On Name-Based Inter-Domain Routing[☆,☆☆]

Jarno Rajahalme[a,*], Mikko Särelä[b], Kari Visala[c,d], Janne Riihijärvi[e]

[a]*Nokia Siemens Networks, Linnoitustie 6, 02600 Espoo, Finland*
[b]*Ericsson Research Nomadiclab, 02420 Jorvas, Finland*
[c]*Helsinki Institute for Information Technology HIIT, Metsänneidonkuja 4 / Pilotti, Espoo, Finland*
[d]*Aalto University, Espoo, Finland*
[e]*Institute for Networked Systems, RWTH Aachen University, Kackertstrasse 9, 52072 Aachen, Germany*

Abstract

Locating objects with topology-independent identifiers has emerged as a key functionality in recent content networking approaches. Numerous designs have been proposed to address the obvious scalability and efficiency challenges such systems face in Internet-scale deployments. These designs have often been based on implicit assumptions of full deployment and a homogeneous autonomous system structure. Considering incremental deployment in a heterogeneous inter-domain setting, however, reveals both new scalability challenges and deployment and operation related disincentives.

In this paper, we propose an inter-domain rendezvous design that combines policy-based name routing between adjacent networks with hierarchical interconnection overlays for scalable global connectivity. This hybrid design enables partial deployment and explicitly addresses the different operational incentives and policies of network service providers and enterprise networks. Extensive domain-level simulations show good performance for our solution in terms of overlay-induced latency, inter-domain path stretch and routing load distribution.

Keywords: Name-based inter-domain routing, incentives, naming, network architecture, rendezvous

[☆] This work was supported by EU FP7 PSIRP project under grant INFSO-ICT-216173.

[☆☆] This work was supported by TEKES as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT).

[*] Corresponding author

Email addresses: `jarno.rajahalme@nsn.com` (Jarno Rajahalme), `mikko.sarela@nomadiclab.com` (Mikko Särelä), `kari.visala@hiit.fi` (Kari Visala), `jar@mobnets.rwth-aachen.de` (Janne Riihijärvi)

1. Introduction

Address-based inter-domain packet forwarding has been a foundational building block in the Internet architecture since its inception [1]. Applications, however, typically have little interest in the underlying network topology, and thus prefer using network topology independent names. This dichotomy between network practice and application preference requires applications to first resolve a name to a set of addresses, and then, as a separate transaction, send a communication request using one of the returned addresses to establish an end-to-end communication session.

The separation between name resolution and communication establishment phases can be problematic from both efficiency and reliability viewpoints. Recent work (e.g., [2]) has demonstrated that significantly better name resolution performance can be attained with more distributed systems. Reliability concerns come up when some of the multiple name servers involved in a query become unavailable, hence potentially preventing communication even when the end systems still have mutual connectivity.

Breaking the customary pattern, several architectural proposals [3–8] have morphed the name resolution phase into the communication set up phase. In these designs, the network routes the initial session establishment packets based on some notion of a *name* or an abstract *identifier*, rather than an inter-domain address. Further packets may, depending on the proposal, be mapped to lower-layer inter-domain paths for increased forwarding performance. The designs differ in their choice of *namespace properties*, such as the structure and semantics of the names they route on. What they share in common, however, is that the names employed in these designs can be used to name objects of different types (e.g., content, hosts, services), and that the names cannot be easily aggregated based on the topological location of the objects in the network.

Observing the multiple scalability, policy compliancy, and deployment related challenges with the prior art cited above, we propose an *inter-domain rendezvous architecture*, that routes communication requests to available replicas of named objects or services, without relying on central entities, such as DNS root servers. The design is based on the following objectives: 1. Use of a flat, self-certifying namespace, providing for efficient channel-independent security, 2. enterprise domains appear only as endpoints of any communication path through the rendezvous architecture, addressing the concerns for operational incentives, 3. rendezvous topologies are formed by willing participants only, so there need not be direct mapping to the underlying forwarding topology, enabling partial deployment, 4. communication locality is preserved whenever possible, addressing traffic policies, and 5. rarely referred reachability state is not distributed globally, allowing efficient operation and scalability.

To address the above mentioned objectives, our rendezvous architecture is comprised of two main subsystems: Firstly, domains in explicit name routing relationships may form *rendezvous networks*. The rendezvous networks are built according to the specific relationship types between domains (e.g., customer-provider, peering), accumulating all local name routing state toward the highest-tier providers of each rendezvous network [7]. This enables optimal inter-domain name routing paths to be taken within each rendezvous network. Secondly, these networks are connected together with *interconnection overlays*. These overlays consist of hierarchical distributed hash table structures [9], hosted by the top-tier providers of the participating rendezvous networks. This combination assures both strict locality guarantees and that name requests are never routed through arbitrary enterprise networks. Our evaluation shows that the resulting architecture has good performance in terms of overlay-induced latency, inter-domain path stretch, and routing load distribution.

In the following we first describe our research methodology (Section 2). Then we examine some of the fundamental aspects of networking namespaces (Section 3). We continue with an overview of the architectural aspects of our inter-domain rendezvous structure (Section 4). Then we describe our evaluation model (Section 5), and present the evaluation results (Section 6). Finally, we highlight our design choices against related work (Section 7, Table 1 on page 25), and conclude in Section 8.

2. Materials and Methods

We analyze the name routing related research reports from the period of last ten years, and find various scalability, policy compliancy, and deployment related challenges. We defer that material to Section 7 to allow for a proper comparison to our contribution (Table 1, on page 25). Based on our analysis of deployment incentives, we derive a novel architecture for inter-domain name-based routing, and evaluate the main metrics of the architecture using a domain-level[1] simulation model (see also Section 5).

[1] Packet, or even router level model would have been both more imprecise (as the router-level structure of the Internet is not known) and practically impossible due to the intractable processing and memory requirements. A

2.1. Evaluation Metrics

Our evaluation metrics are:

1. Additional initial routing *latency* induced by name-based routing

2. Policy-compliant inter-domain path *stretch*

3. Name-routing node *load distribution*

4. Caching efficacy

The *additional initial routing latency* is the most important of the above metrics, since name routing precedes the often user-initiated payload communication phase [11]. This metric models the time overhead incurred due to name-based routing, compared to forwarding on the destination address directly. In a name routing design, however, the destination address is not necessarily required, so the normally preceding name resolution step can in most cases be omitted, balancing the overhead incurred.

We define *policy-compliant inter-domain path stretch* as the ratio of the domain-level path length taken by the name routing messages to the optimal *policy-compliant* domain-level path from the source node to the destination rendezvous node.

