A Secure and Scalable Internet Routing Architecture (SIRA)

Beichuan Zhang
bzhang@cs.arizona.edu

Vamsi Kambhampati
vamsi@cs.colostate.edu

Daniel Massey
massey@cs.colostate.edu

Ricardo Oliveira
rveloso@cs.ucla.edu

Dan Pei
peidan@research.att.com

Lan Wang
lanwang@memphis.edu

Lixia Zhang
lixia@cs.ucla.edu

ABSTRACT

Today's Internet routing architecture faces many challenges, ranging from scaling problems, security threats, poor fault diagnosis to inadequate support for traffic engineering and customer multihoming. By analyzing these challenges and learning lessons from previously proposed solutions, we gain two fundamental insights for designing a secure and scalable routing architecture: cutting a clear boundary between customer networks and transit providers, and embedding essential information in the address structure. We propose the Secure and Scalable Internet Routing Architecture (SIRA), a clean-slate design that separates provider networks from customer networks and embeds organization and location information in address structure. The resulting system provides dramatic improvements in scalability, security, fault diagnosis, and multihoming and traffic engineering support. We also identify new design issues raised by SIRA and sketch out straw-man solutions.

1. INTRODUCTION

The current Internet routing architecture, developed two decades ago, has proven to be a great success; it enabled the Internet evolution to a commercialized global system. However the routing architecture today faces fundamental challenges in critical areas such as scalability and security. In addition, it also suffers from lack of network fault isolation, inadequate support for traffic engineering and customer multi-homing. Incremental modifications have been suggested and had limited success. Other more groundbreaking ideas have been proposed [11, 8, 23, 10, 7], but each has its own perceived shortcomings that need further investigation. Fostering new innovations requires a fresh look at the underlying problems and guiding principles. With these motivations in mind, this papers examines lessons learned from the current state of the art and proposes a new Secure and Scalable Internet Routing Architecture, SIRA.

SIRA is based upon two core concepts:

- A logical separation between network transit providers and network customers; and
- A new address structure that embeds information of both network organization and metropolitan location.

Unlike the current Internet, SIRA places customer and provider routing in completely distinct spaces, and uses a mapping service to bridge the two routing spaces. SIRA's logical separation achieves scalability and stability in the provider space, eliminates re-numbering when customers change providers, and raises the barrier against malicious attacks aimed at the routing infrastructure. At the same time, the mapping service provides an effective means for customers to express their preferences on traffic flows.

The SIRA design also includes a new address structure. Although the current Internet intended a provider-based addressing, there exists *no* explicit association between provider identities and addresses, and resulting problems range from prefix hijacking to ineffective route aggregation. The current routing system also suffers from inadequate information to support traffic engineering and routing policies, which are essential for operating in a competitive environment. SIRA directly encodes the network organization and metropolitan location in an address.The complete SIRA address identifies the essential *units* in network connectivity which include (1) network organization, (2) metropolitan location, (3) subnet, and (4) network interface, and also encodes the *relation* between them, making a network address both informative and versatile in naming various types of components in the network topology.

In the rest of this paper, we first examine the lessons from both the operational Internet and several previously proposed solutions in Section 2, then use the identified principles to guide the design of SIRA in Section 3. Section 4 identifies new issues raised in SIRA design and sketch out solutions, and comparison of SIRA with related work is discussed in Section 6.

2. LESSONS

This section briefly reviews essential lessons learned from the operational Internet and also considers lessons from proposed designs that did not (or have yet to) succeed in deployment. Motivated by these lessons, the next section draws two main principles to guide the SIRA design.

2.1 From The Operational Internet

The operational Internet provides a rich set of important lessons for future designs. We sort the major ones into four categories: *scalability*, *security*, *fault diagnosis*, and *routing policy support*.

2.1.1 Scaling: A Tale of Two (Network) Worlds

We measure the Internet routing scalability in three dimensions: the size of the routing system, the size of the global routing table, and the number of routing updates. The Internet is made of an interconnection of a large num-

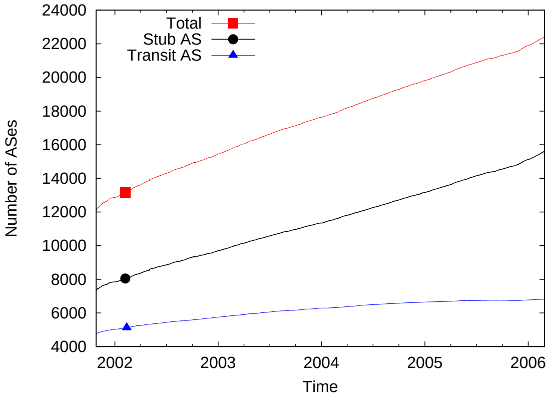

Figure 1: The Growth of Autonomous Systems (based on RouteViews data)

ber of Autonomous Systems (ASes) and BGP is the routing protocol that propagates reachability information to all the ASes. Figure 1 shows the Internet growth as measured by the number of ASes. Stub ASes, which correspond to customer networks, account for 70% - 80% of all ASes and grow nearly eight times faster than transit ASes (provider networks). There are also customer networks that do not show up in the global routing system as an AS, instead their reachability is advertised through the service providers. Thus the actual number of customer networks is much higher than the number of stub ASes. With the continuing penetration of Internet into our society, we expect customer networks will continue to drive the Internet growth. At the same time, our measurements also show that the interconnects among transit ASes are rapidly increasing, leading to an increasingly densely connected core with more and more customer networks at the edges. However because of the *flat-routing* nature at the AS level, a failure of any edge AS triggers routing updates to be propagated to *all* the other ASes.

The topology growth, however, is only one of the factors contributing to the rapid growth of the Internet backbone routing table [20]. Other factors include customers' desire of using provider-independent (PI) prefixes to avoid renumbering after switching providers, the increasing trend of site multihoming, and traffic engineering. A customer network C with a provider-assigned (PA) prefix P from provider $P1$ may also buy Internet connectivity from provider $P2$, i.e. *multihoming* with $P1$ and $P2$, for improved network reliability and performance. Because P is a provider-assigned prefix, C's multihoming to $P2$ not only requires $P2$ to advertise prefix P but also forces $P1$ to de-aggregate its own routing announcements. To make matters worse, today's traffic engineering practices further divide one prefix into multiple more specific ones and inject them into the global routing system. Multihoming and traffic engineering have been two major factors driving the global routing table growth lately.

Along with the routing table size growth, the growth in the number of routing updates also raises a big concern. Given Internet's sheer size, it seems inevitable that some customer networks may be improperly managed or inadequately connected. Due to the flat nature of the Internet routing, unfortunately, any single unstable customer network (a stub AS) can cause routing flaps to flood throughout the system [12], and configuration errors at a customer network often cause large scale damage [17]. Our own measurements also show that much of the "noise" in routing updates comes from a small number of unstable edge networks [24], and even more so when the network is under stress [33]. To compound the problems further, when a routing change propagates through the densely connected core, routers may explore multiple alternative paths before settling on the best one, i.e. the dense core further *amplifies* routing changes generated by edge nodes.

In summary, the growth of the Internet routing domain is mostly at the edges; the three major factors that contribute to the rapid routing table growth are the growth of edge networks, edge multihoming, and traffic engineering; and a major source of routing updates is the edge networks. These observations point us to the direction of separating customer networks from provider networks in order to build a scalable global routing backbone.

2.1.2 Security: Emergence of New Threats

In recent years the Internet has seen a rapid increase in malicious attacks, most of them were launched from compromised hosts. One recent survey[1] found that more than 56% of end hosts either had no anti-virus protection or had not updated it within the last week, 44% did not have a properly-configured firewall, and 38% lacked spyware protection. It is a clear indication that large quantities of compromised hosts will continue to exist for a long time to come. Using compromised end hosts, an attacker can easily launch DDoS attacks against one or more critical routers to bring down services vital to a business, or attempt to exploit software bugs or configuration errors to gain control over critical routers. Although routers in provider networks are not supposed to be involved in any end-to-end communication (except for operational purposes), nothing in the current architecture prevents end hosts from accessing routers in provider networks.

Today's routing protocols themselves are also vulnerable to attacks and operational errors. In one example, a provider router can mistakenly originate routes to prefixes it does not provide connectivity to, hijacking traffic destined to these prefixes. Current routing operations depend heavily on manual configurations, and such manual configurations are prune to errors. Operational mailing lists such as NANOG[22] provide numerous examples of (often unintentional) prefix hijacking events, such as those caused by configuration errors. However more recently malicious route hijacks are on the rise [27]. It is also difficult to detect false routing announcements today, because routers lack necessary information to distinguish between a valid origin AS change and a prefix hijacking event. Techniques such as [15, 34, 40] attempt to *patch* into existing routing system the missing information of prefixes to valid origin AS mapping, but these piecemeal patches also add complexity and are one driving factor behind calls for clean-slate designs.

Overall security threats against the routing infrastructure can come from multiple dimensions, but compromised hosts controlled by attackers are a main source of threats. Setting up a boundary between end systems (customers) and the routing infrastructure (providers) can both limit the threats and more clearly define the vulnerabilities in each system. It would also be useful to *design* necessary information directly into the routing system that could be used to detect faults and enforce security policies. For example, prefix hijacking would be easy to detect if prefixes directly carry information that identifies their origins and/or locations.

2.1.3 Fault Diagnosis: Why It Is Hard

Current routing protocols were primarily designed to route traffic around failures, such as link and node failures. They offer little help to diagnose the failures. However the Internet routing infrastructure is a complex system and network routing can be disrupted by various kinds of unpredictable failures that occur frequently. Examples of such failures include, but are not limited to, software bugs, memory corruption, link/interface instability, or mis-configurations. In order to promptly recover from these failures, network operators need accurate information about the location of the problems and the routers involved. Given a stream of update messages, however, it is often difficult (at best) to determine what event or events caused the update stream [35], making it difficult to react quickly and correctly.

