NIRA: A New Internet Routing Architecture

Xiaowei Yang*
MIT LCS
yxw@lcs.mit.edu

ABSTRACT

This paper presents the design of a new Internet routing architecture (NIRA). In today's Internet, users can pick their own ISPs, but once the packets have entered the network, the users have no control over the overall routes their packets take. NIRA aims at providing end users the ability to choose the sequence of Internet service providers a packet traverses. User choice fosters competition, which imposes an economic discipline on the market, and fosters innovation and the introduction of new services.

This paper explores various technical problems that would have to be solved to give users the ability to choose: how a user discovers routes and whether the dynamic conditions of the routes satisfy his requirements, how to efficiently represent routes, and how to properly compensate providers if a user chooses to use them. In particular, NIRA utilizes a hierarchical provider-rooted addressing scheme so that a common type of domain-level route can be efficiently represented by a pair of addresses. In NIRA, each user keeps track of the topology information on domains that provide transit service for him. A source retrieves the topology information of the destination on demand and combines this information with his own to discover end-to-end routes. This route discovery process ensures that each user does not need to know the complete topology of the Internet.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.2 [**Network Protocols**]: Routing Protocols

General Terms

Design, Economics

1. INTRODUCTION

This paper is concerned with the question of how Internet traffic is routed at the domain level (at the level of the Autonomous System (AS)[1]) as it travels from source to destination. Today, users can pick their own ISPs, but once the packets have entered the network, the users have no control over the overall routes their packets take. ISPs make business decisions to interconnect, and technically the BGP routing protocol [33] is used to select the specific route a packet follows. Each domain makes local decisions that determine what the next hop (at the domain level) will be, but the user cannot exercise any control at this level.

This paper argues that it would be a better alternative to give the user more control over routing at this level. User choice fosters competition, which imposes an economic discipline on the market, and fosters innovation and the introduction of new services. An analogy can be seen in the telephone system, which allows the user to pick his long distance provider separately from his (usually monopolist) local provider. Allowing the user to select his long-distance provider has created the market for competitive long distance, and driven price to a small fraction of their pre-competition starting point. The original reasoning about Internet routing was that this level of control was not necessary, since there would be a large number of ISPs, and if a consumer did not like the wide area choice of a given local access ISP, the consumer could switch. Whether this actually imposed enough pressure on the market in the past might be debated. But for the consumer, especially the residential broadband consumer, there is likely to be a very small number of competitive local ISPs offering service. With cable competing only with DSL, the market is a duopoly at best (at the facilities level) and often a monopoly in practice. So in the future,the competitive pressures on the wide area providers will go down.

*Xiaowei Yang was sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0553.

ACM SIGCOMM 2003 Workshops August 25&27, 2003, Karlsruhe, Germany.

[1]In this paper, an AS is also referred to as a "domain". An AS that provides transit service is sometimes called "a provider" or "a service provider."

While it is only speculation, one can ask whether the lack of end to end quality of service in the Internet is a signal of insufficient competitive pressure to drive the deployment of new services. Certainly, there are many consumers who would be interested in having access to enhanced QoS, if it was available end to end at a reasonable price. Such a service might have driven the deployment of VoIP, of various sorts of teleconferencing and remote collaboration tools, and so on. If the consumer could pick the routes his packets took, this might entice some provider to enter the market with a QoS offering, and a set of ISPs might in time team up to make this widely available. But there is no motivation to offer such a service today, since the consumer has no way to get to it. So one can speculate that lack of competition in the form of user-selected routes is one cause of stagnation in Internet services today.

We cannot prove this hypothesis as a business proposition. Only an experiment in the real world–a real market deployment–can reveal what users actually want, what creative providers will introduce, and what might happen to pricing. But this paper takes as a starting point that this experiment is a worthy one, and explores the technical problems that would have to be solved to give users this sort of choice.

We present the New Internet Routing Architecture (NIRA) — an architecture that is designed to give a user the ability to choose domain-level routes. A domain-level route is a sequence of domains a packet traverses, as opposed to a router-level route, which is a sequence of routers a packet traverses. We emphasize domain-level choices rather than router-level choices because user choices are intended to stimulate competition among providers. It does not help to create competition to give a user choices over a domain's router-level routes. A domain may offer this choice to users according to its own policies. NIRA neither mandates a domain to do it, nor prevents a domain from doing it. We focus our discussion on domain-level choices. In the rest of our paper, without explicit explanation, a "route" refers to a domain-level route. The word "a user" in this paper refers to an abstract entity. It is not necessarily a human user. It could be a piece of software running in a human user's computer, making route selections according to user configured preferences. We view it as a rather separate component, and will not discuss the design of it in this paper.

Many design problems arise from enabling this ability. The most obvious one is: how does a user discover a route so that he can use it? The physical structure of the Internet and the transit policies of each domain together determine the set of possible domain-level routes between two end users. Routes, in this sense, change slowly. We sometimes call such routes "static routes." In contrast, the dynamic conditions of a route may change frequently. We use "route availability" to refer to whether the dynamic conditions of a route satisfy a user's requirements, such as whether a route is free from failures, or whether a route has sufficient available bandwidth. Intuitively, if mechanisms for static route discovery are also required to supply route availability information, there will be less variety in design choices. Therefore, we separate the design problem of route availability discovery from that of route discovery.

The next problem to address is route representation. Users must specify their route choices in packet headers. So how should routes be represented?

Finally, in a commercialized Internet, if a provider cannot be properly compensated, it has little motivation to forward packets according to a user's route specification. Thus, the design of an architecture should take into consideration how a provider is compensated if a user chooses to use its services.

We narrow down the solution space by identifying several key design requirements:

- Scalability. With the fast growth of the Internet and the possibility that millions of small devices will be connected to the Internet, we require that no single routing component has bandwidth, memory, or processing power that scales with the number of users on the Internet.
- Robustness. This requirement has two components. First, if a route is unavailable, a user should be able to find an alternate route within a few round trip times if it exists. We think this failover time is reasonable for Internet applications. Second, the failures, mistakes, or malicious behaviors of one routing component should not have a global impact.
- Efficiency. The various overheads added for supporting domain-level route selection should be minimized for common case.
- Heterogeneous user choices. Individual users in the same domain should be allowed to choose different providers. If a local provider connects to multiple wide area service providers, users of the local provider should be allowed to choose wide area providers of their own choices.
- Practical provider compensation. Provider compensation should not require per-packet detailed accounting or micropayment [29]. We think these schemes have little market appeal.

