SIP SERVICES AND INTERWORKING WITH IPv6

P. Flykt and T. Alakoski

Nokia, Finland

INTRODUCTION

This paper presents a research project that has implemented a Mobile IPv6 enhanced native IPv6 network on which different TCP and UDP based protocols are used. The most important protocol is considered to be the Session Initiation Protocol, which defines the infrastructure and end client behavior for efficient end-to-end session setup.

An overview of the project targets is first given and each technology used is shortly described after that. Topics covered are IPv6 and Mobile IPv6, transition mechanisms and the Session Initiation Protocol (SIP). The project achievements and possible future work is also presented. A conclusion of the technologies used in the research project is given at the end of the paper.

THE RESEARCH PROJECT

The research project was started in order to provide an understanding of the benefits of an IPv6 based non-telecom approach to internet applications and the infrastructure behind it with the focus being on telephony applications. The project also aimed at getting at least a glimpse of the benefits and drawbacks a fictional IT centric competitor could face when starting literally from somebody's less used garage.

The research project was focused on building a SIP infrastructure functional both in IPv4 and IPv6 networks. The target of the research projects was to find solutions that would be independent of the layer 2 network architecture. Especially reducing the overall complexity was an important goal with the limited personnel resources the research project was facing.

The research project consisted of two parts. The first part of the project focused on building a Mobile enhanced IPv6 network with the addition of native IPv6 services. Translation between IPv6 and IPv4 was also on the project agenda. The goal of the second part was to create a SIP server infrastructure. The SIP server was required to be capable of supporting session initialization between any number of SIP User Agents, i.e. SIP clients. The target was also to enable SIP sessions and media streams between native IPv6 and IPv4 hosts by using a dual stack SIP proxy and a IPv6 to IPv4 protocol translator for the media streams. Centralization of the location information was thought to be done by using the Lightweight Directory Access Protocol (LDAP). It was hoped that LDAP would give good scalability and that more LDAP servers could be added to increase location information transaction capabilities.

The ultimate goal of the research project was to have an IPv6 network where not only a SIP but also other services could be used on a daily basis. Sessions from and to the IPv6 network would be transparent to the IPv4 hosts requesting or providing these services. In reality the project does not yet have all the resources needed to make this happen but luckily more and more applications start to support IPv6 natively which means that using these applications nowadays is merely a question of downloading and compiling the source code.

IPV6 – MOBILE IPV6

IPv6

The IPv6 protocol (1) was designed in 1995 by IETF to solve the address exhaustion problem with IPv4. Other demands which had come up within the 20 years IPv4 had been used were also improved.

The address length of IPv6 was increased to 128 bits. The big address space allows addresses to be organized in more hierarchical ways, which will speed up the routing in the core networks. The header size is now fixed to increase the performance and new option headers are included to be able to add new extensions in the future without modifying the base protocol.

IPv6 eases the work of operators by including a stateless autoconfiguration of addresses. DHCP and other address acquisition protocols may also be used. There are three types of addresses: unicast, multicast and the new anycast. Unicast and multicast function as in IPv4. The new anycast address can be given to multiple hosts. When a packet is destined to an anycast address it goes to the nearest host which has that address configured.

IPv6 includes a Neighbor Discovery protocol (2) which finds hosts and routers on the local link. This replaces the IPv4 ARP protocol. IPv6 has added security by including mandatory support for the IPSec protocol (3) to do authentication and encryption individually for each packet.

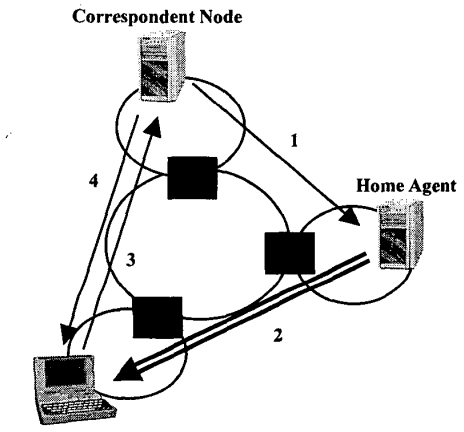
Mobile IPv6

Mobile IPv6 (4) allows hosts to always maintain connection to the Internet using the same IP address.

