HIDRA: HIERARCHICAL INTER-DOMAIN ROUTING ARCHITECTURE

Bryan Clevenger
Computer Science Department
California Polytechnic State Univ.
San Luis Obispo, CA, USA
email: bcleveng@calpoly.edu

Daniel Nelson
Research done while student at:
California Polytechnic State Univ.
San Luis Obispo, CA, USA
email: daninels@cisco.com

John M. Bellardo
Computer Science Department
California Polytechnic State Univ.
San Luis Obispo, CA, USA
email: bellardo@calpoly.edu

ABSTRACT
The size of the Internet's forwarding table is growing rapidly, generating concerns about the ability for high performance routing equipment to economically keep pace. The primary contributors to this growth are end site multihoming, traffic engineering, and in the foreseeable future, IPv6 deployment.

This paper presents HIDRA, a hierarchal network architecture designed to reduce both the immediate size of the Internet's forwarding table as well as its growth rate while maximizing compatibility with the existing Internet architecture. This includes the ability to use existing high performance routers, existing routing protocols, and existing number allocation policies.

HIDRA is prototyped on a small network testbed and shown to work in a limited set of circumstances, including normal network operation, link failures, traffic engineering, and mixed "legacy" Internet and HIDRA topologies. The potential reduction of the Internet's forwarding table is also analyzed.

KEY WORDS
Communication Networks, Routing, Internet, Network Architecture, HIDRA

1 Introduction

The Internet's forwarding table has roughly 300,000 [9] entries, and is growing at a rate faster than the rate of growth of the Internet [1]. This places an disproportionate demand on the medium and high speed routers that comprise the core of the Internet. In order to forward traffic at line speed, which can easily reach into the hundreds of gigabits per second with just a relatively small number of 10-gbit interfaces, these routers use customized hardware [21] to store and perform lookups in the forwarding table. This hardware is expensive, partially due to its small production scale when compared to commodity computer components such as DRAM. In addition, most routers don't provide a field upgrade path solely for the forwarding table capacity, forcing network operators to either replace entire line cards or suffer degraded forwarding performance when the capacity is exceeded.

The primary contributors to the growth of the default free zone's (DFZ) forwarding table have been identified as end site multihoming and traffic engineering [1]. In addition, IPv6 deployment has been predicted to have a noticeable negative impact on the number of DFZ routes [1]. This trend exists despite the current methods of policy control on the number of multihomed sites and best-practice guidance on prefix deaggregation. The problem has reached the point that some sites will not accept a DFZ prefix longer than /24, creating connectivity problems [14].

This paper proposes HIDRA, a Hierarchical Inter-Domain Routing Architecture, as an approach to reducing the size of the DFZ forwarding table. HIDRA divides the Internet into a two-level hierarchy, using IPv4 encapsulation to forward packets between different levels of the hierarchy. This paper also describes the initial HIDRA prototype implementation and testbed. It shows results confirming basic operation, including successfully handling fail-over scenarios with a multihomed site. It also discusses the potential impact that HIDRA can have on the DFZ and its ability to operate in a mixed environment with "legacy" networks.

HIDRA's overriding design constraint is deployability. The incomplete transition to IPv6 has shown [6] that the privatized Internet will not effectively adopt an incompatible protocol or a change that requires a substantial investment without an extremely compelling business case. As such, HIDRA goes to great lengths to maximize compatibility with existing protocols such as IPv4 and BGP, current network hardware, number resource policy, and existing business constraints.

The remainder of this paper is organized as follows. Section 2 introduces general background concepts. Section 3 discusses related work. Section 4 presents HIDRA. Section 5 presents our HIDRA prototype software, testbed, and presents proof-of-concept results. Section 6 details the future directions of HIDRA, and section 7 concludes the paper.

2 Background

Forwarding table expansion has been looked at since the early years of the Internet. Kleinrock and Kamoun recognized the potential expansion of networks to "even possibly thousands of nodes" in 1977 [10], and designed a hierarchical routing system to address the expansion gracefully. Since then, many researchers have investigated hierarchical routing and other measures designed to reduce table growth [20, 4, 18, 16, 2]. In hierarchal routing the entire network topology is divided up into levels. Nodes within any one specific level only need fine granularity routes to each other, and coarse routes to all other locations in the hierarchy. It is these coarse routes that decrease the size of the forwarding table.

Of particular note are Krioukov's overviews of

large-scale routing[11, 12]. The most important point presented by Krioukov is his description of a Locator-Identifier (L-I) split in addressing. An L-I split enables aggregation of identifiers into locations for global distribution. Nodes in the current Internet architecture use IP addresses to serve as both the locator and identifier. Thus, a different network architecture is required to take advantage of the benefits of the L-I split.

There are two broad categories of routing protocols seen in hierarchal proposals. The first is proactive routing. In proactive routing all routes are distributed *before* they are used, and updated or withdrawn as the underlying network topology changes. Border Gateway Protocol (BGP), the protocol currently used on the Internet, is proactive.

The second category is reactive protocols. These delay looking up the routing information until the first time a route is used. A side effect of this technique is that the first packet sent along any one route has substantially higher latency. To avoid the latency in future packets the routing information is cached. Proactive protocols have no extra first packet latency, however they require constant background management traffic to update the routes, may have problems with extremely large tables, and may take a long time to converge before the network is operational.

This work differentiates between the *forwarding information base*, or FIB, and the *routing information base*, or RIB. The FIB is size constrained and implemented in expensive hardware [21]. The RIB stores all the routes learned for every prefix. Only the best route from the RIB get installed into the FIB. As explained in [8], most modern routers have the capability to selectively prevent a RIB route from being installed in the FIB.

