Scaling IP Routing with the Core Router-Integrated Overlay

Xinyang Zhang
Cornell University
Ithaca, NY
jzhang@cs.cornell.edu

Paul Francis
Cornell University
Ithaca, NY
francis@cs.cornell.edu

Jia Wang
AT&T Labs Research
Florham Park, NJ
jiawang@research.att.com

Kaoru Yoshida
The University of Tokyo
Tokyo, Japan
kaoru@hongo.wide.ad.jp

***Abstract*— IP routing scalability is based on hierarchical routing, which requires that the IP address hierarchy be aligned with the physical topology. Both site multi-homing and switching ISPs without renumbering break this alignment, resulting in large routing tables. This paper presents CRIO: a new approach to IP scalability for both global and VPN routing. Using tunneling and virtual prefixes, CRIO decouples address hierarchy and physical topology, effectively giving ISPs the ability to trade-off routing table size for path length. Though CRIO is a new routing architecture, it works with existing data-plane router mechanisms. Through static simulation on a Rocketfuel-measured Internet topology and traffic data from a Tier 1 ISP, we show that CRIO can shrink the BGP RIB by nearly two orders of magnitude, the global FIB by one order of magnitude, and the VPN FIB by ten to twenty times, all with very little increase in overall path length.**

I. Introduction

Historically there has been only one way to scale IP routing in the Internet: *topological hierarchy*. Here, a topologically contiguous portion of the network (a "cloud"), such as a campus network or an ISP, is given a contiguous block of addresses. That address block (or *address prefix*) is aggregated in routing updates so that routers outside of the cloud require only one routing table entry to represent everything inside the cloud. This basic approach was proposed and analyzed early in the development of IP (for instance [10]) and has served as the sole means of IP scalability ever since.

Topological hierarchy constrains how networks are deployed and addresses are assigned. This constraint is not a good fit for how people like to deploy networks and assign addresses in practice, in two ways. First, a network *site* (e.g. an ISP customer network) cannot change ISPs without either changing its address prefix, or requiring the new ISP to advertise its address prefix into BGP[1], thus breaking the coupling between topology and address. The former forces the site to renumber all of its IP devices, while the latter results in routing table growth in the BGP routing core (the *default free* zone)[2].

One often-proposed fix to this problem is to use geography-based addresses [18], [23] instead of ISP-based addresses. Similar to how phone numbers are assigned, a site's prefix would reflect the geographic location of the site, not its ISP. This approach requires that all ISPs serving a geographic region be topologically interconnected within that region. ISPs have resisted this constraint. Other approaches include using coordinates (for example [15]), but these also constrain topology and are not realistic for the Internet.

The second way in which topological hierarchy constrains deployment is with *site multihoming*, that is, where sites connect to more than one ISP. Here, each site must obtain its address prefix from only one of the ISPs. The remaining ISPs must advertise the prefix, again breaking the coupling between topology and address for those ISPs, again resulting in routing table growth [22]. This growth is only checked by ISPs refusing to propagate prefixes larger than a 24, thus limiting the number of sites that can multihome.

This paper proposes a new method of scaling global and VPN IP routing called the Core Router-Integrated Overlay (CRIO). Through the use of tunnels, CRIO allows the decoupling of addressing from topology. As a result of this decoupling, ISPs are given a new tuning knob: one that trades-off routing table size for path length. Using a model based on the actual Internet topology and traffic statistics from a Tier 1 ISP, we show that FIB size for global routing can be reduced by ten times with very little path length penalty, and by twenty times with a modest path length penalty. The FIB size for VPN routing in a Tier 1 ISP can be reduced ten to twenty times with very little path length penalty. Finally, we also show that the size of the BGP RIB can be reduced by nearly two orders of magnitude.

We illustrate the idea behind CRIO with a simple example. Figure 1a shows a simple two-level traditional hierarchical network with two ISPs X and Y, each with two attached sites and two routers. The routing table for Router D shows that it only requires one routing table entry for all the sites attached to ISP Y, because their addresses all have the form 2.* (the '*' represents a wild-card in the second digit of the 2-level address).

Figure 1b shows the same topology, but using CRIO instead of the traditional hierarchy. Note that address prefixes are no longer associated with ISPs. Rather, they are associated with individual routers. For instance, Routers D and F are associated with the prefix 2.*. This means that

[1]This paper does not provide references for well-known protocols.

[2]Actually, a third alternative exists, which is for the site to use Network Address Translation (NAT), and use private addresses internally. While this approach is popular, the Internet standards community, through IPv6, is trying to move away from NAT.

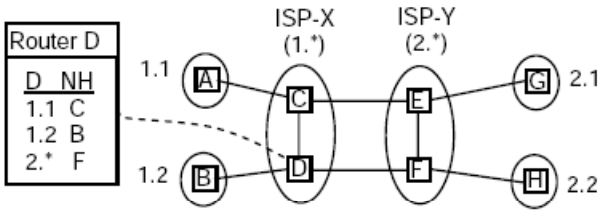

Figure a: Traditional Addressing Hierarchy

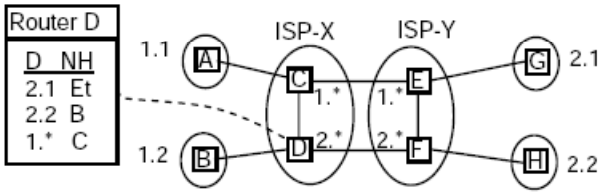

Figure b: CRIO Architecture (Virtual Prefixes)

Fig. 1. An example of traditional hierarchy and CRIO using a simple 2-level hierarchical address. In the routing table for Router D, the column labeled 'D' shows the destination, and the column labeled 'NH' shows the next hop, or in the case of CRIO, tunnel endpoint.

Routers D and F must both know how to reach all sites with a 2.* prefix. Through the use of tunneling, however, it is not necessary for Routers D and F or the 2.* sites to be topologically contiguous. For this reason, we call these prefixes *virtual prefixes*. Router D, for instance, can forward packets to site 2.1 by tunneling packets to Router E (the 'Et' entry in the routing table denotes "tunnel to E"). This of course means that Router D must know the next hop to E (which is not explicitly shown in Figure 1b), but in practice this amounts to storing a route for each Internet POP, which number is the low thousands. In practice the tunnels may be IP-over-IP, or inter-ISP MPLS.

