A DHT-inspired clean-slate approach for the Next Generation Internet

Oliver Hanka, Christoph Spleiß, Gerald Kunzmann, Jörg Eberspächer
Technische Universität München
Institute for Communication Networks
80333 Munich, Germany
{oliver.hanka, christoph.spleiss, gerald.kunzmann, joerg.eberspaecher}@tum.de

Abstract

The Internet architecture of today is the result of a constant evolution during the past 25 years. However, this layering of add-ons, bug fixes and extensions has grown into a tremendously complex and therefore increasingly static platform. On the contrary, a multitude of new challenges related to issues never conceived in the original design, such as security and mobility have grown more and more important. Hence, we propose a clean-slate approach based on a two-tier locator identifier split. Combined with Distributed Hash-Tables (DHT), we develop a scalable long term-alternative to the current Internet architecture.

1. Introduction

The current Internet has many shortcomings [12, 1, 7, 3] and faces even more challenges when looking into the future. This has led to a re-thinking of the Internet architecture. A number of proposals are addressing some of these issues (e.g. [9, 15, 6, 2, 14]). While many of these approaches manage to address a single, specific issue, they mostly add yet another layer of complexity on top of the current architecture. In this paper, we will discuss a clean-slate approach for a Next Generation Internet architecture. Our contribution binds together available concepts with DHT mechanisms in a novel way to address the major limitations of today's Internet.

In five critical areas new solutions are needed and covered by our approach:

- **Naming and addressing** Todays architecture only allows to address single hosts. We believe a mixture of host and content addressing is the way to go for a Next Generation Internet architecture.
- **Scalability** It is a widely know fact, IPv4 addresses are limited [7] and IPv6 is not established yet. Other problems arise from large routing tables at the Internet core and congestion at network nodes, calling for a different routing/lookup mechanism and ways to globally handle QoS issues. We propose a new addressing scheme to solve this issue.
- **Decentralization** The Internet was designed to decentralize information, control and maintenance. A centralized architecture like the DNS-system contradicts this paradigm. Our concept, therefore, introduces distributed lookup and management mechanisms.
- **Mobility** While mobility is supported by some extensions (like Mobile IP [10]) it is still not possible to roam between different subnets without a connection loss at the transport layer. We address this issue with a locator/identifier split [4].
- **Security** This is probably one of the most critical issues with todays architecture. Only add-ons do exist, but no system wide concept is deployed. Our proposal introduces security mechanisms in lower layers.

We propose a *Distributed Hash Table (DHT)* based concept, as future architectures require efficient and flexible mechanisms for addressing, routing, user-management and data storage. DHTs proofed to provide these characteristics. By introducing this technology as the key element, we are able to address most of the topics mentioned above. The one missing—security—is not directly covered by the DHT. Its integration, however, is simplified by mechanisms enabled by the DHT. Figure 1 summarizes the major advantages we gain from this approach.

Before digging into the methods and reasons of our concept, we state that the customer, alias the user, will benefit from our proposal. With our concept, the user will be able to freely roam between different networks, establish trusted and secure connections and will be able to quickly retrieve the information he is interested in from the network.

We are aware, however, that the concept must be accepted by the providers to become a successor to today's architecture. Providers can benefit from the concept besides the gain generated by a higher customer satisfaction. We introduce mechanisms to implement location awareness for data transfer—which will save costly inter-provider traffic—and enable providers to operate and connect heterogeneous networks to each other without the need for additional address and protocol translation. In addition, our proposal aims to reduce the number of protocols needed for communication. This will lessen complexity on both the customer and the provider end, hence resulting in cheaper

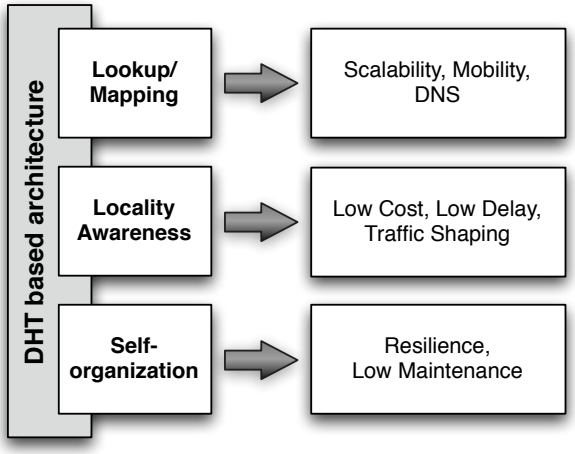

Figure 1. Benefits of the DHT based architecture

and easily manageable implementations. Additionally, existing network structures like *autonomous systems (AS)* can be reused to ease migration.

