Network Working Group Karim
El Malki
INTERNET-DRAFT Gonzalo
Camarillo
Expires: December 2003
Ericsson
Jasminko
Mulahusic
Mikael Lind
TeliaSonera
Hesham
Soliman
Flarion
June
17, 2003

IPv6-IPv4 Translators in 3GPP Networks
<draft-elmalki-v6ops-3gpp-translator-00.txt>

Status of this memo

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Abstract

There have been discussions on the v6ops mailing list and at IETF meetings regarding the suitability of translators (e.g. NAT-PT) as mechanisms for IPv4 to IPv6 transition. It has often been stated that NAT-PT is not a mechanism to be recommended in general to solve the

IPv6-IPv4 transition problem and some modifications to NAT-PT have been proposed. However there have also been discussions regarding special scenarios where some form of translators could be deployed if their use is documented appropriately. The aim of this draft is to document the rationale for using translators in 3GPP networks, in particular for IPv6-only IMS (IP Multimedia Subsystem) and to describe possible solutions to the problem and the interactions with SIP.

TABLE OF CONTENTS

1. Introduction

3GPP has adopted IPv6 as its only mechanism to deploy new IP multimedia subsystem (IMS) services such as messaging or voice and video over IP. 3GPP networks have different constraints from other types of networks, therefore it is important to consider the special requirements which make translators an attractive solution for transitioning 3GPP networks. The 3GPP scenarios and analysis drafts [1][2] describe the 3GPP network and transition mechanisms which could be used in such networks. These should be used as reference together with RFC 3114 [3] when reading this document. The aim of this draft is to document the reasons why translation can be an attractive mechanism in 3GPP networks and to formulate a solution to the 3GPP IPv6-to-IPv4 translation problem. This solution considers the impacts on SIP, which is used in the IPv6-only 3GPP IMS, and aims to reuse solutions and approaches from the SIP and SIPPING WGs.

2. 3GPP Network Requirements, SIP Requirements and constraints

A 3GPP host communicates using PDP Contexts, which are layer-2 point-to-point communication channels between 3GPP hosts and the 3GPP network. Before being able to send any IP packets, a host needs to activate a PDP Context. It is during the PDP context activation that a host normally acquires an IP address. One of the special characteristics of PDP Contexts is that a PDP context can only be used to carry IPv4 or IPv6 packets but not both. The ôPDP Typeö which is requested by the 3GPP host when establishing a PDP Context will be either set to ôIPv4ö or ôIPv6ö.

The 3GPP IMS (IP Multimedia Subsystem) will be used to provide new multimedia services (e.g. messaging, video, voice, audio) to 3GPP hosts. In order to access IMS services the 3GPP host must use a PDP Context of type IPv6 (we will call this an IPv6 PDP Context from now on). The IMS is based on SIP [4].

One essential requirement in 3GPP networks is that 3GPP hosts using IMS applications over IPv6 must be able to communicate with non-3GPP IPv4 hosts (e.g. on Internet) that use SIP applications. In order to achieve this, some kind of translation must be available between 3GPP network realms and the Internet.

Another important requirement is to minimize the number of active PDP

Contexts a host has on any given time. A reason for this is that there are practical constraints on the number of PDP Contexts which a 3GPP host may establish. If a host uses many PDP Contexts it consumes extra resources in the 3GPP network. That is because each PDP Context requires a state to be maintained in the 3GPP network. In addition, each PDP Context would normally require radio signaling and a new radio channel to be established to the 3GPP host. Therefore each additional PDP Context also consumes extra radio resources required to establish the radio channel. For these reasons, any transition solution should support the case where a 3GPP host utilises only one IPv6 PDP Context, without the need to activate additional IPv4 PDP Contexts.

As specified in [4] SIP messages may be end-to-end integrity protected, therefore it may not possible to modify them en-route. In general the SIP WG discourages the use of intermediaries which alter the contents of SIP messages. This is a very important consideration for a 3GPP Translator solution.

