UNIVERSITY CARLOS III OF MADRID

Department of Telematics Engineering

Master of Science Thesis

Extending the address space of the Internet

Author: **Michal Kryczka**

Supervisor: **Dr. Arturo Azcorra**

Leganes, September 2009

Abstract

In the nearest future Internet in its current shape will have to face the problem of shortage of IPv4 addresses. IPv4 architecture reserves 32 bits for addressing purposes, which occurs not to be enough in the face of a rapid growth of the Internet. To overcome this problem, the successor of IPv4, IPv6, was designed with 128 bits of addressing space. However, deployment of IPv6 goes very slowly mainly because of the cost of transition, even if many transition mechanisms were already proposed. Therefore, it could be advantageous to explore solutions which can occur better than IPv6 in terms of deployment ease or will allow for IPv4 and IPv6 coexistance with minimal cost.

This thesis presents new approach for solving a problem with lack of addresses. xIP architecture creates new address space by extending IPv4 address scheme. xIP can be incorporated as a completely new protocol or can be used as a mechanism which can improve IPv6 and IPv4 coexistance (xIP6). The common characteristics of both of the architectures are integrated routing and addressing and reusing extensive part of IPv4 world. This solution seriously decreases management cost during coexistence period and makes deployment less expensive and easier to accept by users.

Table of Contents

List of Figures

Chapter 1

Introduction

Exhaustion of IPv4 addresses was forecast long time ago and several solutions to address this problem were proposed. However, none of them has been widely deployed. As a result, exhaustion of IPv4 address space is nowadays a serious problem which present Internet needs to overcome. The current predictions say that we will run out of addresses between 2011-2013 [1]. This may lead to important connectivity issues and could possibly stop further growth of the Internet.

IPv6 was standardized by IETF as a replacement of IPv4 in 1998 [2][3], however the deployment of new architecture has been very slow. In November 2008, on the occasion of 10th anniversary of the IPv6 standardization, Google has conducted the research to determine the state of IPv6 deployment [4]. The results indicated that the percentage of IPv6 prevalence in overall Internet traffic is less than 1% in each country, with Russia as a leader (0.76%). These numbers show clearly that, even if IPv6 is a stable and well tested technology, its presence in the Internet is still very low, and it is not an unjustifiable assumption that deployment of IPv6 in its current shape may in fact never occur, or at least it will take very long time.

In order to tackle above problem, we propose xIP and its variant xIP6. The main approach of the architecture is to extend address space while keeping significant part of IPv4 world. xIP uses the mechanism of integrated routing and addressing. New address space is created as an suffix to present IPv4 addresses. In consequence, new addressing and also routing is integrated with legacy IPv4 one. It not only allows to reuse IPv4 world but it also seriously decreases management cost during the coexistance period of the new protocol and IPv4. In xIP architecture only one routing topology exists, thus there is no need to duplicate routing tables nor routing policies. We believe that such an approach is better than one of IPv6, which is oriented on abandoning IPv4 - the most popular IP protocol in present Internet.

The rest of the thesis is organised as follows. Section 2 provides brief information about previous solutions which attempted to overcome the address shortage problem. In Section 3 the xIP architecture is presented. The idea of integrated routing and addressing is described and ways of communicating of hosts with xIP are presented. In Section 4 plan of deployment of xIP is described and case scenarios are included before concluding in Section 5.

Chapter 2

State of the Art

The problem of address exhaustion was predicted quite early and there were many solutions which were focused on expanding address space to overcome IPv4 shortage address. Next sections contain information about solutions which base on NAT and new protocols, including IPv6, which was designated by IETF as a successor of IPv4.

2.1 NAT and NAT Improvements

Network Address Translator (NAT) [5] is so far the only widely used solution for lessen the IPv4 address shortage problem. It extends address space by separating private and public address blocks and providing translations between them. Even if widely deployed, NAT has several well-known disadvantages [6][7]. The biggest one is that it violates Internet transparency [8]. NAT is a reason of non-global addressing of the host as it can translate only connections initiated by nodes from private space. It seriously limits connectivity, however gives some form of security. The other problem is end-to-end inconsistency as addresses of the packets needs to be translated on their path. It creates incompatibility with some of the applications which carries addresses in the application data like FTP for example. As a result, additional Application Gateways needs to be implemented isnide the NAT device. Despite these disadvantages, NAT was able to gain big popularity because of its simplicity, cheapness and compatibility with IPv4 infrastructure. NAT was able to slow down IPv4 exhaustion, however, growth of the Internet and arising number of end hosts put us in the situation that this solution is not enough.

To overcome problems with the NAT and to extend its funcionality several improvements were proposed. One of the solution, called Network Address Port Translation (NAPT) is the extension of the NAT described in [9] and offers a possibility of sharing one public address through many devices. It is done by adding to the NAT a funcionality of translating transport idenitifer, like TCP or UDP port numbers. Realm Specific IP [10] makes possible to obtain a public address by a host in a private realm, if it needs to contact with a peer with public address. The address is leased by a host during some time and then it is returned to the pool and can be used by other hosts. This solution gives better transparency than pure NAT (no Application Gateways are necessary as end-to-end consistency is assured) but in fact it does not provide more address space. Another interesting architecture, IPv4+4, was presented in [11]. The main approach was to assure global addressing of the hosts behind the NAT.

IPv4+4 bases on encapsulated IPv4 in IPv4 packets which keeps two addresses: public address of the NAT which connects realm to the Internet, and private one of the host. These addresses are swapped in NAT devices. However, solution demanded not only NAT but also end-hosts to be changed, what seriously limits the deployment.

2.2 New Protocols

One of the first protocol which was designed in order to tackle with IPv4 address shortage problem was Extended Internet Protocol (EIP) [12]. It is a proposal which extends IPv4 header for keeping identifier of the network, while current IPv4 address is supposed to be locally unique and identifies end hosts. Solution reuses IPv4 architecture, however, because of serious changes it implies, like chosing new global rotuing scheme or updating backbone and border routers, the architecture was never adopted.

Oxygen [13] is an proposal of extending IPv4 address space for 32 additional bits. For this purpose a new IP option is defined and additional source and destination address is located in option field. Thanks to this approach Oxygen maintains considerable degree of backward compatibility and can be deployed with quite low complexity.

Another proposal which extends address space is called TRIAD [14] and it is mainly focused on content distribution. Main component of TRIAD is name based routing performed by Directory Relay Protocol (DRP) and Wide-Area Relay Protocol (WRAP). All end to end identification is based on Fully Qualified Domain Name (FQDN) rather then IP addresses. IP addresses are significant only locally. However, TRIAD is not very scalable solution (name-based routing is hard to perform as domain names are not really aggregated) and as it changes global semantic of addresses, it implies many upgrades in current infrastructure.

FQDN and name based routing are also used in IP Next Layer (IPNL) proposal [15]. This architecture bases on IPv4 infrastructure and adds an extra layer between IP and TCP/UDP layers. The topology of IPNL network contains huge middle realm (continuation of present Internet) connected with private realms or directly attached hosts. Moreover, private realms can be aggregated in realm groups and reach middle realm with a single middle realm IPv4 address. Full IPNL address is 80-bits long and contains 32-bits long middle realm IPv4 address, 16-bits long realm group number and 32-bits long end host IPv4 address. However it is FQDN not IPNL address which is used as end host identifier. IPNL routers can route packets based both, on FQDN and IPNL address. At the beginning of the connection, FQDN is used, but when IPNL addresses are learnt, routing is based on IPNL addresses due to performance reason. IPNL partially reuse present infrastructure but also has some drawbacks including the need of significant change in TCP/IP stack. Because of the high complexity of the architecture, IPNL is not easy to deploy.

