

Identifying Hosts in Home Networks

Anu Markkola
Helsinki University of Technology
anu.markkola@tkk.fi

Abstract

The current Internet architecture uses IP addresses and transport layer port numbers for identifying connections between hosts. This model requires changes because IP addresses are ephemeral and rarely unique today. This is caused by mobility, NAT (Network Address Translation) and dynamic addressing. IP addresses are therefore impractical as transport layer identifiers and should be separated to operate only as locators. Home networks have evolved rapidly and contain desktop computers, mobile devices and privately administered servers. Home user needs have to be considered when designing a solution for the technical problems of the Internet. This paper analyzes the proposed solutions, Host Identity Protocol (HIP), Unmanaged Internet Architecture (UIA) and NUTSS architecture from the home user perspective.

KEYWORDS: Host Identity Protocol, Unmanaged Internet Architecture, NUTSS, security, home networks

1 Introduction

The Internet was established in the sixties and its growth in both size and the range of services has exceeded the expectations of the original designers. The architecture has been extended to answer to new requirements, but conflicting design decisions made along the way make it complex and difficult to manage.

The need for more IPv4 addresses is not the only factor that makes the current architecture inadequate. People use mobile devices for their daily activities and want their devices as well as the public network services to be reachable from anywhere. Peer-to-peer (P2P) applications are popular and users expect them to work between private address realms.

A fundamental problem with the current architecture is that many protocols use IP addresses as transport layer end-point identifiers. Static IP addresses are not suitable for mobile hosts. NAT (Network Address Translation) boxes break end-to-end connectivity which is a problem especially for peer-to-peer applications. The security of the current Internet is largely based on firewalls, which allow or disallow flows based on IP packet header information. It is challenging to enforce a wise security policy based on this information and often legitimate flows get blocked by firewalls. Privacy of communication is important for some users, but end-to-end security is not provided automatically for all network applications.

In order to fulfill the new challenges, the Internet needs

to change. IPv6 would help by providing more addresses and support for mobility and security services. However, it is not going to solve all problems. Dynamic addresses and firewalls remain a problem even when there are enough addresses for everyone. This paper analyzes the work that has been done in order to create a new secure Internet architecture.

Some proposed solutions aim to solve the issues by inserting a new layer to the networking stack. This new layer offers a new namespace for global host identification. These protocols operate on top of the current IP based network layer and offer secure network services to the transport layer. The enhancements provided by solutions such as HIP (Host Identity Protocol)[11] and UIA (Unmanaged Internet architecture)[1] are indisputable, but some argue that the end-to-end security model they provide is still not enough [3]. The claim is that this model of security is over the hill and that the new architecture should include also the network in the middle to participate in providing security. NUTSS (NATs, URIs, Tunnels, SIP and STUNT)[4] architecture implements this end-middle-end model and provides other enhancements using higher level signaling protocols such as SIP (Session Initiation Protocol)[13].

This paper looks at the suggested solutions from the viewpoint of home networks. Challenges of the Internet architecture are viewed in more detail in section 2. Home user concerns are analyzed in section 3. Sections 4, 5 and 6 present HIP, UIA and NUTSS respectively.

2 Challenges of the current Internet architecture

This chapter presents the most important problems in the Internet architecture and defines a set of requirements for the solution.

2.1 Identifying hosts and users

The Internet architecture uses IP addresses to locate and identify hosts in the network. Host names used by humans and higher layer applications are mapped to numeric IP addresses through DNS (Domain Name System). For static IP addresses this works well. However, the addresses are no longer unique because of NATs and private address realms. Neither are they static, since public IP addresses are often assigned randomly with DHCP (Dynamic Host Configuration Protocol). The same address of a dynamically managed network can therefore be assigned to different hosts at different

times. Host mobility leads to use of temporary addresses, because moving hosts need to change their locator information to remain reachable. For all these reasons the usage of IP address as an identifier is no longer feasible and therefore another way of identifying hosts and users is needed.

