MIT-CSAIL-TR-2006-025
April 14, 2006

Revisiting Internet Adressing: Back to the Future!

Mythili Vutukuru, Nick Feamster, Michael Walfish,
Hari Balakrishnan, and Scott Shenker

Revisiting Internet Addressing: *Back to the Future!*

Mythili Vutukuru[*], Nick Feamster[*], Michael Walfish[*], Hari Balakrishnan[*], and Scott Shenker[†]

Abstract

IP prefixes undermine three goals of Internet routing: accurate reflection of network-layer reachability, secure routing messages, and effective traffic control. This paper presents *Atomic IP (AIP)*, a simple change to Internet addressing (which in fact reverts to how addressing once worked), that allows Internet routing to achieve these goals.

1 The Past

> *Those who cannot learn from history are doomed to repeat it.* —George Santayana

In the early days of computer networking, each nascent technology came with various idiosyncratic mechanisms (error checking, flow control, sequencing, etc.), making interconnection difficult. To tame this daunting diversity, Cerf and Kahn proposed a universal internetworking layer (IP) that would provide, among other things, a unified and global addressing scheme [5]. IP's original addressing scheme (then called "TCP addressing") was elegant and simple. Each address had separate network and host components, and routers inspected only the network component of the address until the packet found its way to the destination network. The network addresses were implicitly assumed to be *flat*, with little correlation between network proximity and numerical similarity. This feature remained when the architecture moved to class-based addressing, a move precipitated by the combination of network growth and differing network sizes.

This design was well suited to the early Internet with its relatively small size and unified (or at least coöperative) administration. In the 1990s, however, the Internet's transformation into a large and federated infrastructure required two major changes in its addressing architecture. First, the rise of autonomous systems (ASes) introduced a different granularity of control; collections of networks, controlled by different organizations, had to be represented in routing decisions. This requirement led to the introduction of a new set of names—the AS numbers—that now play a crucial role in interdomain routing.

Second, the Internet's rapid growth led to an "incommensurate scaling" problem because portions of the address space (class B addresses) were being depleted rapidly. To handle this problem, the IETF engineered a minor change in addressing—the introduction of *classless* addressing (CIDR) [11], which allows a more flexible demarcation between the network and host address components—along with a major change in routing, namely the large-scale shift to *prefix-based interdomain routing*. The scaling properties of prefix-based routing depend critically on the correlations between network and numerical proximity, which was a radical departure from the flat addressing used for networks originally.

These two modifications (AS numbers and prefix-based addressing) were more a short-term engineering response to pressing needs than a principled, long-term architectural vision, and they brought much complication and clutter to the Internet's addressing scheme. The problems are not merely aesthetic; in §2, we argue that these *ad hoc* modifications introduced serious problems that continue to plague Internet routing and addressing. In particular, we highlight the difficulties in routing around failures, routing securely, and traffic engineering—problems with interdomain routing that are directly traceable to these new developments in addressing and routing.

In this paper, we revisit the question of Internet addressing. As we describe in §3, we propose a return to a clean two-level addressing scheme, with the higher level flat and globally known, and the lower level private and local to the particular network named by the higher level. Because network numbers no longer suffice for this higher-level address (given that they do not represent AS-level concerns), we introduce the notion of *atomic domains* (ADs), which are localized subsets of today's ASes. We propose that addresses take the form of `AD:LID` pairs, where `LID` is a local handle, the semantics of which only the AD knows. We articulate the design and discuss how it deals with the issues of avoiding failures, secure routing, and traffic engineering.

While we are not aware of any other proposal incorporating all of these elements, our work borrows from, and combines, many previously proposed ideas, and we point out our debts as we describe our mechanism. We also discuss some less directly related work in §4. With the simplicity of our proposed design and its resemblance to the Internet's original approach, we note that our contributions are less in technical innovation and more in the rationale for why we should return to the past to confront the future.

[*] MIT CSAIL

[†] UC Berkeley and ICSI

2 The Present

We learn from history that we learn nothing from history. —George Bernard Shaw

In this section, we argue that the two recent modifications to Internet addressing—AS numbers and prefix-based routing on classless addresses—make it difficult to route around failures, secure routing and perform traffic engineering.

