Inter-Domain Rendezvous Service Architecture
PSIRP Technical Report #TR09-003

Jarno Rajahalme
Nokia Siemens Networks

Mikko Särelä
Ericsson Research Nomadiclab

Kari Visala
Helsinki institute for Information Technology HIIT

Janne Riihijärvi
RWTH Aachen

30 December 2009

Abstract

Locating objects with topology-independent identifiers has emerged as a key functionality in data and information centric approaches to networking. Numerous designs have been proposed to address the obvious scalability and efficiency challenges such systems face in Internet-scale deployments. However, dealing with evolvability and deployability concerns in environments encompassing multiple administrative domains has remained a relatively untouched subject.

Taking on this challenge, we base our inter-domain rendezvous design on the incentives and policies of networking stakeholders. The design uses a BGP-like routing structure between the enterprise customers and service providers, and an interconnection overlay between the service providers. We use domain-level simulations to verify that the scalability and efficiency objectives are not compromised.

Keywords Deployment incentives, Internet architecture, naming, network architecture, rendezvous, routing.

Chapter 1

Introduction

The problem of locating objects with identifiers independent of network topology in Internet-scale networks is a common thread in many areas of recent networking research [2, 14, 19, 27]. Identifiers employed in these designs can be utilized to name objects of different types (e.g. hosts, services, information elements, to name a few), but they cannot be easily aggregated based on the location of the objects in the network topology.

This nature of topology-independent identifiers gives rise to a seemingly fundamental tradeoff between scalability and efficiency in inter-domain deployments. Wide scale scalability is enabled by distributing state maintenance burden among the network nodes, so that per-node load grows sub-linearly to the number of objects in the system. However, whenever the destination object related state is not present in the nodes along the shortest policy-compliant path, efficiency suffers. However, even while scalability and efficiency are required properties of Internet-scale solutions, these alone are not sufficient system properties for practical deployability.

As deployment takes place gradually, roughly one administrative domain at a time, an overlay connecting the deployed domains over the existing architectures is needed as part of the architectural solution [25]. This aspect is missing, by definition, from many of the clean-slate approaches [9] cited above.

In this paper, we propose a rendezvous architecture that enables registering and locating arbitrary network objects using a flat identifier space. It is based on a BGP-like routing structure between enterprise domains and service providers (SPs), and an overlay between participating service providers. BGP-like state distribution allows enterprise multihoming, mobility, and direct peering between the enterprises without affecting the overlay(s) operated by the SPs.

Domain-level incentives (and the resulting traffic policies) are the deciding factors determining stakeholder's willingness to deploy new technology. Ignoring incentive considerations as part of the technology design effort can place far too much market power in the hands of a single organization, or lead to severe inter-domain deployment challenges. With the benefit of hindsight the latter can be clearly seen from the fate of numerous past standardization efforts (e.g.

IP multicast [7], IPv6 [31], and inter-domain QoS [16]).

The incentives of service providers and enterprises can be vastly different. This is significant, as the vast majority, more than 90%, of the ASes represent enterprises [6]. The other ASes can be categorized as transit, content hosting and access service providers. The share of these is getting smaller, as the share of enterprise ASes exhibits somewhat super-linear growth, while the other types are leveling off with sub-linear growth.

The service providers are cooperating, albeit sometimes reluctantly, to provide global connectivity services, while the enterprises might be competitors not wanting to deal with each other, route each other's messages, or have a competitor route their messages. For these reasons and the desire for universal reachability, although cooperating enterprises could form peer-based overlays at the edge, we propose a rendezvous architecture based on service providers.

The roadmap for the rest of this paper is as follows: Section 2 covers our major design considerations, while Section 3 describes the proposed inter-domain rendezvous service architecture in detail. Next, Section 4 describes the modeling used for our rendezvous overlay evaluation, while Section 5 gives the evaluation results. Finally, Section 6 discusses the work of other in this area, and Section 7 concludes this work.

Chapter 2

Design Considerations

The aim of the rendezvous architecture is to provide a shared global infrastructure for finding objects of different types, such as network nodes, services, or information in general. Optimizing for incentive compatibility, route stretch and state requirements, routing on flat topology independent identifiers could be done with BGP like peering hierarchy [14, 19] or overlay routing methods such as hierarchical DHTs [11]. However, both of these approaches have their share of problems.

In a likely partial deployment scenario there may be hundreds of "pockets of deployment" in the global network, which makes global use of BGP-like peering hierarchy problematic, especially if many of the tier-1 providers refuse to carry the burden of maintaining the reachability state for the whole global object population. Such arrangement makes it necessary for many small ISPs to invest in large data centers to hold pointers to all existing identifiers and to maintain rendezvous peering relationships with a large number of other ASes. The second approach based on hierarchical DHTs suffers from incentive incompatibilities as the initiation packets may travel through networks of competing enterprises.

We resolve these difficulties in our design by the separation of the "edge-based" rendezvous networks, running BGP-like routing protocol(s) between neighboring domains, from "core-based" rendezvous overlay(s). This separation enables the reach of the aggressive state distribution of the edge networks to remain bounded, while the rendezvous overlays provide wide area object reachability in a highly scalable and efficient fashion.

It is important to note that not all objects are expected to be equally popular. Especially studies on content delivery platforms, social networks and the world-wide web have shown that popularity of objects tends to have a long tail, but with only a small proportion being subject to most of the requests [4, 13, 18]. Power laws or Zipfian distributions have been found to be a good match for object popularity in many of these contexts, and we expect similar behavior to occur for most rendezvous systems as well. As we shall see later this has significant impact on use of caching in the architecture.

Chapter 3

Rendezvous Architecture

The rendezvous service model is simple: Objects are registered in any rendezvous node the object owner has a relationship with. Correspondingly, object users ask their local rendezvous nodes to locate that object for them.

For organizing the rendezvous service, we propose an approach where enterprise ASes within an edge network rely on service provider ASes (e.g. ISPs, or content or access hosting providers) serving the edge network to interconnect using an overlay. In such a *rendezvous network* enterprise ASes only touch messages either sourced from or destined to themselves, avoiding incentive incompatibility problems identified earlier. The edge-based rendezvous networks may span from a fraction of a large AS to a set of multiple ASes.

Both the explicit BGP-like relationship and the interconnection overlay structure allow bypassing non-deployed or non-participating domains, and thus enable gradual deployment of the rendezvous service.