2.2 Middleboxes

The most important types of middleboxes that interfere with the traffic in the Internet are firewalls and NATs. NAT was originally meant to be only a temporary remedy to IPv4 address shortage. The failure in standardizing the functionality has brought up many different solutions which makes it challenging to solve the issues of NAT usage. NATs work well when a client in a private network connects to a public server in the Internet using TCP or UDP. However, other types of connections such as those used by vastly popular P2P applications require that also the connections initiated from outside of the private realm reach the hosts. Problems of P2P connectivity through NATs are well-known[16] and special NAT traversal techniques such as STUN (Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs))[14] have been developed to overcome the issues of reaching hosts in private networks.

Firewalls provide security by enforcing access control rules for data flows. Data traffic that does not comply with security policy rules is filtered in the network before it reaches the protected network or host. Access control on the traffic is done mostly based on the IP addresses of the connection endpoints, port numbers and protocol type. Ephemeral IP addresses reduce the capability of the firewalls, or rather the policy designers, to determine which traffic is allowed and which is not. Because of this, legitimate connections may get blocked because of an over-conservative firewall policy. Real security policies are hard to formulate as firewall rules because the users cannot normally influence the firewalls, as they can be located anywhere along the path. Neither can the firewalls in normal scenarios communicate their policy requirements to the user.

2.3 Security

End-to-end security in the Internet is mostly provided by higher layer protocols, such as TLS (Transport Layer Security), for certain types of applications. Enterprises use VPN (Virtual Private Network) technologies to ensure the security of their employees' data traffic to the office, but the traffic elsewhere in the Internet is mostly not protected in any way.

Most of the security in the current Internet is provided by the endpoints themselves. Firewalls define an accepted set of protocols and ports but usually do not go beyond that. Security is therefore as strong as are the application programs that handle the data traffic.

DoS attacks are a threat that can not be prevented by the endpoint itself. Therefore it has become evident that the infrastructure has to take a stronger role in preventing the attacks.

3 Home user needs

Devices in home networks often connect to Internet through NAT devices which may or may not be configurable by the users themselves. The effects of NATs need to be considered when designing new protocols for the Internet. Ideally, the users in home networks are able to create connections globally through multiple NATs and mobile users must be provided means to connect to their home network from anywhere. Of course, the mobile users must also be able to use the network they are currently located at, provided that they have authorization for it.

The security needs of home users vary greatly. Some use home network connection for telecommuting. Others perhaps for private matters or entertainment. All users need some level of security but most of them do not have the required knowledge to make sure that their network is safe enough. Users generally do not follow the news about security vulnerabilities of software so it is mostly in the hands of the software providers to deliver the security updates automatically and thus keep the computers of the users up-to-date. Privately administrated servers are vulnerable to various types of attacks. Many can be prevented by administering the servers properly, but DoS attacks, for example, can not be prevented by the servers themselves.

A regular home nowadays has multiple computers and also phones, digital TV-recorders, web servers and other devices that provide services. Having devices and services at home, it is only natural that users want to connect to their devices from outside their home also and even allow their friends to access their home network. As a result, the configuration of a home network and its access control mechanisms can be a tedious task even for users with the required technical skills. Therefore the configuration of the home network and the devices should be easy to manage.

Home users cannot easily be forced to dramatic changes in their software or hardware. Without thorough knowledge of the impact of the changes, the motivation for updating the home computers to support a new kind of architecture can be low. Therefore new solutions should be easy to deploy and they should interoperate with legacy networks, protocols and software.

