising the overlay-level path will cause the overlay to switch to another overlay-level path, hence avoiding the physical-level failure.

RON RON [Andersen et al. 2001] was the first large-scale routing overlay implementation and testbed dedicated to providing improved resiliency and performance over default network-level paths. The RON overlay consists of nodes deployed at various locations across the Internet, logically interconnected to form a full mesh. The RON overlay nodes are deployed by end users without any support from the ISPs for their operation (the P2P model).

The main objectives of the RON overlay are to find the best possible path to a destination and to provide resiliency and quick recovery in case the chosen path fails. Thus RON nodes constantly probe each other to obtain topology and performance information. The collected information is then disseminated throughout the overlay network using a link-state protocol. To provide the best routes, the source RON node creates a forwarding table based on the collected information for one of three metrics: (i) latency; (ii) packet loss; and (iii) throughput. Failure resistance also depends on the probes. If a probe is lost, the low frequency probes are replaced by a succession of probes separated by a short interval. If the probes are not responded to after a certain threshold, the path is considered broken.

Based on the active probes and intelligent path selection, RON nodes can generally detect failures earlier, route around failures, and improve the end-to-end performance of applications over traditional unicast. The RON architecture, however, has several shortcomings, including its lack of scalability due to the full-mesh topology, high-frequency probing, uninformed selection of nodes, and lack of topology awareness. Subsequent work on routing overlays have improved upon many aspects of measurement-based overlay networks like RON. We look at these in more detail in Section 3.

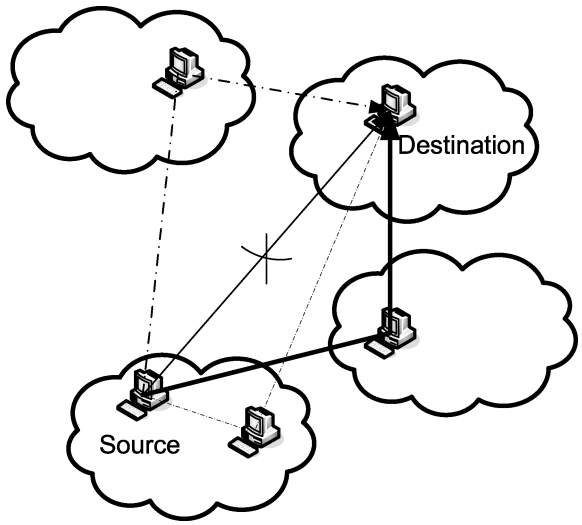

Fig. 7. A one-hop indirection scheme with k = 3 (adapted from SOSR).

Routing overlays like RON generally impose a second overlay routing scheme on top of the native layer's routing. The scalability and overhead of these overlays is inherently limited due to the complexity of these schemes and the routing overhead involved in their operation. However, the authors of RON observed that most of the benefits that could be achieved by overlay routing were effectively captured by routing through a single intermediate node instead of routing through multiple nodes. Gummadi et al. [2004], further develop this idea into scalable one-hop source routing (SOSR). The routing scheme is quite simple: when the source wants to send data, it picks k intermediate overlay nodes at random and attempts to send packets through them. It then chooses the intermediary with the first response to route traffic through. During transmission, if the in-use intermediary fails, the source continues to retry the same set of intermediaries. If the next try also fails, the source selects a new set of k random intermediaries. For example, in Figure 7, the source picks k = 3 intermediate nodes on failure of its main path and tries them all. The path with the best response time is chosen. Experimental observations show that setting k = 4, gives close to the maximum benefits available. Han et al. [2005] make similar observations based on their measurements of single vs multiple hop overlay routing.

The impact of single-hop overlay routing is significant. It shows that the benefits of overlay routing can be achieved without background monitoring and the involvement of complex routing schemes. Additionally, the added latency incurred by the processing required at multiple overlay nodes can be avoided by using a single overlay node.

Akamai Sureroute: Akarouting or Sureroute [Akamai a] is an overlay-based routing service provided by Akamai. The Sureroute overlay is based on observations similar to the results obtained by Detour [Anderson et al. 1999]. An average gain of 15 to 30% was observed when routing through an alternate path rather than the default path. Similar to RON, the Sureroute overlay consists of an overlay network that uses ping data to collect topology and performance information (called the Map Maker component) of its overlay nodes. The placement of overlay nodes is topologically distributed,

which enables the overlay to have a large number of alternate paths. The best paths are chosen (by The Guide) using the concept of races between available paths. Periodically, simultaneous downloads (races) are employed through multiple paths, and the winner is recorded as the path for the near future. Finally, to provide the best last-hop performance, Sureroute is used in conjunction with Akamai EdgeSuite [Akamai c]. Sureroute is a redirection technique that directs the user to the most optimal edge overlay node.

Much of the Akamai technology is proprietary, so information about Akamai Sureroute is limited. While RON and Sureroute aim to utilize the implicit redundancy in the Internet through the multiple paths provided by overlay networks, there have also been proposals that seek to extend this redundancy explicitly. Next, we consider a couple of approaches that provide such explicit redundancy.

MONET: MONET [Andersen et al. 2005] proposes an overlay network similar to RON. However, the authors seek to extend the inherent redundancy in the Internet with explicit redundancy in the form of multihoming. Multihoming is provided through multiple edge ISPs, contacting multiple server replicas, obtaining multiple paths in the overlay network and multiple DNS requests to mask DNS failures. Additionally, to reduce the overhead imposed by path probing, MONET uses a selective path-probing scheme. A path is probed only if it is likely that previous attempts have failed. The delay between probes is chosen based on the variance observed in RTT values. If a path shows a stable RTT over time, the path is less likely to be probed in the future. This ensures that the overhead produced in MONET by path probing is minimal, but it can serve path requests without too much delay. A similar approach is extended by Zhu et al. [2007]. They motivate the design of their architecture from the perspective of an OSP that seeks to provide better service to its customers while turning in a profit for itself. To this end, they provide heuristics for the optimal placement of overlay nodes in the ISP networks and the selection of ISPs for multihoming.

We continue our discussion of routing overlays with two alternative approaches to traditional measurement-based routing overlays. The design of the first draws inspiration from the closely related area of structured peer-to-peer networks. The second approach shows that the benefits of measurement-based overlays can be obtained with much lesser overhead and complexity.

Structured P2P-based: Structured P2P overlays inherently possess failure resistance due to large amounts of path redundancy in their architecture. For example, in Tapestry [Zhao et al. 2004], overlay nodes are assigned node-ids, and destinations are assigned keys with each key being mapped to one of the node-ids. Overlay routing tables consist of a list of overlay node-ids and their keys. Thus each overlay node can maintain a number of different routes to each destination in its routing table. If a failure is detected, one of the backup routes can be chosen to route around the failure.

This routing scheme is called feedback-based proactive routing [Li and Mohapatra 2004a] and is implemented by Zhao et al. [2003] using the Tapestry structured overlay. The advantage offered by this scheme is that the overlay can immediately route around failures by picking one of the backup paths without waiting for the overlay or native routing protocol to detect and recover failed paths. While this approach can provide failure resistance, it may not provide the best possible paths, and hence best end-to-end performance. In fact, the use of a structured overlay in the SOS overlay [Keromytis et al. 2002] suggests an increase in end-to-end latency by a factor of more than 8.

	Detour	RON	MONET	Sureroute	P2P based
Service provided	Improved e2e performance	Resilient routing, improved e2e performance	Resiliency, improved availability and e2e performance	Resilient routing, improved e2e performance	Resilient routing
Basic Startegy	Find best route to destination via measurements between nodes	Link state based reactive routing Probes detect failures & gather path information	RON type overlay combined with multihoming	Races between paths and ping data to choose best paths	Use inherent redundancy in P2P overlays to find alternate paths
Deployment	P2P model, end users provide nodes	P2P model, end users provide nodes	Not specified	Third party service provider	Not specified
Failure resistance	Not specified	Probes detect failed paths, routing finds new path	Similar to RON, multihoming adds resitance	Probes to detect failed paths, route around failures	Feedback based proactive routing uses backup paths when primary fails.

Fig. 8. Comparison of various routing overlays, Detour [Anderson et al. 1999]; RON [Andersen et al. 2001]; MONET [Andersen et al. 2005]; Sureroute [Akamai a]; P2P-based [Zhao et al. 2003].

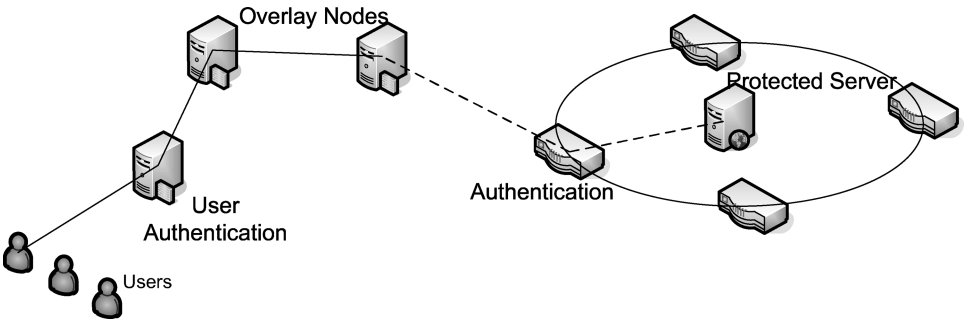

Fig. 9. Security overlay model (adapted from Mayday).

Other enhancements to routing overlay networks in topology design and operation have received a lot of attention in the research community in the recent past. We look at some of these later in Section 3. Figure 8 provides a tabular comparison between some of the different routing overlays described in this section.

2.5. Security and DoS Defense

Denial of service (DoS) attacks are a significant threat to the health and utility of the Internet. Several authors have explored overlay-based schemes to provide DoS resistance to DoS-vulnerable servers. The main objective of these overlay-based approaches is to provide a receiver-controlled communication service. This implies that the receiver (the protected target) can choose who it wants to receive traffic from and prevent unwanted traffic coming from others. Figure 9 shows an example of a typical overlay used for DoS defense.

