# On the Feasibility of Core-Rooted Path Addressing

Cong Guo and Martin Karsten
David R. Cheriton School of Computer Science, University of Waterloo
{c8guo,mkarsten}@uwaterloo.ca

***Abstract*—Addressing, routing, and forwarding in the Internet must form a coherent architecture that satisfies user and technical requirements, such as performance, robustness, and efficiency. This paper presents the Core-Rooted Path Addressing (CRPA) architecture, which is a novel combination of mostly known architectural components. CRPA is designed as a network layer for physical topologies and combines a generic rendezvous service with the path addressing principle. The benefits of CRPA include forwarding components that are completely independent of dynamic routing, as well as compact forwarding tables. The key challenge for such a design is the overall practicality of its addressing scheme. This paper investigates the characteristics of CRPA using Internet topology analysis and simulation, and the findings indicate that it is practicable. Forwarding table sizes are quite manageable, and the frequency and scope of dynamic updates are reduced. While actual deployment might be unrealistic in the current Internet ecosystem, its straightforward design can make CRPA a valuable benchmark for systematic reasoning about network architecture.**

## I. Introduction

The current Internet addressing, routing, and forwarding architecture is based on three fundamental design decisions: 1) each network interface is assigned a single address, 2) hop-by-hop forwarding is based on that address, and 3) a variant of distance-vector routing is used at the inter-domain level. If the network topology would be a proper tree, addresses could align with the tree structure and forwarding would be trivial. However, in a non-tree graph, single-address forwarding requires dynamic routing to produce consistent forwarding tables. If the topology would be an unstructured graph, there would probably be little opportunity for improvement. However, the Internet inter-domain topology is neither a pure tree nor a completely unstructured graph: Business relationships between network providers result in a tree-like hierarchy of provider networks. The set of fully meshed core networks along with multihoming make the tree a multirooted tree. Peering links add further connectivity, but the resulting graph still has significant hierarchical structure. This generally accepted view of the Internet topology [1] has not changed since the seminal work by Faloutsos et al. [2]. IP address aggregation exploits this structure to limit the size of routing and forwarding tables. However, multihoming results in multiple paths, which fundamentally conflicts with a single address and address aggregation. For the same reasons, inter-domain multipath routing cannot be supported without significant changes or additions to the architecture. While extremely useful and robust, the above design characteristics are ultimately also responsible for challenges facing the Internet architecture:

978-1-5090-0223-8/16/$31.00 

- Multihoming results in address de-aggregation that threatens the scalability of backbone routing and forwarding by increasing table sizes [3], [4].
- Distance-vector routing computes consistent forwarding tables for hop-by-hop forwarding using distributed recursion, which can be complex and error-prone in combination with local policies [5].
- Multipath routing cannot be supported without substantial changes to the architecture [6].

Furthermore, the single-address paradigm and overloading of IP address semantics as both identifier and locator has been recognized as an architectural limitation and addressed by a substantial body of work [7], [8]. In particular, the proliferation of mobile devices requires agile yet scalable identity/locator mapping services.

This paper proposes and evaluates *Core-Rooted Path Addressing* (CRPA) to address the above challenges. Compared to the current Internet architecture, CRPA better matches the actual inter-domain topology and reduces the overall system complexity. Scalability and performance are improved by moving functionality from on-path components to end systems and an off-path lookup service. However, it is not obvious whether CRPA would be ultimately practicable. The contributions of this paper are two-fold: 1) combining independently known design concepts into a novel and coherent architecture, and 2) evaluating key questions raised by this and other proposals by studying current and historical Internet topology data. The results indicate that CRPA is scalable, forwarding tables are compact, and the frequency and scope of dynamic updates are reduced. In contrast, alternative internetwork architectures in the literature are either designed as overlay networks, or are not as thoroughly verified with respect to the actual Internet topology.

The rest of the paper is organized as follows. Related work is surveyed in Sec. II to provide background and context. Sec. III presents the detailed CRPA architecture, including the requirement for an off-path rendezvous service. The major challenges for CRPA are evaluated and assessed in Sec. IV. This is followed by a discussion of additional properties of CRPA compared to the existing Internet architecture in Sec. V. The paper is wrapped up with a brief conclusion in Sec. VI.

## II. Background and Related Work

The BGP path-vector algorithm makes no particular assumptions about the topology of the network graph. While this is the most general and robust approach, it cannot exploit specific features of the Internet topology, which is inherently

hierarchical [2], [9] with a majority of traffic using so-called *valley-free* paths. This fundamental structural characteristic of the Internet topology has not changed since the commercialization of provider networks [1]. It presumably reflects inherent properties of business relationships and value chains between independent network providers in a federated global network and can thus be taken into account for architecture proposals. For example, the Hybrid Link-state Path-vector routing protocol (HLP) [10] leverages the hierarchical structure of the Internet by segmenting the topology into multiple hierarchies and limiting update propagation between different hierarchies. On the other hand, the combination of hop-by-hop forwarding, path-vector routing, and local policy settings is notorious for its complexity, instability, and/or faulty outcomes. This has sparked substantial research efforts (cf. [11] and references therein), but the absence of a proper solution might be indicative of inherent shortcomings in the underlying architecture.

In theory, Internet routing computes "optimal" least-cost paths to efficiently utilize the available topology. However, in practice, the shortest-path metric is merely used as a global criterion to ensure that the distributed route computation converges to loop-free forwarding tables, while network providers use local policies irrespectively of global optimization. In general, selfish routing by end systems decreases the overall network efficiency somewhat [12], but it has also been shown to perform reasonably well in Internet-like environments [13] with certain caveats that are addressed by the balanced approach of CRPA. It has been shown that selfish routing in combination with max-min fair congestion control leads to stable routing decisions at good efficiency [14]. Thus, selfish source routing is a relevant alternative to destination-based hop-by-hop forwarding. Previous work on routing scalability and complexity advocates for some form of source routing [15], [16], [17], i.e., transferring control from network providers to end systems. It has been recognized before that source routing provides substantial benefits in terms of flexibility and traffic engineering [18], but most traditional source routing approaches are based on an independent global addressing scheme and require knowledge of the complete network topology to construct a path on demand. Sources must maintain all the path information through a route computation service [19]. In contrast, CRPA encodes path information directly in addresses, so that both source and destination nodes can control part of the path information.

