Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 29, 2011

B. Sarikaya
F. Xia
Huawei USA

January 25, 2011

NAT64 for Dual Stack Mobile IPv6
draft-sarikaya-behave-mext-nat64-dsmip-02.txt

Abstract

This memo specifies modifications required to the home agent to integrate NAT64 with Mobile IP so that IPv6 only mobile nodes (MN) receiving host-based mobility management using Dual Stack Mobile IPv6 (DSMIPv6) can communicate with IPv4 only servers. The protocol is based on home agents maintaining a table similar to NAT64 and linking it to the binding cache. The changes include better keepalive management in order to preserve battery on the mobile node as well as multicast support for NAT64 integrated into the current multicast support scheme in Dual Stack Mobile IPv6 so that IPv6 only mobile nodes can receive multicast data from IPv4 only content providers.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 29, 2011.

Table of Contents

1. Introduction

With IPv4 address depletion on the horizon, many techniques are being standardized for IPv6 migration including NAT64 [I-D.ietf-behave-v6v4-xlate-stateful]. NAT64 together with DNS64 [I-D.ietf-behave-dns64] and the translation algorithm [I-D.ietf-behave-v6v4-xlate] enables IPv6-only hosts to communicate with IPv4-only servers.

NAT64 is designed for fixed hosts. When used for mobile nodes several problems occur as described in [I-D.haddad-mext-nat64-mobility-harmful]. In this document we redesign NAT64 for host based mobility protocol called Dual Stack Mobile IPv6. The design uses DNS64 as is and integrates NAT64 operation with the binding cache of Dual Stack Mobile IPv6.

The document continues in Section 3 with a set of requirements on a solution for NAT64 for Dual Stack Mobile IPv6. In Section 4 the protocol design is presented, multicast translation is explained in Section 5 while handover and route optimization cases are covered in Section 6. In Section 7 extensions to DSMIPv6 are described.

2. Terminology

This document uses the terminology defined in [RFC3775], [RFC5555], [I-D.ietf-behave-v6v4-xlate-stateful], [I-D.ietf-behave-v6v4-xlate], [RFC6052] and [I-D.ietf-behave-dns64].

3. Requirements

NAT64 has two main problems if used for the mobile nodes: the first one is related to mobility and the second one is related to NAT keepalives.

DNS64 uses the IPv6 prefix assigned to the NAT64 IPv6 interface in the domain in translating IPv4 address of the server to an IPv6 address. This prefix will be referred to as Pref64 as in [I-D.ietf-behave-v6v4-xlate-stateful]. [RFC6052] defines two types of prefixes: Well-Known Prefix or Network-Specific Prefix. If the well-known prefix of 64:FF9B::/96 is used then the mobile node would always get the same mapping wherever it moves so no problems can be anticipated. However, for various reasons this is not expected to be the case in general.

If Network-Specific Prefixes (NSP) are used problems can be anticipated especially for mobile nodes

[I-D.korhonen-behave-nat64-learn-analysis]. This happens because DNS64 server used by the mobile node may use a different NSP that NAT64 box is not configured with.

When the mobile node moves to a foreign network, the mobile node's DNS requests can be done in two ways: either mobile node tunnels it to the home agent and the home agent sends it to the DNS server in the home network or the mobile node sends it locally and it goes to the DNS server of the foreign network. The former case poses no problems as DNS64 server is synchronized with NAT64 server at the home network. The latter case poses problems because NSP in the IPv6 address synthesized by the local DNS64 is not recognized by the home NAT64 server, i.e. its interface is not configured with this NSP. In this case the mobile node's IPv6 packet may not reach the destination IPv4-only server. This is called prefix mismatch problem.

Mobile nodes in Dual Stack Mobile IPv6 initiate route optimization with the correspondent nodes when they move to a foreign network by sending first a home test init (HoTI) message to the home agent. This and subsequent messages (Care-of Test Init (CoTI), Home Test (HoT) and Care-of Test (CoT)) contain IPv6 extension headers. NAT64's translation algorithm [I-D.ietf-behave-v6v4-xlate] does not translate IPv6 extension headers. As a result, HoTI and similar messages would be rejected at the NAT64 device and the mobile node would end up receiving an ICMP message.