Different solution for IPv4 address exhaustion is proposed in [16] and is known as a Multi-Tier Architecture for the Internet of Internets (MTAII). The main idea presented in the paper is to not make any changes in hosts (they should still use IPv4 addresses) and to isolate IPv4 on hosts from the WAN architecture. Transmission of IPv4 packets should occur through WAN architecture and is conducted by Specific Internet Protocol (SIP) and transmission of SIP traffic is supported by Multi-Tier Internet Protocol. MTAII solves typical NAT problems like end-to-end inconsistency or non-global addressing. Its big advantage is a possibility of incremental deployment, however architecture assumes a tree-like topology of new Internet. New packets has also significant overhead as they carry addresses of all gateways connecting different tiers between source and destination.

2.3 IPv6

As it was stated on the beginning, IPv6 which was designed as a cure for a problem with lack of IPv4 addresses is still not widely used. The problems and reasons for the slow deployment of IPv6 have already been widely discussed [17] [18] [19]. The most important conceptual issue of IPv6 is that it was designed for transition, not for coexistence with IPv4. IPv6 is a completely new protocol, almost totally unrelated to IPv4. It was assumed that all the Internet applications, hosts, routers, middleboxes, management systems, etc. in the whole Internet would be migrated in a few years, and after this everything would work very well. However, the problems during this so called ”transition phase” were not addressed in depth. This assumption led to a design that contains many relevant drawbacks during this ”transition phase”.

Firstly, IPv6 format packet is different from IPv4 one. This implies that IPv6 packets cannot be processed, even partially, by existing IPv4 routers, middleboxes and other systems. Whole equipment needs to be changed or translation techniques need to be used. An additional problem is that IPv6 addressing assignment is unrelated to the current IPv4 one. This implies that during the transition, there are duplicated addressing schemes. Because of that, routing topology, structure and policies are also duplicated and unrelated between IPv4 and IPv6. What is more, IPv6 lacks really innovative and significant improvements compared to IPv4. Most of such improvements (like enhanced security with IPSec) can be successfully ported to IPv4. For this reason many users do not migrate to IPv6 and decide to live with IPv4+NAT scheme as a cheaper, albeit more restricted alternative.

Many transition methods were proposed in order to speed up and facilitate IPv6 deployment. Because of the fact that IPv4 is still the most common Internet protocol, some of the techniques which allow passing IPv6 packets through IPv4 environment are needed.

The most often used solution is tunneling. The most basic method, called 6in4 uses protocol 41 of IPv4 in order to encapsulate IPv6 packets [20]. This method is used in some other proposals which try to allow IPv4 and IPv6 coexistance. One of the most common protocol is 6to4 [21]. It allows IPv6 packets to be transmitted through the IPv4 network without explicit configuration of the tunnel. 6to4 assigns IPv6 prefix in such a way that it corresponds with IPv4 address. It is done by prepending hexadecimal value of 2002 to public IPv4 address. As a result, any IPv6 site with at least one IPv4 public address can communicate with IPv6 site, even without IPv6 support from ISP.

The other transition mechanism, Teredo [22], is present in operation systems from Microsoft Windows family. It allows to reach IPv6 site by a device which possesses private IPv4 address behind NAT. It is done by encapsulating IPv6 packets within IPv4 UDP datagrams.

Both of the solutions (6to4 and Teredo) need some support from the infrastructure in the network. Teredo bases on Teredo servers and relays, while 6to4 needs relay routers accessible with anycast IPv4 address. Teredo and 6to4 are intended to be temporary solution, and they assume, that in not so far future all IPv6 hosts will use IPv6 native connectivity.

IPv64 described in [23] is an IPv6 extension which allow IPv64 packets to be backward compatible with IPv4. New packets can be processed by IPv6 and IPv64 routers as a conventional IPv6 packets, and in the same time they can be treated as IPv4 packets by IPv4 devices. Such an approach is advantageous in the situation when transit network is composed of many IPv6 and IPv4 clouds.

Even if many proposals designed for facilitating IPv6 deployment were presented, transition to IPv6 does not happen. It inclines to look for other solution which can solve problems with lack of addressing or can speed up IPv6 deployment.

Chapter 3

xIP - IP Protocol with eXtra Address Space

This Section presents main assumptions of the new solution which is called xIP. The architecture of xIP is introduced and discussed. We present main idea about addressing, packet format and way of communication between the hosts. Next, we introduce xIP6 as a way of improving IPv4 and IPv6 coexistance. In the end, some implications with DNS are discussed.

3.1 Assumptions

Based on problems with IPv6 deployment, which are mentioned in previous Section, we have derived several assumptions regarding our solution. The central idea of the new architecture is to not abandon IPv4 but rather extend it. This is because IPv4 is dominant protocol in the current Internet and overwhelming majority of users utilize it. Forcing all of them to change the well-working protocol can appear to be impossible, as example of IPv6 shows. It also needs to be noted that it is Network Address Translator (NAT) which is so far the only widely deployed solution used for lessen the IPv4 address shortage problem. In spite of some disadvantages of NAT (violating Internet transparency), it was widely accepted by community, mainly because of its approach, which is adding some functionalities to IPv4 with minimum required changes.

From the above, it results that it is essential to make possible to deploy xIP gradually, without serious modification of the current environment. It relates not only to the network equipment but also to all mechanisms involved in correct functioning of the network. xIP has to be able to coexist with IPv4 and changing the protocol from IPv4 to xIP should go smoothly with keeping connectivity between both protocols. This leads to following assumptions:

- Deploying xIP should not raise significantly management cost. xIP addressing scheme, routing topology and tables should be integrated with IPv4. Having one consistent routing topology simplifies network management.
- It should be possible to introduce xIP into the site without changing the core of the

network or devices of ISP. It implies that legacy network devices should be able to process xIP packets.

- It should be possible to introduce xIP into the site only partially. Full connectivity between new xIP hosts and old IPv4 hosts should be guaranteed without complex translation techniques. Additionally, this should not raise management complexity in significant way.
- Legacy hosts should be able to benefit from xIP, with certain limitations. As the legacy hosts are not aware of the extended addressing scheme, translation techniques in xIP-aware devices are needed. Thus, some of the benefits of new addressing architecture, like end-to-end consistency cannot be provided.
- xIP should be easy to implement, upgrade and manage. The simplicity of the new architecture is a key factor to gain success.
- In long term, xIP should restore full transparency of the Internet.

We believe that such an approach, which is evolutionary rather than revolutionary, has bigger chances for the final success than solutions with ”change-everything” conception.

3.2 Architecture

In this section we present the architecture and key elements of xIP protocol.

3.2.1 Integrated Addressing

In present Internet we have two different kind of IPv4 addresses: private one (defined in [24]) which can be assigned for local use and public one which are globally unique and are assigned by IANA. The private addresses need to be unique only locally, however they are not valid on the Internet and need to be translated into public address when packets leave local domain. Nowadays, we are suffering the exhaustion of public IPv4 addresses, which can be used for transporting packet in public Internet. That is why IPv6 was designed with new public address space, much bigger than IPv4 one, but completely unrelated with it.

xIP uses a solution which is called Integrated Addressing. It is based on the idea that newly created address space is hierarchically dependent on the IPv4 one. It is in opposite to IPv6 which creates new, unrelated to IPv4, address space. Instead of creating totally new, independent address space, xIP creates new subspaces. Each subspace works as an extension for one public IPv4 address. Each interface within subspace is identified by extended version of address, which contains public IPv4 address and suffix. As a result, one IPv4 public address can be used for addressing many hosts depending on the length of the suffix. Additionally, opposite to the NAT, which uses lease of public addresses, all of the hosts are globally addressable with their own static address.

Such an approach strongly simplifies network management in comparison to IPv6 approach. It is not necessary to maintain two addressing schemes and hence such things like routing topology and routing policies are not duplicated. It saves not only resources but also time which network administrators spend on configuring and managing. Because of one consistent scheme instead of two separated ones, integrated addressing architecture makes troubleshooting and reacting for potential problems easier and faster.