Given the density of network topologies, multipath forwarding would allow utilizing network resources more efficiently [6]. Link-state routing can facilitate multipath forwarding under certain circumstances [20], but at the inter-domain level, multipath forwarding typically requires changes to the addressing system, or tunneling [21]. Proposals for multipath forwarding introduce multiple addresses and some form of source routing [16], [22]. Standardization proposals for decoupling identifiers from locators also effectively introduce multiple addresses per entity [7], [8].

Nimrod [15], NIRA [16], Pathlet routing [22], and SCION [23] all combine design elements of multiple paths and source routing, similar to CRPA. However, all these proposals still use dynamic routing, while the focus of CRPA is to explore the limits of a strictly addressing-based forwarding scheme. SCION does not directly encode the relationship hierarchy in the forwarding paths, and requires an additional service to find shortcuts like a common ancestor provider or a peering link. NIRA makes peering links available for source routing, which requires dynamic routing and might be contrary to business models. CRPA uses a balanced approach, such that the relationship hierarchy is encoded in addresses, while network providers can locally control the forwarding along peering links. CRPA is conceptually much simpler than NIRA, but requires a more comprehensive evaluation, which is done using the wealth of information available from Internet topology databases. In contrast to other previous work, the compact path addresses in CRPA usually do not result in higher header overhead than IPv6. In addition, CRPA is designed as a hypothetical replacement of the IP architecture, instead of an add-on overlay like Pathlet. Thus, CRPA must operate under the stricter assumptions of a physical topology, and cannot assume a global virtual graph.

## III. Core-Rooted Path Addressing

The key concept of Core-Rooted Path Addressing (CRPA) is that an address directly represents a path from the core of the Internet towards a node. Thereby, CRPA supports multipath and source routing and does not require any dynamic routing system, because path addresses are advertised to end systems.

### A. Addressing

CRPA assigns multiple path addresses to each network by strictly following the provider/customer hierarchy, similar to NIRA. The scheme starts out by assigning unique labels as top-level prefixes to all core networks in the *default-free zone* (DFZ) through a centralized authority, such as IANA. These networks are fully meshed by definition and can thus directly exchange packets using their respective prefixes. Networks enumerate their internal nodes and customer networks and assign suitable local labels. The concatenation of provider prefix and customer label forms the prefix for a customer network. A network with multiple providers thus might receive multiple prefixes. Further down the hierarchy, a network might then receive a set of prefixes from each of its providers. Each resulting address is a unique sequence of labels that forms a locator encoding a path starting at a core network. Dynamic prefix allocation can be done with a DHCP-type protocol between each pair of provider and customer network. Addressing is illustrated in Figure 1. Network $V$ is multihomed and has two prefixes.

For clarity of presentation, addresses are shown as sequence of labels in dot notation. In principle, neither the size nor the number of labels in an address have to be globally uniform. In particular, the CRPA scheme can also be used to enumerate transport protocol instances, thus eliminating the need for port numbers as a separate address space. However, for high-speed packet processing, it is important to establish fixed-size

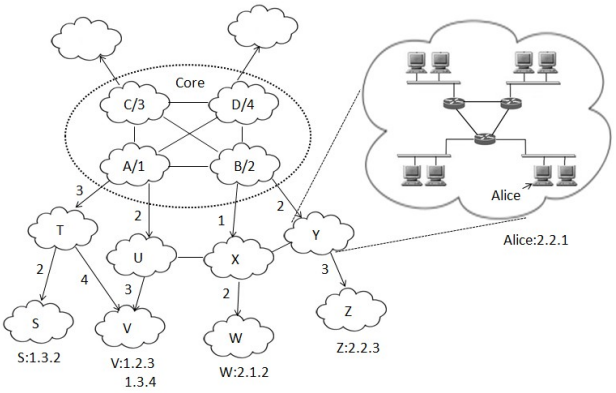


Fig. 1: Address Allocation

addressing fields in a packet header. The evaluation presented in Sec. IV-A1 confirms that 128 bits (equal to an IPv6 address) are sufficient to cover an overwhelming part of the Internet topology with CRPA. For fringe regions at the edge of the Internet, optional header extensions can be used to encode additional labels. These extensions only need to be processed at the edge, presumably at moderate line speeds.

### B. Peering

No address allocation takes place across peering links. Instead, peering links are used implicitly in the forwarding process (see next section for details). This is an important difference from NIRA, which uses a dynamic routing protocol to discover peering links and assigns explicit prefixes to them. However, network providers typically consider their network structure as business-critical proprietary information. Therefore, CRPA employs a more balanced approach to topology transparency. Provider/customer relationships are visible in the inter-domain topology and can be inferred with a high success rate through topology analysis [24] already. On the other hand, peering relationships provide a crucial ingredient for the design of a provider network and are more difficult to infer indirectly. Correspondingly, CRPA directly exposes hierarchical relationships through its addressing scheme, but it does not mandate the advertisement of peering links outside of the participating networks. A technical side effect of this design decision is that it obviates the need for the link-state routing protocol and topology database proposed for NIRA.

In particular, if two networks are connected via a peering link, they advertise a subset of their respective network prefixes (depending on policies) to the peering partner. For example, in Figure 1, Network $U$ might advertise its Prefix 1.2 to Network $X$ using the peering link between $U$ and $X$ and vice versa for 2.1. In pure CRPA, a peering prefix is not re-advertised across another peering link, i.e., in Figure 1 Prefix 1.2 is not re-advertised from Network $X$ to network $Y$. However, a possible exception is discussed in Sec. V.

