UFO: A Resilient Layered Routing Architecture

Yaping Zhu *, Andy Bavier *, Nick Feamster †, Sampath Rangarajan ‡
and Jennifer Rexford *
* Princeton University, † Georgia Tech, ‡ NEC Labs America

This article is an editorial note submitted to CCR. It has NOT been peer reviewed. Authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

Conventional wisdom has held that routing protocols cannot achieve both scalability and high availability. Despite scaling relatively well, today's Internet routing system does not react quickly to changing network conditions (e.g., link failures or excessive congestion). Overlay networks, on the other hand, can respond quickly to changing network conditions, but their reliance on aggressive probing does not scale to large topologies. The paper presents a layered routing architecture called UFO (Underlay Fused with Overlays), which achieves the best of both worlds by having the "underlay" provide explicit notification about network conditions to help improve the efficiency and scalability of routing overlays.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.2.3 [**Computer-Communication Networks**]: Network Operations

General Terms

Routing Architecture, Overlay Networks, Network Monitoring

1. INTRODUCTION

The Internet today must provide a routing infrastructure which satisfies the requirements of scalability, high end-to-end availability, and customized route selection in a cost-effective way. At the IP layer, today's "one size fits all" routing system scales well, but at the expense of availability and customized route selection. In particular, since the Border Gateway Protocol (BGP) faces the fundamental challenge of scalability, only limited BGP routes are announced according to the policy. Moreover, BGP suffers from poor convergence caused by path exploration, since the BGP update messages are not tagged with the root cause, routers can not react effectively. Transient disruptions during routing-protocol convergence [1, 2] degrade the performance of interactive applications like Voice over IP (VoIP) [3, 4, 5] and online gaming. In addition, today's routing protocols do not satisfy the diverse performance requirements of different services. For example, some applications may prefer high-throughput paths while others prefer low loss and low delay, but today's routing protocols direct all traffic over the same set of paths.

At the application layer, a Resilient Overlay Network (RON) [6] trades scalability for high end-to-end availability and customized route selection. RON uses a small collection of hosts to form a topology where each overlay link traverses one or more hops in the underlying network. RON provides high end-to-end availability by using probes to detect changes in the underlying network, and reroute according to the needs of applications. Studies have shown that RON can react more quickly than IP routing to changes in network conditions (e.g., links failures or excessive congestion) [7]. Moreover, with alternative routes available, RON also provides customized route selection, to meet the diverse requirements of different applications. Unfortunately, frequent probing limits its scalability to support efficient detection of network events among a large number of hosts [6].

This paper argues that it is hard to satisfy all the requirements of scalability, high end-to-end availability, and customized route selection in the existing global routing infrastructure. Instead, our goal is to provide highly-available communications between only a subset of the nodes which form a routing overlay. This is sufficient because in practice, not all pairs of nodes in the Internet need fast recovery mechanism.

Our approach is to integrate routing overlays into the existing routing infrastructure. We call our system UFO (Underlay Fused with Overlays) to emphasize the cross-layer nature of our design. Our two-layer routing architecture preserves the benefits of overlay routing, including high end-to-end availability, and customized route selection for applications. In addition, UFO provides the abstraction of a subscription service for network events occurring along the underlying paths between the overlay nodes. Explicit cross-layer notification about changing network conditions improves the efficiency of reactive routing at the overlay layer without compromising scalability, since notification messages are propagated only to the participating overlay nodes, using a lightweight multicast mechanism.

In this paper, we focus on how to design a scalable notification system by answering the question of who to notify about what kind of events. We design an efficient notification subscription mechanism, which is similar to joining a multicast tree. When network failure or excessive congestion happens, a notification message is multicasted only to the overlay nodes affected by the failure, which are the overlay nodes included in the subtree rooted at the offending underlay link. After receiving the notification, the overlay performs its own reactive routing, and lazily re-registers the link later.

In practice, we envision ISPs could view the two-layer design as an extension to their existing routing infrastructure, and offer highly-available communication services to their customers. Although the proposed scheme requires additional router support, we believe that these changes to routers are relatively modest and, further, that ISPs have sufficient incentive to augment their routers to provide this support. Many ISPs already run overlay nodes of their own, for VoIP and IPTV services. Providing explicit feedback about the performance of overlay links allows ISPs to offer better service to their customers, giving them a competitive edge over non-participating ISPs.

The remainder of the paper is structured as follows: we begin with Section 2 which presents the design of how the IP routers generate efficient notification messages when overlay links fail or become highly congested, to support scalable monitoring of the overlay network. Discussion of related work follows in Section 3. Lastly, Section 4 concludes with a discussion of the economic incentives for deployment.