2. Concept

To be able to achieve the goals set in the introduction, we need to redesign the protocols of layer three and four of the OSI model. Yet, instead of defining new layers, we introduce functionality-blocks, which provide their services to higher layers. Figure 2 illustrates the new layer model. In this way we avoid inflexible layers and the current workaround with cross layer designs.

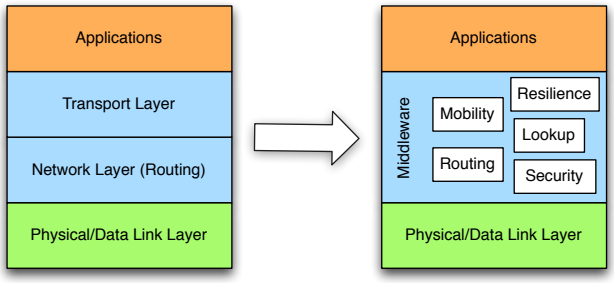

Figure 2. Current OSI-Layer model (left) and NGI layer model (right)

2.1. Locator/Identifier Split

In addition to the DHTs, we found two other design decisions very supportive. The first one is a *Locator/Identifier-Split*. By separating the locator from the identifier we are able to support true mobility. Higher level connections based on a unique and never changing identifier are not prone to interrupt, when the node changes its location within the network. The locator also plays another important role. As suggested by Moskowitz et al. [9], the locator is also used as a public key and hence serving as a fundamental security anchor in the architecture. As long as a node is able to verify its communication partner, it will be able to provide any other security related requirements as privacy and integrity. As a last point, the Locator/ID-Split enables us to introduce content addressing. As many identifiers can point to a single locator, it will not be a difference whether the identifier belongs to an actual node or a document. The identifier only specifies, what can be found in the network, and the locator tells us the current position within. We choose 128 bit for the identifier to support a huge amount of addressable nodes and content.

To further improve scalability, we introduce a hierarchical addressing scheme within the core network. Today's routing tables of core network router are growing at an exponential rate due to variable subnetting and multihoming. By reintroducing a hierarchy in the addressing scheme, the routing tables will shrink dramatically. Yet, we will not loose the benefits of variable subnetting and multihoming. As described above, higher layer connections will be based on flat identifiers and a change of the locator does not play a major role like in today's architecture.

2.2. Locator/Identifier Mapping with DHTs

The DHTs are responsible to map between identifiers and locators. Additionally, they map between domain names and locators, hence rendering the current DNS obsolete. To balance the load, all core network routers participate in a global DHT. These router—most of them currently serving as border gateway router—are interconnected by a one-hop DHT protocol like *D1HT* [8], as they experience very low churn rates. In this way, the lookup delay can be kept to a minimum. A one-hop DHT resolves each routing request within one hop because each node maintains a complete topology table of the DHT. Of course, this means that any topology update must be broadcasted to any participating node. As already stated above, the expected churn rates are very low and hence, not many of those topology updates must be sent.

The global DHT, however, only stores the AS an identifier is currently registered with. More specific, it only stores the address of the gateway routers, the AS can be reached by. A local DHT within the AS is responsible for the actual mapping between the identifier and the locator. If a node wants to communicate with another node, it initiates a request to the local DHT. If the requested node is within the same AS, the request is resolved and the global DHT is not involved in the mapping process. Otherwise, the request is relayed to the global DHT and to the local DHT of the requested node's AS.

By choosing DHTs to handle name resolution and mapping between locators and IDs we to be able to meet our

requirements of a decentralized and stable system. No authority or single person will be able to control major parts of the network. Even manipulating the DHT by an attacker can be prevented [13]. Additionally, reliability will be increased by this approach. Redundant storage within the DHT will keep the lookup system operational even if several ASs become detached from the rest of the network.

3. Evaluation

One critical part of this concept is the load of the core network routers participating in the global DHT. To estimate the load of each core network routers, we take the following assumptions: Based on LRZ traffic statistics [11] (AS 12816 - 1.000qps with about 40.000 nodes) every node starts in average 0.025 queries per second (qps). If we assume 6 billion participating nodes and a core network consisting of 100.000 routers, we get 1.500qps at each core network router without caching.