Routing around failures: A desirable property of a routing protocol is to ensure that every route in a routing table corresponds to reachable destinations. Interdomain routing violates this property because routing handles (*i.e.*, IP prefixes) lack *failure atomicity*. A routing handle is failure-atomic if, from any given external location, either all addresses within the routing handle are reachable, or none are. A failure-atomic routing handle allows route updates to accurately reflect the reachability (or lack thereof) to destinations named by the handle.

To see the problem with IP prefixes, consider a customer network with prefix 2.1.0.0/16 served by a provider with a prefix 2.0.0.0/8.[1] The provider will typically advertise only the aggregated /8 prefix rather than individually advertise the prefixes of each of its customers. When the link to customer A fails, the provider often continues to advertise the whole /8 prefix rather than only advertise the prefixes currently available. As a result, other ASes wrongly believe that addresses within 2.1.0.0/16 are reachable via the advertised route when, in fact, they are not. More generally, the consequence of such scalability-driven aggregation is that routers today receive—and attempt to use—invalid routes that don't correspond to usable paths. Even if a customer network has a backup link via another provider (a growing trend), the alternate path may never be discovered or used by routers in the rest of the Internet.

Previous work supports this point. For example, Feamster *et al.* observe that, for a set of hosts around the world, many end-to-end path failures (in most cases, over 70%) are not reflected in BGP [8, Fig. 13]. The authors hypothesize that route aggregation is the cause for this poor correspondence. Moreover, recent work has shown that IP prefixes correlate poorly with geographic locality [10]; we believe (but have not yet demonstrated) that locality is highly correlated with failure, suggesting further that prefixes are improper routing handles.

Secure Routing: There is no inherent relationship or authoritative database linking an organization's AS number to its allocated IP addresses. The simultaneous use of two logically distinct and numerically unconnected addressing schemes causes the *origin authentication* problem, in which a rogue (or buggy) AS can "hijack" IP address space (*i.e.*, advertise an IP prefix that it does not own) and "blackhole" traffic. Its companion is the *path authentication* problem: although each route contains information about the sequences of ASes traversed by the announcement, the routing infrastructure does not guarantee either the accuracy of this information or its correspondence to the AS sequence that traffic actually follows en route to the destination.

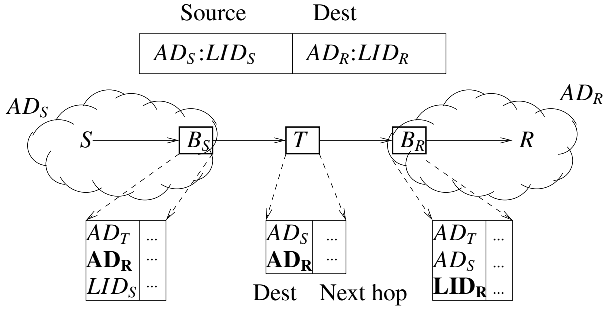

Figure 1: Forwarding behavior in AIP. Source S in AD_S sends a packet to receiver R in AD_R. T is a transit router. The AIP addresses in the packet are shown. The forwarding tables at the border routers (B_S, B_R, T) are also shown; the entry used for forwarding is shown in bold.

Traffic engineering: Today, many networks intentionally advertise more-specific prefixes (deaggregation), exploiting longest prefix matching to control how traffic reaches them. Yet, other ASes may aggregate sets of smaller contiguous prefixes to reduce routing table size. These conflicting operations often violate the originating network's policies, reduce stub networks' control over traffic, and lead to unpredictable traffic flow. We posit that interdomain traffic engineering is both complex and unpredictable because the granularity of today's routing handles (*i.e.*, IP prefixes) can be manipulated by entities not originating them.