This lack of diagnosis information also impacts routing protocol design. For example, route dampening mechanisms are intended to protect the network from update flooding by dampening the updates caused by unstable links. As we mentioned earlier, however, a single link status change can cause path exploration to produce numerous updates and falsely trigger route dampening[19]. If routing updates could carry more information for diagnosis purpose, that would enable one to eliminate false route dampening[39], dramatically improve routing convergence[25], and bring many other benefits. Diagnosis would also be easier if the sheer volume of updates can be reduced by a *separation* of customer and provider networks.

2.1.4 Routing Policy: Cumbersome Implementation

ISPs typically use routing policies to control the distribution of incoming and outgoing traffic in order to maximize resource utilization and revenue, and minimize cost. As transit providers get increasingly densely connected, effective support for routing policies is essential in the highly competitive transit system.

However routing policy implementation today is a tedious and error prune task. Although the current practice is called provider-based addressing, the basic routing unit is a prefix which contains no provider information. Large providers requested and obtained a large number of prefixes over the years, and as we mentioned above, this large set of prefixes get further fragmented due to traffic engineering and customer multihoming. Operators must manually configure routing policies against this large set of *prefixes*. Had the address structure embedded provider information, policy configurations and routing decisions could have been simplified.

Limited information may also lead one ISP's routing policies to conflict the policies of its neighbors and damage end user performance. For example, consider the scenario in Figure 2. Packets enter the ATT network in San Francisco and head to a destination reachable via Sprint. Given only this limited information, ATT attempts to minimize its cost by finding the nearest exit to Sprint (e.g. Seattle). Furthermore, Sprint also sends return traffic to ATT via nearest NYC link. Much better service could have been achieved if the traffic in both direction were routed via Chicago.

Furthermore, this hot-potato routing can also lead to large-scale traffic shift when the interior routing cost changes [30].

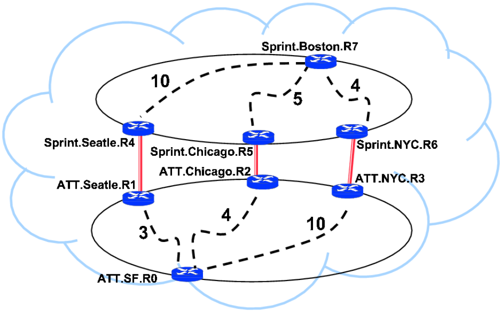

Figure 2: Traffic Engineering

Worse yet, each side may also try to use tools of questionable effect, such as AS prepending or Multi-Exit Discriminator (MED), to influence the entry point of inbound traffic. For example Sprint might adjust the MED value to make ATT send Boston traffic via NYC link, while ATT doing the same to make Sprint send return traffic via Seattle link, resulting in lose-lose scenarios as described in [18]. If both providers knew the topological locations of their next-hop neighbors with respect to the destination, they might be able to agree on a compromise for better overall routing decisions, ending in a win-win (rather than lose-lose) situation.

Customer networks may also have their own routing policies. A customer multi-homed to two providers may want each provider to carry a specific percentage of traffic. Another customer may have its preferences as well. If these two customers cooperate, they may satisfy both traffic preferences and produce a win-win situation. Unfortunately there is no means for them to express such preference explicitly in today's Internet.

Overall, what we have today is a large, ever-increasing set of address prefixes that provide no information to help routing policy settings. Furthermore, providers often set conflicting routing polices without mutual awareness. The end result is a complex and inefficient route selection process.

2.2 From Proposed Solutions

As the Internet developed and encountered problems, a number of alternate routing designs have also been proposed. Notable designs include Hinden and Deering's ENCAPS[11, 8], Deering and Hinden's metro based addressing [7], and Hain's Geo-based addressing [10]. These proposed solutions share major goals of avoiding renumbering from switching providers and scalable support for multihoming, however their approaches fall into one of the two categories, 1) separating customer and provider address space and 2) encoding location information in an address. Although these designs were not (or have yet to be) adopted for deployment, they offer important insights on both the new ideas themselves and the reasons why they have yet to materialize.

2.2.1 Separating Local and Global Routing

The current Internet runs on a single address space. Although today's IP address allocation is called provider-based addressing, it seems a misleading term. First of all, prefixes, as the basic routing unit, carry no provider information. Although providers can be identified by AS numbers, AS numbers are not part of the address structure. Second, there exist a very large number of provider-independent prefixes owned by customers, which enable them to avoid renumbering when changing providers. Finally, even when a customer network gets its prefix from a provider, it can announce the prefix out through another provider when multihoming, defeating any prefix aggregation attempt by the first provider.

Recognizing the fundamental conflict between address aggregation by providers for scalability, multihoming, and customer desire for provider-independent addresses, Hinden & Deering proposed ENCAPS in 1996 [11, 8] which separates providers and customers into two address spaces. Tunnels are used to carry packets from source customer networks over the provider space to reach destination customer networks. To encapsulate packets into tunnels, however, requires a lookup service to map the destination customer address to the address of the tunnel exit point which connects to the destination customer.

O'Dell [23] made another new routing proposal, named GSE, in 1997, where the basic idea is to divide IPv6's 16-byte address into two parts, with the lower N bytes being used for the End System Designator (ESD) and local routing, and the higher (16 - N) bytes (called Route Goop, or RG) used for routing between providers. One novelty in this design is to hide a customer site's RG from its internal hosts: the upper (16-N) bytes in the source address, which represents the site's provider, are filled in only when packets exit the customer site, and a multihomed site will have multiple RGs, one for each provider. In essence the upper (16-N) bytes represent the address space in the provider domain, hence GSE shares the fundamental idea with ENCAPS in envisioning a network where customers and providers have distinct address spaces.

However this separation of address space was considered a fundamental change of the original end to end Internet model and raised a number of open issues. Understanding the tradeoff between the gains and cost of such different designs takes time to develop, and the proposals could not be adopted without addressing those open issues first.

2.2.2 *Location-Based Addressing*

Another way to resolve the fundamental conflict between address aggregation by providers, multihoming, and customer desire for provider-independent addresses is to make the address allocation based on locations, instead of by providers. One such proposal made in the early 90's is to use metro-area based address as an alternative to today's provider based address [7]. The main goal was to avoid customer renumbering when changing providers.

More recently another location-based addressing scheme, Geo-based addressing[10], was proposed. Although there exist certainly differences between this proposal and metro-based addressing, for example the encoding of latitude and longitude information into the address instead of metro-area ID, the two proposals bear fundamental similarities. They are both proposed as one of the ways, but not necessarily the only way, to allocate IPv6 addresses, both envisioned coexistence of location-based addresses and provider-based addresses, and which type to use would be based on the need of individual parties.

However there has been a fair amount of resistance to these proposals, because routing based on those location-based addresses would not be able to reflect interconnectivity among providers, and support for routing policies is a common requirement for all routing decisions. Realistic and economically viable routing policies must reflect the interests of providers.

2.2.3 *Lessons Learned*

The early proposals for separate customer and provider address spaces were motivated by the fundamental conflict between address aggregation by providers for scalability, multihoming, and customer desire for avoiding renumbering. At the time, however, the routing scaling issue was not as acute as today, the number of multihomed sites was relatively small, the large volume of routing dynamics generated by edges had either not occurred or not been recognized, and malicious attacks from hosts had not been seen as a serious problem. Thus the tradeoff of making a fundamental change at the time was unclear. However the developments of above problems over last several years have made us revisit the previous proposals and gave us a deeper understanding of the advantages from separating customer and provider spaces. But our objective is far more than just providing distinct address spaces for scalability. We believe the objective is to provide truly distinct components with a sharp boundary, to help with not only routing scalability, but perhaps more importantly security and fault diagnosis.

We note that the earlier proposals for encoding location information into an address were mostly to get around the problems raised by provider-based address allocations. However an economically viable design must take provider economic interests into account as *first* priority, thus addressing the issue of "facilitating the routing of money". Replacing the current address allocation with location-based approach, even partially, is not a feasible approach. However, an address structure that contains provider information as a first priority could be enhanced with location information to open the door to a wide variety of new routing functionality and policy support.

3. THE SIRA ARCHITECTURE

Summing up the lessons learned from both the operational Internet and previous alternative design proposals, we come to the following design principles.

Principle 1: The routing architecture should draw a sharp boundary between customers and providers. The primary role of a customer network is to act as a source or destination for packets; The primary role of a provider network is to provide transit service and forward packets across the network. They are fundamentally different in terms of scaling, security, diagnosis, and traffic engineering concerns and objectives. A departure from the current Internet model, this principle makes a necessary tradeoff to carry the global routing into next stage of Internet evolution.

Principle 2: The address structure should encode organization, location, and network specific (e.g. subnet/interface) information, which are needed for aggregation, security, diagnosis, and traffic engineering. Current difficulties in supporting routing policy come from the lack of ISP information in the routing units (prefixes) to which the policies need to be applied. Previously proposed solutions, using separate address space and inserting metro information into address provide an incomplete solution. Metro addressing lacks strong policy support and separate customer/provider addressing lacks designs to support the new solutions (e.g. a mapping service between the two spaces). Both provider

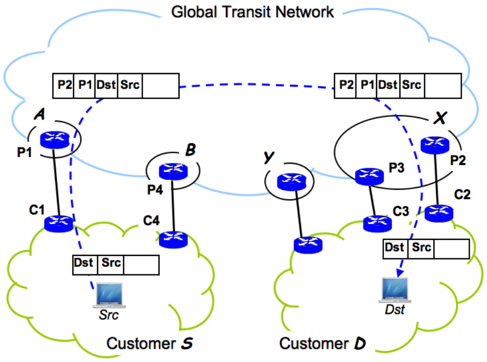

Figure 3: A Sample SIRA Session from Src to Dst

and location information are important.