3.2.2 xIP Network

New architecture bases on xIP networks (x-networks) which are inside current IPv4 site and function in IPv4 environment. Each of the subnet has an edge router (x-router) which serves as the gateway of the x-network. This router has public (globally unique) IPv4 address assigned. The x-network contains hosts which have their private addresses unique inside the site. Moreover, all devices which belong to the x-subnet have assigned extended xIP address (x-address). This address contains two parts. First one is common for a whole x-subnet, and it is IPv4 address of the router. Second part of the address is an extension (suffix) to IPv4 address and is unique for every interface within the subnet. In consequence, each host can be identified by two addresses. One is private IPv4 address which is assigned basing on the same rules as now and is used for IPv4 communication and the other is x-address which is built by connecting public address of the router and x-part of the address. This address is used for a communication with nodes from remote x-network.

The example of such a network in IPv4 environment is presented on Fig. 3.1. As it is visible on the illustration, xIP architecture does not need to be introduced in whole realm at once what allows to spread the costs over a longer period. In the picture we have a network A which is only IPv4 inside IPv4 site. Then, there is a possibility of introducing x-network B only with putting xIP router, which will be the gateway for legacy IPv4 hosts which are inside the x-network. From that time, subnet B is xIP aware and hosts inside are globally addressable by usage of extended address, even if each device inside is only IPv4 aware. However, in this case address and packet translations made by x-router are necessary. Finally, with introducing xIP aware hosts we can create x-network C with xIP aware hosts. Thanks to this, whole communication will be fully transparent and no address translation will be necessary. What is important is that during whole deployment process, full connectivity inside the local network is maintained.

3.2.3 xIP Packets

One of the aims of xIP is to utilize a significant part of IPv4 architecture. That is why one of the claims is that xIP packets should be easily processed by IPv4-only devices. It implies that xIP packet should have a format of IPv4 packet with information about extended addressing hidden in a way, which will not be changed in any way by IPv4 devices

A way of differentiating an InRA packet from an IPv4 packet needs to be provided. It should be assured that IPv4 devices will not erase or change information stating that the packet is InRA compatible. To accomplish this, first bit of the seventh octet of the IPv4 header can be used (i.e. 48th bit of the header, beginning with bit 0). This bit, (informally called ”evil bit”), is now not used by IPv4 protocol and according to specification in RFC 791 [25] it must be set by the source to zero value and ignored by processing devices. xIP devices should set this bit when putting in packet extra address information. Thanks to this, all xIP-aware devices will be able to distinguish if packet contains additional information regarding addressing while IPv4-only devices will still be able to process packet as conventional IPv4

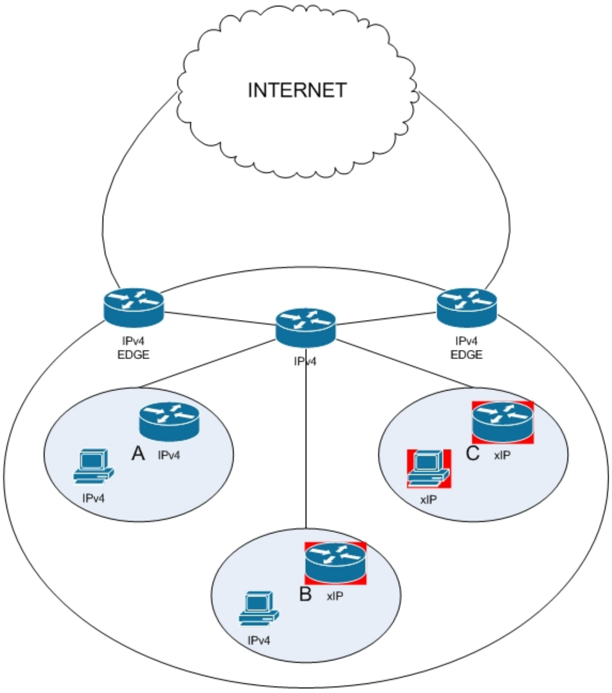


Figure 3.1: Example of xIP network in IPv4 environment

packet. We carried out several tests in the Internet and packets with the bit set to one were processed correctly and the value of the bit was not changed.

The other issue is a way of putting extra address information into IPv4 compatible packets. We have three possible solutions to do that: using a fields from current IPv4 header, using IPv4 options or tunneling. They are shortly described in next subsections.

Using fragmentation field

In IPv4 header we have two fields which are used for dealing with fragmented packets. These are ID field (16 bits long) used for identifying fragments and Fragment Offset (13 bits long) used for correct reassembling. However, fragmentation of IPv4 packets in current Internet is not very common and additionally you can ensure that packet will not be fragmented by setting Don't Fragment Flag. As a result, we have 29 bits of IPv4 header which are rarely used in communication. Thus, we could use them for carrying extra address information. This would allow us to create extra address with the length of 14 bits (we need to carry both, source and destination) or 13 bits, in case we do not want to separate bits of address. xIP header which uses this solution is presented on Fig. 3.2

However, modifying Fragment Offset field, makes the IPv4 router thinks that packet is fragmented. It has following implications

- xIP packets cannot be translated with NAT. When NAT gets packet with Fragment Offset field bigger than zero, it thinks it has a disordered, fragmented packet. Then, it keeps packet and waits for the first fragment of the packet in order to get information which are necessary to make address translation (like TCP port for example), and

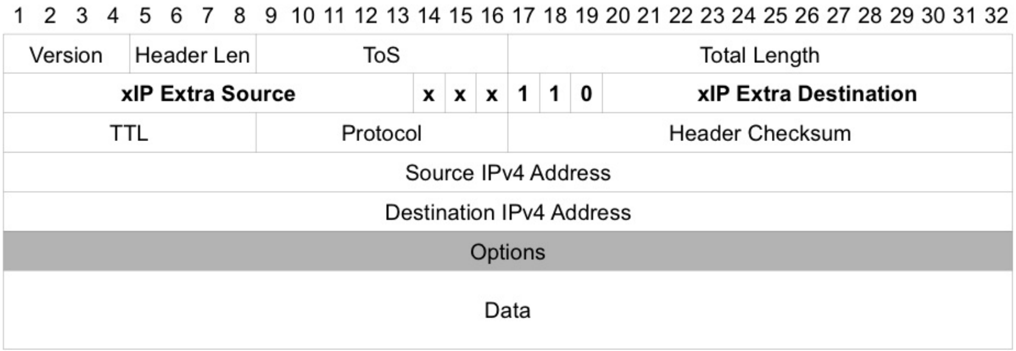


Figure 3.2: xIP header with extra addresses hidden in fragmentation field

which in case of fragmented packets are located in first fragment. This is not a problem for xIP, as xIP packet should be used together with IPv4 public addresses, and should not be NAT-ed.

- Some network attacks uses fragmentation in order to foolish firewalls, thus, some firewalls keep incoming fragmented packets in order to inspect whole packet. Because of that, correct configuration of firewalls is necessary to make it possible using xIP with extended address in Fragment Offset Field. In general it should not be difficult to accomplish. Firewalls mainly inspect incoming packets, so we could assume such a firewall is controlled by a site which wants to introduce xIP and can be easily configured by it.

- From experiments we carried out, we noticed that routers which are based on Linux system have implemented the function which makes them wait for all fragments of fragmented packet. In case if packets are disordered, router waits for all of them and then forwards them in correct order. Additionally, if it is possible (if MTU of next link allows for it), router reassembles the packet. It allows for optimization of usage of the link, but it also violates standard [25] which states that reassembling packets can only take place in the destination. This behaviour, makes it impossible to use xIP with Fragment Offset field for carrying extra address information, with current Linux-based router. In this case, the router will always keep a packet and will delete it after some period of time. Conventional routers (i.e. from Cisco manufacturer) on which we performed tests, do not behave in a way described above, and they process packets according to the standards.

Using ID field and Fragmentation Offset field for carrying extended addressing does not change the size of IPv4 header, what is a big advantage. The size of extra address also looks sufficient as with one public address, more than 8000 devices could be addressed. However, problems with Linux routers can seriously limit the deployment.