We follow the design of past systems and give objects statistically unique cryptographically generated identifiers [1]. Rendezvous takes place with these identifiers, and hence we call them *rendezvous identifiers* (*RIds*). The cryptographical nature of the identifiers is instrumental for securing the protocol operations. Due to the similarity in the identifier structures, we refer to AIP [1] for identifier related security properties.

Figure 3.1 shows a simple topology example of our rendezvous design. Stub ASes are organized as *rendezvous networks* with their providers and the providers organize as an *interconnection overlay* to provide global reachability. These concepts are further explained in the following subsections.

3.1 Rendezvous Networks

The stub ASes shown in Figure 3.1 run their own internal rendezvous systems and get inter-domain rendezvous connectivity from their rendezvous providers (which again may be served by their providers). BGP-like routing protocols enable policy-compliant [12] paths for the signaling messages. Object *registration*

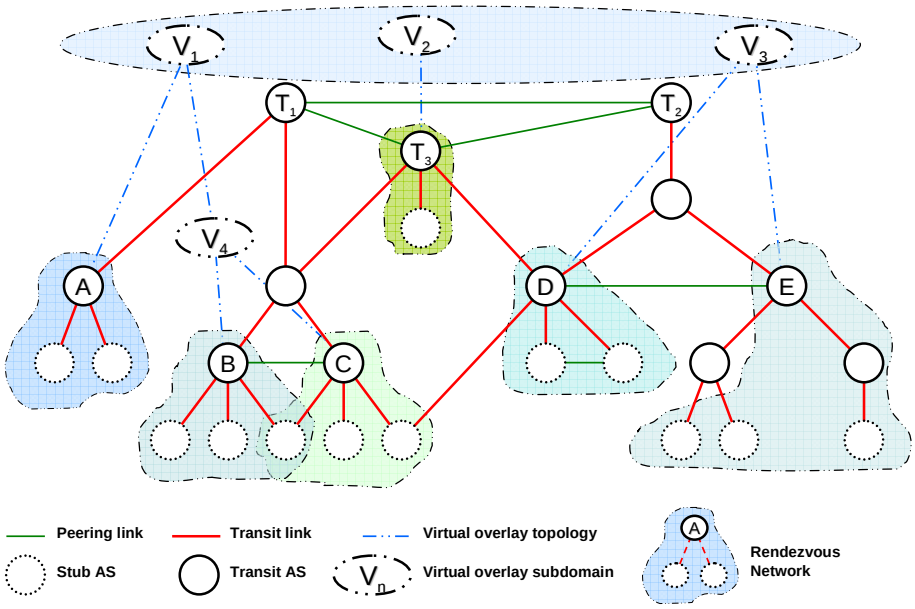

Figure 3.1: Rendezvous network interconnection.

messages containing the RIds of the objects are propagated in the domain hierarchy upstream to providers and peers until a rendezvous overlay participant is reached. Each AS along the way stores the RId and a pointer towards the object. The object location is then registered to the overlay, so that queries from any part of the Internet will be eventually routed to the rendezvous node the object has been originally registered in.

Similarly, when a host queries for an object, the request is forwarded upstream, unless a pointer to the object is found locally. Finally, if the object reaches the topmost node in the edge network, and the object pointer is still not found, the request is forwarded on the interconnection overlay, as described below.

3.2 Rendezvous Overlay Structure

The edge-based rendezvous networks are interconnected using a hierarchical overlay structure composed of two parts: At the lower level, domains organize into a local hierarchical overlay structure based on Canon [11] and at the global level these hierarchies are connected together separately to minimize stretch and latency.

At each level of the overlay hierarchy, and in the global overlay, each object has a designated overlay node that is responsible for maintaining a pointer to

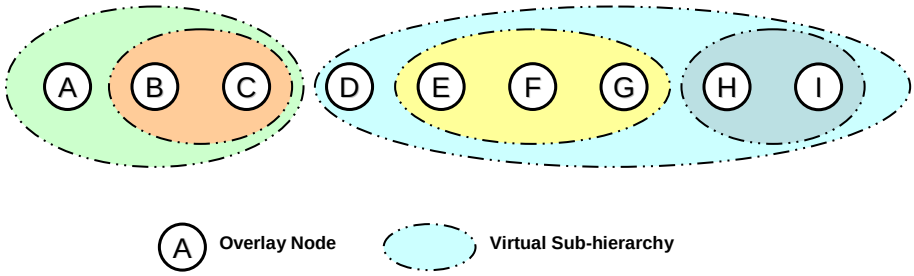

Figure 3.2: Example of a hierarchical structure of overlay nodes.

the rendezvous node through which the object was originally registered. The overlay node is selected by comparing the overlay node identifiers to the object identifier: It is the overlay node with the highest identifier value not overshooting the object identifier, wrapping the identifier space around at zero.

We anticipate that domains physically close to each other, e.g. within the same country, may wish to contain their mutual rendezvous traffic to their own networks. This is enabled by hierarchical interconnection of their rendezvous overlay nodes (Figure 3.2). The hierarchy is formed by joining the overlay structures one by one, starting from the bottom of the hierarchy, and proceeding to the higher levels. Routing locality is then preserved by observing the resulting structure while forwarding rendezvous signaling messages. As a bonus, the hierarchical structure exhibits good *convergence of inter-domain paths*, enabling efficient caching of object pointers.

When the desire for locality is exhausted, routing shifts to a global mode. This structure is based on identifier prefix groups [11], but organized in compact routing fashion [20]: Each node belongs to a prefix group formed by all the overlay nodes sharing the same identifier prefix (Figure 3.3). Nodes maintain overlay links to all other nodes within their own prefix group. Each node also maintains at least one overlay link to *some node* in each other prefix group, *starting from the prefix following its own, up to the prefix of the node's successor in the top-most overlay hierarchy.* This latter emphasized part is a new optimization that has not been reported earlier. As the prefix length determines the total number of prefix groups, by properly selecting the prefix length the nodes can minimize the needed number of overlay links. The overlay links to other groups are selected so that the group-to-group latency is minimized by choosing as short links as possible.

Finally, replicating object pointers to a few other nodes in the prefix group makes it possible to find the pointer sooner. Thus routing all the way to the algorithmically designated node is not always needed. In this setting, it is possible to maximize service availability and minimize the average overlay routing latency at the same time. However, for scalability reasons wider replication should be limited to the most popular objects [24].