Path lengths in CRIO can be longer than shortest path. For instance, consider a packet from Site 1.1 to Site 2.1. The packet would go to Router C, which would forward it to Router D, the nearest router associated with virtual prefix 2.*. Router D would then tunnel the packet to Router E, possibly forwarding the packet back via Router C. Router E would detunnel the packet, and forward it on to Site 2.1.

In spite of the fact that CRIO represents a new approach to scaling, it can be built with existing tools. This is in part because CRIO uses prefix-based routing (albeit logical), and so can be handled by BGP as-is, and in part because of the widespread use of tunneling today, for instance for VPN service. Indeed, CRIO is reminiscent of Mobile IP, which uses tunnels rooted at Home Agents to decouple addresses and topology.

Given this simple idea, there are many questions to be answered concerning both the static and dynamic performance of CRIO. The core question for the static performance of CRIO is, what would the routing table size versus path length trade-off be in today's Internet, for both global routes and VPN routes? Related to this are question such as: How big should the virtual prefixes be? Where should virtual prefix routers be positioned? What additional tunnels should routers be configured with? This paper focuses on these questions.

The core question for dynamic performance is, how are tunnels installed and maintained (i.e. in the face of failures) in routers? This paper briefly speculates on these and other dynamic performance issues, and suggests that their solutions are relatively straightforward. Nevertheless, this paper does not fully answer them. While these questions are obviously crucial, we want to first insure that the static performance of CRIO provides significant benefits before moving on to the admittedly more difficult issue of dynamics.

The contributions of this paper are:

- A description of CRIO: a new approach to scaling global and VPN IP routing that effectively decouples topology and addressing through the use of virtual prefixes and tunnels.
- A static analysis of CRIO using the actual Internet topology as measured by Rocketfuel, and using real traffic data from a Tier 1 ISP.

The rest of the paper is organized as follows. We present the CRIO architecture in detail in Section II. Section III gives the static evaluation of CRIO for both global and VPN routing. We discuss various issues related to CRIO, including dynamic performance, in Section IV. Finally, we discuss related work in Section V and conclude with Section VI.

II. CRIO Architecture

CRIO uses uni-directional inter-ISP tunnels that start and end at provider edge routers (here called PE, for Provider Equipment) in Tier-1 and Tier-2 POPs. These tunnels may be based on MPLS or GRE. Routers know which tunnel to choose for a given IP prefix based on *mapping* entries distributed to PEs, where each entry consists of the tuple:

$$\langle \text{IP prefix; tunnel endpoint; (optional) policy} \rangle$$

where *tunnel endpoint* is the IP address of a PE which may be reachable via an IP-GRE (IP-GRE-IP) tunnel or an MPLS label. Multiple mapping entries with the same prefix are used for multi-homed sites, one per access link. The optional policy field can be used to determine how PEs choose among the multiple tunnel endpoints. Once a packet sent by a source arrives at an ingress PE (normally through default routes), the forwarding process at the PE is then a two step process: (i) determine the tunnel endpoint for a given prefix (if any); (ii) tunnel the packet into an outer IP header or MPLS header, and find the next hop to the tunnel endpoint. BGP itself is required only for the latter step.

In what follows, we fill in the details of the architecture in a step-by-step fashion. We start by giving an example of packet traversal under CRIO. We then describe the tunnels themselves (Section II-B). Next, we describe the tunnel mapping table and its characteristics (Section II-C). Finally,

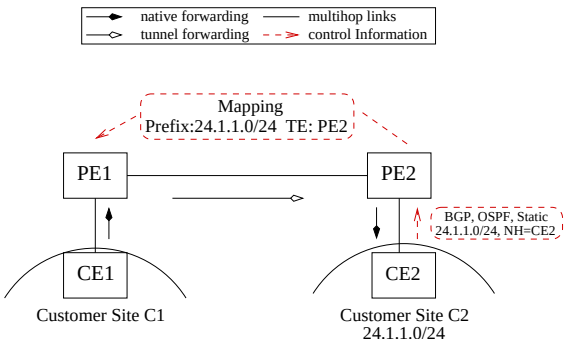

Fig. 2. CRIO path through the internet

we describe virtual prefixes and how they can be used to greatly reduce the size of the tunnel mapping table seen at any given router, and therefore the size of router forwarding tables (Section II-D).

A. An Example

Figure 2 illustrates the control plane information flow under CRIO, and the corresponding packet forwarding. This is for a packet originating from customer network (site) $C1$ to an address in $C2$ as it passes through the provider networks. Router $PE1$ is an ingress provider edge router and $PE2$ is an egress provider edge router. The "links" between $PE1$ and $PE2$ in this figure, as is the case with other similarly constructed figures in this paper, are not physical links. In fact, $PE1$ and $PE2$ may not locate in the same administrative domain. Rather the physical path between $PE1$ and $PE2$ could be a multi-AS paths through the Internet.

The dashed line in the figure represents the control plane information flow. Initially, the site edge router $CE2$ exchanges routing information with $PE2$, either through a dynamic routing protocol such as BGP, or through static configuration. If the site needs to convey any policy information to the provider, this can be done in a BGP community attribute, or statically. Having received the site prefix, $PE2$ creates the mapping. In this example, the mapping consists of the site prefix 24.1.1.0/24, the Tunnel End (TE) value, in this case the IP address of $PE2$, and the optional policy (Section II-C). The mapping is then distributed across the provider networks through some external mechanism and installed in $PE1$.

Once the mapping is established, a packet from network $C1$ to network $C2$ is forwarded as follows. First, $CE1$ sends the packet to $PE1$ as normal. $PE1$ has a tunnel mapping entry (Prefix=24.1.1.0/24, TE=$PE2$) indicating that the packet should be tunneled to $PE2$. The packet transmitted by $PE1$, then, will have an outer tunnel header (which could be MPLS or IP depending on the tunneling mechanism, discussed later) and the original header as the inner header. $PE2$ detunnels the packet upon receiving it, then forwards the original IP packet to $CE2$. Note that the tunneled portion was between $PE1$ and $PE2$. In the terminology of this paper, we refer to $PE1$ as the *Tunnel Startpoint* (TS), and to $PE2$ as the *Tunnel Endpoint* (TE).