Using IP option field

The other solution for carrying extended addresses is to use IPv4 options. This mechanism will allow to adjust size of extra address to our needs and make it bigger than 14 bits

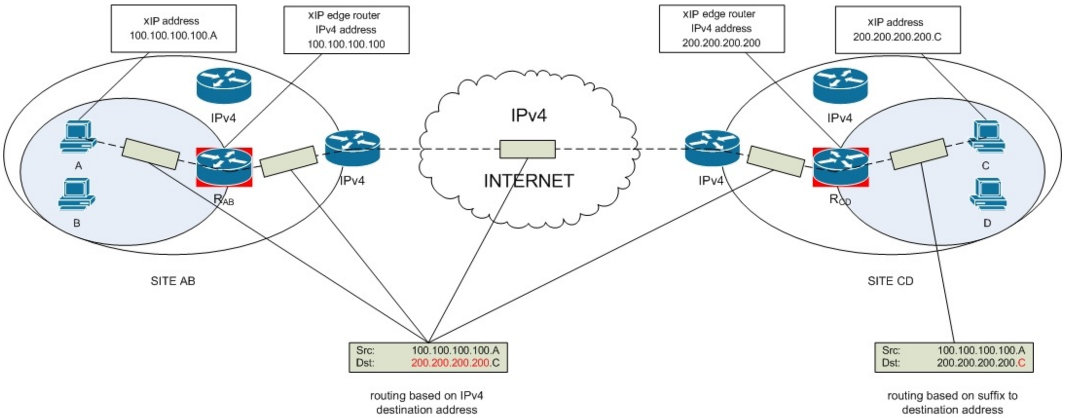


Figure 3.3: xIP Integrated Routing and Addressing

(which is maximum size for the option described above) if necessary. However, this solution produces some overhead. Moreover it needs to be noted that some of IPv4 equipment can erase IPv4 options or process the packet incorrectly, what can cause that information about extra addressing can be lost.

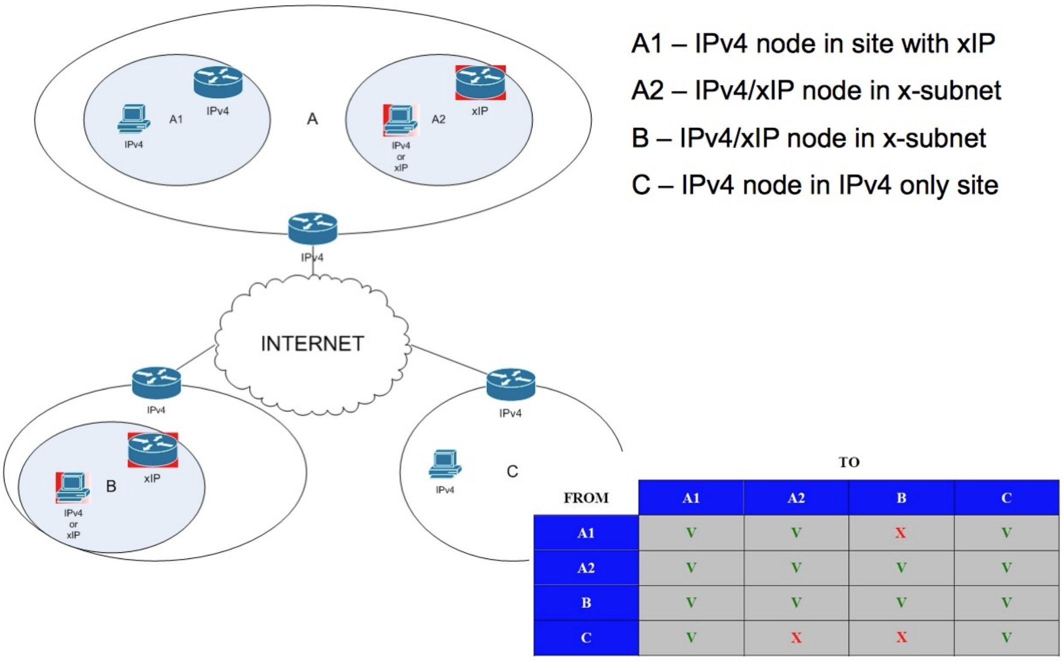
Tunneling

The last solution is to use tunneling and to encapsulate xIP packet into IPv4 packet. This allows to attach not only information about addressing but also whole new xIP header with some additional functions and options. It gives the biggest flexibility, but also produces the biggest overhead.

3.2.4 Integrated Routing and Communication

xIP uses not only Integrated Addressing scheme but also Integrated Routing - routing of xIP packets is connected with IPv4 one. Routing topology and routing tables does not need to be duplicated but only extended in x-aware devices. xIP packets do not have to be translated and they can traverse IPv4 clouds without any problems. It is a big advantage in comparison to IPv6, where translation and tunelling techniques are used to allow passing IPv6 packets through IPv4 environment. What is more, introducing IPv6 to the site implies duplicating whole routing topology what increases complexity and cost management.

xIP packet can traverse through IPv4 environment and routing is based on IPv4 destination address only. This address identifies the xIP-aware edge router of xIP subnet. Once it is reached, packet is delivered to end host in xIP subnet basing on suffix to IPv4 destination. This approach allows to leave current routing behaviour in the core of the Internet unchanged. What is performed it is still IPv4 routing based on longest prefix match, however, entries in routing tables will contain longer prefixes, as with xIP the site will need fewer public IPv4 addresses. It makes routing tables bigger, which cannot be avoid while extending address space. It needs to be assured that announcements with longer prefixes will not be discarded by the routers. The idea of addressing and packet traversing is presented in Fig. 3.3.


A1 – IPv4 node in site with xIP
A2 – IPv4/xIP node in x-subnet
B – IPv4/xIP node in x-subnet
C – IPv4 node in IPv4 only site
A
A1
IPv4
IPv4
A2
xIP
IPv4
or
xIP
IPv4
INTERNET
IPv4
B
xIP
IPv4
or
xIP
IPv4
C
IPv4
TO

FROM	A1	A2	B	C
A1	V	V	X	V
A2	V	V	V	V
B	V	V	V	V
C	V	X	X	V

Figure 3.4: Connectivity during coexistance period

As it was mentioned earlier, hosts in x-networks can be identified with two addresses. They should have assigned IPv4 private address (PAX). Then, local communication inside the site is straightforward and bases on traditional IPv4 routing. Hosts inside x-network sends IPv4 packets with PAX as source address and IPv4 as destination address. They are also reachable by other hosts inside the local site by using PAX. Communication with IPv4 hosts in public network is performed in the same way as today for hosts with private addresses. IPv4 packet with PAX as a source address will be send to edge router where NAT translation will take place.

The communication between two hosts from x-network which are not in the same local network uses xIP packets and extended x-address. If host in x-network is not xIP aware (it is IPv4 legacy hosts), then it sends a packet using a PAX as its source. This packet is translated into x-packet by x-router, which put x-addresses. Then packet crosses IPv4 environment until reaching x-router of the destination, from where it is forwarded to the destination. Description of translating process is presented in next Section (xNAT). With introducing fully x-aware hosts, address translation in x-router can be omitted.

The table in Fig. 3.4 shows possible reachability of hosts. As we can see, the only case when availability is not assured is connection between IPv4 host from IPv4 only site to host from x-network in different site, but opposite communication is possible. This type of reachability cannot be done straightforward and requires upgrades on the IPv4-only site or using techniques like bidirectional NAT.