### C. Forwarding

Forwarding in CRPA essentially follows the valley-free model [9], but supports exceptions as explained in Sec. V. The forwarding process, shown in Algorithm 1, is comprised of 3 stages covering the upstream, peering, and downstream part of a forwarding path. A forwarding node maintains 3 tables with forwarding-related state: 1) *own* network prefixes, 2) *peering* network prefixes, and 3) local and customer *labels*.

**Algorithm 1** Packet Forwarding

$src$: source address
$dst$: destination address
$own$, $peering$, $labels$: state tables

1: $prefix \leftarrow own$.match($dst$)
2: **if** valid($prefix$) **then**
3: $\quad nextlabel \leftarrow$ getnextlabel($dst$, $prefix$)
4: $\quad nexthop \leftarrow labels$.get($nextlabel$)
5: **else**
6: $\quad nexthop \leftarrow peering$.match($dst$)
7: **if** invalid($nexthop$) **then**
8: $\quad$ **if** upstream-source-routing **then**
9: $\quad\quad nexthop \leftarrow own$.match($src$).provider
10: $\quad$ **else**
11: $\quad\quad nexthop \leftarrow$ default provider

In the first stage (Line 1), it is determined whether any of the node's own network prefixes match the destination address. If yes, the packet is in its downstream phase and the next label in the destination address is used to make a forwarding decision (Lines 3,4). If not, the destination address is compared to the various peer network prefixes and if a match is found, it is used for forwarding (Line 6). If this is not the case, the packet is in its upstream phase. During the upstream phase and depending on policies, the upstream path can be controlled via the source address (Line 9) or a default provider network is chosen (Line 11). The packet exchange between core networks can be regarded as traversing a peering link (Line 6).

The key evaluation metric for this addressing and forwarding scheme is the structure and size of the 3 forwarding state tables, which are shown to be compact in Secs. IV-A. Another important benefit of CRPA, explained in Section V, is that the update frequency of these tables is significantly reduced compared to tables in the current Internet.

### D. Path/Address Selection

In CRPA, topology discovery takes place during address allocation by assigning multiple path addresses to each node. This provides end systems with an opportunity to actively choose from multiple forwarding paths where available. A lookup system is necessary to manage the association between a node identity and its path address(es). This service is not unlike the current DNS or other (application-level) directory services. In particular, the number of entities that need to be indexed is the same as in the current Internet architecture, but each index is typically associated with multiple values. The term *rendezvous service* (RS) is used in this paper to refer to a service that associates a node or service identity with multiple path addresses.

End systems or network proxies report path addresses to the RS. Senders select the forwarding path by choosing a source path address from their local addresses and a destination path address from the RS. Path metrics can be attached to the path addresses to facilitate traffic engineering (cf. Sec. V). An ancillary benefit of a generic RS is that it can subsume multiple naming services and facilitate architectural evolution. An identity in the RS may refer to a service, a content object, or any other concept, because the flexible addressing scheme of CRPA can also comprise transport addressing.

### *E. Rendezvous Service*

The basic functions of the rendezvous service (RS) are: 1) providing identity-to-address mappings for end systems to establish sessions, and 2) propagating network state updates to end systems to adjust forwarding paths, if necessary. The first function is straightforward and implemented in DNS and many other existing services. The dynamics of the second function are similar to mobile registration services, such as home location registers [25], or home agents [26], as well as other locator/identifier services when used for mobile end systems, such as HIP [8] or LISP [7], [27]. In addition, previous work has shown that it is possible to implement a much more agile naming service than DNS [28]. Given that some of these services are deployed at a significant scale already, the RS functionality should be quite feasible. Therefore, we only sketch the conceptual design of a possible RS here.

The RS, illustrated in Figure 2, is a hybrid structure containing elements of a content-delivery network (CDN) and DNS. It is comprised of three conceptual layers. The middle layer is formed by rendezvous and replication servers. The actual identity-to-address mappings are stored at rendezvous servers, while recent copies of the mappings are cached at replication servers. A distributed index system is used to organize the identity name space and thus rendezvous servers, like higher-level name servers organize the domain name space in the DNS system. The rendezvous server for an identity is expected to be controlled and published by the owner of the identity. Similar to a CDN, its replication servers are also registered in the index system. In the bottom layer, clients or proxy caches first query the distributed index system to find the rendezvous server or a nearby replication server for a particular identity. That server is then queried for the available path addresses, one of which is used to establish the communication session.

The index system is independent of the network topology or forwarding policies, therefore clients can aggressively cache the addresses of rendezvous and replication servers. The structure and bootstrap of the index system is like DNS. An existing design and prototype like LISP-TREE [29] can be used for the index system. However, the actual identity-to-address mappings stored in rendezvous servers can be more volatile, because topology and policy changes must be captured using dynamic updates.

The RS employs a replication mechanism for address mappings to reduce server load and client-side lookup latency. In DNS, clients can cache results autonomously with a time-to-live (TTL) hint from the server, because mappings are relatively static overall. In contrast, the RS for CRPA must propagate address updates fast, because they potentially reflect changes in the network topology. The semi-managed replication servers and autonomous proxy caches can handle these updates resulting in moderate message load without resorting to a more complex mechanism. This design is assessed in Sec. IV-B.

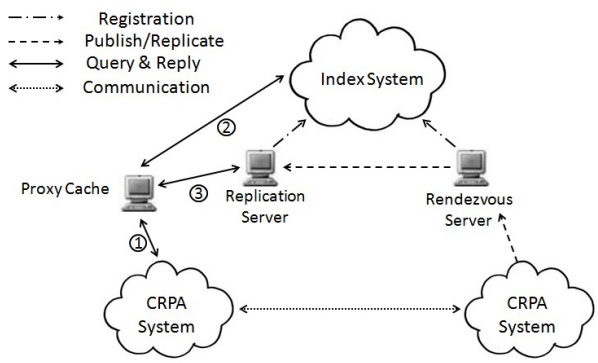


Fig. 2: Rendezvous Service (RS)

### *F. Topology Changes*

Topology events are caused by either transient link failures or permanent topology changes. Link failures that cause network partitioning cannot be handled under any circumstances and are not considered here. When a failure is detected, affected networks notify their respective authoritative rendezvous server to suspend the failed addresses. These updates are forwarded to replication servers in a timely fashion. If a sender detects the failure of an ongoing session and has other path addresses of the destination available, it can try other paths first. Otherwise, it sends a specially marked request to the RS, which triggers an immediate refresh of cached address mappings at replication servers, regardless of the TTL counter.