The *routing node load distribution* and *caching efficacy* metrics are important from the system dimensioning viewpoint, as the user visible latency will increase if the system becomes overloaded.

2.2. Network Topology Model

We use the CAIDA inter-domain relationship datasets [12], which accurately represent the inter-domain transit structure of the Internet, but which are known to lack most of the lower-tier peering relationship detail [13]. We use the inferred inter-domain relationships information (i.e., the *customer-provider*, *peering*, or *sibling* relationships) to emulate the typical, incentive-derived BGP routing policies [14], selecting exported paths on the following order of preference: 1. Customer paths (*additional revenue*), 2. paths to siblings' customers (*revenue for siblings*), 3. peering paths (*revenue neutral* or *cheaper than transit*), and finally, if nothing else is available, 4. provider paths.

Point-of-Presence (PoP) level model would be more accurate and still tractable, but they are still work-in-progress within the research community [10].

Within each category, we select the shortest path among the possibly many available paths. Also the domains to which these paths are *exported to* are selected based on their mutual relationship. All *selected* paths are exported to customers and siblings, while only paths to customers and siblings are exported to peers and providers. These policies leads to inter-domain paths, that are, on the average, about 25% longer than the shortest possible valley-free paths [15].[2]

To assess the effect of the up to 90% of peering links reportedly missing from the CAIDA datasets [13], we form alternate inter-domain topologies by augmenting the CAIDA topology with 900% additional peering links according to the following simple rules:

1. All peering relationship at and above the domains with Route Views route monitors [16] are known, so none is added to these domains.

2. No peering for singly-homed stub domains. We assume that domains with interest for peering would first find interest for multi-homing.

3. No peering with (transitive) customers. Conversely, no peering with (transitive) providers.

Observing these rules we pick several different augmented datasets, each with peering domains selected with our traffic model (see below), so that the domains most likely to originate traffic are also the ones that are more likely to peer than others. As the peering relations are the most dynamic part of the Internet topology, the augmented datasets also provide a view on how our results might change when the Internet topology evolves.

To obtain estimates for the link latencies, we use simple averages over published path latencies and lengths: 34 ms for inter-domain hops and 2 ms for intra-domain router hops [17]. The number of intra-domain router hops used between overlay nodes residing in the same domain is approximated by $1 + \lfloor \log D \rfloor$ where D is the degree of the domain [18].

2.3. Traffic Model

We form the traffic model for the system by categorizing domains into different types, each playing a different role in the system both in terms of participation in name routing operation and in generating traffic. The categorization is given in [19], characterizing each domain in terms of the traffic volumes of three types of network *utilities*: *business access utility* ($U_{\texttt{ba}}$), *web hosting*

[2] We refrain from considering the hypothetical shortest paths in the non-annotated autonomous system graph.

utility ($U_{\texttt{web}}$), and *residential access utility* ($U_{\texttt{ra}}$). Business access utility is derived from the inter-domain transit hierarchy only, while the web and residential access utilities are derived from the transit hierarchy as random variates, observing the measured utility rank correlations and measured power-law distributions [19].

We assume the popularity distribution for the target objects to be Zipfian, following the established practice in the field [20–23]. Due to the power-law distribution, even modest caching is helpful in catching the demand for the most popular objects. However, due to the assumed heavy tail of the distribution, the system design can not rely on very high caching efficiency. We model the behavior of a typical cache by assuming that the pointers for most popular destinations (using Zipf exponent 0.91 as reported for DNS in [22]) can be found from the cache. To err on the safe side we set the in-memory cache size small enough for the average hit ratio to stay below 50%.

3. Namespace Design Considerations for Name-based Routing

Name routing designs separate their networked namespace from the underlying network topology. This enables both finding the closest replica of content or service, providing a name-based anycast primitive, and using multiple sources for efficient and robust transport and high availability. When routing on names, the structure and properties of the namespace becomes a major factor in determining both the amount of routing state in the network, and the processing load imposed on the networking elements. Scalability is not, however, the only significant design factor. Having lost the explicit binding to the underlying network topology, security becomes the driving concern in the system design. Finally, security and scalability need to be balanced with usability, lest the system remain a research proposal only.

In the following, we explore these properties in the name routing design space. We start by considering the aspects that are most visible to the end user (usability). Then we look at the more system-level aspects (security and trust), and finally the most technical aspects (efficiency and scalability).

3.1. Usability

First of all, networked names need to be usable for their intended uses. Currently, the most obvious use of names is when end users enter them in a browser's address bar. This use, while the most visible, represents only a small fraction of the total uses of networked names. Most often the names are invisible to the end user, as they are hidden behind symbolic (e.g., textual or graphic) links. Furthermore, the direct entry use has been greatly replaced by the use of search engines, recently furthered by the integration of the search and address bars in modern web browsers. This reveals the preference of the users to use plain language words to refer to the content they seek, rather than remembering and typing in the actual content names, or even consulting a bookmark list.

The problem with mnemonic names is that they are mnemonic, i.e., they map to existing symbols or concepts in the end user's head. This gives rise to phishing attacks, where an adversary intents to misuse these implicit mappings to gain the trust of the end user, when no such trust is warranted. As a result, the attacker will be able to speak for a trusted entity, while scheming to betray this (misplaced) trust for the detriment of the end user.

Consequently, building on both the lessened use, and inherent dangers of mnemonic content names, some (e.g., [7]) conclude that the human-mnemonic content names should be altogether dismissed with. We believe, however, that for the time being globally recognized mnemonic content names serve a useful purpose (e.g., in print advertising, bumper stickers). This means that the overall networking system should support using such names, but it does not necessarily follow that the network-wide name routing system should route on them. When the network does not act on human readable names, local, or user-level bindings can provide much of the needed functionality [24].

Finally, there is the issue of the usefulness of network names for applications. If a piece of content is intended for global use, its name must be usable as a reference from anywhere in the global network. This requirement necessitates global significance of the networked names.

3.2. Security and Trust

To be secured against attacks, name routing systems must be explicitly secured, as the (weak) security from "return routability," or the binding to the network topology, is not available any more. That is, when the aim is to use any available copy of the named content, rather than a copy from a known source, a *secure binding* between the content name and the content itself is needed.

Adding to the discussion on usability above, we conclude that *all* uses of mnemonic content names must be verified. That is, the authority of the content being associated with a specific mnemonic name must be explicitly checked each time any action is taken on the basis of such a name. Arbitrary names can be securely associated with any content using public key certificates, secure statements about name/content bindings given by trusted entities (e.g., trusted peers, or certificate authorities) [25]. However, this dependence on trusted third parties can be a burden on network elements, if they need to establish additional communication channels to verify whether given bindings should be trusted or not.