3 AIP: A Modest Proposal for the Future

The best of prophets of the future is the past. —Lord Byron

In this section, we propose an alternative addressing structure, called *atomic IP (AIP)*, which attempts to redress the problems raised above. AIP's design is guided by the following three principles:

1. *Failure-atomic units*: The administrator of a network should be able to define that network as a failure-atomic unit.
2. *Immutable granularity of routing handles*: If a route corresponding to a failure-atomic unit originates from a given organization, then no other organization should be able to change the granularity of any routing handle that includes that organization.

[1] In CIDR, provider-dependent addressing is critical for scalability.

3. *Address-to-organization binding*: Every interdomain address should be bound to the organization that owns the address.

In AIP, each failure-atomic network, called an *atomic domain* (AD), has a unique identifier (its AD number). In practice, each AS would be divided into one or more ADs (we address this issue later in this section). The interdomain address of each host has the form `AD:LID`, where `AD` is the AD number of the host and `LID` is the local identifier (*e.g.*, a local address) of the host within the AD. The domain name system (DNS) would include an AIP-record containing the AIP address(es) for a hostname.

The `LID` component of an AIP address uniquely identifies a host within an AD, but AIP does not mandate or specify the `LID`'s form or structure. Each AD could pick its own `LID` format based on its intradomain network layer (*e.g.*, IPv4, IPv6, an MPLS tag, a virtual LAN identifier, etc.), or even use a topology-independent end-point identifier (as in HIP [19], UIP [9], or DOA [27]).

Interior and border routers in an AD maintain routing information on a per-AD basis for destinations in other ADs; *i.e.*, an AIP routing table maps AD numbers to "next hop" locations but does not maintain any information about `LID`s in other ADs. The border routers participate in an interdomain routing protocol (*e.g.*, BGP) to exchange these mappings. As in the current Internet, each router also participates in an interior routing protocol (*e.g.*, OSPF) to maintain routing information to the `LID`s within the AD. Note that AIP does not specify or mandate any particular choice of interdomain or internal routing protocol.

As shown in Figure 1, a host sends a packet to some destination by specifying the destination's `AD:LID`. Until the packet reaches the destination AD, each router forwards the packet based only on the `AD`. When the packet reaches some border router in the destination AD, that router forwards the packet to its destination based on the packet's `LID`.

The rest of this section discusses how AIP deals with the issues of failure atomicity, routing security, and better traffic control.

3.1 Failure Atomicity

To achieve failure-atomicity, each stub network in the current Internet located in a single city, or point-of-presence (PoP), would constitute a failure-atomic AD. Similarly, for a medium- or large-sized AS, the set of addresses in a single PoP would constitute an AD.

To estimate the number of ADs under this construction, we worked with other researchers who have access to various data sets classifying prefixes according to origin AS and geographic location; we found that the Internet has between 80,000 and 100,000 distinct ADs [25]. Current router hardware can easily support routing and forwarding tables of this size. In fact, this size is smaller than the number of prefixes in today's core border routers. The reason for the smaller number is explained by a recent study [10], which shows that each AS today announces many discontiguous prefixes from the same location (because of improper address space allocation). These prefixes cannot be aggregated into one routing table entry even though all of these prefixes "share fate" when any external link fails.

We need to distinguish between the routing handles used in a route to name the destination and the objects that are used merely for internal computations in the routing protocol (*e.g.*, AS numbers in the AS path of BGP). AIP uses AD numbers to name destinations. However, we need to use names with a larger granularity than ADs to describe the path in the route. To see why, if a transit AS wanted to shift traffic between PoPs, the AD path of routes through the affected PoPs would change, requiring a transit AS to propagate a new route. To alleviate this problem, we propose the following optimization: transit networks should have an AD number for the entire AS (called the "AS-level" AD number), in addition to the failure-atomic ("PoP-level") AD number. When the AD is being used as a transit network for an external destination, the AIP border router inserts the AS-level AD number in the AD path of the route announcement, rather than the failure-atomic AD number. In this way, the transit AS could shift traffic between PoPs without introducing any new routing update messages (presuming the rest of the AD path remained the same). When an AD originates a route though, that AD's border router inserts the failure-atomic AD number in the AD path of the BGP update so that ADs within the AS are themselves failure-atomic.