4 Host Identity Protocol

Host Identity Protocol [11] separates the locator and identifier roles of the IP address and creates a new namespace based on cryptographic Host Identities. The new cryptographic namespace decouples transport and network identifiers by providing statistically global endpoint identifiers. Transport layer protocols use these Host Identifiers (HIs) for identification instead of IP addresses, while the network layer uses IP addresses for routing. A Host Identifier is the public key of an asymmetric key pair. It can be used for strong authentication because it is computationally difficult to forge. Hosts establish HIP connections by initiating a four-way handshake called the base exchange. During the base exchange, two hosts authenticate each other, execute Diffie-Hellman exchange and form two IPsec security associations for each direction. The actual data traffic of the con-

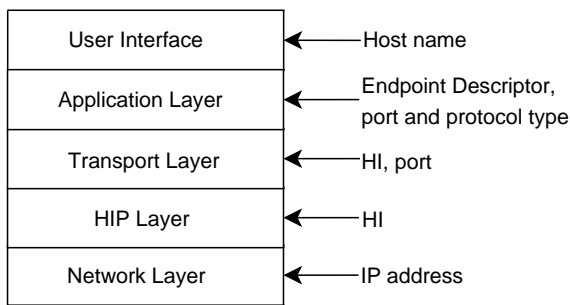


Figure 1: The HIP namespace in the networking stack and identifiers used in different layers

nection is protected with IPsec ESP-tunneling [6]. HIP name layer is added to the networking stack as shown in Fig. 1.

Separating the IP address from the identity works because a locator allows HIP to provide support for host mobility and multihoming [5]. Transport layer uses HIs instead of IP addresses and connections can therefore survive IP address changes. Mobile hosts can remain reachable by using a HIP Rendezvous server [9] which is a globally accessible public server that is used to relay HIP control packets.

Another benefit of using HIs instead of IP addresses is the interoperability of IPv4 and IPv6 applications. HIP hides the actual IP address from the transport layer, which makes it possible to create connections between hosts using applications that support different IP-versions.

HIP protects against DoS attacks with computational puzzles that the client must solve before any state is created at the responder. Hi^3 [12] offers even more protection by using an overlay for routing control messages. The overlay routers can be delegated to send the puzzles on behalf of the server, so that the load of a DoS attack will not fall on the server itself.

HIP NAT traversal [8] is provided with UDP encapsulation and HIP rendezvous servers. When a host discovers the presence of a NAT using STUN, it encapsulates all HIP control messages and data traffic into UDP packets. This allows the packets to traverse NATs. Global connectivity through NATs is provided by rendezvous servers that are located in the public Internet. Servers behind NATs, that wish to be reachable from outside of the private network also, register to a rendezvous server using HIP registration procedure [10]. The rendezvous server forwards HIP control packets to guarantee successful NAT traversal. However, the hosts try to find a direct path for ESP between them using UDP hole punching and resort to triangular routing only when necessary.

Users prefer to use host names when initiating connections and therefore DNS or other name mapping service is used to resolve host names to HIs. HIP can also be used in opportunistic mode if the HI of the other endpoint can not be resolved before connecting. In opportunistic mode the connection initiator sends the first packet of the Base Exchange to the IP address of the recipient with empty HI field. The recipient delivers its HI to the initiator in the second packet of the Base Exchange. Opportunistic mode leaves the connection initiator vulnerable to middleman attacks, but is still

more secure than connection with no protection at all.

4.1 Analysis

HIP offers security enhancements for authentication and data encryption. DoS protection is also an important improvement. HIP provides mainly end-to-end security, but Host Identifiers can also be used at the firewalls in the network to identify communication endpoints. A firewall can intercept the control messages that pass through, which allows it to create a mapping between Host Identity and IP address-SPI (IPsec Security Parameter Index) pair. As a result, IPsec data flows can be filtered based on Host Identity instead of plain IP address. For this to work, however, the control messages containing the information necessary for creating the mapping, and data traffic must use the same path.

Host identifiers are not user-friendly meaning that DNS or other name service must be used to map the identifier. HIP related technical concepts, such as public keys, identities and fingerprints, are not likely to be familiar and are therefore less usable to regular home users. This implies that special attention must be paid to the user interface design of HIP management.