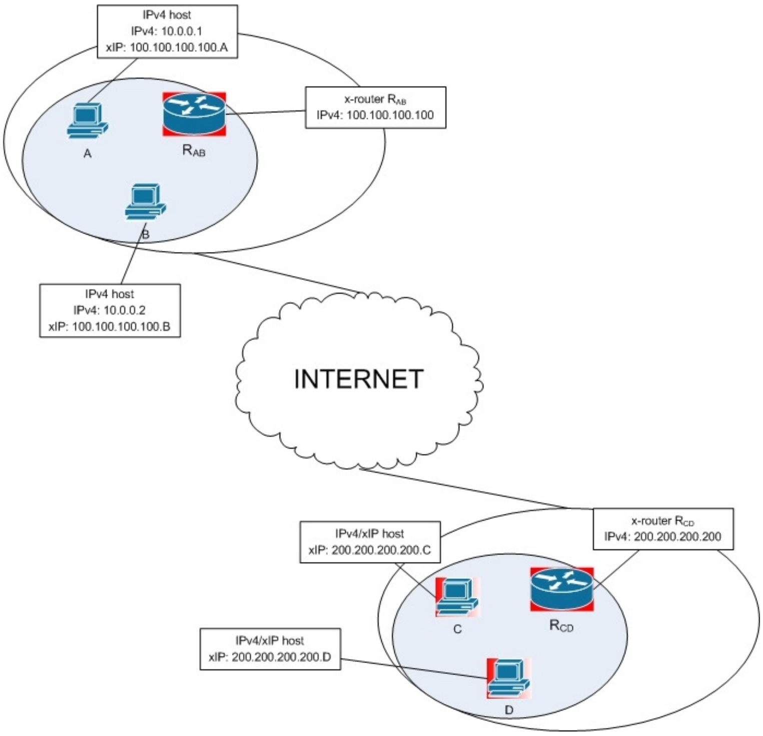


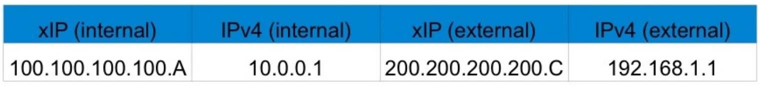
Figure 3.5: xNAT - topology

xNAT

One of the claims of xIP is to make it possible to use with legacy IPv4 host. This solution implies implementing some form of NAT box in x-router. It allows to introduce xIP without any modification of the hosts but it also brings certain limitations like necessity of address translation and end to end inconsistency. That is why in long term we expect introducing xIP hosts, however xNAT can be very useful in short term.

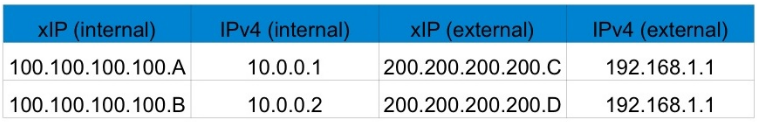
The main idea of xNAT is to spoof DNS answers for queries which are sent by legacy hosts. It is need to be noted, that legacy hosts can identify remote xIP host only with a name (FQDN) as they are unaware of new address scheme. One class of IPv4 private addresses should be reserved only for DNS spoofing. These addresses cannot be used by an host in the local site. xNAT box manages this class of addresses and dynamically associate IPv4 address with remote extended address. This mapping needs to be unique only for the single IPv4 host. It means, that the same IPv4 address can be reused for representing different extended addresses for different IPv4 hosts. As a result, this solution is very scalable, as a number of addresses which are necessary for making mappings does not depend on number of the hosts in the network but only on the number of distinct destinations which single host communicates to. xNAT translates only addresses and does not translate transport header. Additionally, one mapping can be used for many different connections with the same source-destination doublet, as only addresses are translated and transport header is not inspected.

Lets assume that we have two different sites (Fig. 3.5). The first one contains IPv4 hosts A (IPv4 address 10.0.0.1 and x-address 100.100.100.100.A) and B (IPv4 address 10.0.0.2



xIP (internal)	IPv4 (internal)	xIP (external)	IPv4 (external)
100.100.100.100.A	10.0.0.1	200.200.200.200.C	192.168.1.1

Figure 3.6: xNAT table 1



xIP (internal)	IPv4 (internal)	xIP (external)	IPv4 (external)
100.100.100.100.A	10.0.0.1	200.200.200.200.C	192.168.1.1
100.100.100.100.B	10.0.0.2	200.200.200.200.D	192.168.1.1

Figure 3.7: xNAT table 2

and x-address 100.100.100.100.B) which are in x-subnet with x-router R$_{ab}$ (IPv4 address 100.100.100.100). Second subnet contains hosts C (x-address 200.200.200.200.C) and D (x-address 200.200.200.200.D) which can be both IPv4 or xIP hosts.

When A wants to initiate connection with C, it has to send a DNS query type A with the name of C. This query needs to reach R$_{ab}$. Then, R$_{ab}$, which is aware of x-addressing, gets a full extended address of C. Next, it looks if has already proper entry in its table for a A-C doublet. If it is not the case, it chooses address from reserved class, which needs to be unique only to host A (i.e. 192.168.1.1), and creates DNS response with that address. As a result, x-router has entry as in Fig. 3.6 and IPv4 host has IPv4 address for x-destination. During the communication between A and C, R$_{ab}$ translates IPv4 addresses to x-addresses for outgoing packets and x-addresses to IPv4 addresses for incoming packets.

Second example show the situation when D wants to send packets to B. Node D gets x-address of B with DNS query (or can know x-address directly if it is xIP-aware) and sends a packet. When router R$_{ab}$ gets a packet with x-destination of B, first it checks if already has a proper entry in table for a B-D doublet. In case it does not have, R$_{ab}$ chooses address from reserved class, which needs to be unique only to host B (i.e. 192.168.1.1) and creates entry in xNAT table. As a result, R$_{ab}$ has entries as in Fig. 3.7. Similarly as above, during the communication between B and D, x-router translates IPv4 addresses to x-addresses for outgoing packets and x-addresses to IPv4 addresses for incoming packets.

3.3 xIP6

The idea of Integrated Addressing and Integrated Routing seriously simplifies network management in comparison to IPv6 approach. There is no need of duplicating the address scheme, routing topology or routing policies. Because of the fact that xIP can coexist with IPv4, operating and network troubleshooting is easier and faster than in the same network with two parallel addressing scheme (IPv4 and IPv6). Opposite to IPv6, xIP can be deployed gradually without raising special management difficulties. Consequently, it could be more acceptable by community.

However, we need to notice that IPv6 is already a mature technology ready for deployment. It is very well developed, albeit not deployed protocol. There is a strong community connected with IPv6 and some of the current research projects (for example, in mobility area) strongly bases on IPv6 features. Moreover, we already have IPv6-ready applications,