Also, it is preferred to limit impacts to the installed IPv4 user agent base and aim for a solution where most of the changes are made to the 3GPP user agent and IMS. That is because it will obviously be harder to require changes to SIP user agents on Internet than to require new functionality in 3GPP user agents which still have to be deployed.

3. Analysis of current SIP solutions for IPv6/v4 transition

A complete solution for IPv6/v4 transition needs to handle both the SIP layer and the media layer (e.g. RTP). Vanilla SIP can handle heterogeneous IPv6/v4 networks at the SIP layer as long as proxies are properly configured. However, end-points using different address spaces need to implement extensions in order to exchange media between them. These extensions affect the session description protocol in use (e.g. SDP) and the SIP offer/answer state machine.

3.1 SIP Layer

A SIP user agent is typically reachable through the SIP server that handles its domain. If the publicly available SIP URI for a particular user is sip:user@example.com, requests sent to that user will be routed to the SIP server at example.com. The proxy or user agent sending the request will perform a DNS lookup for example.com in order to obtain the IP address of the SIP server. Therefore, if the SIP server of a domain is a dual-stack proxy that supports IPv4 and IPv6, it will be able to receive requests from IPv4-only and from IPv6-only hosts. Then, the SIP server will relay the request to the user agent using the address provided by the user agent at registration time (which could be IPv4 or IPv6).

The SIP server that receives a request using IPv6 and relays it to the user agent using IPv4, or vice versa, needs to remain in the path traversed by subsequent requests between both user agents. Therefore, such a SIP server should always be configured to Record-Route in that situation.

3.2 Media Layer

SIP establishes media sessions using the offer/answer model [5]. One end-point, the offerer, sends a session description (the offer) to the other end-point, the answerer. The offer contains all the media parameters needed to exchange media with the offerer; codecs, transport addresses, protocols to transfer media, etc.

When the answerer receives an offer, it elaborates an answer and sends it back to the offerer. The answer contains the media parameters that the answerer is willing to use for that particular session. Offer and answer are written using a session description protocol. The most widespread session description protocol at present is SDP [6] and 3GPP IMS uses SDP, thus we will focus on it. Session descriptions are transmitted end-to-end and are not modified by proxies. In this document we sometimes use SIP INVITEs and 200 (OK)

Responses for simplicity to identify the offer and response model, but it should be noted that support for other SIP messages carrying the SDP offer/answer is implied.

Vanilla SDP only allows an end-point to provide a single IP address per media stream. However, using the ALT extension [7] it is possible to include several IP addresses in the description of a media stream. Using ALT, an offerer can provide, for instance, an IPv4 and an IPv6 address for a particular media stream. The answerer will choose the address of the type it supports or prefers.

An end-point can use several mechanisms to obtain the different addresses to be placed in its ALT group in its session description. It can be a dual-stack host that configures IPv4 and IPv6 addresses or it can use protocols like TURN [8], RSIP [9], STUN [10] or TEREDO [11] to discover extra IP addresses which it is reachable at.

ICE [12] describes how to couple address discovery procedures with the offer/answer model. ICE is useful when the user agents are in different private addresses spaces, where more than one offer/answer exchange is needed to discover a reachable address for the peer.

4. IPv4/v6 Transition Solution for IMS

As mentioned previously, one important requirement for 3GPP networks is that 3GPP hosts running SIP-based IMS applications over IPv6 must be able to communicate with IPv4 SIP hosts on the Internet. This requires the following to be performed at the borders of the

3GPP
network:

1. Ensure that the IP addresses in SDP offers/answers are of
the
appropriate type for a communication to proceed.

2. Enable media communication by performing IP address and
port
mapping of the media traffic (e.g. RTP/UDP) exchanged
between the
IPv6 IMS user agent and the non-3GPP IPv4 user agent.