The SIRA design follows these two principles, and here we describe the architectural changes and discuss their impacts.

3.1 Separating Providers from Customers

SIRA design separates provider networks and customer networks into two different routing spaces. The *provider space* comprises individual networks that provide data forwarding service. The *customer space* comprises stub networks that are sources or sinks of data traffic. Generally speaking a customer network connects to one or more provider networks, however occasionally two customer networks may also directly connect to each other. A stub AS in today's Internet corresponds to a customer network in SIRA. A transit AS in today's Internet splits into two parts in SIRA: the AS's customers become customer networks in SIRA's customer space, and all the IP boxes associated with transit services compose a provider network in the provider space. The resulting provider network includes all the transit routers and border routers that connects to customers, as well as servers such as those at network operation centers (NOC).Provider networks interconnect to form the Global Transit Network (GTN). The resulting system is illustrated in Figure 3.

SIRA implements the separation of providers from customers by the following three means. First and foremost, providers use a separate routing space. A provider routing protocol is operated among routers in the provider space to maintain reachability to all other providers only. It fundamentally differs from BGP in *its confinement within the provider space*. There is *no* routing protocol operating across the links between the provider and customer spaces. Each customer runs a routing protocol to maintain routes to reach internal subnets and its immediate neighbors (its providers or other directly connected customer networks).

Second, SIRA assigns separate *address spaces* to providers and customers. The two address formats have the same length, however they are clearly distinguished by the first N bits in the address. The next section describes SIRA's address structure in detail.

Third, SIRA allows *no direct communication across the two spaces*. Any packet that carries a customer source address and a provider destination address (or vice versa) is invalid and will be dropped by the first GTN router it encounters. In fact, the design of the routing protocols are such that a transit router in provider space does not have a route or forwarding entry to a customer address (and vice versa)[1]. Any *necessary* access from customer space to provider space, e.g., when operators remotely login to a router, must go through special proxies. These proxies have interfaces in both address spaces and must authenticate and authorize any access between customer and provider space. In fact, this model of operation is currently being used by a number of ISPs, where routers do not allow remote access except those from a pre-configured proxy with authentication credentials. Instead of relying on operators to manually configure such boundaries, SIRA provides the separation *by default* to enhance the security of the provider space.

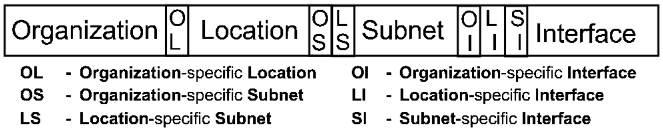
Organization
O L
Location
O S
L S
Subnet
O I
L I
S I
Interface
OL - Organization-specific Location
OS - Organization-specific Subnet
LS - Location-specific Subnet
OI - Organization-specific Interface
LI - Location-specific Interface
SI - Subnet-specific Interface

Figure 4: Address Format

Viewed from customer networks, the provider space is a single logical hop connecting all customer networks. End-to-end data delivery across this provider hop is achieved by encapsulating customer packets in a GTN packet header, with the source address as the GTN entry router and the destination address as the GTN exit router. A mapping service is needed to map the customer address to corresponding provider router. Figure 3 illustrates a typical example. When a source host *Src* (in customer network S) sends a packet to a destination host *Dst* (in customer network D), the packet will be forwarded to one of S' providers, say A. The ingress GTN border router (P_1) uses the information from the mapping service to find the egress GTN border routers (P_2) that connect to D. P_1 then encapsulates the packet with its own address as the source and P_2's address as the destination, and forwards it to P_2. Upon receiving the packet, P_2 will decapsulate it and send it to D.

We claim that the SIRA design conforms to Internet's original end-to-end transparency model. Although SIRA is composed of two routing spaces and introduces a dependency on the mapping service, the latter's impact is similar to that of the existing DNS service. On the surface the encapsulation step in crossing GTN seems like NAT (Network Address Translation), but SIRA allows any customer host to talk directly to any other customer host by guaranteeing the uniqueness of host addresses in customer space. The separation of, and the mapping between, two address/routing spaces are used to isolate customers from the global delivery backbone for critical security and scalability purposes; they have no impact on the end-to-end model.

3.2 SIRA's Address Structure

The address structure is the centerpiece of a routing architecture. A network address represents the attachment point of a device, be it a host or a router, in the topology. Because the Internet is a global interconnection of different organizations, packets are delivered by first forwarding them to the destination organization, then to the actual destination location, next to the particular destination subnet and

[1] A GTN border router will have forwarding entries for any customers directly attached to the border router.

finally to the destination interface. The current IP address structure lists the subnet and interface, but does not directly specify the organization or location. These last two components are *not* included in today's IP address structure, but are needed for making routing decisions. As a result, current routing practices find other means to embed and infer this information. For example, manual configurations of routing metrics between router pairs and routing policies based on individual prefixes indirectly reflect organization and location information.

SIRA designs a new address structure that includes each of the four essential components for fully describing a network attachment point: organization ID, location ID, subnet, and interface ID. Figure 4 shows the network address structure used in SIRA. We describe each component below. The same address structure is used for both provider and customer address spaces, where the first N bits of the address flags the distinction between the two address spaces.

Organization Component. Each organization is assigned a globally unique organization ID and all of its addresses begin with this ID. An organization is either a customer or a provider. If the first N bits[2] of the organization ID are zero, it is a *Provider ID*, otherwise it is a *Customer ID*.

Location Component. The second component is geographic location ID of the address, with a total length of L bits. The first L_1 bits encodes Continent ID, the next L_2 bits encodes Country ID, and the remaining bits encodes *metropolitan area* ID, or metro-ID in short. SIRA design assumes a list of globally defined metropolitan areas, and the metro-ID for each is the encoded longitude-latitude value of the metropolitan area location (similar to [10]). Each of the three sub-fields in the location ID is globally unique by itself without depending on the other two. This flexibility accommodates (perhaps rare) cases where one country may cross multiple continents, supports route aggregation at continent or country level, and routing to a specific metro area independent from whether the political structure in the region has changed (e.g. the merger of East and West Germany).

Subnet and Interface Components. The last two components of the address specify the subnet and interface and are similar to that in the current IP address structure. They are used to deliver packets to specific network attachment points.

Component Relationships. In addition to the four components described above, SIRA's address structure also contains flag bits to indicate the relationship among the components (Figure 4). Each bit indicates whether a later component is specific within the scope of an earlier component. The location component begins with an OL bit that is normally 0, meaning that the network attachment point is at a location within a given organization. The subnet component begins with an OS bit indicating whether the subnet spans multiple organizations and an LS bit indicating whether the subnet spans multiple locations. For example, a layer-2 switch connecting multiple providers at an exchange point would have $OS = 1$ and $LS = 0$, indicating that it connects different organizations at the exchange point location. Different combinations of the three OI, LI, and SI bits in front of the interface ID indicate whether the interface ID is unique across multiple organizations, locations, or subnets respectively. E.g., when the SI bit is 0, the interface ID is specific to the subnet. $SI = 0$ also implies that the interface inherits the subnet's dependency on location and organization, i.e., $OI = OS$ and $LI = LS$. When the 3 bits are set to 001, the interface ID is independent from the subnet field but dependent on the location and organization, which can be used as subnet-independent router ID. When the 3 bits are set to 011, the interface ID represents a unique ID inside an organization (and must be allocated as such). When all the 3 bits are set, the interface ID will be a globally unique ID from some global allocation service. In other words, an address cannot have all the 3 bits in front of the interface ID set unless the interface ID is obtained from a global allocation service. Other combinations follow similar logic. The net result is a flexible address structure that identifies organization, location, subnet, interface, and the relationship between each of these components.

3.3 Discussion

The two simple design ideas, *placing providers and customers in separate routing spaces* and *identifying various network components in the address*, provide fundamental advantages over today's Internet routing architecture. Below we discuss several of the important ones.

Scaling and Stabilizing the Routing Infrastructure. As discussed in Section 2, the main factors driving the global routing table growth are the growth of customer networks and site multihoming, and the main factor for routing dynamics is connection instability of customer networks. We first show how SIRA design contributes to the scalability and stability of the global routing infrastructure. Later this section, we show how SIRA can effectively support multihoming and traffic engineering with no impact on the routing table size.

Figure 1 shows that the number of transit ASes is only about 20% of the total ASes in today's Internet. In addition, the number of provider networks grows at a much slower rate compared with customer networks. Since GTN routers are only concerned with routing among providers, the global routing system will have a core with relatively small routing table size. Compared with today's Internet, SIRA's provider routing space is not affected by the dynamics of customer network connectivity, hence routing in GTN is expected to see much lower update rate and shorter convergence delay.

SIRA's new address structure also enables route aggregation at multiple levels of granularity, including organization, continent, country, metro area, and subnet. One can decide to keep more specific routes to prefixes that are close-by, as measured by either organizational distance (e.g. neighboring provider networks) or geographic distance (e.g. metro areas within a country), and aggregate routes to prefixes that are further away. Thus routing table size can be minimized without compromising providers' capability to implement routing policies and traffic engineering. This picture is

[2] Given there will be significantly more customer networks than provider networks, instead of designating the first address bit as the flag, we define the provider flag as having the first N-bits equal to zero. Assuming the organization field is O-bits total, we can have up to 2^{O-N} provider networks and $(2^O - 2^{O-N})$ customer networks.

in sharp contrast to the situation in today's Internet, where multiple prefixes announced by the same network are often different numerically and cannot be aggregated.