Secure Overlay Services: SOS [Keromytis et al. 2002] was the first to propose the use of overlay networks to provide DoS resistance in the Internet. In SOS, traffic is routed through a series of overlay nodes in a manner similar to onion routing [Goldschlag

et al. 1999] via a Chord [Stoica et al. 2003] overlay to a special node (secret servlet) in the overlay network. The target to be protected from DoS attacks is protected via a filtering ring formed by the routers around it. Only traffic from the secret servlet is allowed through the filtering ring. The security of SOS depends on two factors: (i) the location of the secret servlet is hidden because of the anonymous nature of the overlay routing and (ii) the secret servlet will receive traffic only through the overlay and only after it has been authenticated at the overlay edges. Hence, to access a SOS protected server, a user needs to be preapproved by the server (receiver control) and then authenticated at the edges of the overlay network.

The SOS scheme while effective, incurs overhead in the size of the overlay network required and the circuitous routing that is required to reach the secret servlet. There have been several enhancements to the SOS architecture, including WebSOS [Stavrou et al. 2005] which allows SOS to authenticate previously unapproved users with human recognition tests. Mayday [Andersen 2003] generalizes the SOS architecture to provide more routing and filtering choices. MOVE [Stavrou et al. 2005] suggests a protection service that does not require infrastructure support. Stateless multipath overlays in which the user spreads her traffic across multiple overlays [Stavrou and Keromytis 2005] in a psuedo-random manner to avoid directed attacks on active overlay nodes and FONet [Kurian and Sarac 2007], which we will discuss shortly.

i3-based solution: The Internet Indirection Infrastructure (i3) [Stoica et al. 2002] is an indirection layer that allows hosts to control what packets they receive. I3 uses the concept of a rendezvous point between the sender and the receiver. Instead of sending a packet directly to the receiver, the packet is associated with an identifier (trigger) that maps it to a rendezvous point. To receive data, the receiver asks (inserts a trigger) to receive packets with a specific id. The network searches for packets with the specified id and forwards them to the receiver.

By the inherent nature of its operation, i3 allows hosts to explicitly control what packets it chooses to receive. Through the i3 overlay, end hosts can hide their IP addresses and choose to avoid receiving packets at arbitrary ports. By providing only trusted senders with a private trigger and rate-limiting traffic received from public triggers, a server can protect itself from malicious attacks [Adkins et al. 2003]. If a trigger is being misused for DoS attacks, the server can remove the trigger to stop receiving traffic from the trigger completely, at the expense of possibly dropping legitimate traffic.

DefCOM: Traditional DoS defense mechanisms operate at one or at most two "locations" in the three locations of interest in DoS defense, the source end, the victim end, and the network core. In DefCOM [Oikonomou et al. 2006], the authors argue that for effective DoS defense, all three of these locations have to be successfully monitored. In particular, the authors argue that attack detection is best done at the victim end, rate limiting is more effective at the network core, and source end traffic monitoring mechanisms can be used to detect and separate attack traffic from legitimate traffic.

In addition to the three-pronged defense deployment, messaging for coordination between all three ends is established through an overlay network. Nodes at different ends collaborate with each other by exchanging messages, packet marking, and marking-based processing. The overlay network is dynamically constructed in a manner similar to multicast tree construction. A DefCOM node that desires to join the network sends a join message towards the destination. If a tree is present, the node joins the overlay at the point where its message is intercepted by a node in the tree.

	SOS/Mayday	WebSOS	FONet	i3-based
Service provided	DoS protected communication for known users	DoS protected communication for previously unknown but human users	Different services based on the requirement of the customer	Receiver controlled communication
Size of ovelay	Large number of nodes Small coverage	Large number of nodes Small coverage	Large number of nodes Large coverage	Large number of nodes Large coverage
Compromised nodes proection	Not explicitly, but implicit protection due to large number of nodes	Similar to SOS	Not explicitly, but nodes are protected similar to network infrastructure by ISPS	Not explicitly
Routing	Chord ring for SOS Mayday suggests several alternatives with different properties	Chord /CAN overlay routing	Path vector protocol	Trigger based forwarding
Sweeping attacks	Possible	Possible	Not possible , overlay nodes cannot be directly attacked	Possible
Network support required ?	Yes, for filtering	Yes, for filtering	Yes, for filtering and tunneling	Yes, for flow maintanence filtering and throttling

Fig. 10. Comparison of various security overlays, SOS [Keromytis et al. 2002]; Mayday [Andersen 2003]; WebSOS [Stavrou et al. 2005]; FONet [Kurian and Sarac 2007]; i3-based [Adkins et al. 2003].

The operation of DefCOM is based on alerts generated at the victim end in the presence of an attack. These alerts are flooded in a controlled manner. Nodes that forward traffic to the victim then form a traffic tree and each node monitors its peers to identify highly aggressive flows. Nodes with higher aggressiveness than a specific threshold are considered malicious, as they have failed to rate-limit their outgoing traffic and their traffic is marked as unstamped. Unstamped traffic may be dropped or heavily rate-limited in the rest of the network.

FONet: In FONet [Kurian and Sarac 2007], we have proposed a large-scale federated overlay network for DoS defense in the Internet. FONet nodes are deployed across the Internet and are interconnected via DoS-resistant tunnels. Unlike SOS, in FONet the overlay nodes are explicitly protected against DoS attacks by restricting access to them to authorized traffic from within their domains only, or neighboring FONet nodes. This simplifies routing (anonymous routing to protect the nodes is not required) and makes the architecture as a whole more secure. FONet nodes provide authentication based on user credentials and securely route user traffic through the overlay network to the destination. The FONet architecture also generalizes the services offered by SOS and seeks to provide a hierarchy of protection services with tradeoffs between openness and security.

Security overlays are an intriguing area of overlay application. By allowing security to be provided at a higher layer, security overlays can potentially counter many of the problems associated with the open nature of the unicast Internet. Figure 10 provides a tabular comparison between the different security overlay approaches described in this

section. Besides DoS defense, overlays may also find applications in authentication, traceback [Stone 2000], and anonymity.

Tor: Anonymous networks are built with the purpose of providing users with protection against traffic analysis and user profiling widely prevalent in the Internet. A significant amount of work (overlay based and otherwise) has been devoted to this purpose [Dingledine et al. 2004; Abe 1999; Anonymizer ; Freenet Project ; JAP]. In this section, we briefly look at Tor [Dingledine et al. 2004] a second-generation onion router [Goldschlag et al. 1999]. In onion routing, messages travel from source to destination via a sequence of proxies ("onion routers"), which reroute messages in an unpredictable path to provide anonymity to the sender.

Tor addresses some of the limitations of onion routing by adding perfect forwarding secrecy, congestion control, integrity checking, and a low latency communication service. Tor provides an overlay network of onion routers that provide two services: (1) anonymous outgoing connections for clients and (2) anonymous hidden services for servers.

Providing anonymous outgoing connections is the primary objective of Tor. Tor users install a Tor proxy in their machines, which connects to one of the Tor router nodes. A virtual path to the destination is established through the overlay nodes via onion routing. The data is forwarded through the Tor network until it reaches an exit point from where the packet is directly forwarded to the destination. The original Onion router design could not provide perfect forwarding secrecy, as a single hostile node could record traffic in the network. In Tor, ephemeral keys are used at each hop, so once the keys are destroyed, old traffic cannot be decrypted.

Tor uses a general-purpose (SOCKS proxy) interface which allows it to work with virtually any application that supports TCP (thus Tor provides anonymity at the TCP stream level). This is a unique feature to Tor, whereas in onion routing separate interfaces were required on a per-application basis. Tor provides congestion control via end-to-end ack messages and strict congestion checking at the edges of the network. For use with web applications, in practice Tor is combined with an application layer proxy, Privoxy [Privoxy] which adds further anonymity at the application layer.

Tor can also provide anonymous hidden services to servers. Such a Tor-hidden server can only be accessed through clients that also speak Tor. A Tor-specific .onion-level domain name is used for this purpose. The Tor network can resolve this .onion domain name and handle user traffic to these domains to reach the destination server anonymously.

In this section we have described some of the more important overlay applications. As we mentioned before, overlay networks have been proposed for a much larger set of applications in several disparate and interesting areas. Due to lack of space and to concentrate on the most important applications, we stop our discussion of overlay applications here and continue this article with a discussion of the enhancements that have been made to some of the architectures described above. To avoid a specific implementation-based discussion, in the next section we start discussion with generalized overlay models for routing, QoS and security overlays, and consider the enhancements from the perspective of these generalized models.

3. OVERLAY ENHANCEMENTS

In the first part of this section we describe generalized overlay models for routing, security, and service overlays. We concentrate on these applications due to the protracted research interest in these three areas. We use these models throughout our discussions

in the rest of the article. These generalized overlay models have been greatly enhanced by researchers over the years. In the second part of this section we examine some of the most relevant work in overlay enhancements.

3.1. Generalized Overlay Models

3.1.1. Routing Overlay Model. The generalized routing overlay model is mainly due to RON [Andersen et al. 2001]. The purpose of a routing overlay is to provide an improved routing performance over that provided in the Internet. It does so by finding the best paths to destinations and quickly detecting failures to route around them. The overlay nodes perform probes to its neighboring nodes in the overlay. These probes help detect whether an overlay link has failed and help the overlay node to measure performance statistics to each overlay node. To perform routing, the overlay network executes a routing protocol (usually link state) between the nodes in the overlay network. This allows the nodes to select best paths (based on some metrics, including latency, throughput, loss rate, and bandwidth) and route around failures in the network as they occur. The routing overlay generally performs better than the native layer routing because it is able to choose paths that are not available at the native layer (due to various reasons including policy restrictions and physical restrictions) and detect failures earlier than the native layer.

3.1.2. Service Overlay Model. The generalized service overlay model is mainly due to SON [Duan et al. 2003]. A service overlay network is deployed by a third-party service provider to provide an added service to its customers. The service provided is generally QoS, but may also be resilient routing, content delivery, and security. The service overlay is usually provided some support (for a price) for its operation from the ISPs. For example, a QoS-enhancing service overlay network may purchase bandwidth and delivery guarantees for its traffic from ISPs [Duan et al. 2003].

3.1.3. Security Overlay Model. The generalized security overlay model (Figure 9) is due to SOS [Keromytis et al. 2002] and Mayday [Andersen 2003]. The primary purpose of a security overlay is to provide DoS-resistant communication for its participants. A security overlay typically utilizes two primary services for its operation (i) an anonymous routing service that hides the overlay traffic and the location of overlay nodes to prevent attacks on the nodes and internode traffic and (ii) a filtering service around the protected target to allow only overlay traffic through. These are usually combined with a user authentication service at the edges of the overlay network to identify legitimate users of the protection service. Depending on the type of authentication used, the routing service used, and the filtering parameters used, the overlay can provide different levels of protection for different types of users.