3. Ensure that IP version 4 is used for transport of SIP
messages
between the IMS domain and external IPv4 domains.

IMS user agents need a means to obtain a public IPv4 address
plus a
port number to place in their session descriptions in order
to
receive media and an IPv6 address plus port number to send
media to.
For incoming (to IMS) media packets, the public destination
IPv4
address plus port number will be mapped to the 3GPP user
agentÆs own
IPv6 address plus port number at the edge of the 3GPP
network. For
outgoing (from IMS) media packets, the destination IPv6
address plus
port number will be mapped to the public IPv4 address plus
port
number of the non-3GPP IPv4 user agent.

A solution to these problems is given in the following
sections.

4.1 Reference Architecture for the solution

We introduce two network elements: the SIP Edge Proxy and the
IP
Address and Port Mapper (IPAPM). The reference architecture
is shown
in Figure 1.

```
                                                -------    ------
                                               |  IMS  |  | SIP  |
                                          IPv6 |  SIP  |  | Edge |
--------
                                            ---| proxy |  | Proxy|
IPv4|        |
       ------               ------   /   | (CSCF)|--|
|-----|        |
      |      |             |      | /     -------    ------
|        |
      | 3GPP |             | GGSN |/            IPv6  |
|        |
      | IPv6 |=============|      |\                  |         |
IPv4  |
      | host |  IPv6-only  |      | \             -------        |
Net   |
      |      | PDP Context |      |  \  IPv6     |       | IPv4
|        |
       ------               ------    -----------| IPAPM
|-------|        |
                                                 |       |
|        |
                                                  -------
--------
```

Figure 1 - SIP Edge Proxy and IP Address/Port Mapper (IPAPM)
in the
3GPP Network

We will refer to ôIncomingö SIP messages as IPv4 messages
going from
an IPv4 host towards the SIP Edge Proxy, while ôOutgoingö
messages
are from the SIP Edge Proxy towards the IPv4 host.

Note that a user agent on the IPv4 network (Internet) may support receiving and transmitting media over both IPv4 and IPv6 (dual-stack) or only over IPv4. This is independent of whether the user agent is using dual-stack or IPv4-only SIP proxies and registrars. Therefore an intermediate node cannot deduce the media IP-type capability of a user agent from these characteristics.

4.1.1 SIP Edge Proxy

The SIP Edge Proxy will naturally be a dual-stack node with both IPv6 and public IPv4 addresses configured on its interfaces. It will perform Record-Routing, as described in Section 3.1 and will be in the path of all the requests coming from and going to the IPv4 network.

The SIP Edge Proxy must store and manage a local pool of IPv6 and public IPv4 addresses which have been previously configured on the interfaces of an IPAPM node. The SIP Edge Proxy may have multiple IPv6/v4 address pools each belonging to different physical IPAPM nodes. This would enable the SIP Edge Proxy to perform load sharing or utilise IPAPMs which are best placed for the communication (e.g. by comparing IP addresses).

Note that the SIP Edge Proxy is a logical entity which may be implemented as a part of other SIP proxies. The IPv4 DNS records for the domain will point to the SIP Edge Proxy and all the outgoing requests with an IPv4 address as the SIP next-hop will be routed to it. Since in the 3GPP model it is the S-CSCF proxy which receives all incoming SIP messages to the IMS domain, the SIP Edge Proxy could be integrated in that node.

4.1.2 IP Address and Port Mapper (IPAPM)

The IPAPM (IP Address and Port Mapper) is needed because the 3GPP IPv6-only host and the IPv4-only host cannot send media traffic to each other due to IP layer incompatibility. The IPAPM will simply perform the IP address mapping for the appropriate IP address, port, protocol tuples on both incoming and outgoing media packets. The SIP Edge Proxy will install and delete this bidirectional state in the IPAPM (see 4.4). It should be noted that the IPAPM operation is similar to that of a bidirectional NA(P)T-PT [16] after having installed state for a particular connection. That is, the translation algorithm (SIIT) is the same, the main difference is the method used to install state in the translator. Hence, if needed, an IPAPM may also operate as a normal NA(P)T-PT for other (non-IMS) traffic for which it does not have an address/port binding.