Securing the Routing Infrastructure. SIRA significantly raises the barrier against malicious attacks targeted at the global routing infrastructure. First, compromised hosts in customer space can no longer directly attack the provider infrastructure. The SIRA design prevents unauthorized accesses to any backbone routers by prohibiting direct access between customer and provider spaces. This separation disables the use of today's diagnostic tools such as `traceroute` by hosts to discover the path details in the provider space, but we believe that this is a reasonable sacrifice for the gained protection, and new diagnosis tools can be developed for SIRA.

Attackers can still use the compromised hosts within a customer network to DDoS local GTN border routers, but this has localized impact and is relatively easy to diagnose. Attackers may also use compromised hosts from multiple sites to DDoS the routing infrastructure by flooding packets to some remote customer destinations. However given the GTN topology is opaque, attempting to DDoS any specific component in the provider topology becomes difficult.

Second, because this separation is implemented by encapsulating customer packets, the encapsulation header has the entry router address to GTN as the source and the exit router address from GTN as the destination. In the absence of border router compromises, this encapsulation step *eliminates spoofed source address problem within the provider space*. In the current Internet, some provider border routers check the source address in packets coming from stub networks and reject spoofed addresses. However, this requires the provider border router to be configured with the list of prefixes owned by the customers and in addition there is no clear way of telling which provider fails to apply source filtering. In SIRA, the provider router simply needs to check the organization ID and verify if it matches the customer's ID. Furthermore, the provider router must attach *its own address* when encapsulating the packet and thus a provider router that fails to filter is clearly identified. When a DDoS attack occurs, aiming at either the routing infrastructure or a remote customer site, the entry routers of the attack traffic are readily identified and necessary steps can be taken to curtail the attack.

Third, by putting the organization ID into the address structure, SIRA eliminates false origin route announcements. Whenever a network announces someone else's prefixes, the immediate neighboring routers can readily detect the fault and stop it. For example, all AT&T's neighbor networks expect that any prefix originated by AT&T routers starts with AT&T's Organization ID. If an AT&T router originates one of Sprint's prefixes, neighbor networks can easily tell that this is a false announcement and drop it.

Although SIRA makes it difficult for attackers to gain access to GTN routers, it is still possible that routers in GTN may get compromised and cause damages within. Because SIRA makes launching effective attacks against GTN from customer networks difficult, attackers may also attempt to become a provider and enter GTN. Detecting compromised routers and misbehaving providers within GTN remains an open research challenge. However, we believe that SIRA design significantly raises the barrier to malicious attacks. We also expect that the reduced provider routing space and SIRA's address structure make the detection much easier, compared with the situation in today's Internet.

Faults Diagnosis. With the organization and location information, once a faulty machine's address is obtained, its location is known as well, which can be very useful in handling the fault. Administrators can make use of relationship bits during address assignment to facilitate fault diagnosis. For example, assume there's one router connecting three different subnets in the same metro location and organization. In SIRA, the administrator has the option of giving the three attachment points the same interface number and setting the SI bit to 1. This means the interface number identifies a box instead of a physical network interface, and can make fault diagnosis easier in many cases.

Multihoming and Traffic Engineering Support. SIRA's separation of provider and customer routing spaces eliminates the scaling issues associated with the current multihoming practice. In SIRA, a customer network's provider list is available from a mapping service somewhat analogous to the current DNS service. A SIRA sender consults the mapping service to obtain the destination address and *the destination's provider list*. This mapping service provides one step of indirection that can be utilized for effective support of multihoming and traffic engineering. A multihomed customer network (Dst) has an entry in the mapping service that includes its provider list (e.g., X and Y can be used to reach Dst) and any preferences associated with these providers (e.g., Dst would prefer to receive 80% traffic via provider X and 20% traffic via provider Y). The sender learns the receiving site's preference through the mapping service and can now make an informed decision based on both sender's and receiver's connectivity and preferences, taking full advantage of multihoming.

Traffic engineering within GTN can also be supported better. A pair of provider networks often interconnect at multiple locations. Knowing the locations of both the destination address and the interconnection points to neighbor provider networks, routers can make informed decisions to route packets efficiently. For example, recall the lose-lose scenario from Figure 2. In SIRA, $ATT : SF : R0$ may exclude $R1$ as the egress point because Seattle ($R1$) does not lead toward the destination (Boston). Then between the two remaining choices, $R0$ can pick the closer egress point, $R2$ at Chicago. Under heavy traffic load from $ATT : SF : R0$ to $Sprint : Boston : R7$, however, $ATT : SF : R0$ may consider splitting traffic between egress routers $R1$ and $R2$ in proportion. Knowing from which location the traffic enters GTN, Sprint network may also constrain from manipulating routing policy to force ATT sending traffic through NYC location. In this example, the location information helps avoid many drawbacks in today's routing policy support.

The above list is intended to highlight some benefits of SIRA, but not as a complete list. SIRA provides many other benefits. For example, Hinden and Deering's original goal of allowing customers to easily change providers without renumbering is met by SIRA; a customer simply needs to update its mapping entry after a provider change. In summary, with two simple ideas, the SIRA design brings many key benefits to network security, routing, scalability, fault diagnosis and so on.

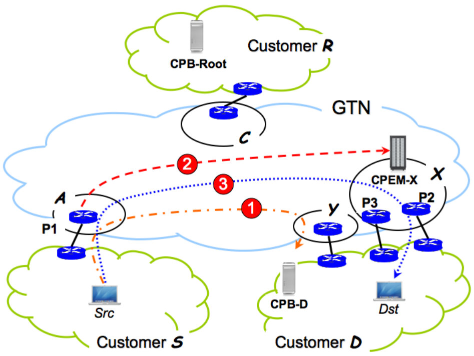

Figure 5: Mapping Service (CPB and CPEM)

4. SYSTEM COMPONENT DESIGN

Every coin has two sides. The separation of provider space from customer space provides solutions to a number problems facing the Internet today, at the same time it also creates a number of new issues which must be resolved in order to make SIRA work. The two primary new issues are the mapping service from customer address to the corresponding provider address, and the handling of failures at the border between customer and provider spaces.

Because SIRA makes a number of fundamental changes from today's practice, it also becomes necessary to redesign the routing protocols for both *Customer Routing* and *Provider Routing*. One would like to select source and destination providers to take full advantage of customer multi-homing. One would also like to fully utilize the rich information carried in the new addresses to reduce the complexity of routing system and facilitate traffic engineering in the provider space.

4.1 Mapping from Customers to Providers

The separation requires a service that maps a customer destination address to the set of providers that can deliver packets to the destination customer. This mapping service is not only necessary for end-to-end packet delivery, but also essential to fully utilizing customer multi-homing.

The main objective is to ensure the accessibility of the service, but not violate the separation of customers and providers. To achieve this we introduce a two-step mapping service. First, a Customer-to-Provider Binding (CPB) service identifies providers that can be used to reach the destination. Second, a Customer-to-Provider Edge Mapping (CPEM) service identifies the specific provider router. Figure 5 shows the major steps in sending from *Src* to *Dst*:

1. *Src* looks up *Dst* in CPB. After traversing the CPB hierarchy, the query reaches D's CPB server and returns D's provider list and preferences. *Src* chooses one destination provider, say X, includes it as a header option, and sends the packet.

2. Once P_1 receives the packet, it will query X's CPEM server to obtain the list of egress routers (P_2 and $P_3$and preferences.

3. P_1 chooses one egress router, encapsulates the packet and sends it.

Customer-to-Provider Binding (CPB). Given a customer address, the CPB service returns 1) the list of providers that connect to the customer, and 2) the customer's preferences for these providers. A source host uses this information to choose the destination provider. Since it is the customer network itself, not any provider, that has full knowledge of CPB information, CPB is located in customer space and is maintained by customer networks. In our example, *Src* learns the providers for *Dst* are X and Y. We propose implementing this service as a DNS extension.To be able to traverse the DNS hierarchy, a bootstrapping mechanism is needed for accessing the root servers. In the existing DNS, a local caching resolver (e.g. querier) is configured with the DNS root server addresses. Similarly, SIRA queriers are configured with the root server addresses *and the providers for these servers*. Just as in DNS, the querier first queries servers in the list and, provided at least one address is correct, obtains the current set of root server addresses and providers. When an higher-level DNS zone (e.g. edu) refers a querier to a lower-level DNS zone (e.g. ucla.edu), the referral specifies the lower-level server addresses *and their provider information*[3].

Customer-to-Provider Edge Mapping (CPEM). Given a destination customer address and provider for the destination, the CPEM service returns 1) the list of provider border routers that connect to the destination and 2) the provider's preferences for receiving traffic via these border routers. This service is located in provider space and each provider maintains its own customers' CPEM information. In our example, *Src* selected X as the provider; the CPEM identifies P_2 and P_3 as the provider border routers.To implement the CPEM, a provider can place many CPEM replica servers in its network. These servers have different interface ID, but share the same well-known prefix – `PID:AnyLocation:CPEM`. A CPEM query is sent to this prefix without specifying the interface ID, and it will be routed to the closest CPEM server by the provider's internal routing.

The two-step mapping service divides the mapping information and its maintenance along the customer-provider boundary. It not only conforms with the separation of two spaces, but also gives organizations the right incentives to improve their service as they only maintain information for their own interests. Providers are willing to offer good service to their own customers, and customers are willing to take care of their own information.

CPB lookup has the same delay as DNS, but CPEM incurs extra delay compared with current Internet. Caching can be used to reduce query delay effectively. Techniques such as pre-fetching for popular destinations are also useful. In case the source is using outdated mapping information, the destination can piggyback the current version number of its own mapping record in return packets, so that the source may do another lookup. CPB records can be secured by DNSSEC [4]. CPEM records can be authenticated using the provider's signature.

4.2 Handling Border Failures

In the current Internet, routing protocols take care of all

[3] Note that this does not mean that the DNS query to the destination's DNS server can be skipped because providers to this DNS server may not be the same providers to the destination host.