Permanent topology changes require address allocation changes and corresponding RS updates. Updated address prefixes are distributed to customer networks via the regular address allocation mechanism. Affected networks in turn withdraw old addresses and announce their new addresses in the rendezvous service. A change that affects a peering link is handled locally by sending appropriate advertisement messages to peers, but peering prefixes are not re-advertised. A key advantage for CRPA is that update latency is not as critical as for BGP convergence, because topology changes are often pre-arranged with a lead time, while failures can also be handled by end systems' path selection.

## IV. Evaluation

This section assesses the basic feasibility of CRPA. Topology analysis is used to show that both the size and number of CRPA addresses are limited, while forwarding tables are small and compact. Also, the size of the $peering$ table in Algorithm 1 is manageable. Finally, simulation experiments are used to assess the effects of topology dynamics and the resulting updates in comparison to BGP routing.

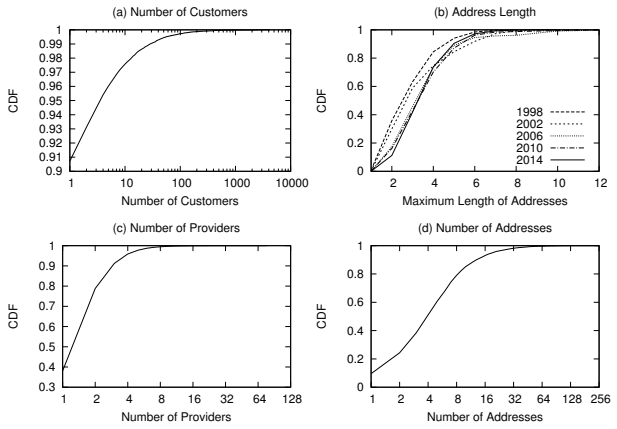


Fig. 3: Address Allocation - Simulation

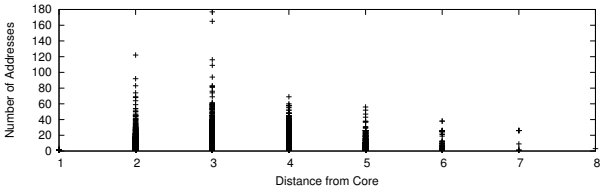


Fig. 4: Number of Addresses by Distance from Core

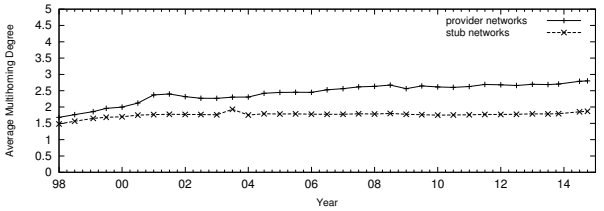


Fig. 5: Evolution of Multihoming

### A. Addressing and Forwarding

While the design of CRPA is not necessarily tied to the current notion of *autonomous systems* (AS) in the Internet, the AS-level topology is the closest approximation of a "typical" planet-scale communication network. Thus, the AS topology is used to simulate CRPA address allocation to evaluate its feasibility. This study is based on the CAIDA AS relationships dataset from October 2014 [30]. The topology data comprises 46120 ASes and 172275 links.

Some inferred downstream paths appear to have more than 20 hops, although the actual maximum AS path length in the routing data is 12 hops, while the average AS path length is 4.17 [31], [32]. This distribution of AS path lengths has been largely stable during the last 12 years [1]. Two paths may be joined and appear as a long path in the inference results, but the inferred path is not actually used. These long paths are not completely removed from the data set, but filtered out during the address allocation simulation as follows: An *AS triple* is defined as a set of two consecutive links $X \rightarrow Y \rightarrow Z$. If an inferred AS triple $X \rightarrow Y \rightarrow Z$ is not found in the actual BGP routing data [31], [32], Y does not allocate to Z its prefixes from X.

*1) Size and Structure of Addresses:* CRPA does not mandate a specific size of labels that make up network prefixes. However, to assess the size of addresses (and for operational simplicity) it is useful to think of an address as a sequence of fixed-size labels. The label size is directly related to the number of customers for each provider network, the distribution of which is shown in Fig. 3(a). 85% of ASes are stub ASes, which have no customer. The average number of customers is 13, while the maximum number is 4232. Thus, even with the inclusion of internal nodes, a label size of 16 bits seems sufficient to cover most cases. If insufficient for a large internal network, operators can use an extra level of labels to enumerate internal nodes. The label size implies the size of the $labels$ table in Alg. 1, which can be implemented as a compact array.

The number of labels in an address depends on the depth of the AS relationship graph. A simulation of CRPA address allocation on the approximated Internet AS graph at various points in time is shown as the resulting distribution of address lengths in Fig. 3(b). The historical Internet graphs are also from the CAIDA dataset [30]. The results confirm that the vast majority of AS addresses require 8 labels or less and, more importantly, the AS graph does not expand substantially over time. Therefore, a fixed-size header field of 8 labels with 16 bits each, resulting a total address length of 128 bits, can cover an overwhelming majority of the Internet graph. As discussed in Sec. III-A, this can be enhanced by an optional header part that would only be processed in fringe regions of the Internet. Alternatively, network operators are free to choose arbitrary label sizes, as long as all assigned prefixes are unique.