3.2. Advanced Overlay Topology Design

One of the first considerations in designing an overlay network is to choose an overlay topology to connect the various overlay nodes. The overlay topology chosen is generally application-specific to suit the particular requirements of the service it is designed to provide. The topology chosen has a direct impact on the performance, scalability and overhead, security, and failure resistance of the overlay network. In this section, we discuss topology design issues for different kinds of overlay networks.

Topology construction for routing overlays: Routing overlays are designed to improve the resiliency and performance over the underlying Internet paths. In routing overlays, the overlay topology and the routing protocol have a direct impact on

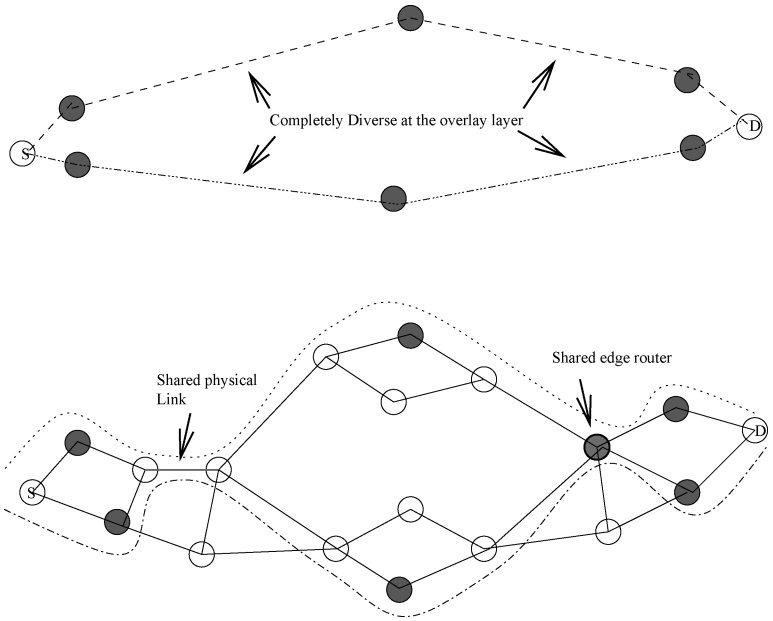

Fig. 11. Overlay path diversity.

the failure resistance and recovery of the overlay network [Li and Mohapatra 2004a]. One overlay topology design characteristic that directly affects failure resistance is path diversity. Path diversity can be considered in two layers: (1) overlay layer and (2) physical layer. Overlay-layer path diversity ensures that overlay-level paths share as few overlay-level links as possible. However, this does not necessarily translate into diversity at the physical layer (see Figure 11). If diverse overlay paths share the same physical links, the failure of the shared physical link breaks both the paths simultaneously. Therefore, building overlay networks with physical diversity provides more resistance to failures and improves recovery.

Among the routing overlays, RON uses overlay-layer path diversity where it forms a full-mesh among all nodes that monitor their connectivity to every other overlay node. This enables a RON node to quickly detect path outages and to choose the best possible alternate path to reach a remote destination. On the other hand, the full-mesh design is not scalable and has a large operational overhead of $O(N^2)$ where N is the number of overlay nodes in the system. One proposal to reduce this overhead is to randomly interconnect overlay nodes bound by certain degree constraints [Chu et al. 2000; Li and Mohapatra 2004a] but this comes with a tradeoff of reducing the degree of connectivity and alternate paths.

An alternative approach for selection of overlay nodes in a physical topology-aware manner is proposed in Han et al. [2005]. In this work, the authors utilize offline analysis of a large quantity of traceroute and ping measurement data for this purpose. From the measurements, they calculate path diversity and latency as metrics to choose the best placement of overlay nodes. Path diversity is calculated as the number of overlapping nodes between the indirect overlay path through an overlay node and the direct native layer path. If two paths through the same node have a high correlation, the paths are assumed to be part of the same cluster. Based on this clustering scheme, a simple heuristic to choose the number and placement of overlay nodes is to

choose one node at random from each cluster. A similar clustering is applied for latency measurements between source destination pairs through an overlay node. The final heuristic uses latency measurements to pick the desired set of nodes from the set of nodes chosen by the path diversity heuristic. Similar ideas are explored in Cui et al. [2002], where the authors explore the problem of assigning backup paths with the aim of minimizing the joint probability of failure between the primary and backup paths. This probability is minimized by selecting backup paths with minimal correlation at the physical level to the primary path.

Another underlying topology-aware topology construction scheme was proposed in Qiu et al. [2003]. In this work, the authors describe a distributed binning scheme to take advantage of the physical proximity between the nodes in an overlay. In this scheme, overlay nodes partition themselves into "bins" based on ping measurements to certain landmarks, for example, DNS servers. Nodes that are physically close to each other group themselves into bins (clusters). Clusters that are proximate to each other can then be clustered together to form a higher-order cluster, and so on. This binning scheme can be used to build an overlay topology with a better routing performance than random node selection. For example, a simple strategy used by the authors to build such an overlay is to have an overlay node pick half of its neighbors from the nodes closest to itself (approximated by picking them at random from its bin) and the other half at random (to maintain connectivity). Even this simple scheme was shown to perform better than constructing a random overlay network.

The relationship between overlay topology and its performance is formalized in Zhang et al. [2006]. In particular, the authors identify three graph-theoretic metrics for the design of highly efficient routing overlay topologies: (i) characteristic path length (CPL); (ii) average cut size; and (iii) the weighted node degree sum. The first, characteristic path length (CPL) is defined as the median of the means of the shortest path lengths connecting each vertex to all other vertices. A small value of CPL provides a better routing performance because the path lengths to be traversed are smaller. The average cut size is a measure of the path diversity available in a graph. A larger cut size implies a more richly connected graph, and hence better performance. While the CPL and average cut size most directly affect routing performance, a third metric, weighted node degree sum (WNDS), is required to compensate for the difference in the utilization levels of links between nodes with different degrees. WNDS assigns larger weights for smaller degrees. Hence, a smaller value of WNDS corresponds to a richly connected graph, and thus a better performance. Based on these metrics, the authors propose a heuristic for node selection and topology design that aims to find a subgraph with small CPL and WNDS but a large cut size.

Failure recovery is an important consideration in the topology construction of a routing overlay. When a failure occurs, the overlay chooses a new overlay-level route in an attempt to route around the failure. The success of routing around the failure depends on two factors: (i) the availability of alternate routes and (ii) the quality of available alternate routes. The availability of alternate routes depends directly on the topology used to build the overlay. The full mesh, for example, provides the maximum number of alternate routes, and hence a better success of failure recovery [Li and Mohapatra 2004a]. The quality of the available alternate routes depends on the correlation in the physical links comprising the alternate overlay paths with the physical links comprising the failed primary path. If there is a high correlation between the paths, the probability that the alternate path is also affected by the same failure as the primary path is high. This highlights the importance of minimum correlation between native layer paths during the construction of the overlay network. Han et al. [2005] develop heuristics for the construction of overlay networks with maximum path diversity (and hence

minimum correlation) through a knowledge of the native layer topology. Similar ideas are explored by Cui et al. [2002], where the authors explore the problem of assigning backup paths with the aim of minimizing the joint probability of failure between the primary and backup paths. This probability is minimized by selecting backup paths with minimal correlation at the physical level to the primary path. The importance of native-layer topology awareness in overlay construction is also highlighted in Li and Mohapatra [2004a]. The authors show that topology-aware approaches have comparable failure recovery ratios (the ratio of paths recovered to total number of failures) and recovered path penalties (the added penalty due to the selection of a less optimal recovery path) to the optimum cases at much less routing overhead when compared to full mesh.

Topology construction for service overlays: QoS-enhancing overlays like SON [Duan et al. 2003] and QRON [Li and Mohapatra 2004b] aim to provide service and bandwidth guarantees to applications. For this purpose, the generalized SON overlay for QoS requires the provisioning of bandwidth-guaranteed tunnels between overlay nodes. The actual topology to be chosen is less evident in this case, but it stands to argue that good end-to-end latency and scalability would be beneficial in SON design. In QRON [Li and Mohapatra 2004b] the authors propose the construction of a global-scale SON to provide QoS guarantees, hence scalability becomes an important consideration. The topology is constructed such that nodes within the same domain are fully meshed, and there is at least one tunnel between two neighboring domains. To improve the scalability of this architecture, the authors propose a hierarchical naming scheme which logically groups overlay nodes into disparate clusters (see Section 2.3).

Vieira and Liebeherr [2004b] propose methods to guide the topology design of a large-scale SON. They aim to minimize the costs associated with the interconnection of overlay nodes across multiple ISPs while providing the best access to end users. The problem is formulated as an optimization problem and proven to be NP-hard. The authors propose multiple heuristics to approximate the optimal solution. Fan and Ammar [2006] study the problem of dynamically reconfiguring SON topologies to suit communication requirements. Such a reconfiguration can allow the SON to better adapt to changing communication requirements, but with an associated cost. The authors aim to minimize the overall cost, which includes the cost of delivering traffic over the network and the cost of reconfiguring the overlay. The problem is shown to be NP-hard, and the authors propose several approximations for it.

Topology construction for security overlays: In security overlays (Section 3.1.3) traffic is tunneled through a series of overlay nodes to reach a protected destination. The primary objective of the overlay design is to provide DoS-resistant communication service to its users and to protect the overlay nodes from DoS attacks. Latency and improving end-to-end performance are not part of the overlay design. For example, the SOS [Keromytis et al. 2002] overlay topology follows a Chord ring [Stoica et al. 2003] which is vital to the DoS-resistant service provided by SOS. The SOS provides an effective solution to DoS defence, but the tradeoff is that the end-to-end latency takes a hit and is increased by a margin of 5 to 8 times over unicast latency. In WebSOS, Stavrou et al. [2005], implement, in addition to Chord, a CAN topology [Ratnasamy et al. 2001] with comparable performance metrics. In FONet [Kurian and Sarac 2007], we move away from using the circuitous routing used in previous proposals, to providing DoS defence using the SON model of service. A third-party OSP deploys overlay nodes with bandwidth guarantees to protect inter-FONet traffic from DoS attacks. Additionally, individual overlay nodes are protected against DoS attacks via the filtering of

undesired traffic. By avoiding the need for circuitous routing and protecting the overlay nodes from attack, FONet improves on the end-to-end performance of applications using these overlays without compromising security.