3 Related Work

There are a number of current proposals for reducing the number of DFZ routes, the majority of which involve explicitly implementing a Location-Identifier split. Many of these proposals have components in common with HIDRA. For instance, a number of protocols use encapsulation and proactive routing. However we feel HIDRA is unique in its pragmatism of providing backwards compatibility with existing equipment and other non-technical aspects of operating interdomain networks. This section reviews some of the more popular proposals and those proposals that are the most similar to HIDRA. Due to space constraints is does not review all proposals.

The design of ViAggre [8] is motivated by the same observation that major architectural changes are unlikely without an incremental deployment strategy that doesn't require upgrading router hardware of software. It leverages the difference between the FIB and the RIB. HIDRA uses an IP encapsulation scheme, as opposed to MPLS tunneling, that provides global scalability benefits when used by multiple peer sites.

Shim6 [15] is an end-host protocol stack modification that provides both load-sharing and failover capabilities to multihomed sites without the requirement for provider independent addresses. Since multihomed sites are a major contributer to the DFZ, this can dramatically reduce the FIB size. Unlike HIDRA, Shim6 requires IPv6 deployment, explicit support in all communication endpoints, and only focuses on multihomed sites.

NIRA [22] describes a new comprehensive, policy based network architecture. It employs a hierarchical provider-rooted addressing scheme that can reduce the FIB size, however it includes many changes that will impede adoption. For instance, it uses new proactive and reactive protocols, a new representation for routes, and a different business model for provider compensation.

IPNL [7] uses a two-level routing hierarchy with IPv4 as L_0 and a new IPNL protocol at L_1. The identifier address is based on fully qualified domain names. NAT is used to transform packets to traverse L_0. Two key benefits are addressing v4 address exhaustion and the ability to renumber a site without much difficulty. HIDRA does not directly address exhaustion, but it does remove the major hurdle to issuing more sites *provider independent* addresses without increasing the size of the DFZ.

HLP [17] is a proposed routing protocol shown to have better scalability and performance characteristics than BGP. Internally it uses path-vector routing between hierarchies and link-state within. These techniques were intended to replace functionality found in BGP. In contrast, HIDRA uses encapsulation to explicitly forward packets along AS routes and, for compatibility reasons, uses BGP to exchange routes. These proposals can be complementary. A more efficient way of distributing routes can benefit HIDRA and using explicit encapsulation reduces the burden on the routing protocol.

TRRP [19] is a similar proposal in terms of both goals and implementation. They also propose encapsulating traffic in an IPv4 packet that is compatible with the preexisting router base, and adding encapsulation/decapsulation devices to the network edges. However, TRRP requires a reactive routing protocol, and uses DNS to create a mapping between hostname and destination address. While HIDRA can be expanded to incorporate a similar reactive scheme, it is not immediately required to realize benefits. Additionally, TRRP requires the use of an existing, preassigned IPv4 address as the endpoint of the tunnel. In contrast, HIDRA uses a new addressing scheme to quickly differentiate between different layers of the hierarchy, and derives endpoints directly from already existing number allocation policy.

LISP [5] is another proposal that uses encapsulation to deliver packets across the Internet. LISP defines the specific packet formats, including the use of UDP encapsulation and reactive lookup. It also provides behavioral constraints for the mapping protocol. LISP-ALT [5] is an instance of LISP that creates an overlay network using GRE tunnels with multi-protocol BGP peering sessions inside the tunnels. This overlay is used for the reactive lookup portion of LISP. Unlike HIDRA, LISP's reactive routing uses new, untested protocols. LISP does not support a purely proactive scheme.

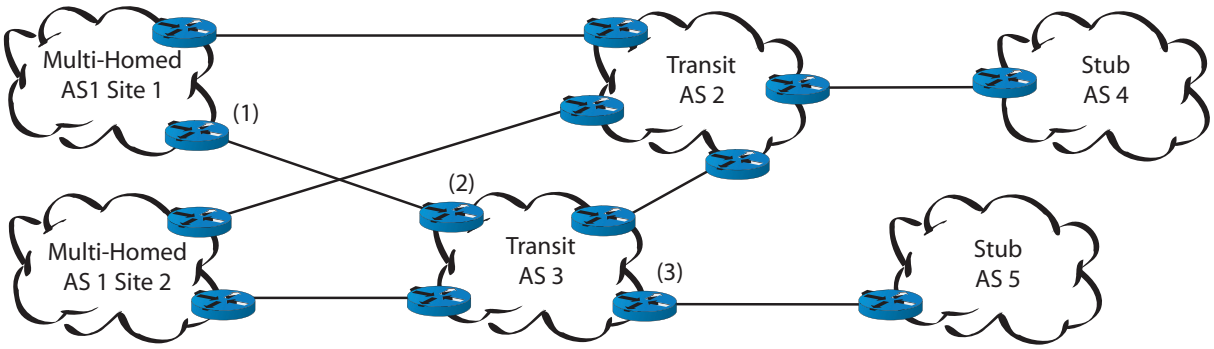


Figure 1. A multi-site multi-homed network. From the standpoint of AS1, encapsulation can occur at point (1) if the AS itself provides encapsulation, point (2) if its upstream provider encapsulates, or at the originating end-host if a reactive routing scheme is used. Decapsulation can occur either at point (1) for end-site decapsulation, or point (3) for ISP decapsulation. (3) provides for slower DFZ FIB growth and better failover characteristics.

4 HIDRA

HIDRA is a hierarchical network architecture. The top level of the hierarchy, level 0 or L_0, always uses IPv4 as the network layer. Using location-identifier parlance, the destination location address is in the L_0 header. Level 0 consists of all transit networks. A transit network is generally responsible for carrying traffic between disparate networks under different administrative control. End sites, whether or not they are multihomed, are not part of L_0.

