Design Considerations for a Network of Information

(position paper)

Bengt Ahlgren* Matteo D'Ambrosio† Christian Dannewitz‡
Marco Marchisio† Ian Marsh* Börje Ohlman§
Kostas Pentikousis¶ René Rembarz§
Ove Strandberg‖ Vinicio Vercellone†

ABSTRACT

The existing Internet ecosystem is a result of decades of evolution. It has managed to scale well beyond the original aspirations. Evolution, though, highlighted a certain degree of inadequacies that is well documented. In this position paper we present the design considerations for a re-architected global networking architecture which delivers dissemination and non-dissemination objects only to consenting recipients, reducing unwanted traffic, linking information producers with consumers independently of the hosts involved, and connects the digital with the physical world. We consider issues ranging from the proposed object identifier/locator split to security and trust as we transition towards a Network of Information and relate our work with the emerging paradigm of publish/subscribe architectures. We introduce the fundamental components of a Network of Information, i.e., name resolution, routing, storage, and search, and close this paper with a discussion about future work.

Categories and Subject Descriptors

C.2.1 [**Computer-Communications Networks**]: Network Architecture and Design

Keywords

Network of Information, information-centric networking

*SICS, bengta@sics.se, ianm@sics.se
†Telecom Italia, matteo.dambrosio@telecomitalia.it, marco.marchisio@telecomitalia.it, vinicio.vercellone@telecomitalia.it
‡University of Paderborn, cdannewitz@upb.de
§Ericsson Research, borje.ohlman@ericsson.com, rene.rembarz@ericsson.com
¶VTT, kostas.pentikousis@vtt.fi
‖Nokia Siemens Networks, ove.strandberg@nsn.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ReArch'08, December 9, 2008, Madrid, SPAIN
Copyright 2008 ACM 978-1-60558-234-4/08/0012 ...$5.00.

1. INTRODUCTION

Jacobson *et al.* [9] argue for the need to transition into the third generation of networking. The first generation dealt with connecting wires and laying down infrastructure. The second one placed end nodes, instead of the interconnecting points, at the forefront, leading to the emergence of the WWW and widespread Internet adoption. The third generation, which we call *Network of Information* (NetInf), will refocus the point of attention to what humans care the most about: information.

We recently presented [6] a set of scenarios highlighting the inadequacy of the current host-centric approach and the conceptual advantages of the *information-centric* NetInf approach. In short, by taking information *per se* as the starting point, it will be possible to design a communication infrastructure which is much better adapted to the task of distributing and exchanging information compared to today's host-centric approach.

While NetInf's major advantage is in large scale information dissemination, its design also accommodates non-dissemination applications, including interpersonal communications; inherently supports mobile and multiaccess devices, capitalizing on their own resources (for instance, storage) to deliver higher levels of information availability; and links the physical and digital worlds.

NetInf extends the concept of identifier/locator split with another level of indirection and possibility for recursive lookups in order to decouple objects from their storage location(s). As hosts take a secondary role and information ascends into center stage, objects have to become self-certifiable, unlike today's Internet where information is assumed to be valid because the sender appears legitimate. In NetInf, the host plays a lesser role as users can focus on objects instead of having to focus on their locations, as is done today, e.g., with URLs. NetInf addresses current problems such as unwanted traffic, denial of service, and intermittent connectivity. A central research issue to conclude is whether or not IP needs to be replaced. It is important to point out that NetInf is not an application-layer overlay and that it is in a unique position to use other technologies ranging from virtualization to network coding to in-network manage-

ment, also developed within the EU-funded project 4WARD (www.4ward-project.eu).

Next, we introduce the issues we are concerned with, discussing the pros and cons of different solutions. Then, in Section 3, we initiate the discourse on the necessary NetInf architectural elements. Finally, Section 4 summarizes our contribution and outlines future work.

2. INFORMATION MODELLING

NetInf elevates information to first-class network citizens, in the form of so-called *Information Objects* (IOs). We concur with Jacobson *et al.* [9], that an important incarnation of an IO is A/V content, Web pages, and email. However, to truly revolutionize networking, we argue that the scope should be broadened to include streaming and real-time services, (video-)telephony, and the virtual representation of physical objects.