Each mobile node has a permanent home address, which is a global IPv6 address in the Mobile Node's home network. The home network has a new network element called the Home Agent which keeps record of the current location of its Mobile Nodes.

When the Mobile Node moves to a new foreign network it acquires a temporary care-of address using e.g. stateless autoconfiguration or DHCP. It then informs its Home Agent of its new location by sending it a registration message. Upon receiving and accepting the registration the Home Agent captures all packets destined to the Mobile Node's home address and tunnels them to the registered care-of address.

The Mobile Node also sends a registration message to all of its Correspodent Nodes. The Correspondent Nodes make an entry into their binding caches in order to associate the Mobile Node's home address to the new care-of address. Any future packets from the Correspondent Nodes are then sent directly to the care-of address thus optimizing the data flow.



1 : The Correspondent Node (CN) sends a datagram to the Mobile Node's (MN) home address.

2 : The Home Agent (HA) captures the datagram and tunnels it to the MN's Care-of Address.

3 : The MN receives the datagram and sends a reply containing the binding update message.

4 : The CN updates its binding cache and the future datagrams are send directly to the MN's Care-of Address.

Figure 1 Mobile IPv6

TRANSITION MECHANISMS

Moving from IPv4 to IPv6 will likely to be a slow and a gradual changeover. The transition period will be years when IPv4 hosts coexist with IPv6 hosts. To make the transition smooth a set of transition mechanisms has been introduced by the IETF.

At the early stage of IPv6 deployment there will be isolated IPv6 sites, "IPv6 islands", which are surrounded by the IPv4 network, "IPv4 ocean". Transition mechanisms provide a way to carry IPv6 packets between the isolated islands and to allow IPv6 hosts talk with IPv4 hosts.

Transition mechanisms can been divided into three groups: dual stack, tunneling and translators.

Dual stack

Dual stack hosts have both IPv4 and IPv6 stacks implemented (5). A host with a dual stack can send and receive both IPv4 and IPv6 datagrams and is the easiest technique for communicating in both IP realms. Both IPv4 and IPv6 addresses are needed for the host. Tunnel endpoints and translators must also have dual stacks.

Tunneling

With tunneling the IPv6 datagrams are encapsulated within IPv4 headers at the source end and then sent through the IPv4 network to the destination. The destination host removes the outer IPv4 header and processes the IPv6 packet normally.

Automatic and configured tunnels. Tunnels can be used on hosts or on intermediate routers to connect separate IPv6 hosts or isolated IPv6 networks together (5). With configured tunnels the endpoints of the tunnel has to be manually configured by the network administrators. Usually configured tunnels are used to connect IPv6 sites where IPv6 traffic will be exchanged regularly.

In automatic tunneling the encapsulating host determinates the endpoint of the tunnel from the IPv6 address of the datagram. The destination IPv6 address has to be an IPv4 compatible address from which the 32 low order bits of the IPv6 address are extracted and used as the tunnel endpoint IPv4 address.

6 over 4. The 6 over 4 tunneling (6) allows IPv6 hosts or sites to interoperate with each other by using IPv4 multicasting and tunneling. 6 over 4 hosts are dual stack systems with IPv4 addresses and have the ability to join multicast groups.

A virtual link is created by using IPv4 multicast over which IPv6 neighbor and router discovery messages are transmitted. The hosts encapsulate these messages in IPv4 multicast packets and send them to predefined IPv4 multicast addresses. Instead of the hardware address of the interface the host replies with its IPv4 address in the neighbor discovery message. Since the IPv4 addresses of the hosts are known, a host-to-host connection may then be created with an IPv6-in-IPv4 tunnel.

6to4. 6to4 (7) uses a unique IPv6 address prefix to route datagrams via tunneling over the IPv4 ocean. It is

used to connect isolated IPv6 islands together through an IPv4 network. The prefix of the 6to4 address has the IPv4 endpoint of the tunnel embedded in it. The 6to4 address is illustrated in Figure 1.

The border router of the IPv6 island extracts the IPv4 address from the destination's 6to4 address. The IPv4 address is the global IPv4 address of the border router of the destination's network. The sender's border router then uses this IPv4 address to automatically configure an IPv6-in-IPv4 tunnel between the IPv6 islands.