B. CRIO Tunnels

This example is in some respects nothing new. Any tunneling-based services offered in the Internet today, for instance VPNs, tunnel in much this fashion. The main difference between this example and current practice is the fact that, in CRIO, tunnels extend between arbitrary pairs of PEs in different provider networks.

Inter-provider tunnels are slowly becoming more common, primarily in the context of multi-provider VPN services. In CRIO, any given PE requires a tunnel to every PE for which it has a mapping. This could easily constitute the large majority of PEs in the Internet core. To cope with this proliferation of tunnels, we envision the use of *one-ended* tunnels, whereby the TE accepts tunneled packets regardless of where they came from. In the case of MPLS, what this means is that the same label is used for a given TE by an intermediate MPLS router independent of the source of the packet. In the case of IP-GRE, this means that the TE does not filter on the source address of tunneled packets as they typically do today.

The use of one-ended IP-GRE tunnels simplifies tunnel configuration considerably. All that is required is that the TE information in the mapping entry include the relevant tunnel information, such as a GRE Key. The use of one-ended MPLS tunnels is more involved, since the labels have to be assigned along the path from source to destination. Either way, the scaling properties are similar: the information required in each PE scales linearly with the number of PEs. We have not considered in detail any issues related to establishing one-ended labels on such a large scale, in part because the IP-GRE approach appears to be straightforward. As such, in the remainder of this paper, we tacitly assume IP-GRE tunnels, with the understanding that we see no outstanding reason why MPLS tunnels can't be used.

CRIO tunneling effectively partitions the path into three distinct parts:

- **UP:** IP routing from the source to the TS.
- **ACROSS:** Tunneled from the TS to the TE.
- **DOWN:** IP routing from the TE to the destination.

We characterize these as up, down, and across because the TS and TE tend to correspond to provider edge routers where packets enter and leave the provider infrastructure. As such, we can think of the path as going up from the source network to the ingress provider edge router, across to the egress provider edge router, and down to the destination network. By this characterization, the across part constitutes the bulk of the path—the core of the Internet.

CRIO frees the routers in the across part of the path from having to compute BGP routes to all Internet destinations. Rather, BGP itself only needs to compute routes to the *TE prefix*. Here, the TE prefix is an aggregation of a collection of TE addresses. For instance, all the TE within one POP can be

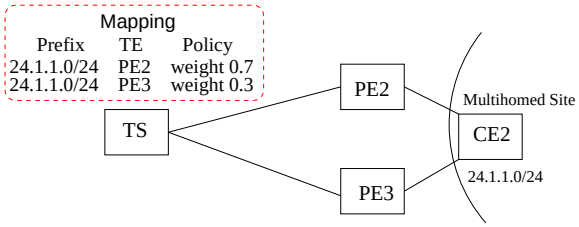

Fig. 3. Multihomed Site (Reachable Via Router CE2)

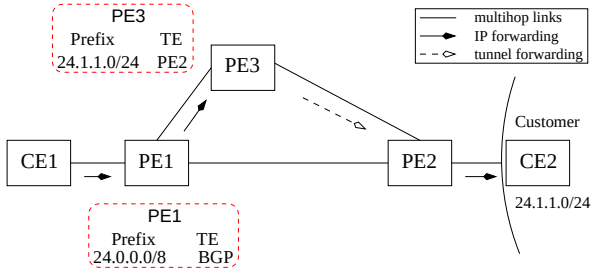

Fig. 4. Virtual prefix: $R3$ is VP for 24.0.0.0/8

aggregated into a single prefix. Of course, the providers can assign TE addresses and advertise the prefixes in anyway that is suitable. Limiting a TE prefix to a single POP, however, appears to be a natural way to deploy. It provides for very fine-grained policies without compromising robustness. The number of provider POPs in the Internet is in the thousands, not tens of thousands [12], [16]. This gives two distinct benefits. First, the size of the BGP computation is much smaller. Second, the routes that are being computed are for very stable destinations.

C. Tunnel Mappings

Mappings are created at, or more accurately, near the TE associated with the mapping. Mappings tend to be created through an interaction with the Autonomous System (AS) that contains the prefix (i.e., the site), and the AS that contains the TE (i.e., the ISP).

If a site obtains its connectivity through multiple POPs, then it will correspondingly have multiple mappings. This is true whether the POPs are in the same or different ISPs. For instance, in Figure 3, the site prefix 24.1.1.0/24 reachable through router $CE2$ is multihomed to two TEs, $PE2$ and $PE3$. As such, there are two mappings for 24.1.1.0/24, $PE2$ and $PE3$.

In today's BGP, if a multihomed site was multihomed to two different ISPs (that is, $PE2$ and $PE3$ were different ISPs), then the prefix 24.1.1.0/24 would have to be advertised globally even if it was otherwise aggregatable in one of the ISPs. With CRIO, BGP is not burdened by the multihoming—routers only compute routes to the POP prefixes containing $PE2$ and $PE3$. Whats more, in CRIO, the traffic engineering policies of 24.1.1.0/24 may be encapsulated in the mappings. For instance, in Figure 3, the $PE2$ mapping is annotated with the policy "weight 0.7", and the $PE3$ mapping is annotated with the policy "weight 0.3". This indirectly conveys to the TS the desired ratio of traffic the site wishes to receive on its incoming access links.[3]

If the site becomes unreachable via one TE, the mapping between the TE and prefix becomes invalid. This fact must be distributed to the all TSs where the mapping is used, as discussed in Section IV.

[3]To be clear, the TS does not have to honor the policy: it may and typically will give priority to its own traffic engineering and routing policies, and only honor the site's policy when doing so doesn't adversely impact its own operation.

D. Virtual Prefixes

Up to this point in the description, all we've done is shift routing information from BGP to some kind of mapping table distribution protocol. We have not shrunk the total amount of information that would reside in any given router's forwarding table. To the contrary, since one of our goals is to allow for an increase in the amount of multihoming, we would expect to see substantial overall growth in forwarding information.

Virtual prefixes allow us to shrink the forwarding tables. A virtual prefix is a super-prefix that spans a large portion of the address space, for instance a /8, even though the individual prefixes in the virtual prefixes are otherwise non-aggregatable. Virtual prefixes themselves are carried by BGP/IGP. The routers that advertise a given virtual prefix are TSs that hold the mappings for *every prefix within the virtual prefix*. We call such a TS a VP-TS. Any ISP can independently determine which routers should be made VP-TSs for which virtual prefixes. In general, ISP should not advertise virtual prefixes to any neighboring AS other than its customers. Otherwise, it might carry transit traffic for its peering or even upstream provider.