HIP requires changes to the operating system networking stack which may slow down deployment. APIs for using HIP have been developed for both HIP-aware and legacy applications [7]. Most of the legacy applications can operate normally using the HIP legacy API, but some IPv4 applications, FTP for example, might not always operate as expected. This, however, is a relatively small problem when considering the benefits of HIP.

HIP is still work in progress, but the technical details as well as more general impacts of HIP are studied by IETF (Internet Engineering Task Force) HIP working group and HIP research group. Implementations exist but wide-scale deployment has not been achieved yet.

5 Unmanaged Internet Architecture

Unmanaged Internet Architecture (UIA) [1] addresses the identification problems by introducing user-friendly personal names for devices. Users can assign names to their own devices in their own namespace. The names are mapped into globally identifiable Endpoint Identifiers or EIDs which can then be used in routing through an overlay network. UIA provides peer-to-peer connectivity for mobile devices in the user's social neighborhood. This means that devices track the locations of other devices that are often contacted by the user, such as the devices of friends and family members. No central name service or support from the infrastructure is needed to accomplish this.

UIA operation is divided between naming and routing layers. The UIA naming layer is responsible for managing the personal namespace of the user. Users give simple names to their devices, such as *phone* or *laptop*, and register the names and corresponding EID to other devices through UIA introduction procedure. To other users the device is reachable through the owner's name, for example *phone.Alice*. Users can easily share their namespace with their friends so that the friend will have a sub-namespace containing the devices

that the user wants to share. The naming layer maintains the state of the namespace across all devices of the owner and all devices in other namespaces linked to the user's namespace. New namespace entries are delivered to devices with a method that combines gossiping and rumor mongering types of information delivery. In the gossiping phase the device broadcasts all new information about namespace changes to other devices it is connected to at the moment. In rumor mongering phase the device will ask from other devices if there is any new information available.

The UIA routing layer maintains a list of active overlay peers that can be used in the routing phase. These peers should preferably be in the public Internet to make NAT traversal possible, but less stable peers can be used also to support connections in ad hoc environments. Another list of potential peers is also gathered which contains EIDs and location information of all devices that have been introduced to the device directly. Additionally, the potential peer list is updated by exchanging information periodically with devices in social neighborhood. If some of the active overlay peers is unavailable, the overlay peer list can be complemented by searching the list of potential peers.

The overlay network locates the target using the EID provided by the naming layer. First the connection initiator tries direct TCP-connection to the location where the target was during last connection. If this fails, a location request is broadcast to the peers of the overlay network which forward the request if unable to answer. A counter value is updated in the request to determine when to stop forwarding. A device that has a TCP-connection open to the target will respond to the request.

UIA is still at a very experimental level. A prototype implementation exists and some results of performance simulations are presented in [1].

5.1 Analysis

UIA supports legacy NAT-traversal by using stable overlay peers in public Internet for message forwarding. Hole punching will possibly be supported in the future. UIA mobility support is based on introducing a new global identifier the same way as HIP does but UIA identifiers are personal rather than host-specific. Locating a mobile device is different from HIP since no infrastructure is used but instead an overlay network resolves the locations of the devices in close social distance.

UIA is targeted for end-users and it is therefore designed to be as easy to use with little or no configuration effort needed. Personal namespace is convenient for non-technical users and device introduction procedure offers an intuitive way for setting up a network between users' own and their friends' devices.

6 NUTSS

The NUTSS[4][3] (NATs, URIs, Tunnels, SIP and STUNT) architecture is based on separating data and control planes. Users, machines, applications and data flows in the network are identified by URIs. The architecture leaves behind the

end-to-end model of the current architecture. In the architecture the network in the middle authenticates and authorizes the data traffic passing between hosts. NUTSS uses a combination of off-path and on-path signaling procedures to achieve this. NUTSS-aware policy-boxes in the network are used to enforce security policies.