Use of self-certified names [26], however, limits the verification burden on one public key signature, establishing the trust that the name owner (i.e., the holder of the corresponding public key) has produced the signature over the content, thus proving that the name–content binding is valid. The binding from the content owner to the name is securely embedded in the name itself, as the name is essentially a secure hash of one of the owner's public keys. For some uses, such as securing the inter-domain routing system, this level of security is enough, as it allows the network to reject non-authentic routing state.

It should be noted that this reliance on self-certified identifiers shifts the burden for securing human readable name bindings to the end systems. However, as discussed above, not all uses of the network need be based on human entered content names, as, e.g., self-certified hyperlinks within secured documents are effectively certified by the signature covering the whole document, and thus need not be separately verified before used by the user. This suggests that *the overall verification burden in the networking system is reduced, when bindings to human readable names are verified in the end systems*, which is needed only when such names are actually entered by the end user.

Finally, end systems have more choices for trust establishment mechanisms and policies than what is possible for shared infrastructure. As an example, consider the difficulty of all network operators having to agree on a set of trusted Certification Authorities, when this problem is barely tractable within the context of a web browser application.

3.3. Efficiency and Scalability

All of the above factors have an effect on system-wide efficiency and scalability. As our aim is for Internet-scale system scalability, each design decision must be reflected against the scalability requirement.

The portion of the namespace visible to the basic network elements (e.g. routers) must therefore be as simple and as self-contained as possible. The remaining parts of the namespace can be managed by more specialized entities (e.g., content caches), and can have different efficiency and scalability requirements. This separation of concerns in the namespace design is reflected in our namespace design, defined in the following chapters.

4. Rendezvous Architecture

We define *rendezvous architecture* as an inter-domain networking system routing *communication requests* to available replicas of named objects or object collections. The semantics of the request processing upon arrival to the registering *rendezvous node* is not the concern of the rendezvous architecture. This separation allows the rendezvous infrastructure to remain generic, while specific application-level semantics can be implemented on top of it, using additional information elements in the rendezvous request messages.

The *rendezvous service model* consists of two phases of operation: Firstly, objects are registered in any rendezvous node(s) the object owner has a relationship with. Secondly, object users ask their local rendezvous nodes to route communication requests regarding the named object to a rendezvous node responsible for that object. The task of the rendezvous architecture is to bridge the gap between the respective rendezvous nodes.

Most of the prior work in the area is based on an implicit assumption of a homogeneous autonomous systems (AS) business structure. The AS structure comprising the Internet is, however, formed by domains having different operative incentives. The vast and growing majority, already more than 90%, of the ASes can be categorized as *enterprises*, or users of network services [27]. Conversely, the rest of the ASes can be categorized as *network service providers.* While it is common for an end-to-end forwarding path to pass through many competing transit networks, the end customers of the network service providers, however, expect their traffic not to pass through arbitrary other enterprise networks. Also, a typical enterprise network would not expect its network to provide transit services to other networks [15].

In practice, network service providers find incentives for new deployments at different times. This invalidates the common research assumption of availability of a new networking architecture between all pairs of domains neighboring on the forwarding level [28, 29].[3] These concerns call for *design for deployment* – mandating at least partial overlay solutions, as support by all domains can not be assumed – as well as for a design where end-to-end traffic is not handed to arbitrary

[3] This assumption is explicit in the *clean-slate* design approach [30].

domains in the Internet.[4]

Addressing the security, policy-compliancy, deployment, and scalability concerns, we focus on the following objectives for the inter-domain rendezvous architecture:

O1: Use of a flat, self-certifying namespace.

O2: Enterprise domains appear only as endpoints of any communication path through the rendezvous architecture.

O3: Rendezvous topologies are formed by willing participants only, so there need not be direct mapping to the underlying forwarding topology.

O4: Communication locality is preserved whenever possible.

O5: Rarely referred reachability state is not distributed globally.

We achieve these objectives by architecting a solution comprising of known component technologies as follows:

Namespace. Rendezvous namespace is based on flat self-certifying cryptographic labels without any predefined application layer semantics. We expect these labels to be used to name object collections that share common distribution and access semantics and call these collections *scopes*. This level of indirection takes care of a part of the global scalability challenge imposed by the flat namespace. The rendezvous requests may include additional naming information identifying individual objects, but these are processed by the rendezvous nodes and caches responsible for those objects, and need not concern the inter-domain name routing system.

Rendezvous Networks. Rendezvous nodes in enterprise networks advertise the availability of scopes in their domains by sending the related reachability state to their *rendezvous peers* and *rendezvous service providers*. These service providers further propagate this state to their rendezvous peers and providers. The top tier rendezvous service providers in this *local* hierarchy retain all registration state advertised by their peers and (transitive) customers. This process makes the registered object collections reachable in the set of domains thus defined. We call this kind of domain set a *rendezvous network* (see the shaded areas in Figure 1).

[4] The second consideration significantly limits the available path choice in the inter-domain networking system. This fact can be factored in the design, however, enabling effective and realistic path selection mechanisms.