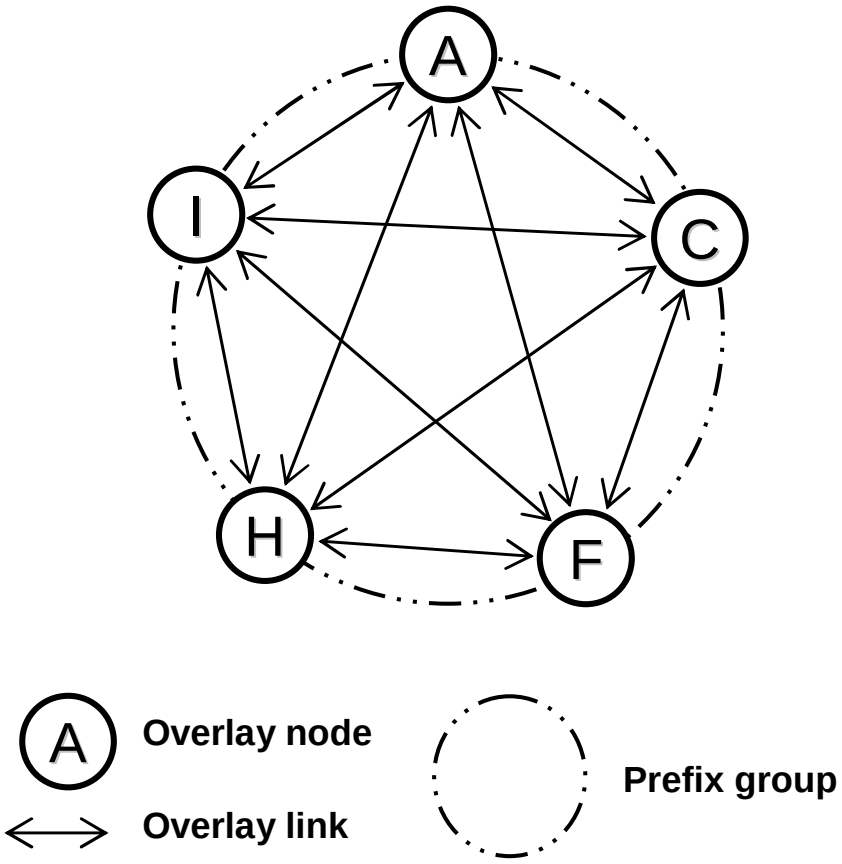

Figure 3.3: Group of nodes all sharing the same identifier prefix linking to each other.

3.3 Message Routing and Caching

Rendezvous requests are routed in three phases:

1. Within the edge-based rendezvous network, following inter-domain adjacencies.
2. Within the locality-preserving hierarchical overlay, bottom-up.
3. Within the global latency-optimized overlay, first towards the object's prefix group, and then towards the closest node replicating the object pointer.

During each phase, the routing node's local cache is examined to short-circuit the rest of the routing process whenever possible.

Whenever an object pointer is found, the rest of the routing process is dismissed and the rendezvous request is forwarded to the object's home rendezvous node. To avoid issues with non-transitive connectivity [10], the home rendezvous node should send the response utilizing the established overlay links.

Routing response through the established overlay links also enables intermediate nodes to cache the requested object pointer. This takes advantage of the convergence of the paths for the same object: If the object pointer is cached along the path of the request message, then the message is immediately redirected towards the home location of the object from there.

The degree of caching, or the storage and processing capacity devoted to caching can be determined by each domain locally. The benefit of caching to the requesting node is obvious, but the caching domain also benefits as it does not need to use its connectivity resources to forward the request, and handle the response.[1]

3.4 Overlay Reliability and Trust

The utilized overlay structure is virtual and thus does not require specific support from non-deployed networks. However, the overlay depends on the participants having a common agreement on their relative position in the virtual structure. This requires a degree of mutual trust, and therefore entails obligations between the participants.

On the hierarchical level, we expect the participating service providers to cooperate only with parties that they can enforce contracts with. On the global level, trust must be handled differently. When a next hop for a rendezvous message is determined, the aim is to choose a node that is 1. trusted, 2. topologically close, and 3. hosts the destination object pointer with high likelihood.

Note that within edge-based rendezvous networks the above properties are assumed to hold by the virtue of chosen rendezvous providers routing the messages and caching the returned object pointers.

[1] Unless the cached entry was stale, in which case the caching node will receive an error response and will need to forward the request normally.

3.5 Object Collections

In addition to bare "objects" we define *object collections* to capture the natural containment structure within many types of objects. For example, individual photographs in an album may be considered an object collection.

We expect most objects within the inter-domain rendezvous overlay to be object collections and most other objects be stored in one or more object collections. The collection itself is registered in the inter-domain overlay and is, thus, globally reachable. Individual objects within collections that are not separately registered in the rendezvous system are reachable only through the containing object collection. This arrangement creates a layer of indirection between the user and object of interest.

We expect the rendezvous service to offer natural incentives, favoring the use of collections vs. individual object registrations. Firstly, globally reachable registrations will likely cost something. Secondly, using a collection may reduce the user-visible latency, as there is no additional latency from individual object resolutions, provided that the whole collection is serviced through the same rendezvous node. Thirdly, using collections enables custom, collection-level access controls. These can be very useful for various users ranging from corporations to individuals.

While it is possible to utilize more than one level of indirection of identifiers, we refrain from this generalization here for the simplicity of exposition.

3.6 Object Multihoming and Mobility

The proposed architecture supports object mobility and multihoming in a natural way. Object mobility requires the owner to register the new location and unregister the old location. The unregister message is passed through the same rendezvous node from which the registration was originated.

For a multihomed object, the rendezvous system will route the rendezvous request to the closest rendezvous node serving the object, or alternatively to multiple rendezvous nodes in parallel. To enable this, the rendezvous nodes receiving the multiple registrations store all the parallel pointers.

In many cases, however, a cached object pointer may be found already within one of the local hierarchies, in which case the query will automatically be sent following the pointer.

3.7 Alternative Rendezvous Structures

The naming architecture described in this work is not limited to the system architecture described in this Section. Such naming architecture could be deployed in many ways ranging from cooperating enterprises building peer-to-peer based rendezvous systems to a single "oracle" entity which takes the responsibility for routing rendezvous messages between all the edge networks. However, this approach is fraught with challenges related to e.g. trust, market structure (e.g.

likely monopoly), faith-sharing, and incentives. Some of these challenges could be solved by free competition between multiple such *rendezvous providers*. The providers would compete on global identifier-level routing coverage and thus they would want all the rendezvous networks to register their state (changes) with them.

However, it is simpler for the edge networks, if their rendezvous providers hide the complexities of global interconnection from them. Using simple state flooding between the providers [14, 19] would lead to an organization somewhat resembling the peering Tier-1 structure at the top of the Internet transit structure today [15]. However, there are obvious problems with scalability, or the cost of providing such a service without matching incentives to invest in new data centers when the supported identifier name space is sufficiently large.