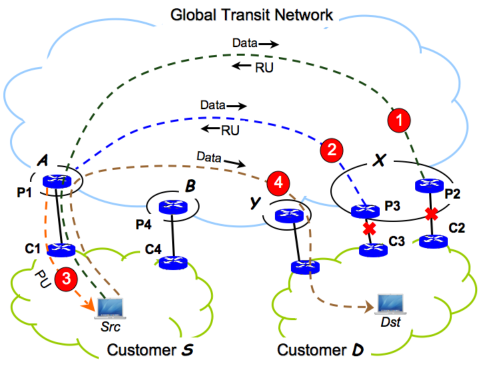

Figure 6: Demand-Driven Notification

topological changes (e.g., link/router failures and recoveries) by adjusting routing tables accordingly. In SIRA, provider routing and customer routing adapt to topological changes *within* their own space, but the border links between them are not in either customer or provider space.

A border link failure will trigger routing updates in the customer network. For example, in Figure 6, if link P_1–C_1 fails, routing updates will be propagated throughout customer network S to withdraw the path for provider A. When *Src* sends packets to *Dst*, S will automatically choose B as the source provider. This *proactive update* approach is the same as in conventional routing protocols.

However, provider networks do not maintain paths to customers and they should not be burdened with the routing dynamics caused by customers. For example, suppose link P_2–C_2 fails. Proactive updates would propagate the impact of a routing event to all the places, even though it may only affect data traffic from a few places. In SIRA, no routing updates are sent inside X, but the system must still react to this event so that future data traffic will be directed to alternative link P_3–C_3. To accomplish this, a new mechanism called *demand-driven notification* to handle border failures. If link P_2–C_2 in Figure 6 fails, demand-driven notification works as follows:

1. If a packet destined to D arrives at P_2, P_2 will drop the packet and send the ingress GTN router (P_1) a *Router Unreachable* (RU) message as a notification. It is similar to current ICMP Unreachable message.

2. P_1 uses RU message to update its mapping cache. For later data packets, the CPEM lookup will result in the alternative egress router (P_3) within the same destination provider.

3. If P_3–C_3 fails too, P_1 will receive another RU message. There is now no egress router available in the destination provider. P_1 will drop subsequent packets and send a *Provider Unreachable* (PU) message back to the sender host *Src*.

4. *Src* uses the PU message to update its mapping cache, and selects another destination provider (Y) to retransmit the packet.

PU and RU messages carry a TTL, after which they will timeout, and the corresponding router or provider will be regarded as available again. Ideally the TTL value should be set to the expected recovery time. This is relatively easy for planned maintenance, but hard for unexpected failures. Assuming most failures are short [13], TTL value of a few hours should be good for unexpected failures.

The main advantage of demand-driven notification is that it limits the impact to only *active* sources that will be affected by the particular topological change. Assuming at any moment, a customer network only communicates with a small number of other networks, then demand-driven notification avoids unnecessary churn for the rest of the Internet. On the flip side, demand-driven notification incurs extra delay and consumes more bandwidth for the first packet sent to the destination, because the packet gets dropped after crossing GTN, and retransmissions will not succeed until the notification message is received.

The tradeoff depends on the scale of the system and failure impact. Proactive update is more suitable where the overhead of propagating updates everywhere is manageable or when most of the destinations are very popular. Demand-driven notification is more suitable when the system is very large or when most destinations have only a few active sources. In the Internet, there are a large and increasing number of customers that may generate lots of border failures, while each failure may have only a small scope of impact. We believe that the stability of the global routing infrastructure outweighs the performance overhead.

4.2.1 *Performance Enhancements*

The performance overhead of demand-driven notification can be mitigated by several optional mechanisms. First, mapping entries can be updated after border failures or recoveries. Providers can update their CPEM and customers can update their CPB records. This helps inform new data sources of the topological changes and allows them to choose the right egress router.

Second, a destination provider (X) with a failed customer link can look up the destination customer address (*Dst*) in its own CPEM, find an alternative egress router (P_3) and forward any packet sent to the failed link. This is in addition to the RU notification message so subsequent packets will avoid the failed link.

Third, once the source provider (A) receives RU messages, it can share the information among all its border routers, so that subsequent packets coming from any border router will be able to choose the right egress GTN router. This can be done by multicasting RU messages to all A's border routers.

These mechanisms, like value-added services, cost more to providers. They can be configured on per-customer basis, i.e., only do certain things for certain customer networks. They are not mandated for SIRA to work, but can be offered by providers as premium services to their own customers.

4.3 Routing in Customer Space

One of the goals is to empower users to fully exploit multi-homing and stimulate competition among providers. The main questions are who makes the decision and how to choose providers.

4.3.1 *Provider Selection*

In SIRA, *source hosts select the destination provider* for end-to-end communication because they initiate the CPB lookup and have all the information (e.g., list of providers, preference, application type) to make a good decision. Once packets are sent into the customer network, the network can only get limited information about remote providers from the packet header.

On the other hand, *the source network is responsible for delivering packets to the right source provider based on its traffic condition and local policy*, because it has the aggregated view of local traffic and can make the best decision for the customer network as a whole. It can use traffic engineering mechanisms to shift the load from one source provider to another. More specifically, each border router can advertise a default route to AnyPID. In the absence of more specific routes, a packet will flow to the closest border router. The network administrator can influence the flow of traffic by manipulating the preferences associated with the default routes.

4.3.2 Internal Routing

A customer network chooses a protocol for its own internal routing. Existing protocols, such as OSPF [21], IS-IS and EIGRP [3], need to be modified to understand the new address structure, forwarding rules, route announcements, and route aggregation. New protocols may be developed to consider location information in routing decisions. The routers maintain paths to the customer's internal prefixes as well as its providers. Each customer border router injects a route for the provider it directly connects to. If a customer network connects to the same provider at multiple locations, each border router will announce its own route to the same PID. Each customer border router also announces a route to $AnyPID$ as a default route to reach the Internet regardless of through which provider. The routing table at each internal router has one entry for each of its providers and one default entry for the nearest provider. Forwarding packets follows simple rules: If the destination customer address matches any internal prefix, forward the packet along the best path to the prefix; Otherwise, forward the packet along the best path to source provider, which can be $AnyPID$.

A simple way is to choose source provider according to local network's traffic preference, and choose the destination provider according to remote network's traffic preference. It can also be based on policy, e.g., do not use Abilene as the source provider if the destination customer is not connected to Abilene. The decisions on source and destination providers can be related, e.g., if both customer networks connect to the same provider network, then choose this provider. It can also be application dependent, e.g., use high-bandwidth providers for file transfers, and low-delay-jitter providers for VoIP.

The customer network selects the source provider based on the destination provider information in the packet. The customer network has the aggregated view of local traffic and can make the best decision for the customer network as a whole, such as load balancing among multiple providers. On the other hand, the host has complete knowledge of the application and knows the destination's provider list and traffic preference. Thus the host may be able to make a better decision of source provider for the its application. SIRA allows the host put its choice of source provider in the packet, but whether to accept it or not is up to the customer network. The network may ignore the host's choice for short-term, but in the long run it should take host's choices into consideration in planning and improving network performance.

4.4 Routing in Provider Space

Fundamentally, routing provides two types of information for path computation. **Reachability information** indicates which way one can go to reach destination D. **Routing Metrics** help determine which path is best if there is more than one way to reach D. Therefore, in order to compute the paths to the *destinations within the GTN (Global Transit Network)*, the provider-space routing needs to address the following issues: (a) how to scale the reachability protocol to the number of destinations within GTN; (b) how to ensure that all the routers in the GTN have consistent reachability information; and (c) how to collect routing metrics to facilitate path selection. We now present these design issues and the proposed solutions.

4.4.1 Design Issues

Scalability of Reachability Information. Our provider-space routing has two scalability goals: 1) a small routing table size, and 2) avoiding unnecessary routing changes. By the design of SIRA, the GTN routing table contains only routes to provider-space entities, thus we have already reduced the scale of the routing system and have removed the edge instability from the routing system. However, we are still dealing with a system with several thousands of providers, each of which could have up to hundreds of routing entries. If the providers implement load balancing similar to the current Internet, we could still end up with big routing tables and unnecessary routing instability. We achieve small routing table size by enabling efficient address aggregation, and avoiding instability via incorporating RCN [25] and FRTR [32].

Consistency of Reachability Information. Current Internet routing computes reachability at two levels: *inter-domain* routing uses BGP to exchange information between domains and calculate global reachability, while *intra-domain* routing protocols such as OSPF calculate the reachability within a single domain. If some routers run only an intra-domain protocol, the inter-domain protocol and the intra-domain protocol need to synchronize their information via route redistribution. However, route redistribution has become a major source of operational errors. Moreover, inside a provider, BGP routers need to use full-mesh iBGP connections (typically over multiple router hops) to exchange routing information with each other, which leads to a major scalability problem for large providers. Route reflector and confederation have been introduced to improve the scalability of iBGP, but these short-term solutions can cause new problems [9]. To avoid these problems, we propose to use a single protocol, SIRA Path Vector Protocol (SPV) to provide *reachability* information for both inter- and intra-domain routing.

Routing Metrics. Routing metrics cannot be maintained by a single protocol, because there are no agreed upon metrics at the *inter-domain* level. Features such as knowledge of a competitor's topology or link capabilities are simply not available and may even be intentionally obscured due to competition between providers. We propose that each provider uses an internal protocol solely for the purpose of collecting topological information and computing the metrics associated with internal paths.

The combined result works as follows. SPV computes one or more paths to each destination. When presented with multiple paths to a destination, a router applies a combination of routing policies and internal network information learned from the topology maintenance protocol to select a best path. The forwarding table is updated appropriately and the reachability information is announced.

4.4.2 SIRA Path Vector Protocol (SPV)

In SIRA, SPV is used to maintain both inter- and intra-domain reachability. In other words, every router in the provider space runs SPV, so there are no route re-distribution and iBGP scalability problems.