The identifier address is found in L_1. When an L_1 packet is traversing L_0 it gets encapsulated with the appropriate L_0 header. Before traversing the end site network, the packet will be decapsulated and all subsequent forwarding decisions will be based on the L_1 header. The protocol for L_1 is unspecified, however the initial design and implementation assume it is either IPv4 or IPv6. IPIP encapsulation is used when IPv4 is the L_1 protocol.

IPv4 is chosen as the L_0 protocol to maximize compatibility with existing hardware. By making calculated use of standard IPv4 forwarding logic, all present routers are able to forward HIDRA traffic and carry both L_0 and L_1 routes without hardware modifications or software upgrades. Additionally, most existing routers can be active participants in HIDRA routing with small configuration changes and the inclusion of an external encapsulation/decapsulation device. This is an extremely important part of facilitating HIDRA adoption.

4.1 L_0 Addresses

The anticipated immediate use of HIDRA employs IPv4 as both the L_0 and L_1 protocol. Because most networks will carry both L_0 and L_1 traffic simultaneously, it is important to be able easily differentiate the packets even though both layers are using the same logical address space. To perform this classification on existing equipment, HIDRA sets aside a well-known /8 prefix to contain all the L_0 addresses. Using this technique it is possible to install a set of routes that explicitly treat all L_0 and non-L_0 packets separately.

The L_0 addresses are computed as a direct function of existing *autonomous system numbers* (ASNs). This helps to leverage existing number allocation policy and network topologies. Employing ASNs in this way is a natural extension of their current use. Presently each transit, multihomed, or single homed site with unique routing policy is assigned a single ASN. This number already logically corresponds to the L_0 location. That is, the network the communication end point is attached to. Defining a mapping between ASNs and L_0 addresses also enables the reuse of two key pieces of the Internet infrastructure: number resource allocation mechanism and policy, and the BGP routing protocol. A secondary benefit is the ability to project the future size of the DFZ FIB based on historical number consumption, as seen in Section 4.7.

The actual mapping used to create the L_0 address in HIDRA is to use the /8 prefix for the high-order 8 bits of the address, and set the low-order 24 bits to the low-order 24 bits of the ASN. This technique only uses the lower-order 24 bits of the 32-bit ASN. This is not a large limitation because current ASN being issued have all 8 high-order bits set to 0 [3]. Additionally, as discussed in section 4.7, the projected consumption rate of ASNs in HIDRA is such that it will take far in excess of 10,000 *years* before it is necessary to use any of the high-order 8 bits.

4.2 Encapsulation

Encapsulation is the act of placing an L_0 header on an L_1 packet. This is an expensive operation in both time and space. It requires performing a lookup on the L_1 destination to determine the corresponding L_0 destination. Such a mapping potentially requires a lookup table at least the size of the current DFZ FIB. Performing this expensive mapping close to the transmitting host is very important. First, it moves the incremental encapsulation burden of supporting additional devices from the core L_0 network equipment to edges of the network. It also provides for much better lookup cache locality, which in turn makes encapsulation more efficient. HIDRA uses at least one encapsulation device, and typically more to provide failover and load balancing, near the access links for each end site. Ideally the end hosts themselves will encapsulate the packets before transmission, removing the burden

from the network entirely. HIDRA also works well *without* this optimization, which is not explored in this paper.

It is unreasonable to expect all end hosts to participate in encapsulation. Upon initial deployment there will be almost no devices that have encapsulation software. Given the number of special purpose embedded hosts used today, it is also unreasonable to expect *every* "legacy" device in a network to become HIDRA aware. Therefore the network must support a transparent encapsulation service. This service will be initially provided by dedicated network hardware, then transitioned to border routers as their software and load permits. When encapsulation is done using an external dedicated device, that device should be topologically close to the border routers to minimize stretch.

Figure 1 illustrates the possible encapsulation points within a network. Large stub and multihomed sites are expected to provide their own encapsulation service, with smaller site leaving that responsibility to their upstream access provider.

4.3 Decapsulation

A packet is decapsulated as it traverses the L_0 – L_1 boundary. Decapsulation is a much faster operation than encapsulation because it only requires removing the outer-most header from the packet before forwarding it using standard techniques; there is no inherent requirement for a large, slow lookup operation. The only technical requirement placed on the decapsulation point is that it sits in both the L_0 and the L_1 networks. HIDRA takes advantage of this flexibility to minimize the number of routes in the L_0 DFZ.

Initially the decapsulation service will be provided by an external device, typically the same device that provides encapsulation. Later it will be transitioned to border routers as their software permits. Like encapsulation, when decapsulation is done using an external dedicated device, that device should be topologically close to the boarder routers to minimize stretch.

The obvious point of decapsulation is when a packet enters the destination site (either stub or multihomed). This is depicted as point (1) in figure 1. This is termed "end-site decapsulation." End-site decapsulation requires every AS to originate an additional route to the DFZ FIB, the route for the site's L_0. This will be a /32 route.

Point (3) in figure 1 is the immediate upstream provider's external gateway. Performing decapsulation here is termed "ISP decapsulation." ISP decapsulation has a number of advantages over end-site decapsulation. Most importantly, it enables traffic engineering for a multihomed site. The site can select the appropriate ingress link by advertising that *provider's* L_0 address as the decapsulation point. It also enables efficient multi-site networks. Since the decapsulated L_1 packet traverses the provider's network, it can take the most efficient path from the decapsulation point to any of the customer's sites.

