Towards Publish/Subscribe Internetworking

Jarno Rajahalme
Nokia Siemens Networks Oy*
firstname.lastname@gmail.com

Abstract

The internetworking architecture of the Internet of today has served us well, enabling, perhaps surprisingly, the exponential growth of the network so far. Some of this growth has been gained at the cost of architectural evolvability. The unplanned deployment of Network Address Translators, for example, makes the Internet architecture too rigid for further evolution. It seems an architectural overhaul is required to overcome the current limitations. This paper reviews the growing body of research proposing directions for Internet Architecture renewal, and then shows how these contribute to a Publish/Subscribe based Internet in context of the vision of Content-Centric Networking.

KEYWORDS: Internetworking, Network Architecture, Routing, Publish/Subscribe, Content-Centric Networking

1 Introduction

Recently, there have been proposals for creating a new Internet architecture based on the Content-Centric Networking concept: instead of focusing on sending packets from one network interface to another, the network should be centered around the data (or information) being requested and transferred [16], [18]. Most of the popular applications are already providing the user an information centric view on the Internet: Users are not concerned about where the information they seek is located,[1] but that they get the information they want.

The current send/receive model of internetworking is a particular extension of an Inter-Process Communication (IPC) model of computer operating systems[8]. Choosing a different model may yield a different network architecture. The publish/subscribe model is a promising approach to an alternative internetworking paradigm.

The publish/subscribe model is deceivingly simple: The clients of a publish/subscribe network may *advertise* specific *publications* and correspondingly *subscribe* to specific publications. The publish/subscribe network builds internal state based on this, so that it is then ready to *deliver* any *published* publications to the clients with matching subscriptions. What is characteristic about this model is that *the network nodes need not be explicitly named*. Exploiting this characteristic will fundamentally impact the resulting architecture.

*Manuscript received April 17, 2008. This work was supported by the EU FP7 PSIRP project under grant INFSO-ICT-216173.

[1] Except for the trustworthiness of the provider of the information.

In this paper we first review the relevant state-of-the-art (section 2). Then we distill the main inferences from the presented material (section 3). After that we outline how this, in the context of the simple publish/subscribe model, leads us towards publish/subscribe internetworking architecture (section 4). Finally, we conclude the paper in section 5.

2 Review of selected Internet architecture research

In this section we introduce a selected subset of the relevant existing work on which we build our results. This comes in two rough categories: Overall architecture issues (section 2.1), proposed routing schemes (section 2.2).

2.1 Internet architecture models

2.1.1 FARA: Reorganizing the Addressing Architecture

FARA [5] defines an abstract model for an end-to-end network architecture. FARA abstracts the whole packet network as *communication substrate*,[2] which provides packet routing and forwarding between *entities*, the abstract endpoints of network communication. Communication substrate carries the packets labeled with *forwarding directives* (FDs) through the network and to a *slot* where the destination entity is bound to. The entities manage multiple *associations* internally with *association IDs* that are defined by the destination entity itself. Notably the association ID is separated from the forwarding directive, so the entity may move to new locations (new FDs) without a need to change the association identity. There is no global namespace for the entities or the associations.

To make entities reachable without a global namespace, FARA defines a *rendezvous* mechanism and the *FARA directory service*. Rendezvous allows communication discovery with service names (the *rendezvous information string*), followed by initiation with a handshake to create a new association (with internal association state). The postulated directory service provides mapping from generic names (e.g. FQDNs or local names) to FD, RI-string pairs. The final disambiguation between potentially multiple FD, RI-string pairs is the responsibility of the rendezvous mechanism.

FARA itself does not cover the structure of the network at all, thus it does not give any specification for the forwarding directive. M-FARA, the accompanying instantiation of FARA, defines mobility agents that provide some ren-

[2] Corresponding to the 'communications subsystem' in the original end-to-end arguments paper [20].

dezvous infrastructure on top of the assumed packet delivery substrate. M-FARA proposes to use addressing realm specific source routing (represented within the FD) as a solution to span multiple address spaces.

FARA sets an example for how network architecture can be designed and reasoned on an abstract level, and only after sufficient coherence has been reached are the details of protocol design brought in via instantiating the abstract architecture to a (protocol) engineering solution.