Chapter 4

Evaluation Model

Given the incentive- and trust-motivated architecture described above, the next step is to assess its practicality in a network ecosystem resembling today's Internet. To achieve that we construct a domain-level rendezvous simulation model utilizing the present-day autonomous system structure and traffic patterns. Packet-level simulation on the host level was not feasible on this scale, and it is also not necessary for the level of detail and realism we seek.

The targeted evaluation metrics are:

1. Policy-compliant inter-domain path *stretch*
2. Additional overlay routing *latency*
3. Caching efficacy
4. Overlay node *routing load distribution*

From the end user and content service provider points of view the most important of the above metrics is the additional latency caused by the rendezvous operation, because in typical usage rendezvous signaling precedes the actual payload communication phase. As the rendezvous system is part of the *control plane* of the network, the rendezvous messaging bandwidth is not as important, as it is assumed that the actual data traffic forms the bulk of the total network load. Additionally, the effect of caching to the rendezvous overlay performance is of interest.

The simulation model captures the essential structures of the design presented in the preceding section: The edge-based rendezvous networks, locality-preserving hierarchical overlays, and the global level overlay structure.

The simulation proceeds as follows: Having constructed the network (Section 4.1), we generate rendezvous requests based on the employed traffic model (Section 4.4), measure the path length and latencies while routing the request through the model as described in Section 3.3, and accounting for the overlay nodes acting on the message. Finally we compare the accumulated figures to

the alternative of being able to route the message from the source AS to the destination AS directly using the shortest policy-compliant path.

The following subsections describe the components of the simulation model in more detail.

4.1 Network Formation

We assume that within the edge-based rendezvous networks there is enough state around to enable the rendezvous messages to follow policy-compliant paths [19]. Thus we do not simulate the actual edge network routing state, instead we deduce the policy-compliant paths with sufficient accuracy based on the underlying network topology model (Section 4.3).

Formation of the simulated rendezvous networks is based on a simple model, where the transit service providers offer rendezvous as a service for their stub customers as well as small transit customer networks, when the total number of customers is not too high. This provides a sufficient approximation of the division between enterprise and non-enterprise domains.

The rendezvous networks are then combined together in hierarchical overlays following the Canon design [11]. The sub-hierarchies are built based on locality of the rendezvous providers, given our assumption that topologically close networks are usually capable of forming coalitions and can benefit from the ability to contain their mutual rendezvous messaging within their networks.

Lack of multihoming in the overlay hierarchy is a limitation in the current simulation model.

When the possibilities for proximity-based clustering are exhausted, the global layer of interconnection is established as described in Section 3.2 above.

4.2 Network Dimensioning

To build the simulation network, we need estimate the number of overlay nodes required by each rendezvous service provider. The required number of nodes depends on primarily on two factors: 1. the expected number of registrations created by the customers of the rendezvous service provider, and 2. the amount of memory needed to store each object registration in the rendezvous overlay.

We assume that the total number of globally reachable objects in the system is at least an order of magnitude higher than the number of registered domain names in the Internet, i.e. around 10^{10} objects or object collections in total. The distribution of these on the individual ASes hosting these objects is based on the traffic model described in Section 4.4 below.

Each rendezvous registration takes about 56 bytes: 32 bytes for the object identifier, up to 16 bytes for the next hop IP address, and some reserve for the overhead of the data structures. Given the capabilities of a typical server nowadays, we can estimate that a typical rendezvous overlay node could thus store around 10^8 such records in memory.

In our overlay design there often is a copy of each object pointer at each level of the overlay hierarchy. Additionally, on the global routing level replication is used both for fault tolerance and better latency performance. Therefore the total storage space requirement of each registration is multiplied by the factor of $(d+r)$, where d is the average depth of the Canon hierarchy, and r the average replication count on the global level. To be on the safe side, we assume that this factor reduces the average hosting capacity of an overlay node to around 10^7 object registrations.

The above implies that even with such a high total object count, around 10^3 servers would suffice from the memory point of view to support the collection of object registrations of *all* rendezvous networks. As most of the overlay nodes are underutilized compared to their full capacity, in practice the number of nodes in the system is somewhat higher.

4.3 Network Topology

We use CAIDA's AS relationships dataset [3] as our Internet inter-domain topology model. One of our primary uses of the dataset is to compute path lengths for valley-free paths between ASes, observing the typical routing policies resulting from the commercial interests of the autonomous systems [12].

The CAIDA dataset consists of a full AS graph derived from a set of RouteViews BGP table snapshots. While it gives a good picture of the Tier-1 connectivity and provider–customer links between ASes, it is known to lack many of the peering links [22]. In the case of large content provider networks, it is reported that as much as 90% of their peering links can be missing, because these links are invisible to the set of available route monitors. Naturally, this inaccuracy of the model affects our results, but nevertheless gives us a conservative estimate of the scalability of the system as some existing links have not been utilized.

The total number of ASes in the dataset used is 25881 and number of (bidirectional) links is 52407. The links are annotated as being either peer–peer, provider–customer, or sibling–sibling ones.

4.4 Traffic Model

We form the traffic model for the system by categorizing ASes into different types, each playing a different role in the system both in terms of participation in rendezvous network operation and in generating traffic. The used categorization is given in [5], characterizing each AS in terms of the traffic volumes of three types of network usage *utilities*.

The three utility types are: *web hosting* ($U_{\texttt{web}}$), *residential access* ($U_{\texttt{ra}}$) and *business access* ($U_{\texttt{ba}}$). Business access models the cumulative transit provided through the AS. Each of these utilities follows a power-law (i.e. Zipfian) distribution, parameters of which can be found from [5]. Based on these results, we

annotate the AS graph by generating random variates from the Zipf distributions for different utilities, and assigning these to the ASes so that the observed rank correlations are obtained.

We assume that target scopes of queries are distributed to ASes proportional to $U_{\texttt{web}} + \alpha U_{\texttt{ra}}$.[1] The queries themselves are originated using the residential access ($U_{\texttt{ra}}$) distribution only.

The popularity of objects is also assumed to follow a Zipfian distribution in line with several studies in content delivery networks and other such services [4, 13, 18]. This property should enable highly effective caching as part of the overlay solution (see Section 3.3).

