OVERLAY NETWORKS

Madhumita De
Department of Computer Science and Engineering
University of Texas at Arlington
mxd0438@omega.uta.edu

Abstract

This paper gives an overview of overlay networks, which are isolated virtual networks deployed over existing networks. It also discusses the main applications of overlay networks and concentrates on Resilient Overlay Network (RON), discussing how it works, its architecture and the critical issues related to it. RON provides an architecture, whereby fault detection and recovery from path outages can be improved substantially. RON also demonstrates the benefits of moving some of the control over routing into the hands of end-systems.

Keywords: Overlay Networks, Resilient Overlay Networks, QoS, Mobile Computing

1 Introduction

An overlay network is an isolated virtual network built on top of one or more existing networks. It is essentially a collection of overlay sockets, which are connected with an overlay protocol, such as Hypercube [6], Delaunay triangulation [6], etc. The overlay sockets have properties, which are stored as attributes associated with the overlay network. These attributes include the type of overlay protocol, and socket adapter, which are associated with an Overlay ID, which uniquely identifies an overlay network. Overlay Networks provide an alternative to changing an existing network layer

A new overlay network is created by generating a unique overlay ID string and by specifying the attributes of the overlay network that are associated with the overlay ID. To join an existing overlay network, a node must obtain the overlay ID and the attributes of the overlay network.

Attributes can be of two types- key attributes and configurable attributes. Key attributes are bound to an overlay ID and are determined when an overlay ID is created. Key attributes cannot be modified once the overlay ID has been created for an overlay network. An overlay socket participating in an overlay network can modify configurable attributes. All attributes have a name and a value, both of which are strings. The following table gives the names and values for key attributes that are expected in most cases:

ATTRIBUTE NAME	VALID VALUES
KeyAttributes	List of key attributes, separated by commas
Socket	HCAST2-0 (HyperCast)[6]
SocketAdapter	ScktAdptTCP (default: TCP2-0) ScktAdptUDP
Node	HC2-0 (hypercube) DT2-0 (Delaunay triangulation)(default)
NodeAdapter	NodeAdptUDPServer NodeAdptUDPMulticast
NodeAdptUDPMulticast.UDPMulticastAddress	Multicast IP address/port number

Table1: Key Attributes

The attribute, Socket contains the version number of the overlay socket while as SocketAdapter indicates the protocol used by the socket adapter. Node contains the acronym of the overlay protocol with version number, NodeAdapter indicates the protocol used by the socket adapter, and NodeAdptUDPMulticast.UDPMulticastAddress indicates the IP multicast address used by NodeAdptUDPMulticast. The last mentioned key attribute denotes an additional attribute called UDPMulticastAddress for the attribute NodeAdptUDPMulticast that describes a parameter of the latter as a subattribute.

There are a few things that must be noted while creating a new overlay network.

- All attributes that are essential for establishing overlay socket communication in an overlay network should be selected to be key attributes.
- All overlay sockets that participate in the same overlay network must have identical key attribute values. If two overlay sockets have the same set of key attributes, then they should be able to communicate in an overlay network.

Another way of looking at an overlay network is that it is composed of hosts, routers, and tunnels. Tunnels are paths in the base network, and links in the overlay network. Hence, each link can translate into several hops on the underlying network. The characteristics of a link, such as latency, bandwidth, and losses are the aggregate of the underlying network links over which it travels [7]. Hosts are packet sources or sinks, and routers are packet transits, as in conventional networks. Individual components (routers or hosts) can participate in more than one overlay at a time or in multiple ways (router, host) in a single overlay.

The popularity of overlay networks lies in that without changing the existing network layer, its properties can be changed and several different applications- mobility [12,17], routing [5], quality of service [1,4] addressing (6bone), security [11,14], and multicast [8] make use of overlay networks.

The two primary uses of overlay networks are containment, and provisioning [10]. Containment refers to the ability of an overlay network in restricting the visibility of its contents. This is implemented by the use of tunneling, which encapsulates the packets of a new protocol.

Containment was one of the first uses of overlays in the early 1980's, and motivated their re-emergence in the early 1990's for the M-Bone and later 6-Bone [10]. The M-Bone, now obsolete, was used for multicasting and the 6-Bone, which is IPv6 on IPv4 and is the main current deployment of the latter, addresses addressing problems in IPv4 by providing more address space than the underlying network. Provisioning uses reservation of components and capacity along tunnels to provide service guarantees to the overlay network. Here, worth mentioning is the X-Bone, which discovers, configures, and monitors network resources to create overlays over existing IP networks [10].