3.2 Routing Security

In §2, we argued that BGP lacks mechanisms that provide origin authentication or path authentication. In this section, we explain how AIP eliminates the former and simplifies the latter.

Origin authentication (*i.e.*, a secure mapping between IP prefixes and ASes) is a hard problem today because no authentic mapping exists between routing handles (prefixes) and the entities announcing them (ASes). Maintaining the integrity of this indirection requires a "routing registry" (a public database storing the AS owner of each prefix). Unfortunately, keeping such a database accurate and up-to-date has proven difficult [12]. S-BGP [16] suggests that ASes obtain certificates from a trusted authority (*e.g.*, an Internet Registry) proving ownership of the prefixes they announce. These certificates must still be kept up-to-date as ASes delegate address space to their customers, which proves unwieldy. AIP *eliminates* the need for origin authentication by removing the indirection between the names of routing handles and the entities announcing them. The routing handle *is* the AD number and thus trivially identifies the entity that owns it.

Proposals like S-BGP [16] and SPV [14] solve the path authentication problem by having each AS along the AS path sign the AS path attribute of the BGP update and append the signature to the update before sending the update to other neighboring ASes. A router receiving a BGP update then establishes the authenticity of the AS path in the update by verifying the signatures of all ASes in the AS path. This description is incomplete, however, because saying that routers can verify signatures implies that routers somehow know the public keys of the ASes. Giving routers this knowledge today requires a public key infrastructure (PKI), *i.e.*, a trusted third party issuing certificates of the form "AS X's public key is K." Establishing such a PKI is burdensome.

AIP eliminates the need for a PKI because *ADs can be named by the public keys themselves* (this mechanism is inspired by other systems that use *self-certifying* names [18, 19]). Naming ADs after public keys is not a radical departure from the status quo: AS numbers, after all, are already flat names and encode no semantics. Of course, public keys are large, and putting several in a single route advertisement might be impractical; accordingly, we propose that AD numbers be *hashes* of public keys rather than the public keys themselves. This optimization mandates a bootstrapping mechanism for exchanging the actual public keys. To this end, routers could simply exchange public keys in separate routing BGP messages, a lookup service could be used to resolve the hash of an AD to its public key as in Secure Origin BGP (soBGP) [28], or the route announcements could periodically contain the public keys.

Once ADs are named by hashes of public keys, path authentication in AIP proceeds exactly as in (and provides the same guarantees as) S-BGP: some router in AD_i signs the AD path $[AD_{i+1}\ AD_i\ \ldots\ AD_0]$. A router receiving an announcement with this AD path verifies every signature in the route update before installing the route in its routing table. Thus, each route advertisement must be signed once by each AD along the AD path; a router that receives a route must verify $N-1$ signatures, where N is the number of labels on the AD path.

Critics have argued [14] that S-BGP's cryptography is prohibitive, but we disagree. First, route processors on routers are becoming faster. For example, Cisco's CRS-1 route processor is 1.2GHz. Second, S-BGP could use a more efficient cryptosystem such as ESIGN [21]. Microbenchmarks show that in the ESIGN cryptosystem, a 3 Ghz processor can perform sign and verify operations on 2048-bit keys in 150 and 100 microseconds, respectively [17, §7.2.2]. Assuming that a router may learn on the order of 100,000 ADs, with two or three routes per AD, each having an average path length of 4, a router would be able to verify an entire routing table's worth of signatures on the order of minutes; of course, peak load would be substantially less. Third, public key operations can easily be offloaded to a set of dedicated computers in the AD.

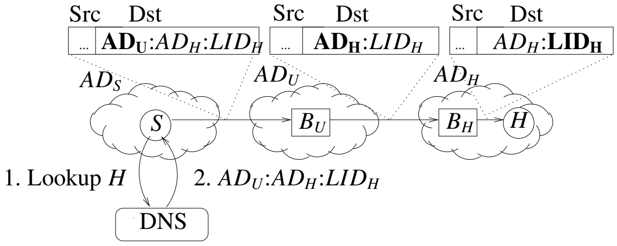

Figure 2: Traffic control using delegation. Sender S resolves the DNS name of traffic controlled host H to get its delegation record AD_U:AD_H:LID_H. Traffic from S to H goes via AD_U. The AIP header field used for forwarding at each stage is in boldface.