ISP decapsulation further reduces the size of the L_0 DFZ, because the only entries necessary are for transit provider's decapsulation points. As shown in section 4.7, there are almost an order of magnitude fewer transit providers than single or multi-homed networks. Finally, the FIB burden of accepting new customers is placed on the upstream provider(s) that contract with the customer. Each customer will add one L_1 entry to the provider's FIB for each route the provider accepts from the customer. In contrast, the provider will only occupy a single slot in the L_0 forwarding table, regardless of the number of customer routes or the length of customer prefixes it accepts.

4.4 Routing

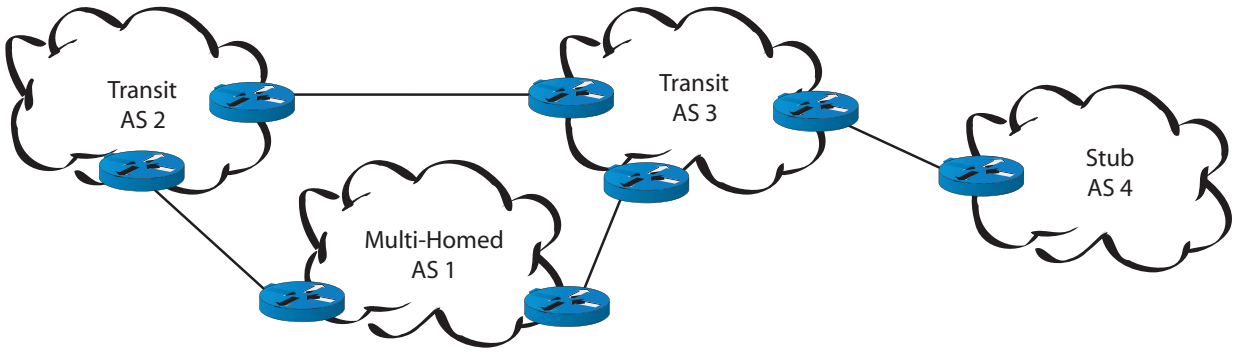
The current version of HIDRA uses a proactive routing protocol. The reactive protocol for HIDRA is ongoing work, as mentioned in section 6.

To maximize compatibility and interoperability with existing network infrastructure, HIDRA uses BGP as its proactive routing protocol. Unmodified BGP already contains all the information necessary to map a L_1 address into the corresponding L_0 address. Each L_1 route has its own entry in the BGP table, so the normal longest prefix lookup can be used to extract that entry. Instead of using the next-hop value from that entry, which is standard on today's Internet, the AS path attribute is used. The AS path attribute is an ordered list of all the ASNs a packet traverses while it follows the path to the destination. HIDRA is only concerned with the final ASN in the path. This will be the ASN of the destination site. This ASN is extracted from the AS path and transformed into the corresponding L_0 address as previously described.

Selecting BGP as the proactive routing protocol for HIDRA enables the reuse of all *existing* route advertisements on today's Internet. It further enables the reuse of network administrator's knowledge of and device support for manipulating BGP advertisements for purposes such as traffic engineering and primary/backup link designation.

In addition to using the existing advertisements for proactive lookups, all HIDRA enabled ISPs must advertise their L_0 route via BGP. These routes are originated by every device within the site that can perform decapsulation. These devices may be dedicated decapsulation boxes or decapsulation routers themselves. Either way, this system uses any-cast to replicate the decapsulation service within a site. The encapsulation devices also need to originate a default route that is not propagated outside the site. This route redirects unencapsulated L_1 traffic not destined for the current L_1 site to be encapsulated before the trip across L_0.

Routers in HIDRA are configured in a similar fashion to current routers. That is, they exchange full L_0 and L_1 routes, both internally and externally, via BGP. All these routes are present in every router's RIB. However all HIDRA aware routers are configured to prevent non-customer L_1 routes from entering the FIB. L_0 routes are easily identified by the well known prefix. Since all HIDRA routers have a complete RIB, it is possible to have a long chain of alternating HIDRA aware and legacy sites each with the information they need to successfully forward packets across the entire network. This permits



Legacy Internet

Peering Link	Routes Originated
AS1⇒AS2	**192.168.1.0 ⇝ AS1**
AS1⇒AS3	192.168.1.0 ⇝ AS1
AS2⇒AS1	**192.168.2.0 ⇝ AS2**
AS2⇒AS3	192.168.2.0 ⇝ AS2
AS3⇒AS1	**192.168.3.0 ⇝ AS3**
AS3⇒AS2	192.168.3.0 ⇝ AS3
AS3⇒AS4	192.168.3.0 ⇝ AS3
AS4⇒AS3	**192.168.4.0 ⇝ AS4** **192.168.44.0 ⇝ AS4**

HIDRA with End Host Decapsulation

Peering Link	Routes Originated
AS1⇒AS2	**ASN.0.0.1 ⇝ AS1** 192.168.1.0 ⇝ AS1
AS1⇒AS3	ASN.0.0.1 ⇝ AS1 192.168.1.0 ⇝ AS1
AS2⇒AS3	**ASN.0.0.2 ⇝ AS2** 192.168.2.0 ⇝ AS2
AS3⇒AS1	**ASN.0.0.3 ⇝ AS3** 192.168.3.0 ⇝ AS3
AS3⇒AS2	ASN.0.0.3 ⇝ AS3 192.168.3.0 ⇝ AS3
AS3⇒AS4	ASN.0.0.3 ⇝ AS3 192.168.3.0 ⇝ AS3
AS4⇒AS3	**ASN.0.0.4 ⇝ AS4** 192.160.4.0 ⇝ AS4 192.160.44.0 ⇝ AS4