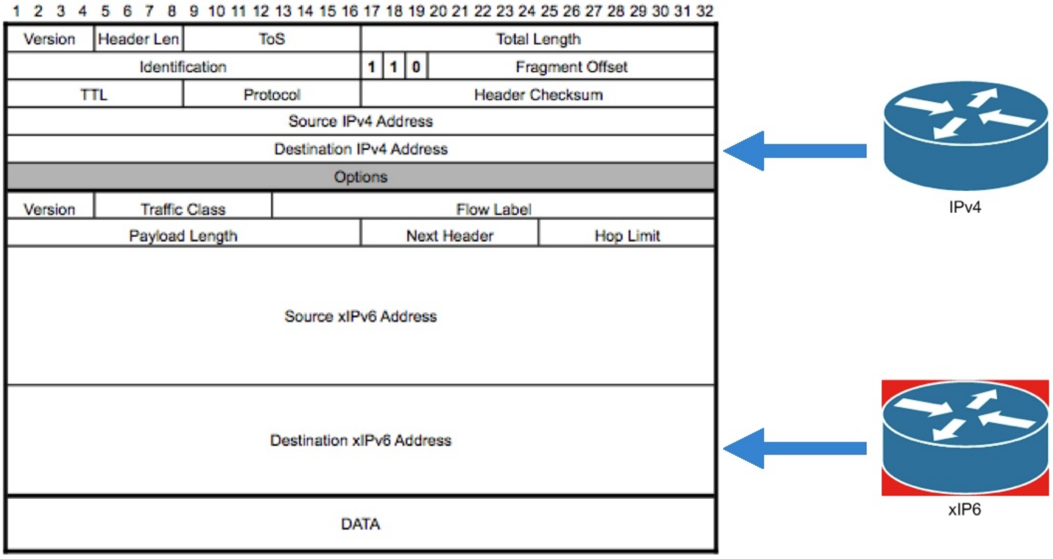


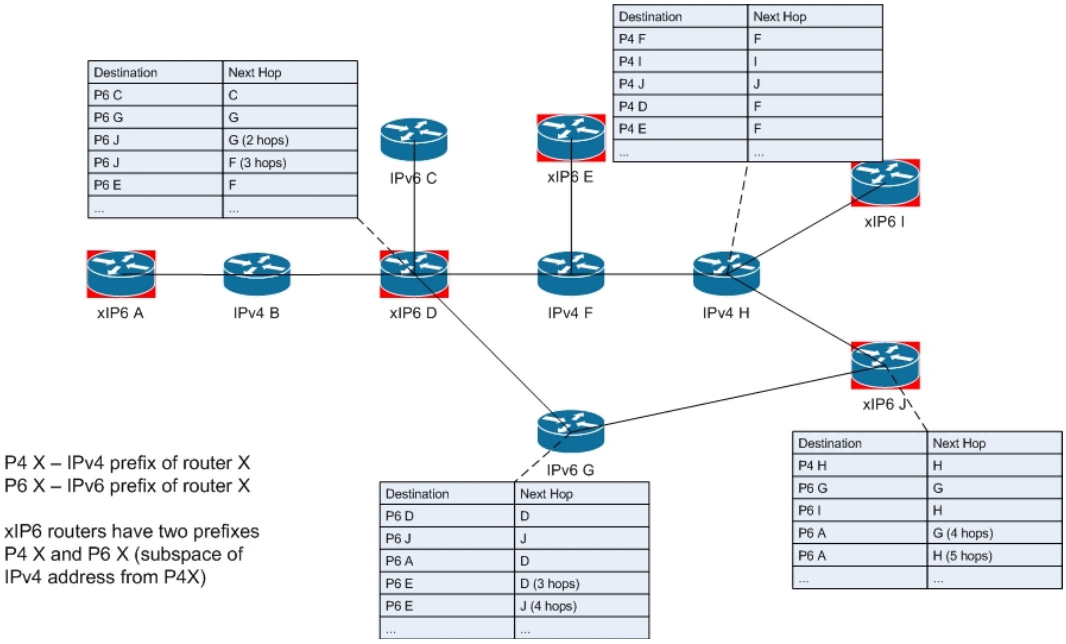
Figure 3.8: xIP6 header

IPv6 DNS record or IPv6 aware protocols like DHCP or ARP.

The idea of xIP could be used for adjusting IPv6 technology to IPv4. With this approach (xIP6), IPv6 and IPv4 addressing scheme are integrated. IPv4 unicast addresses would serve as the network part of IPv6 addresses and the remaining bits of IPv6 address are used in the same way as xIP address suffix described in previous Sections. To avoid the necessity of renumbering all IPv6 networks, it could be possible to assign IPv6 prefix for new xIP compatible IPv6 sites. Similar approach about IPv6 and IPv4 addressing is presented in 6to4 [21], but 6to4 still keeps IPv6 routing behaviour. xIP6 packet keeps both headers, IPv4 and IPv6, so packet can be routed in the core of the Internet by IPv4 infrastructure basing only on IPv4 address. Moreover, thanks to keeping IPv6 header, none of the features of IPv6 is lost and many already working IPv6 based solution can be reused. The example of xIP6 header is presented in Fig. 3.8.

xIP6 integrates also IPv4 and IPv6 routing tables. The aim is to have one consistent routing topology which we believe is more effective that two unrelated routing schemes. This approach allows to leave some part of routing behaviour in the core of the Internet unchanged. What is performed, it is still IPv4 routing based on longest prefix match, however, as with xIP6 the site will need fewer IPv4 addresses, it will announce networks with longer prefixes. It will cause that routing table will be bigger, which simply cannot be avoid when extending address space. It needs to be assured that advertisements with longer prefixes are not discarded by routers. IPv4 devices will take part in IPv6 routing but they will have knowledge only about prefixes with maximum length of /32.

Additionally, xIP6 devices can have a knowledge about prefixes longer than /32. They also have access to IPv6 header, thus they can know full IPv6 address of destination. This is in opposite to IPv4 only devices which see only first 32 bits of the address. Thanks to this, xIP6 devices can for example choose better routes or perform some other operation on packets, basing on fields from IPv6 header.



Destination	Next Hop
P6 C	C
P6 G	G
P6 J	G (2 hops)
P6 J	F (3 hops)
P6 E	F
...	...

Destination	Next Hop
P4 F	F
P4 I	I
P4 J	J
P4 D	F
P4 E	F
...	...

Destination	Next Hop
P6 D	D
P6 J	J
P6 A	D
P6 E	D (3 hops)
P6 E	J (4 hops)
...	...

Destination	Next Hop
P4 H	H
P6 G	G
P6 I	H
P6 A	G (4 hops)
P6 A	H (5 hops)
...	...

P4 X – IPv4 prefix of router X
P6 X – IPv6 prefix of router X

xIP6 routers have two prefixes
P4 X and P6 X (subspace of
IPv4 address from P4X)

Figure 3.9: xIP6 Integrated Routing

As a result, routing table of xIP6 device contains three type of addresses: IPv4, IPv6 and xIP6. IPv6 destinations can be reached by IPv6 or xIP6 routers. Similarly, IPv4 destinations are reachable by IPv4 and xIP6 routers. Next hop for xIP6 destinations can be of any type: IPv4, IPv6, or xIP6.

Sample topology with IPv4, IPv6 and xIP6 routers is presented in Fig. 3.9. It can be seen that xIP6 devices (like D or J) have knowledge about full topology including IPv4 and IPv6 routers. Thanks to this, router D keeps two routes to router J, one through IPv6 router G and the other through IPv4 routers F and H, and can route packets through any of them.

Such an approach strongly simplifies network management in comparison to pure IPv6 approach. It is not necessary to maintain two addressing and routing schemes and hence such things like routing topology and routing policies are not duplicated. It saves not only resources but also time which network administrators spend on configuring and managing. Because of one consistent scheme instead of two separated ones, troubleshooting and reacting for potential problems with xIP6 is easier and faster.

For a period of five years University of Carlos III has been in charge of REDI Madrid, the regional research network for the research and education institutions in Madrid area. Both of the protocol, IPv4 and IPv6, run in the network. Running two completely different protocols caused many maintenance problems. Based on our experience, we consider that xIP6 which integrates IPv6 and IPv4 addressing and routing could significantly simplify network management reducing operational costs in the range of 20% to 30%

3.4 xIP and Domain Name Service

Currently, to perform communication in the Internet, we need to identify the other part of the transmission with its IP address. IP address can be known in advance or can be obtained using Domain Name Service (DNS). In second case, a device knows only FQDN of destination, and it sends DNS query with this name to get IP address. We need to notice the fact, that if we want to offer xIP functionality to the legacy hosts (with usage of xNAT), first solution with using extended xIP addresses directly for identifying the remote host cannot be used. This is because legacy hosts are not aware of extended addresses and they understand only IPv4 addresses. Thus, identifying through the FQDN is the only possibility. This does not refer to xIP aware hosts.

Using DNS to obtain extended address implies introducing new DNS record and new type of query. However, in short term, before new record will be widely understood and served, some partial solution can be needed. It will allow introducing xIP in a site immediately without waiting for the support from global Internet. The easiest solution is to obtain extended address using two DNS queries. First part of xIP address, which is IPv4 address of edge router, can be obtained in the same way as today, with type A DNS record. The extended part of address can be obtained with type A (32 bits long) or type AAAA (128 bits long) query, depending on the size of extended part, which we are not defining now. The second query can carry the same domain name as first query used to get IPv4 part of the address, but preceded with some agreed upon unique string. Lack of the answer for second query, together with successful answer for the first query, would mean that destination is IPv4 only.