4.2 IMS Generated INVITE

When a 3GPP user agent sends an SDP offer (e.g. INVITE) to an Internet user agent with only IPv6 addresses in the SDP, the Internet user may be dual-stack (in which case there should be no address incompatibility problem) or it may be IPv4-only. If it is IPv4-only, the 3GPP user agent will get a final error response back. This final error response will typically be a 488 (Not acceptable here) response with a warning header with warn code 300 (Incompatible network protocol) .

This response will traverse the SIP Edge Proxy, which will locally assign a public IPv4 address and port number to the IPv6 3GPP user agent for this session (Call-id, To tag, From tag) from a local pool of addresses/ports. The unique address/port combination should stay allocated to the same 3GPP IPv6 user agent for the duration of the SIP session. The SIP Edge Proxy must install this mapping state information in the IPAPM when it also obtains the 3GPP userÆs IPv6 address (in the successive SDP offer, see below).

The SIP Edge Proxy should add the assigned IPv4 media address and port assigned to the 3GPP user agent to the 488 (Not Acceptable Here) response. Note that the SIP Edge Proxy should not modify the contents of SDP, but append the IPv4 media address to the message. This is in line with what is described in [13], which recommends against SDP editing and puts requirements to achieve the same goal using a better solution. Therefore this work in the SIPPING WG addresses our problem and its completion should be encouraged. The SIP Edge Proxy utilises such a mechanism to append the assigned IPv4 media address and port to the response. The IPv4 address must be public. The 3GPP user agent will, upon reception of this response, generate a new SDP offer that contains both the IPv4 and the original IPv6 addresses and uses ALT [7]. This SDP offer will traverse the SIP Edge Proxy. Therefore we are effectively adding a requirement to [13] that the solution should allow proxies to request the use of certain IP addresses and ports in SDP offers and answers. The SIP Edge Proxy can now install a bidirectional mapping in the IPAPM between the 3GPP userÆs IPv6 media address/port and the assigned public IPv4 address/port for the session.

When the IPv4-only user agent sends back a SDP answer containing at least a public IPv4 address/port pair, the SIP Edge Proxy locally assigns an IPv6 address and port to the IPv4 user agent from a local pool of addresses/ports. The unique address/port combination should stay allocated to the same IPv4 user agent for the duration of the SIP session. The SIP Edge Proxy must install this bidirectional mapping state information in the IPAPM. Then the SIP Edge Proxy appends this IPv6 address plus port number to the SDP answer. As mentioned previously, SDP editing should be avoided and a solution satisfying the requirements in [13] should be used. This IPv6 address and port will be used by the 3GPP IPv6-only user agent to send media to the IPv4 user agent. The IPAPM will map this IPv6 address/ port pair to the IPv4 address contained in the SDP answer. Media can now flow in both directions through the IPAPM. In this paragraph we have effectively added a requirement to [13] that the solution should allow proxies to request the use of certain IP addresses and ports as destination of the media flows.

4.3 Internet Generated INVITE

In order to limit the impact on IPv4 user agents on Internet, the SIP Edge Proxy will perform a different procedure in the case of SDP offers (e.g. INVITE) sent by IPv4 user agents with at least a public IPv4 address in their session descriptions.

Upon receiving this offer, the SIP Edge Proxy will parse the SDP and establish that the IPv4 user agent does not currently have IPv6 addresses but has at least one public IPv4 address. The SIP Edge Proxy should then locally assign an IPv6 address plus port to the IPv4 user agent for this session. At this point the SIP Edge Proxy has enough information to install a bidirectional mapping in the IPAPM between the IPv4 user agentÆs public IPv4 media address/port and the IPv6 address/port assigned to it for the session. It will also allocate an IPv4 public address/port to the 3GPP IPv6 user agent, even though it cannot establish the binding until it obtains the 3GPP IPv6 user agentÆs media address in the SDP answer. The SIP Edge Proxy should then append the IPv6 media address and port, assigned to the IPv4 user agent, and the IPv4 media address and port, assigned to the 3GPP IPv6 user agent, to the SDP offer (e.g. INVITE). To achieve this it will not do SIP editing but will use the mechanism already described in 4.2 in relation to [13].