2.1.2 Addressing Reality: An Architectural Response to Real-World Demands on the Evolving Internet

In [6] Clark et al. introduce the concept of *controlled transparency*, and argue for the *packet* as a fundamental invariant in the Internet Architecture. This leaves open the role of current packet design (the 'stateless connectionless datagrams') in future architectures.

2.1.3 Holding the Internet Accountable

In [1] Andersen et al. argue for a new Internet addressing structure, where the subnet prefix is replaced with self-certifying Autonomous Domain identifier (AD) and the remaining suffix (called the interface identifier in the IPv6 addressing architecture) is replaced with a self-certifying host identifier (EID). IP addresses would then take the form AS:EID. The private keys bound to the AS and EID are held by the domain and the host, respectively.

The design is based on the finding that the inter-domain routing protocols would scale much better, if the routing is done on the domain level, and that the self-certifying property would protect against the nowadays common BGP spoofing attacks (hijacking more specific prefixes etc.). The EID part is globally unique by itself, which would enable hosts moving of multi-homing between domains so that the correspondent will have assurance that it is still the same host.

2.1.4 Steps Towards a DoS-resistant Internet Architecture

In [13] Handley and Greenhalgh present seven steps towards an Internet architecture that is resistant against denial-of-service (DoS) attacks. We'll summarize the steps here:

1. *Separate Client and Server Addresses*: This prevents peer-to-peer communication and effectively limits distribution of Internet worms.

2. *Nonglobal Client Addresses*: Clients don't need global addresses. Authors propose domain-by-domain path-based addresses that are formed as clients' packets are forwarded towards servers. Servers will then be able to respond to clients using these domain-level paths. The labels have only a local significance, so they cannot be exchanged between servers.

3. *RPF Checking of Server Addresses*: Reverse-path forwarding checks at domain boundaries make it harder for servers to spoof source addresses.

4. *State Setup Bit*: Packets causing state set-up (e.g. TCP SYN) must be marked with a specific bit. This way these packets can be chosen for closer inspection e.g. at domain boundaries. The use of the bit is enforced by all (compliant) nodes refusing to set up any state due to packets that don't have the bit set.

5. *Nonce Exchange and Puzzles*: These offer tools to balance the costs of setting up communication state between clients and server. Busy servers (maybe under attack) may demand clients to solve cryptographical puzzles before accepting any communication state. Also middleboxes such as firewalls could validate the clients with nonces and rate-limit clients with puzzles.

6. *Middlewalls*: Moving filtering functionality closer to clients would help against distributed denial of service attacks. Locating this functionality to specific middlewalls offers an alternative deployment model for the proposed pushback mechanisms [15]. The middlewalls would not normally filter the traffic, but could do so by (authorized) request.

7. *Multicast*: Only source-specific multicast [3] is supported, as the traditional multicast model [9] is security wise beyond repair. With the proposed addressing architecture only server addresses can send multicast, and only client addresses can receive it.

2.2 New Internet routing schemes

2.2.1 ROFL: Routing on Flat Labels

ROFL [4] explores a totally flat addressing and routing architecture for an Internet-scale network. The routing architecture borrows heavily from Virtual Ring Routing and other DHT technologies previously employed on top of an internetworking architecture as overlays. ROFL applies these technologies to a network without an underlying IP internetworking layer.

ROFL suffers from considerably higher *stretch* than the current BGP policy constrained Internet, and can remedy that only with huge amounts of memory for caching routes and increasing the number of *fingers* each *virtual node* needs to maintain. ROFL concludes that while it really does not scale, the idea of flat addressing should not be dismissed as impossible.

2.2.2 WRAP: Wide-Area Relay Addressing Protocol

WRAP is a Loose Source Record and Route (LSRR) [2] IP encapsulated overlay forwarding scheme developed in the TRIAD project [12]. WRAP defines a shim header between IP and transport layers. This header contains the source route, i.e. list of IPv4 addresses of WRAP gateways to traverse. At each hop between successive WRAP gateways, the IP header contains the address of the previous WRAP gateway as the source address, and the IP address of the next WRAP gateway as the destination address.