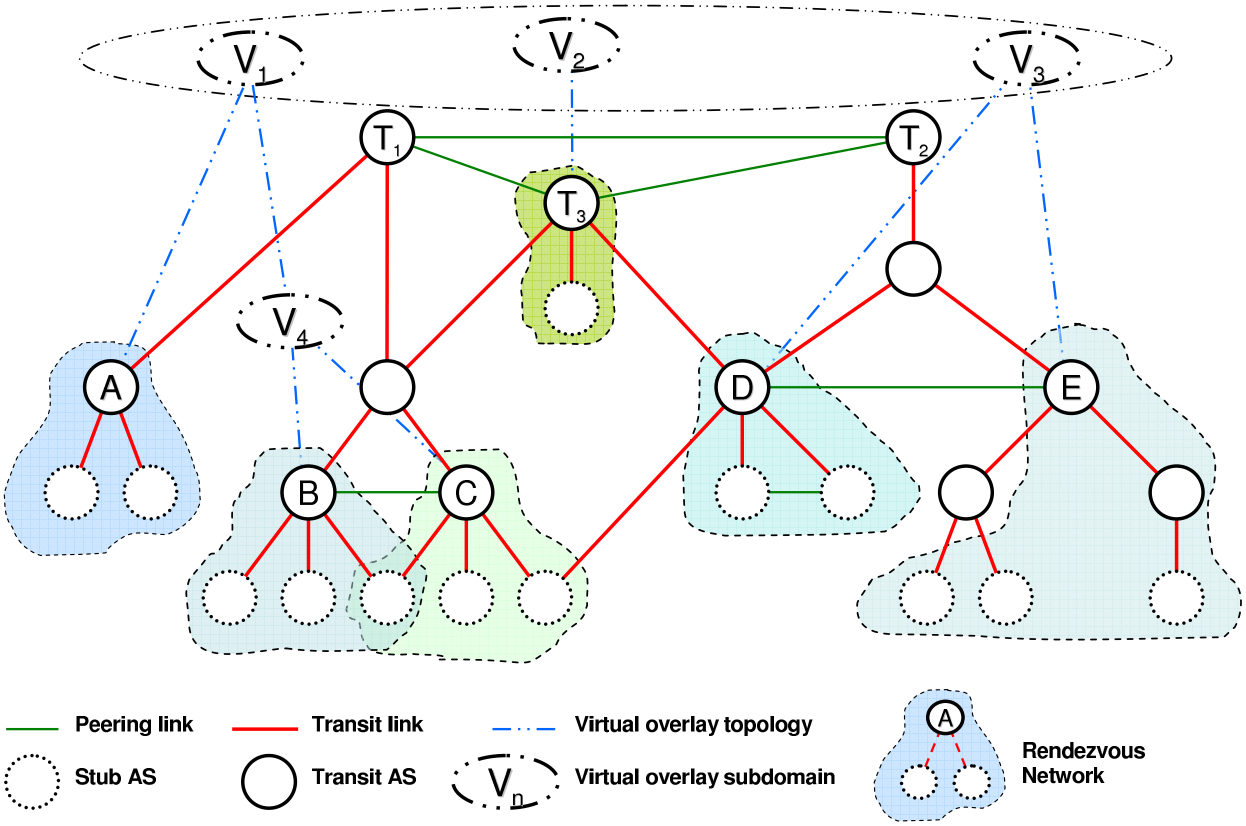


Figure 1: Rendezvous network interconnection. Solid lines represent the packet forwarding level inter-domain structure. Dashed lines represent the interconnection overlay. Rendezvous nodes are located in rendezvous networks (the shaded areas), individual rendezvous nodes not shown.

Interconnection overlay. The rendezvous service providers serving as roots of their respective rendezvous networks interconnect using an *interconnection overlay* (the dashed lines in Figure 1). The overlay is used to store (*scope identifier*, *pointer*) tuples, pointing to the rendezvous nodes maintaining the reachability state for the named object collections. Immediately upon encountering such a pointer in the overlay, the communication request is forwarded to the responsible rendezvous node using the forwarding path stored with the pointer. The destination rendezvous node can be hosted by any network reachable either on the underlying packet network or via the interconnection overlay.

Amongst the many available overlay designs, we have chosen Canonical Chord [9], also used in ROFL [6]. However, we apply the overlay strictly between willing rendezvous service providers, thus maintaining the objectives O2 and O3 on the overlay level. The stated locality preservation objective (O4 above) is maintained by storing pointers to responsible rendezvous nodes on each Canon hierarchy level, starting from the rendezvous network hosting the responsible rendezvous node.

On the top-level of the interconnection overlay we use identifier prefix based latency optimization [9], where each node maintains links to all other nodes in its own prefix group, and at least one link to any node in each of the other prefix groups. In practice, this is an additional global overlay layer on top of the locality-preserving hierarchical overlays. However, the combined structure allows for significant reduction in overlay link maintenance overhead, as each overlay node needs top-level links only for prefixes within its own portion of the identifier space in the Canon hierarchy. Thus, the more there are nodes in the local hierarchy, the less global level links are needed at each node, and vice versa.[5]

The overlay structure is virtual and does not require specific support from non-participating networks, such as upstream transit providers. 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. To this end, we expect the service providers on common hierarchical levels (depicted by e.g. V_4 in Figure 1) to cooperate only with parties they can enforce contracts with (e.g. domains B and C forming the V_4). On the top level (V_1–V_3), a weaker level of trust has to be assumed, as all rendezvous service providers need to participate.

Rendezvous Request Routing. In summary, rendezvous requests are routed in phases as follows:

1. *Intra-domain* within the requesting domain. Sufficient name-level forwarding state is maintained at local rendezvous nodes to keep local traffic local.

2. Within the *local rendezvous network*, following inter-domain rendezvous adjacencies, and observing the established rendezvous state [7].

3. Within the *locality-preserving interconnection overlay*, hierarchically bottom-up, following the overlay forwarding state [9].

4. On the *top-level of the interconnection overlay* using scope identifier prefix based shortcuts.

[5] To our knowledge this result has not been reported before.

5. *During any of the above steps*, whenever an explicitly instantiated or cached object pointer is found, the rendezvous request is forwarded to the rendezvous node responsible for the named object, using the included path. The path may take place via the overlay (e.g., to address issues with non-transitive connectivity [31]), or directly via the underlying packet network, whenever such connectivity exists.

6. Cached entries can be explicitly invalidated by routing the request back to the node that acted on a stale cached pointer. The request routing then resumes normally, as if the cached entry did not exist. The forwarded request also indicates that the specific cache entry is invalid, preventing other nodes acting on stale forwarding state.

7. Rendezvous nodes send responses via a path recorded in the request message. This path needs to have one entry for each traversed underlying routing domain. In the case of global connectivity at the underlying packet layer the response is sent directly to the requesting rendezvous node, allowing for minimal latency.

8. Based on the rendezvous response, the requesting rendezvous node may instantiate cached pointers in the overlay as guided by its local policy.

5. Evaluation Model

We assess the practicality of our rendezvous design in a network ecosystem resembling today's Internet. To that end, we construct a domain-level rendezvous model using the present-day autonomous system structure (see Section 2.2) and traffic patterns (see Section 2.3). The model captures the essential structures of the design presented in the preceding section: The rendezvous networks, locality-preserving hierarchical overlays, and the global level overlay structure. Using these structures, we execute rendezvous request routing (phases 1–5 described above), and use the resulting routing paths to compute the distributions for our evaluation metrics (Section 2.1). As stated in Section 2.3 above, we implement an abstracted caching model, assessing cache hits based on target scope popularity ranks. Thus we need no warm-up nor actual storage for a cache in each simulated node.

In order to tease apart the performance contribution of each of these components we also evaluate the same traffic cases with alternative structures, as follows:

Rendezvous Networks with Canon(3) With rendezvous networks, and with 3 levels of Canon hierarchies, clustered according to the topological distance between the overlay nodes, meeting all of our design objectives.

Rendezvous Networks With rendezvous networks, but without the Canon hierarchies. This option invalidates our objective O4.