4.5 Network Latency Model

To obtain estimates for the link latencies, we use the following delay values: 34ms for inter-AS node-node hops and 2ms for intra-domain router hops [33]. The number of intra-domain router hops used between overlay nodes residing in the same AS is $1 + \lfloor \log D \rfloor$ where D is the degree of the AS. This is based on findings in [29] where a strong correlation[2] between the number of routers in an AS and the degree of the AS is found, and on the assumption that the intra-domain routing topology is efficiently designed.

4.6 Caching Model

Our caching model is rather simple, but the results (Section 5) are indicative nonetheless. We found that due to the Zipfian popularity distribution, rather modest cache will be enough to cache the most popular scopes. However, the required cache size for good cache hit rate depends heavily on the power-law exponent realized in the object population.

Apart from the cache size, cache freshness is an important issue. As reported in [18] for DNS caching, the Time-To-Live (TTL) value of 15 minutes will yield higher than 80% cache hit ratio for the first level cache.

For individual requests the cache hit ratio is determined by the number of the queries received in the TTL period for the given object identifier. Naturally the hit rate is higher for the most popular destinations, and for rarely requested destinations the hit rate goes quickly to zero, as the query interval exceeds the TTL period. Based on these findings we stochastically determine the cache availability for each request so that the higher the popularity (query frequency) for the object, proportionally higher the likelihood is that the entry is still valid in caches.

[1] In our simulations we used the value 0.5 for parameter α, that is, web hosting generates more scopes than home users do. However, a later sensitivity analysis showed that our metrics are not particularly sensitive to the choice of α (in the vicinity of this value, that is).

[2] In May 2001 the coefficient of correlation was 0.959.

Chapter 5

Simulation Results

We ran the simulations described above and found it generating, in a typical case, ∼1220 rendezvous networks with the total of ∼2700 overlay nodes. The average locality-preserving hierarchy depth is ∼3.3.

The figures reported below are generated using multiple simulation rounds with different random seeds, including topology generation.

In Figure 5.1 we report the cumulative distribution function (CDF) of the measured stretch, the multiplier to AS-level hop count required to route a rendezvous message from a randomly selected source domain to the domain where the sought after object is located, via the overlay structure vs. routing directly using a routing policy compliant valley-free path (bypassing the overlay). The small number of requests with stretch below one are due to the overlay node functioning as a detour [26], offering a shortcut, typically utilizing peering links otherwise not usable for policy-compliant end-to-end paths.

In Figure 5.2 we show the histogram for Chord hops needed to reach the scope pointer.

Figure 5.3 shows the cumulative distribution of overlay routing latency in milliseconds, depicting the time to reach an overlay node holding the scope pointer, starting from the user's home AS. The latency model used is described above (Section 4.5).

Figure 5.4 shows the distribution of node load, measured by the number of times each node forwards or handles a rendezvous request message during a simulation run of 100000 rendezvous requests. It can be seen that most of the nodes are lightly utilized, and the heaviest load concentrates on a rather small set of nodes. However, the ratio between the most heavily loaded nodes and the average is not too big, and caching reduces that ratio considerably.

5.1 Sensitivity Analysis

After conducting the simulations reported above, a factorial analysis was performed to get an understanding of the sensitivity of the model to some of the

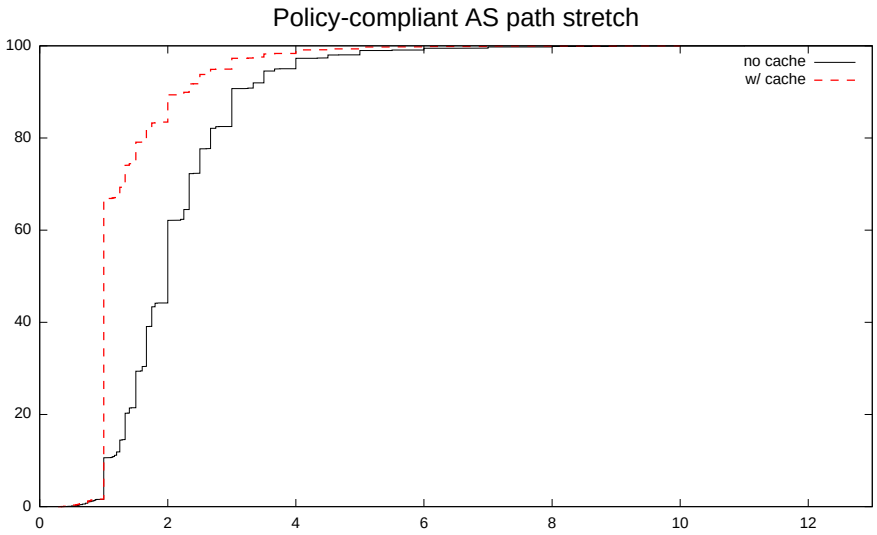

Figure 5.1: AS-level stretch CDF: 2.11 (mean), 3.75 (95%) (without caching); 1.35 (mean), 3.00 (95%) (with caching).

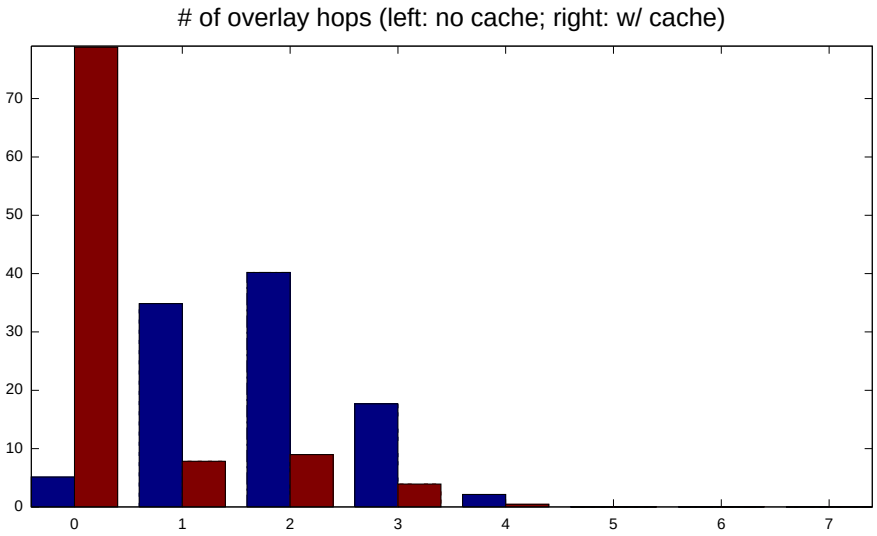

Figure 5.2: Rendezvous overlay hops needed to reach the scope pointer (% of requests): 1.77 (mean), 3.00 (95%) (without caching); 0.39 (mean), 2.00 (95%) (with caching).