Researchers [35] [44] have proposed using a domain-level link state protocol to distribute the topology information of the Internet and the transit policies of each domain. In these proposals, the border router of a domain or a specialized server in a domain keeps track of the topology information and transit policies of each domain, and makes route selections for users. These proposals do not satisfy our design requirements. To support heterogeneous user choices, it is insufficient to specify transit policies at the granularity of domains. For example, if a local provider interconnects with UUnet, but a user in the local provider's network does not sign a business agreement for UUnet's service, then packets destined to the user cannot be sent through UUnet. Hence, it is necessary to specify transit policies at the granularity of individual users. It is not scalable to keep

this amount of information at a single router or a server. For this reason, we can not adopt the previous proposals.

NIRA divides the task for route discovery between the two halves, source and destination, so that each half only needs to know his part of the network as the network grows. Each user discovers topology information on domains that provide transit service for him according to contractual agreements. We call those domains a user's providers. The source retrieves topology information on the destination's providers on demand. Combining these two pieces of information, the source specifies a route to reach the destination. This route discovery process does not require a single router or a server to keep the topology information and transit policies for every user. The tradeoff is that users may spend extra round trip times for route retrieval. Since we deliberately separate the dynamic route availability discovery from route discovery, routes retrieved on demand can be cached for later use. We expect that caching will amortize the overall cost on round trip times.

In NIRA, reactive notification and proactive notification are used in combination to facilitate route availability discovery. If a route is unavailable when a user tries to use it, a router sends back a reactive notification to the user, or a user detects it via time-out. A user may also receive proactive notifications on the dynamic conditions of domains that provide transit service for him.

In the current Internet, to send packets to another user, a user only needs to find the address of the destination user, and puts his address and the destination address in the packets' headers. Compared to the current Internet, an architecture that supports domain-level route selection is likely to increase the connection setup overhead for route discovery and the packet header overhead for route representation. To minimize the overheads, we introduce an addressing scheme such that a common type of domain-level route can be represented by a pair of addresses. When sending a packet, in most cases, a user only needs to pick a source address and a destination address, and puts the two addresses in a packet header.

NIRA suggests a practical provider compensation scheme. Providers decide whether they would provide transit service for a user based on contractual agreements between the user and the providers, and may install policy filters to prevent illegitimate route usage. Providers will be properly compensated by billing their customers based on the contractual agreements.

In the following sections, we first compare NIRA to related work. Then we will describe in detail the design of NIRA. We start with the description of NIRA's addressing scheme. We then discuss route discovery, route availability discovery, and provider compensation.

2. RELATED WORK

We organize the related work into three parts: related routing architecture proposals, route representation schemes, and current route selection technologies. We discuss them in turn.

2.1 Routing Architecture Proposals

Nimrod [10, 36, 32] proposed a hierarchical map distribution based routing architecture. It did not address how to fit the design into the policy-rich inter-domain routing environment. As a result, no inter-domain routing protocol has ever evolved from Nimrod. NIRA's addressing, forwarding, and topology information propagation schemes directly incorporate the contractual relationships between different parities of the Internet.

The Inter-domain Policy routing (IDPR) [35] protocol was designed according to the original policy routing ideas described by Clark [13]. IDPR is a link state based protocol. Each AS has a route server that keeps a link state database for the domain-level Internet topology and computes policy routes on the request of hosts. The IDPR proposal does not handle destination policies. NIRA does not require the presence of per domain route servers and domain-level topology information does not flow globally in NIRA.

In the Scalable Inter-Domain Routing Architecture [17], the inter-domain routing contains both a hop-by-hop node routing (NR) component and a source demand routing (SDR) component. The NR component installs default forwarding paths, and is similar to BGP. The SDR component is similar to IDPR. The authors in [42] suggested two approaches for SDR to discover routes: Routing Information Base (RIB) query and Path Explorer. RIB query utilizes the RIBs constructed by the NR component. Path Explorer is the technique that a source requests an on-demand route computation to reach a destination. Intermediate nodes do not exert their route selection preferences but rather propagate routes that obey their transit policies to the source. Recent research has shown quite a few problems of the NR component in the current routing system [26], [25], [22], [43], [27]. Inserting another component into the routing system is likely to cause more problems. NIRA has different route discovery mechanisms and does not require a separate routing component to establish default routes.

There are two well-known hierarchical routing schemes: the cluster-based hierarchical routing proposed by Kleinrock et al [24] and the landmark routing proposed by Tsuchiya [40, 41]. Both schemes do not address inter-domain policy routing issues and are best suited for intra-domain routing.

IPNL [20] is a recent architecture proposal. It is designed to solve the IPv4 address depletion problem and the routing scalability problem, and does not address user choices. TRIAD [12] is an Internet architecture that aims at explicit support for a content layer and has different goals from NIRA. The Wide-area Relay Addressing Protocol (WRAP) in TRIAD adds a layer on top of IPv4 to support path-based addressing, while NIRA is designed to handle IP layer routing and addressing issues. The Feedback Based Routing system [44], like NIRA, separates routing information into topology information and dynamic information. A domain's access router keeps the topology information and transit policies of the entire Internet, and performs

route selection and path monitoring. User choices are not supported.

2.2 Route Representation Schemes

This category of related work focuses on how to represent provider-level routes and how the corresponding forwarding mechanism works, but the work does not address how routes are discovered and how providers are compensated.

Dated back to the seventies, Sunshine [38] discussed a route specification scheme using a sequence of local port numbers (termed as "switch addresses" in the paper) traversed by a packet to reach a destination. Both IPv4 [31] and IPv6 [14] contain specifications for source routing. A number of IPng proposals, including Pip [18], SIP [16], TUBA [23], SIPP [19], all provide loose source routing capabilities.

NIRA's route representation scheme differs from the previous approaches in that it leverages the policy-level structure of the Internet, and uses a pair of addresses to represent a common type of domain-level route. Similar to previous approaches, NIRA uses multiple addresses to represent more complicated routes.

2.3 Current Route Selection Technologies

As there is no global framework supporting route selection, some local and small scale solutions are created to satisfy users' needs for selecting routes. There are both commercial products and academic efforts.

Several companies, including netVmg, Opnix, RouteScience, Sockeye [1, 2, 3, 4, 5], emerged to offer "route control" services or products. "Route control" technologies can help a multi-homed site to choose the access link to its providers dynamically. The control is limited to dynamically selecting the next hop provider for outbound traffic only. The limitation of these technologies is that they cannot choose beyond the next hop, and usually cannot be afforded by individual consumers.

Another effort for providing alternative routes that are different from those determined by the underlying routing architecture is to build overlay networks. An overlay network exploits a small group of end hosts to provide packet forwarding and/or duplication service for each other [6] [39] [34]. The limitation of the overlay networks is that they are not ubiquitous. Only nodes on the overlay network can control their paths by tunneling traffic through other nodes on the overlay network. It is unlikely that overlay networks can scale up to include every user on the Internet. Besides its limited scope, an overlay architecture is less efficient than source routing. An overlay path may traverse duplicate physical links.