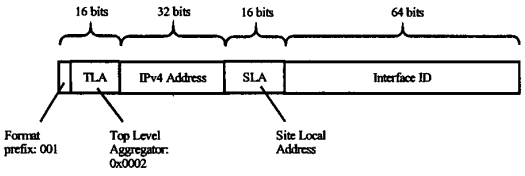


Figure 2 6to4 address prefix

Tunnel broker. The idea behind the tunnel broker (8) is to have dedicated servers, called tunnel brokers, to automatically manage IPv6-in-IPv4 tunnels for users. The tunnel broker also assigns an IPv6 address for the client and to the other endpoint of the tunnel. This scheme fits well for isolated IPv6 hosts and small IPv6 sites.

The dual stack client makes a connection to the tunnel broker and sends authentication information to the server. The connection may use e.g. the http protocol. After successful authentication the tunnel broker determines and configures the tunnel endpoint server for the client. It sends a reply telling the global IPv6 addresses of the client and the tunnel server. It also registers both addresses with the DNS server. Every tunnel has a lifetime and will be removed after its expiration. The tunnel broker might also have some tunnel management method of keeping track of unused tunnels which it would then remove without waiting for the expiration of the tunnel lifetime.

Realm Specific IP (RSIP). RSIP (9) defines a way for a RSIP gateway to allocate resources for RSIP hosts. RSIP hosts are dual stack systems. When an IPv6 host, i.e. a RSIP host, wants to initiate a session to IPv4 hosts outside of the RSIP realm, it first contacts the RSIP gateway. The RSIP gateway allocates an IPv4 address for the RSIP host. The RSIP host then creates an IPv6-in-IPv4 tunnel to the RSIP gateway and transmits all IPv4 datagrams through the tunnel to the destination IPv4 hosts.

There are two methods to allocate resources for RSIP host:

1. RSA-IP (Realm Specific Address IP). Each RSIP host is allocated a unique IPv4 address from the pool of addresses.
2. RSAP-IP (Realm Specific Address and Port IP). Each RSIP host is given an IPv4 address, which might be shared with other hosts, and some unique port numbers.

RSIP is not only restricted for IPv4-IPv6 transition, the general framework can be used to connect any two protocol realms.

Translators

Translators allow IPv6 only hosts to communicate with IPv4 hosts. Two technologies are available: header conversion and transport relay. In header conversion the IPv6 headers are converted to IPv4 headers and vice versa. Some of the IPv6 option headers can not be converted. Header conversion has problems if the application layer embeds IP addresses or port numbers in the payload. Example of such application is FTP. Header conversion techniques include SIIT and NAT-PT.

A transport relay stands in the middle of TCP and UDP session. When IPv6 TCP or UDP session is requested it first stops in the relay server. The server creates a connection with the IPv6 host giving its own address as the other end of the connection. Then it establishes a second connection with the real destination, which is the IPv4 host. When both of the connections are established, the relay server reads data from one connection and writes it in to the other. Like header conversion the transport relay has problems with IP addresses and port numbers embedded in application data. SOCKS64 (12) is a good example of this technique.

A solution for a protocol with an IP address embedded in the payload is to use an Application Level Gateway (ALG). An ALG is an application specific translation agent in the translator server, which modifies the application payload and performs other necessary functions to make the application work. ALGs have to be made for each application separately. Note that IPSec computes the authentication also over the IP header so IPSec doesn't work with the header conversion.

Stateless IP/ICMP Translator (SIIT). SIIT (10) describes a mechanism in which the IPv4 headers are translated to IPv6 headers and vice versa. SIIT is stateless because it converts each datagram independently without any information from the other datagrams. The translator is usually a box on the border of the IPv6 island. When the IPv6-only host sends a datagram to the IPv4 host it uses an IPv4-mapped IPv6 address. This datagram is then routed to the nearest SIIT box, which performs the header conversion. The source IPv6 host gets assigned a temporary IPv4 address, which is used in the new IPv4 header of the datagram. After conversion the SIIT box sends the datagram to the destination IPv4 address.

SIIT doesn't define how to get the temporary IPv4 address for IPv6 hosts, e.g. DHCP and DNS could be used. SIIT doesn't translate any IPv4 header options, nor does it translate IPv6 routing headers, hop-by-hop extension headers or destination options headers.