2. SCALABLE OVERLAY MONITORING

Routing overlays [6] typically perform two functions: monitoring to detect network failures, and recovery by rerouting over alternative paths. In this paper, we integrate the design of routing overlays into the routing infrastructure. As opposed to conventional routing overlays, we propose that a more efficient division of function is to provide (1) monitoring support at the IP layer, and (2) customized route selection and recovery using alternate paths at the overlay layer. Notification of changes in network conditions from IP layer is more efficient than monitoring at the overlay layer, because network equipment has faster ways to detect failures and trigger notifications. For example, failure notifications can be triggered by a "loss of light" signal in SONET, or lost "hello" packets in OSPF. By receiving notifications from the IP layer, routing overlays may decide to reroute and recover earlier than the IP layer.

One goal of our notification mechanism is to preserve the overlay link abstraction: overlays are deployed without knowledge of the IP topology, and the routers can send notifications to overlays about overlay links without exposing properties of individual IP layer links. To achieve this goal, we design the registration scheme for routers to store states about the overlay links that traverse them.

Our registration and notification design is scalable, because: first, the number of overlay nodes and links is only a small portion of all the nodes in the networks. Second, we use multicast based design to improve the scalability of our registration and notification scheme. Lastly, multiple overlays running on the same overlay node can share the registration and notification states and messages.

In this section, we present an efficient and scalable notification system for the overlays. First, we describe how overlays register and store states at routers to get explicit notifications in response to various network events. Next, we illustrate how notification messages are multicasted only to the downstream overlay nodes affected by the offending link. Lastly, we explain how overlays can recover without explicit recovery notification from the IP layer.

2.1 Registration of Overlay Links

An overlay node registers a unidirectional overlay link by sending a registration message with the event types of interest. We can capitalize on ideas from multicast protocols to improve the efficiency of registration. In particular, we use a multicast tree rooted at the overlay destination to ensure that each registration packet traverses a link only once. The overlay sources join the multicast group to register for notifications about changes in reachability to the overlay destination. For example, in Figure 1, the four overlay sources join a multicast group for overlay links terminating at D. Suppose S1 joins the group first. Then, S1's registration message (i.e., join request) to D would traverse the path R3-R2-R1-D, and each node along the way would keep track of the outgoing link to forward packets sent to the multicast group. Later, when S2 joins the multicast group, router R3 grafts link R3-S2 into the multicast tree, but does not need to forward the registration request further. This process reduces the storage, bandwidth, and processing overhead for registration requests. After the four overlay nodes all finish registration, a multicast tree will be formed. As illustrated by arrows in Figure 1, the multicast tree is rooted at destination D and have four overlay nodes as leaves.

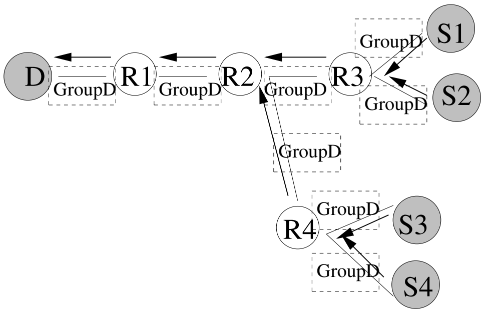

Figure 1: Multicast based registration

Many multicast protocols exist, with different complexity. Because of relatively simple requirements, we envision using Protocol Independent Multicast (PIM) [8]; PIM "sparse mode" is suitable since the number of overlay nodes is small, relative to the number of IP routers and links. We can also capitalize on existing data-plane support for multicast forwarding in IP routers. However, multicast forwarding typically requires a multicast group address. We could conceivably assign a multicast group address to each overlay destination, at the expense of administrative overhead and consuming a large part of the multicast address space. Instead, we envision that the data plane of the routers would treat registration packets (destined to D) as implicit multicast join messages. Similarly, the router could treat notification messages as implicit multicast packets destined to the downstream members of the multicast group.

Overlay nodes join the multicast groups, so that routers can scalably disseminate notification messages upon failure. In a unicast design, a fully-connected overlay with n nodes has n^2 overlay links, each requiring registration state on one or more underlay links. In the multicast design, the routers need only to participate in at most n multicast groups. In fact, multiple overlay destinations (participating in different overlay networks) could conceivably *share* a single multicast group, if they were using the same IP address. Suppose a second overlay node E runs on the same overlay node as D. Then, notification messages could be distributed over a single multicast tree rooted at D. The number of multicast groups would be limited to the number of participating overlay sites.