WRAP gateways should be placed in domain border gateways, thus allowing domain-level loose source routing. The source edge of the network is responsible for computing a

set of alternative paths through the network, and adding and filling the WRAP header to all packets being transmitted. The path computation requires listening to BGP messages and maintaining a complete map of the Internet. This gives the prefixes of each domain in the network, but how this is translated to specific IPv4 addresses of each WRAP gateway is not specified.

2.2.3 NIRA: A New Inter-Domain Routing Architecture

NIRA [25] gives the user the ability to choose a provider/domain level end-to-end path. This is argued on the end-to-end rationale [20]: Only users know whether a path works or not. Here the user can be a software agent on the end user's computer, or a network device on the edge of the end user's network (e.g. NAT/FW) could select the paths on behalf of the end user. The argumentation for the user choice of the path builds on fostering ISP competition as a tool for encouraging ISPs to offer better Internet services [7]. Moreover the authors cite evidence [21] that usually the BGP offered default paths are not the best ones possible, as alternative paths offer lower loss rates.

Authors reason that any new Internet architecture needs to take the business relationships between ISPs into consideration. The bilateral relationships (*transit*, *peering*) between ISPs effectively prune the physical network topology, restricting the path choices available for end-to-end traffic [22]. They then use this policy-constrained topology as the one that needs to be covered by the routing architecture.

The policy-constrained Internet topology leads to *valley-free* domain level routes.[3] For each sender and receiver, the network is divided into three regions:

1. The core region covers the tier-1 ISPs (who do not buy transit service from any other ISPs),
2. The uphill region covering all possible paths from the sender up to the core in the sender's up-graph, and
3. The downhill region covering all possible paths from the core down to the receiver.

Distinct routing solutions can then be employed in each of the three regions.

NIRA design includes the Topology Information Propagation Protocol (TIPP) for maintaining the topology of each up-graph in the network. TIPP establishes the uphill, downhill, and peering routing tables within the up-graph, employs path-vector mechanism to build and distribute address information to the users, and utilizes a link-state mechanism to distribute topology information (link status etc.).

TIPP nicely scopes the topology maintenance to distinct edge regions on the network – there is no global topology management for the whole network. The Core routing is managed with BGP like today.

A sender can select the uphill route based on the path state information maintained with TIPP, but for the downhill path selection a Name-to-Route Lookup Service (NRLS) is needed. Finally, the end-points can negotiate route selection between themselves after their initial contact.

The TIPP-built path vectors are represented as IPv6 addresses, where the first 16 bits are reserved for the identification of the Core ISPs,[4] and the last 32 bits are reserved for intra-domain addressing. The 80 bits in the middle are used to encode the domain level path vectors through the up-graphs. Peering links will be represented like "local cores", i.e. they are also encoded into the first 16 bits of an address.

The uphill route is encoded in the source address, and the downhill route in the destination address. Forwarding takes this into account, so that uphill forwarding is done based on the source address, and the destination address is used for the core and the downhill regions.

The evaluation of NIRA is based on inferred Internet topology (~20000 domains, ~80000 links, ~200 tier-1 providers). 90% of domains will have less than 20 prefixes, 30 link records, and 100 forwarding entries. Even the worst cases stay within a couple of thousand forwarding entries. What is remarkable for NIRA is that in each region (uphill, core, downhill) only prefixes assigned internally within the region are present in the routing tables. In this sense NIRA takes the provider-based addressing model to its ultimate limit. Also other performance numbers for NIRA fare admirably (TIPP convergence time, message overhead, and NRLS lookup latency) to the state-of-the-art.

NIRA follows TRIAD/WRAP [2] to use the path-based addressing scheme. The concept of valley-free domain level routes if from Gao [11]. NIRA leverages the route monitoring algorithm from "Feedback based routing" proposal for rapid route fail-over. In border-line cases NIRA is similar to SHIM6 [14]. NIRA source address based routing in the up-graph is somewhat similar to the IPv6 Provider Based Autoconfiguration [23].

2.2.4 Routing as a Service

In [17], Lakshminarayanan et al. propose to solve the internet routing *tussle* [7] between the end users and ISPs by outsourcing route computation to 3rd party *Routing Service Providers* (RSPs). RSPs would buy *virtual links* from the current ISPs, and have *virtual routers* interconnecting them. Customers would interface with the service via RSP gateways, which could exist as software on a customer laptop, or as a network device in the edge of a corporate network, or deeper in the infrastructure.