One disadvantage to naming ADs with public keys is that revoking a public key results in a change in the AD number of the AD. As a result, either all of that AD's route advertisements must be withdrawn and re-advertised with the hash of a new public key or additional BGP messages that signal change in the AD number should be propagated with BGP updates. Revocation also requires updating DNS entries for end hosts whose ADs change. But we believe the benefits of eliminating a PKI (which would have to handle revocation anyway) outweigh these problems with key revocation (which we assume will be a rare event).

3.3 Traffic Control

By design, the AD number in a route update is flat and cannot be manipulated. Hence the mechanisms used to control inbound traffic in today's Internet (*e.g.*, announcing more specific prefixes of a larger prefix) cannot be used with AIP. In this section, we propose a mechanism that allows organizations to control inbound traffic in AIP. (Outbound traffic control follows today's approach: setting the import policy of routes at the border routers.)

Inbound traffic control allows a network with two or more upstream providers to specify that it prefers one of those upstreams to carry traffic destined to a particular host (or set of hosts).[2] For example, an organization with two Web servers within the same AD and two upstream providers might want traffic destined to those servers to arrive via different upstream ADs.

AIP uses DNS to perform inbound traffic control at the granularity of individual hosts (unlike BGP, which provides only prefix-level control). If AD_H wants to control inbound traffic to one of its hosts, H, it modifies H's DNS entry to be instead $\mathbf{AD_U}$:AD_H:LID_H (rather than simply AD_H:LID_H); here, AD_U is the upstream AD preferred by

[2] An "upstream" here refers to an AD that is willing to carry traffic for a downstream AD, not necessarily an immediate upstream provider.

AD_H. Hosts wishing to send traffic to H then resolve H's DNS name, get back this *delegation record*, and insert it into the destination address of the packet. This scheme is inspired by the idea of "delegation" used in other systems [22, 27].

Figure 2 shows a slight modification to the forwarding scheme described in §3 required to enable delegation. Routers forward packets based on the AD number at the top of the stack. When the topmost AD is reached, the border router pops off the topmost AD number from the destination address stack. If the next address is the stack is an LID, the router forwards the packet to a host within the AD. Otherwise, it forwards the packet to the next AD in the stack of destination address.

This approach of controlling traffic to a host works when the DNS name of the host is looked up before initiating communication; but AIP must also let ADs control *return* traffic to their hosts when their hosts have *initiated* connections. To this end, a host initiating communication (a "client") inserts its *own* delegation record into the source address field of the first packet of the communication (*e.g.*, a SYN packet in the case of TCP); clients can get their delegation records from their ADs using DHCP or by hard-coding. The client's remote peer then places this delegation record in the destination address of the packet when replying to the client.

The delegation mechanism described above is susceptible to a man-in-the-middle attack: an adversary could place its own AD as the upstream AD in a spurious delegation record of a packet, causing return traffic from the server to go to the adversary instead of to the client. To defend against this attack, an end-host acting on a delegation record AD_U:AD_H:LID_H to send traffic to H must verify that AD_H requested the delegation. So AD_H must sign the delegation record AD_U:AD_H:LID_H with its private key; hosts sending packets to H then verify this signature.

4 Related Work

You can observe a lot just by watching.
—Yogi Berra

Although AIP is the first proposal to suggest that addresses should satisfy failure atomicity, several other projects have recognized the shortcomings of routing on IP prefixes and proposed various alternatives. Both HLP [23] and BGP policy atoms [26] propose changing the granularity of routes. However, HLP's goals are complementary to AIP's: HLP aims for more scalability, better isolation, and faster convergence. Also, HLP's routing handles are AS numbers; in contrast, AIP proposes finer-grained AD numbers. BGP policy atoms reduce routing table state by grouping prefixes that are subject to the same policy [26]. Unlike policy atoms, AIP explicitly groups routing handles according to failure atomicity; one can think of an AD as a "failure atom".