In addition to DoS attacks, security overlays can also be vulnerable to compromise of overlay nodes. Since overlay nodes are generally made up of end systems deployed by users rather than core routers deployed by ISPs, they are in general more vulnerable to malicious attacks and intrusion. Some authors have explored mechanisms to enhance overlay networks with protection against such attacks. Walters et al. [2010], employ data mining techniques to detect outliers in data reported by overlay nodes.[2] The reasoning behind the proposed technique is that a malicious insider will have difficulty in lying consistently to (i) every other node (spatial outliers) and (ii) over time (temporal outliers). The problem of ensuring Byzantine resiliency and intrusion tolerance has received more attention in P2P overlays [Johansen et al. 2006; Sit and Morris 2002; Singh et al. 2004; Castro et al. 2002].

Wang [2005] and Wang et al. [2005] analyze the vulnerability of structured overlay topologies to two types of intrusion attacks: (i) penetration attacks that aim to directly attack a protected server by compromising nodes in the path to the protected server and (ii) proxy depletion attacks that aim to disable the overlay (proxy) network by compromising all nodes in it. The authors demonstrate that without added protection measures like proxy migration and reconfiguration,[3] the overlay network can be vulnerable to penetration attacks. The vulnerability to penetration attacks is linear to the depth of the overlay topology that is, the number of nodes to traverse to reach the target. In proxy depletion attacks, the overlay topology plays an important part in its resiliency. Specifically the authors observe that topologies with a lower vertex degree and balanced connectivity overall showed better resiliency to depletion attacks. With these requirements, Chord, with its high connectivity, is shown to perform poorly against depletion attacks. Topologies like CAN [Ratnasamy et al. 2001] and the de Brujin graph with lower degrees and well-distributed connectivity are expected to exhibit better resiliency to depletion attacks [Frechette 2005]. For DoS attacks on the nodes, they demonstrate that overlay networks can provide scalable resistance to large-scale DoS attacks. The size of the overlay topology has a linear impact on the volume of attack the overlay can withstand. In contrast to the results on latency observed by SOS and WebSOS [Stavrou et al. 2005], the authors also contend that the overlay network can *improve* the end-to-end performance of the end user due to the presence of long-lived TCP connections between overlay nodes. For a detailed explanation of the results, the reader is referred to Wang [2005].

3.3. Overlay Routing

The feasibility of overlay routing in improving the policy-based network layer in routing performance has been validated by several experimental [Anderson et al. 1999, 2001; Rahul et al. 2006; Zhang et al. 2006] and analytical results [Zhang et al. 2006; Qiu et al. 2003]. Much work has gone into enhancing overlay routing with an aim to improving its performance, enhancing its scalability, and reducing its overhead. Figure 12 shows an overview of some of the relevant work. In this section, we group these works under three topics, as follows:

[2]An outlier is defined as an observation that lies an abnormal distance from other values in a random sample from a population.

[3]Proxy migration and reconfiguration refer to changing the location of nodes randomly, and dynamically changing the overlay topology.

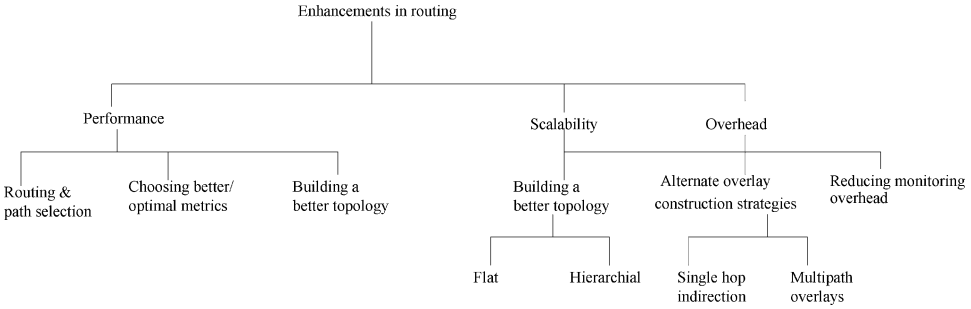

Fig. 12. Overlay routing enhancements tree.

Performance: The end-to-end performance of the overlay network depends on (i) routing protocols and path selection algorithms; (ii) path selection metrics; and (iii) minimal hit-time after failures.

Routing protocol and path selection: The most common routing and path selection approach in overlay routing is a link-state-based proactive approach [Andersen et al. 2001]. In this approach, we assume knowledge of global topology and link-state information. The shortest path is chosen (using Dijkstra's algorithm) for each flow-based on the desired routing metric. Another approach is a link-state, based reactive approach proposed in Zhu et al. [2006]. In the reactive approach, link-state advertisements and global knowledge are assumed as in the proactive case. The difference is that the one chosen initially as the best one is maintained for the subsequent flows, unless the existing path is no longer suited to provide certain performance guarantees. The authors contend that the reactive routing scheme leads to a more stable overlay routing scheme with fewer path changes.

A recent study presented in Rahul et al. [2006] suggests that overlay paths typically have very high persistence (in the order of hours), suggesting that a reactive approach can be well-suited for most overlay needs without sacrificing performance. In QRON [Li and Mohapatra 2004b], the authors suggest two alternative path-selection algorithms that can provide load balancing in addition to satisfying the performance requirements. Another routing approach is the feedback-based approach proposed in Zhao et al. [2003]. In the feedback-based approach, each overlay node maintains a small number (usually two) of backup routes to every other overlay node in its routing table. When the overlay node detects that the primary path is lossy or not available, it switches to one of its backup routes. The backup routes are disjoint at the overlay level (not necessarily at the physical level), and hence have a reasonable probability of being available even if the primary path fails [Li and Mohapatra 2004a].

Metrics: The metrics used for path selection depend on the type of application the overlay intends to support. Choosing the correct metric is important to ensure the best performance for the routing process. The most common metric used is latency as proposed by RON [Andersen et al. 2001]. Latency is well-suited for most network applications, and is the most easily measured via path probes. RON additionally proposed two other metrics: path loss and throughput. Path loss is more difficult to measure as it has to be estimated from the two-way path loss probability of a probe packet. A simplifying assumption is that the bidirection loss is equally divided in both directions. Throughput can be estimated by using the TCP throughput equation based on the observed latency and loss rate. Zhu et al. [2006], propose the use of available (overlay) bandwidth as a metric for path selection. Available overlay bandwidth is defined as

the minimum available bandwidth of all physical links comprising the overlay link. The authors argue that latency, loss rate, and throughput are not directly indicative of traffic load in the path (latency depends primarily on propagation latency rather than traffic load, losses occur only after congestion has already happened, while throughput as measured, in RON is the TCP throughput which depends on factors like flow size and advertised window [Zhu et al. 2006]). However, available bandwidth is not easily measured, and estimation techniques have to be used which can incur added load in the network.

Amir et al. [2005] propose a two-metric routing decision for VoIP applications. In VoIP, the goal is to maximize the number of packets that arrive with a certain threshold for playback at the receiver. Packets that arrive after the threshold are useless, but limited loss of packets can be tolerated. The metric proposed depends on both loss rate and latency of the link, and is called expected latency. In QRON, Li and Mohapatra [2004b] suggest two alternate metrics to use in conjunction with Dikjstra's shortest-path algorithm. Since, in QRON, path selection aims to satisfy the QoS requirement, the main criterion used is available bandwidth. Thus both metrics proposed in QRON are dependent on the available bandwidth and additionally on the available computational capacity of the nodes in the overlay.

Reducing hit time during failure recovery: We define hit time as the time period after the failure of a native link comprising an overlay path during which there is no data flow between a source-destination pair that was using the overlay path. Note that our definition assumes that the overlay *always* finds a new path, and is more general than the usual definition of hit time that defines it for a single overlay link [Seetharaman and Ammar 2006]. The more general definition helps us to account for multipath overlays and other techniques for reducing hit time. In the generalized routing overlay, hit time is comprised of the time to detect the fault, the route convergence time during which all the nodes in the network are aware of the fault and a new route is calculated, and the time taken to switch to the new route. In single-hop indirection overlays [Gummadi et al. 2004], since there is no routing convergence, the hit time is comprised of the time to detect the failure and the time to switch to the new route. Finally, in multipath overlays [Andersen et al. 2003], assuming that there is at least one redundant route between the source and destination that is active, the hit time is zero.

In general routing overlays, the hit time depends directly on the frequency of active probing between the nodes. However, as shown by several authors [Keralapura et al. 2004; Seetharaman and Ammar 2006], higher probe rates lead to an increase in negative interactions between overlays and native traffic, referred to as route flapping. An improved awareness of the native-layer routing process at the overlay layer can reduce the number of route flaps. Multipath overlays [Andersen et al. 2003] provide an answer to this problem at the cost of increased network traffic. Mesh routing is used in these overlays to add redundant packets into the network by duplicating traffic along multiple redundant routes. Multipath routing can be used in conjunction with general routing overlays to tradeoff between the hit-time and the redundancy required in the network.

Scalability and overhead: We consider overhead and scalability together because of their close correlation. In RON Andersen et al. [2001], state that the RON overlay is scalable to around 50 nodes. This limit in scalability is caused by the overhead introduced by the overlay operation. There are three components that make up this overhead: (i) probing or ping overhead between overlay nodes; (ii) link-state broadcasts to announce up-to-date link state; and (iii) the computational overhead required at an overlay node to process state and data traffic. We ignore the computational overhead

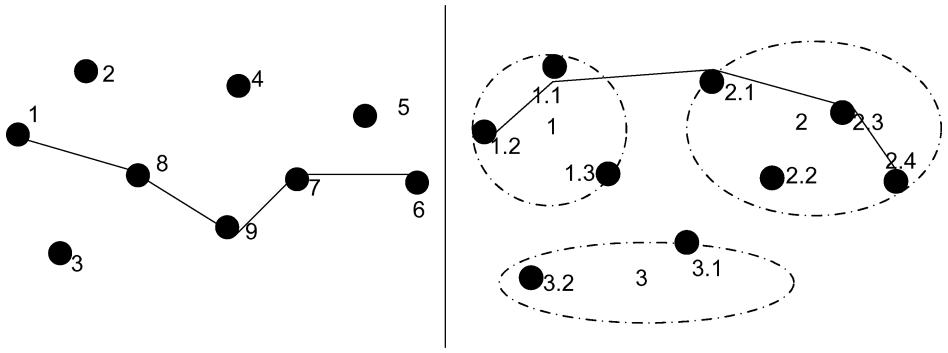

Fig. 13. Flat vs hierarchial organization.

because it is not necessarily an overlay design problem (although as done in QRON [Li and Mohapatra 2004b], residual computational capacity of the nodes can be considered in the routing process). In the generalized routing overlay model, the probing overhead is generally unavoidable (except in SOSR [Gummadi et al. 2004]). In link-state routing protocols (we do not consider feedback-based approaches because, as per Li and Mohapatra [2004a] they are less scalable than link-state approaches) the link state advertisements are also required. The overhead imposed by these two factors becomes excessive in RON due to the full-mesh overlay topology used. It has been shown that a 50-node full-mesh overlay introduces about 30 Kbps routing overhead [Andersen et al. 2001].