Based partly on lessons learned in our work in BGP routing, we propose a design for SPV that adresses several limitations in BGP such as slow convergence and path exploration, lack of security, poor aggregation, potential for policy oscillations, and so forth. We start with the basic concept of associating a path with each route, but a fundamental new change to be explored is the granularity of the path. At a minimum, the path specifies the sequence of provider IDs used to reach the destination, but the path can include more specific information that may be useful for fault diagnosis, traffic engineering, and security. For example, the path (Sprint:NY, ATT:NY, ATT:LA, Qwest:LA) indicates that packets using this path will be passed from Qwest to ATT in Los Angeles, delivered across the US by ATT, and then passed to Sprint in New York. Within ATT, the path may be even more specific and include (ATT:NY:subnet3, ATT:chicago:subnet2, ATT:LA:subnet9, Qwest:LA). This level of detail may be useful within ATT, but could reveal sensitive information to other providers. Therefore, before announcing the path to Sprint, the ATT border router can abbreviate this path to (ATT:NY, ATT:LA, Qwest:LA) or simply (ATT, Qwest).

Since SPV routers can aggregate routes at different granularities, e.g. Provider ID (PID) only, or PID plus metro-ID, or PID plus metro-ID plus subnet-ID, routing table size can be minimized without compromising providers' capability to implement routing policies and traffic engineering. For example, instead of using multiple AS numbers for its networks in different geographic areas, AT&T could adopt a single PID and use the first bits of the metro area field to distinguish between the different areas. An SPV router's routing table may contain entries for ATT, ATT:northamerica.us, ATT:northamerica.canada, ATT:europe, ATT:asia, and so forth. The first route provides a default for reaching any AT&T address, and the other routes provide more specific routes to AT&T in the US, Canada, Europe, and Asia respectively. More specific routes are also supported and become increasingly more useful as one gets closer to the destination itself. A small provider may route to AT&T addresses using only the entries above. However within AT&T itself, the routing table would contain more specific routes to ATT:metro:subnets or even specific routers, but AT&T does not announce these detailed internal routes to external peers.

To reduce unnecessary instability, achieve fast convergence, and enable fault diagnosis, our new path vector protocol will also include some mechanisms that we have previously developed to improve BGP. Root Cause Notification (RCN)[25] explicitly signals the location of the failure, based on which routers can significantly reduce path exploration and slow convergence after failure or policy change. It also improves dampening of unstable routes[39], and aids in diagnosis[16]. Another mechanism, Fast Routing Table Recovery (FRTR)[32], allows routers to send periodic refreshes to ensure routing table consistency and enables faster recovery from session failures.

4.4.3 Topology Maintenance Protocol

To facilitate path selection, each provider uses a topology maintenance protocol, most likely a link-state variant, to collect the current network state. This protocol *does not carry any reachability information to prefixes.* Instead, it monitors the state of each link as well as bandwidth, delay and so forth. It can also compute the metrics associated with any path inside the provider network at the request of SPV. For example, an SPV router may need to choose between two internal paths. It can pass the two paths to the topology maintenance protocol to obtain their metrics. Then it will apply local routing policies to determine which path is more preferred.

5. EVALUATION

In this section we analyze some performance aspects of SIRA, including the routing table size scalability and CPEM lookup penalty. To evaluate routing table size we use one year of BGP data of RouteViews Oregon collector [2], from January 15^{th} to December 15^{th} 2005. To assess the CPEM lookup penalty we use one month of DNS logs from a university campus network.

5.1 Routing Table Size

We estimate the size of the global routing table using RouteViews Oregon collector RIBs, taking a sample of one day for each month of 2005. Figure 7 shows three curves, one for the table size in current BGP, counted as number of prefixes in the Routing Information Base (RIB), and the other two represent estimations for table size in SIRA, considering both one entry per AS and one entry per metro area. We define a metro area, based on the population distribution over a geographic region[7], i.e., cities with a high number of inhabitants will form a metro area, while less populated cities will be covered by the nearest metro area. For example, in the US, there are 70 cities with more than 250,000 inhabitants [5].If we assume this as reasonable threshold for a major city, we can then divide the US into 70 metro areas. We did a similar analysis for other countries and computed the number of metro areas based on the country's total population [5]. We then used Regional Internet Registry (RIR) data [28] to find the country to which each AS was allocated. We assume each AS contributes to the table size with a number of metro areas of the respective country where it is allocated. For this analysis we only considered ASes that belonged to ISPs (we excluded stub ASes).

As shown in the figure, if SIRA uses one entry per metro area, the table size will be very close to that of current BGP, but if SIRA uses one entry per AS, the size becomes more than an order of magnitude lower than the current BGP table size. Note that with subnet addressing, the numbers for table size in SIRA can be much higher than these. However, we expect that route aggregation in SIRA will compress the table size, reducing the number of entries for remote ISPs. Furthermore, if the information in the metro area is struc-

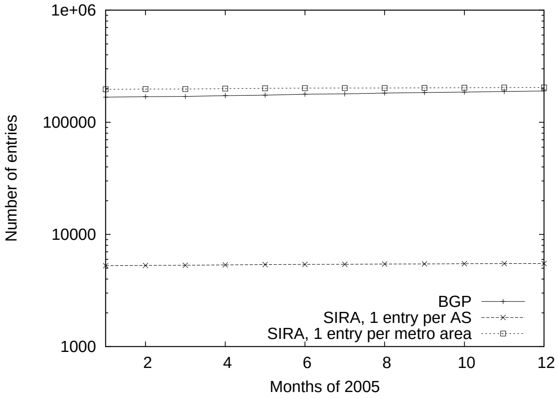

Figure 7: Estimated routing table size for SIRA.

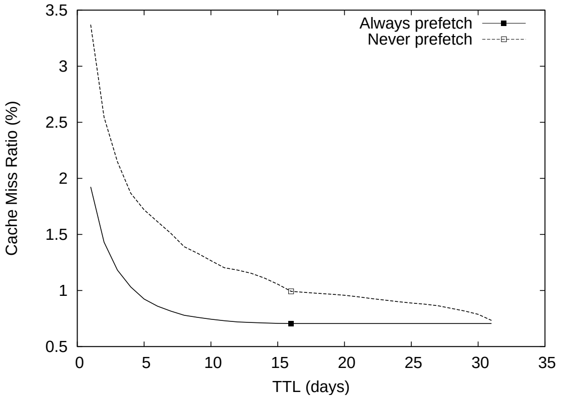

Figure 8: Cache miss ratio for different values of TTL, with and without prefetching.

tured , e.g., divided in country, region, city, etc, aggregation based on geographic information can become very effective. Also, note that the size of the current BGP table has a steady growth over time, as reflected by the addition of new stub ASes to the network. Whereas in case of SIRA, the size remains almost constant over the 1-year interval. This is mainly because the set of providers originating entries in the RIB barely changes over time.

Note that some of the prefixes originated by service providers are addresses of their customers. Some customers run private BGP sessions with their providers and delegate to them the prefix advertisement. The provider then removes the private AS number and announces the prefix to the global routing system. Therefore the analysis done here is actually a worst-case analysis, since we are assuming all prefixes originated by providers in current BGP will still be present in the P-space of SIRA, whereas in reality, only a fraction of these will be advertised, and the remain will be in C-space.

5.2 Analysis of CPEM Caching Mechanism

In this section we analyze the cache performance of CPEM lookups. In addition, we investigate how cache prefetching can improve the overall performance of the system. In SIRA, GTN border routers perform CPEM lookups whenever destination customer network information is not available in the local cache, i.e., when there is a cache miss. When the lookup succeeds, the entry is inserted into the local cache. Note that the lookup is only performed for the *first packet* of a flow, for a given destination. Instead of measuring lookup costs per packet, we use DNS logs to model *flows* generated from a customer network, i.e. we assume each DNS request in the trace corresponds to a different flow originated from C-space. For the purpose of this evaluation we use one month of DNS logs from January 15^{th} to December 15^{th} 2005, taken from the DNS server of a university computer science department. We cleaned the DNS logs by removing invalid entries and converted the IP addresses returned by the queries to prefixes based on the RIB snapshot from November 10^{th}, 2005 from RouteViews Oregon collector. The resulting DNS log file has $1,203,065$ flows.

In order to evaluate the lookup penalty, we simulated the cache behavior for each request in the trace assuming each record in cache had the same TTL value in each simulation run (we tried values of TTL from 1 to 31 days). We then computed the *cache miss ratio* per flow, i.e. the fraction of flows that would result in a CPEM lookup. We compare cache miss ratio without prefetching and with prefetching. When using prefetching, whenever the TTL for the record expires in cache, the GTN router will automatically fetch the entry again from the CPEM server. However, this prefetch operation is only done once per record, i.e. when the record expires a second time, it will be removed from cache. Note that without prefetching, the entry is removed from the cache when the TTL expires.Figure 8 shows the cache miss ratio for different values of TTL. First, we observe that the cache miss ratio decreases as the value of TTL increases. This is due to the fact that entries will stay longer in cache with larger TTL values. However, longer TTLs increase the probability of finding a stale entry in cache. Second, from Figure 8 we can see that when using prefetching, the cache miss ratio reduces by almost a factor of 2 for small TTL values. Third, we see that as the values of TTL approach the duration of the trace (31 days), the leverage of prefetching is not so relevant, mainly because only a small number of entries will expire from cache. Note that when using prefetching, a cache miss can only occur when a request arrives after 2xTTL days since the time the record was inserted in cache, or if the request is for an entry that was never inserted in the cache. By looking at Figure 8 we observe that the miss ratio with prefetching is close to 0.7% for any TTL value above 10 days. This is because windows of more then 2x10=20 days are likely to cover almost all requests during the 31 days trace. In addition, when using prefetching and a TTL of 1 day, the probability of a cache miss is only 2%, which we believe is a good balance between finding a stale entry in cache verses not finding an entry in cache.