Figure 4 illustrates the usage of virtual prefix. Assume that router $PE3$ is a VP-TS for 24.0.0.0/8, and advertises this prefix into BGP. Router $PE1$ receives a packet for a destination within 24.1.1.0/24. $PE1$ does not have a mapping for this prefix, so it forwards the packet towards $PE3$. When $PE3$ receives the packet, it does have a mapping for 24.1.1.0/24 (and indeed all prefixes within 24.0.0.0/8), and so tunnels the packet to TE $PE2$. Notice that this packet took a longer path than it would have taken if $PE1$ had a mapping entry for 24.1.1.0/24. Instead of taking the path $CE1 - PE1 - PE2 - CE2$ (and intermediate routers not shown), it took the path $CE1 - PE1 - PE3 - PE2 - CE2$.

The existence of $PE3$'s virtual prefix provides $PE1$ with a tuning knob that allows it to trade-off forwarding table size for path length on a per-prefix basis. This is a good trade-off to be able to make. Most of the traffic crossing the Internet is for a small fraction of destinations. The large majority of prefixes in a router's forwarding table are for destinations for which there is very little traffic [9]. This means that most routers could shed most of their prefixes with very little overall increase in traffic volume. Unfortunately, this may

unfairly punish low-traffic-volume sites that just happen to end up with very bad paths. In this case, the ISP can choose to install the mappings of those particular sites to prevent the long paths.

III. CRIO Static Analysis

In this section, we evaluate the static performance of CRIO by simulation. We evaluate the effect of CRIO on global routing tables using a POP-level map of Tier 1 ISP and the traffic matrix from one Tier 1 ISP. We also evaluate the effect of CRIO on VPN routing tables for a large VPN provider and one of its national-sized customers.

A. Data Gathering

Our analysis uses three sets of data for both global Internet and VPN: network topology, prefix-TE mapping, and traffic matrices.

Internet Topology: We obtained a POP level topology from Rocketfuel [3], an ISP topology mapping engine. Our topology, which includes ISPs that are classified as Tier-1 by [12], consists of the Tier-1 subset of Rocketfuel's POP-level ISP map which includes 23 ISPs with 1219 POPs, and 4159 inter-POP links.

Internet TE-prefix Mapping: We derive the TE-prefix mappings using the Rocketfuel raw traces. For each prefix, we first gather all traces destined to that prefix (possibly from different sources). Then for each trace, we back-track until we hit the first POP in our topology. The POP is then identified as a TE for the prefix. Because our topology only consists of Tier-1 ISPs, it is possible that the POP we identified is not the immediate upstream POP for the prefix. The fact that we completely ignored the non-Tier1 topology will certainly have an impact on our analysis. Most noticeable is that path lengths (in terms of hops) will be shorter than they actually are. However, this reduction in path lengths is an overall effect equally affecting all the paths. Since we are not interested in the actual value of the path length, we believe that the limitation on the network topology will not invalidate our analysis.

Internet Traffic Matrices: We also compute the prefix-level traffic flow matrices across all the POPs in our topology. The element in the ith row and the jth column in a traffic flow matrix is the amount of traffic that originate from POP i and terminate at prefix j. To build traffic flow matrices, we use router-level Netflow [2] records from multiple border routers across a large Tier-1 ISP's backbone during the week from 12/01/2004 to 12/07/2004.

For each flow, NetFlow maintains a record in the router cache containing a number of fields including the source and destination IP addresses, source and destination routing prefixes, source and destination ASes, source and destination port numbers, the protocol type, type of service, flow starting and finishing timestamps, number of bytes and number of packets transmitted. A flow can be counted both at the ingress router and at the egress router. We remove the duplicate counted flow records before aggregating flows at the prefix level. Since we are interested in the inter-POP traffic flow matrices, we remove all the aggregated flow records that originate from and destinate to the same POP.

Note that we only have access to one Tier-1 ISP's traffic data. We are only able to calculate the traffic flow data for all POPs in that ISP. In other words, we can only fill the ith row in the traffic flow matrix if we have the traffic data for POP i. For the remaining POPs in our topology, we generate their traffic flow data by taking the existing row and randomly permuting the values in the columns.

VPN Topology: We use one of the 23 ISPs, which is also a large VPN provider. Simulation is done at the router level instead of the POP level due to the much smaller topology[4].

VPN TE-prefix Mapping: Since we have access to router's routing table dumps and configuration files for every PE in the VPN, deriving TE-prefix mapping is straight-forward. There exists a mapping between TE x and Prefix p if p is advertised in BGP by the PE x.

VPN Traffic Matrices: We also compute the prefix-level traffic flow matrices across all the PEs in our VPN topology. The element in the ith row and the jth column in a traffic flow matrix is the amount of traffic that originate from PE i and terminate at prefix j. To build traffic flow matrices, we use router-level Netflow records from customer CE routers.

B. Simulation Overview

The focus of the simulation is to understand various aspects of CRIO, in particular, the tradeoff between path length and forwarding table size under different virtual prefix deployment strategies. The two metrics we use to quantify the tradeoff is average path length (or average path length inflation) and average table size.

We implement the simulator using two separate components. The first component is the BGP simulator, which is used to simulate reachability to all TEs and virtual prefixes. The second component is a forwarding simulator that uses output from the BGP simulator to simulate the actual forwarding path. In what follows, we discuss this in detail.

The BGP simulator we used is C-BGP [1]. For the Internet experiments, we simulate inter-POP behavior only. The input to the simulator is the POP-level topology and the VP-TS placement determined by the policies. Each POP is represented as one router in the simulation. In the case of VPN, the simulations are done at PE router level. Following are the configurations we used.

Router connectivity: The connectivity between routers follows directly from the POP/router level connectivity in our topology.

Intra-domain routing configuration: Every router is configured to run OSPF. We use the *unit weight* scheme to set link weights. The path length between any two routers is measured by hop count.

[4]The number of PE routers, prefixes, and CE routers used for these experiments is omitted for privacy reasons.