The goal of an information-centric network is to make all available information in the network easily accessible to the user. For the remainder of this paper, we refer to the actual bit-patterns that contain the information as *Data Objects* (*DO*s, also referred to as Bit-level Objects). These are usually files, streams, or other representations of data in a specific format, e.g., an mp3 file with a certain encoding. Such Data Objects can further be divided into *chunks*, i.e., smaller pieces, to support download features like swarming.

In many cases, however, a user is not really interested in a specific Data Object but in the information a Data Object represents. For example, a user might be interested in a certain song (e.g., Beethoven's 9th symphony), but does not care about the encoding (mp3 with 128 kbps or wma with 196 kbps). These higher level semantics are expressed through the actual *Information Objects*. An IO may, e.g., refer to a certain song without specifying the concrete encoding or the performing orchestra. IOs enable users to find content independently of its specific representation and independently of certain characteristics that *might* not be relevant to the user. IOs can be composed of other IOs or can directly point to one or multiple Data Objects that contain the content itself.

Metadata enables us to further express the semantic meaning of Information Objects, e.g., describing its content or its relation to other objects. Existing research in this field provides an excellent starting point for integrating these features into the network layer, particularly with regard to description languages such as the Resource Description Framework or the ability to automatically establish relations between IOs[1].

With this view of an IO, it is easily possible to accommodate streaming in the NetInf world. E.g., an episode of a TV series could be represented by an IO. With the great degree of freedom the NetInf concept offers, one could bind a trailer of the yet unpublished episode to that IO, and when the episode is published, change the binding to the actual episode. When the binding changes, users can be notified, either explicitly by sending a notification or implicitly by sending the newly bound content. The DOs that contain the video can either be files that are downloaded or actual streaming media. Likewise, a telephone conversation could be represented by an IO, with voice and potentially video streams attached. The IO might also be a "dead end" with no Data Object associated, e.g., in case of a live stream that has not yet started.

Thanks to the very general Information Object model, NetInf is also able to better integrate information access into the user's world by representing real-world objects as Information Objects [5]. Those IOs aggregate Data Objects related to the represented physical entity. For example, an IO could represent the Eiffel tower and could point to related Data Objects like pictures, a wiki page, and a service to buy tickets.

Versioning and Revocation

An additional challenge is to handle dynamics, i.e., the different versions of an IO. An IO like *Today's issue of The New York Times* is frequently changing. A simple solution is to represent each issue by a different DO which the IO can bind to. When a new issue becomes available, the IO can change its binding to the new DO. The old DO can then be rebound to the IO *Yesterday's issue of The New York Times*. Versions can also be an explicit attribute of an IO. For objects that have self-certifying names (e.g., based on a hash over the file), it is not possible to have versions of the objects as each new version, by definition, would get a new name.

Deletion and revocation of NetInf object are challenging issues. To enable deletion of all available "copies" of an object, it would require some central register to keep track of them. In disconnected operation it is not possible to guarantee consistency between partitions. Consistency will also be an issue when partitions rejoin. A more promising way to deal with this is to create an architecture where objects can be invalidated themselves. To implement this type of revocation, objects can be given an attribute that says that before it can be used it needs to be recertified by calling a recertifying function. When using encrypted objects, a similar mechanism could require users to go back to the source and request an encryption key (see also next section). When the object shall no longer be public, certification or decryption keys are no longer issued.

Security Considerations

In current node-centric networks, security is mostly concerned with securing communication between hosts and people. The integrity and authenticity of the actual data transmitted are usually established indirectly by trusting the other party. In an information-centric network, this model is not sufficient. Integrity and authenticity of bit-level Data Objects must instead be provided for the objects themselves, independently of the host delivering the object.

Integrity and authenticity can be directly tied to the names

[1] `http://www.w3.org/RDF/`, `http://www.w3.org/2001/sw/`

of objects with *self-certifying names*, meaning that there is a cryptograhic relation between the name and the object.

Verifying authenticity is harder without consulting a third party. Using (a hash of) a public key as part of the self-certifying name is a common way of providing authenticity. This is the case in HIP, the Host Identity Protocol [14], and in the SFS filesystem [13]. Verification of authenticity, however, requires a handshake with the entity holding the corresponding private key.