Canon(3) Without rendezvous networks, but with 3 levels of Canon hierarchies. Also enterprise ASes participate in the overlay, invalidating the policy objective O2.

Global Prefix-root only All ASes in the overlay, and no local Canon hierarchies. This invalidates objectives O2 and O4.

Chord Global Chord overlay without any hierarchical structure. Also this invalidates objectives O2 and O4.

All distributed structures above use aggressive latency optimization by preferring topologically close nodes in their overlay link selection. In addition to the above structural variants we also evaluate our target design (Rendezvous Networks with Canon overlay) with an alternative AS topology, adding 900% more peering links (See Section 2.2).

To form the simulated rendezvous networks we assume the transit service providers to offer rendezvous as a service for their customers. The customers can be either stub networks, or small transit networks, measured by the number of domains reachable via their customer links. This provides a sufficient approximation of the division between enterprise and network service provider domains. Next, the rendezvous networks are clustered together in hierarchical overlays. Clustering is based on locality of the rendezvous providers, given our assumption that networks topologically close to each other might be willing to contain their mutual rendezvous messaging within their networks (the objective O4 above).

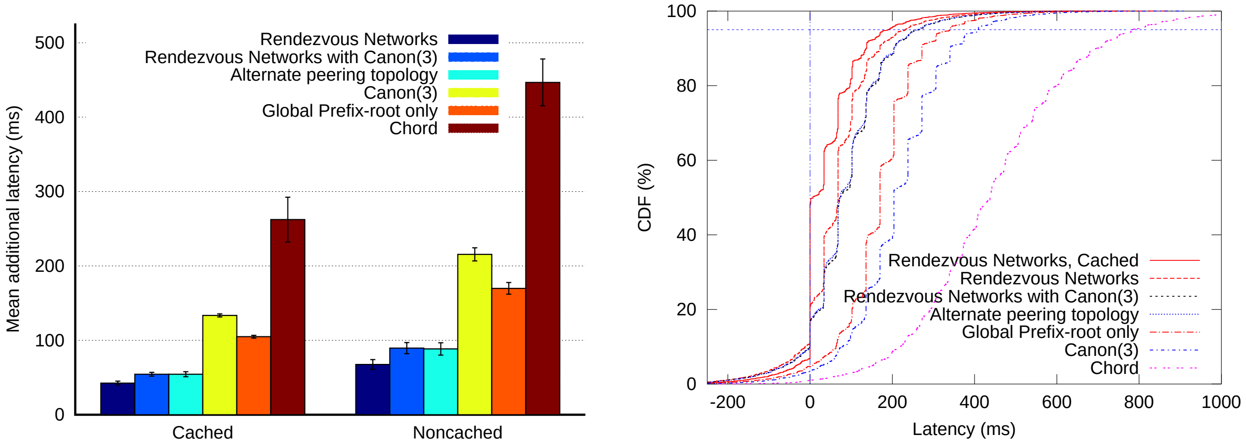
The number of required overlay nodes is an important aspect to the network formation, and depends primarily on three factors: 1. The number of ASes hosting overlay nodes; each needing a minimum of one node, 2. The expected numbers of both registrations created by the content providers and requests initiated by the customers, and 3. the required in-memory storage for object pointers and overlay overhead. To be on the safe side, we assume the total number of globally reachable object collections in the system to be at least an order of magnitude higher than the number of registered domain names in the DNS today, i.e., around 10^{10} globally resolvable scopes in total. This does not limit the number of reachable individual objects in any way, as each scope can provide access to multiple objects.

We assume each object pointer to take about 64 bytes: 32 bytes for the object identifier, up to 16 bytes for the next hop overlay node identifier or IP address, if routing directly on the underlay, and some reserve for the overhead of the indexing data structures. We store an object pointer in each level of the overlay hierarchy, and also distribute the pointer at the global level to achieve both fault tolerance and better latency performance. Taking these overlay overheads into account, we estimate that a typical server could hold on average of 10^6–10^7 unique local object registrations in memory. Taking all three factors together, we get an overlay of a few thousand nodes for our target design, or about 10x more for the evaluation variants without rendezvous networks.

The major deficiencies in the presented model include the lack of modeling of link or node failures, or deflection, the untrustworthy operation by some of the overlay participants. The model also lacks estimation of the actual computational load imposed by the overlay maintenance protocols and the request routing and cache management processes themselves. Addressing the latter concern, we point to the existence of servers processing similar loads today (e.g., DNS, TLS). For the former, we use replication to keep the rendezvous pointers in multiple overlay nodes. Thus, only the failure of all these nodes would make the related information unreachable. Also, in practical systems, such failures are detected and the related state is further replicated to remaining overlay nodes. Addressing lying nodes, overlay nodes could ask a random set of other nodes to test the reachability for the objects registered by the first node, thus revealing any errant behavior. Then, 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 failing or lying node from the overlay.

6. Evaluation Results

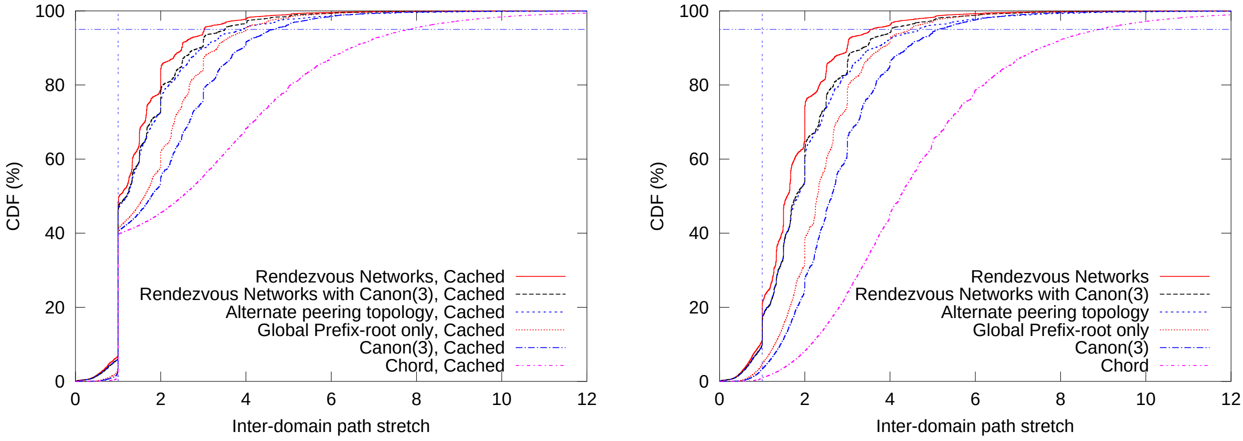
Figure 2(a) shows the mean latencies of each of our evaluation cases, the most important performance indicator from the network user viewpoint, with and without caching. Figure 2(b) shows the cumulative distribution functions (CDFs) of select cases, most cases shown without caching to amplify the differences. Negative values are due to overlay nodes offering shortcuts to otherwise



(a) Rendezvous overhead mean latencies (ms).