Connection establishment starts with off-path signaling which allows authentication of the endpoints, acquiring authorization from middleboxes and negotiation of connection parameters. The connection endpoints are identified by human-readable URIs. The network addresses needed for data traffic are exchanged during off-path signaling and used only in routing the data traffic which provides support for host mobility and multihoming. Policy-aware boxes that are used during off-path signaling are usually, but do not have to be, located off the data path, and can be physically far away from the connection endpoints. Location-independence of policy-boxes helps in protecting the hosts and the network from DoS attacks. As a result of the off-path signaling the endpoints receive a secure token from the policy boxes which is then used during on-path signaling to prove the authorization of the flow to the middleboxes. In consequence, firewalls, NATs and routers can map the passing data flows to a certain user or application on a specific host.

6.1 Off-path signaling

Off-path signaling uses static URIs for routing the control messages. NUTSS architecture uses a slightly modified version of SIP[13] as the off-path signaling protocol. In the NUTSS architecture each domain in the network that has a policy concerning data traffic, has at least one off-path policy box. The policy box is responsible for the security policy of its domain. Hosts register to the off-path policy box of their domain by providing an endpoint descriptor and the current IP address to the policy box. The descriptor contains the URI, information about the user, connection and service types, software and other optional data. If the domain is not directly connected to the public Internet, the policy box registers the received endpoint descriptor to the policy-box of the parent domain with its own IP address. The chain of registrations will cause the connection invitations targeted at a host to travel through all policy-boxes responsible for domains through which the target host is connected to the Internet.

Off-path signaling messages for connection initiation are routed from the source through the local policy box and from the policy boxes of parent domains to the public Internet. Messages pass across the Internet to the outermost policy box of the target domain which forwards the messages towards the recipient. Finally the policy box responsible for the target subdomain can forward the message to the recipient. In figure 2 Bob is sending an invitation to Alice. The off-path signaling message containing the invitation travels through the policy-box of Bob's ISP through the Internet to the policy-box of the ISP that Alice's domain belongs to. The policy-box of the subdomain can then deliver the invitation to Alice.

Off-path policy boxes, through which the signaling messages pass, can apply access control policies to the signal-

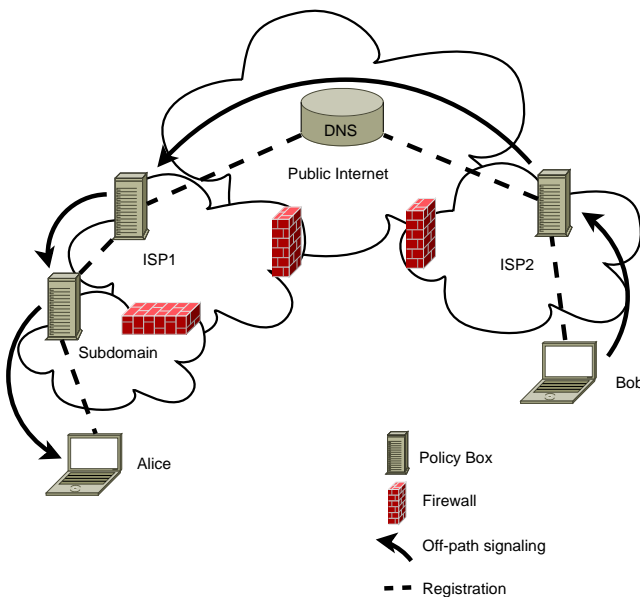


Figure 2: NUTSS Registration and off-path signaling

ing messages and set requirements for connection parameters. Any off-path policy box can also request authentication from the user. Authentication requests and responses follow the same path as other off-path signaling messages. The policy requirements for data traffic and a token for proving the authorization are attached to the passing off-path signaling messages.

6.2 On-path signaling

On-path signaling is based on network addresses and its purpose is to establish the actual data flow between the endpoints. This includes taking the necessary actions for traversing NATs and allowing the NUTSS-aware middleboxes to verify the authorization of the data flow.