*2) Number of Addresses:* The number of prefixes allocated to each network is an important metric, because it determines the size of the $own$ table in Alg. 1 and also ultimately dictates the number of values that need to be managed in the rendezvous service. The number of prefixes essentially depends on the degree of multihoming along the paths from the Internet core to each network. Fig. 3(c) shows the distribution of the number of providers. Although 62% of the ASes have more than one provider, the 91st percentile is only 3 providers. Simulated address allocation gives the number of addresses that are allocated to each network and the result is shown in Fig. 3(d). 90% of ASes receive less than 13 prefixes, the 99th percentile is 42 prefixes, and the largest number is 213.

Assuming a uniform multihoming degree, it could be surmised that the number of allocated addresses increases towards outer regions of the network graph. This is investigated by breaking down the number of addresses in relation to the length of a network's shortest AS path (*distance*) to the Internet core, which turns out to be six at most. Fig. 4 shows that the number of addresses does not increase with increasing distance from the core. To understand the historical perspective and to extrapolate future trends, the evolution of the average multihoming degree over time is shown in Fig. 5, again based on the CAIDA dataset. The degree of multihoming does not increase rapidly and thus the number of addresses will likely also not increase dramatically in the future.

*3) Peering:* The size of the $peering$ table in Algorithm 1 is determined by the number of peer networks in the inferred

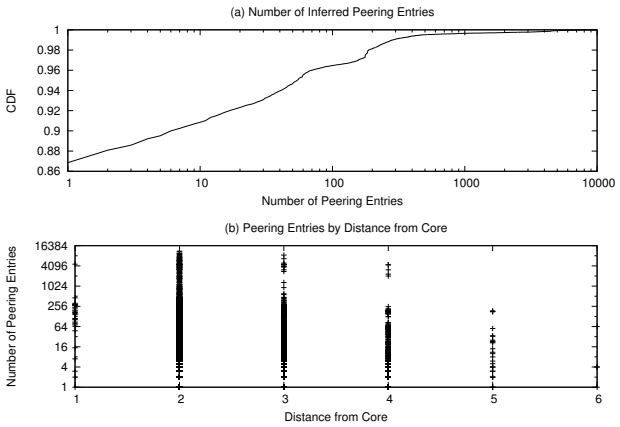


Fig. 6: Peering Entries - Simulation

topology. In the resulting distribution of the number of peers, more than 85% of ASes have no peer, the 95th percentile is 8 peers, and the maximum number of peers is 2891. However, the *peering* table size depends on the number of peering entries, which is the accumulation of address prefixes from all the peers and shown in Fig. 6(a). The 95th percentile is 54 peering entries. The breakdown by AS path distance from the core in Fig. 6(b) shows that the largest number of peering entries is found in networks that are close to the core, which most likely operate high-end packet forwarding infrastructure.

The topology inference results are based on public BGP routing data and are known to miss a number of peering links [33], especially between lower-level ASes. The impact of missing links is assessed by estimating the relative number of peering links missing. DIMES [34] is a distributed research project to study the structure and topology of the Internet with the help of a volunteer community. DIMES agents are diversely spread over the Internet and observe AS links that are only propagated locally due to export rules and thus not seen by BGP monitors, such as peering links. For the purpose of this study, links appearing in both DIMES data and IXP participant data [35], [36] are considered relevant peering links. Because the DIMES project has not published new data since early 2012, this is done on topology data from January 2012. It turns out that the inclusion of DIMES data increases the overall number of peering links by only about 7%.

Recent literature claims that inference based on publicly available data can typically reveal at most 30% of the actual peerings at IXPs, while in fact, 67% of all possible AS pairs at a typical IXP form a peering relationship [37]. Therefore, the address simulation described in Sec. IV-A1 is extended to estimate the impact of peering links missing from public data: Using member lists from multiple IXPs of various sizes, a random sample of 67% of all the possible AS pairs at each IXP are assumed to form a peering relationship.The number of additional peering entries is the sum of prefixes that need to be added per peering relationship. The minimum, median and maximum numbers of additional peering entries for each IXP are shown in Fig. 7. The general trend is that the number of additional entries increases with the size of the

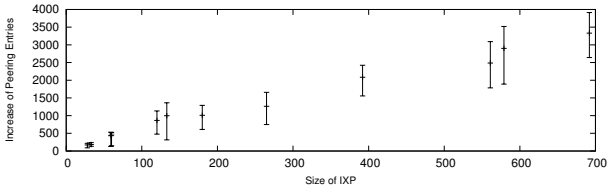


Fig. 7: Additional Private IXP Peering Entries

IXP. According to this estimate, joining an IXP with more than 500 members creates about 2500 additional peering entries. There are only four IXPs in the world having more than 500 members today [36] and most IXPs have fewer than 100 members. On average, when an AS adds a peering link at an IXP, it receives about 8 additional peering entries. Ultimately, IXP peerings do not result in an explosive growth of peering entries, because most IXP members are at high layers of the addressing hierarchy and most of them have only tens of address prefixes to advertise to peers, as shown in Figure 4. Thus, the overall estimated size of peering tables is still manageable. In summary, even if the number of peering entries at high-end backbone routers surpasses the above estimate by an order of magnitude, a contemporary 20 Mbit TCAM [38] is still sufficient to store all those entries.

A recent trend of a "flattening" Internet topology is reported in the literature [39]. Such a trend would challenge any architecture that employs hierarchical addressing. However, hierarchy is still an important property of the Internet topology [40] and the topology is far from fully meshed. In fact, both studies report that a majority of peering traffic flows to and from content providers, rather than between transport providers. Therefore, we conjecture that these observations do not affect the basic assumption for CRPA.