There have been two general approaches to solving this scalability problem: (i) approaches that deal with flat topologies and (ii) approaches that deal with hierarchical topologies. Flat topologies like the full-mesh are not scalable, hence several other topologies have been proposed by researchers [Li and Mohapatra 2004a]. Li and Mohapatra [2004a] analyze several such topologies for their routing overhead and conclude that there are several topologies available which, unlike full-mesh, can scale linearly with overlay size. The impact of these alternate topologies on routing performance is quantified in Rewaskar and Kaur [2004]. They observe through large-scale Internet measurements that reducing the degree of connectivity of the overlay topology by a factor of 2 reduces the overhead by a factor of nearly 4. However, this reduction in degree affects the availability of paths with lower latency and loss rate than the default path by a 40% and 30% percent, respectively. Similar observations for probing duration show that doubling the probing duration reduces overhead by half while generating stale routing information for 10% and 30% for latency and loss rates, respectively. The tradeoff is apparent, while full mesh provides the best performance, alternative topologies with a lesser degree of connectivity can provide acceptable results for most requirements. Another approach to reducing the overhead would be to reduce the amount of monitoring required to maintain the overlay network. The task of monitoring $O(N^2)$ paths can be reduced to the task of monitoring k (k is approximately $O(\log N)$) linearly independent paths [Chen et al. 2004]. A detailed discussion of the algorithms is beyond the scope of this survey, and we avoid further discussion.

The overlay topologies considered so far have been flat, requiring every overlay node to have a complete global picture to perform overlay routing. A viable alternative is to use a hierarchical topology (Figure 13). The hierarchical approach is an axial method to enhance the scalability of routing protocols (e.g., OSPF) in the Internet. Hierarchical methods depend on the ability to "aggregate" routing information without incurring

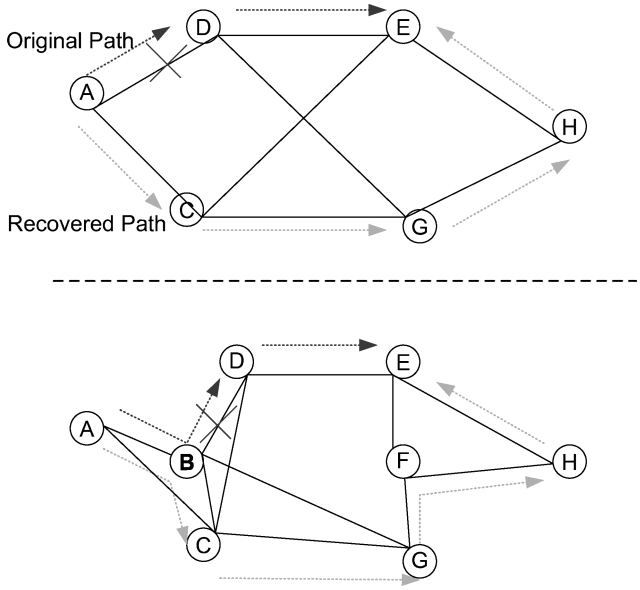

Fig. 14. Overlay recovery after failure.

significant penalties associated with the loss of information. These ideas are extended to overlay networks in QRON [Li and Mohapatra 2004b]. The authors propose a hierarchical organization with nodes organized into clusters based on their proximity. Clusters are further organized into higher-level clusters, and so on, with Level-1 clusters forming the highest level of aggregation. Link state advertisements are made only within the cluster, and gateway nodes in each cluster aggregate the local information to form a full-mesh topology connecting them. A similar clustering approach is proposed in Kostic and Vahdat [2002]. They observe that the hierarchical approach exponentially improves network overhead and scalability, while its performance penalty is within 15% of the optimal.

A third approach to improving the scalability of overlay networks is the single-hop overlay indirection approach [Andersen et al. 2001; Gummadi et al. 2004; Han et al. 2005]. We discussed this approach in detail previously in Section 2.4.

4. INTERACTION BETWEEN OVERLAY AND NONOVERLAY TRAFFIC

An important consideration in the operation of routing overlay networks is their interactions with each other and nonoverlay traffic. Since different overlays and background traffic share many of the same network resources, they may compete for these resources with each other. There are two different types of interactions: (i) interactions between overlay routing and underlay routing and (ii) interactions between different overlay routing schemes.

For an example of how the interaction between overlay and underlay routing can affect both layers negatively, consider Figures 14, 15, and 16 (Note: In these figures, the native layer or underlay is at the bottom, and the overlay layer is at the top). In Figure 14, when the native link BD (at the bottom) fails, the overlay recovers and switches from path ADE to path ACGHE. Some time later, the native layer recovers from the failure and chooses a new path ABCD to route from A to D (Figure 15). The

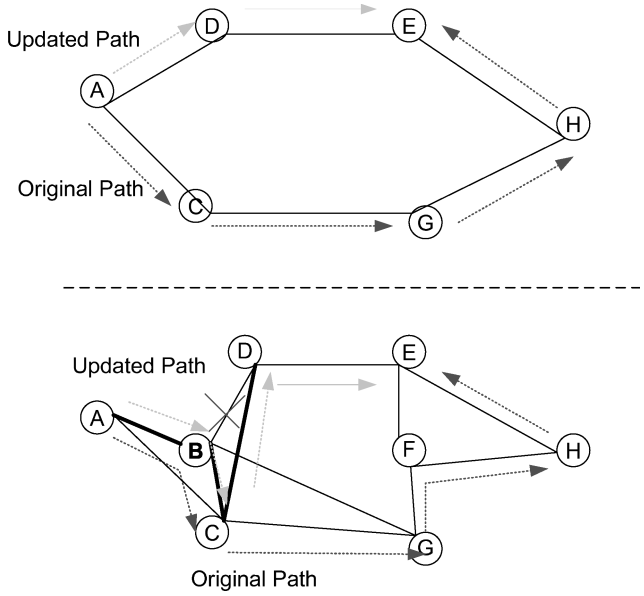

Fig. 15. Overlay rerouting after native recovery.

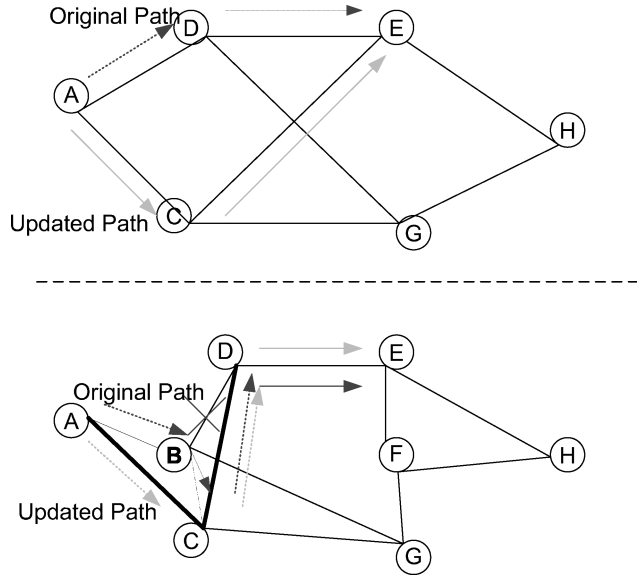

Fig. 16. Overlay recovery after TE reroute.

overlay layer, due to its probing, detects a new, shorter path to E and hence switches back to ADE. Suppose that this switch causes the native link BC to be overloaded. If traffic engineering is present in the domain, it may switch to AC instead of ABC in the original routing tables (Figure 16). The overlay now detects that path ACE is more advantageous than ADE, and hence switches to ACE. These route flaps are caused due to the lack of interaction between the different layers during recovery. Several

authors have explored these interactions and how to accommodate for them in overlay architectures.

Qiu et al. [2003] study interactions between different overlay and underlay routing in the absence of traffic engineering (i.e., the underlying network-level routing remains the same). They observe that in the absence of traffic engineering (TE), different routing schemes can interact well with each other. Overlay routing can provide nearly optimal latency at the expense of added network cost and link utilization. The goal of traffic engineering, however, is to reduce network costs by varying network-level routing to changes in traffic conditions. In this context, both overlay routing and traffic engineering continually adapt to each other to minimize their respective cost functions. The authors separately study this effect with traffic engineering provided by an OSPF route optimizer and an MPLS optimizer. In the OSPF case, they observe that there is a significant performance degradation, so much so that the nonoptimized case outperforms the optimized case. MPLS, based traffic engineering on the other hand performs significantly better than the OSPF case. The authors contend that the MPLS optimizer has much more fine-grained control over overlay traffic, as opposed to the OSPF optimizer which allows it to adjust its routing matrix more effectively.

Liu et al. [2005] further explore the effects of overlay routing on traffic engineering. They model the interaction between the conflicting objectives of overlay routing and traffic engineering as a noncooperative, nonzero sum two-player game.[4] The authors demonstrate that when modeled this way the game has a stable and unique Nash equilibrium point.[5] A discussion of the Nash equilibrium is beyond the scope of this article; but note that Nash equilibrium is not an efficient state for either player. The selfish behavior of overlay routing and its interaction with traffic engineering as a result degrades the performance of regular users and the underlay network as a whole.

Keralapura et al. [2004], examine the interactions between overlay routing and traffic engineering in the presence of unexpected events like failures. They identify that overlay routing violates two basic assumptions made by ISPs in their traffic engineering policies: (i) traffic demand is relatively constant over shorter periods of time and (ii) changes in the path within a domain do not impact traffic demands. This leads to frequent oscillations in routing and makes traffic matrices more dynamic and difficult to predict. Traffic engineering is also responsible for implementing load-balancing policies of the ISP. Overlay routing can bypass these load-balancing requirements-violating the ISPs load-balancing intent. Another consideration is the case of a single overlay that spans multiple AS domains. The effects of a failure of a physical link in one of the domains can cause the overlay to switch its paths, affecting the load on links in other domains. This is an undesired effect and can lead to oscillations in domains due to an event in another domain.