3. NETWORK MODEL, ADDRESSING, ROUTE REPRESENTATION AND FORWARDING

3.1 The Network Model

NIRA's addressing and route representation scheme relies on the policy-level structure of the Internet. A domain decides whether it will provide transit services between a pair of its neighboring domains, or for certain address prefixes, based on its business relationships with other domains. A domain's decisions are reflected in the domain's transit policies. There are three most common business relationships [21] between two interconnecting domains: the provider-customer relationship, the peering relationship, and the sibling relationship[2]. A typical domain-level route is said to be "valley-free." That is, a packet sent by an end user is first "pushed" up along its provider chain, and then flows down along the receiver's provider chain. There exists a densely connected region of the network [37] where packets can not be further "pushed" up. We call this region the *Core* of the Internet. Outside the *Core*, the interconnection of the network is relatively sparse, where the dominant interconnection relationship is the provider-customer relationship [21]. A packet does not have to be pushed all the way up to the *Core* to reach its destination. Alternatively, a low-level peering link may connect the sender's provider chain and the receiver's provider chain. A packet can take the short-cut to reach its destination. This model is consistent with research results on the Internet's structure [21] [37].

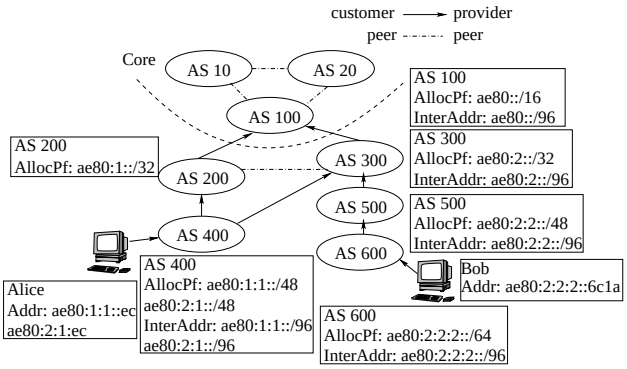

Figure 1: An example of NIRA's network model and addressing scheme.

Figure 1 illustrates the network model. AS 10, AS 20, and AS 100 are providers in the *Core*. Other ASes are outside the *Core*. Routes `400 200 100 300 500 600` and `400 200 300 500 600` are both valley-free routes.

3.2 Addressing

NIRA uses a hierarchical provider-rooted addressing scheme to reduce the overhead for constructing and representing a route. A domain inside the *Core* is called a "top-level provider." Each top-level provider will be allocated a globally unique address prefix. A top-level provider then allocates address prefixes from its address space to its customers. The customers in turn allocate these prefixes to their customers. This address allocation process happens recursively at each domain: when a domain receives an address prefix from one of its providers, it allocates an address prefix from the address space of this prefix to its customers if it has any. The

[2]In a sibling relationship, two domains provide mutual transit service for each other.

process of allocating an address prefix to a customer can either be automated, or be manually configured by network operators.

We use the phrase "route segment" to refer to a partial route, as opposed to a route that connects two end users. NIRA's addressing scheme ensures that for any route segment that starts at a domain, and reaches the *Core*, and consists of only customer-provider interconnections, there is an address prefix that uniquely identifies it. If a domain has multiple route segments to the *Core*, it has multiple address prefixes.

The current design of NIRA assumes that all addresses have a fixed length L, and $L = 128$ bits. An address is a concatenation of an inter-domain address and an intra-domain address. The inter-domain addresses of nodes in the same domain have the same length. Thus, a node could keep a unique intra-domain address in all its addresses, which simplifies intra-domain communication. An inter-domain address is formed by padding a prefix allocated to a domain with zeros to the length of its inter-domain address.

Figure 1 shows an example of NIRA's addressing scheme. Addresses are represented using IPv6 conventions [15], where "::" represents variable number of zeroes. The length of an address prefix or an inter-domain address is specified after "/." In this example, the top-level provider AS 100 is allocated a unique prefix `ae80::/16`, and it has an inter-domain address `ae80::/96`. AS 100 allocates prefix `ae80:1::/32` to its customer AS 200 and prefix `ae80:2::/32` to its customer AS 300. AS 200 and AS 300 allocate address prefixes to their customers in turn. AS 400 has two route segments to the *Core*, `400 200 100` and `400 300 100`, and therefore has two address prefixes, `ae80:1:1::/48` and `ae80:2:1::/48`. *Alice* is a host in AS 400, and has addresses `ae80:1:1::ec` and `ae80:2:1::ec`; *Bob* is a host in AS 600, and has the address `ae80:2:2:2::6c1a`.

3.3 Route Representation

We utilize the feature that an address prefix uniquely identifies a route segment to develop a novel route representation scheme. Valley-free routes that do not go across a low-level peering interconnection can all be represented by a pair of source and destination addresses. Each of these routes consists of two route segments. One route segment is the chain of the providers that allocate the source address; the other is the provider chain that allocates the destination address. The two segments either meet at a common provider, or both reach the *Core*. For example, in Figure 1, the pair of addresses `ae80:1:1::ec` and `ae80:2:2:2::6c1a` uniquely identify the domain-level route, `400 200 100 300 500 600`, between hosts *Alice* and *Bob*. Routes that can be represented in this fashion are called canonical routes. Other routes, such as route `400 200 300 500 600`, are called non-canonical routes.

To forward a packet with such a route representation, the forwarding algorithm needs to inspect not only the destination address, but also the source address. From the destination address, a router is able to tell whether the destination domain has been reached or not. If it has not been reached, then the router decides whether the turn-around point has been reached or not based on the source and the destination addresses. If the two addresses share a common prefix that belongs to the current domain's address space, then the turn-around point is reached; or if the two addresses do not share a common prefix, but the the current domain is in the *Core*, then the turn-around point is also reached. Before the turn-around point, the packet will be forwarded "up-hill" according to the source address; after it, the packet will be forwarded "down-hill" according to the destination address. We term this forwarding algorithm as "valley-free" forwarding.

A non-canonical route representation requires more addresses. Consider the non-canonical route `400 200 300 500 600` in Figure 1 between *Alice* and *Bob*. The turn-around point of the route is at AS 200, but the combinations of *Alice*'s addresses and *Bob*'s address can only represent routes with turn-around points either at AS 100 or at AS 300. If we stick with a stateless route representation scheme, i.e. no virtual path setup, it is necessary to use more addresses to represent these routes. In the worst case, a non-canonical route is represented by a sequence of addresses with each address denoting a domain in the route. An address with an all-zero intra-domain portion is used as the all-router intra-domain anycast address, which can represent an intermediate domain. However, in most cases, a non-canonical route also has a compact representation. If the route contains a valley-free segment that does not go through a low-level peering link, the entire route segment could be represented by two addresses, with one representing the starting domain and the other the ending domain[3]. For example, the route segment `300 500 600` in the route between *Alice* and *Bob*, can be represented as `ae80:2:: ae80:2:2:2::6c1a`. The complete representation for the route `400 200 300 500 600` is `ae80:1:1::ec ae80:1:: ae80:2:: ae80:2:2:2::6c1a`.