The 3GPP IPv6-only user agent will receive the SDP offer (e.g. INVITE) and process the appended IPv6 and IPv4 address/port pairs. The 3GPP user agent will use the appended IPv6 address/port to send media to the IPv4 user agent. It will then send the SDP answer (e.g. 200 OK). The SDP answer will contain both its newly assigned IPv4 address/port (appended to the offer) and its IPv6 address/ port and uses ALT [7].

The SDP answer will traverse the SIP Edge Proxy. At this point the SIP Edge Proxy can install the bidirectional mapping state in the IPAPM between the 3GPP user agentÆs IPv6 address and the public IPv4 address/port it was locally assigned earlier (which is also contained in the SDP answer itself). The IPv4 user agent will use the public IPv4 address and port in the SDP answer to send media to the 3GPP IPv6 user agent. Media can now flow in both directions through the IPAPM.

4.4 IPAPM Operation and State Installation

The installation of state in the IPAPM is intimately coupled with the generation of session descriptions (offers and answers) by the user agent.

For incoming media packets (arriving at the IPAPMÆs IPv4 interface), the IPAPM should modify source and destination address and port pairs as follows. The IPAPM should make an address/port mapping for packets having the public IPv4 source address plus port number that the IPv4 user agent placed in its session descriptions. The IPAPM should map the source address/port of these IPv4 packets to the IPv6 source address plus port number assigned by the SIP Edge Proxy to the IPv4 user agent for this session. The IPAPM must also look for packets having the public IPv4 destination/port address corresponding to that assigned to the IPv6 user agent by the SIP Edge Proxy. These must be mapped to the IPv6 address/port pair contained in the session description sent by the IPv6 user agent.

For outgoing media packets (arriving at the IPAPMÆs IPv6 interface), the IPAPM should modify source and destination address and port pairs as follows. Packets having the IPv6 source address plus port number that the 3GPP user agent placed in its session descriptions, must be mapped to the IPv4 source address and port assigned by the SIP Edge

Proxy to the 3GPP user agent for this session. The IPAPM must also look for packets having the IPv6 destination/port address corresponding to that assigned to the IPv4 user agent by the SIP Edge Proxy. These must be mapped to the IPv4 address/port pair contained in the session description sent by the IPv4 user agent

Note that the protocol used for communicating the address/port mapping information from SIP Edge Proxy to the IPAPM is beyond the scope of this document. Two alternatives are MEGACO [14] and the MIDCOM protocol being developed [15].

4.5 Private Addressing in IPv4 User Agent

The procedures described above work fine when the IPv4 user agent has a public IPv4 address and provides it in its session description. However, many IPv4 user agents are behind NATs. Therefore it is necessary for them to discover the public IPv4 address/port which they get assigned by the NAT, to be able to use it in end-to-end SDP messages.

To resolve this situation the 3GPP IMS user agent may choose to use ICE when communicating with user agents from different domains than its own. The 3GPP user agent would add ôa=stunö lines to its media lines grouped by ALT, as described in [7] and would run STUN servers on those transport addresses. The IPv4 user agent would be able to discover public addresses for itself by communicating with these STUN servers. Using ICE and STUN this way allows user agents to discover new addresses which allow connectivity to the SIP peer, as described in [12]. This mechanism does not require introduction of new servers in IMS, but requires support in the 3GPP user agent and in the IPv4 user agent as described in the sections below.