Using our observation from section 5.3.3, we can use a TTL value of 1 day, knowing that less than 7% of provider changes occur within that period, hence the probability of finding a stale entry in cache will be less than 7%. With prefetching, this will give us a cache miss ratio of 1.9% (Figure 8).

If m is cache miss ratio, the mean latency of CPEM lookups, Δ, will be given by:

$$\Delta = (1-m)d_{cache} + m(d_{cache} + d_{CPEM}) \qquad (1)$$

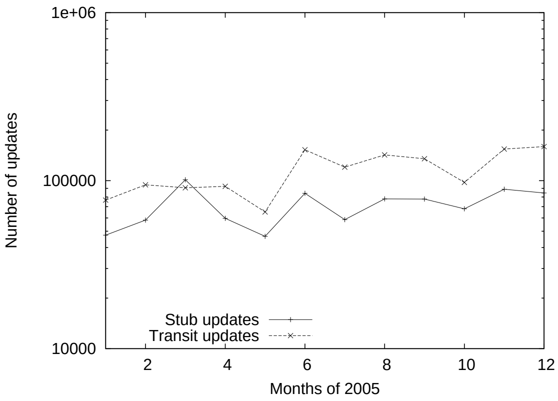

Figure 9: Average number of BGP updates per router.

Where d_{cache} is delay to access CPEM cache and d_{CPEM} is delay to access CPEM server. Assuming $d_{CPEM} \sim 100ms$, based on access delay to DNS servers [14], and considering d_{cache} is negligible, we can estimate $\Delta \sim 2ms$, with a probability less than 7% of finding stale entries in cache. We can extend this analysis to DNS cache, since in our scheme each customer provider list is stored in DNS, hence in order to find adequate values for TTL we still need to take in account how often customers change providers, and we will find similar numbers as above.

5.3 Orphan subsections

5.3.1 Number of BGP Updates

SIRA will only use BGP to distribute routing information inside the P-space, thus routing updates caused by edge dynamics in current BGP will not be present in SIRA. In order to estimate the number of BGP updates generated by each router in SIRA we used update traces from RouteViews Oregon collector, taking a sample of one day for each month of 2005. Figure 9 shows the upper and lower bounds for the average number of updates originated per router in SIRA. We used the following heuristic to compute the upper and lower bounds: if a BGP update has an AS_PATH that begins with a provider, we are sure the update resulted from some event in P-space. These updates are a lower bound of the real number of updates per day we will see for each router in P-space in SIRA.If a BGP update has a stub AS as the origin in AS_PATH, then we cannot tell if the update was a result of events in P-space or C-space. In summary, if upd_{total} is the total number of updates we see in current BGP, then $upd_{total} = upd_{stub} + upd_{provider}$, where upd_{stub} and $upd_{provider}$ are updates that have a stub or a provider as the AS_PATH origin, respectively. The SIRA lower bound is given by $upd_{provider}$, whereas the upper bound is given by upd_{total}. The figure shows a reduction of up to %40 in the number of update messages originated per router in SIRA when compared to current BGP. Also, note that this reduction is actually the worst case scenario due to the reasoning presented at the end of section 5.1 (since some of the providers in current BGP will no longer be providers in SIRA).

5.3.2 Number of Notification Messages

As mentioned in section **??**, the connectivity between a destination provider and its customer may be temporarily unavailable. In that case, the destination provider has to send *Provider Unavailable* messages to all the active source hosts currently using this provider to reach the customer. In addition, when the customer recovers the connection with its previous provider, the customer can notify again the active source hosts, so that they can mark the link as available, instead of waiting for the TTL to expire. Because *link up* information can be piggy-backed in data packets, we will only consider only *Provider Unavailable* (PU) messages in the following analysis, and assess the resulting overhead in SIRA.

We model link failures between customers and providers as a Poisson process, with an average interval between link failures of $\frac{1}{\lambda}$ days, and negligible failure durations, i.e., we only consider non-overlaping failures per customer. If a customer is multi-homed, we consider the routes to reach it are evenly divided between its providers. So if there are S active sources at any given time with active connections to the same customer, and its degree of multihoming is d, there will be $\frac{S}{d}$ sources using each provider, and there will be $\frac{S}{d}$ notification messages sent for each link failure. Let X represent the random variable of number of failures per day for a given link. The average number of notification messages per day M_{PU} of a customer with d providers is given by:

$$\begin{aligned} M_{PU} = \quad & d \cdot \tfrac{S}{d} \cdot E[X] \quad = \\ = \quad & S \textstyle\sum_{n=1}^{\infty} n \frac{\lambda^n e^{\lambda}}{n!} \quad = S\lambda \end{aligned} \qquad (2)$$

In current BGP, all link changes must be propagated to all the nodes in the network N, which means that the number of update messages caused by customer link dynamics is given by $M_{update} = N\lambda$ (excluding path exploration).Comparing this with (2), we observe that BGP originates $r = \frac{M_{update}}{M_{PU}} = \frac{N}{S}$ more messages caused by edge dynamics. Taking in account the observed number of simultaneous active flows in backbone links of a tier-1 provider [26], we can estimate $S \sim 30$. The current number of ASes in the Internet as of Jan 29th 2006 can be estimated as $N \sim 21,312$, yielding $r = \frac{N}{S} = \frac{21312}{30} = 710$. SIRA achieves a reduction of almost 3 orders of magnitude in number of messages caused by edge dynamics over current BGP.

5.3.3 Provider Changes

How frequently does a customer change providers? In order to answer this question we analyzed an AS level topology file from Jan 29th 2006 [31], containing information about AS level links.This file is created from RIBs and BGP updates collected from RV, RIPE, looking glasses, route servers and routing registries. The file has information about:

- AS numbers of each end of the link
- First time the link was observed: t_0
- Last time the link was observed: t_1
- Link type: edge, core,...

The first step was to filter only edge links, i.e. those connected to stub ASes, based on the link type information. A given link may exist in the Internet, but be captured only

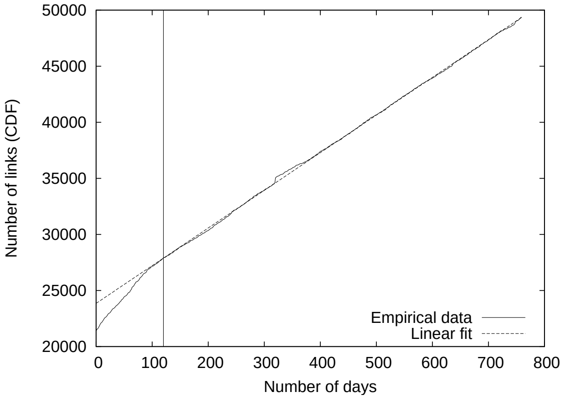

Figure 10: Distribution of $t_0 - T_0$ for edge links, $B =$ 120 days.

some days after the start of the collection process. On the other hand, a link can exist in the topology file, but not be observed for some days, although the link is still present in the Internet. Therefore, we must find values for thresholds B and D such that every change we observe in the interval [T_0+B, T_1-D] reflect a real change in Internet topology, where T_0 is initial collection time and T_1 is last time the topology file was updated. If for a given link $t_0 > T_0 + B$ then we are sure the link appear in the Internet at time t_0. Conversely, if for the same link $t_1 < T_1 - D$ we can be certain the link disappear from the Internet at time t_1.

In order to find B, we plot in Figure 10 the values of $t_0 - T_0$ for all links, i.e. the time elapsed since we start the collection until the time the link was first observed.

We assume edge links grow linearly over time [37], so the initial non-linear phase we observe in the figure must be a distortion introduced by the collection process. In fact, it is expected that in the initial phase of the collection process we capture more links, including those that were already present in the topology. After this initial period, we will capture mainly the new links that are added to the network. By setting $B = 120$ days we will capture only the linear growth phase.

Figure 11 represents the distribution of the values of $t_1 - T_1$ for all edge links. As for link appearance, we assume links disappear from the Internet at a linear rate [37], therefore the initial non-linear phase we observe is caused by the collection process. We capture only the linear phase by setting $D = 180$ days. We can now determine safely (1) the exact time the link appear in the Internet if $t_0 > T_0 + B$ and (2) the exact time the link disappear from the Internet if $t_1 < T_1 - D$.

Based on above observations we can now analyze how customers change their providers. Note that a change can be either a provider addition or a provider removal. In Figure 12 we plot the distribution of the time interval between changes of providers for each stub AS.

The figure shows that about %50 of changes occur within an interval of 35 days and %90 of changes occur within an interval of 246 days (∼8 months).

We already know how often there is a change in the provider set of each customer, but we do not know how long each customer stays with a given provider. We define *service period*

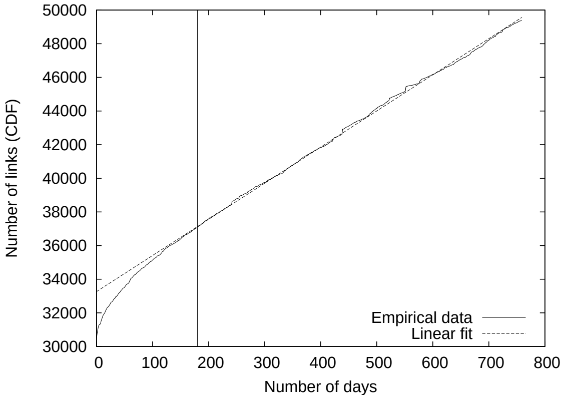

Figure 11: Distribution of $t_1 - T_1$ for edge links, $D =$ 180 days.