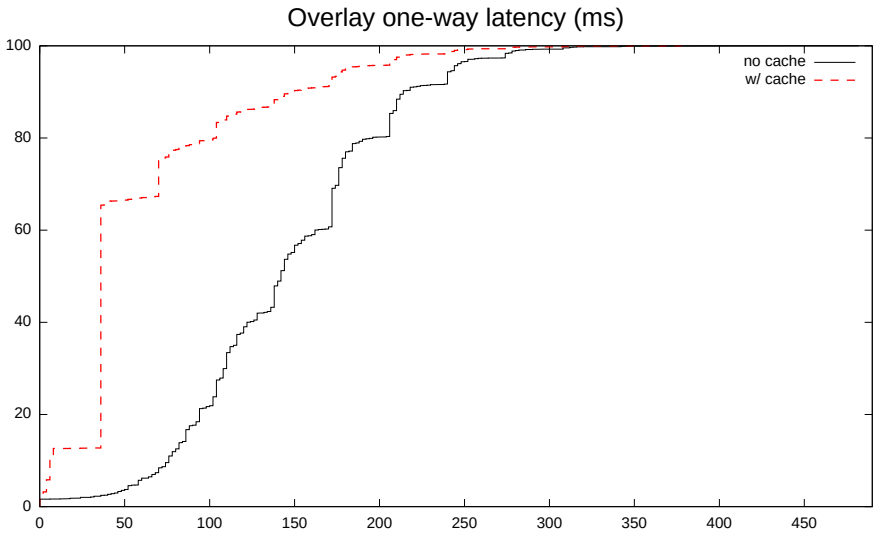

Figure 5.3: Query latency CDF (ms): 144 (mean), 244 (95%) (without caching); 62 (mean), 180 (95%) (with caching).

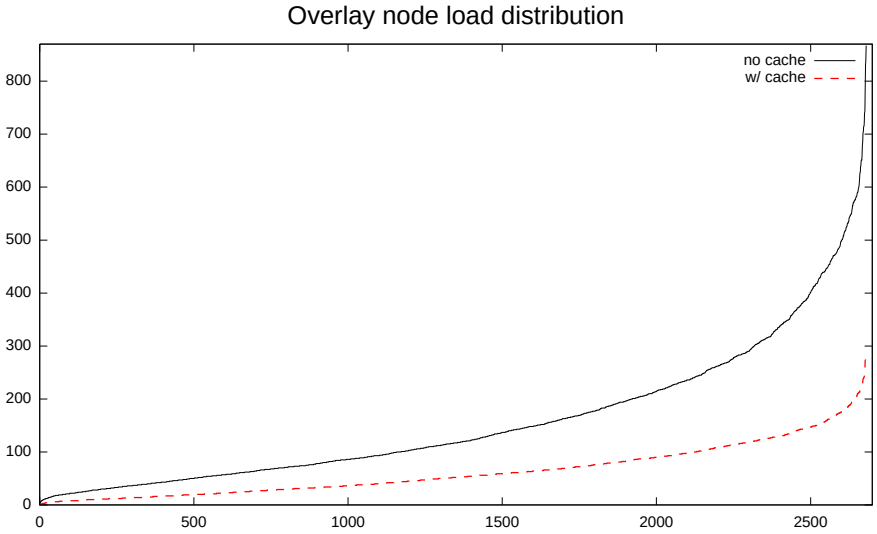

Figure 5.4: Node load distribution of 100000 rendezvous requests on the overlay nodes: 159 (mean), 441 (95%) (without caching); 63 (mean), 157 (95%) (with caching).

Parameters:	Node Mem.	SmallNetLim.	Util.Frac.	Min.Nodes	Repl.Count	Links
Base level:	4GB	2	0.1	1	1	70
Alt. level:	2GB	8	0.3	2	4	140
Effect:						
Stretch (mean):	-0.1691	-0.1142	-0.0781	0.1775	-0.5747	-0.1090
CI (95%):	0.0001	0.0019	0.0002	0.0002	0.0009	0.00004
Stretch (95%):	-0.2991	-0.2373	-0.1149	0.2978	-1.0550	-0.2002
CI (95%):	0.0030	0.0202	0.0053	0.0077	0.0122	0.0024
Latency (mean):	-10.4120	-8.4217	-6.8646	13.1244	-37.3722	-9.7002
CI (95%):	0.1047	5.8838	1.4313	0.3427	0.5786	0.3221
Latency (95%):	-13.9063	-18.5000	-9.9375	12.3594	-51.7656	-11.0469
CI (95%):	1.6636	18.5044	7.7257	3.5080	6.8988	0.7126

Table 5.1: Sensitivity analysis results (8x64 replications, no caching). Numbers indicate the average effect of a given variable change from base to alternative level. CI lines report the half length of the 95% confidence interval for the parameter effect above.

simulation parameters. The analysis was conducted without caching to better show the effect of individual parameters. The non-caching case can be considered the "worst case scenario", as even modest caching dramatically improves the average performance figures. Analysis results based on 8×64 replications are summarized in Table 5.1.

As can be seen from the confidence interval figures in the Table 5.1, the average stretch is highly stable with modest number of replications (8), while analyzing the tail (95 percentile) of the latency would in the case of some parameters need more replications in order to yield statistically significant results.

The difference in the parameter values for the base level and the alternate level were set on the intuition that alternate values should produce better performance. This was indeed the case for most of the variables:

Node Memory Amount of memory nodes dedicate to store object pointers: Less memory, more nodes needed to serve the target 10^{10} scopes.

SmallNetLimit Limits the size of networks accepted as Rendezvous network customers. Bigger rendezvous networks lead to lower average latency, as their internal routing is policy-compliant, but amount of state that needs to be managed by a single rendezvous service provider grows.

Util.Fraction Allows networks with small impact on load to become rendezvous customers, higher value leads to lower total number of rendezvous networks created, but more load on individual rendezvous service providers.

Min Nodes Minimum number of overlay nodes for each rendezvous service provider. More than one may be necessary for fault tolerance (compare to DNS operative guidelines).

Repl.Count Number of replicas of object pointers to keep on the global level. One is the functional minimum, more than one may be needed for fault tolerance. Wider replication should help lowering the routing latency, but requires more storage space in the system.

Target Links 70 is close to functional minimum with the simulation parameters used. More overlay links implies denser connectivity; more opportunities to select shorter routes and higher the maintenance cost for managing the links.