RSPs would be able to offer specialized routing service that current ISPs cannot offer (other than within their own domains, if at all). These include *avoiding certain ASes for policy reasons*, blocking unwanted traffic (potentially closer to the sources, see [15]), and QoS routing.

The case for RSPs rests on the assumed demand for these specialized services that could not be provided by the default routes computed by BGP. Also, some of the proposed services can be provided with other means. For example your ISP could offer a service for blocking traffic you don't want without any inter-domain coordination, if the normal blocking at your own firewall is not flexible enough.

[3] [11] defines *Valley-Free* as: "After traversing a provider-to-customer or peer-to-peer edge, the AS path cannot traverse a customer-to-provider or peer-to-peer edge."

[4] E.g. via their AS numbers.

Also, the paper does not show how the RSPs would interoperate when the sender and the receiver subscribe to different RSPs, or how the RSPs could start offering services without first buying service from all current ISPs.

2.2.5 The Case for Separating Routing from Routers

In [10] Feamster et al. propose separating routing from individual routers. Here routing consists of routing protocol exchanges within and between ASes and route computation for the AS and individual routers. The proposed Routing Control Platform (RCP) can be deployed in step with individual ASes benefiting from internal deployment in form of easier management of routing policies. RCP can interface with existing routers through the BGP4 route reflection mechanism.

The paper proposes three architectural design principles for routing: 1. Compute routing using consistent state (domain level view), 2. Minimize unexpected or unwanted routing protocol interactions, and 3. Support flexible, expressive policies directly at the inter-domain routing level.

The proposed separation would have numerous benefits. Currently routing policies are implemented via intricate tuning of BGP routing parameters of individual routers, which requires AS-level management systems to ensure that the per-router configurations build up to a coherent whole on the AS-level. RCP would enable network-wide path selection and policy.

2.2.6 Don't Secure Routing Protocols, Secure Data Delivery

In [24], Wendlandt et al. present *Availability Centric Routing* (ACR) dramatically increasing the robustness of inter-domain routing against both BGP control- and data-plane attacks. While the authors do not explicitly mention it, they use the end-to-end argument [20] to conclude that data integrity and confidentiality cannot be provided by secured routing protocol, and that *availability* remains the only security property to be provided for by the routing system.

Availability Centric Routing builds on Availability Providers offering the end systems multiple paths to any destination they might communicate with. End systems then monitor the end-to-end integrity and performance of the chosen path, changing paths whenever there is a problem with the current one. End systems can also split their traffic over the multiple paths for better performance. These functions could be provided for the actual end system by an edge device or the Availability Provider the source is using.

Sources use IP encapsulation over different paths to forward traffic through the *deflection points* in the Availability Provider's network. The encapsulation header includes a *deflection forwarding identifier* for directed forwarding using an alternate forwarding table to the normal BGP. The alternate table is populated with /24 prefixes to avoid BGP attacks on more specific subnets.

Authors report on simulation results showing that a single tier-1 Availability Provider would dramatically increase the reachability of a destination under a spoofing attack. Local ingress filtering for advertised prefixes further improves the performance to 100% success rate, while both single-path BGP and intelligent multi-homing both fail under this kind of attack.

Finally, the authors present incentive argumentation making the model attractive for deployment a tier-1 operator, as the deflection service would be cheaper to produce than normal transit service (no new physical infrastructure), and the deflection service would draw in additional traffic and thus more revenue.

3 Key inferences from state-of-the-art

Several key inferences to make from the presented prior work include:

1. The notion of an administrative domain is the defining factor for the inter-domain topology of the Internet. The contractual relationships between domains (the inter-domain policies) define the internet topology, not the underlying physical connectivity [11], [22].

2. The policy view on the network leads to the valley-free routing model, which can be utilized to localize and thus scale the routing infrastructure. Tier-1 providers form the Core of the network.

3. The concept of the up-graph, this is the policy determined network from a customer to the default free zone (Tier-1 core of the Internet). This is likely candidate for an invariant for any Future Internet architecture. You must follow the money!