Network Address Translator - Protocol Translation (NAT-PT). The NAT-PT mechanism (11) makes the IP protocol conversion as described above but suggests a method of assigning an IPv4 address for the IPv6 hosts. The NAT-PT box would be the router on the border of IPv6 island. Two separate methods are available:

1. NAT-PT has a pool of globally routable IPv4 addresses, which it assigns to open sessions. IPv6 to IPv4 address mapping is cached for the duration of the session. Sessions can be bi-directional, i.e. to and from the IPv6 island.
2. All IPv6 addresses are mapped into a single IPv4 address by also translating the port number of the connection. This is called Network Address Port Translator - Protocol Translation (NAPT-PT). With NAPT-PT the sessions can only be outbound from the IPv6 island.

With NAPT-PT the number of available ports on the system limits the number of sessions a NAPT-PT box can have simultaneously. NAPT-PT could be combined with normal NAT-PT by having a pool of IPv4 addresses to use with the port translation.

Because of the nature of the port translation, a NAPT-PT translator with a single IPv4 address can only have one IPv6 host assigned for each inbound connection with a certain port number. E.g. all connections with a destination port number 80 would be redirected to the same host.

SESSION INITIATION PROTOCOL

The Session Initiation Protocol (SIP) (13) provides a contact infrastructure whereby hosts can contact each other and set up different kinds of sessions between themselves. A SIP message consists of two parts, where the first part contains the information that is relevant when forwarding the message through the SIP infrastructure to the intended recipient and the second part describing the end-to-end session between the hosts. For audio/video content the Session Description Protocol (SDP) is used, but also other kinds of payload describing the session can be sent, e.g. chess application setup data, text, pictures, etc. Any number of these payloads can be sent in a SIP message.

The SIP infrastructure consists of SIP User Agents, i.e. the hosts, and SIP servers where the User Agents initiate and accept SIP sessions and the servers create the necessary infrastructure that connects the SIP User Agents.

There are three different kinds of SIP servers, namely SIP proxies, SIP redirect servers and SIP registrars. Of these the SIP registrar server accepts location information updates from SIP User Agents and stores it for future use. The SIP proxy and the SIP redirect servers are used when routing a SIP session from the initiator to the destination. Of these, the SIP proxy server accepts the SIP message and forwards it to the next location, whereas the SIP redirect server replies with the location information to the inquirer. The proxy and the redirect server acquire the location information by acting as SIP registrars or by querying a SIP registrar server using for example LDAP.

RESEARCH PROJECT RESULTS

Network environment

The construction of the test network was focused on building an IPv6 network with support for Mobile IPv6. The IPv6 network was built exclusively on Linux due to the availability of the source code and expertise. The Mobile IPv6 functionality was freely available from Helsinki University of Technology and was created by the MIPL student project. The network environment also supports the 6to4 transition mechanism, which enables easy connectivity between isolated IPv6 network "islands".

The IPv6 network is also able to support UDP interworking between IPv6 and IPv4. This is accomplished by adding a NAT-PT translator into the network. The translator itself uses the translation functionality described by SIIT together with NAT techniques mapping one IPv4 address to one IPv6 address. The network translation functionality is implemented as a user space daemon.

Native DNS over IPv6 could also have been supported using the BIND 9 DNS software but unfortunately the main C library did not support DNS queries over IPv6. Therefore only static mappings from host names to IPv6 addresses were used in the network.

Network applications

In the network several applications have been compiled and installed. Currently it is possible to use at least the Kphone SIP User Agent using the Dissipate SIP stack, SSH, Apache web server, Mozilla web browser, FTP client and server, Exim SMTP mail server, Courier IMAP server and client, and Xtris and BattallaNaval games natively in the IPv6 network. Of these applications only the Xtris game and the Dissipate SIP User Agent needed some coding to support IPv6, the others are used with minor modifications.

SIP infrastructure

The SIP infrastructure, which consists of SIP servers and SIP User Agents, is the most important application

supported by the research IPv6 network. The SIP server code was built in-house and provided us with good experience of SIP. Simple SIP message authentication was also implemented both in the SIP servers and the SIP client.