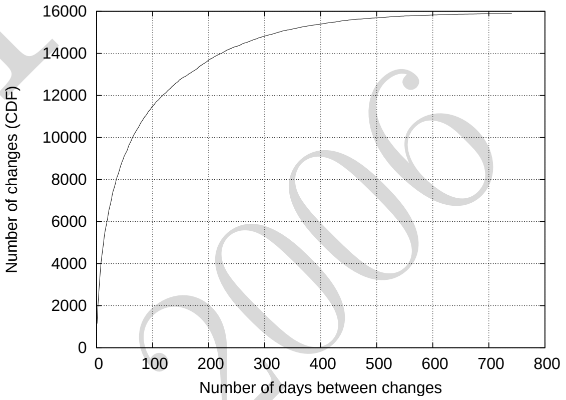

Figure 12: Interval between provider changes.

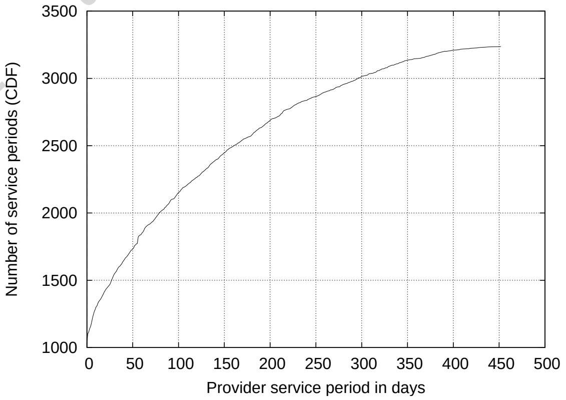

Figure 13: CDF of provider service period per customer.

of a provider as the interval $[t_0, t_1]$, where $t_0 > T_0 + B$ *and* $t_1 < T_1 - D$, B and D are the thresholds computed previously, t_0 and t_1 are first seen and last seen time stamps of the link between the customer and provider. This means that in a service period, we are certain about the appearance and removal instants of the link, t_0 and t_1 respectively. Figure 13 shows the distribution of the provider service period per customer. First observation is that there are 1,031 service periods (corresponding to 32% of cases) with a time of 0 days, meaning that the link between customer and provider was only used within a single day. We conjecture these cases happen in customers with very stable links to its providers that eventually fail in a certain day, making the customer change to a backup provider. However, the failed connection is restored within the same day, making the customer switch back to the original provider.

We also observe that 90% of service periods have a duration within 266 days (∼9 months).

6. RELATED WORK

Section 2 examined some lessons learned in today's Internet operations and examined some proposed (but not deployed) designs on separating customer and provider address space and adding location information to addresses. To see far and reach high, we tried our best to climb on the shoulders of giants. The design of SIRA was influenced by O'Dell and especially Hinden and Deering's idea of separating customer routing from provider routing. SIRA was also inspired by the idea of incorporating metro-location information into address structure, originally proposed for very different purposes.

More recently, the design of HLP [29] focuses specifically on improving the scalability and stability of today's BGP operations. It shares a common goal with SIRA of isolating edge instability from the backbone core by using a hybrid link-state and path-vector approach, with the core running BGP as is, and compartmentalizing the lower tiers in Internet's topological hierarchy into separate regions using link-state routing. HLP can provide improved scalability and stability in core routing, but it does not directly address the issue of protecting the core routing infrastructure.

[6] took a top-down approach to develop a new Internet address architecture, starting with an abstract architectural model. FARA abstracts communications between end-to-end entities as associations, and assumes the existence of a communication substrate that can deliver data according to an abstract notion of Forwarding Directive (FD). However FARA by itself does not specify the content of FD or how it may be implemented. In that respect one may view SIRA as a complement to FARA, with the main focus on the design of the packet delivery substrate.

[36] perhaps seems more closely related to our work than the above two, in that it also utilizes the address structure to achieve specific routing goals. However NIRA's design solely focuses on how to empower end users with the ability to select their own provider routes. SIRA selects providers for exiting and entering customer networks, but routing in the provider space is left strictly to provider routing protocols.

7. CONCLUSIONS

This paper presented the key concepts in SIRA design, separating providers from customers and embedding essential information in address structure. We also identified major issues raised by SIRA design and sketched out some preliminary solutions. SIRA provides significant advantages in system scalability, security, fault diagnosis, and multihoming support. It also offers great opportunities to improve other parts of the routing infrastructure by taking advantage of SIRA. Our ongoing work includes analyzing engineering trade-offs in a complete realization and exploring new advantages offered by SIRA. A more detailed discussion of SIRA and its design trade-offs can be found in [38].

8. REFERENCES

[1] AOL/NCSA Online Safety Study. `http://www.staysafeonline.info/pdf/safety_study_2005.pdf`, December 2005.

[2] Advanced Network Technology Center and University of Oregon. The RouteViews project. `http://www.routeviews.org/`.

[3] B. Albrightson, J. J. Garcia-Luna-Aceves, and J. Boyle. EIGRP–A Fast Routing Protocol based on Distance Vectors. In *Networld/Interop 94*, May 1994.

[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. *RFC 4033*, 2005.

[5] T. Brinkhoff. City population statistics [cited 2006-01-29]. http://www.citypopulation.de.

[6] D. Clark, R. Braden, A. Falk, and V. Pingali. FARA: Reorganizing the Addressing Architecture. In *ACM FDNA Workshop*, August 2003.

[7] S. Deering. Metro-Based Addressing: A Proposed Addressing Scheme for the IPv6 Internet. Presentation, Xerox PARC, July 1995.

[8] S. Deering. The Map & Encap Scheme for Scalable IPv4 Routing with Portable Site Prefixes. Presentation, Xerox PARC, March 1996.

[9] T. G. Griffin and G. Wilfong. On the correctness of IBGP configuration. In *ACM SIGCOMM*, August 2002.

[10] T. Hain. An IPv6 Provider-Independent Global Unicast Address Format. draft-hain-ipv6-pi-addr-09.txt, 2006.

[11] R. Hinden. New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG. *RFC 1955*, 1996.

[12] G. Huston. 2005 – A BGP Year in Review. APNIC 21, March 2006.

[13] G. Iannaccone, C.-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of Link Failures in an IP Backbone. In *ACM SIGCOMM IMW*, November 2002.

[14] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. Dns performance and the effectiveness of caching. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 153–167, New York, NY, USA, 2001. ACM Press.

[15] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (Secure–BGP). *IEEE JSAC*, April 2000.

[16] M. Lad, D. Massey, and L. Zhang. A graphical tool for capturing BGP routing dynamics. In *NOMS*, April 2004.

[17] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP Misconfiguration. In *ACM SIGCOMM*, August 2002.

[18] R. Mahajan, D. Wetherall, and T. Anderson. Negotiation–Based Routing Between Neighboring ISPs. In *NSDI*, May 2005.

[19] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz. Route Flap Damping Exacerbates Internet Routing Convergence. In *ACM SIGCOMM*, August 2002.

[20] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang. IPv4 Address Allocation and BGP Routing Table Evolution. In *ACM SIGCOMM CCR*, Janurary 2005.

[21] J. Moy. OSPF Version 2. RFC 2328, SRI Network Information Center, September 1998.

[22] The NANOG Mailing List. `http://www.nanog.org/`.

[23] M. O'Dell. GSE – An Alternate Addressing Architecture for IPv6. draft-ietf-ipngwg-gseaddr-00.txt, February 1997.

[24] R. Oliveira, R. Izhak-Ratzin, B. Zhang, and L. Zhang. Measurement of Highly Active Prefixes in BGP. In *IEEE GLOBECOM*, November 2005.

[25] D. Pei, M. Azuma, D. Massey, and L. Zhang. BGP–RCN: Improving BGP Convergence Through Root Cause Notification. *Computer Networks*, 2005.

[26] S. I. project [cited 2006-01-29]. http://ipmon.sprint.com.

[27] A. Ramachandran and N. Feamster. Spamming with BGP Spectrum Agility. NANOG 36, February 2006.

[28] RIPE. Regional internet registry [cited 2006-01-29]. ftp://ftp.ripe.net/pub/stats.

[29] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, Z. M. Mao, S. Shenker, and I. Stoica. HLP: A Next Generation Inter–domain Routing Protocol. In *ACM SIGCOMM*, 2005.

[30] R. Teixeira and J. Rexford. Dynamics of Hot-Potato Routing in IP Networks. In *ACM SIGMETRICS*, 2004.

[31] I. topology collection [cited 2006-01-29]. http://irl.cs.ucla.edu/topology.

[32] L. Wang, D. Massey, K. Patel, and L. Zhang. FRTR: A Scalable Mechanism for Global Routing Table Consistency. In *DSN*, 2004.

[33] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Observation and Analysis of BGP Behavior Under Stress. In *ACM SIGCOMM IMW*, 2002.

[34] R. White. Securing BGP Through Secure Origin BGP. *IPJ*, September 2003.

[35] J. Wu, Z. M. Mao, J. Rexford, and J. Wang. Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network. In *NSDI*, May 2005.

[36] X. Yang. NIRA: A New Internet Routing Architecture. In *ACM SIGCOMM FDNA Workshop*, 2003.

[37] B. Zhang, R. A. Liu, D. Massey, and L. Zhang. Collecting the internet as-level topology. *Computer Communication Review*, 35(1):53–61, 2004.

[38] B. Zhang, D. Massey, D. Pei, L. Wang, L. Zhang, R. Oliveira, and V. Kambhampati. A Secure and Scalable Internet Routing Architecture (SIRA). Technical Report TR06-01, University of Arizona, April 2006.

[39] B. Zhang, D. Pei, D. Massey, and L. Zhang. Timer Interaction in Route Flap Damping. In *ICDCS*, June 2005.

[40] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang. An Analysis of BGP Multiple Origin AS (MOAS) Conflicts. In *ACM IMW*, Oct 2001.