In this paper, the main concentration will be on Resilient Overlay Networks (RON), which is a project undertaken by the MIT Laboratory for Computer Science on overlay networks to provide fault tolerance. Section 2 of the paper is on related works; sections 3 and 4 look at the working and architecture of RON, followed by some results in section 4. Section 5 is the conclusion of the paper.

2 Related Work

The idea of overlay networks is old; in fact, the Internet itself is an example of an overlay deployed over the telephone network. However, [1] claims that RON is the first wide-area network overlay system that can detect and recover from path outages and periods of degraded performance within several seconds. Few overlay networks have been designed for efficient fault detection and recovery, although some have been designed for better end-to-end performance. While RON shares with the Detour framework [18] the idea of routing via other nodes, it differs from Detour in three significant ways. First, RON seeks to prevent disruptions in end-to-end communication in the face of Internet failures. RON takes advantage of underlying Internet path redundancy in a few seconds, reacting responsively to path outages and performance failures.

Second, RON is designed as an application-controlled routing overlay because each RON is closely tied to the application using it; RON readily integrates application-specific path metrics and path selection policies. Third, experimental results from a real-world deployment of a RON to demonstrate fast recovery from failure and improved latency and loss-rates even over short time-scales are provided. An alternative design to RON would be to use a generic overlay infrastructure like the X-Bone and port a standard network routing protocol (like OSPF or RIP) with low timer values. However, this by itself will not improve the resilience of Internet communications for two reasons [1]. First, a reliable and low-overhead outage detection module is required to distinguish between packet losses caused by congestion or error-prone links from legitimate problems with a path. Second, generic network-level routing protocols do not utilize application-specific definitions of faults. Various Content Delivery Networks (CDNs) use overlay techniques and caching to improve the performance of content delivery for specific applications such as HTTP and streaming video. The functionality provided by RON may ease future CDN development by providing some routing components required by these services [1].

3 Resilient Overlay Network

The resilient overlay network (RON) is an architecture that allows distributed Internet applications to detect and recover from path outages and periods of degraded performance within several seconds, improving over today's wide-area routing protocols that take at least several minutes to recover [1]. The high recovery time in the case of wide-area protocols is due to the use of the Border Gateway Protocol (BGP-4) running at the border routers between autonomous systems that filters out the information shared with other autonomous systems and providers. Filtering of the information leads to several topological details being hidden, little information being available about traffic conditions, and damping of routing updates, which results in BGP's fault recovery mechanism taking tens of minutes to work.

RON is an application-level routing and packet-forwarding service that gives end-hosts and applications the ability to take advantage of network paths that traditional Internet routing cannot make use of [1]. At a glance, RON nodes examine the condition of the Internet between themselves and the other nodes, and based upon how the network looks, decide if they should let packets flow directly to other nodes or if they should send them indirectly via other RON nodes. This lets RON detect and react to path failures within several seconds instead of several minutes.

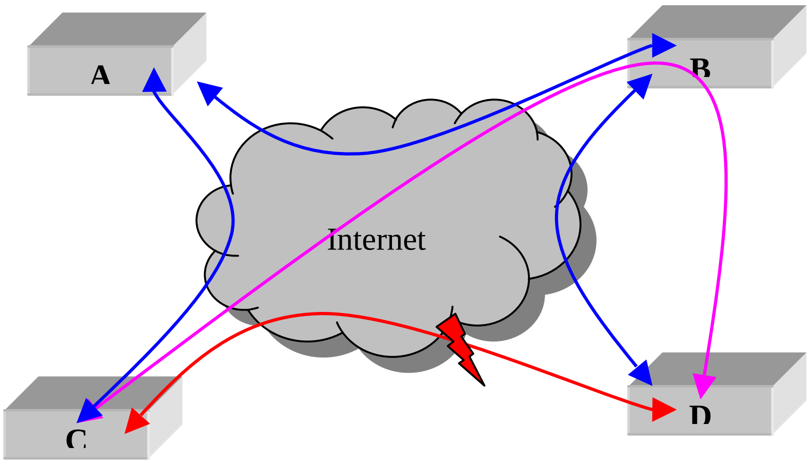

Figure 1: Routing around Internet Failures

The figure above represents the main goal of RON, which is to enable a group of nodes to communicate with each other in the face of problems with the underlying Internet paths connecting them. Here, RON nodes C and D communicate with each other via another RON node B so as to route around an Internet failure in the direct link between C and D.