It is possible to mandate ICE implementation in 3GPP user agents, but support of ICE/STUN in IPv4 user agents is necessary to make this communication work. Since at the current time it is uncertain whether IPv4 user agents on the Internet will support ICE/STUN, it is not possible to guarantee that this procedure will work. Should this procedure fail then the user agents will know that communication is not possible.

We assume that the IPv4 user agent utilizes a SIP Proxy which has one or more public IPv4 addresses. Therefore this proxy can communicate with the SIP Edge Proxy at the edge of the IMS domain which also has at least one public IPv4 address.

4.5.1 IMS Generated INVITE

As described previously, this solution is based on the ICE mechanism [12]. In this case the 3GPP user agent sends an INVITE to the IPv4 user agent. The IPv4 user agent happens to have only IPv4 private addresses. As described previously (see 4.2), this results in an Error response from the IPv4 user agent. The SIP Edge Proxy then locally assigns an IPv4 address/port to the 3GPP user agent, gets ready to install state in the IPAPM and appends this address/port to the Error Response. The 3GPP user agent then generates a new INVITE and uses the procedure described in ICE [12]. In particular it should start STUN servers on the IPv6 addresses it will use in its offer. The 3GPP user agent then sends the offer containing ôa=stunö lines to its media lines grouped by ALT [7]. One of the media addresses must be the public IPv4 address which the SIP Edge Proxy appended to the previous Error Response. The SIP Edge Proxy now has all the information to install bidirectional state in the IPAPM for the 3GPP user agent.

The IPv4 user agent (assuming it supports ICE) runs the ICE procedure upon receiving the offer (INVITE) from the 3GPP user agent. In this way it discovers at least one public IPv4 address/port pair for itself and uses this in its SDP answer. The procedure then follows as described in 4.2. Note that it is not strictly necessary that the 3GPP user agent runs STUN after receiving the response since it does not need to discover new addresses for the communication.

If ICE is not supported by the IPv4 user agent then the communication will ultimately fail. The IPv4 user agent will return only private IPv4 addresses in its SDP answer. The response will traverse the SIP Edge Proxy which will not be able to allocate IPv6 address/port pairs mapped to private IPv4 addresses. The 3GPP user agent will receive the response, will return an ACK and will immediately send a BYE message to terminate the call since it cannot accept the private IPv4 address in the SDP response.

4.5.2 Internet (private IPv4) Generated INVITE

As described previously, this solution is based on the ICE mechanism [12]. The IPv4 user agent (which only has private IPv4 addresses) sends an SDP offer (e.g. INVITE) to the 3GPP IPv6-only user agent utilising private addresses. It would add ôa=stunö lines to its media lines and would run STUN servers on those transport addresses. The SDP offer will then traverse the SIP Edge Proxy. The SIP Edge Proxy is unable to make the local assignment of an IPv6 address/ port pair to the IPv4 user agent (see 4.3) because of the private IPv4 addressing in the SDP offer. However it is able to make a local assignment of an IPv4 public address/port to the 3GPP IPv6 user agent for this session, and will append this address/port to the SDP offer. The mechanism to append this information to the SDP offer is described in 4.2.

When the 3GPP user agent receives the SDP offer it will send back an SDP answer (e.g. 200 OK) to allow the STUN procedure to proceed (i.e. it can see that the offerer is using STUN). The SDP answer will contain the newly assigned public IPv4 address/port (previously appended to the SDP offer by the SIP Edge Proxy) and its IPv6 media address/ports. The 3GPP user should add ôa=stunö lines to its media lines and run STUN servers on those media addresses (i.e. excluding the IPv4 address since it is an IPv6-only host).

The SDP answer will traverse the SIP Edge Proxy. The SIP Edge Proxy will now be able to install the bidirectional mapping in the IPAPM between the 3GPP user agentÆs IPv6 media address in the SDP answer and the public IPv4 address which it locally assigned previously (this IPv4 address is also contained in the SDP answer).