Seetharaman and Ammar [2006] study the behavior of networks in which the overlay layer and the native layer operate independently of each other. The problem arises due to lack of coordination in the failure-recovery mechanisms of the two layers. As described in Section 3.1.1, a routing overlay uses probe messages to detect failures (often quicker than the native layer) and finds its own alternate path. This independence results in a dual rerouting at the two layers, and hence to oscillations. Completely avoiding this recovery process is also not ideal and can lead to partitioned overlays.

[4]In a noncooperative, nonzero sum game, the players act in their own self interest, and a gain by one player does not necessarily imply a loss by the other player.

[5]Nash equilibrium: When everyone plays their best move to everyone elses best move, no one is going to move.

Hence, the authors suggest an improved "awareness" of the underlay recovery at the overlay layer. The overlay layer suppresses or delays its own rerouting process in deference to the recovery process at the native layer. The authors demonstrate that, in this way, the number of oscillations due to dual rerouting are reduced. However, the tradeoff is the time required to recover from the failure at the overlay layer, which may now depend on the speed at which the underlay recovers. Seetharaman et al. [2007] suggest preemptive strategies for each layer that try to prevent the other layers from needing to make a readjustment which could potentially lead to oscillations. For the overlay layer, this amounts to making available bandwidth measurements on the native-layer links and limiting overlay bandwidth consumption to be less than the available bandwidth. For the native layer, the strategy takes into account the fact that overlay routing will choose paths with the lowest latencies. So during its TE calculations, the native layer tries to ensure that the hop-count of paths between source-destination pairs stays within a small threshold of its previous value.

Keralapura et al. [2005] examine the interactions that occur when multiple overlays coexist. Since each overlay takes independent routing decisions without knowledge of the other, oscillations in network load and routes are both possible. The authors identify three conditions for such oscillations : (i) a failure or path degradation event which triggers the recovery event in overlays; (ii) a shared link between the two overlays; and (iii) correlation between probe periods of the overlays. The aggressiveness of each overlay plays an important part in the probability that two overlays will get synchronized and go into oscillations. Generally, an increase in aggressiveness (defined as the the ratio of probe timeout and probe interval) translates to a higher probability of synchronization.

Oscillations are detrimental to both overlay and nonoverlay traffic. A careful consideration of the impact of a new overlay needs to be done before deployment. Overlays potentially can also benefit from a common probing layer similar to that suggested by Nakao et al [2003].

5. THE PLURALIST VS PURIST ARGUMENT AND OVERLAY DEPLOYMENT MODELS

Recently, there has been much debate in the network community about the long-term impact of overlay networks in the Internet. The traditional or purist view sees overlay networks as a means to an end. As per the purist view, overlay networks are testbeds for the implementation and experimentation of novel architectures in the Internet [Peterson et al. 2004; Ratnasamy et al. 2005]. Some authors have advanced the purist view [Ratnasamy et al. 2005] by presenting solutions within the existing architecture that allow it to evolve and solve some of the problems that seem to necessitate overlay networks. The pluralist, on the other hand, views overlay networks as an end in itself. As per the pluralist view, in the next-generation Internet, multiple overlay networks will be deployed by third-party OSPs to cater to the various requirements of different users. In such an Internet architecture, also called the virtualized Internet [Peterson et al. 2004], multiple applications will coexist, and traditional IP will be one such application.

In this article we do not seek to forward either view of overlay networks. However, the presence of overlay networks in the future of the Internet is undeniable. It is in this context that we motivate an overlay deployment model in which ISPs will actively participate in the deployment and operation of overlay networks. In our discussions so far we have seen two different overlay deployment models, the SON model, which has some participation from the ISPs, and the P2P model, where the ISP is unaware of the overlay presence. In the proposed third model of overlay deployment, henceforth called the provider-provisioned overlay model (PON) [Kurian and Sarac 2007], ISPs

will be involved statically with the OSPs for the deployment of overlay networks and dynamically during overlay operation by exchanging information between the native and overlay layers.

In the PON model, overlays will be built in a collaborative manner by a number of participating ISPs and OSPs. ISPs deploy PON nodes in their domains and offer them to OSPs. ISPs may also provide additional support for the operation of the overlay to the OSP and charge the OSP for the resources and support provided. The OSP will lease a number of PON nodes from multiple ISPs and deploy overlay-based applications on top of them. It offers these applications as value-added services to interested users and charges the users for the service. PON model described above has several advantages over existing deployment models, and can potentially better support many of the overlay applications in the Internet today. Some of the advantages of the PON model include the following:

ISP, OSP, and end-user friendliness: The PON is deployed in a federated manner, with multiple ISPs leasing out their local PON nodes to the OSP. This avoids the need for a single entity (the OSP) to spend potentially prohibitive amounts of money and resources to deploy overlay nodes across multiple domains. By charging the OSP for the resources and support, the ISP can gain added revenue from the overlay. This ISP friendliness provides an incentive to ISPs to deploy PON overlays in their networks. The PON model also empowers the end user of the value-added service. In the current model, the user is provided a guarantee (if any) inside its provider ISP domain only. In the PON model, OSPs are responsible entities for guaranteeing end-to-end services.

Deploying end-to-end services: In the Internet today, the deployment of end-to-end services (for example, QoS, multicast, IP traceback) have proven to be notoriously difficult. The lack of global cooperation and incentives for deployment are cited as the main reasons behind this difficulty [Peterson et al. 2004]. In PON, the presence of the OSP to coordinate between different ISPs solves this coordination problem in the deployment of end-to-end services.

Building better overlay architectures: The PON model allows for the active involvement of the ISPs during overlay operation and deployment. This enables PON overlays to better support some of the applications considered by traditional overlays. Routing overlays [Andersen et al. 2001] can be enhanced with locally available routing information to reduce probing costs and to provide best routes. During the construction of routing overlays, it has been shown that an awareness of the underlying native-layer topology is important for overlays, with better overall performance and failure resistance [Li and Mohapatra 2004a; Han et al. 2005]. In probing-based routing overlays, costly oscillations may occur if the overlay and native layers are unaware of each other. The involvement of ISPs in overlay operation can help avoid such oscillations by provisioning for overlay traffic in traffic engineering calculations. QoS provisioning overlays [Duan et al. 2003] can be enhanced with MPLS-based path protection and bandwidth guarantees, packet marking and labeling at the edges of the domains, selection of best nodes and routes for building the overlay, and a solid business model. DoS defense through overlays is another application that has received a lot of attention in the research community in recent years [Keromytis et al. 2002; Andersen 2003; Shi et al. 2006; Adkins et al. 2003]. Again, the PON model can be used to enhance generalized overlay-based architectures to better support the application. As an example, we have proposed the FONet architecture [Kurian and Sarac 2007], which is a PON-based overlay to build a better overlay architecture (called FONet) for DoS defense in the Internet.

The pluralist vs purist views of overlay deployment offer an intriguing perspective into the future model of Internet deployment. The PON model can serve both the purist and pluralist models of overlay deployment. If the pluralist view prevails, PON can serve as a transition model to the next generation virtualized Internet model. The overlay nodes that are deployed in the core of the network can be high-capacity computing resources that may correspond to next-generation switching devices in the context of the next-generation Internet environment. If the purist view prevails, ISPs can use PON overlays to earn added revenue by providing end-to-end value-added services to their customers.

6. CONCLUSION

In this article we have presented a comprehensive survey of overlay networks, the important applications, and the strengths and weaknesses of the proposed architectures for each application. We also formulate generalized overlay architectures for three of the important applications, routing, QoS, and security. Based on these generalized models, we have described proposed enhancements in topology design, routing, failure resistance, and the interaction between overlay and native layers. Finally, we briefly present the pluralist vs purist view of overlay networks and propose the PON model of overlay deployment which meshes well with both views.

REFERENCES

ABAD, C., YURCIK, W., AND CAMPBELL, R. H. 2004. A survey and comparison of end-system overlay multicast solutions suitable for network centric warfare. *Proc. of SPIE*.

ABE, M. 1999. Mix-networks on permutation networks. In *Advances in Cryptology. Proceedings of ASIACRYPT*. Lecture Notes in Computer Science, vol. 1716, Springer, Berlin. DOI: 10.1007/b72231.

ACCELIA. Accelia Durasite Technologies. http://www.accelia.net/.

ADKINS, D., LAKSHMINARAYANAN, K., PERRIG, A., AND STOICA, I. 2003. Taming IP packet flooding attacks. In *Proceedings of the 2nd ACM HotNets Workshop*. ACM, New York.

AGARWAL, S., PADMANABHAN, V. N., AND JOSEPH, D. A. 2005. SureMail: Notification overlay for email reliability. In *Proceedings of the ACM HotNets IV Workshop*. ACM, New York.

AKAMAI. Akamai sureroute for failover. http://www.akamai.com/en/html/services/sureroute for failover.html.

AKAMAI. Akamai Technologies Inc. http://www.akamai.com.

AKAMAI. Turbo-charging dynamic web sites with Akamai edgesuite. http://www.developer.akamai.com/ pdf/WP TCD.pdf.

ALMEROTH, K. 2000. A long-term analysis of growth and usage patterns in the multicast backbone (MBone). In *Proceedings of the IEEE INFOCOM*. IEEE, Los Alamitos, CA.

AMIR, Y., DANILOV, C., GOOSE, S., HEDQVIST, D., AND TERZIS, A. 2005. 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*. ACM, New York, 51–56.

ANDERSEN, D. 2003. Mayday: Distributed filtering for Internet services. In *Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems*. USENIX Association, Berkeley, CA.

ANDERSEN, D., BALAKRISHNAN, H., KAASHOEK, M., AND MORRIS, R. 2001. Resilient overlay networks. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles*. ACM, New York, 131–145.

ANDERSON, D.G., BALAKRISHNAN, H., KAASHOEK, M.F., AND RAO, R. 2005. Improving web availability for clients with MONET. In *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI)*.

ANDERSON, D. G., SNOERENY, A. C., AND BALAKRISHNAN, H. 2003. Best-path vs. multi-path overlay routing. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*. ACM, New York.

ANDERSON, T., ROSCOE, T., AND WETHERALL, D. 2003. Preventing Internet denial-of-service with capabilities. In *Proceedings of the 2nd ACM HotNets Workshop*. ACM, New York.