RON nodes exchange information about quality of paths among themselves via a routing protocol and build forwarding tables based on a variety of path metrics, including latency, packet loss rate and available throughput [1]. The RON nodes obtain the path metrics by using a combination of active probing and passive observations of ongoing data transfers.

The second goal of RON is to integrate routing and path selection with distributed applications more tightly than is traditionally done. This integration includes the ability to consult application-specific metrics in selecting paths, and the ability to incorporate application–specific notions of what network condition constitute a “fault” [1]. When a RON node receives a packet intended for another RON node, it looks for the destination in an application-specific forwarding table, encapsulates the packet in a RON packets and sends it to the next RON node. In this way, the packet traverses the overlay via a series of RON nodes until it reaches the destination. On the other hand, BGP primarily uses a hop-counting mechanism to determine routes and it produces a single best route for forwarding packets. This may result in sub optimal routing.

4 Architecture

Each program that communicates with the RON software on a node is a RON client. A RON client interacts with RON across an API called a *conduit*, which the client uses to send and receive the packets [1].

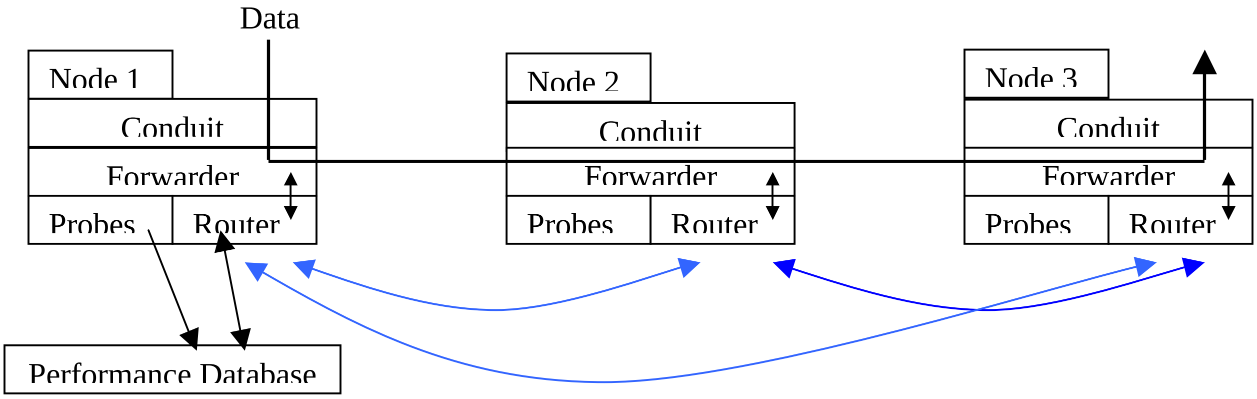

Figure 2: The RON system architecture

Data enters the RON from the RON client via the *conduit*at an *entry node*, which classifies the data packet to determine the type of path it should use (example, low latency, high throughput, etc.). The *entry node* determines the path from its topology table, for which the RON forwarder consults with its router. Then the packet is encapsulated into a RON header, tagged with some information that simplifies the forwarding by downstream RON nodes and it is forwarded on. Each subsequent RON node simply determines the next forwarding hop based on the destination address and the tag. The final RON node that delivers the packet to the RON application is called the *exit node*. When the packet reaches the RON *exit node*, the forwarder there hands it to the appropriate output conduit, which passes the data to the client. While selecting paths, RON monitors the quality of its virtual links using active probing and making passive measurements as mentioned before. RON nodes use a link-state routing protocol to disseminate the topology and virtual-link quality of the overlay network.

5 Results [1]

Results from two sets of measurements of a working RON deployed at sites across the Internet demonstrate the benefits of the RON architecture. For example, over a 64-hour sampling period in March 2001 across a twelve-node RON, there were 32 significant outages, each lasting over thirty minutes, over the 132 measured paths. RON's routing mechanism was able to detect, recover, and route around *all* of them, in less than twenty seconds on average, showing that its methods for fault detection and recovery work well at discovering alternate paths in the Internet. Furthermore, RON was able to improve the loss rate, latency, or throughput perceived by data transfers; for example, about 5% of the transfers doubled their TCP throughput and 5% of our transfers saw their loss probability reduced by 0.05. The loss rate between MIT and ArosNet ranged up to 30%, but RON was able to bring it down to well below 10% by routing data through Utah. Again, in the case of latency, in 92% of the samples, the latency of the packets sent over a RON-like path was better than the Internet latency. The average latency over the measurement period dropped from 97ms. to 68ms. At most one intermediate RON node was found to be sufficient while forwarding packets to overcome faults and improve performance in most cases.