Applying the same technique for bit-level Data Objects means that there has to be a principal capable of securely keeping the private key. To enable off-line verification, the principal has to pre-compute a signature which then is attached to the Data Object, as in DONA [10], for example. This solution, however, has the drawback of making revocation more difficult, should the private key be compromised, or the object be updated. Another drawback is that the name of the object has to change if the principal's key has to be changed, or if the object is transferred to a new principal.

A possible alternative design is to let the object name be cryptographically bound to the content with a hash, securing the integrity, but to let authenticity verification be a separate function. This also makes it possible to verify against multiple principals. Pre-computed signatures for this purpose can still be distributed together with the object.

Relation to the Publish/Subscribe Paradigm

The *publish/subscribe* communication paradigm [8] is very attractive as interface or interaction model for a network of information. In this paradigm, receivers indicate their interest in receiving particular *events* by subscribing to those events. Senders independently publish events, which results in the receivers with matching subscriptions getting notified. The paradigm provides decoupling between the communicating parties, the sender and the receiver, both in time and space.

The key part of a publish/subscribe system is the event notification service, which provides subscription management and storage for delivering the event notifications. The mechanism used to match events with subscriptions is crucial for functionality as well as scalability and performance. This mechanism can be compared with information searching, but it is not clear to us that an event notification service can replace a general search service. If we conclude that a general search service is in any case needed, it may be possible to simplify the event notification service and thus overcome the scaling issues.

3. NETINF COMPONENTS

A key functionality of NetInf is to retrieve Data Objects based on their unique identifiers. This process typically includes two main steps, described in the next two subsections. First, *name resolution* locates an object in the network. Then, *routing* forwards (a) the object retrieval query to its storage location(s) and (b) the Data Object from its storage location(s) to the requesting client. How data storage itself can benefit from an information-centric network approach will be discussed in Section 3.3. Preceding the overall information retrieval is a *search*. Advanced search mechanisms can be enabled by the information-centric network approach that complement today's dominating full text search, as described in Section 3.4.

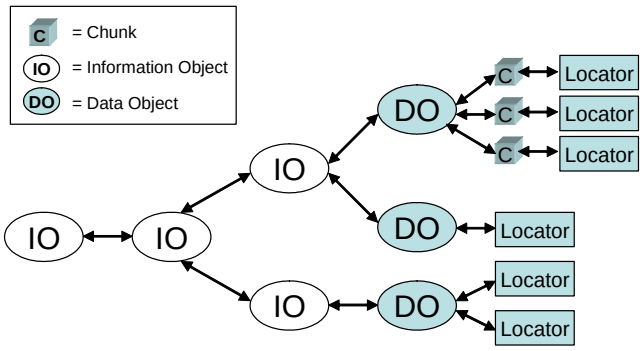

Figure 1: Network of Information Bindings

3.1 Name Resolution

Name resolution (NR) mechanisms resolve IDs into one or more locations. NR should work on a global scale, ensuring correct resolution for any globally available resource, just as the Internet works today. We call this the *Global Resolution Property*. NR should also work in a intermittently connected network if a Data Object is locally available. We call this the *Local Resolution Property*. One way of implementing the Local Resolution Property is to support multiple coexisting NR systems, some of which could have global scope and some could have local scope. In other words, NR systems that can resolve any ID worldwide can naturally coexist with NR systems that deal with a local ID space (e.g., company-internal). This important feature eliminates the need for permanent global connectivity and allows for efficient implementations using anycasting and locality-aware content distribution.

Implementing an identifier/locator split has certain side effects that have to be considered carefully. For example, when a laptop hosting numerous Data Objects changes its network location, all locations of hosted Data Objects change too, leading to a potentially large number of updates in the NR system. Hence, further optimizations are required when applying the identifier/locator split to an information-centric network.

In addition to the identifier/locator binding, the information model described in Section 2 calls for at least three more bindings between IOs, DOs, and chunks that need resolution. In Figure 1, IOs are bound to other IOs and DOs. DOs may be further split into chunks enabling swarm-like dissemination. These binding types have to be represented via 1:n or m:n mappings: DOs and IOs can both be bound to multiple IOs, hence, m:n mappings are required. In contrast,

the DO – chunk and DO – locator bindings are of type 1:n. Chunks are semantic-free and therefore only bound to a single DO, and a DO can be replicated at multiple locations, each of which is unambiguous.