When the IPv4 user agent receives the SDP answer, it will run STUN towards the public IPv4 address supplied by the 3GPP user agent in SDP as described in [12]. This will allow it to check connectivity to the IPv4 address in the answer and learn about public IPv4 addresses which it is reachable at.

At this point the IPAPM will not have a binding which allows it to map the IPv4 user agentÆs incoming STUN requests from IPv4 to IPv6. Therefore the IPAPM must be STUN-aware to allow this procedure to succeed. The IPAPM must locally maintain a separate pool of IPv6 addresses configured on its interface which are not handled by any SIP Edge Proxy. Upon receiving an incoming STUN request it must create a local bidirectional binding which maps the source address of the IPv4 user agentÆs STUN request to an IPv6 address allocated from its local pool. The STUN Request will therefore reach the 3GPP user agent having the IPAPM-allocated IPv6 address as its source address. The IPAPM mapping established previously will allow the subsequent STUN response to traverse the IPAPM and reach the IPv4 user agent.

Once it has found new public IPv4 addresses which allow connectivity to the 3GPP user agent, the IPv4 user agent should issue a new offer (e.g. re-INVITE or UPDATE) to pass the newly discovered public IPv4 address to the callee. Now that the IPv4 user agent has at least a public address/port pair it can complete the procedure successfully as described in 4.3.

If the IPv4 user agent does not support ICE, the communication would fail. One alternative could be to deploy servers (e.g. STUN) on the edge of the 3GPP network which IPv4 user agents could utilise to discover public IPv4 address/ports which they can use in SDP offers.

4.6 Examples

TO BE DONE

5. Application proxies and NAT-PT for non-IMS services

As mentioned previously, a 3GPP host should be able to access IMS-applications over a single IPv6 PDP Context. In addition to this it would be preferable if the same IPv6 PDP context could be used for other applications than IMS specific (e.g. web, email etc.).

The alternative to using a translator is to activate a new IPv4 PDP context. This new session over the new IPv4 PDP Context will, in most cases, be using private addresses since most 3GPP operators don't have and cannot get enough IPv4 addresses. This will mean that the choice is between an IPv4 session with NAT or an IPv6 session with a protocol translator. Using a protocol translator might have some drawbacks in comparison to using IPv4 and NAT since NATs are more mature and widely deployed.

However, looking from a 3GPP perspective, using protocol translation might be more advantageous than NAT as it will eliminate the need for creation of additional IPv4 PDP Contexts, which is a big advantage.
This makes protocol translators a viable alternative in 3GPP networks.

Using a protocol translator is not the only alternative to get rid of the extra PDP contexts when communicating with an IPv4 host. Instead, tunneling of IPv4 over IPv6 can be used. This approach has the same benefits as the translator (i.e. no additional PDP contexts need be created) but it has another drawback, which is extra overhead. Since the 3GPP network is a wireless network with limited bandwidth, increased overhead is quite an issue and has to be avoided in the largest extent possible. Note that this would not be an issue if IPv4 in IPv6 header compression is used. Another drawback is that it would require the 3GPP host to obtain an IPv4 address through some means and it is not certain that all 3GPP networks will have the appropriate infrastructure (e.g. DHCPv6 server since only stateless address configuration is mandated in 3GPP). Normally this IPv4 address would be a private address, therefore the traffic would have to be both tunneled and passed through a (IPv4) NAT. The alternative would be to hand out public IPv4 addresses. Even if an operator had enough IPv4 public addresses to share between its subscribers, it is not clear how this can be done efficiently. Normally the UE would be assigned an IPv4 address when it established a PDP context and this address is not changed for the lifetime of the PDP context (which can be many hours). Hence, to share addresses efficiently, UEs will need to know that their IPv4 address is no longer needed and terminate the PDP context if appropriate. When the address is needed again the UE will need to re-establish the PDP context. Clearly this process will add significant delays and will be inefficient over the radio interface.