6 Conclusion

An overlay network undoubtedly has its advantage because of which it finds application in several different fields. The advantage lies in it being easy to configure, not requiring router support, and having an application-specific architecture. It also provides an excellent test-bed for designing and testing network protocols. However, none of the papers mention the drawbacks of overlay networks- one more level of indirection, and the overlay nodes not always being located at the best locations, which may affect the performance of any overlay network adversely. The Resilient Overlay Network suffers from its own drawbacks. RON creates the possibility of misuse or violation of BGP transit policies [1]. The potential criticism for RON is that while they may perform well in limited settings, they may not when they start to be deployed on a large-scale [1]. Moreover, Network Address Translators (NATs) pose one more problem for RON- when both hosts are behind NATs, they may not be able to communicate directly, which will be interpreted by RON as an outage. RON would try to route around it thereby establishing communication between the two but possibly resulting in sub optimal routing. Inspite of these drawbacks, the research results prove that RON is a good platform on a variety of resilient distributed Internet applications may be developed.

References:

[1] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. In *Proc. of the 18th. ACM Symposium on Operating Systems Principles(SOSP), October 2001.*

[2] John Janotti. Network Layer Support for Overlay Networks. IEEE OPENARCH 2002.

[3] Sherlia Shi, Jonathan Turner. Placing Servers in Overlay Networks. www.arl.wustl.edu/Publications/2000-04/spects02ss.pdf

[4] Lakshminarayanan Subramanian, Ion Stoica, Hari Balakrishnan, and Randy Katz. OverQoS: Offering QoS using Overlays. HotNets-I: 1st HotNets Workshop, Princeton, NJ (2002).

[5] Ben Y. Zhao, Yitao Duan, Ling Huang. Brocade: Landmark Routing on Overlay Networks. First International Workshop on Peer-to-Peer Systems (IPTPS 2002).

[6] http://www.cs.virginia.edu/~mngroup/hypercast/designdoc/PDF/

[7] Yair Amir, Baruch Awerbuch, Claudiu Danilov, and Jonathan Stanton. Global Flow Control for Wide-Area Overlay Networks: A Cost Benefit Approach. IEEE OPENARCH 2002.

[8] Sherlia Y. Shia and Jonathan Turner. Multicast Routing and Bandwidth Dimensioning in Overlay Networks. JSAC/IEEE 2002.

[9] John Byers, Jeffrey Considine, Michael Mitzenmacher, and Stanislav Rost. Informed Content Delivery Across Adaptive Overlay Networks. NSF Career awards. www.cs.bu.edu/techreports/pdf/ 2002-007-informed-overlay-delivery.pdf.

[10] Joe Touch. Dynamic Internet Overlay Deployment and Management Using the X-bone. Proc. ICNP 2000, Osaka, Japan. www.isi.edu/touch/pubs/comnet2001/, www.isi.edu/x-bone (White paper)

[11] Angelos D. Keromytis, Vishal Mishra, and Dan Rubenstein. SOS: Secure Overlay Services. SIGCOMM'02.

[12] M. Stemm and R.H. Katz. Vertical Handoffs in Wireless Overlay Networks. ACM Mobile Networking (MONET), Special Issue on Mobile Networking in the Internet, 1998.

[13] Tzi-cker Chiueh. Resource Virtualization Techniques for Wide-Area Overlay Networks. www.ecsl.cs.sunysb.edu/von/

[14] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, and Dan S. Wallach. Secure Routing for structured peer-to-peer overlay networks research.microsoft.com/users/mcastro/security.pdf.

[15] Stephen Cogdon and Ian Wakeman. How Bad Are Overlay Networks? www.ee.ucl.ac.uk/lcs/papers2002/LCS098.pdf

[16] Minseok Kwon and Sonia Fahmy. Topology-Aware Overlay Networks for Group Communication. NOSSDAV'02, ACM.

[17] Fan Du, Lionel M. Ni, and Abdol-Hossein Esfahanian. HOPOVER: A New Handover Protocol for Overlay Networks. IEEE, 2002.

[18] A. Collins. The Detour Framework for Packet Rerouting. Master's thesis, University of Washington, Oct. 1998.