This fundamental restriction of IPv6-IPv4 translation is avoided in this document by an additional requirement not to initiate the route optimization with IPv4-only servers.

NAT64 is a NAT device which keeps NAT table as the NAT state. NAT state is soft state and it expires if it is not refreshed during a certain time interval. NAT devices delete existing bindings at the end of a time interval if no activity is detected during that interval. Timer values of a minimum of two and maximum of five minutes for UDP [RFC4787] and 2 hours and four minutes [RFC2663] for TCP [RFC5382] are recommended [I-D.ietf-behave-v6v4-xlate-stateful]. However, existing NAT devices are known to have non-deterministic and typically short expiration times especially for UDP-based bindings.

Outbound refresh (mobile node initiated) is necessary for allowing the client (mobile node) to keep the mapping alive. NAT keepalives are used for this purpose [RFC5245]. Mobile nodes go to sleep mode when inactive in which battery usage is minimized. However sending NAT keepalive messages for outbound refresh may drain the mobile node's battery because it has to cut short its sleep mode.

NAT keepalives should be avoided for the mobile nodes. This

requirement is met by integrating NAT64 state with binding cache that the home agent creates for the mobile node in order to keep track of its mobility and by having the home agent to refresh NAT binding with the NAT device.

While resolving issues of NAT64 related to mobility, it is desirable to keep compatibility with fixed hosts. This requirement is met by reusing DNS64 for mobile nodes as well.

The behaviour of IPv4-only or dual stack mobile nodes using host based mobility protocol Mobile IPv6 is specified in [RFC5555]. However [RFC5555] does not specify how IPv6-only mobile nodes can access IPv4-only servers. Hence this specification complements [RFC5555].

NAT64 is designed for unicast communication, the translation algorithm is defined in [I-D.ietf-behave-v6v4-xlate] does not translate multicast packets. IPv6 only hosts receiving multicast data from IPv4 only servers is not covered.

For many applications multicast communication for mobile nodes in a dual stack Mobile IPv6 environment is a requirement. This requirement is met by designing a multicast translation scheme for Dual Stack Mobile IPv6. This technique applies to any source multicast (ASM) as well as Source Specific Multicast (SSM).

4. Modifications to DSMIPv6 for NAT64

This section discusses extensions to NAT64 to support mobility. Multicast extensions are discussed next in Section 5. It is assumed that NAT64 and HA can be hosted in different machines, however it is also possible that HA and NAT64 coexist in the same node.

Mobile nodes reverse tunnel their packets to the home agent when roaming and at the home network the home agent is the default router. When forwarding packets sent by the mobile node, the home agent first checks the Source Address field of the inner header in the binding cache to find the corresponding binding cache entry for this mobile node's home address. A further check is made if the destination address' prefix matches Pref64 in the prefix table. In case of a match, IPv6-only flag in the binding cache entry for the mobile node is set if it was not set already.

If NAT64 and HA are collocated, HA creates a "NAT state" of

<MN source address, IPv6 source port> <--> <IPv4 Interface address, IPv4 source port>

To this NAT state this specification adds keepalive interval K which is used to make sure HA/NAT64 initiates NAT64 keepalives. MN does not have to shorten the time it spends in dormant state and drain its battery.

Translation into IPv4 packet takes place at the NAT64 server. If NAT64 server is collocated then the home agent translates IPv6 packet into an IPv4 packet following the algorithm presented in [I-D.ietf-behave-v6v4-xlate].

HA (collocated with NAT64 or not) keeps IPv6-only flag and Pref64 in the binding cache. This state is linked to the binding cache entry for MN. The home agent forwards IPv6 packet towards NAT64 server.

When forwarding any subsequent packets for the same session corresponding to <MN source address, source port>, HA collocated with NAT64 finds the corresponding entry in the NAT table and creates the corresponding IPv4 packet using this entry. The above procedure of new NAT64 state creation is repeated only when a new session is started by MN.