TABLE I
ROUTER FORWARDING TABLE ENTRIES

Name	Stored in	Type
TE prefix	TS	BGP
Virtual prefix	TS	BGP
TE-induced prefix	TE	BGP/Mapping
VP-induced prefix	VP-TS	Mapping
Perf-induced prefix	TS	Mapping

BGP configuration: : Every router is a BGP speaker. There is an iBGP peering session between every pair of routers in the same AS. Each router announces its own address (i.e., TE address) and its assigned virtual prefixes via BGP. There is an eBGP peering session between neighboring routers that are not in the same AS. There is no specific interdomain policies: routers exchange all TE and VP reachability among their eBGP peers.

Note that we use C-BGP only to generate the converged routing tables, not to simulate BGP dynamics per se. The simulator starts from the initial route announcements and terminates when the routing system reaches convergence. Upon finishing, C-BGP will output each router's routing table, thus defining the paths to all TEs and virtual prefixes.

We simulate the forwarding path from a given router (the *starting router*) to a destination prefix as following. (i) If the starting router has a mapping in its forwarding table for the prefix, then the path is the same as the one between the starting router and the TE router (which can be determined from the output of C-BGP). (ii) If the starting router does not possess a mapping, the prefix will be matched to a virtual prefix and be routed to the corresponding VP-TS (again using the routing information from C-BGP). The VP-TS router will in turn tunnel the packet to the appropriate TE. Finally, the simulation outputs the forwarding path.

The average path length is computed by weighting the path length by the fraction of traffic the prefix carries and summing up the weighted path length over all prefixes.

Without yet getting into details, calculating table size involving summing up prefixes from various categories that constitute the routers' forwarding table. Section III-C discusses this calculation in detail.

C. Simulation Details

We start by describing the different categories of prefixes in router forwarding tables under CRIO. Then we show how various policies can change the contents of the forwarding table. In doing so, we illustrate the variables we used in the simulation to alter the table size. In the remainder of this section, for simplicity of presentation, we assume that each router serves as TE, TS, and VP-TS.

Router Forwarding Table: As shown in Table I, the prefixes that need to be stored in router forwarding table can be classified as five types: *TE prefix*, *virtual prefix*, *TE-induced prefix*, *VP-induced prefix*, and *perf-induced prefix*.

The *TE prefixes* are BGP routes to all TEs. The number of TE prefixes is a constant, the value of which depends on the network topology that is used in the simulations.

The *virtual prefixes* are of fixed size and the number of virtual prefixes is a constant for all the routers. In our simulations, we examined the use of a range of aggregates from /8 to /16 as virtual prefixes. We found virtually no difference between them, and so we only show the /8 results in this paper. The number of virtual prefixes at each router is therefore 256. Since only the virtual prefixes and TE prefixes appear in the BGP RIB, the total number of prefixes in the CRIO architecture is 1219+256, or roughly 1500. This number may certainly vary by a few hundred, but even so the number of prefixes is still roughly two orders of magnitude smaller than those in today's BGP RIB.

The *TE-induced prefixes* stored on a router are the prefixes for which that router is the TE. Similarly, if a router is the VP-TS for certain virtual prefix, the *VP-induced prefixes* stored on that router are all the prefixes for mappings that are covered by this virtual prefix. The number of TE-induced and VP-induced prefix entries varies depending on the router in question, as well as the VP deployment policy in use.

The *perf-induced prefixes*, short for *Performance-induced prefixes*, include mappings for important customers' prefixes, and prefixes carrying heavy traffic. The latter prefixes are installed in routers to optimize traffic load in the ISP. The number of perf-induced prefixes stored in the forwarding table can vary by several orders of magnitude. A router can have zero entries in this category by having all its traffic handled by appropriate VP-TS routers. On the other extreme, a router can have a full forwarding table of hundreds of thousands entries so that it has optimal tunnel paths for every prefix, even those for which it rarely forwards packets. The primary focus of our simulation is to understand the effects of the number of perf-induced prefixes on forwarding path lengths under certain VP placement policies (to be defined later).

Of the five categories of router forwarding table entries, the TE-induced, VP-induced prefixes, and perf-induced prefixes can be used to control table size. The first two are changed through VP placement policy schemes, the latter through performance optimization policies. Next, we discuss the various policies used in the simulation.

Internet VP Placement Policies: In this analysis, we examine three different policies and show how they effect the router forwarding table entries, TE-induced prefixes and VP-induced prefixes.

Rand: A simple but unrealistic policy is to randomly assign a virtual prefix to each router. The router's virtual determines the set of VP-induced prefix mappings that the router has to maintain. Furthermore, the router keeps native routes to all prefixes for which it is the TE. Notice that with this simple policy, the traffic that has both origin and destination in the same AS might nevertheless be first forwarded to a VP-TS router outside of the AS, and then tunneled back in again to the destination. The following two

policies, on the other hand, allow an ISP to keep its internal traffic internal.

Rand+TP: With this policy, a router stores routes not only for its own TE-induced prefixes, but also those of all other routers in the same AS. Another way of viewing this policy is to consider TE-induced prefixes to be customer prefixes. Every router then keeps mappings for all customer prefixes in the same AS. As a result, not only will the traffic originated and terminated in the same AS stay within the but these intra-AS paths will all be shortest path.

Rand+VP: The third scheme also forces local traffic to be routed within the AS, but in a different way from Rand+TP. We ensure that within every AS, there is at least one VP-TS that covers any given virtual prefix. This way, an internal VP-TS will always be chosen over an external one, and internal traffic will remain within the AS. Within an AS, though, VPs are assigned randomly.

VPN VP Placement Policies: The policy design for VPN experiments has a different focus than those for the global Internet experiments. Here, we define policies that take into account additional router information, such as the number of TE-induced prefixes or traffic volume. In the following placement policies, each virtual prefix has only a single router as its VP-TS.

Cust: With this policy, the VP-TS for a virtual prefix is the router that has the highest number of TE-induced prefixes within that virtual prefix. Intuitively, this means that a router will advertise a virtual prefix if many of its customer prefixes are within in that virtual prefix.

Traf: With this policy, the VP-TS for a virtual prefix is the router that sends the highest total amount of traffic to all prefixes within that virtual prefix. Intuitively, this means that VP-TS will be the router that sends the most amount of traffic to that virtual prefix.

Rand: The VP-TS for each virtual prefix is a randomly selected router.