Jung et al. [5] analyzed the caching behavior of DNS resolvers and experienced a 80 - 86% cache-hit-rate for an AS. If we take the lower value, we get 300qps at each core network routers with caching in the lower hierarchy enabled. In addition to simple host lookups, our system should be able to also resolve content-identifiers. We, therefore, increase the expected query rate by a factor of 10.

Finally, we analyze the data rate at each core network router the queries will generate. For simplification, we assume each address field has the size of 16byte. Each query-packet contains 4 fields (rx/tx identifier and rx/tx locator) some flags and some bytes for the packet-header. This generously sums up to 200bytes per packet. The query itself causes 4 packets to be transmitted. First, the query itself; then the query forwarded into the DHT; the reply from the DHT and finally the response to the query. Therefor, we have the following equation:

$$200byte * 3000qps * 4 = 2,4Mbyte/s$$

As a result, each core network router needs to handle a maximum data rate of 20MBit/s for the host- and content-addressing lookup system. A modern hardware-architecture should easily be able to handle this load.

4. Conclusion

We introduced a new DHT based concept for a Next Generation Internet architecture. The proposal solves several problematic issues of todays' Internet relating mobility, security, address shortage and routing table growth.

We believe both, customers and providers, will benefit from this concept. Users will gain more flexibility and added functionality while ISPs will be able to generate new business models. Particularly, costly inter-AS traffic, caused, e.q. by peer-to-peer file transfer, can be limited.

Our next step is to run supporting simulations. Currently we are implementing the major building blocks of the architecture to be able to run simulations and to test it in an experimental testbed. This work has been supported by the German Federal Ministry for Education and Research (BMBF).

References

[1] R. Braden, D. Clark, S. Shenker, and J. Wroclawski. Developing a Next-Generation Internet Architecture. White paper, DARPA, July 2000.

[2] D. Farinacci, V. Fuller, D. Oran, and D. Meyer. Locator/ID Seperation Protocol (LISP). Internet draft 08, IETF, July 2008.

[3] A. Feldmann. Internet clean-slate design: what and why? *SIGCOMM Comput. Commun. Rev.*, 37(3):59–64, 2007.

[4] L. Iannone and O. Bonaventure. On the cost of caching locator/ID mappings. In *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, New York, NY, USA, 2007. ACM.

[5] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. Dns performance and the effectiveness of caching. *IEEE/ACM Trans. Netw.*, 10(5):589–603, 2002.

[6] 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 *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 181–192, New York, NY, USA, 2007. ACM.

[7] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang. IPv4 address allocation and the BGP routing table evolution. *SIGCOMM Comput. Commun. Rev.*, 35(1):71–80, 2005.

[8] L. R. Monnerat and C. L. Amorim. D1HT: A Distributed One Hop Hash Table. In *Proceedings 20th IEEE International Parallel and Distributed Processing Symposium*, page 21. IEEE, June 2006.

[9] R. Moskowitz and P. Nikander. Host Identity Protocol. RFC 4423, IETF, May 2006.

[10] C. Perkins. IP Mobility Support for IPv4. RFC 3775, IETF, August 2002.

[11] L. Rechenzentrum. DNS Statistiken - http://dnsstat.lrz-muenchen.de/cgi-bin/dns.cgi. Webpage - 11/05/2008.

[12] M. Siekkinen, V. Goebel, T. Plagemann, K.-A. Skevik, M. Banfield, and I. Brusic. Beyond the Future Internet–Requirements of Autonomic Networking Architectures to Address Long Term Future Networking Challenges. In *11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'07)*, pages 89–98. IEEE, 2007.

[13] E. Sit, R. Morris, P. Druschel, F. Kaashoek, and A. Rowstron. *Security Considerations for Peer-to-Peer Distributed Hash Tables*, volume 2429, pages 261–269. Springer-Verlag Berlin, Heidelberg, 2002.

[14] X. Yang. NIRA: a new Internet routing architecture. *SIGCOMM Comput. Commun. Rev.*, 33(4):301–312, 2003.

[15] X. Zhang, P. Francis, J. Wang, and K. Yoshida. Scaling IP Routing with the Core Router-Integrated Overlay. In *ICNP '06: Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*, pages 147–156, Washington, DC, USA, November 2006. IEEE Computer Society.