The Kphone SIP User Agent application that includes the Dissipate SIP protocol stack was used as the SIP client application in the project. The client was originally written to support only IPv4 but has been modified to use IPv6 addresses. Mobile IPv6 handles the mobility of the SIP clients and of other hosts. SIP mobility is used when the user changes to a different terminal which has SIP User Agent capabilities

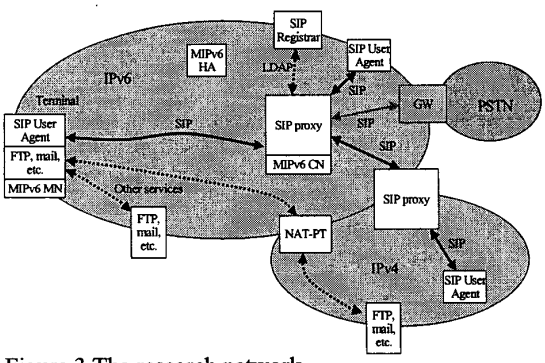


Figure 3 The research network

FUTURE RESEARCH ITEMS

To achieve higher loads for each SIP server the load has to be shared between more than one server. This can be accomplished by letting DNS take care of the load balancing when giving out multiple IPv6/IPv4 addresses for one domain name. The other option is to have the SIP server process only those messages that correspond to a specific connection, which could be determined for example by examining the remainder when dividing the last octet of the IP address with the number of SIP servers in the load balancing pool. By distributing the load over several servers each server needs the registered location information from the SIP registrar server. Loading the registration information from the registrar server could be based for example on the LDAP protocol. Also SIP messaging scenarios and switching between unicast and multicast sessions could be implemented and studied with the basic infrastructure in place.

CONCLUSIONS

It has become almost possible to support IPv6 as the only protocol for communication reaching not only other IPv6 hosts but also the existing IPv4 Internet infrastructure. The emphasis is on the word "almost" as it requires some more engineering in filling the last gaps. Fortunately more and more software are starting to take IPv6 into account, even the DNS resolver is being developed towards natively supporting IPv6.

There also exists many solutions to solve the interworking between IPv6 and IPv4 networks. IPv6 communication can be transported over IPv4 in a variety of ways and native IPv6 hosts are able to exchange messages with IPv4 hosts. There still needs to be more work done to support protocols, e.g. FTP, that store IP layer information in their protocol messages.

REFERENCES

1. S. Deering and R.Hinden, 1995, " Internet Protocol, Version 6 (IPv6) Specification", RFC1883
2. T. Narten, E. Nordmark and W. Simpson, 1996, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 1970
3. S. Kent and R. Atkinson, 1998, "Security Architecture for the Internet Protocol", RFC2401
4. D. Johnson and C. Perkins, 2000, "Mobility Support in IPv6", draft-ietf-mobileip-ipv6-12.txt (work in progress)
5. R. Gilligan and E. Nordmark, 2000, "Transition Mechanisms for IPv6 Hosts and Routers", RFC2893
6. B. Carpenter and C. Jung, 1999, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC2529
7. B. Carpenter, K. Moore, 2000, "Connection of IPv6 Domains vial IPv4 Clouds", draft-ietf-ngtrans-6to4-07.txt (work in progress)
8. A. Durand, P. Fasano, I. Guardini and D. Lento, 2000, "IPv6 Tunnel Broker", draft-ietf-ngtrans-broker-05.txt (work in progress)
9. M. Borella, D. Grabelsky, J. Lo and K. Tuniguchi, 2000, "Realm Specific IP: Protocol Specification", draft-ietf-nat-rsip-protocol-07.txt (work in progress)
10. E. Nordmark, 2000, "Stateless IP/ICMP Translation Algorithm (SIIT)", RFC2765
11. G. Tsirtsis and P. Srisuresh, 2000 "Network Address Translation - Protocol Translation (NAT-PT)", RFC2766
12. H. Kitamura, 2000, "A SOCKS-based IPv6/IPv4 Gateway Mechanism", draft-ietf-ngtrans-socks-gateway-05.txt (work in progress)
13. M. Handley, H. Schulzrinne, E. Schooler and J. Rosenberg, 1999, "SIP: Session Initiation Protocol", RFC 2543