HIDRA with ISP Decapsulation

Peering Link	Routes Originated
AS1⇒AS2	192.168.1.0 ⇝ †
AS1⇒AS3	192.168.1.0 ⇝ †
AS2⇒AS3	**ASN.0.0.2 ⇝ AS2** 192.168.1.0 ⇝ AS2
AS3⇒AS1	**ASN.0.0.3 ⇝ AS3** 192.168.4.0 ⇝ AS3 192.168.44.0 ⇝ AS3
AS3⇒AS2	ASN.0.0.3 ⇝ AS3 192.168.4.0 ⇝ AS3 192.168.44.0 ⇝ AS3
AS3⇒AS4	ASN.0.0.3 ⇝ AS3 192.168.1.0 ⇝ AS3
AS4⇒AS3	192.168.4.0 ⇝ † 192.168.44.0 ⇝ †

Figure 2. Example prefixes and ASNs (⇝) advertised in a small multi-homed network using legacy BGP, HIDRA with end site decapsulation, and HIDRA with ISP decapsulation. Routes in bold are the L_0 FIB. Duplicate L_0 routes are not in bold. All other routes are L_1. † – Route originated with a private ASN.

an ISP to deploy HIDRA without requiring a customer to reconfigure its end of the peering sessions.

Figure 2 depicts a small, four site network, and illustrates how routes are announced and propagated under the current Internet architecture, HIDRA with end-site decapsulation, and HIDRA with ISP decapsulation. In the legacy example, all routes are originated by their own AS and installed in the DFZ FIB, for a total of 5 entries. When HIDRA and end-site encapsulation is employed, each AS originates a single L_0 route and enough proactive routes to advertise their entire address space. This results in 4 entries in the DFZ FIB.

The final example in figure 2 shows HIDRA with ISP decapsulation. Each ISP originates a single L_0 route, for a total of 2 entries in the DFZ FIB. In addition, the stub and multihomed sites advertise their prefixes with a *private* ASN to their immediate upstream providers. The providers routers automatically remove the private ASN from the AS path before propagating the route, so it appears as though the route originates from the provider's network. This ensures the packet is addressed to the provider's L_0 decapsulation address and *not* the customer's private ASN.

4.5 Multi-homed Networks

Another benefit of using BGP to distribute proactive routes is robust support for multihomed sites. AS1 in figure 2 is a multihomed AS. When either one of its access links fail, the traffic will automatically be routed through the other link regardless of the decapsulation point. Both of these scenarios are illustrated in the next paragraphs.

Assume HIDRA is using end-site decapsulation. The multihomed site will be advertising its L_0 route to both its upstream providers, AS2 and AS3. AS3 will also hear the multihomed site's L_0 route from its peering session with AS2, however it will prefer the direct link between AS3 and AS1 due to the shorter AS path. AS4 will hear AS3's best route, which will be AS3–AS1. Further assume that a host in AS4 is communicating with a host in AS1. The packets get encapsulated at AS4's border gateway and then sent along the L_0 route to AS1. When the AS1–AS3 link fails it will be detected by BGP. AS3 will fall back on the next best L_0 route, AS1–AS2–AS3. This modified route will be propagated to AS4, and all L_0 packets will continue to reach AS1 using the updated route.

In the second scenario HIDRA is using ISP decapsulation. In this case AS2 is originating a proactive route to AS1's prefixes, as is AS3. These routes have AS2 and AS3, respectively, as the last AS in the AS path. AS4 will only hear AS3's advertisement since that route has a shorter AS path. In addition both AS2 and AS3 are advertising L_0 routes. Packets sent from a host in AS4 to a host in the multihomed site getS encapsulated with AS3's L_0 address, because that is the best proactive route. When the link between AS1 and AS3 fails, BGP will withdraw the *proactive* route originated by AS3. This causes AS3

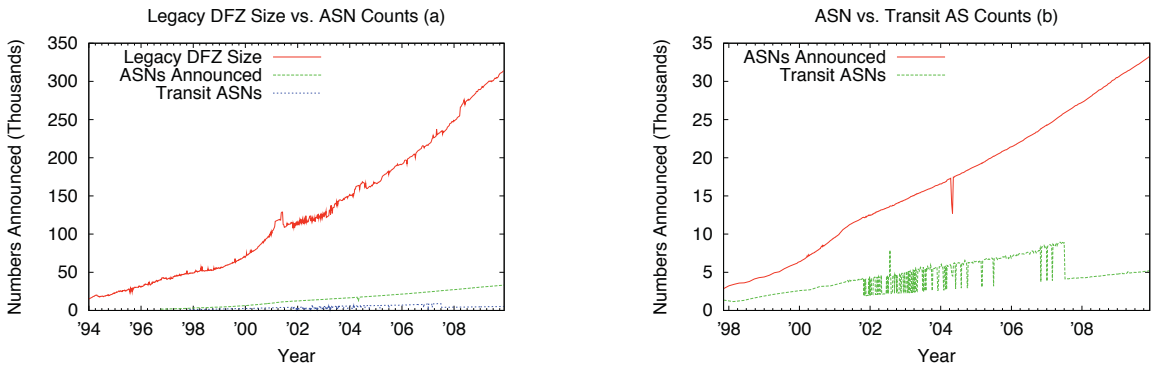


Figure 3. Historical size of the DFZ table. The "Legacy DFZ Size" shows the historical and current size of the DFZ table. The "ASNs Announced" line shows the number of unique ASNs visible on the public Internet. "Transit ASNs" is the number of unique ASes that appear in the middle of at least one BGP AS path. The former AS count approximates customer decapsulation and the later ISP decapsulation. Graph (b) focuses on the ASN data. All data is from the public potaroo [9] web site from the RouteViews [13] vantage point. For readability only every hundredth data point is plotted in the graph.

to advertise its next best proactive route, the route that AS2 is originating. BGP then updates AS4 with the new route, and all future encapsulations will use AS2's L_0 decapsulation address.