6.3 Analysis

NUTSS architecture is still on experimental level but a proof-of-concept implementation is available. NUTSS uses well-known NAT traversal techniques for passing legacy NATs. Connection reversal, STUNT (Simple Traversal of UDP Through NATs and TCP too)[2] and TURN (Traversal Using Relay NAT)[15] relaying are tried during on-path signaling phase in order to traverse the NATs on the path. Public off-path STUN servers are used to discover the external address of a NAT if present.

Mobility in NUTSS is managed with off-path signaling. Hosts use the off-path signaling messages to inform the other communication endpoint about the IP address changes. NUTSS adds an extra session layer to the networking stack which hides the details of transport layer connections from the applications. Application layer connections can be preserved although the IP address changes by tracking the TCP-connection parameters such as number of bytes received or sent, and initializing a new connection with these parameters when the new addresses are known.

NUTSS off-path signaling is protected with SIP authentication and TLS (Transport Layer Security) encryption if needed. The actual data traffic can be protected with TLS also. Usage of cryptography is agreed on during the off-path signaling. The greatest security benefit introduced by NUTSS is that it allows firewalls and other middleboxes in the network to take a larger role in providing the security. Middleboxes can affect the data traffic through more advanced security policies. This protects especially privately administrated servers that may have very limited access control policies.

Using SIP URIs for naming hosts provides user-friendliness. URIs are familiar to users because they resemble e-mail addresses. Additionally the technology behind NUTSS, especially SIP, is already used by many through instant messaging and VoIP applications. The upper layer naming concepts of NUTSS are therefore a strong candidate for regular users. Access control handling should be easier to users, because of simple program-like format of policy definitions and default-off policy which means that all connections that are not authorized are dropped by default. ISPs can also participate in securing their clients by providing more intelligent firewall policies, content filtering and authentication rules for communication endpoints.

As stated earlier, many components of NUTSS architecture are already widely used and therefore the effects of their usage are well-known. NUTSS can be deployed incrementally. Hosts can use publicly available policy-boxes until their own domain has a policy-box. NUTSS implementation provides APIs for both NUTSS-aware and legacy applications. To use NUTSS through a legacy application the user needs to encode the hostname to include connection parameters. This can be cumbersome, since the encoded hostname will be a string presentation of all parameters needed by NUTSS library, for example recipient type and application, that would normally be filled in by the user application.

NUTSS architecture development is currently at a very experimental level. Researchers at Cornell University have created a proof-of-concept implementation of NUTSS. More research and work on implementations is needed to reach confidence of the applicability of NUTSS in Internet-wide deployment.

7 Conclusion

This paper presented three proposals for improving the Internet architecture. The proposals, Host Identity Protocol, Unmanaged Internet Architecture and NUTSS architecture, were analyzed from the viewpoint of a regular home user. All provide solutions to the deficiencies of the current Internet architecture. HIP and UIA add a new name layer to the networking stack between the network and transport layers whereas NUTSS provides a higher layer solution with signaling above the transport layer.

UIA answers to home user needs with zero-configuration network connectivity and intuitive management of personal namespace. NUTSS concepts such as URIs resembling e-mail addresses are likely to be familiar to home users and therefore easily acceptable. All proposals provide at least some level of legacy application support which reduces de-

ployment issues. Both HIP and NUTSS require some support from the infrastructure which will certainly restrain wide-scale deployment. NUTSS is strongly dependable on central administration of the user identifiers, unlike HIP and UIA which use cryptographic identifiers that can be generated by the host itself. UIA operates completely without public infrastructure but is not likely to scale to Internet wide deployment. As a solution to small home networks it seems to be ideal.

HIP offers extensive security features but the end-middle-end model of NUTSS security is interesting also, since it can considerably strengthen the protection provided by firewalls and moves some load of the access control to the network. A hybrid solution would be possible as NUTSS off-path signaling could be used for resolving HIs and HIP connection parameters for data traffic. The HI could then work as the secure token to prove the authorization to the middleboxes.