ANDERSON, T., SAVAGE, S., AGGARWAL, A., BECKER, D., CARDWELL, N., COLLINS, A., HOFFMAN, E., SNELL, J., VAHDAT, A., VOELKER, G., AND ZAHORJAN, J. 1999. Detour: A case for informed internet routing and transport. *IEEE Micro 19,* 1, 50–59.

ANONYMIZER. Internet privacy and secuity solutions. http://www.anonymizer.com/.

BALAKRISHNAN, H., STEMM, M., SESHAN, S., AND KATZ, R. H. 1997. Analyzing stability in wide-area network performance. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. ACM, New York, 2–12.

BALLANI, H. AND FRANCIS, P. 2005. Towards a global IP anycast service. *ACM SIGCOMM Comput. Comm. Rev. 35*, 4, 301– 312.

BANERJEE, S., BHATTACHARJEE, B., AND KOMMAREDDY, C. 2002. Scalable application layer multicast. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. ACM, New York, 205–217.

BASET, S. A. AND SCHULZRINNE, H. G. 2006. An analysis of the Skype peer-to-peer Internet telephony protocol. In *Proceedings of the IEEE INFOCOM*. IEEE, Los Alamitos, CA.

BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., WANG, Z., AND WEISS, W. 1998. An architecture for differentiated services. Rfc2475.

BLUMENTHAL, M. AND CLARK, D. 2001. Rethinking the design of the internet: The end to end arguments vs. the brave new world. *ACM Trans. Internet Technol. 1,* 1, 70–109.

BRADEN, R., CLARK, D., AND SHENKER, S. 1994. Integrated services in the internet architecture: an overview. Rfc 1633.

BRESLAU, L., CAO, P., FAN, L., PHILLIPS, G., AND SHENKER, S. 1999. Web caching, and zipf-like distributions: Evidence, and implications. In *Proceedings of the IEEE INFOCOM*. IEEE, Los Alamitos, CA.

CARPENTER, B. E. AND NICHOLS, K. 2002. Differentiated services in the internet. *IEEE Proc. 90*, 9, 1479–1494.

CASTRO, M., DRUSCHEL, P., GANESH, A., ROWSTRON, A., AND WALLACH, D. 2002. Secure routing for structured peer-to-peer overlay networks. In *Proceedings of the 5th USENIX Symposium on Operation System Design and Implementation* (OSDI). USENIX Association, Berkeley, CA.

CHANDRASHEKAR, J., ZHANG, Z.-L., DUAN, Z., AND HOU, Y. T. 2003. Service oriented Internet. In *Proceedings of the 1st International Conference on Service Oriented Computing*.

CHAWATHE, Y. 2003. Scattercast: An adaptable broadcast distribution framework. *ACM Multimedia Syst. J, 9,* 1, (Special issue on Multimedia Distribution). 104–118.

CHEN, Y., BINDEL, D., SONG, H., AND KATZ, R. 2004. An algebraic approach to practical and scalable overlay network monitoring . In *ACM SIGCOMM Comput. Comm. Rev. 34*, 4, 55–66.

CHU, Y.-H., RAO, S. G., AND ZHANG, H. 2000. A case for end system multicast. In *Proceedings of ACM SIGMETRICS*. ACM, New York.

CHUN, B., CULLER, D., ROSCOE, T., BAVIER, A., PETERSON, L., AND WAWRZONIAK, M. 2003. PlanetLab: An overlay testbed for broad-coverage services. *ACM SIGCOMM Comput. Comm. Rev. 33,* 3, 3–12.

CODEEN. CoDeeN: A CDN on PlanetLab. http://codeen.cs.princeton.edu/.

CROWCROFT, J., HAND, S., MORTIER, R., ROSCOE, T., AND WARFIELD, A. 2003. Qos's downfall: At the bottom, or not at all! In *Proceedings of the ACM SIGCOMM Workshop on Revisiting IP QoS: What Have We Learned, Why Do We Care?* ACM, New York.

CUI, W., STOICA, I., AND KATZ, R. H. 2002. Backup path allocation based on a correlated link failure probability model in overlay networks. In *Proceedings of the 10th IEEE International Conference on Network Protocols*. IEEE, Los Alamitos, CA.

DINGLEDINE, R., MATHEWSON, N., AND SYVERSON, P. 2004. Tor: the second-generation Onion router. In *Proceedings of the USENIX Security Conference*. USENIX Association, Berkeley, CA.

DUAN, Z., ZHANG, Z.-L., AND HOU, Y. T. 2003. Service overlay networks: SLAs, QoS, and bandwidth provisioning. *IEEE/ACM Trans. Network. 11*, 6, 870–883.

DUVVURI, V., SHENOY, P., AND TEWARI, R. 2003. Adaptive leases: A strong consistency mechanism for the world wide web. *IEEE Trans. Knowl. Data Eng. 15,* 5, 1266–1276.

EDGESTREAM. Edgestream Videostreaming platform. http://www.edgestream.com/.

FAN, J. AND AMMAR, M. 2006. Dynamic topology configuration in service overlay networks: A study of reconfiguration policies. In *Proceedings of the IEEE INFOCOM*, IEEE, Los Alamitos, CA.

FRANCIS, P. Yoid: Extending the internet multicast architecture. Unrefereed report. http://www.isi.edu/ div7/yoid/docs/ycHtmlL/htmlRoot.html.

FRECHETTE, S. 2005. A proxy-network based overlay topology resistant to dos attacks and partitioning. In *Proceedings of the 19th IEEE IPDPS*. IEEE, Los Alamitos, CA.

FREEDMAN, M. J., LAKSHMINARAYANAN, K., AND MAZIRES, D. 2006. OASIS: Anycast for any service. In *Proceedings of the 3rd USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*.

FREENET PROJECT. The free network project : A distributed anonymous information storage and retrieval system. freenetproject.org.

GLOBULE. Globule: the Open-Source Content Distribution Network. http://www.globule.org/index.html.

GOLDSCHLAG, D., REEDY, M., AND SYVERSONY, P. 1999. Onion routing for anonymous and private internet connections. *Comm. ACM 42,* 2, 39–41.

GUMMADI, K., MADHYASTHA, H., GRIBBLE, S. D., LEVY, H. M., AND WETHERALL, D. J. 2004. Improving the reliability of internet paths with one-hop source routing. In *Proceedings of the 6th USENIX Symposium on Operating Systems Design and Implementation (OSDI)* USENIX Association, Berkeley, CA.

GWERTZMAN, J. AND SELTZER, M. 1996. World-wide web cache consistency. In *Proceedings of the USENIX Technical Conference*. USENIX Association, Berkeley, CA.

HAN, J., WATSON, D., AND JAHANIAN, F. 2005. Topology aware overlay networks. In *Proceedings of the IEEE INFOCOM*. IEEE, Los Alamitos, CA.

JANNOTTI, J., GIFFORD, D., JOHNSON, K., KAASHOEK, M., AND O'TOOLE, J. 2000. Overcast: Reliable multicasting with an overlay network. In *Proceedings of the 4th USENIX OSDI*. USENIX Association, Berkeley, CA, 197–212.

JAP. Jap: Java anonymous proxy. http://sourceforge.net/projects/anon/.

JOHANSEN, H., ALLAVENA, A., AND VAN RENESSE, R. 2006. Fireflies: Scalable support for intrusion-tolerant network overlays. In *Proceedings of EUROSYS*.

KERALAPURA, R., CHUAH, C.-N., TAFT, N., AND IANNACCONE, G. 2005. Can coexisting overlays inadvertently step on each other? In *Proceedings of the 13th IEEE International Conference on Network Protocols*. IEEE, Los Alamitos, CA.

KERALAPURA, R., TAFT, N., CHUAH, C.-N., AND IANNACCONE, G. 2004. Can ISPs take the heat from overlay networks? In *Proceedings of the ACM HotNets Workshop III*. ACM, NewYork.

KEROMYTIS, A. D., MISRA, V., AND RUBENSTEIN, D. 2002. SOS: Secure overlay services. In *Proceedings of the ACM SIGCOMM*. ACM, New York.

KOSTIC, D., RODRIGUEZ, A., ALBRECHT, J., AND VAHDAT, A. 2003. Bullet: High bandwidth data dissemination using an overlay mesh. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles*. ACM, New York.

KOSTIC, D. AND VAHDAT, A. 2002. Latency versus cost optimizations in hierarchical overlay networks. Tech. rep., Duke University.

KRISHNAMURTHY, B., WILLS, C., AND ZHANG, Y. 2001. On the use and performance of content distribution networks. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*. ACM, New York.

KURIAN, J. AND SARAC, K. 2007. Provider provisioned overlay networks and their utility in DoS defense. In *Proceedings of the IEEE Globecomm*. IEEE, Los Alamitos, CA.

KWON, M. AND FAHMY, S. 2002. Topology-aware overlay networks for group communication. In *Proceedings of NOSSDAV*.

LABOVITZ, C., AHUJA, A., BOSE, A., AND JAHANIAN, F. 2000. Delayed internet routing convergence. In *Proceedings of ACM SIGCOMM*. ACM, New York.

LI, Z. AND MOHAPATRA, P. 2004a. The impact of topology on overlay routing service. In *Proceedings of the IEEE INFOCOM* IEEE, Los Alamitos, CA.

LI, Z. AND MOHAPATRA, P. 2004b. QRON: QoS-aware routing in overlay networks. *IEEE J. Select. Areas Comm. 22,* 1, (Special Issue on Recent Advances on Service Overlay Networks). 29–40.

LIANG, J., KO, S., GUPTA, I., AND NAHRSTEDT, K. 2005. Mon: On-demand overlays for distributed system management. In *Proceedings of the 2nd USENIX Workshop on Real, Large, Distributed Systems*. USENIX Association, Berkeley, CA.

LIU, Y., ZHANG, H., GONGY, W., AND TOWSLEY, D. 2005. On the interaction between overlay routing and underlay routing. In *Proceedings of the IEEE INFOCOM*. IEEE, Los Alamitos, CA.

LUA, E. K., CROWCROFT, J., PIAS, M., SHARMA, R., AND LIM, S. 2004. A survey and comparison of peer-to-peer overlay network schemes. *ACM Comput. Surv. 36,* 4, 335–371.

MACEDONIA, M. AND BRUTZMAN, D. 1994. MBone provides audio and video across the Internet. *IEEE Computer 27,* 4, 30–36.