A forwarding algorithm that works with such a representation needs to inspect two consecutive addresses for "valley-free" forwarding. However, not every two consecutive addresses in a non-canonical route representation denotes a valley-free segment. The two addresses `ae80:1:: ae80:2::` do not mean to represent `200 100 300`. Instead, they should represent `200 300`. A different forwarding algorithm, which interprets the address `ae80:2::` as the next domain to reach, rather than the end of a valley-free segment, should be used.

We add an extra field, Forwarding Indicator, denoted by F to inform a router on how to interpret two adjacent addresses. Different values of F indicate whether two addresses represent a valley-free route, where the second address is the end of the valley-free route, or two addresses represent a domain-level interconnection, where the second address is the next domain to reach (which we call next-domain forwarding). A route representation in a packet header is shown in Figure 2.

A canonical route representation only contains fixed fields, where a_s and a_d are the source address and the

[3]The address for the starting domain or the ending domain may be omitted if that domain is in the *Core*

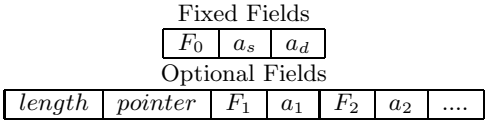

Figure 2: A route presentation in a packet header is shown.

destination address. A non-canonical route representation contains optional fields. Fields a_s and a_d contain the first two addresses, and F_0 specifies the forwarding algorithm between them. In the optional fields, *length* denotes the total number of addresses in the optional fields; *pointer* points to the current F_i. Field a_i is the ith address in the optional field. Preceding each a_i is F_i.

Forwarding decisions are made upon the values in fields F_0, a_s, and a_d in a packet's header. When a packet has reached the current a_d, the field F_c pointed to by *pointer* is moved to the field F_0; a_c, which is the address following F_c, is moved to a_d; a_d is shifted to a_s; values in F_0 and a_s are preserved at F_c and a_c. *Pointer* is moved to the next F_i. Return routes can be generated by reversing the fields a_s and a_d and the sequence of (F_i, a_i) pairs.

Suppose $F = 0$ indicates "valley-free" forwarding and $F = 1$ indicates next-domain forwarding. As an example, the representation for route `400 200 300 500 600` is shown in Figure 3.

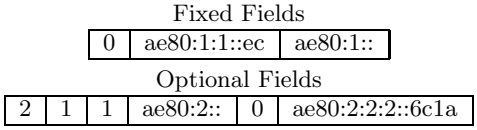

Figure 3: The representation for route `ae80:1:1::ec ae80:1:: ae80:2:: ae80:2:2:2::6c1a` is shown.

When a packet with this header arrives at AS 200, the representation will change to what is shown in Figure 4. The Forwarding Indicator becomes 1. A router will correctly forward the packet to the next domain `ae80:2::`.

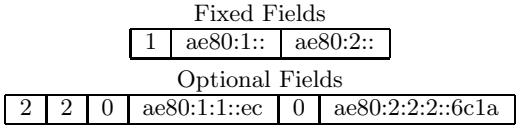

Figure 4: When a packet reaches AS 200, the modified representation for route `ae80:1:1::ec ae80:1:: ae80:2:: ae80:2:2:2::6c1a` is shown.

It is worth noting that the non-canonical route representation shares similarity with IP's source routing option, where multiple addresses are used to encode a source route. Source routing is discouraged in today's Internet because of the perceived security reason [8]. That is, without source routing, though an attacker can spoof a source address, replies will be sent back to the real host with that address. This makes impersonation difficult. However, with source routing, an attacker can spoof a source address and insert his address in the middle of a source route. As replies will follow the reversed source route, an attacker is able to intercept them. If address-based authentication is used (an address is associated with the identify of the host with that address), the attacker is able to impersonate the host with the spoofed address. We argue that address-based authentication provides weak security guarantees, because it assumes that every component on the path from the sender to the receiver is trustable. As more secure end-to-end authentication tools are widely available today, address-based end-to-end authentication should become obsolete, and should not be the obstacle that impedes the deployment of source routing.

3.4 Establishing Forwarding States

Outside the *Core*, border routers exchange, but do not propagate, information on address reachability to establish forwarding states. To ensure that a packet with a canonical route representation is forwarded along the address allocation paths of a_s and a_d, a customer domain announces to a provider only the address prefixes allocated by the provider. A domain announces all or selected set of its addresses to non-provider domains.

Inside the *Core*, a top-level provider must have forwarding states for address prefixes of other top-level providers. A top-level provider is not necessarily connected to every other top-level provider. Thus, forwarding states for address prefixes of the top-level providers will be established by a dynamic inter-domain routing protocol, such as BGP. If a packet has a canonical route representation, the route a packet traverses between two providers in the *Core* will be chosen by the routing protocol. We call a contiguous region of the Internet that runs an inter-domain routing protocol "a routing region." The most important property of a routing region is that from any domain in the routing region, there is at least one policy-allowed route to reach any other domain in the routing region without going outside the routing region. Our assumption is that the number of domains in the *Core* is small compared to that of the Internet. Running a dynamic routing protocol in a small region will not cause any performance and scaling problem.

3.5 Optimizations on Route Representation

The design choices of provider-rooted addressing and making the *Core* a routing region make an efficient route representation scheme for common case. We assume that in practice, the business relationships between ASes in the *Core* are complicated. ASes may belong to different countries and have sophisticated commercial contracts. Therefore, in most cases, we expect that users do not care to choose domain-level routes between two providers in the *Core*, or are not allowed to choose due to the complications of provider compensation problem, or cannot choose because there is only one policy-allowed route. Thus, a pair of source and destination addresses are sufficient to represent a user selected route in most cases. In cases where a user does

not care to choose a particular domain-level routes, such as for connections with a few packets, this route representation is also efficient. However, a non-canonical route still requires more than two addresses to represent. To further reduce the overhead, we introduce two optimizations.

3.5.1 Extended Addressing

We can extend NIRA's addressing scheme to make more routes canonical. A global unique prefix can be allocated to a peering interconnection. Each participating AS will be allocated a prefix from the address space of this prefix. Each AS allocates address prefixes to its customers using the same addressing scheme as described in Section 3.2. Thus, a valley-free route that contains the peering interconnection can also be represented by a pair of addresses.