Several other proposals suggest changes to Internet addresses, though for different reasons from AIP.[3] The GSE (or "8+8") [6, 20] proposal suggests that end-host addresses contain lower bits that identify the host, and upper bits that, in contrast to AIP's AD number, reflect topology. These proposals (and other measurement studies [3]) lament the current fragmentation of IP space. AIP gets around this problem by eliminating aggregation altogether, whereas GSE encourages provider-based aggregation by letting hosts—which are unaware of their top address bits—renumber easily when they switch providers. NIRA's [29] addresses, like AIP's, have interdomain and intradomain pieces but are hierarchical. NIRA lets an end-host select its transit networks; thus, end-hosts have multiple addresses, one for each path to the core. Nimrod [4] focuses on scalability, TRIAD [13] on content-based routing and stitching together NATed realms; both projects have comprehensive scope, whereas AIP inherits much of the status quo (*e.g.*, BGP, DNS).

AIP's addresses support heterogeneity by allowing each AD the freedom to use its own local addressing scheme. Plutarch [7] also connects heterogeneous networks (called "contexts") but eschews a global addressing scheme like AIP's. AIP modifies addressing, while Plutarch admits IPv4 as one of many "contexts". UIP [9] also connects networks of different types but does so with a global identifier space overlaid on IP; UIP could run on top of AIP instead of IP.

AIP's addressing scheme relates to several existing proposals to secure routing. In §3.2, we discussed how self-certifying route handles enable path authentication schemes such as S-BGP [16] and SPV [14] without requiring a PKI. Listen and Whisper [24], like AIP, uses public key cryptography yet avoids a PKI; its purpose is to detect inconsistencies in AS paths, a function subsumed by the path authentication AIP inherits from S-BGP.

AIP allows a network better control over how traffic reaches its network. Akella *et al.* studied the benefits of providing this control to the increasing number of networks with multiple upstream networks [1] and have also proposed a DNS-based system to provide these networks control over inbound traffic at the AS level [2]. AIP shares the goals of this system and also uses DNS for fine-grained inbound traffic control, but it also proposes direct AD-level control of inbound traffic.

5 Summary

This is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning. —Winston Churchill

In this paper, we argued that IP prefixes are the wrong granularity for Internet routes because they are not failure-

[3] See the NIRA [29] and BANANAS [15] papers for more comprehensive lists of routing architecture proposals.

atomic, they have no correspondence to the identity of the network that owns them, and they can be manipulated by other networks in ways that make traffic engineering more difficult. We proposed an alternative two-level addressing scheme called *atomic IP (AIP)*, where each failure-atomic unit of the network is named by a flat address called an atomic domain (AD). Routing on ADs allows the routing protocol to accurately reflect reachability while still scaling to a large number of end hosts. We argued that naming each AD by its public key can make it easier to secure interdomain routing without requiring a public key infrastructure, by simplifying path authentication and eliminating the need for origin authentication altogether. We discussed how delegation can help an AD control inbound traffic in a way that, unlike IP prefixes, cannot be manipulated by remote networks.

But what about adoption? Many in the networking research and operations community have recognized that the current interdomain routing architecture is bursting at the seams, and is ripe for change. A useful direction for the community would be to take the mix of proposed directions and ideas, and develop consensus on the best way forward. We believe that failure-atomicity and better routing security are desirable, and hope that the simple ideas proposed herein (which are a throwback to the past) can inform the debate about the future of Internet routing.

References

[1] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman. A measurement-based analysis of multihoming. In *SIGCOMM*, Aug. 2003.

[2] A. Akella, S. Seshan, and A. Shaikh. Multihoming performance benefits: An experimental evaluation of practical enterprise strategies. In *USENIX Annual Technical Conference*, Boston, MA, June 2004.

[3] T. Bu, L. Gao, and D. Towsley. On characterizing routing table growth. In *Global Internet*, 2002.

[4] I. Castineyra, N. Chiappa, and M. Steenstrup. The Nimrod routing architecture. RFC 1992, Aug 1996.