Registration messages are stored as soft state: the overlay nodes periodically refresh the registration and the routers discard stale registrations. Each registration packet carries a version number or timestamp that the routers store and include in notification packets, so overlay nodes can safely ignore outdated notification packets.

When processing a registration message, a router verifies that the two overlay nodes are allowed to register an overlay link. To prevent denial-of-service attacks and unauthorized use of the service, the routers must verify that the registration packet is authorized. Verification may involve consulting a separate server or a locally cached list of valid participating nodes. The ISP could also prevent source-address spoofing by network ingress filtering. Another option for authentication is to have registration packets carry a capability. In this way, access to the registration and notification service will be efficiently controlled by light-weight authentication cookies, such as those found in L2TPv3 [9].

2.2 Notification of Network Events

According to the various requirements of applications running on top of routing overlays, the IP network should provide explicit notification about different kinds of events that affect the performance of overlay links (e.g., failures, congestion). Different types

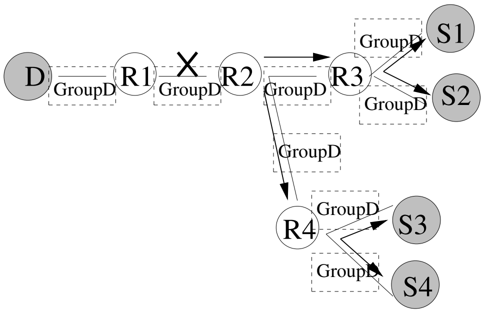

Figure 2: Multicast based notification

of overlays may require different types of notifications. For example, an overlay designed for video might be more concerned with jitter than one designed for bulk file transfer.

The router can notify registered overlay nodes about various kinds of network events that would affect the performance of an overlay link, including physical failure of routers or links, heavy congestion, or lost reachability due to policy changes or session failures. For example, for failure notification, the notification packet contains the failed overlay links (with the IP addresses of the source and destination overlays of this link). To reduce the overhead of the notification system, we are not interested in providing the overlays with complicated statistics which would take a long time to compute. Instead, we focus on notifying overlays about network events and statistics computed from passive measurement about the performance of an overlay link.

Only the router that is directly incident to the offending underlay link sends a notification. Upon detecting the event, the router has to determine the overlay groups affected by the failure. In Figure 2, suppose the link R1-R2 experiences some fault. Since group D uses the left interface of R2 (which is affected by the offending link) to reach destination D, R2 must send notification to group D.

As an optimization, the affected multicast group can be computed in advance when the registration packets perform lookups in the routing table during registration. In this example, since the next-hop interface of destination D is the left interface, R2 can associate group D with the left interface by storing state of group D at the left interface. Storing per-interface state at each router helps to quickly identify the right multicast group to notify.

For each affected multicast group, the router determines how to disseminate notification message within the group. The affected overlay nodes are the ones in the subtree rooted at the router that detected the network event. In this example, overlay nodes S1 to S4 are all in the subtree rooted at R2, and therefore can not reach destination D because link R1-R2 fails. R4 multicasts a notification packet to all *downstream* members in the multicast tree. To reduce the overhead of notification, only one copy of the notification packet would traverse each downstream link, as shown by the arrows on links R2-R3 and R2-R4 in Figure 2.

2.3 Lazy Re-registration of Overlay Links

After learning about a problem with an overlay link, the overlay can quickly reroute (at the overlay level) to circumvent the problem. Yet, an important question remains about how the overlay learns that the overlay link has recovered from failure. One seemingly natural approach would be for the router to send another notification message when the overlay link has recovered. However, determining that the underlying *path* (the overlay link between two overlay nodes) has recovered is difficult, since no one router has sufficient visibility into the path-level performance. For example, even if a failed link recovers, the incident router may not know that all of the other routers have converged to a new, stable path using that link. In addition, some of the routers on the new path may not have any registration state for the overlay link, if they were not on the old path that received the registration packet. As such, in our design, the routers do *not* send recovery messages to the overlays.

Fortunately, fast recovery of overlay links is not as important as fast notification of reachability or performance problems. Overlay networks typically have a rich (logical) topology, with numerous alternate paths to circumvent a problematic overlay link. As such, we rely on the overlay nodes to re-register the overlay link at some time in the future. The re-registration process is the same as the registration we discussed before. The overlay node could be conservative and wait for several seconds (or minutes) for the underlying path to reconverge before attempting to re-register the overlay link. This keeps the design simple without compromising the high availability of the overlay routing layer.