The choice of an appropriate NR mechanism will be heavily influenced by the characteristics of NetInf namespaces. Desirable attributes of future namespaces are *persistence* of names and *contention freeness* with respect to ownership disputes [17]. Those attributes can be met by using flat namespaces. But a flat namespace prevents the use of concepts similar to today's DNS, which is based on a hierarchical architecture and requires, accordingly, a hierarchical namespace. Therefore, a new NR approach will be sought.

For flat names, Distributed Hash Table (DHT) based systems are a promising approach. DHTs are decentralized, highly scalable, and mostly self-organized, limiting the need for administrative entities. There are several compact routing protocols (see [12]), typically used in P2P overlay networks, which can route messages in $O(\log N)$ routing steps, with compact routing tables of $O(\log N)$ states, where N is the number of nodes. $O(\log N)$ resolution steps may, however, result in unacceptably large latencies. Recently proposed promising approaches can guarantee a constant number of hops per lookup [16, 15]. Although convergence time may still be an issue, these approaches can reduce the number of required hops in exchange for larger routing tables and increased overhead in case of churn. Therefore, they are well suited for NR systems with local scope, e.g., within an ISP network. Here, a limited number of "carrier-grade" NR nodes is deployed, i.e., tens or thousands of nodes, that are expected to be highly stable and reliable, with almost no churn.

On a global scope, however, a DHT-based NR system becomes more problematic. Due to the flat namespace and the intrinsically non-cooperative nature of Autonomous Systems (ASs) and other administrative domains, there is an issue with binding placement and control. Scalability and increased churn also need to be taken into account. As we generally believe in the benefits of a flat namespace over a hierarchical one, a possible solution to the dilemma lies in integrating aspects of promising initiatives from the area of routing on "flat" identifiers into the NR system, such as the Late Locator Construction (LLC, see next section). It is one of the key challenges of the NR design to find a satisfactory trade-off between scalability (aggregation based on hierarchical names) and the name persistence offered by the flat namespace.

3.2 Routing

Once the name resolution system has resolved the location-independent identifier into a topologically meaningful locator, the underlying topological routing is used to route all messages among network locations.

In the future, continuing with the current growth, the Internet will have to cope with tens or hundreds of thousands of ASs, millions of prefixes, and billions of hosts. These numbers can still be considered low estimates as the advent of sensor networks, the Internet of Things, and Information-centric Networking can easily lead to much larger numbers of addressable entities.

We are evaluating several options for future routing schemas. A first alternative is to use a traditional topology-based routing scheme, based on shortest path algorithms and hierarchical routing, like the ones used in current Internet (OSPF, ISIS, BGP), or a topological-based compact routing scheme. But recent results in routing research [11] are not encouraging, since logarithmic scaling can not be reached over real networks whose topologies are not static; in fact, network dynamics involves communication costs which cannot grow slower than linearly with the number of nodes and often increase at a very high rate [2].

A second alternative to investigate is to use name based routing which integrate both the resolution path and the retrieval path; this might result in better performance.

Name-based routing combines the name resolution step with the routing step. Name-based routing mechanisms perform the routing of Data Objects based on their identifiers instead of their locations. In a very strict definition, name-based routing mechanisms perform routing by directly mapping the identifiers of Data Objects to a route, without at any point translating the identifier into an address containing topological information. In general, however, most name-based routing mechanisms translate identifiers into addresses at some point, but hide all location information from the transport layer, i.e., address generation and usage become internal to the network layer [1, 10].

We will evaluate name-based routing schemes and related mechanisms like NodeID [4, 3] and LLC [7] to solve the problem arising from global NR systems (cf. Section 3.1). An interesting property of these latter proposals is that they break the routing problem into three, possibly more tractable, parts; routing through an ingress edge domain, through a core network domain and then the egress edge domain. The core domain can reuse traditional routing mechanisms as the large number of new hosts and the churn caused by their movements are isolated in the edge domains. New routing mechanisms for the edge domains can be designed to handle that dynamicity, in a scalable manner, as they do not have to solve the end-to-end routing problem but only have to route to and from the core network.

The approaches for increasing scalability of global IP routing currently being investigated in the IRTF Routing Research Group also divide the problem into core and edge domains. The main goal is to reinforce the ability to aggregate IP routes, and not in general to support churn stemming from host and network mobility.