Performance Schemes: We implement a simple performance scheme that is based solely on traffic volume. Given a router, we select its perf-induced prefixes in the following way. First, we rank all prefixes based on the volume of traffic sent to each. This information is obtained from the router's traffic flow matrix. We then pick a target volume of traffic (which we refer to as V_{traf}), expressed as a fraction of the total traffic that should be tunneled (and therefore take the shortest path). We then select the N highest-volume prefixes as perf-induced prefixes such that the target fraction V_{traf} is reached. By varying V_{traf} (and therefore N) we can tradeoff the size of the forwarding table for the volume of traffic that is routed shortest path.

D. Simulation Results

Internet Simulation Results: Figure 5 shows the tradeoff between the forwarding table size and the total fraction of traffic carried by perf-induced prefixes V_{traf} averaged across all starting routers. We vary the value of V_{traf} from 0.75 to 0.99. We observe that even small forwarding tables can result in most packets going shortest path. For example, with a forwarding table only 1/3 the size of the native BGP table, 99% of the traffic is shortest path. In addition, Figure 5 also

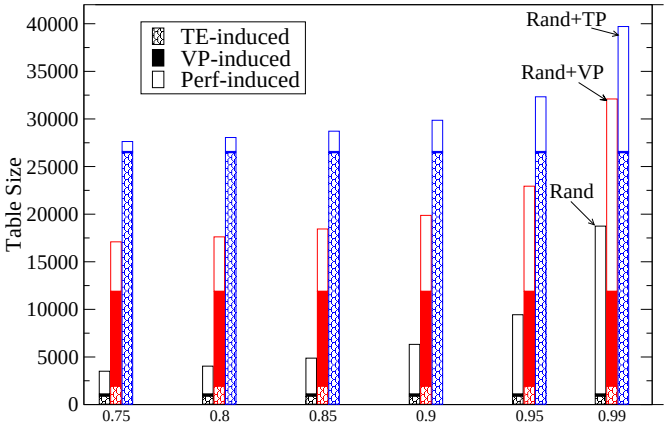

Fig. 5. Tradeoff between average table size and fraction of traffic carried by perf-induced prefixes

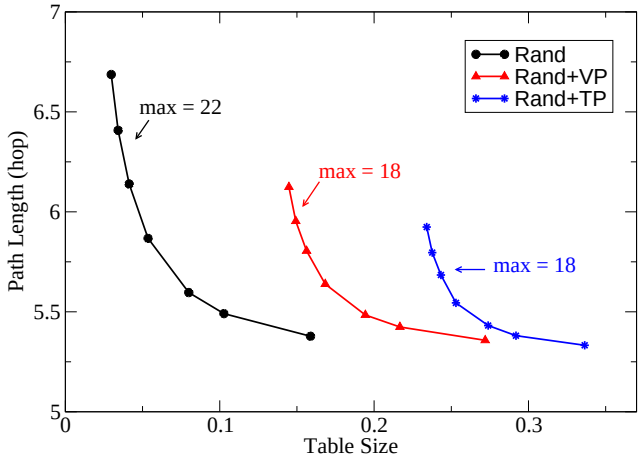

Fig. 6. Tradeoff between average path length and forwarding table size

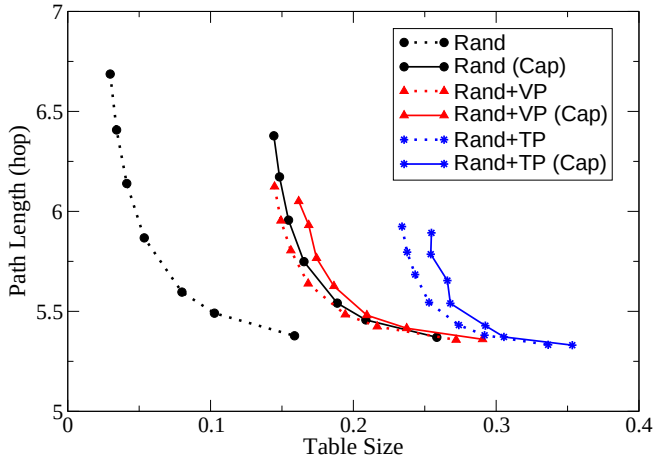

Fig. 7. Tradeoff between average path length and forwarding table size, with the maximum path length capped at 15 hops

shows the portion of the forwarding table derived from each type of prefix. For instance, in order to enforce that intra-AS traffic be routed within the AS, both Rand+TP and Rand+VP have to include additional forwarding table entries.

Figure 6 shows the average path length versus forwarding table size for the three policies. The data points shown in the figure are averaged over all routers. We make the following key observations. First, all the curves show that using virtual prefix does increase the path length. Second, the average path length converges quickly as the number of perf-induced prefixes increases. Overall, CRIO can reduce FIB size by an order of magnitude if we are willing to let intra-ISP packets travel outside of the ISP, and by roughly five times otherwise, with very little path length penalty.

The above observations hold for all three VP placement polices. However, different policies reach near-optimal path lengths at different table sizes. In particular, Rand+VP pays a table-size penalty for insuring that all intra-AS traffic stays within the AS. Rand+TP pays an additional table-size penalty for insuring that all intra-AS traffic is shortest path (even for destinations that carry very little traffic).

Though the average path length shows little increase, the use of virtual prefixes may cause certain prefixes to experience a particularly long path. In Figure 6, we indicate the maximum path length under each policy. While the maximum path is roughly three times longer than the average path, it should be understood that the POP diameter (the longest shortest path between all pairs of POPs) in our simulated topology is 15 hops. Given this, the maximum path experienced with virtual routers is not much longer than the maximum shortest path.

In spite of this, we are interested in knowing what the effect on table size and average path length is if we cap the maximum path length to the POP diameter of the topology: 15 hops. We do this by taking every prefix that has a path longer than 15 hops, and including that prefix in the set of perf-induced prefixes (i.e. adding it to the mapping table). Figure 7 shows the result. We see that for both Rand+VP and Rand+TP, capping path length increases table size by noticeable but never-the-less small amount.

The main take-aways from this simulation analysis are the following:

- The number of prefixes in the BGP RIB is reduced by nearly two orders of magnitude.
- The number of prefixes in the FIB is shrunk by roughly five to ten times with virtually no penalty in path length.