However, option with two DNS queries has some small disadvantages like additional delay made by two queries and potential errors if a host with the FQDN containing unique prefix exists. Therefore, introducing new DNS record is highly expected. The problem of obtaining x-address does not occur with xIP6, where we can reuse AAAA record.

Chapter 4

Deployment and Case Scenarios

In this Section we will present the deployment scheme for xIP. Next, two case scenarios when xIP could be easily used will be discussed.

4.1 Deployment

One of the reasons why deployment of IPv6 takes so much time is because it significantly raises complexity in the network. If we do not want to run two distinct protocols in the network, it is necessary to set a flag date, when all the devices and services will be turned into IPv6. If we decide to keep two protocols in the site, IPv4 and IPv6, we also need to keep two completely distinct routing topologies and addressing schemes. It obviously raises management cost and complexity. Additionally, significant part of network equipment needs to be changed. What is more, it should be done not only locally, but on whole path, which will be taken by IPv6 packets what consequently leads to change whole Internet. Meanwhile, to connect two IPv6 island through IPv4 infrastructure, tunneling must be used what raise complexity and produces overhead.

To make xIP successful, it needs to be assured that deployment scheme is easy and can be performed gradually in rather evolutionary than revolutionary way. It should be also assured that user after introducing xIP can exploit benefits immediately without waiting for support from ISP or global infrastructure. That is the reason, why it is desired that xIP packet are IPv4 compatible, so they can traverse the networks without translation mechanisms.

Introducing xIP does not have to be done in whole realm at the same moment. As it was mentioned, on the begininng it can be limited to one router per site. This router will serve as edge router of the x-network and will provide x-capability to the legacy IPv4 hosts. These hosts are not aware of their x-address, so translation of the addresses needs to be implemented in x-router (xNAT). Introducing x-routers can be done gradually to satisfy current need while full connectivity inside the realm is still assured without any translation mechanisms.

In long term, we expect x-hosts to be introduced into x-network, what will make the process of communication much easier without translation of the addresses on the way. It does not need to be done immediately but it will assure that end-to-end consistency which is violated by NAT will be restored. The new host will be able to recognise if destination is x-capable, to obtain x-address and to create x-packets. At the same time, it needs to be able to process IPv4 packets correctly, and to assure connectivity inside the realm, besides x-address it will still have private IPv4 address assigned.

4.2 Connection of Two Remote Sites

One of the case scenario when xIP can bring benefits is the example of two remote sites. It could be two offices of the same company (then both networks are under the same administrative control) or for example two universities, which cooperate doing some research. In both cases, networks contain many devices which need to communicate with each other through IPv4 network. Nowadays, we could solve it in following ways:

- Assigning public IPv4 addresses to all devices. This solution, even if the simplest, is very costly because of high consumption of IPv4 public addresses.
- Migration to IPv6. As we stated earlier this solution creates many problems, raises complexity and requires migration every host and service to IPv6. It also requires IPv6 support from ISP otherwise tunneling needs to be performed.
- VPN tunnels. This technique requires uniqueness in private addresses assigned to hosts. It is not so easy to do in case when networks are not under the same administrative control. VPN also gives connectivity only between places where it is established. Additionally it raises cost of management.

With xIP we can assure full connectivity with introducing one x-router per each site and putting all devices which need to be reachable into one x-network. The biggest benefit is that we only use one public IPv4 address per site as the devices to be reached have assigned private addresses (which can overlap between two sites) but still they remain accessible. Additionally, these devices are reachable not only by these two particular sites, but also by all xIP-aware sites.

Business model of implementing xIP has strong points, because as it is presented, xIP can bring benefits the same day it is installed in a company, what justifies expenses which are spent on it. Thus, on the beginning of transition period, even if xIP is not widely implemented in global network, the company can still exploits benefits. With introducing xIP to more sites, these benefits will be bigger as there will be more devices which could be reachable using xIP.

4.3 xNAT as NAT46

The idea of the xNAT could be successfully used for IPv4 and IPv6 coexistance improvement. xNAT could work as NAT46, and it could be used in order to allow IPv4 hosts behind NAT46 box to connect to IPv6-only devices. Such a scenarios can be not so rare in the nearest future. Because of IPv4 address shortage some of the sites will be IPv6-only, while many sites will not migrate and will stay IPv4-only. To avoid a situation when we have two poorly connected Internets, a mechanism which will assure connection between the sites is demanded.

Moreover, NAT46 makes possible to introduce IPv6 addressing to the legacy site, and makes not modified IPv4 devices globally addressable with IPv6 address. Such devices would be globally reachable by IPv6 hosts, even if they would be aware only about its IPv4 address assigned (no matter if private or public one). In this case, some benefits of IPv6 (extended address space) could be introduced into the IPv4 site, with installing just one device in the edge of the network. Such a solution can potentially speed up IPv6 deployment.

The base idea of NAT46 is very similar as one of xNAT. NAT46 is connected to IPv6 Internet and has allocated IPv6 prefix P6::/64. Local network of NAT46 box is IPv4 only. Every device in it has assigned IPv4 address (public or private one). Every IPv4 device can be additionally addressed using IPv6 globally unique address. This address is created by prepending to current IPv4 address of device (i.e. a.b.c.d) IPv6 prefix P6::/96. The newly created IPv6 address has a form of P6::a.b.c.d. Additionally, NAT46 possesses a class of IPv4 addresses P4, which needs to be locally unique inside the network. These addresses are used to represent IPv6 addresses to local IPv4 hosts.

Translation of the packets is made using SIIT mechanism [26]. Translation of the addresses is made using NAT46 table. Each entry contains a triplet: INT4 (IPv4 address of local machine), OUT6 (IPv6 address of remote machine) and OUT4 (locally unique IPv4 address from P4 class, which is used for representing OUT6 for a particular INT4). The 4th value of INT6 (IPv6 address which represents IPv4 internal device in IPv6 Internet) can be calculated by prepending P6::/96 prefix to the INT4. When NAT46 translates IPv4 packet into IPv6 packet, it also translates addresses from INT4 to INT6 and from OUT4 to OUT6 according to the entry in the table. Similarly, for every IPv6 packet translated to IPv4 packet, NAT46 also translates following addresses: INT6 to INT4 and OUT6 to OUT4.

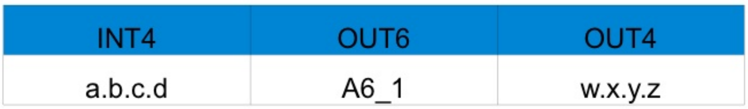
It is worth to be noted, that NAT46 is a very scalable solution. One entry can serve many connections between a pair of IPv4 and IPv6 hosts, as NAT46 translates only end-addresses from IP header, without inspecting transport header. Additionally, the size of P4 class does not depend on the size of the network only on the number of connection kept by single IPv4 host from local network with the distinct IPv6 destinations.

Detailed examples of IPv4 and IPv6 initiated connections are described below.

4.3.1 IPv4 to IPv6 connection

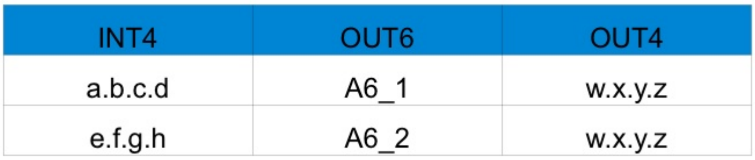
When IPv4 device (i.e. with address a.b.c.d) wants to initiate connection to remote IPv6 host (i.e. with IPv6 address A6_1), the only way to identify this host is by using Fully Qualified Domain Name (FQDN). It is because unmodified IPv4 hosts are not aware of IPv6 addressing. In this situation, IPv4 sends DNS type A query to NAT46. NAT46 tries to resolve the query, and in case it succeed, it means that destination is IPv4-aware, and no translation needs to be done. If it is not the case, then NAT46 tries with AAAA query with the same FQDN. If it succeed, it means destination is IPv6-only and NAT46 obtains IPv6 address of it (A6_1), otherwise destination does not have IPv4 neither IPv6 address assigned, and NAT46 returns notification of failure to IPv4 device.