To reduce the number of addresses a node has, if several ASes peer with each other, the entire peering group is allocated one global unique prefix, rather than one prefix per pair. Similarly, in a sibling interconnection, an AS allocates an address prefix to its sibling from each prefix allocated to the AS by its providers. An AS that does not connect to the *Core* can get its own address prefix, and allocate it to its customers. More addresses make more valley-free routes canonical.

3.5.2 Multiple Routing Regions

If in some region of the Internet, there are many non-provider-customer interconnections, and getting more addresses is undesirable, we can demarcate the region as a routing region. For example, a few ASes that voluntarily provide transit service for each other may form a routing region. Thus, if a user is satisfied with the route chosen by the underlying routing protocol, he does not have to explicitly specify the route segment in the routing region.

3.6 Feasibility of NIRA's Addressing Scheme

NIRA's efficient route representation scheme is brought about by its addressing scheme. However, there are two concerns about NIRA's addressing scheme. First, how many addresses will a node have? If this number is unreasonably large, users with limited memory and processing power are not able to handle their own addresses. Second, in the address allocation process, when a prefix reaches an edge domain, how many domain-level hops has it taken? If this number is large, it is impossible to use a fixed-length address format. However, we prefer a fixed-length address to a variable length one because of its simplicity.

In theory, both numbers could be quite large. Suppose there are d levels of hierarchies. At each level, a domain has k higher level providers. Then by NIRA's addressing scheme, an edge domain has k^{d-1} address prefixes, which grows exponentially with d. Our hypothesis is that in reality, financial constraints will keep an AS from having an unreasonably large number of providers and will also keep the depth of provider hierarchy shallow. Thus, NIRA's addressing scheme should prove practical.

We evaluated our hypothesis using Internet topology deduced from BGP table dumps. We obtained several BGP table dumps dated from November 1997 to January 2003 from the Route Views server [28]. Using the AS path attributes, we derived the AS-level Internet topology. We used a heuristic algorithm similar to those in [21] [37] to infer the provider-customer relationships between ASes and to identify the *Core* of the Internet. The heuristic algorithm is based on the observation that in the domain-level Internet topology, a provider network usually has a larger degree than its customer networks. The *Core* C is initialized with the AS that has the largest degree. Let the current number of ASes in C be $|C|$. The classification algorithm iterates through all ASes in the decreasing order of their degrees. For each AS D , the algorithm counts the number of edges $Nconn$ between D and ASes in C. In each iteration, the AS with the largest $Nconn$ is added into C. The classification algorithm stops when the largest $Nconn$ in one iteration is less than $\alpha * |C|$, where α is a tunable parameter between 0 and 1. When $\alpha = 1$, the *Core* is a complete graph. Provider-customer relationships can be approximately deduced from the following assumptions: a valid domain-level path is valley-free; domains in the *Core* are top-level providers; a provider has a larger degree than its customers. From our inferred provider-customer relationships, we apply our addressing schemes, and collect data.

We show the results when $\alpha = 1$. More than 80% of interconnections are classified as having a provider-customer relationship; and more than 90% of ASes are allocated at least one prefix[4]. Our results show that the number of ASes in the *Core* over a period of six years is very small, at most 15. When relaxing the inference algorithm with $\alpha = 0.1$, we find the size of the *Core* to be between 56 and 123. This validates the assumption that the size of *Core* is small compared to that of the Internet and does not grow much.

Figure 5 shows the mean, median, and largest number of prefixes an AS should have been assigned in NIRA over the years. In the worst case, an AS may have more than three hundred address prefixes. On average, an AS has fewer than 20 prefixes. For a unique prefix, Figure 6 shows the mean, median, and largest number of AS hops the prefix has traversed. The mean and median are both around 4; and the largest is at most 11. These numbers are reasonable from 11/1997 to 01/2003, even during the booming period of the Internet. We see no significant growth trend for these numbers. We think our experiments suggest that NIRA's addressing scheme is practical.

4. ROUTE DISCOVERY

A user can use any general mechanism to discover routes. NIRA provides two infrastructure services to as-

[4]Due to Route Views server's limited vantage points, we can not observe all AS-level connections from the routing table dumps. An AS is not allocated a prefix if it does not have a path to the *Core* that consists of only customer-provider links in our inferred topologies.

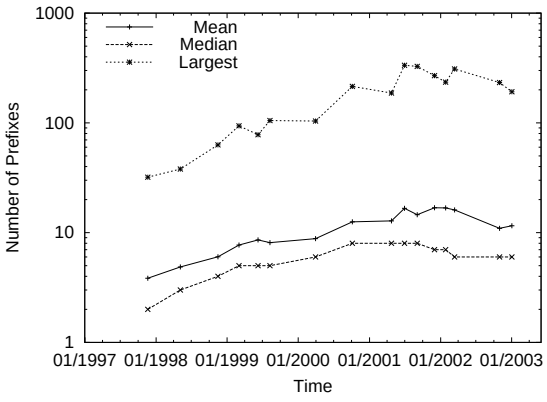

Figure 5: The mean, median, and largest number of prefixes an AS has.

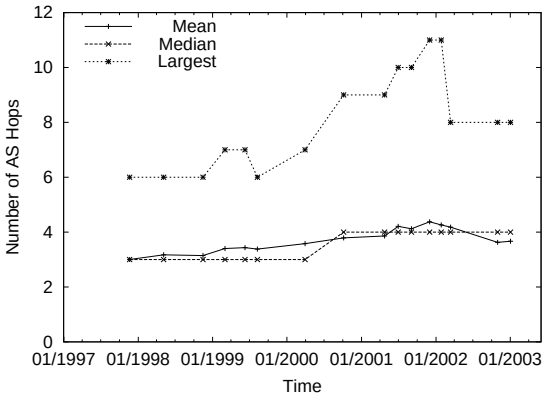

Figure 6: The mean, median, and largest number of AS hops a unique prefix has traversed.

sist route discovery. The first service, a policy-constrained topology information protocol, helps a user to discover the topology information on domains that provide transit service for the user. Usually, a user can use this information to find his route segments that reach the *Core*. The second service, a name-to-route resolution service, helps a user to solve the bootstrapping problem — how to send the first packet to another user. This service assumes that a user knows the name of his correspondent, and returns the user the topology information on route segments of his correspondent. The route segments of a user and his correspondent should intersect to render a route between the two.

4.1 The Topology Information Propagation Protocol (TIPP)

TIPP propagates to a user his inter-domain addresses, the route segments associated with these addresses, and possibly other information on domains that provide transit service for the user.

With TIPP, a domain summarizes its interconnections with its neighboring domains and sends the information to its neighbors. Domains also propagate interconnection information for each other. Basic TIPP messages are triggered by topology or transit policy changes, and do not include the dynamic conditions of the interconnections.