(b) Rendezvous overhead CDF (ms).

Figure 2: Rendezvous overhead mean latencies with 95% confidence intervals and latency overhead cumulative distributions for select overlay structures (ms). Lines for 0 ms and 95% included for reference.



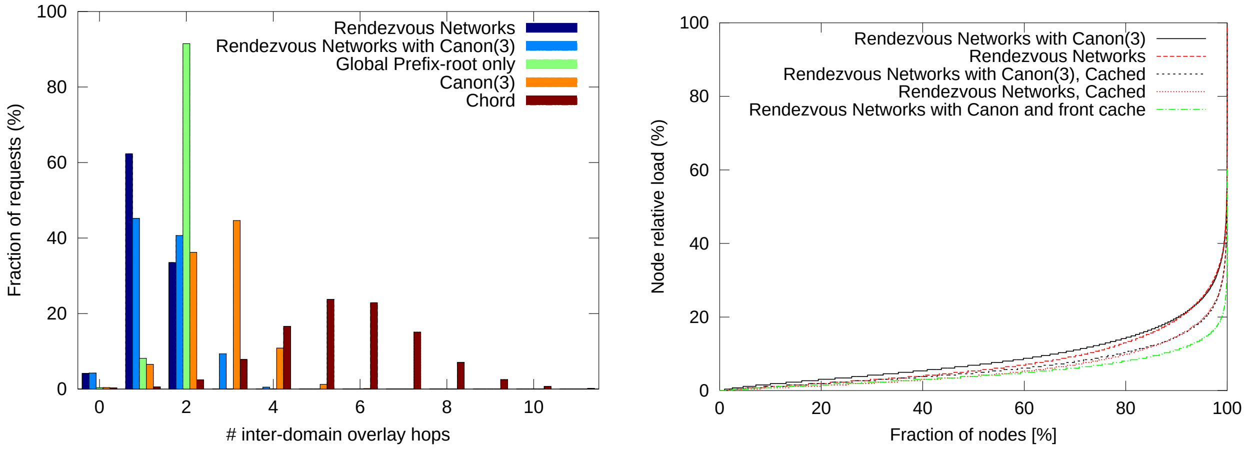
(a) Domain-level stretch CDF with Cache.

(b) Domain-level stretch CDF without Cache.

Figure 3: Rendezvous stretch cumulative distributions, with and without caching. Lines for stretch 1 and 95% included for reference.

policy-constrained end-to-end paths. Most of the 95^{th} percentile figures compare favorably to the above 1000 ms figures reported for DNS [22]. Local caching in enterprise ASes and end nodes (not modeled) will make the user perceivable performance even better.

Figure 3(a) reports stretch, the ratio of domain-level rendezvous path to the comparable shortest policy-compliant path. As with the negative latencies, the small number of requests with stretch below one are due to overlay nodes offering a shortcut via links otherwise not available for policy-



(a) Inter-domain rendezvous overlay hop distributions.

(b) Relative overlay node load distribution.

Figure 4: Overlay inter-domain hops and node load distribution, with and without caching.

compliant end-to-end paths. The obvious comparison point is DONA [7], which provides stretch of 1, when mapped universally to the underlying packet level transit hierarchy. Our figures are good, considering the decrease in the required number of servers and our avoidance of the universal deployment requirement. Also, this stretch only applies to the initial rendezvous exchange, as the payload communication takes place over optimal policy-compliant paths.

We see that our objective of locality preservation has lead to a slightly increased latency and stretch compared to the case where Canon hierarchies are not used. In retrospect this is as expected, since each level of hierarchy typically adds an additional overlay link to go through. More accurate caching and latency models are needed to fully quantify this difference.

The distributions of inter-domain overlay hops for the different overlay structures are shown in Figure 4(a), while Figure 4(b) shows the distribution of node load, measured by the relative number of times each overlay node forwards or handles a rendezvous request message during a simulation run of 30000 requests, repeated for 20 independent replications.[6] The x-axis indexes the overlay nodes, and the y-axis shows the relative node load, compared to the most loaded node of all the simulation cases included. 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.

The effect of caching on node load was firstly surprisingly low, but closer analysis shows that a

[6] With 300000 rounds the distributions remain the same and means stay within the reported confidence intervals.

share of the overlay load is due to the rendezvous network local demand, rather than load imposed by the other overlay nodes. The bottom dashed line ("front cache") shows the overlay node load in the case where local caching is separated from the overlay nodes.

We have included several important reference cases in Figures 2 – 4: One with an alternate AS topology with additional peering links, and several full overlay variants covering all ASes (dismissing our rendezvous network model). We report the cases with no caching to better show the differences.

The mean effect of the additional peering is small, but some queries face higher stretch, but more because of the shorter average path lengths, than due to longer overlay distances, since the additional latency shows no real difference between the two cases. The ground truth likely lays between these cases, as our simple model for additional peering disregards distance, and in many cases adds peering links between ASes that could not physically peer with each other. Overall, it seems that our model is relatively robust to even large shifts in the underlying AS topology.

The full overlay cases allow comparing the performance of our provider-based model against variants where there are only singular rendezvous networks, i.e. a model where each AS operates as their own rendezvous service provider. It comes as no surprise that a bigger overlay leads to higher overheads, in addition to defeating our policy objectives. This shows that our incentive-based separation between the roles of the ASes may indeed perform better than the variant without such separation.

7. Related Work

Many of the basic insights to network architecture deployability are stated in [28]: New architecture evolution starts with partial deployment, which is made possible by anycast service provided by the underlying, old architecture. However, [28] simply assumes existing contractual agreements and market structure, which seems to be at odds with their revenue flow assumption: There are no ASes which would universally benefit from operating as new architecture detours, as even tier-1 ISPs may need to compensate for excess traffic they send to their peers.

Therefore, in our architecture we assume contractual agreements between the users of the network and their rendezvous providers. These could be existing or new contracts, especially created for rendezvous purposes. In some scenarios the compensation for rendezvous providers may be possible via other channels of revenue, e.g., advertising.

Deployability concerns have also been studied in [32–36], but deployment incentives and their effects on technology design are rarely considered, and usually only from the the point of view of a single investing entity.