The only (initially) counter-intuitive response was with the minimum number of nodes in a rendezvous service provider domain: While increasing overlay node count via reducing the per-node memory allocation resulted in consistently (somewhat) better performance, increasing the minimum number of overlay nodes in all rendezvous networks from 1 to 2 resulted in worse performance. This may be explained by the fact that the minimum applies also with rendezvous service providers with lightly loaded overlay nodes, hence the main effect is somewhat increased stretch in the system.

	Stretch (mean)	Stretch (95%)	Latency (ms,mean)	Latency (ms,95%)
Without caching:				
Best:	2.075	3.79	142	243
Worst:	3.217	5.94	222	347
With caching (∼75% hit ratio):				
Best:	1.3410	2.74	62	186
Worst:	1.5950	4.058	79	281

Table 5.2: Best vs. worst parameter combination performance (with and without caching).

To assess the effect of caching on the parameter sensitivity, we compared the performance of the worst performing combination and the best performing combination of the parameters in the Table 5.1, with and without caching. The performance figures are summarized in Table 5.2. Contrary to our anticipation, caching helps also the 95 percentile latency performance (243 ms vs. 186 ms for the best performing parameter set). Also contrary to our anticipation, caching *did not* mask the performance difference between the best and worst performing parameter sets: Both with and without caching the increase in the 95 percentile latency is ∼50% more with the worst performing parameter set, compared to the best performing parameter set.

Finally, we compared the performance of our provider based model against the variant where there are only singular rendezvous networks, i.e. a model where each AS operates as their own rendezvous service provider. Similar hierarchical clustering was conducted in each case. On the average over multiple simulation rounds the alternative resulted in about 25% increase in both stretch and latency. This shows that our incentive-based separation between the roles of the ASes may indeed perform better than the variant without such separation.

5.2 Discussion of the Results

Our sensitivity analysis shows that our model gives rather robust performance indicators for our chosen metrics, other than for the 95 percentile latency, which exhibits significant statistical fluctuation between simulation runs for some parameters. Even so, we conclude that the additional (95 percentile) latency contribution of our rendezvous design compares favorably to DNS, where the comparable worst case performance is measured in seconds [18].

The analysis above shows that our system performance is better than the variant without the incentive motivated separation of the edge-based rendezvous networks from the rendezvous service providers. Also, it can be argued that our stretch and latency figures show that the system performance is close enough to stretch-1 systems, when considering the huge decrease in the required number of servers and the related incentive challenges.

The presence of transit loops and the lack of policy-compliant connectivity between some domains in the CAIDA dataset forced us to add a degree of realism we did not originally plan for in the simulation. To manage the observed non-connectivity, we probed the connectivity in the dataset for each candidate overlay link and only used links that had policy-compliant connectivity. This also enabled us to mimic the real-world latency measurement capabilities used in overlay structures, resulting in realistic analysis of the effect of choosing topologically short links.

The major deficiency in the presented analysis is the lack of modeling of deflection, or untrustworthy operation by some of the overlay participants. However, there are couple of approaches that could be utilized in practice to detect and route around such behavior. Firstly, overlay nodes could ask a random set of other nodes to test the reachability for the object(s) registered by the first node. This would help against lying nodes that would respond favorably the the original registrar, but still claim non-availability to the other requesting nodes. Secondly, having detected untrustworthy behavior, the first node could initiate wider replication of its object registrations, thus quickly minimizing the effect of the misbehaving node. Furthermore the first node could initiate measures for excluding the lying node from the overlay.

Chapter 6

Related Work

Many of the basic insights to network architecture deployability are stated in [25]: New architecture evolution starts with partial deployment, which is made possible by anycast service provided by the underlying, old architecture. However, [25] simply assumes existing contractual agreements and market structure, which seems to be at odds with its revenue flow assumption: There are no ASes which would universally benefit from operating as new architecture detours.

Deployability concerns have also been studied in [7, 8, 16, 30, 31], but deployment incentives and their effects on technology design are rarely considered, [17] being a welcome exception.

The concept of *rendezvous-based communication abstraction* was introduced in Internet Indirection Infrastructure ($i3$) [27]. However, $i3$ operates directly on Chord [28], making all the data traffic pass through the DHT structure. Unfortunately, Chord has no regard for domain-specific routing policies, so the $i3$ nodes operate as arbitrary "detours" [26], while, as stated above, there are no ASes for which such operation would be universally beneficial without new, $i3$-specific revenue.

ROFL [2] proposes routing on flat labels in the Internet-scale without the underlying IP forwarding assumed. ROFL, while also DHT-based and providing a general packet routing service like $i3$, addresses some of the DHT-related incentive concerns by enabling policy-compliant routes to be taken *after* the initial packet has passed through the ROFL routing stage. ROFL borrows this from NIRA [32], which defines a specific link-state protocol for maintaining an up-to-date view of the *upgraph*, the available uphill and downhill [12] paths between any user of the network and the other domains either via the Tier-1 networks or some other peering links.

Nevertheless, ROFL suffers from a subtle kind of policy-compliancy issue: Since each AS is a participant in the ROFL inter-domain DHT, it is possible that traffic (at least the initial packets) between any two enterprise ASes is routed via a third enterprise AS, who may be a direct competitor of either of the two first ASes. This behavior is comparable to BGP prefix hijack [21] and other man-in-the-middle attacks. In our rendezvous architecture, we design

around this issue by separating the concerns of enterprise ASes from the Service Provider (SP) ASes.

TRIAD [14] and DONA [19] use a BGP-like inter-domain routing protocols, similar to the one used in our work to form the edge-based rendezvous networks, to distribute route information: TRIAD for servers identified with DNS names and DONA for data with self-certifying identifiers. They both assume that all ISPs are naturally willing to peer on the name level, if they are already peering on the IP level. Due to this unrealistic assumption these designs result in policy compliant paths, but also scalability challenges. TRIAD resolves the problem by restricting the managed namespace to service names while DONA argues that large data centers can handle the load. In DONA, the Tier-1 ASes are burdened with memory, processing and communication overheads scaling linearly with the number of registered publications globally. This might be acceptable for a few Tier-1 ASes, if they had the incentives for such investments. However, the rendezvous-based communication abstraction may enable better utilization of peering links and/or local storage, reducing the transit traffic [23]. If Tier-1 ASes do not deploy, the high costs will multiply to many smaller ASes.