In both scenarios recovery automatically takes place on the same time scale and using the same mechanisms as recovery in today's Internet. This is true when using both end-site decapsulation and ISP decapsulation.

4.6 Load Balancing

An accomplishment of HIDRA is retaining as many existing network management practices as possible. Load balancing is an important example. This technique is still available within HIDRA, but requires ISP Decapsulation. Given a HIDRA network with ISP decapsulation, the end-site AS can balance incoming traffic by advertising different portions of its netblock to different upstream ISPs. The netblock split is under control of the end site, as are the MEDs, communities, AS prepending, and other BGP traffic engineering techniques.

4.7 Projected Impact on DFZ Table Size

A secondary benefit of using ASNs as the basis for the L_0 address is the large body of historical ASN utilization data that is available. Since 1998, [9] has been archiving enough data to project both the absolute size and growth of the L_0 table for legacy routing, HIDRA routing with end-site decapsulation, and HIDRA with ISP decapsulation. The two graphs in figure 3 show this data.

The data supports an immediate projected reduction from approximately 315,000 entries to 34,000 entries, equivalent to one order of magnitude. This assumes every existing stub and multihomed site performs its own decapsulation, which is a reasonable initial deployment scenario. Long term, stub and multihomed sites should migrate to ISP decapsulation, reducing the table size from 34,000 to approximately 5,000, another order of magnitude.

In addition to immediately reducing the DFZ FIB size, HIDRA also substantially changes the table's growth trend. The current table is growing either exponentially or super-linearly with a steep slope. Either way, the growth trend in ASN usage is linear with a much shallower slope. Additionally, the growth in transit ASNs is even flatter than total ASN growth. Under the assumption that new non-transit sites would *not* be issued ASNs after HIDRA implementation, the transit ASN trend is the one that will most closely predict future L_0 growth. Therefore widespread adoption of HIDRA can both immediately reduce DFZ size by at least one order of magnitude, and flatten out the expansion trend so much that the L_0 table will not reach the size of today's table again for at least 50 years, if not significantly longer.

5 Prototype

HIDRA has been prototyped in a laboratory setting. The prototype provides early operational experience and increases the confidence in the technical aspects of the architecture. The prototype consists of two parts. The first part is the software stack necessary to encapsulate packets, decapsulate packets, and use proactive routing to lookup a destination ASN based on the destination address in the packet header. The second piece is an actual network that uses HIDRA to route packets. The following sections describe both aspects of the prototype, and discuss some of the experiments that have been performed.

5.1 Software

HIDRA was implemented and tested using generic Linux computers to provide encapsulation and decapsulation. There are two primary parts to the implementation, a user-level daemon and a kernel-level encapsulation / decapsulation module.

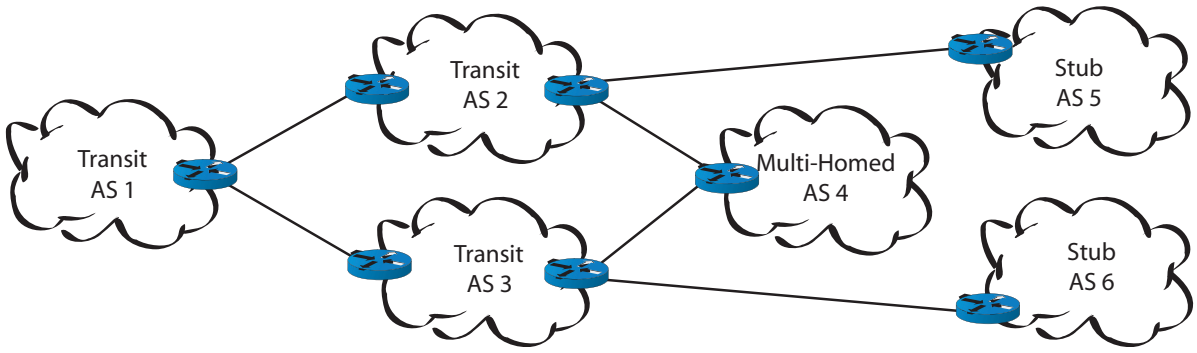


Figure 4. The network testbed used to evaluate HIDRA.

The implementation uses Linux's kernel firewall hooks to identify packets for encapsulation or decapsulation. The HIDRA kernel module handles all decapsulation. It also contains an encapsulation cache that maps L_1 addresses into L_0 addresses. If a packet requires encapsulation, and there is a corresponding entry in the mapping cache, the kernel module will encapsulate the packet. Otherwise the packet is sent to the user space daemon.

The HIDRA daemon resolves the L_1 destination into the appropriate L_0 destination. Once the mapping is resolved, the daemon creates a new entry in the kernel's mapping cache. This ensures that all subsequent packets sent to the same destination will be correctly encapsulated by the low overhead kernel module, instead of the high overhead daemon. In addition to resolving the mapping, the daemon is responsible for encapsulating and re-injecting all packets redirected to it from the kernel module. It also manages the HIDRA specific firewall rules.

The daemon exchanges iBGP information with the other routers inside the AS. The testbed routers are configured to peer with the HIDRA daemon. Receiving the full BGP table enables the proactive mapping lookup. It also allows the daemon to update the kernel's mapping cache and firewall rules automatically as routes are advertised and withdrawn. This is a critical part of handling network failures.