### *B. Dynamics*

Aside from static topology characteristics that determine the inherent scalability of the addressing scheme and forwarding tables, it is important to understand the dynamic behavior of CRPA in response to topology updates, preferably in comparison to the current architecture. Typical metrics to assess these dynamics include update latency, message load and affected region. The update latency of BGP describes the convergence of the distributed route computation as observed at various routers, while in CRPA address updates are propagated through the RS and latency is observed at replication servers. Message load is not directly comparable either, because BGP nodes operate on the forwarding path and thus BGP message processing is relatively more expensive than CRPA's message processing in off-path servers. The region affected by a dynamic event in BGP consists of all the networks receiving update messages. In CRPA, the affected region also includes the networks that transfer update messages to rendezvous servers and replication servers. However, a somewhat indirect comparison with BGP still helps with assessing CRPA.

An initial assessment of CRPA's update behavior can be gleaned by studying the so-called *customer cones* [41] in the inferred Internet topology. In the inferred topology, 99% of

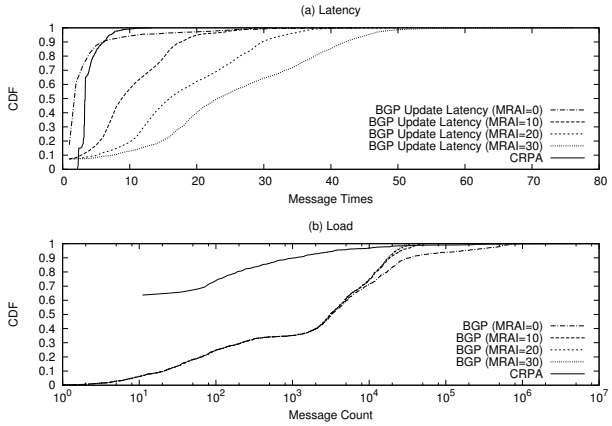


Fig. 8: Topology Updates

the networks have less than 45 direct and indirect customers. Thus, it is reasonable to expect that the number of update messages and affected nodes in CRPA is fairly limited.

A simulation experiment is carried out using an existing BGP simulator [42] and a simple CRPA simulator implemented for this project. A topology generator, which is included in the BGP simulation package [42], is used to generate topologies resembling the Internet AS graph with relationship annotations. For BGP, each node in the graph represents one AS with one IP prefix, while for CRPA it represents one network with possibly multiple CRPA prefixes. The experiment has been repeated several times using different topologies. Because the results are fairly consistent, only one example result is presented here, based on a network graph with 10000 nodes and 50387 links. During the experiment, 1000 provider-customer links to multihomed customers are randomly picked to fail and each failure is handled independently. The simulation records the message events that are triggered by the link failure.

In a real-world CRPA deployment, the placement of replication servers would be determined by each user of the RS based on their own requirements. For this experiment, a simple greedy algorithm [43] is used to decide the placement of replication servers. For the topology used here, replication servers are placed at 10 network nodes chosen according to this algorithm. The update latency is taken as the average time period between link failure and arrival of the update notification at all replication servers. Because push updates are used in this CRPA simulation, it represents the best case latency for CRPA. However, as discussed in Sec. III-F, update latency is less of an issue for CRPA in general.

In case of BGP, the update latency is measured as the average time period between link failure and restoration of a usable path, i.e., routing convergence, as observed at all routers affected by the failure. The latency is measured in unit message transmission times. BGP's minimal route advertisement interval (MRAI) is configured using the same time unit.

Fig. 8(a) presents the CDFs for the average update latency across all simulated link failures for CRPA and BGP with various settings of MRAI, which is an important BGP parameter that is notoriously difficult to configure [44]. Only with an (unrealistic) MRAI setting of 0, BGP propagates updates for most events faster than CRPA. Fig. 8(b) shows the corresponding CDFs for the number of messages in CRPA and BGP respectively. It is obvious that CRPA produces far fewer update messages for a majority of events. In addition, for $87\%$ of the link failures, BGP affects more nodes in the network than CRPA. For more than $52\%$ of the failures, BGP even affects 100 times more nodes. In BGP, $31\%$ of link failures are visible for more than half of the nodes. In comparison, $65\%$ of link failures trigger updates to less than 12 nodes in CRPA.

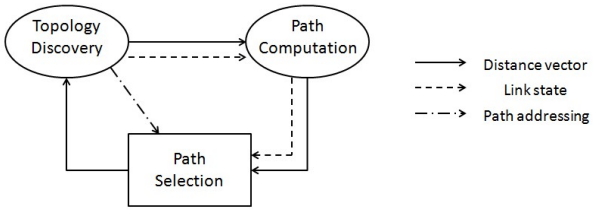


Fig. 9: Routing Components

The reason for the fewer updates in CRPA is that update messages are sent only to customers, peers, and rendezvous servers. Hence, if an event occurs close to the edge of the topology, very few messages are generated and propagated to a small region. In BGP however, because of the distributed recursive path computation, messages may be propagated to a large region regardless of their origin. In summary, even with the fundamental caveats of comparing BGP and CRPA, these results corroborate that CRPA causes significantly less message overhead than BGP, while resulting in competitive update latencies.

## V. Discussion

The analysis and simulations reported in the previous section confirm the conjecture that CRPA is basically feasible and beneficial. The forwarding procedure is simple and the state complexity of the various forwarding tables would be low. Topology dynamics can be handled with reduced overhead compared to the current BGP-based architecture for the majority of topology events. Furthermore, the basic inter-domain structure of the Internet is stable and appears to be an inherent characteristic of a federated network of networks. In this section, we sketch additional properties of CRPA to illustrate its potential - much of which needs verification in the form of future research though.

**Routing:** Aside from trivially enabling multipath forwarding, the advantage of CRPA becomes apparent when considering fundamental routing components and their interaction, as illustrated in Figure 9. Determining and establishing forwarding paths can be separated into three generic components: topology discovery, path computation, and path selection. Of those components, path selection plays a central role in the comparison, because the resulting forwarding state is read-accessed for each packet that is transmitted. Therefore, reducing forwarding state complexity and churn improves efficiency.