OceanStore [3] presents an architecture for global, location independent storage of versioned files. OceanStore clients connect to “data pools”, which manage all the data using highly interconnected networks. OceanStore uses self-certified names file names [24, 26], which are essentially pseudo-random, binary strings. Human readable names are resolved via directories, some of which are chosen by users to serve as their (trust) roots. The OceanStore design employs two separate strategies for routing based on the file name: Firstly, via a probabilistic local algorithm, using attenuated Bloom filters [37], and secondly via a global distributed hash table (DHT) structure, with probabilistic locality guarantees. On the contrary, our design explicitly addresses the administrative domains comprising the Internet, seamlessly combines the local and global routing producing low stretch, and employs an explicitly hierarchical DHT structure with strict locality guarantees, that are not available in flat DHTs.

[38] describes a geographical tree structure used for finding geographically close location records for data identified with flat identifiers. We use similar but more general structure (not restricted to the tree structure) within our *rendezvous networks*, beneath the Canon hierarchies, but build the hierarchy on the network service provisioning topology, rather than a separate, non-contractual geographical topology. The upper levels of the tree (e.g., ”world”) are essentially replaced by the prefix-based Canon top-level structure.

SFR [39] presents a design for *semantic free referencing*, allowing the untangling of the Web from DNS. SFR uses a flat DHT (e.g., Chord) for mapping SFRTags (a hash including a public key and a salt value) to *o-records*, that contain the objects location(s). Instead, we follow the route-by-name paradigm, eliminating the separate lookup phase. However, our use of a hierarchical DHT structure could also be applied to SFR design.

CoDoNS [40] has clearly established that a more distributed structure to name resolution could bring considerable performance benefits. However, while we use such a structure in our design, we forgo with the prevailing resolution-before-communication-setup model altogether, and route communication requests directly on names.

The concept of *rendezvous-based communication abstraction* was introduced in Internet Indirection Infrastructure (*i*3) [5]. However, *i*3 operates directly on Chord [41], making all the packets pass through the overlay structure. Unfortunately, Chord has no regard for domain-specific routing policies, so the *i*3 nodes operate as arbitrary "detours" [42] for all traffic. Arguably, there are no ASes for which such operation would be universally beneficial without new, *i*3-specific revenue. In our design rendezvous serves as a combined resolution and set up phase for end-to-end communication; the rest of the communication exchange is forwarded separately, using optimal policy-compliant paths.

ROFL [6] proposes Internet-scale routing on flat labels without assuming underlying IP forwarding. ROFL, while providing a general packet routing service like *i*3, addresses some of the overlay-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 [43], which defines a specific link-state protocol for maintaining an up-to-date view of the *upgraph*, the available uphill and downhill [15] paths between any user of the network and the other domains either via the tier-1 networks or some lower tier peering links.

Nevertheless, ROFL assumes that all ASes serve in similar roles, and as a result suffers from a policy-compliancy issue similar to *i*3: Since each AS is a participant in the ROFL inter-domain overlay, it is highly likely that the initial packets between any two enterprise ASes are routed via third party enterprise ASes. In the extreme, this behavior may enable arbitrary third parties to listen in on the communication set up phase of other networks. We design around this issue by separating the concerns of enterprises from those of the network service provider in the rendezvous architecture, and use the hierarchical Chord structure [9] only for rendezvous network interconnection.

TRIAD [4] and DONA [7] are examples of inter-domain networking designs where a name-based routing phase precedes the payload communication phase. TRIAD uses a BGP-derived routing design, but on the level of servers identified with DNS names ("BGP with names"). DONA, however, uses self-certified identifiers instead of human readable names, and makes the *registration* and *find* messages explicitly follow the underlying provider and peering hierarchy.

Both TRIAD and DONA assume all ISPs to be naturally willing to peer on the name level, if they are already peering on the underlying packet forwarding level. Due to this arguably unrealistic assumption their request messages are always forwarded on optimal policy-compliant paths. This has a downside, however, as the amount of name-level forwarding state needed to achieve this also leads to significant scalability challenges. TRIAD limits the problem by restricting the managed namespace to service names, while DONA argues that large data centers can handle the load. Due to the allowed *wildcard principal* practice, DONA design burdens the tier-1 domains with memory, processing and communication overheads scaling linearly with the number of *individually registered* publications. While it is possible that all the tier-1 transit providers find incentives for the implied investments (as detailed in [7]), we do not assume that to be the case. On the contrary, we assume the incumbent tier-1 transit providers to not be among the early adopters of any networking architecture with the potential for better utilization of either peering links or local storage, since such efficiencies will limit the growth of transit traffic [44]. *This assumption shifts the global DONA registration state burden to large number of smaller administrative domains.* We divert from both of these designs by *not* assuming the rendezvous topologies to necessarily mirror the underlying packet forwarding topology, especially so at the upper tiers of the transit structure.

CCN [8] defines a combined content networking and transport layer framework. CCN naming is based on hierarchical textual names, making it necessary for the network equipment to fetch and verify certificates along arbitrary certification chains when securing the state maintained by the network [25]. In our analysis, this enables intractable attack vectors against the network, as attackers can devise arbitrary certification chains. Our use of self-certified names allows such authenticity checks to be performed in one step without any third party involvement. Also, the CCN inter-domain routing model relies on DNS for resolving destination server addresses when direct CCN relationship does not exist between adjacent administrative domains. Our rendezvous model can be considered addressing this missing inter-domain aspect of the CCN design.

As a summary, we highlight our design choices in the last row of Table 1 (page 25).

8. Conclusions

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

Deployment of any new architecture takes place one step at a time, each step taken by individual stakeholders acting on their own incentives. It is possible, and even likely, that this process never achieves full deployment over the existing networks. Accepting this, we recognize the need for solutions for interconnecting the deployed parts. Furthermore, while interconnection overlays are needed for deployment purposes, the participation in such overlays may need to be limited to network service providers. This does not, however, limit the ability of the enterprise networks to form their own overlays, where only known and trusted entities may participate.

While the presented heterogeneous design is necessarily more complex than any of the homogeneous predecessors, 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 incentive challenges in the proposed rendezvous based communication abstraction, and more generally in name-based inter-domain routing systems. Even though the performance figures for our design seem encouraging, the intricate issues of trust in shared inter-domain structures require more careful analysis.

References

[1] B. Leiner, V. Cerf, D. Clark, R. Kahn, L. Kleinrock, D. Lynch, J. Postel, L. Roberts, S. Wolff, A brief history of the Internet, ACM SIGCOMM Computer Communication Review 39 (2009) 22–31.