The daemon uses the iBGP peering sessions to originate the L_0 decapsulation route for its AS, and originate a default route that directs all L_1 traffic to itself for encapsulation before the packets leave the AS. The routers are configured to tag all routes in their L_1 with a well-known BGP community. The encapsulation boxes uses this community to determine which L_1 destinations are directly reachable and which ones require encapsulation.

5.2 Network

A testbed network is necessary to more accurately evaluate HIDRA. The AS-level network topology for this prototype network is depicted in figure 4. This network was physically created in one of the laboratories at our institution. Each AS consists of one Cisco 3640 or 3820 router with multiple 100Base-T or 1000Base-T network interfaces. Each AS also has an older 500Mhz Pentium III PC with 256MB of RAM running the HIDRA software stack. BGP is used to exchange both L_0 routes and proactive L_1 routes. The network uses private addresses and is otherwise not related to the commodity Internet.

In addition to HIDRA-based IPv4, the testbed routes IPv6 traffic between all ASes. This traffic does *not* use HIDRA. IPv6 is used as a control plane to coordinate experiments. All other traffic uses IPv4 and HIDRA.

5.3 Experiments

All experiments were run on the network topology depicted in figure 4. To ensure complete connectivity, an all-pairs ping was conducted before performing any experiment. The initial experiment was a simple base-line ping test between machines located in AS 5 and AS 4. This test was first run with the network in a legacy configuration. The legacy configuration is similar to how the Internet is operated now. The testbed was reconfigured for HIDRA using end-host encapsulation and decapsulation, and the ping was re-run. With end-host encapsulation each packet is encapsulated by the actual host it is being sent from before it is transmitted across the network. Decapsulation also takes place on the end-host. The results in both cases were very similar; both tests averaged pings of slightly under 1 millisecond, with the HIDRA network performing slightly slower as shown in figure 5. The performance difference is due to the extra CPU overhead of encapsulation and decapsulation.

The next test utilized the same two network configurations and demonstrates the failover capabilities of both networks. After the 30th packet was sent and the response received, the connection between AS 4 and AS 2 was manually broken. In both cases, this causes the active path between AS 4 and AS 5 to fail. BGP takes roughly 10 packets to route around the failure and use the longer path, AS 5 – AS 2 – AS 1 – AS 3 – AS 4. Because the packets traverse two extra hops, the round trip latency increases noticeably. After the 90th ping response was received, the link was restored. In both instances BGP detected and utilized the recovered link after a short period of time. This is visible in the graph when the round trip latency dropped back down to the original time.

Figure 6 shows the round trip ping latency between

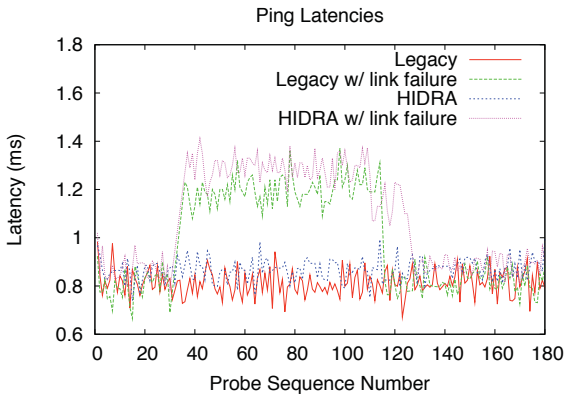


Figure 5. Ping latencies for HIDRA with end-site decapsulation between a host in AS4 and AS6 from figure 4. In two of the tests the link between AS3 and AS4 fails at time 30. Both architectures recover and continue forwarding traffic.

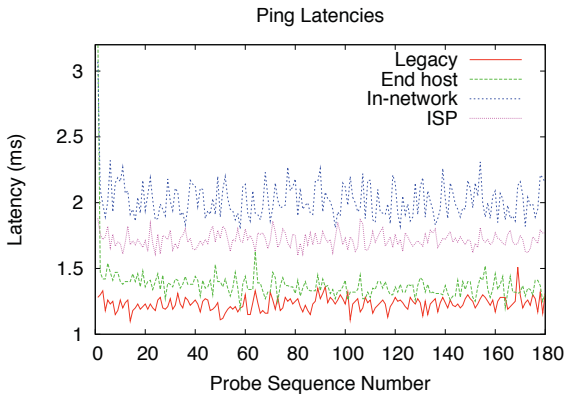


Figure 6. Ping latencies for HIDRA between a host in AS4 and AS6 from figure 4. Compares legacy, end-host encapsulation, within AS encapsulation, and ISP encapsulation.

AS 6 and AS 5 across the range of possible HIDRA configurations. The configurations include legacy and end-host encapsulation, both of which were described earlier. New configurations tested include in-network encapsulation and ISP encapsulation.

In-network encapsulation is tested by adding an additional machine to both AS 6 and AS 5. One machine was running a recent Ubuntu release and the other Windows XP. Neither machine has the HIDRA software installed. These two new machines were used to measure ping latency. For the packets to get encapsulated correctly they get routed to an encapsulation device in the same site as the host. The L_0 header is placed on the packet by this device before the packet leaves the site. The same is true for decapsulation. There is a noticeable increase in latency because the packet traversing an additional 4 links (to and from both the encapsulation and decapsulation device).