The propagation and formation of TIPP messages are subject to policy constraints. In particular, TIPP messages do not propagate globally. A domain does not tell any neighbor the interconnections between itself and its customers, and a domain does not propagate messages it receives from its customers regarding interconnections within its address space. For a domain D, if it provides transit service between its neighbors N_1 and N_2, then it tells N_1 about the interconnection between itself and N_2, propagates messages heard from N_1 to N_2, and vice versa; otherwise, it does not.

It is optional for domains inside a routing region to run TIPP between their neighbors. If domains in a routing region do not run TIPP, then any two domains in a routing region appear to domains outside the routing region as if they have a virtual interconnection.

It can be verified that these policy constraints ensure that only messages related to domains on a user's providers are propagated to the user, and therefore, keep the bandwidth overhead low and the number of states maintained by a user small. A user assembles TIPP messages to form a partial view of the Internet topology. Figure 7 depicts *Alice*'s view and *Bob*'s view of the topology shown in Figure 1 respectively[5]. Had we chosen to use a link-state protocol, both of them would have seen the complete topology.

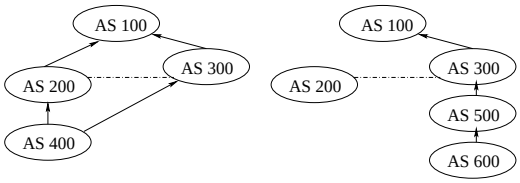

Figure 7: *Alice*'s view (left) and *Bob*'s view (right) of the topology shown in Figure 1.

From the discussion in Section 3.6, a domain has on average fewer than 20 prefixes. Each prefix on average has an allocation path of length 4. Assume all prefixes are allocated from disjoint paths. An average user may receive TIPP messages about 80 interconnections. If there are 250 bytes of data about each interconnection, a user needs 20KB of memory to store the information, which is unlikely to be a problem for the average Internet user.

4.2 The Name-to-Route Resolution Service (NRRS)

A user can utilize NRRS to discover the other user's route segments. NRRS is designed as a distributed

[5]The example we show here assumes providers in the *Core* have the policy not to propagate the topology information in the *Core* to their customers.

name lookup service. A user stores his route segments at a designated server. We call the servers that store such information "route servers." Inspired by the success of DNS [30], route servers are organized according to a hierarchical namespace.

The bootstrapping mechanism and the lookup process of NRRS are similar to those of DNS. A resolver is hard-coded with the route segments of the roots. A user is hard-coded with the route segments of his resolvers. At each level of the resolution, the route segments of route servers that are in charge of a lower-level namespace is returned. The lookup process stops when the route segments regarding the queried name is returned.

Unlike in DNS, where the IP addresses of the root DNS servers do not change when the topology of the Internet changes, the route segments regarding the roots may change. For this reason, we require that the roots reside in the *Core*, as the addresses of a server in the *Core* are resistant to topology changes. A resolver can use any of its addresses that are allocated from the *Core* as a source address, plus the address of a root server, to form a canonical route to reach the server.

In the simplest case, a user only has canonical route segments. Thus, all a user needs to store at his route servers are his addresses. In general, a user may have non-canonical route segments. However, a user does not have to store them at his route servers, if his canonical route segments have sufficient redundancy. Once a user sends the first packet to another user, the two users may exchange the entire set of route segments, and choose a route they both like.

4.3 Route Records Update

When the domain-level topology changes, a user's addresses or route segments may change. These changes will be propagated to a user via TIPP. It is up to users to decide how to update their records. NIRA does not restrict how updates should be done. Stale information of a user may make him unreachable, but does not affect the reachability of other users. It is possible that a single topology change may affect a large group of users, e.g., when a second-level provider changes its top-level providers, the customers of the second-level provider should all change their addresses. We admit that this coupling is inconvenient, but it is the logical consequence of any topology dependent addressing scheme [40] [24]. However, we do not think this coupling effect would cause significant problems for two reasons.

First, the static Internet topology changes at a quite low frequency. According to the study conducted by Chen et al [11], the AS birth and death rate of the Internet is less than 15 per day, the AS-level link birth and death rate is less than 50 per day. Only those changes would affect a user's addresses or route segments. Their study also shows that most of the new or dead ASes are of degree 1 or 2, thus probably being edge ASes. Hence, the changes are likely to affect just users in those domains. Second, static topology changes are caused by the changes in business relationships and will happen in a controlled manner. Network administrators and users could deploy creative scheduling algorithms to reduce the service disruption and route server update overhead. A grace period may be granted before a provider cuts off its service completely so that a user has sufficient time to update its route server records. Randomized algorithms may be used to prevent users from simultaneously update their route server records, and avoid the overloading of the network and route servers.

5. ROUTE AVAILABILITY DISCOVERY

In NIRA, the architectural level support for route availability discovery is a combination of reactive notification and proactive feedback. In addition to sending basic TIPP messages carrying its static interconnection information, a domain may send advanced TIPP messages that include dynamic information of its interconnections. A user is proactively notified of the dynamic states concerning his providers via these TIPP messages. Therefore, when initiating a connection, the user knows which of his route segments are available. To prevent a user from receiving a large number of TIPP messages, a domain may use standard rate limiting techniques to suppress excessive TIPP messages, or only propagate TIPP messages regarding the changes of a particular dynamic attribute, such as link up and down.

As TIPP messages do not propagate globally, a user in general does not know the availability of the route segments of other users. When a user wants to send packets to another user, it is possible that he chooses a route segment of the other user that is unavailable. In NIRA, if a router detects that a route specified in a packet header is unavailable, the router must try its best to send a control message to inform the original sender. Such a control message may include reasons why the route is unusable. When a user receives such a reactive notification, as he is aware of both his route segments and those of the other user that he is trying to contact, he could switch to an alternate route on the order of a round trip time. In this case, NIRA enables fast route fail-over. In cases where a router is unable to send a failure notification, e.g., a router is overloaded, users shall use timeout to detect route failures. The fail-over time then depends on the timeout parameters.

The combination of proactive notification and reactive notification reduces the dynamic routing information of which a user needs to keep track. However, reactive notification may increase connection setup time when user selected routes are unavailable. In principle, if routers use local repair mechanisms to send packets over alternate routes, connection setup time will be shortened. For intra-domain routes, routers shall always use such mechanisms for rapid failover. Failures that invoke reactive notification are those that result in inter-domain disconnections. Since a user has expressed his domain-level route choices in a packet header, and an intermediate route does not know the user's route preference, which is probably related to a user's financial considerations, it is best for the user to decide on an alternate route. So in NIRA, we prefer to use reactive notification to local repair for inter-domain fail-

ures. Users shall cache states for recently used routes and avoid using unavailable routes for a new connection. We expect that the amortized set up time will be reduced by caching. Besides mechanisms provided by NIRA, users or service providers can use additional mechanisms such as probing or sending proactive notification to specific users to facilitate route availability discovery. For instance, a local provider may offer a route monitoring service to its users. The provider runs a server that actively probes the dynamic attributes of popular routes and provides the timely information on route availability to its customers.