To minimize the use of translation, application specific proxies can be used. Currently deployed 3GPP networks already contain application proxies therefore it should not be a complicated matter to make them dual-stack so that they are able to allow IPv6 hosts to access IPv4 servers. However it is not possible to exclude that 3GPP IPv6 hosts will use non-IMS applications for which there are no application proxies. This will not be a large amount of traffic, however such communication must be supported. A protocol translator (NAT-PT [16]) can be used for this purpose.

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. Contributors

Gabor Bajko (Nokia) has contributed to this work.

9. Acknowledgements

TBD

10. Author's Addresses

Karim El Malki
Ericsson AB
LM Ericssons Vag. 8
126 25 Stockholm
Sweden
Phone: +46 8 7195803
E-mail: Karim.El-Malki@ericsson.com

Gonzalo Camarillo
Ericsson
Advanced Signalling Research Lab.
FIN-02420 Jorvas
Finland
E-mail: Gonzalo.Camarillo@ericsson.com

Mikael Lind
TeliaSonera
Vitsandsgatan 9B
SE-12386 Farsta
Sweden
E-mail: mikael.lind@teliasonera.com

Jasminko Mulahusic

TeliaSonera
Vitsandsgatan 9B
SE-12386 Farsta
Sweden
E-mail: jasminko.mulahusic@teliasonera.com

Hesham Soliman
Flarion
E-mail: H.Soliman@flarion.com

11. References

[1] Soininen, J. (Ed.), et al., "Transition Scenarios for 3GPP Networks", draft-ietf-v6ops-3GPP-scenario-03 (work in progress), March 2003.

[2] Wiljakka, J. (Ed.), et al., "Analysis on Ipv6 Transition in 3GPP Networks", draft-ietf-v6ops-analysis-04 (work in progress), June 2003.

[3] Wasserman, M. (Ed.), et al., öRecommendations for IPv6 in Third Generation Partnership Project (3GPP) Standardsö, RFC 3314, September 2002.

[4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E., "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[5] Rosenberg, J., Schulzrinne, H., ôAn Offer/Answer Model with the Session Description Protocol (SDP)ö, RFC 3264, June 2002.

[6] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[7] Camarillo, G. , Rosenberg, J., ôThe Alternative Semantics for the Session Description Protocol Grouping Frameworkö, draft camarillo-mmusic-alt-01 (work in progress), June 2003.

[8] Rosenberg, J., Weinberger, J., Mahy, R., Huitema, C., "Traversal Using Relay NAT (TURN)", draft-rosenberg-midcom turn-01 (work in progress), March 2003.

[9] Borella, M., Lo, J., Grabelsky, D. and G. Montenegro, "Realm
Specific IP: Framework", RFC 3102, October 2001.

[10] Rosenberg, J., Weinberger, J., Huitema, C., Mahy, R., "STUN -
Simple Traversal of User Datagram Protocol (UDP) Through
Network Address Translators (NATs)", RFC 3489, March 2003.

[11] Huitema, C., "Teredo: Tunneling IPv6 over UDP through NATs",
draft-huitema-v6ops-teredo-00 (work in progress), June 2003.

[12] Rosenberg, J., ôInteractive Connectivity Establishment (ICE): A
Methodology for Network Address Translator (NAT) Traversal for

the Session Initiation Protocol (SIP)ö, draft-rosenberg-sipping-ice-00 (work in progress), February 2003.

[13] Rosenberg, J., ôRequirements for Session Policy for the Session Initiation Protocol (SIP)ö, draft-ietf-sipping-session-policy-req-00 (work in progress), June 2003.

[14] Groves, C., Pantaleo, M., Anderson, T., Taylor, T., ôGateway Control Protocol Version 1ö, RFC 3525, June 2003.

[15] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., Rayhan, A., "Middlebox communication architecture and framework", RFC 3303, August 2002.

[16] Tsirtsis, G., Srisuresh, P., ôNetwork Address Translation - Protocol Translationö, RFC 2766, February 2000.

12. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided
by the
Internet Society.