4. The division between inter- and intra-domain routing cases is crucial for scalable and resilient routing. The routing interface between domains should be based on a domain-level view, not just router-to-router view. This enables routing policies to be treated at the level where they are defined (between domains).

5. Routing complexity for the network can be traded off with additional complexity in specific points in the path. Connection state naturally resides at the end-points, but increasingly also in the end-domain edge devices (NAT boxes, stateful firewalls). User's endpoint might not be trustworthy, and they might also not be willing to choose inter-domain paths. Path fail-over control could also be located in the core edges so that routing could be managed in the three segments of the valley-free routing model: the uphill, the core and the downhill.

6. Domain-level path would enable radically more scalable network, solving the looming Internet routing scalability issues for good. The usage of domain level paths would also yield a more secure Internet [13]. Many potential encodings for the path in packets exist (IPv6 addresses (NIRA), loose source routing model (WRAP)), but also new label switching schemes are possible.

7. Totally flat routing will likely scale to intra-domain scope. Subnets are already flat, and layer 2 switching and routing is expanding in scope [19]. Any intra-domain aspects of routing or internal network structure should not be visible at the inter-domain interface. In

the long run there may be no need for IP routing in intra-domain. Future Internet Protocol may become really an Inter-Domain Protocol only!

8. Evolvability: Universal access enables evolving the architecture by implementing new architectures as overlays on top of the old one. Like the Internet was an overlay on top of the phone system, any new architecture should run (also) as an overlay on top of the current Internet, and should in turn enable further overlays to be built on top of it. For evolvability the packet is an invariant, but not necessarily with the unicast datagram model (which is policy! A specific use of the packet mechanism).

9. Separate name-to-path mapping and routing solutions (i.e. rendezvous) are possible. The routing system need not implement them, but must enable bootstrapping these systems on top of itself. These systems must also follow the inter-domain policies, as the peering/transit relationships define the possible paths through the network. The rendezvous phase could also create state that may be needed anywhere in the network for the data communication phase. The underlying topology/routing layer would become simpler, as major policy constraints are already taken care of at the rendezvous level. However, the forwarding plane will need to enforce domain-level paths to take only policy-enabled next-hop domains as packets arrive to the domain.

Next we apply these findings to a publish/subscribe internetworking design.

4 Towards publish/subscribe internetworking

Based on our findings above, we will feel our way to the publish/subscribe internetworking architecture at the inter-domain level.

4.1 Assumptions and definitions

We will utilize domain-level paths for forwarding through the network, trying to realize the promise of highly scalable inter-domain routing structure, flexible policy control, and protection against Denial-of-Service attacks. Furthermore, we assume the endpoints in the network will not be named at all (see section 1), so that when we have reached the end of the domain-level path, the publication identity within the packet will be used to determine which nodes the packet should be delivered to.

The main challenges we will now tackle are:

1. *How to apply domain-level paths in a publish/subscribe internetworking architecture?*

2. *How does the network bootstrap?*

We model each domain as an entity, a singular participant in the overall network. The internal structure of the domain is going to be referred to only when absolutely necessary. Furthermore, we assume that each domain knows its boundaries, i.e. its inter-domain links, and the inter-domain policy applied over these links. This is important, since the peering and transit relationships dictate how the traffic can be forwarded through the domains. While the process of getting from a name to the inter-domain path is out of the scope of this paper, we assume that any such rendezvous functionality will also honor these same policies, so that we will not try to use paths that will not be allowed by the policy.

The domain-level paths are lists (or stacks) of *labels*. The labels are identifiers assigned by a domain to its neighboring domains. This means that the global identifiers of the domains will not be used as forwarding labels, so that trying to use a domain label stack out of context will be futile. However, domain-level multi-homing requires that each domain uniquely identify itself over each inter-domain link, so that paths over multiple links can be seen to lead to the same neighboring domain.

In our publish/subscribe forwarding model the *publication identifiers* will be mapped to *local labels* that may then be used to forward the packet, after the domain-level path is exhausted.

4.2 Bootstrapping the inter-domain forwarding state

In the following we concentrate on the routing related signaling exchange between network domains.

1. To start the bootstrap process, the domain A issues a *forwarding subscription* over its inter-domain links. The subscription is identified with A's self-generated cryptographic domain identifier ID_A.