Chapter 7

Conclusions

Deployment of new architectures takes place one step at a time, each step taken by individual stakeholders acting on their own incentives. It is possible that this process never achieves full deployment over the existing network. Accepting this, the question of how to interconnect the deployed parts of the new network becomes the key problem to address. In many cases, interconnection is required between parties with no existing contractual relationships. Thus, in addition to the architectural changes, deployment may imply the need for new business structures, resulting in an interplay between architecture and business structure development.[1]

In the case of locating objects in Internet-like networks, it seems that balancing between the extremes of BGP-like state flooding and universal overlay designs enables addressing deployability concerns and some of the incentive differences between the different stakeholders in the Internet. The presented design divides the network into edge-based rendezvous networks and structured overlays interconnecting such networks, and provides better performance than the universal overlay option and far better scalability than the global state flooding designs.

While the presented heterogeneous design is necessarily more complex than any of the homogeneous alternatives, the effort seems worthwhile for the unique combination of characteristics rendered. However, the work presented here should be considered as an initial step towards understanding the deployability challenges in the proposed rendezvous based communication abstraction and in general. The design presented here is still far away from a finished product and therefore by no means ready for practical deployment. While the performance figures for our design seem encouraging, the intricate issues of trust in shared inter-domain structures require more careful analysis. In the end, the fate of any networking technology will be determined by the cruel test of the market forces at play in the evolving network we call the Internet.

[1] Adopting the argument of Lawrence Lessig in *Code is Law*.

Bibliography

[1] D. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable Internet Protocol (AIP). In *ACM SIGCOMM'08. Proceedings*, pages 339–350, August 2008.

[2] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker. ROFL: Routing on Flat Labels. In *ACM SIGCOMM'06. Proceedings*, September 2006.

[3] The CAIDA AS Relationships Dataset, August 18th, 2008. http://www.caida.org/data/active/as-relationships/.

[4] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *ACM SIGCOMM IMC'07. Proceedings*, pages 1–14, 2007.

[5] H. Chang, S. Jamin, Z. Morley, and M. W. Willinger. An Empirical Approach to Modeling Inter-AS Traffic Matrices. In *ACM SIGCOMM IMC'05. Proceedings*, pages 139–152, 2005.

[6] A. Dhamdhere and C. Dovrolis. Ten years in the evolution of the Internet ecosystem. In *ACM SIGCOMM IMC'08. Proceedings*, pages 183–196, 2008.

[7] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the IP multicast service and architecture. *Network, IEEE*, 14(1):78–88, 2000.

[8] N. Feamster, H. Balakrishnan, and J. Rexford. Some Foundational Problems in Interdomain Routing. In *Proceedings of ACM HotNets-III*, 2004.

[9] A. Feldmann. Internet Clean-Slate Design: What and Why? *ACM SIGCOMM CCR*, 37(3):59–64, 2007.

[10] M. Freedman, K. Lakshminarayanan, S. Rhea, and I. Stoica. Non-transitive Connectivity and DHTs. In *Usenix WORLDS'05. Proceedings*, 2005.

[11] P. Ganesan, K. Gummadi, and H. Garcia-Molina. Canon in G major: designing DHTs with hierarchical structure. *Distributed Computing Systems. Proceedings*, pages 263–272, 2004.

[12] L. Gao. On inferring autonomous system relationships in the Internet. *Networking, IEEE/ACM Transactions on*, 9(6):733–745, Dec 2001.

[13] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube Traffic Characterization: A View From the Edge. In *ACM SIGCOMM IMC'07. Proceedings*, pages 15–28, 2007.

[14] M. Gritter and D. R. Cheriton. An Architecture for Content Routing Support in the Internet. In *USENIX USITS'01. Proceedings*, 2001.

[15] G. Huston. Interconnection, Peering, and Settlements. In *Proc. Internet Global Summit (INET)*, Jun. 1999.

[16] P. Jacob and B. Davie. Technical challenges in the delivery of interprovider QoS. *Communications Magazine, IEEE*, 43(6):112–118, June 2005.

[17] D. Joseph, N. Shetty, J. Chuang, and I. Stoica. Modeling the adoption of new network architectures. In *Proceedings of the 2007 ACM CoNEXT conference*. ACM New York, NY, USA, 2007.

[18] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. *IEEE/ACM Transactions on Networking (TON)*, 10(5):589–603, 2002.

[19] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A Data-Oriented (and Beyond) Network Architecture. In *ACM SIGCOMM'07. Proceedings*, pages 181–192, 2007.

[20] D. V. Krioukov, K. Claffy, K. Fall, and A. Brady. On Compact Routing for the Internet. *ACM SIGCOMM CCR*, 37(3), 2007.

[21] O. Nordström and C. Dovrolis. Beware of BGP attacks. *ACM SIGCOMM CCR*, 34(2):1–8, 2004.

[22] R. V. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. In Search of the Elusive Ground Truth: The Internet's AS-level Connectivity Structure. *SIGMETRICS Perf. Eval. Rev.*, 36(1):217–228, 2008.

[23] J. Rajahalme, M. Särelä, P. Nikander, and S. Tarkoma. Incentive-Compatible Caching and Peering in Data-Oriented Networks. In *ReArch'08. Proceedings*, 2008.

[24] V. S. Ramasubramanian. *Cost-Aware Resource Management for Decentralized Internet Services*. PhD thesis, Cornell University, January 2007.

[25] S. Ratnasamy, S. Shenker, and S. McCanne. Towards an Evolvable Internet Architecture. In *ACM SIGCOMM'05. Proceedings*, pages 313–324, 2005.

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

[27] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. *Networking, IEEE/ACM Transactions on*, 12(2):205–218, April 2004.

[28] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, 2003.

[29] H. Tangmunarunkit, J. Doyle, R. Govindan, W. Willinger, S. Jamin, and S. Shenker. Does AS size determine degree in as topology? *ACM SIGCOMM CCR*, 31(5):7–8, 2001.

[30] M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering what-if deployment and configuration questions with wise. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pages 99–110. ACM New York, NY, USA, 2008.

[31] D. Waddington and F. Chang. Realizing the transition to IPv6. *Communications Magazine, IEEE*, 40(6):138–147, Jun 2002.

[32] X. Yang, D. Clark, and A. Berger. NIRA: A New Inter-Domain Routing Architecture. *Networking, IEEE/ACM Transactions on*, 15(4):775–788, Aug. 2007.

[33] B. Zhang, T. Ng, A. Nandi, R. Riedi, P. Druschel, and G. Wang. Measurement based analysis, modeling, and synthesis of the internet delay space. In *ACM SIGCOMM IMC'06. Proceedings*, pages 85–98, 2006.