With path-vector routing, each topology event (transient or permanent) triggers a recursive distributed computation that includes several rounds of topology discovery, path computation, and path selection at multiple nodes, until all nodes converge to a new consistent forwarding state. Link-state routing does not have this convergence phase, but topology updates are broadcast to all nodes. With CRPA, topology changes result in address updates, which only propagate downstream from the changed link, and ultimately to the RS. CRPA builds trivially loop-free paths and eliminates the complexity of a separate distributed routing process, but this comes at the price of increased complexity in the RS. However, the RS state is only read-accessed once per session or during topology events and can be deployed outside the forwarding path.

**Traffic Engineering:** Business relationships are encoded in the addressing of CRPA. Networks can apply policies other than basic relationships when they determine which prefixes to propagate to customers and peers. Link metrics can be propagated and aggregated during address allocation along with address prefixes. In terms of link metrics and their aggregation, CRPA is a hybrid between traditional distance-vector and link-state routing. CRPA is scalable and supports information hiding, similar to distance-vector routing. However, the path metric computation is direct and non-recursive.

When end systems publish their path addresses to the RS, they can also apply local policies and attach path metrics, for example to influence the forwarding path of inbound traffic. Sources select upstream paths based on their own policies and path metrics from providers. They can follow the destinations' preferences when selecting destination addresses, or impose their special policies. In that sense, CRPA provides a level of information and control that is commonly associated with link-state protocols and source routing. However, CRPA does not incur the cost of the link-state broadcast and in fact, does not require any dynamic routing at all. Current BGP mechanisms are extremely cumbersome and error-prone [5], because of their inherent global scope. One caveat is that local forwarding policies might inherently conflict and prevent global routing consistency or stability. Because CRPA shifts path selection to end systems via the RS, these problems are largely eliminated from the architecture. In fact, any advanced method for constraint-based routing requires the level of information and control in CRPA to utilize resources efficiently through multipath forwarding [45]. In contrast, meaningful inter-domain traffic engineering is impossible in the current Internet, because inter-domain routing is restricted to single-address and distance-vector routing. With CRPA, it seems promising to investigate proposals that are currently aimed at intra-domain scenarios only. By providing a simple and feasible implementation of sender-based path selection and multipath forwarding, CRPA makes traffic engineering approaches possible that were previously deemed out of reach for inter-domain consideration.

**Deployment:** CRPA is tuned for valley-free transmissions, but the consideration of an exception illustrates how CRPA could be incrementally deployed alongside the current single-address paradigm. Certain inter-domain arrangements are termed *complex relationships* [10] in the current Internet. A complex relationship is characterized by forwarding traffic in violation of the valley-free paradigm, such as traffic destined for other peers or even provider networks. The most common example of a complex relationship is a sibling relationship, where two ASes do not use routing filters between each other. In CRPA, for example, in Figure 1 network $X$ might re-advertise its peering prefix $1.2$ to network $Y$. Another example is an inter-AS relationship where a customer provides selective transit service to a provider. For example, in Figure 1, if network $U$ provides transit service between its provider $A$ and its peer $X$, the forwarding path between $S$ and $X$ would be $S-T-A-U-X$.

In CRPA, such relationships require the re-advertisement of destinations, similar to advertising local prefixes across a peering link. In the previous example, $U$ would advertise the prefix for $X$ to $A$ and $A$ would add this prefix to its $peering$ table. If such non-default prefixes are re-advertised to other networks, this might eventually lead to forwarding loops and/or networks having to choose from multiple possible paths. Ultimately then, such advertisements need to become proper routing messages and need to be processed using regular distributed routing logic to compute consistent forwarding rules. However, this challenge is actually a blessing in disguise. Re-advertised non-local prefixes essentially represent destination addresses and this demonstrates an opportunity for incremental deployment of CRPA. In fact, using this technique, addresses outside the CRPA addressing hierarchy could be assigned, resembling current destination addresses, and treated as peering links during forwarding. Consequently, the existing destination-based architecture can regarded as a special case of CRPA, where only peering tables are populated, and managed by conventional single-address dynamic routing protocols.

## VI. Conclusion

This paper studies Core-Rooted Path Addressing (CRPA) as an architecture for addressing, routing, and forwarding in the Internet. The main advantage over the current single-address BGP-based architecture is the elimination of distributed computations that affect the forwarding state of each router. Instead, complexity is shifted to an off-path rendezvous service (RS) that is fundamentally not more complex than DNS. Compared to NIRA, providers retain more privacy of their business-critical topology and interconnection information. The main contribution presented in this paper is a comprehensive topology analysis and simulation to confirm the basic feasibility of CRPA. While being a clean-slate proposal, CRPA could be used in a hybrid deployment alongside traditional single-address forwarding, as sketched in the paper.

## Acknowledgements

This work is supported by the Natural Sciences and Engineering Research Council of Canada.

## REFERENCES


[1] A. Dhamdhere and C. Dovrolis, "Twelve Years in the Evolution of the Internet Ecosystem," *IEEE/ACM Trans. Netw.*, vol. 19, no. 5, pp. 1420–1433, Oct. 2011.

[2] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," in *Proc. ACM SIGCOMM*, 1999, pp. 251–262.

[3] "BGP Routing Table Analysis Reports," http://bgp.potaroo.net, Accessed August 2015.

[4] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," *IETF RFC 4984*, Sep. 2007.

[5] N. Feamster, R. Johari, and H. Balakrishnan, "Implications of Autonomy for the Expressiveness of Policy Routing," *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1266–1279, Dec. 2007.

[6] J. He and J. Rexford, "Toward Internet-wide Multipath Routing," *IEEE Network*, vol. 22, no. 2, pp. 16–21, Mar. 2008.

[7] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)," *IETF RFC 6830*, Jan. 2013.

[8] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture," *IETF RFC 4423*, May 2006.

[9] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, Dec. 2001.

[10] L. Subramanian, M. Caesar, C. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica, "HLP: A Next Generation Inter-domain Routing Protocol," in *Proc. ACM SIGCOMM*, 2005, pp. 13–24.