2. If the domain receiving the subscription agrees to start forwarding traffic over the inter-domain link, it chooses a forwarding label fl_A, unique in the scope of the subscription. The domain(s) in the scope of the subscription[5] will store the mapping $< ID_A, fl_A >$. This relation need not be told to the domain A itself, as the domain who assigned the label will remove the forwarding label after the forwarding decision is made, but before scheduling packets to the specific inter-domain link.

3. The domains in scope of the subscription will build internal (intra-domain) forwarding state for the label fl_A, establishing shortest routes to the domain A via all the inter-domain links A has issued the subscription. This state will form a concast delivery tree towards the domain A. This delivery tree forms an elementary trunk for potentially all traffic from the neighbor to the domain A. Additional trunks could be created, potentially spanning multiple domains, but for bootstrap purposes these elementary concast trunks are sufficient.

Done by all domains over all inter-domain links this process will lead to all domains having established forwarding state in their neighbor domains that can carry traffic towards themselves.

[5] For the purposes of bootstrap the scope of the immediate neighbor domain is sufficient.

4.3 Using forwarding state for inter-domain rendezvous

Next we will go through the steps that need to be taken to use the above established forwarding state for connecting the inter-domain rendezvous functionality that provides the name to inter-domain path mapping function.

1. The inter-domain rendezvous functionality of all domains (R_x) will need to subscribe to a well known, globally unique publication identifier, say ID_R. This subscription will lead to establishment of intra-domain forwarding state with a local intra-domain forwarding label (fl_{Rx}).

2. R_x monitors the arrival of new neighbors (the process in the subsection above). When the domain A becomes available, R_x speculatively issues a publication with ID stack (ID_A, ID_R) (limiting the scope of the publication to just the neighbor A).

3. The first identifier (ID_A) is mapped locally to the forwarding label fl_A, and the publication packet is delivered to the neighboring domain over the domain specific delivery tree established before. fl_A is removed from the publication packet before crossing the inter-domain link.

4. When arriving to the domain A only the publication identifier ID_R is left in the packet. Assuming that A implements corresponding rendezvous functionality, the mapping from ID_R to a local forwarding label fl_{RA} exists. ID_R is then replaced with fl_{RA} and the publication is delivered over the shortest path to the closest node subscribing to R_A.

5. When R_A receives the publication, it expects to find the publication ID stack (ID_x, ID_R) within the payload of the publication, and can thus finish a handshake with R_x.

Done between the rendezvous functionalities between all neighboring domains in the internetwork, the global rendezvous network becomes connected. Note that alternative rendezvous systems can co-exist, each using a different globally unique publication identifier. For universal access at least one "global scope" rendezvous system is needed. Some rendezvous systems may operate in limited scopes, and some may establish connectivity utilizing other rendezvous systems (e.g. personal overlay networks).

4.4 Wrapping it together

With these constructs arbitrary domain level paths can be established by the inter-domain rendezvous functionalities performing domain level routing on publication identifiers. The domain-level path is recorded as the rendezvous function forwards the subscription request.

Depending on the publication, when the domain-level label path ends, the publication ID may map to an intra-domain forwarding label that causes the domain to prepend new domain-level label paths for the remaining hops. The domain-level label path construct can thus be seen as an optimization by which publication specific forwarding state is established only in places where needed, e.g. for branching a multicast tree, or for performing some traffic control or transformation. Without this optimization all nodes in the global path will need to manage publication-specific state. For large number of publications with small number of subscribers dispersed globally (the long tail) this would lead to intractable scaling issues.

Within the scope of the default free zone (Tier-1 ISPs) the peering relationships will lead to all Tier-1 ISPs having domain specific concast delivery trees to each other. Thus the domain-level route spans the DFZ with just one label: the label of the Tier-1 provider of the subscriber.

Finally, it seems advisable to isolate the ultimate subscribers from the domain level paths at the network edge so that they could be served with automatic repair of failed domain-level paths.

5 Conclusion

In this paper we have covered a number of research proposals for internet architecture renewal, addressing the various problems facing the Internet as we know it. From this body of research we have distilled nine points to be taken into consideration for any new Internet architecture. Finally, we have applied these points in a radically new Content-Centric Networking design and shown how it is possible to bootstrap the inter-domain forwarding functionality in such network. The rendezvous functionality, while a crucial piece of the total architecture, has been left out of scope for this paper.