Home networking solutions are dependent on the development of the Internet architecture and currently the limitations of the architecture affects the evolution of home networks. More research on the technological solutions as well as their effects on the Internet and home networking is needed. Especially the issues concerning deployment should be addressed.

References

- [1] B. Ford, J. Strauss, C. Lesniewski-Laas, S. Rhea, F. Kaashoek, and R. Morris. Persistent personal names for globally connected mobile devices. In *Proceedings of 7th USENIX Symposium on Operating Systems Design and Implementation*, pages 233–248, November 2006.
- [2] S. Guha. Stunt - simple traversal of udp through nats and tcp too. Work in progress, IETF Network Working Group, December 2004. URL <http://nutss.gforge.cis.cornell.edu/pub/draft-guha-STUNT-00.txt>.
- [3] S. Guha and P. Francis. An end-middle-end architecture for a secure internet, January 2007. URL <http://www.cs.cornell.edu/people/francis/sigcomm-nutss-2007.pdf>.
- [4] Saikat Guha and Paul Francis. Towards a Secure Internet Architecture Through Signaling. Technical Report cul.cis/TR2006-2037, Cornell University, Ithaca, NY, 2006.
- [5] T. Henderson. End-host mobility and multihoming with the host identity protocol. Internet-draft, IETF Network Working Group, March 2007. URL <http://tools.ietf.org/html/draft-ietf-hip-mm-05>.
- [6] Moskowitz R. Nikander P. Jokela, P. Using esp transport format with hip. Internet-draft, IETF Network Working Group, February 2007. URL <http://tools.ietf.org/html/draft-ietf-hip-esp-05>.
- [7] M. Komu, S. Tarkoma, J. Kangasharju, and A. Gurtov. Applying a cryptographic namespace to applications. In *DIN '05: Proceedings of the 1st ACM workshop on Dynamic interconnection of networks*, pages 23–27, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-144-9.
- [8] M. Komu, S. Schuetz, M. Stiemerling, L. Eggert, and A. Pathak. Hip extensions for the traversal of network address translators. Internet-draft, IETF Network Working Group, March 2007. URL <http://tools.ietf.org/html/draft-ietf-hip-nat-traversal-01>.
- [9] J. Laganier and L. Eggert. Host identity protocol (hip) rendezvous extension. Internet-draft, IETF Network Working Group, June 2006. URL <http://tools.ietf.org/html/draft-ietf-hip-rvs-05>.
- [10] J. Laganier, T. Koponen, and L. Eggert. Host identity protocol (hip) registration extension. Internet-draft, IETF Network Working Group, June 2006. URL <http://tools.ietf.org/html/draft-ietf-hip-registration-02>.
- [11] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. RFC 4423, IETF Network Working Group, May 2006.
- [12] P. Nikander, J. Arkko, and B. Ohlman. Host identity indirection infrastructure (hi3). In *Proc. of The second Swedish National Computer Networking Workshop 2004*, November 2004. URL http://www.ambient-networks.org/docs/Host_Identity_Indirection_Infrastructure_Hi3.pdf.
- [13] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, IETF Network Working Group, June 2002.
- [14] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN - simple traversal of user datagram protocol (UDP) through network address translators (NATs). RFC 3489, IETF Network Working Group, March 2003.
- [15] J. Rosenberg, R. Mahy, and C. Huitema. Traversal using relay nat (turn). Internet-draft, IETF Network Working Group, March 2006. URL <http://tools.ietf.org/html/draft-rosenberg-midcom-turn-08>.
- [16] P. Srisures, B. Ford, and D. Kegel. State of peer-to-peer(p2p) communication across network address translators(nats). Internet-draft, IETF, February 2007. URL <http://www.ietf.org/internet-drafts/draft-ietf-behave-p2p-state-02.txt>.