The *NR system* has to enable a flexible binding between different entities of the information model like Information Objects and Data Objects, and it has to realize the identifier/locator split in a way that enables an intelligent choice

between multiple copies of an object. The *routing mechanism* has to ensure fast data forwarding while keeping the routing table sizes manageable in spite of an ever increasing number of addressable entities. We will evaluate existing approaches as well as new approaches to implement these functions while applying strategies to reduce and minimize all the communication costs: in particular, the message flow for the maintenance of the name resolution database and the routing protocol messages.

3.3 Storage

We recently [6] provided a glimpse of how a NetInf API could look like. At the very least, NetInf nodes will be able to register and retrieve IOs and DOs. The ability to store information in the NetInf infrastructure is fundamental to our design. By providing storage capabilities, NetInf can enhance information dissemination effectively, alleviating the need for concurrent network presence of information producers and consumers, similarly to publish/subscribe systems. Besides allowing for temporal and spatial decoupling, NetInf supports both object storage and caching. Roughly, an object cache can be seen as a type of dynamic short-term memory, while object storage refers to long term memory. Storage will be provided as a reliable service. On the other hand, we are considering scenarios where caching is used opportunistically by the network to enhance performance. Note that in NetInf, caching capabilities are fully integrated in its native operation mode. This is in contrast with current architectures where caching is introduced via separate performance enhancing proxies or application-layer overlays.

The NetInf distributed storage/caching system can be implemented following two different models which can coexist: In the *network-based storage model*, storage resources are provided in the network infrastructure. For instance, storage units may be integrated with network nodes. Perhaps in addition, dedicated storage servers can be deployed in the network according to certain criteria. Among the advantages of this model are its simplicity and the next to negligible churn of storage nodes. In the *network-managed storage model*, network nodes control portions of the storage memory in the attached user equipment. When a user device connects to the provider access network, it makes part of its storage space available to the network. The network can then use such storage partitions as part of the storage system under its control. User-contributed storage can be used, for example, to store DO chunks encoded with erasure codes, as well as complete DOs. Clearly, in this model, there is room for innovations that can manage high churn and intermittent connectivity.

One can discriminate between registering IOs and DOs with the NetInf NR system, binding them with certain locations, and "uploading" (entire) DOs in the NetInf storage system. In practice, an implementation may opt to perform registration and storage in one go. However, we expect that providing two separate API calls will not only be handy for some dissemination applications but, more importantly, will allow for dynamic and non-dissemination content registration. NetInf applications will be able to use some basic storage functions in a manner similar to the way they use TCP/IP sockets today for establishing communication channels. Such functions could allow to store a DO in the network and optionally, for example, i) make it available to any requester, a certain user group, or just to the owner; ii) attach an expiration time; and iii) use erasure code redundancy. Updating or deleting a stored object can be another call (but recall the discussion in Section 2).

3.4 Search

The information-centric approach discussed in this paper calls for new types of search functions. In principle, search functionality can be an integral NetInf component, with its key goal being to return one or more relevant IOs. Perhaps alternatively, external search functionality, closely integrated with NetInf concepts, can also be introduced. We currently envision new types of search functions than go beyond today's state-of-the-art text-box query search. For example, the integration of physical entities in the information model requires new search functionality based on the real-world attributes of a physical entity, e.g., its GPS position, a Radio Frequency Identification (RFID) tag, or a captured image. As mentioned in Section 2, pointing a camera-equipped mobile device at a monument could lead the user to representative IOs. In this case, a new search system identifies the monument on the image and returns relevant IO(s) pointing to its virtual representation. Then, NetInf can return all relevant DOs bound to this IO, such as a Wikipedia article and ticket prices. Results may also be filtered, e.g., based on file type, to further scope the lookup results.

Generally, the information-centric design focuses the attention on the attributes and metadata of each IO. We anticipate that such attributes and metadata will assist in improving search results. Furthermore, the search function can potentially be used to refine or even add new attributes, tags, and metadata to the IO, something beneficial for subsequent searching.

4. SUMMARY AND FUTURE WORK

We have discussed a number of issues and possible design choices related to the concept of networking of information. Naming and addressing is such a key issue. If we look at today's Internet it is clear that DNS's generality, flexibility and ease of use is one of the key reasons for its success. At the same time IP addressing with its too small address space and semantic overload is causing much of the problems we currently struggle with.