6. PROVIDER COMPENSATION

Provider compensation problem is important because providers have control over various network resources, and if they can not benefit from giving a user the ability to choose from multiple routes, it is unlikely that a provider will honor a user's choices. In the current Internet, a provider is paid by its directly connected customers. BGP provides a good technical support for this provider compensation model. With BGP, each domain can only select the next hop. The common BGP policies state that a domain prefers a customer route to a provider or a peer route, as a customer route potentially brings revenue to the domain; a domain prefers a peer route to a provider route, as a peer route usually involves no payment; the least preferable route is a provider's route, as it costs the domain money.

New compensation models are required to ensure that a provider will be compensated properly when users are able to select routes. The high level principle is that users are restricted to choose from what they have agreed to pay for. We assume a sender and receiver joint payment model, whose extremes include the one-end-pays-all model.

We discuss two possible provider compensation models. Both require that users have contractual agreements with providers before using their service, and therefore, do not need micropayment.

6.1 Direct Business Relationships

The first model is similar to that of today's Internet. Contractual agreements are negotiated between directly connected entities. However, the agreements will take into account the cost for allowing users to choose different routes. Providers may monitor the route usage of a customer and charge a customer differently based on what routes he chooses to use. In Figure 1, for the same amount of traffic, AS 400 may charge *Alice* more if she chooses to use the route `400 200 100 300 500 600` to reach *Bob* than if she chooses to use `400 200 300 500 600`.

A user may try to use an illegitimate route that violates a domain's transit policies. In Figure 1, *Alice* may try to use a route fragment `400 200 300 100`. AS 300 shall not forward packets with such route fragments because neither AS 200 nor AS 100 is going to pay for the service. In direct business relationships, as a domain's transit policy is usually expressed as whether a domain provides transit service between two neighbor domains, providers could use physical security to prevent such misuse as it is done today. If a packet comes from an interface connecting to a neighbor domain, a provider could assume that the packet comes from the neighbor domain. A border router first verifies whether the source address in field a_s matches the interface a packet comes from and then checks the packet header against its policy filters. A policy filter is of format `(`a_s`, `a_d`, action)`. If a packet header matches a policy filter, a domain will take the corresponding action. If the action is "forward", the packet will be forwarded according to the domain-level route specified in the packet header. Policy filter checking is a simplified type of packet classification. Recent research results [7] show that it can be done at a high speed.

6.2 Indirect Business Relationships

The direct business relationships are quite restricted, thus preventing some service models from existence. We imagine that in NIRA, business relationships can be extended beyond directly connected entities. Users may negotiate business relationships with non-directly connected providers.

In indirect business relationships, how to prevent route misuse becomes a challenging problem. Packets coming from the same adjacent domain may be subject to different transit policies. Thus, a provider can not tell what polices to apply to a packet based on from which domain the packet comes. Instead, a provider needs to tell whether the packet is sent to or from one of its customers. The need to authenticate the sender or the receiver of a packet at the packet forwarding time is not unique to NIRA. Even in today's Internet, an overlay provider, or a second-hop provider that wants to sell QoS to a customer, also has this requirement. As this problem is general enough to become a separate problem, we do not discuss it in detail in this paper. Any solution to this problem could be plugged into NIRA to prevent unauthorized route usage.

6.3 Financial Risks of Exposing Routes

In NIRA, we assume a joint payment model at the network layer, though cost could be shared differently at higher layers. In this payment model, it is difficult for a receiver to avoid being sent unwanted traffic. This problem prevails in today's Internet. The manifestations include SPAM and DoS attacks. However, in today's Internet, an edge domain has control over to which provider to announce its address prefixes using BGP. Therefore, the domain could control from which provider its incoming traffic comes. With NIRA, a receiver exposes its route segments and its route preference via its route servers. A non-cooperative sender may override a receiver's route preference and intentionally send large volume of traffic over an expensive route to a receiver. This non-cooperative behavior of the sender could cause a financial loss to the receiver.

This financial risk is the unfortunate side effect of user empowerment. However, there are two possible counter measures. First, with NIRA's canonical route representation, the source address field of a packet reveals the

network location of the sender, and the forwarding algorithm depends on the correctness of the source address. This prevents a user from spoofing an arbitrary source address. If a non-cooperative sender has to reveal his real network location in order to send packets to the victim receiver, this requirement may discourage him from being non-cooperative in the first place. Moreover, if the non-cooperative sender cannot fake his network location in a packet header, the victim receiver could ask his upstream providers to filter out the sender's traffic. Second, as we have mentioned, a user can store at his route servers only part of his route fragments, and keep more expensive route fragments private. He can reveal the private route fragments to those users he trusts after they have exchanged packets via the public route fragments.

Briscoe [9] also suggested a solution to this problem. The solution says that it is customary for both the sender and the receiver to pay, but the ultimate liability should remain with the sender. Any receiver could dispute his payment unless the sender had proof of a receiver request.

We do not claim that the above measures could eliminate the non-cooperative behaviors of users. We think these measures will reduce the financial risk of a receiver to minimal.

7. CONCLUSION

This paper describes the design of NIRA, a new Internet routing architecture. NIRA aims to provide a user the ability to choose domain-level routes. To the best of our knowledge, this is the first work that addresses a full range of the design challenges for supporting domain-level route selection, which include route discovery, route availability discovery, route representation, and provider compensation. We identify several key design requirements for NIRA, which consist of scalability, robustness, efficiency, heterogeneous user choices, and practical provider compensation, and have made design choices to meet these requirements.

In particular, NIRA separates the route availability discovery from route discovery to achieve scalability and robustness, and uses hierarchical provider-rooted addressing for efficiency. NIRA does not require per-packet detailed accounting, or micropayment to compensate providers.

The work presented in this paper is preliminary. We are in the process of implementing a prototype to validate the properties of NIRA.

8. ACKNOWLEDGMENTS

The author would like to thank David Clark for his valuable guidance throughout this work; John Wroclawski and Robert Morris for the discussions that greatly helped the work; Nick Feamster, Simson Garfinkel, Ben Leong, Jinyang Li, Ji Li, and Archit Shah for reading an early version of the draft; and the anonymous reviewers of SIGCOMM and the FDNA workshop for their comments.

9. REFERENCES

[1] netVmg. `http://www.netvmg.com/`, June 2003.