3. RELATED WORK

Our work is motivated by results showing the benefits of route deflection. Detour recognized the benefits in redirecting traffic along overlay paths [10, 11]. Resilient Overlay Networks (RON) [6] subsequently built a system based on Detour to improve availability by quickly detecting paths with high latency or packet loss and rerouting around them. However, routing overlays are usually limited in scale because of inefficient monitoring of overlay link quality. In particular, since the performance of reactive routing depends on whether changes in network conditions are discovered in an accurate and timely fashion, routing overlays usually probe aggressively, which leads to unavoidable tradeoffs between probe overhead and reaction time. Moreover, since these probes cannot easily differentiate performance degradation caused between link failures and transient congestion, the overlay nodes must wait for several lost probes before rerouting the traffic in order to avoid overreacting. By providing IP layer notification for efficient monitoring of overlay link quality, our work circumvents these problems.

Jannotti proposes path reflection to reduce overlay forwarding inefficiency by allowing end hosts to request short-circuit packet routing and duplication in nearby routers [12]. In our proposal, overlays are deployed as part of the infrastructure at routers. Since the overlays are deployed inside the network (e.g., at PoPs) instead of the edge of network, then reflection points for forwarding overlay traffic will be implicitly pushed inside the network. Nakao and Chen propose optimizations to reduce overlay measurement overhead by probing only along some most-disjoint underlay paths, with inference or knowledge of the underlay network topology [13, 14]. Although measurement overhead is reduced by selectively probing a subset of the overlay links, the amount of probing overhead to monitor an individual overlay link is not reduced. In comparison, our proposal pushes this monitoring functionality down to routers, and explicit notification improves the monitoring scalability by obviating the need for probing on each overlay link.

The importance of reliability in communication networks has long been recognized, and there is a large body of recovery techniques at different network layers [15]. At the link layer, SONET recovery is quite fast. However, since it uses shared or dedicated protection rings to provide backup capacity, the cost of resource reservation for both the primary and the backup path is very high. MPLS offers both restoration and protection techniques. MPLS restoration techniques have relatively slow response time, which varies from a few milliseconds to hundreds of milliseconds, which may not be able to meet the requirements of real-time applications.

MPLS protection techniques could reroute traffic upon failure using precomputed and signaled backup paths, also at the cost of resouce allocation for backup capacity. Our registration scheme is similar in spirit to the signaling mechanism in MPLS protection, but more efficient than MPLS in that our registration and notification is designed based on multicast.

At the IP layer, most prior work on improving the Border Gateway Protocol (BGP) reliability addresses control-plane issues, such as reducing convergence time and the number of routing messages [16, 17, 18]. However, BGP convergence faces fundamental trade-offs between the rate-limiting timers to reduce the number of routing updates, and the overhead for routers to process update messages. Recently, several proposals focus on reducing data-plane disruption. REIN [19] uses interdomain paths to protect interdomain links, can only be used to protect few critical interdomain links, because of its reliance on policy negotiation between different ASes. R-BGP [20] works by pre-computing a few strategically chosen failover paths. [21] proposes to use failure carrying packets for faster notification of link failures. Nevertheless, both proposals can only respond to routing failures, since they do not support the monitoring of network performance. Our work can be easily extended for notification of excessive network congestion. [22] provides router support for end-systems to explicitly select non-shortest path routes, but the proposed approach can not be easily extended to the interdomain context. Instead, our work focuses on providing high availability for a few overlay routes.

4. CONCLUSION AND DEPLOYMENT

Routing faces a tension between high availability and scalability. The paper proposes a layered routing architecture that achieves the best of both worlds by explicit notification about changes in network conditions from the IP layer, and therefore benefits scalable monitoring of overlay link quality. This paper has presented an initial design for UFO. As the first routing architecture that embraces routing as inherently multi-layer, we believe that it will offer important insights about how to design routing systems that are both reactive and scalable.

As for deployment, routers can provide line-card support for forwarding packets on overlay links and expose a standard interface to allow a separate control plane (either running directly on the routers or on a remote server) to install forwarding table entries directly on the routers. This improves the efficiency of packet forwarding in the overlay networks: specifically, it not only speeds up overlay packet encapsulation and decapsulation but also reduces the amount of traffic going both inbound and outbound to reach the overlay servers, which are traditionally located at the network edge. Moreover, ISPs can upgrade only a small fraction of their routers to provide routing overlay capabilities.