The final configuration is ISP decapsulation. In this scenario, AS 4, 5, and 6 are all configured as legacy networks and there is no HIDRA specific software running in any of those sites. AS 1, 2, and 3 are configured as HIDRA networks and tag routes originated by their customers as being part of L_1. Packets may traverse unencapsulated from AS 6 to AS 4 because they are part of the same L_1. If the packet is destined for AS 5 is remains unencapsulated, because AS 2 and AS 5 are part of the same L_1. If the packet is destined for another AS the encapsulation device in AS 2 will encapsulate it. The specific path used in this experiment, from AS 5 to AS 6, requires encapsulation. The L_0 packet will be decapsulated as it enters either AS 2 or AS 3, depending on the direction of communication. Again, because the packet is traveling along four extra hops we see an increase in round trip latency, shown in figure 6. Regardless of latency, the success of the pings demonstrates that all the configurations can correctly forward traffic.

Failover in the ISP encapsulation configuration was the final experiment. Pings were sent from AS 6 to AS 4. Since the best route between AS 6 and AS 4 remains in the same L_1, and because ISP encapsulation is used, the ping probes and responses are *not* encapsulated. After the 30th packet, the link between AS 2 and AS 4 is manually failed. Roughly 10 seconds after failure the pings are successful with a higher latency, due to the longer route. However this is more interesting because of the encapsulation involved. To utilize the AS 3 – AS 1 – AS 2 – AS 4 path, the packet must traverse L_0 and must be encapsulated. So, HIDRA automatically routed around the link failure *and* began encapsulating previously unencapsulated packets to do so. The link was restored after the 90th packet, and the network was able to adjust back to the original, unencapsulated path.

6 Future Work

The most important improvement to HIDRA is the inclusion of reactive routing. As described in this paper, proactive routing will reduce the size of the DFZ FIB, but it still requires each router to store the entire BGP table in its RIB. Reactive routing can substantially reduce the size of the RIB. Because it would no longer be required to have the entire BGP table, reactive routing also provides the opportunity to push the encapsulation burden all the way to the end hosts. We have a prototype reactive solution utilizing DNS as its route distribution protocol integrated in the HIDRA software, but it requires more development, especially as it relates to detecting and automatically recovering from link failures.

Support for using IPv6 as the L_1 protocol is another important missing software feature in HIDRA. This is slightly more complicated in the proactive architecture because the existing Internet routers do not universally exchange IPv6 routes with BGP. Integrating IPv6 support with reactive routing is the path of least resistance for IPv6. Future work entails adding both proactive and reactive IPv6 support.

7 Conclusion

As the number of routes on the Internet continues to expand, there is a pressing need for change to enable our hardware to keep pace. The concept of a hierarchical system has presented itself in many of the recent proposals, each with a different way to limit the required size of the DFZ forwarding table. Unlike other proposals, HIDRA offers a path to help effectively reduce both the immediate size as well as the rate of growth of the global DFZ forwarding table in an incremental fashion that attempts to remain fully backwards compatible. It utilizes many preexisting structures and protocols such as existing number allocation policy, BGP, and current router firmware. These pragmatic concerns separate HIDRA from many other proposals. Additionally, HIDRA enables future improvements, such as adding a reactive routing protocol which will further reducing the strain put on core routers. As such, we feel it surpasses many other proposals in that it can be realistically integrated to the existing Internet architecture.

References

[1] A proposal for scalable internet routing & addressing. http://tools.ietf.org/html/draft-wang-ietf-efit-00.

[2] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup. Compact name-independent routing with minimum stretch. 2008.

[3] Autonomous system (as) numbers allocation table. http://www.iana.org/assignments/as-numbers.

[4] L. J. Cowen. Compact routing with minimum stretch. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 255–260. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 1999.

[5] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. Locator/ID separation protocol (LISP). http://tools.ietf.org/html/draft-farinacci-lisp-12.

[6] Final report on ipv6 deployment issues. http://www.6net.org/publications/deliverables/D2.5.3.pdf.

[7] P. Francis and R. Gummadi. Ipnl: A nat-extended internet architecture. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 69–80, New York, NY, USA, 2001. ACM.

[8] Hitesh Ballani and Paul Francis and Tuan Cao and Jia Wang. Making Routers Last Longer with ViAggre. In *Proc. of USENIX Symposium on Networked Systems Design and Implementation*, Apr 2009.

[9] G. Huston. BGP Routing Table Analysis Report. http://bgp.potaroo.net/as6447/.

[10] L. Kleinrock and F. Kamoun. Hierarchical routing for large networks. *Computer Networks*, 1(3):155–174, 1977.

[11] D. Krioukov, K. Fall, and A. Brady. On compact routing for the internet. 2007.

[12] D. Krioukov, K. Fall, and X. Yang. Compact routing on internet-like graphs. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, 2004.

[13] U. of Oregon. Route views project. http://www.routeviews.org/.

[14] E. Reis. Ipv4 bgp table reduction analysis. http://mail.lacnic.net/pipermail/lacnog/2008-May/000046.html.

[15] Shim6: Level 3 multihoming shim protocol for ipv6. rfc5533 ; http://www.shim6.org/.

[16] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica. Towards a next generation inter-domain routing protocol. In *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*, 2004.

[17] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica. Hlp: a next generation inter-domain routing protocol. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 13–24, New York, NY, USA, 2005. ACM.

[18] M. Thorup and U. Zwick. Compact routing schemes. In *Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*, pages 1–10. ACM New York, NY, USA, 2001.

[19] TRRP. http://bill.herrin.us/network/trrp.html.

[20] P. F. Tsuchiya. The landmark hierarchy: a new hierarchy for routing in very large networks. *ACM SIGCOMM Computer Communication Review*, 18(4):35–42, 1988.

[21] R. Whittle. How do modern high-end routers implement the fowarding information base function? http://www.firstpr.com.au/ip/sram-ip-forwarding/router-fib.

[22] X. Yang. Nira: a new internet routing architecture. In *FDNA '03: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pages 301–312, New York, NY, USA, 2003. ACM.