[2] Opnix. `http://www.opnix.com`, June 2003.

[3] Proficient Networks. `http://www.proficientnetworks.com/`, June 2003.

[4] RouteScience. `http://www.routescience.com/`, June 2003.

[5] Sockeye Networks. `http://www.sockeye.com/`, June 2003.

[6] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proceedings of the 18th ACM Symposium on Operating System Principles*, Banff, Canada, Oct. 2001.

[7] F. Baboescu and G. Varghese. Scalable Packet Classification. In *Proceedings of ACM SIGCOMM*, San Diego, CA USA, Aug. 2001.

[8] S. Bellovin. *Security Concerns for IPng*. Internet Engineering Task Force, Aug. 1994. Informational, RFC 1675.

[9] B. Briscoe. The Direction of Value Flow in Multi-service Connectionless Networks. In *Proceedings of International Conference on Telecommunicatons and E-Commerce (ICTEC'99)*, Nashville, USA, Oct. 1999.

[10] I. Castineyra, N. Chiappa, and M. Steenstrup. *The Nimrod Routing Architecture*. Internet Engineering Task Force, Aug. 1996. Informational, RFC 1992.

[11] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. The Origin of Power Laws in Internet Topologies Revisited. In *Proceedings of IEEE INFOCOM*, New York, NY USA, June 2002.

[12] D. R. Cheriton and M. Gritter. TRIAD: A New Next-Generation Internet Architecture. `http://www-dsg.stanford.edu/triad/triad.ps.gz`, 2000.

[13] D. Clark. *Policy Routing in Internet Protocols*. Internet Engineering Task Force, May 1989. RFC 1102.

[14] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*. Internet Engineering Task Force, Dec. 1998. Draft Standard, RFC 2460.

[15] S. Deering and R. Hinden. *Internet Protocol Version 6 (IPv6) Addressing Architecture*. Internet Engineering Task Force, Apr. 2003. Proposed Standard, RFC 3513.

[16] S. E. Deering. SIP: Simple Internet Protocol. *IEEE Network*, 7(3):16–28, May 1993.

[17] D. Estrin, Y. Rekhter, and S. Hotz. Scalable Inter-Domain Routing Architecture. In *Proceedings of ACM SIGCOMM*, pages 40–52, Baltimore, MD USA, Aug. 1992.

[18] P. Francis. A Near-Term Architecture for Deploying Pip. *IEEE Network*, 7(3):30–37, May 1993.

[19] P. Francis and R. Govindan. Flexible Routing and Addressing for a Next Generation IP. In *Proceedings of ACM SIGCOMM*, pages 116–125, London, UK, 1994.

[20] P. Francis and R. Gummadi. IPNL: A NAT-Extended Internet Architecture. In *Proceedings of SIGCOMM*, pages 69–80, San Diego, CA USA, Aug. 2001.

[21] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking (TON)*, 9(6):733–745, Dec. 2001.

[22] T. G. Griffin and G. Wilfong. An Analysis of BGP Convergence Properties. In *Proceedings of ACM SIGCOMM*, pages 277–288, Cambridge, MA USA, Aug. 1999.

[23] D. Katz and P. S. Ford. TUBA: Replacing IP with CLNP. *IEEE Network*, 7(3):38–47, May 1993.

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

[25] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proceedings of ACM SIGCOMM*, pages 175–187, Stockholm, Sweden, Aug. 2000.

[26] C. Labovitz, G. R. Malan, and F. Jahanian. Internet Routing Instability. In *Proceedings of ACM SIGCOMM*, pages 115–126, Cannes, France, Sept. 1997.

[27] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz. Route Flap Damping Exacerbates Internet Routing Convergence. In *Proceedings of ACM SIGCOMM*, pages 221–233, Pittsburgh, PA USA, Aug. 2002.

[28] D. Meyer. University of Oregon Route Views Project. `http://antc.uoregon.edu/route-views/`.

[29] S. Micali and R. L. Rivest. Micropayments Revisited. In B. Preneel, editor, *Proceedings of the Cryptographer's Track at the RSA Conference*, LNCS 2271, pages 149–163. Springer Verlag CT-RSA, 2002.

[30] P. Mockapetris and K. J. Dunlap. Development of the Domain Name System. In *Proceedings of ACM SIGCOMM*, pages 123–133, Stanford, CA USA, Aug. 1988.

[31] J. B. Postel. *Internet Protocol*. Internet Engineering Task Force, Sept. 1981. Standard, RFC 791.

[32] R. Ramanathan and M. Steenstrup. *Nimrod Functionality and Protocol Specifications, Version 1*. Nimrod Working Group, Mar. 1996. Internet Draft, Expires 30 August 1996, `http://ana-3.lcs.mit.edu/~jnc/nimrod/prospec.txt`.

[33] Y. Rekhter and T. Li. *A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force, 1995. Draft Standard, RFC 1771.

[34] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: Informed Internet Routing and Transport. *IEEE Micro*, 19(1):50–59, Jan. 1999.

[35] M. Steenstrup. *An Architecture for Inter-Domain Policy Routing*. Internet Engineering Task Force, June 1993. Proposed Standard, RFC 1478.

[36] M. Steenstrup. *A Perspective on Nimrod Functionality*. Internet Engineering Task Force, May 1995. Internet Draft, Expires November 1995, `ftp://ftp.bbn.com/pub/nimrod-wg/perspective.ps`.

[37] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet Hierarchy from Multiple Vantage Points. In *Proceedings of IEEE INFOCOM*, New York, NY USA, June 2002.

[38] C. A. Sunshine. Source Routing in Computer Networks. *ACM Computer Communication Review*, 7(1):29–33, Jan. 1977.

[39] J. Touch and S. Hotz. The X-Bone. In *Proceedings of IEEE Global Internet Conference*, Sydney, Australia, Nov. 1998.

[40] P. F. Tsuchiya. The Landmark Hierarchy: Description and Analysis. Technical Report MTR-87W00152, The MITRE Corporation, June 1987.

[41] P. F. Tsuchiya. Landmark Routing: Architecture, Algorithms, and Issues. Technical Report MTR-87W00174, The MITRE Corporation, May 1988.

[42] K. Varadhan, D. Estrin, S. Hotz, and Y. Rekhter. *SDRP Route Construction*. Internet Engineering Task Force, Feb. 1995. Internet Draft, Expires August 27, 1995.

[43] K. Varadhan, R. Govindan, and D. Estrin. Persistent Route Oscillations in Inter-Domain Routing. *Computer Networks*, 32(1):1–16, Jan. 2000.

[44] D. Zhu, M. Gritter, and D. R. Cheriton. Feedback Based Routing. In *First Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, NJ USA, Sept. 2002.