[11] A. Wang, L. Jia, W. Zhou, Y. Ren, B. T. Loo, J. Rexford, V. Nigam, A. Scedrov, and C. Talcott, "FSR: Formal Analysis and Implementation Toolkit for Safe Interdomain Routing," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1814–1827, Dec. 2012.

[12] T. Roughgarden and E. Tardos, "How Bad is Selfish Routing?" *J. ACM*, vol. 49, no. 2, pp. 236–259, Mar. 2002.

[13] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, "On Selfish Routing in Internet-Like Environments," *IEEE/ACM Trans. Netw.*, vol. 14, no. 4, pp. 725–738, Aug. 2006.

[14] D. Yang, G. Xue, X. Fang, S. Misra, and J. Zhang, "A Game-Theoretic Approach to Stable Routing in Max-Min Fair Networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1947–1959, 2013.

[15] I. Castineyra, N. Chiappa, and M. Steenstrup, "The Nimrod Routing Architecture," *IETF RFC 1992*, Aug. 1996.

[16] X. Yang, D. Clark, and A. W. Berger, "NIRA: A New Inter-Domain Routing Architecture," *IEEE/ACM Trans. Netw.*, vol. 15, no. 4, pp. 775–788, Aug. 2007.

[17] A. Singla, P. Godfrey, K. Fall, G. Iannaccone, and S. Ratnasamy, "Scalable Routing on Flat Names," in *Proc. ACM CoNEXT*, 2010.

[18] K. Argyraki and D. Cheriton, "Loose Source Routing as a Mechanism for Traffic Policies," in *Proc. ACM SIGCOMM FDNA Workshop*, 2004, pp. 57–64.

[19] O. Ascigil, K. L. Calvert, and J. N. Griffioen, "On the Scalability of Interdomain Path Computations," in *Proc. IFIP Networking Conference*, 2014, pp. 1–9.

[20] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," *IETF RFC 2992*, Nov. 2000.

[21] W. Xu and J. Rexford, "MIRO: Multi-path Interdomain ROuting," in *Proc. ACM SIGCOMM*, 2006, pp. 171–182.

[22] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica, "Pathlet Routing," in *Proc. ACM SIGCOMM*, 2009, pp. 111–122.

[23] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen, "SCION: Scalability, Control, and Isolation on Next-generation Networks," in *IEEE Symposium on Security and Privacy*, 2011, pp. 212–227.

[24] H. Haddadi, M. Rio, G. Iannaccone, A. Moore, and R. Mortier, "Network Topologies: Inference, Modeling, and Generation," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 2, pp. 48–69, Apr. 2008.

[25] J. Li and Y. Pan, "Dynamic Database Management for PCS Networks," in *Proc. IEEE ICDCS 2001*, 2001, pp. 683–686.

[26] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," *IETF RFC 6275*, Jul. 2011.

[27] "LISP Network Monitoring Platform," http://lispmon.net, Accessed August 2015.

[28] V. Ramasubramanian and E. G. Sirer, "The Design and Implementation of a Next Generation Name Service for the Internet," in *Proc. ACM SIGCOMM*, 2004, pp. 331–342.

[29] L. Jakab, A. Cabellos-Aparicio, F. Coras, D. Saucez, and O. Bonaventure, "LISP-TREE: A DNS Hierarchy to Support the LISP Mapping System," *IEEE JSAC*, vol. 28, no. 8, pp. 1332–1343, 2010.

[30] "The CAIDA AS Relationships Dataset, 1998-2014," http://www.caida.org/data/as-relationships, Accessed August 2015.

[31] "RIPE Routing Information Service," http://www.ripe.net/data-tools/stats/ris, Accessed August 2015.

[32] "University of Oregon Route Views Project," http://www.routeviews.org, Accessed August 2015.

[33] Y. He, G. Siganos, M. Faloutsos, and S. Krishnamurthy, "A Systematic Framework for Unearthing the Missing Links: Measurements and Impact," in *Proc. USENIX NSDI*, 2007.

[34] "The DIMES Project," http://www.netdimes.org, Accessed August 2015.

[35] B. Augustin, B. Krishnamurthy, and W. Willinger, "IXPs: Mapped?" in *Proc. ACM IMC*, 2009, pp. 336–349.

[36] "Packet Clearing House," http://www.pch.net, Accessed August 2015.

[37] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger, "Anatomy of a Large European IXP," in *Proc. ACM SIGCOMM*, 2012, pp. 163–174.

[38] "Renesas Network Search Engine," http://www.renesas.com/products/memory/nse/index.jsp, Accessed August 2015.

[39] A. Dhamdhere and C. Dovrolis, "The Internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh," in *Proc. ACM CoNEXT*, 2010, pp. 21:1–21:12.

[40] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet Inter-Domain Traffic," in *Proc. ACM SIGCOMM*, 2010, pp. 75–86.

[41] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas *et al.*, "AS Relationships, Customer Cones, and Validation," in *Proc. ACM IMC*. ACM, 2013, pp. 243–256.

[42] A. Elmokashfi, A. Kvalbein, and C. Dovrolis, "On the Scalability of BGP: The Roles of Topology Growth and Update Rate-limiting," in *Proc. ACM CoNEXT*, 2008, pp. 8:1–8:12.

[43] S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt, "Constrained Mirror Placement on the Internet," in *Proc. IEEE INFOCOM*, 2001, pp. 31–40.

[44] A. Fabrikant, U. Syed, and J. Rexford, "There's something about MRAI: Timing diversity can exponentially worsen BGP convergence," in *Proc. IEEE INFOCOM*, 2011, pp. 2975–2983.

[45] O. Younis and S. Fahmy, "Constraint-Based Routing in the Internet: Basic Principles and Recent Research," *IEEE Commun. Surveys Tuts.*, vol. 5, no. 1, pp. 2–13, Third Quarter 2003.