VPN Simulation Results: Figure 8 shows the CDF of forwarding table size of all of the PE routers under various VPN VP placement policies, normalized to the number of routes each PE would carry in the absence of CRIO. The left set of curves represents various VP placement policy with V_{traf} = 0, that is, no perf-induced prefixes and all packets are forwarded via VP-TS. The right set of curves are policies with V_{traf} = 1, which means that the PE keeps the mappings for every prefix to which it sends traffic (therefore

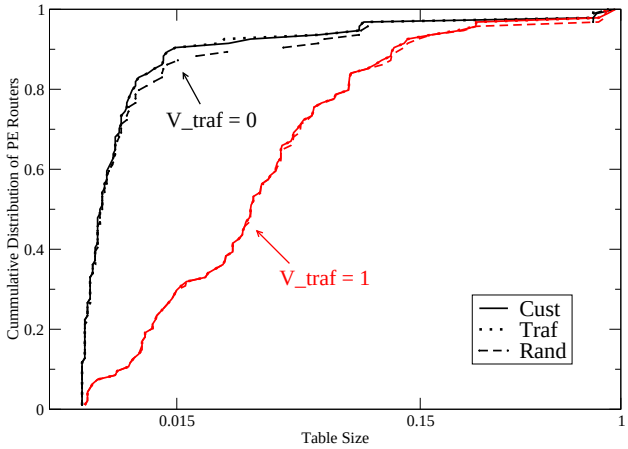

Fig. 8. Comparison between different virtual prefix placement schemes

ensuring shortest path for all traffic). Each set consists of 3 curves corresponding to 3 different policies. We make the following observations. First, both set of curves show that the forwarding table size can be greatly reduced for most of the PE routers. In the case where v_{traf} = 0, nearly 90% of the PE get a reduction by a factor of 60. Second, there are still a few PEs whose forwarding table size is close to the full size. A careful examination of the data shows that these PEs have a large number of customer links, and so must in any event carry these prefixes. For these PEs, TE-induced prefixes are dominant part of the forwarding table, which explains why the use of virtual prefixes has little effect on them.

Interestingly, the different VP-placement schemes have effect little on routing table size, as the CDF curves are almost on top of each other. This is primarily due to the fact that the majority of VPN prefixes are within a single virtual prefix. Intuitively this means that no matter which policy we use, the result is always that one or a few VP-TS routers have large forwarding tables, and the rest of the routers have very small tables. Having said that, note that Figure 8 does not reflect the fact that the policy does effect the location of the VP-TS, which in turn effects path length (at least for the v_{traf} = 0 policies).

Figure 9 shows the trade-off between table size and path length for the VPN. The forwarding table size is normalized by the total number of prefixes for the VPN. The data points shown in the figure are averaged over all PE routers. We label several data points with their coresponding V_{traf} values. First, note that the most right data point labeled with V_{traf}=1 shows that even without paying any penalty in path length, we can still shrink the forwarding table size to 7.9% of the original size. This curve has a similar property as the one for the Internet simulation, that is, the average path length converges quickly as the number of perf-induced prefixes increases. In particular, the V_{traf}=0.99 data point shows that, with negligible increase in path length, we can further shrink the average table size to 5.6%.

The results in this section have shown that CRIO is

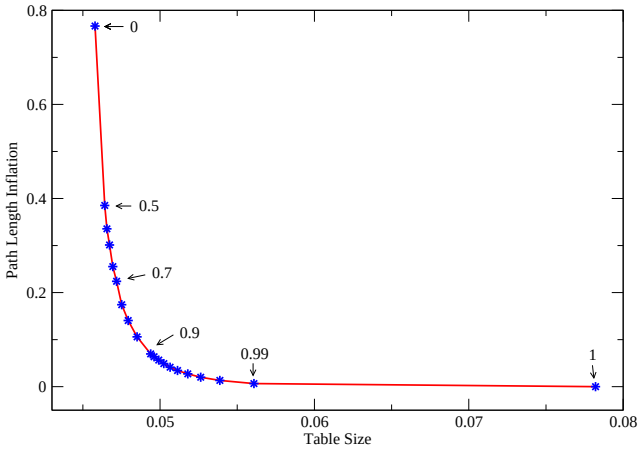

Fig. 9. Tradeoff between average path length inflation and forwarding table size

very effective at reducing table size for this specific VPN customer. Even though our simulation consists only one VPN customer, we suspect that this result will hold for most VPN customers. This is because the most VPNs exhibit a hub/spoke traffic pattern [21]. The percentage of pure "meshed" communication, where every node talks to every other node is relatively small (3%). This property makes CRIO particularly suitable for VPNs. If we deploy VP-TS at hub nodes, then the forwarding tables at spoke nodes can be greatly shrunk without any path length penalty. Furthermore, if different customers have their hub nodes at different PE routers, then the individual large tables will be spread over many PEs, and any given PE will see a large drop in its cumulative VPN table size.

IV. CRIO Dynamics

This section briefly discusses dynamic aspects of CRIO. As described in the previous section, CRIO reduces the size of the BGP RIB by nearly two orders of magnitude, while reducing the size of the FIB realistically between five and ten times. In other words, CRIO has the effect of moving much of the task of routing from BGP to the mapping tables. We argue that this shift in functionality is overall beneficial, not only because FIB sizes are reduced, but also because we suspect that distributing mapping tables is inherently simpler than distributed routing.

A key characteristic of a mapping table entry is that it is the same no matter where it appears. The same cannot be said for routing information. A given BGP entry is subtly different at different routers, for instance because the AS path may be different, or because the next hop router may be different. As a result, it is hard to look at a given BGP routing table entry and know if it is correct or incorrect: its correctness depends on the state in other routers. Assuming that one knows the correct value of a mapping table entry, on the other hand, one can determine if it is right or wrong anywhere.

BGP is notoriously hard to debug [17]. We believe that CRIO affords two benefits to BGP operation. First, there are far fewer entries. Second, the entries that remain refer to ISP POPs, and therefore will be more stable than many routing entries that appear in BGP today [9]. Note that much of the task of setting policy also shifts from the BGP domain to the mapping table domain. Though we don't go into detail here, we suspect for the same reasons that this shift is also overall beneficial.