[5] V. G. Cerf and R. E. Kahn. A protocol for packet network intercommunication. *IEEE Transactions on Communications*, Com-22(5), May 1974.

[6] M. Crawford, A. Mankin, T. Narten, J. W. Stewart, and L. Zhang. Separating identifiers and locators in addresses: An analysis of the GSE proposal for IPv6, Oct. 1999. Internet draft draft-ietf-ipngwg-esd-analysis-05.txt (Work in progress).

[7] J. Crowcroft, S. Hand, R. Mortier, T. Roscoe, and A. Warfield. Plutarch: An argument for network pluralism. In *SIGCOMM FDNA Workshop*, Aug. 2003.

[8] N. Feamster, D. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring the effects of Internet path faults on reactive routing. In *SIGMETRICS*, June 2003.

[9] B. Ford. Scalable Internet routing on topology-independent node identities. Technical Report MIT-LCS-TR-926, MIT CSAIL, Oct. 2003.

[10] M. Freedman, M. Vutukuru, N. Feamster, and H. Balakrishnan. Geographic locality of IP prefixes. In *Proc. ACM SIGCOMM Internet Measurement Conference*, Oct. 2005.

[11] V. Fuller, T. Li, J. Yu, and K. Varadhan. *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*, Sept. 1993. RFC 1519.

[12] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working around BGP: An incremental approach to improving security and accuracy in interdomain routing. In *NDSS*, Feb. 2003.

[13] M. Gritter and D. R. Cheriton. TRIAD: A new next-generation Internet architecture, http://www-dsg.stanford.edu/triad/, July 2000.

[14] Y.-C. Hu, A. Perrig, and M. Sirbu. SPV: Secure path vector routing for securing BGP. In *SIGCOMM*, Aug. 2004.

[15] H. T. Kaur et al. BANANAS: An evolutionary framework for explicit and multipath routing in the Internet. In *SIGCOMM FDNA Workshop*, Aug. 2003.

[16] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol (S-BGP). *IEEE JSAC*, 18(4):582–592, Apr. 2000.

[17] J. Li, M. Krohn, D. Mazières, and D. Shasha. Secure untrusted data repository (SUNDR). In *OSDI*, Dec. 2004.

[18] D. Mazières, M. Kaminsky, M. F. Kaashoek, and E. Witchel. Separating key management from file system security. In *SOSP*, Dec. 1999.

[19] R. Moskowitz and P. Nikander. Host identity protocol architecture, Sep 2003. draft-moskowitz-hip-arch-05, IETF draft (Work in Progress).

[20] M. Ohta. 8+8 addressing for IPv6 end to end multihoming, Jan. 2004. Internet draft draft-ohta-multi6-8plus8-00 (Work in progress).

[21] T. Okamoto and J. Stern. Almost uniform density of power residues and the provable security of ESIGN. In *ASIACRYPT*, pages 287–301, 2003.

[22] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. In *SIGCOMM*, Aug. 2002.

[23] L. Subramanian et al. HLP: A next-generation interdomain routing protocol. In *SIGCOMM*, Aug. 2005.

[24] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz. Listen and whisper: Security mechanisms for BGP. In *NSDI*, Mar. 2004.

[25] R. Sundaram and M. Marathe, July 2005. Private communication.

[26] P. Verkaik, A. Broido, kc claffy, R. Gao, Y. Hyun, and R. van der Pol. Beyond CIDR aggregation. Technical Report TR-2004-01, CAIDA, Feb. 2004.

[27] M. Walfish, J. Stribling, M. Krohn, H. Balakrishnan, R. Morris, and S. Shenker. Middleboxes no longer considered harmful. In *OSDI*, Dec. 2004.

[28] R. White. *Architecture and Deployment Considerations for Secure Origin BGP (soBGP)*, May 2005. Internet Draft draft-white-sobgp-architecture-01.txt; work in progress.

[29] X. Yang. NIRA: A new Internet routing architecture. In *SIGCOMM FDNA Workshop*, Aug. 2003.