6 Acknowledgments

This work has been made possible by the architecture discussions in the PSIRP project, as well as by the group at the TKK Research Seminar on Future Internetworking, led by Mikko Särelä. The presented design is a result of a brainstorming meeting with Janne Tuononen.

References

[1] D. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Holding the Internet Accountable. In *HotNets-VI*, pages 7–12, November 2007.

[2] K. Argyraki and D. R. Cheriton. Loose source routing as a mechanism for traffic policies. In *FDNA '04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pages 57–64, New York, NY, USA, 2004. ACM.

[3] S. Bhattacharyya (editor). An overview of source-specific multicast (SSM). RFC 3569, The Internet Engineering Task Force, July 2003. `http://ietf.org/rfc/rfc3569.txt`.

[4] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker. ROFL: Routing

on Flat Labels. In *ACM SIGCOMM*, pages 363–374, September 2006.

[5] D. Clark, R. Braden, A. Falk, and V. Pingali. FARA: Reorganizing the Addressing Architecture. In *ACM SIGCOMM*, pages 313–321, August 2003.

[6] D. Clark, K. Sollins, J. Wroclawski, and T. Faber. Addressing Reality: An Architectural Response to Real-World Demands on the Evolving Internet. In *ACM SIGCOMM*, pages 247–257, August 2003.

[7] D. Clark, J. Wroclawski, K. Sollins, and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. *Networking, IEEE/ACM Transactions on*, 13(3):462–475, June 2005.

[8] J. Day. *Patterns in Network Architecture: A Return to Fundamentals*. Prentice-Hall, 2008.

[9] S. Deering. Host extensions for IP multicasting. RFC 1112, The Internet Engineering Task Force, Aug. 1989. `http://ietf.org/rfc/rfc1112.txt`.

[10] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. v. d. Merwe. The case for separating routing from routers. In *FDNA '04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pages 5–12, New York, NY, USA, 2004. ACM.

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

[12] M. Gritter and D. R. Cheriton. TRIAD: A New Next-Generation Internet Architecture. `http://www-dsg.stanford.edu/triad`, January 2000.

[13] M. Handley and A. Greenhalgh. Steps Towards a DoS-resistant Internet Architecture. In *FDNA '04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pages 49–56, New York, NY, USA, 2004. ACM.

[14] G. Huston. Architectural Commentary on Site Multi-homing using a Level 3 Shim. Internet-draft, expired, The Internet Engineering Task Force, July 2005. `http://tools.ietf.org/draft/draft-ietf-shim6-arch/draft-ietf-shim6-arch-00.txt`.

[15] J. Ioannidis and S. M. Bellovin. Implementing Pushback: Router-Based Defense Against DDoS Attacks. In *Proceedings of Network and Distributed System Security Symposium, San Diego, C alifornia*. The Internet Society, February 2002.

[16] V. Jacobson. A New Way to look at Networking. Talk at Google, available online at `http://video.google.com/videoplay?docid=-6972678839686672840`, 2006.

[17] K. K. Lakshminarayanan, I. Stoica, S. Shenker, and J. Rexford. Routing as a Service. Technical report, UC Berkeley, 2006.

[18] PARC Incorporated. CONTENT-CENTRIC NETWORKING: PARC's Strategy for Pioneering a Self-Organizing Network That Meets Information Needs. `http://www.parc.com/research/projects/networking/contentcentric/mediabackgrounder.html`, 2006.

[19] R. Perlman. Rbridges: Transparent Routing. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2:1211–1218 vol.2, 7-11 March 2004.

[20] J. Saltzer, D. Reed, and D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4):277–288, November 1984.

[21] 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.

[22] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin. The impact of routing policy on Internet paths. *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2:736–742 vol.2, 2001.

[23] H. Waris. IPv6 Provider Based Autoconfiguration. Master's thesis, Helsinki University of Technology, October 2001.

[24] D. Wendlandt, I. Avramopoulos, D. Andersen, and J. Rexford. Don't Secure Routing Protocol, Secure Data Delivery. In *HotNets-V*, pages 7–12, 2006.

[25] 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.