There are any number of ways that mapping tables could be distributed. If nothing else, they could be distributed in BGP as a new BGP attribute and we would still be better off than we are today. Nevertheless, we believe that a divide-and-conquer approach, whereby a separate overlay infrastructure for the purpose of mapping table distribution is used. This avoids overloading BGP. One could use flooding similar to OSPF to distribute mapping tables. Another approach would be to use a route server [20] or Routing Control Platform [14] infrastructure. One could use a gossip style of distribution [8] or even a pub/sub approach [4]. Note that in all of these approaches, to prevent mapping entries from being polluted by the injection of bogus mappings [5], all mapping entries need to be authenticated. One can imagine a chain of certificate authorities rooted at IANA for this purpose.

V. Related Work

The basic idea of using a mapping table and tunneling to offload the core routers first appeared as an unpublished talk by Steve Deering in 1996. Steve attributes the idea to others (Bob Hinden and Robert Elz).

CRIO most closely resembles the Caida work [19] on Atomized Routing, which also proposes the use of tunneling to limit the BGP computation to the Internet core, and the corresponding mappings (which they call Declared Atoms). CRIO differs from Atomized Routing in two major respects. First, Atomized Routing does not have the concept of virtual routers—all TSs require the full mapping table. Second, Atomized Routing is designed to operate in the customer sites themselves. That is, declared atoms are originated by sites, and TSs and TEs are deployed in sites. Philosophically, CRIO departs considerably from this in that it limits all changes to provider networks. While [19] discusses an enhancement that does put TSs and TEs in provider networks, the enhancement requires considerable cooperation between providers, including those that do not otherwise have a peering relationship.

Several Internet drafts and RFCs [5], [7] have analyzed the current state of interdomain routing and proposed a set of guidelines for next-generation routing protocol. In particular, [7] examines various operational practices and discusses the impacts of these practices on the scaling of interdomain routing system.

[5]These security related concerns are in fact faced by existing routing control infrastructure on today's Internet.

Many architecture designs have also been proposed to address the Internet routing scalability and convergence problem. HLP [11] uses hybrid of link-state and path-vector protocol to provide faster convergence without sacrifying the scalability. Like CRIO, it tries to scale the routing system by providing additional layer of indirection, namely, it performs routing on the granularity of AS and have prefixes mapped to corresponding AS. IETF's Multi6 workgrouping facilitates the same design principle in the context of supporting multi-homing in IPv6 without over burdening the routing system. Unlike CRIO, its solution requires the mapping activity to be implemented on the endhost protocol stack. A different scaling architecture is along the style of replacing distributed routing computation with a centralized server. [14] and [20] are two examples of such systems.

VI. Conclusions and Future Work

This paper presents a new routing architecture, the Core Router-Integrated Overlay (CRIO), that uses tunneling and virtual prefixes to dramatically reduce both the BGP RIB (by nearly two orders of magnitude) and the FIBs due to global routing (by an order of magnitude) and VPNs (by ten to twenty times). These results were produced using the C-BGP simulation tool over a topology measured by Rocketfuel, and using a traffic model based on measured traffic at a Tier-1 ISP.

For future work, we would like to produce a detailed design of the mapping table distribution infrastructure, actually build the system, and try it under varying workloads. In so doing, we would like to explore fitting this model into next generation network control models like the Routing Control Platform (RCP) [13] and 4D network management framework [6]. We would also like to explore the use of CRIO to provide traffic engineering for multi-homed sites.

References

[1] C-BGP. http://cbgp.info.ucl.ac.be/.

[2] Netflow. http://www.cisco.com/warp/public/732/Tech/nmp/ netflow/index.shtml.

[3] Rocketfuel. http://www.cs.washington.edu/research/networking/ rocketfuel/.

[4] Design and evaluation of a wide-area event notification service. *ACM Trans. Comput. Syst.*, 19(3):332–383, 2001.

[5] A. Doria and E. Davies. Analysis of IDR Requirements and History. Internet-draft, 2004.

[6] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Meyers, J. Rexford, G. Xie, H. Yan, J. Zhan, H. Zhang. A clean slate 4d approach to network control and management. *ACM SIGCOMM CCR*, 2005.

[7] G. Huston. Commentary on Inter-Domain Routing in the Internet. RFC 3221, Dec. 2001.

[8] I. Gupta, R. van Renesse and K. Birman. A survey of gossiping and broadcasting in communication networks. *Network 18*, 1988.

[9] Jennifer Rexford, Jia Wang, Zhen Xiao and Yin Zhang. BGP routing stability of popular destinations. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment*, 2002.

[10] L. Kleinrock and F. Kamoun. Hierarchical routing for large networks: Performance evaluation and optimization. *Computer Networks, Vol. 1, pp. 155-174*.

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

[12] L.Subrmanian, S. Agarwal, J. Rexford and R.Katz. Characterizing the internet hierarchy from multiple vantage points. In *Proc. IEEE INFOCOM*, 2002.

[13] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, J. van der Merwe. Design and implementation of a routing control platform. *NSDI*, 2005.

[14] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. Merwe. The case for separating routing from routers. In *Proc. ACM SIGCOMM FDNA Workshop*, 2004.

[15] N. Maxemchuk. Routing in the manhattan street network. *IEEE transactions on communications (1558-0857), Vol.35, Iss.5*, 1987.

[16] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *Proc. ACM SIGCOMM*, 2003.

[17] Olaf Maennel and Anja Feldmann. Locating internet routing instabilities. In *Proc. ACM SIGCOMM*, 2004.

[18] P. Francis. Comparison of geographical and provider-rooted internet addressing. *Computer Networks and ISDN Systems 27(3)437-448*, 1994.

[19] P. Verkaik, A. Broido, kc claffy, Y. Hyun, R. Gao, and R. van der Pol. Beyond CIDR Aggregation. Technical report, Feb. 2004.

[20] R. Govindan, C. Alattinoglu, K. Varadhan, and D. Estrin. Route servers for inter-domain routing. *Computer Networks and ISDN Systems*, 1998.

[21] Satish Raghunath, K. K. Ramakrishnan, Shivkumar Kalyanaraman and Chris Chase. Measurement based characterization and provisioning of ip vpns. In *Internet Measurement Conference*, 2004.

[22] T. Bu, L. Gao, and D. Towsley. On characterizing bgp routing table growth. *Computer Networks*, 2004.

[23] T. Hain. An ipv6 provider-independent global unicast address format. *IETF Internet Draft draft-hain-ipv6-pi-addr-09.txt*, 2006.