Having IPv6 address of the destination, NAT46 checks if it has already a proper entry to do the translation. It is done by looking for the entry in which OUT6 from NAT46 table matches A6_1 and INT4 from NAT46 table matches a.b.c.d. In case there is no such an entry, NAT46 chooses IPv4 address from class P4 to represent IPv6 address A6_1 to IPv4 host (i.e.



INT4	OUT6	OUT4
a.b.c.d	A6_1	w.x.y.z

Figure 4.1: NAT46 table 1



INT4	OUT6	OUT4
a.b.c.d	A6_1	w.x.y.z
e.f.g.h	A6_2	w.x.y.z

Figure 4.2: NAT46 table 2

w.x.y.z). This address (OUT4) needs to be unique only for specific IPv4 host, and can be reuse to represent other (or the same) IPv6 addresses to other IPv4 hosts. Next, NAT46 creates entry in NAT46 table which contains following triplet: IPv4 address of source a.b.c.d (INT4), IPv6 address of destination A6_1 (OUT6), IPv4 address w.x.y.z used for mapping (OUT4). Once entry exists, NAT46 sends DNS reply to IPv4 source with IPv4 address of OUT4. This entry is presented in Fig. 4.1.

From now on, NAT46 translates outgoing packets from IPv4 to IPv6 and translates IPv4 address of source a.b.c.d (INT4) to IPv6 source address P6::a.b.c.d (INT6) and IPv4 address of destination w.x.y.z (OUT4) to IPv6 destination address A6_1 (OUT6) according to the entry it has. Every incoming packet is translated in opposite way, from IPv6 to IPv4.

4.3.2 IPv4 to IPv6 connection

Every IPv4 host behind NAT46 box can be reached with globally unique IPv6 address from P6::/64 prefix. Thus, if IPv6-only device wants to connect to such IPv4 device, it can use IPv6 address directly or it can obtain it from DNS global system. Once IPv6 device has IPv6 address of the destination, it sends a packet which is routed through IPv6 Internet and finally it reaches NAT46 box.

Lets assume IPv6 host (i.e. with address A6_2) sends a packet to IPv4 host (i.e. with address e.f.g.h) using IPv6 address (P6::e.f.g.h). When IPv6 packet comes to IPv6 external interface of NAT46, NAT46 checks if it has appropriate entry to perform translation. Similarly as it is described above, it is done by looking for the entry in which OUT6 from NAT46 table matches A6_2 and INT4 from NAT46 table matches e.f.g.h. In case there is no such an entry, NAT46 chooses IPv4 address from class P4 to represent IPv6 address to IPv4 host (i.e. w.x.y.z). This address (OUT4) needs to be unique only for specific IPv4 host, and can be reuse to represent other (or the same) IPv6 addresses to other IPv4 hosts. Then, proper entry with a triplet of INT4, OUT6 and OUT4 is put into the NAT46 table. This entry is presented in Fig. 4.2.

Since this time, NAT46 translates incoming packets from IPv6 to IPv4 and translates IPv6 address of source A6_2 (OUT6) to IPv4 source address w.x.y.z (OUT4) and IPv6 address of destination P6::e.f.g.h (INT6) to IPv4 destination address e.f.g.h (INT4). Every outgoing packet is translated in opposite way, from IPv4 to IPv6.

Chapter 5

Summary

IPv4 address shortage becomes a very serious problem and xIP (or its variant xIP6) presented in this thesis can occur as a good solution for that. xIP approach has several advantages:

- xIP packet format is IPv4 compatible and can be processed by IPv4 devices
- xIP can be used with IPv4 hosts (with using xNAT)
- with Integrated Addressing and Routing we have a single, integrated, routing table
- it implies a single integrated routing topology and routing policies which can cut operational costs
- deploying xIP can be done gradually and legacy hosts can also exploit xIP benefits

xIP can be incorporated in two different ways. Pure xIP works as an extension of widely used and widely understood technology - IPv4. xIP6 allows for reusing many aspects from IPv6, is compatible with based-on-IPv6 innovations, and can possibly facilitate IPv6 deployment.

This approach is less costly than deploying pure IPv6 and taking into account problems with IPv6 deployment and IPv4 address exhaustion, xIP can be considered as a good alternative and efficient architecture which extends current Internet address space.

References

[1] Geoff Huston. IPv4 address report. *http://www.potaroo.net/tools/ipv4/index.html*.

[2] S. Deering and R. Hinden. Internet protocol, version 6 (IPv6) specification. *RFC1883*, 1995.

[3] S. Deering and R. Hinden. Internet protocol, version 6 (IPv6). *RFC2460*, December 1998.

[4] Lorenzo Colitti. Global IPv6 statistics - measuring the current state of IPv6 for ordinary users. *RIPE 57 Meeting, Dubai*, October 2008.

[5] K. Egevang and P. Francis. The IP network address translator. *RFC1631*, May 1994.

[6] T. Hain. Architectural implications of NAT. *RFC2993*, November 2000.

[7] M. Holdrege and P. Srisuresh. Protocol complications with the IP network address translator. *RFC3027*, January 2001.

[8] B. Carpenter. Internet transparency. *RFC2775*, February 2000.

[9] P. Srisuresh and K. Egevang. Traditional IP network address translator (traditional NAT). *RFC3022*, January 2001.

[10] M. Borella, D. Grabelsky, J. Lo, and K. Taniguchi. Realm specific IP: Protocol specification. *RFC3103*, October 2001.

[11] Z. Turanyi and A. Valko. IPv4+4. *IEEE International Conference on Network Protocols*, 2002.

[12] Z. Wang. EIP: The extended internet protocol. *RFC1385*, November 1992.

[13] A. Azcorra. An option for eXtended IP general addresses (OXYGEN). *Internet Draft*, June 2005.

[14] D .R. Cheriton and M. Gitter. TRIAD: A scalable deployable nat-based internet architecture. *Stanford Computer Sciece Technical Report*, January 2000.

[15] P. Francais and R. Gummadi. IPNL: A nat-extended internet architecture. *SIGCOMM*, 2001.

[16] M. Lee. MTAII: New foundation for the next generation internet. *GLOBECOM*, 2003.

[17] Alain Durand. Deploying IPv6. *IEEE Internet Computing*, 2001.

[18] Monica Domingues, Carlos Friacas, and Pedro Veiga. Is global IPv6 deployment on track? *TERENA Networking Conference*, Denmark, 2007.

[19] Edward Lewis. Moving from IPv4 to IPv6. *ICIN 2008, Bordeaux, France*, October 2008.

[20] E. Nordmark and R. Gilligan. Basic transition mechanisms for IPv6 hosts and routers. *RFC4213*, October 2005.

[21] B. Carpenter and K. Moore. Connection of IPv6 domains via IPv4 clouds. *RFC3056*, February 2001.

[22] C. Huitema. Teredo: Tunneling IPv6 over UDP through Network Address Translators NATs. *RFC4380*, February 2006.

[23] A. Azcorra, A. Garcia, and M. Bagnulo. Internet protocol, version 64 (IPv64) specification. *Internet Draft*, April 2002.

[24] Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, and E. Lear. Address allocation for private internets. *RFC1918*, February 1996.

[25] J. Postel. Internet Protocol, Darpa Internet Program, protocol specification. *RFC791*, September 1981.

[26] E. Nordmark. Stateless IP/ICMP translation algorithm (SIIT). *RFC2765*, February 2000.