MIRKOVIC, J., PRIER, G., AND REIHER, P. 2002. Attacking DDoS at the source. In *Proceedings of the IEEE International Conference on Network Protocols*. IEEE, Los Alamitos, CA.

MOGHUL, J. 2000. Squeezing more bits out of http caches. *IEEE Network 14,* 3, 6–14.

MOORE, D., VOELKER, G., AND SAVAGE, S. 2001. Inferring internet denial-of-service activity. In *Proceedings of the USENIX Security Symposium*. USENIX Association, Berkeley, CA.

NAKAO, A., PETERSON, L., AND BAVIER, A. 2003. A routing underlay for overlay networks. In *Proceedings of ACM SIGCOMM*. ACM, New York.

NETLI. Netli Application Delivery Network. http://www.netli.com/.

NINAN, A., KULKARNI, P., SHENOY, P., RAMAMRITHAM, K., AND TEWARI, R. 2001. Cooperative leases: Scalable consistency maintenance in content distribution networks. In *Proceedings of the World Wide Web 10*.

OIKONOMOU, G., REIHER, P., ROBINSON, M., AND MIRKOVIC, J. 2006. A framework for collaborative ddos defense. In *Proceedings of the Annual Computer Security Applications Conference*.

PARK, K., PAI, V. S., PETERSON, L., AND WANG, Z. 2004. Codns: Improving dns performance and reliability via cooperative lookups. In *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*.

PAXSON, V. 1997. End-to-end routing behaviour in the internet. *IEEE/ACM Trans. Network. 5*, 5, 601–615.

PENDARAKIS, D., SHI, S., VERMA, D., AND WALDVOGEL, M. 2001. ALMI: An application level multicast infrastructure. In *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems*. USENIX Association, Berkeley, CA.

PETERSON, L., SHENKER, S., AND TURNER, J. 2004. Overcoming the Internet impasse through virtualization. In *Proceedings of the 3rd ACM Workshop on Hot Topics in Networks (HotNets-III)*. ACM, New York.

PRIVOXY. Privoxy: A privacy enhancing http proxy. http://www.privoxy.org/.

QBONE. Qbone: A test bed for differentiated services. http://www.isoc.org/inet99/proceedings/4f/4f 1.htm.

QIU, L., YANG, Y. R., ZHANG, Y., AND SHENKER, S. 2003. On selfish routing in Internet-like environments. In *Proceedings of the ACM SIGCOMM*. ACM, New York.

RAHUL, H., KASBEKAR, M., SITARAMAN, R., AND BERGER, A. 2006. Towards realizing performance and availability with a global overlay network. In *Proceedings of Passive and Active Measurement Conference (PAM)*.

RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SHENKER, S. 2001. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM*. ACM, New York.

RATNASAMY, S., SHENKER, S., AND MCCANNE, S. 2005. Towards and evolvable internet architecture. In *Proceedings of ACM SIGCOMM*. ACM, New York.

REWASKAR, S. AND KAUR, J. 2004. Testing the scalability limits of overlay routing infrastructures. In *Proceedings of the 5th Passive and Active Measurements Workshop (PAM)*.

6BONE: A testbed for deployment of IPv6. http://www.6bone.net/old 6bone home page.html.

SAVAGE, S., COLLINS, A., HOFFMAN, E., SNELL, J., AND ANDERSON, T. 1999. The end-to-end effects of internet path selection. In *Proceedings of ACM SIGCOMM*. ACM, New York.

SEETHARAMAN, S. AND AMMAR, M. 2006. On the interaction between dynamic routing in the overlay and native layers. In *Proceedings of the IEEE INFOCOM*. Los Alamitos, CA.

SEETHARAMAN, S., HILT, V., HOFFMAN, M., AND AMMAR, M. 2007. Preemptive strategies to improve routing performance of native and overlay layers. In *Proceedings of the IEEE INFOCOM*. Los Alamitos, CA.

SHI, E., STOICA, I., ANDERSEN, D., AND PERRIG, A. 2006. Overdose: A generic DDOS protection service using an overlay network. Tech. rep., Carnegie Mellon University. http://reports-archive. adm.cs.cmu.edu/anon/2006/CMU-CS-06-114.pdf.

SHIN, Y., GUPTA, M., AND HENDERSON, R. 2006. Separating wheat from the chaff: A deployable approach to counter spam. In *Proceedings of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet*. Vol. 2. USENIX Association, Berkeley, CA.

SINGH, A., CASTRO, M., DRUSCHEL, P., AND ROWSTRON, A. 2004. Defending against eclipse attacks on overlay networks. In *Proceedings of the 11th European ACM SIGOPS Workshop*. ACM, New York.

SIT, E. AND MORRIS, R. 2002. Security considerations for peer-to-peer distributed hash tables. In *Proceedings of the 1st International Workshop on Peer-To-Peer Systems*.

SNOEREN, A. C., BALAKRISHNAN, H., AND KAASHOEK, M. F. 2001. Reconsidering internet mobility. In *Proceedings of 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)*.

STAVROU, A., COOK, D. L., MOREIN, W. G., KEROMYTIS, A. D., MISRA, V., AND RUBENSTEIN, D. 2005. Websos: An overlay-based system for protecting web servers from denial of service attacks. *Elsevier J. Comput. Networks 48,* 5, (Special Issue on Web and Network Security). 781–807.

STAVROU, A., KEROMYTIS, A., NIEH, J., MISRA, V., AND RUBENSTEIN, D. 2005. Move: An end-to-end solution to network denial of service. In *Proceeding of the Internet Society (ISOC) Symposium on Network and Distributed Systems Security* (*SNDSS*).

STAVROU, A. AND KEROMYTIS, A. D. 2005. Countering DOS attacks with stateless multipath overlays. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*. ACM, New York.

STOICA, I., ADKINS, D., ZHUANG, S., SHENKER, S., AND SURANA, S. 2002. Internet indirection infrastructure. In *Proceedings of ACM SIGCOMM*. ACM, New York.

STOICA, I., MORRIS, R., LIBEN-NOWELL, D., KARGER, D., KAASHOEK, M., DABEK, F., AND BALAKRISHNAN, H. 2003. Chord: A scalable peer-to-peer lookup protocol for Internet applications. *IEEE/ACM Trans. Network. 11,* 1, 33–46.

STONE, R. 2000. Centertrack: An IP overlay network for tracking dos floods. In *Proceedings of the 9th USENIX Security Symposium*. USENIX Association, Berkeley, CA.

SU, A., CHOFFNES, D., KUZMANOVIC, A., AND BUSTAMANTE, F. E. 2006. Drafting behind akamai (travelocity-based detouring). In *Proceedings of ACM SIGCOMM*. ACM, New York.

SUBRAMANIAN, L., STOICA, I., BALAKRISHNAN, H., AND KATZ, R. 2004. OverQoS: An overlay based architecture for enhancing Internet QoS. In *Proceedings of the 1st Symposium on Networked Systems Design and Implementation (NSDI)*.

TOUCH, J. D., WANG, Y.-S., PINGALI, V., LARS EGGERT, AND R. Z., AND FINN, G. G. 2005. A global X-bone for network experiments. In *Proceedings of Testbeds and Research Infrastructures for the Development of Networks and Communities*.

TURNER, J. S. AND TAYLOR, D. E. 2005. Diversifying the Internet. In *Proceedings of IEEE GLOBECOM*. IEEE, Los Alamitos, CA.

URGAONKAR, B., NINAN, A., RAUNAK, P. S. M., AND RAMAMRITHAM, K. 2001. Maintaining mutual consistency for cached Web objects. In *Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS-21)*.

VENU, J. 2002. Content delivery networks poised to take off. White paper for ITtoolbox Networking. http://networking.ittoolbox.com/pub/JB061502.pdf.

VIEIRA, S. L. AND LIEBEHERR, J. 2004a. Topology design for service overlay networks with bandwidth guarantees. In *Proceedings of the 12th IEEE International Workshop on Quality of Service (IWQoS)*.

VIEIRA, S. L. AND LIEBEHERR, J. 2004b. An algorithmic approach to topological design of service overlay networks. In *Proceedings of the 12th IEEE International Workshop on Quality of Service (IWQoS)*.

WALTERS, A., BAUER, K., AND NITA-ROTARU, C. Towards robust overlay networks: Enhancing adaptivity mechanisms with Byzantine-resilience. http://www.homes.cerias.purdue.edu/crisn/papers/adapt.pdf.

WANG, J. 2005.T olerating denial-of-service attacks: A system approach. Ph.D. dissertation, Department of Engineering and Computer Science, University of California, San Diego.

WANG, J., LIU, X., AND CHIEN, A. A. 2005.E mpirical study of tolerating denial-of-service attacks with a proxy network. In *Proceedings of the 14th ACM/USENIX Security Symposium*. ACM, New York.

XU, K. AND ZHANG, Z.-L. 2005.R educing unwanted traffic in a backbone network. In *Proceedings of the SRUTI*.

YU, H., BRESLAU, L., AND SHENKER, S. 1999.A scalable web cache consistency architecture. In *Proceedings of ACM SIGCOMM*. ACM, New York.

ZHANG, H., KUROSE, J., AND TOWSLEY, D. 2006.C an an overlay compensate for a careless underlay? In *Proceedings of the IEEE INFOCOM*. IEEE, Los Alamitos, CA.

ZHAO, B. Y., HUANG, L., STRIBLING, J., JOSEPH, A. D., AND KUBIATOWICZ, J. D. 2003. Exploiting routing redundancy via structured peer-to-peer overlays. In *Proceedings of the 11th IEEE ICNP*. IEEE, Los Alamitos, CA.

ZHAO, B. Y., HUANG, L., STRIBLING, J., RHEA, S. C., AND JOSEPH, A. D. 2004. Tapestry: A resilient global-scale overlay for service deployment. *IEEE J. Select. Areas Comm. 22,* 1, 41–53.

ZHU, Y., DOVROLIS, C., AND AMMAR, M. 2006. Dynamic overlay routing based on available bandwidth estimation: A simulation study. *Comput. Networks: Int. J. Comput. Telecommun. Network. 50,* 6, 742–762.

ZHU, Y., DOVROLIS, C., AND AMMAR, M. 2007. Combining multihoming with overlay routing (or, how to be a better ISP without owning a network). In *Proceedings of the IEEE INFOCOM*. IEEE, Los Alamitos, CA.

Received August 2007; revised June 2008; accepted April 2009