[2] V. Ramasubramanian, E. G. Sirer, The Design and Implementation of a Next Generation Name Service for the Internet, in: SIGCOMM'04. Proceedings, volume 34, ACM, pp. 331–342.

[3] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, C. Wells, B. Zhao, OceanStore: An Architecture for Global-Scale Persistent Storage, in: ACM ASPLOS-IX. Proceedings, volume 28, pp. 190–201.

[4] M. Gritter, D. R. Cheriton, An Architecture for Content Routing Support in the Internet, in: USENIX USITS'01. Proceedings.

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

[6] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, S. Shenker, ROFL: Routing on Flat Labels, in: SIGCOMM'06. Proceedings, ACM.

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

[8] V. Jacobson, D. Smetters, J. D. Thornton, M. Plass, N. Briggs, R. L. Braynard, Networking Named Content, in: ACM CoNEXT'09. Proceedings.

[9] P. Ganesan, K. Gummadi, H. Garcia-Molina, Canon in G Major: Designing DHTs with Hierarchical Structure, IEEE Distributed Computing Systems (ICDCS'04). Proceedings (2004) 263–272.

[10] Y. Shavitt, N. Zilberman, A Structural Approach for PoP Geo-Location, in: NetSciCom'10. Proceedings.

[11] J. Brutlag, H. Hutchinson, M. Stone, User Preference and Search Engine Latency, Technical Report, Google, Inc, 2007.

[12] CAIDA, The CAIDA AS Relationships Dataset, August 10th, 2009. `http://www.caida.org/data/active/as-relationships/`.

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

[14] M. Caesar, J. Rexford, BGP routing policies in ISP networks, IEEE Network 19 (2005) 5–11.

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

[16] Routeviews, Route views peers, http://www.routeviews.org/peers/, 2009.

[17] B. Zhang, T. Ng, A. Nandi, R. Riedi, P. Druschel, G. Wang, Measurement-Based Analysis, Modeling, and Synthesis of the Internet Delay Space, in: ACM SIGCOMM IMC'06. Proceedings, pp. 85–98.

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

[19] H. Chang, S. Jamin, Z. Morley, M. W. Willinger, An Empirical Approach to Modeling Inter-AS Traffic Matrices, in: ACM SIGCOMM IMC'05. Proceedings, pp. 139–152.

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

[21] P. Gill, M. Arlitt, Z. Li, A. Mahanti, YouTube Traffic Characterization: A View From the Edge, in: ACM SIGCOMM IMC'07. Proceedings, pp. 15–28.

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

[23] W. Willinger, D. Alderson, J. Doyle, L. Li, More normal than normal: scaling distributions and complex systems, in: Winter simulation 2004, Proceedings, Winter Simulation Conference, p. 141.

[24] R. Rivest, B. Lampson, SDSI – A Simple Distributed Security Infrastructure, Technical Report, MIT, 1996.

[25] D. Smetters, V. Jacobson, Securing Network Content, Tech. Report, Palo Alto Research Center, 2009.

[26] D. Mazières, M. Kaminsky, M. F. Kaashoek, E. Witchel, Separating key management from file system security, ACM SIGOPS Oper. Syst. Rev. 33 (1999) 124–139.

[27] A. Dhamdhere, C. Dovrolis, Ten years in the evolution of the Internet ecosystem, in: SIGCOMM IMC'08. Proceedings, ACM, pp. 183–196.

[28] S. Ratnasamy, S. Shenker, S. McCanne, Towards an Evolvable Internet Architecture, in: ACM SIGCOMM'05. Proceedings, pp. 313–324.

[29] S. Shenker, L. Peterson, J. Turner, Overcoming the Internet Impasse through Virtualization, Proceedings of

ACM HotNets-III (2004).

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

[31] M. J. Freedman, K. Lakshminarayanan, S. Rhea, I. Stoica, Non-transitive Connectivity and DHTs, in: WORLDS'05. Proceedings, USENIX.

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

[33] N. Feamster, H. Balakrishnan, J. Rexford, Some Foundational Problems in Interdomain Routing, in: Proceedings of ACM HotNets-III.

[34] P. Jacob, B. Davie, Technical challenges in the delivery of interprovider QoS, Communications Magazine, IEEE 43 (2005) 112–118.

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

[36] D. Waddington, F. Chang, Realizing the transition to IPv6, Communications Magazine, IEEE 40 (2002) 138–147.

[37] B. Bloom, Space/Time Trade-offs in Hash Coding with Allowable Errors, Communications of the ACM 13 (1970) 422–426.

[38] G. Ballintijn, M. Van Steen, A. Tanenbaum, Scalable Human-Friendly Resource Names, IEEE Internet Computing (2001) 20–27.

[39] M. Walfish, H. Balakrishnan, S. Shenker, Untangling the Web from DNS, in: USENIX NSDI'04. Proceedings.

[40] V. S. Ramasubramanian, Cost-Aware Resource Management for Decentralized Internet Services, Ph.D. thesis, Cornell University, 2007.

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

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

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

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

Design	Namespace	Security	Routing granularity	Deployment	Scalability	Efficiency	Policy compliancy
OceanStore [3]	Secure hash (key+label)	Self-certified, (label not authorized)	Individual files	Partial/ Overlay	Provider-based flat global DHT	Probabilistic local search	No explicit relation to AS-structure
TRIAD [4]	DNS names	none	Server names	Universal	DFZ:[a] whole namespace	Stretch 1[b]	As in BGP
ROFL [6]	Secure hash (key)	Self-certified	Individual hosts	Universal	Hierarchical DHT	High DHT penalty[c]	No AS-level policies
DONA [7]	Secure hash (key) + label	Self-certified, key and label not bound	Individual data items + aggregation	Universal	DFZ: All individually registered data	Stretch 1	Valley-free, no explicit AS incentives
CCN [8]	DNS-based hierarchy	3rd party cert. name/content binding	Individual data packets	Local + bridging via DNS	Not scalable to transit routers	Name resolution penalty[d]	As in BGP
Rendezvous (this paper)	Secure hash of the owner's key	Self-certified	Object collections (scopes)	Partial/ Overlay	Provider-based hierarchical DHT	Locally explicit policy-compl. hierarchies	Hierarchical DHT, providers only

Table 1: Feature comparison between various name routing designs. Last line summarizes our design decisions (Section 4).

[a] Default-free zone, autonomous systems (ASes) without default routes.

[b] Given that all ASes participate and maintain routing state for all names. ASes with only one provider can use default routes.

[c] For the initial packet exchange only.

[d] In non-adjacent inter-domain cases.