We distinguished between *Data Objects*, always encoded in a particular scheme (bit-pattern), and *Information Objects*, i.e., information at a level above particular encodings. We argued that a network of information requires a naming system that supports self-certification, allows for transfer of

ownership, and can handle streaming content. Moreover, we noted that a major challenge is the design of a name resolution system, which, when coupled with a suitable routing strategy, is efficient and scalable in both a local and a global scope.

Within the 4WARD EU project we are designing a NetInf architecture including an information model, local and global name resolution systems, information routing, and models for handling storage in the network. We are building proof-of-concept prototypes, illustrating the benefit of the architecture, showing a serverless web, a personal mobile scenario handling disconnection, and real-world objects integrated into the virtual information world.

5. ACKNOWLEDGMENTS

This work has been carried out in the IST 7th Framework Programme Integrated Project 4WARD, partly funded by the Commission of the European Union. We thank our colleagues in Work Package 6, aptly titled *Network of Information*, for fruitful discussions.

6. REFERENCES

[1] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup. Compact name-independent routing with minimum stretch. In *SPAA '04: Proc. 16th ACM Symp. on Parallelism in algorithms and architectures*, New York, NY, USA, 2004. ACM.

[2] Y. Afek, E. Gafni, and M. Ricklin. Upper and lower bounds for routing schemes in dynamic networks. In *Proc. 30th Symp. on Foundations of Computer Science*, 1989.

[3] B. Ahlgren, J. Arkko, L. Eggert, and J. Rajahalme. A node identity internetworking architecture. In *Proc. 9th IEEE Global Internet Symposium*, Barcelona, Spain, Apr. 28–29, 2006. In conjunction with IEEE Infocom 2006.

[4] Connectivity and dynamic internetworking prototype and evaluation. Deliverable 23, Ambient Networks Phase 2 project, Dec. 2007. FP6-CALL4-027662-AN P2/ D23-E.2.

[5] C. Dannewitz, H. Karl, and D. Warneke. Service platform for real-world / Internet integration in mobile applications. In *Proceedings of the 13. Mobilfunktagung*, Osnabrück, Germany, May 2008.

[6] C. Dannewitz, K. Pentikousis, R. Rembarz, E. Renault, O. Strandberg, and J. Ubillos. Scenarios and research issues for a Network of Information. In *Proc. 4th Int. Mobile Multimedia Communications Conf.*, Oulu, Finland, July 2008.

[7] A. Eriksson and B. Ohlman. Dynamic internetworking based on late locator construction. In *10th IEEE Global Internet Symposium*, May 2007.

[8] P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2):114–131, 2003.

[9] V. Jacobson, M. Mosko, D. Smetters, and J. Garcia-Luna-Aceves. Content-centric networking. Whitepaper, Palo Alto Research Center, Jan. 2007.

[10] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *Proc. ACM SIGCOMM*, Kyoto, Japan, Aug. 2007.

[11] D. Krioukov, k c claffy, K. Fall, and A. Brady. On compact routing for the Internet. *SIGCOMM Comput. Commun. Rev.*, 37(3), 2007.

[12] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Survey and Tutorial*, 7, 2005.

[13] D. Mazières, M. Kaminsky, M. F. Kaashoek, and E. Witchel. Separating key management from file system security. In *Proc. 17th ACM Symposium on Operating Systems Principles (SOSP '99)*, Kiawah Island, SC, USA, Dec. 1999.

[14] R. Moskowitz and P. Nikander. Host identity protocol (HIP) architecture. IETF RFC 4423, May 2006.

[15] A. T. Mýzrak, Y. Cheng, V. Kumar, and S. Savage. Structured superpeers: Leveraging heterogeneity to provide constant-time lookup. In *Proc. 3rd IEEE Workshop on Internet Applications*, Washington, DC, USA, 2003.

[16] V. Ramasubramanian and E. G. Sirer. Beehive: O(1) lookup performance for power-law query distributions in peer-to-peer overlays. In *Proc. Networked System Design and Implementation (NSDI)*, San Francisco, CA, Mar. 2004.

[17] M. Walfish, H. Balakrishnan, and S. Shenker. Untangling the web from DNS. In *NSDI'04: Proc. 1st Symp. on Networked Systems Design and Implementation*, San Francisco, CA, USA, 2004.