An alternative deployment option is for ISPs to host third party overlay networks, offering them extra support for packet forwarding and explicit notification, as an extension of their service hosting business. If routing overlays are run by a third-party, it will bring additional issues of accountability and security, as discussed below.

Accountability between the overlay and the ISP is based on whether the notification correctly characterizes the performance about overlay link conditions or not. Multiple ASes are jointly accountable and responsible for the performance of cross-domain overlay links.

As for security, overlays can set up monitors to keep track of whether the ISP lies by not sending notification during failures or excessive congestion. This can be achieved by using less-frequent probes to prevent the ISP from introducing a persistent bias in the notification. In particular, the overlay can deploy path-quality monitoring protocols [23] that reliably raise an alarm when the packet-loss rate and delay exceed a threshold. In this way, even if the ISP could lie sometimes by refusing to report network failures or congestion conditions, it would eventually get caught if it lies often. Fortunately, since the overlay only wants to verify whether the ISP lies or not, the overlay would not have to collect these measurements and react to them in real time, which introduces much less overhead than real-time measurement and reaction would.

5. REFERENCES

[1] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, “Delayed Internet routing convergence,” in *Proc. ACM SIGCOMM*, 2000.

[2] F. Wang, Z. M. Mao, J. Wang, and L. Gao, “A measurement study on the impact of routing events on end-to-end Internet path performance,” in *Proc. ACM SIGCOMM*, 2006.

[3] C. Boutremans, G. Iannaccone, and C. Diot, “Impact of link failures on VoIP performance,” in *Proc. NOSSDAV Workshop*, 2002.

[4] N. Kushman, S. Kandula, and D. Katabi, “Can you hear me now?! it must be BGP,” *ACM Computer Communication Review*, 2007.

[5] A. Markopoulou, F. Tobagi, and M. Karam, “Assessment of VoIP quality over Internet backbones,” in *Proc. IEEE INFOCOM*, 2002.

[6] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, “Resilient overlay networks,” in *Proc. Symposium on Operating System Principles*, 2001.

[7] N. Feamster, D. Andersen, H. Balakrishnan, and M. F. Kaashoek, “Measuring the effects of Internet path faults on reactive routing,” in *Proc. ACM SIGMETRICS*, 2003.

[8] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, “Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification.” RFC 2362, August 2006.

[9] Cisco, “Layer 2 tunnel protocol version 3.” `http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s23/l2tpv3.htm`.

[10] K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall, “Improving the reliability of Internet paths with one-hop source routing,” in *Proc. Operating System Design and Implementation*, 2004.

[11] A. Collins, “The Detour Framework for Packet Rerouting,” Master’s thesis, University of Washington, 1998.

[12] J. Jannotti, *Network Layer Suppport for Overlay Networks*. PhD thesis, Massachusetts Institute of Technology, 2002.

[13] Y. Chen, D. Bindel, H. Song, and R. H. Katz, “An algebraic approach to practical and scalable overlay network monitoring,” in *Proc. ACM SIGCOMM*, 2004.

[14] A. Nakao, L. Peterson, and A. Bavier, “A routing underlay for overlay networks,” in *Proc. ACM SIGCOMM*, August 2003.

[15] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, and MPLS*. Morgan Kaufmann, 2004.

[16] A. Bremler-Barr, U. Afek, and S. Schwarz, “Improved BGP convergence via ghost flushing,” in *Proc. IEEE INFOCOM*, 2003.

[17] L. Jiazeng and et al, “An approach to accelerate convergence for path vector protocol,” in *Proc. IEEE GLOBECOM*, 2002.

[18] D. Pei and et al, “Improving BGP convergence through consistency assertions,” in *Proc. IEEE INFOCOM*, 2002.

[19] H. Wang, Y. R. Yang, H. Liu, J. Wang, A. Gerber, and A. Greenberg, “Reliability as an Interdomain service,” in *Proc. ACM SIGCOMM*, 2007.

[20] N. Kushman, S. Kandula, D. Katabi, and B. M. Maggs, “R-BGP: Staying connected in a connected world,” in *Proc. Networked Systems Design and Implementation*, 2007.

[21] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica, “Achieving convergence-free routing using failure-carrying packets,” in *Proc. ACM SIGCOMM*, 2006.

[22] X. Yang and D. Wetherall, “Source selectable path diversity via routing deflections,” in *Proc. ACM SIGCOMM*, 2006.

[23] S. Goldberg, D. Xiao, E. Tromer, B. Barak, and J. Rexford, “Path-quality monitoring in the presence of adversaries,” in *Proc. ACM SIGMETRICS*, 2008.