In case of collocated HA and NAT64, an incoming IPv4 packet is processed as follows: When HA receives a packet addressed to its IPv4 interface it searches the NAT table for the corresponding MN IPv6 source address and port. For example the tuple <203.0.113.1, 2000> would match the network-specific prefix (NSP) of 2001:FF00::/64 and the source port of 1500. HA creates an IPv6 packet from IPv4 packet using this information. IPv4 packet is translated into an IPv6 packet following the algorithm presented in [I-D.ietf-behave-v6v4-xlate]. Next HA fetches MN's binding cache entry and finds care-of address of MN. HA encapsulates IPv6 packet and sends it to the mobile node.

If HA and NAT64 are not collocated, NAT64 translates IPv4 packet and forwards to HA as IPv6 packet. HA, after receiving the incoming IPv6 packet to the mobile node's home network, searches its binding cache and finds care-of address of MN and encapsulates the packet and sends it to MN.

Keepalive interval is used to send NAT keepalive messages when HA is collocated with NAT64. NAT keepalive messages are ICMP Echo Request messages [RFC3519]. ICMPv6 Echo Request message MUST be encoded with a UDP header. The packet's destination address is the destination address associated with the keepalive interval. The source address is MN's home address. Keepalive interval is used to keep track of inactivity of the mobile node's session with its NAT64 host, IPv4-only server. UDP header contains the source and destination port numbers of NAT64 binding. Any ICMP Echo Request message sent from

the home agent serves as outbound refresh message for the session and any corresponding ICMP Echo Reply received serves as the inbound refresh.

ICMPv6 Echo Request message encoded in UDP header is translated into ICMPv4 Echo Request message with UDP header at NAT64 server following translation rules defined in [I-D.ietf-behave-v6v4-xlate] since the UDP header preserves the source and destination port numbers that are needed in order to match with NAT64 binding. NAT64 server also refreshes NAT64 state for this session. An ICMPv4 Echo is sent to IPv4 only server as an IPv4 packet with UDP header. IPv4 server replies with IPv4 Echo Reply which is translated into ICMPv6 Echo Reply message and received by the home agent.

Keepalive interval of K seconds controls the frequency of keepalive messages. K is a protocol constant with a default value. The default value should be less that the timeout value used by the NAT server. Because of this K can be set to the default value of 110 seconds [RFC3519].

Home agent collocated with NAT64 forwards any subsequent packets for the same session corresponding to <MN source address, source port> and refreshes the keepalive interval. Home agent does not do any inbound refresh. Home agent MUST not forward ICMPv6 Echo Reply message to MN. Incoming packets for this session do not refresh the keepalive interval since it is the interval for outbound refresh. It is up to IPv4 only server to do the inbound refreshes.

5. Multicast Translation

In this section we specify how mobile node can receive IPv4 multicast data from IPv4-only content provider based on the current multicast support scheme in Dual Stack Mobile IPv6 [RFC3775]. The reverse translation of IPv6 multicast data for IPv4-only receivers is out of scope. Multicast translation specified in this section applies to both cases of collocated HA and NAT64 as well as HA and NAT64 hosted in different machines.

IPv6-only mobile node will join IPv4 multicast group by sending MLD Membership Report message to the home agent. This message is sent in the mobile node-home agent tunnel. Mobile node will use synthesized IPv6 address of IPv4 multicast group address, e.g. a /96 prefix used for any source multicast called IPV6_TRASM_ADDRESS prefix followed by a.b.c.d, IPv4 multicast group address. IPV6_TRASM_ADDRESS prefix takes the form of FFxx::/96, it is non-SSM prefix [I-D.venaas-behave-mcast46]. Multicast router at the home agent receives this join message from the mobile node for the group

IPV6_TRASM_ADDRESS prefix:a.b.c.d.

Each home agent is assigned a unique IPV6_TRASM_ADDRESS prefix. Mobile nodes can learn this value by means out of scope with this document. With this, mobile node can easily create an IPv6 multicast address from the IPv4 group address a.b.c.d that it wants to join.

Home agent as multicast anchor checks the group address and recognizes IPV6_TRASM_ADDRESS prefix. It next checks the last 32 bits is an IPv4 multicast address in range 224/8 - 239/8. If all checks succeed, home agent joins a.b.c.d using IGMP on its IPv4 interface.

Home agent identifies the mobile node from the tunnel and adds the multicast group address to the multicast state associated with the mobile node's binding cache entry. Home agent also sets IPv6-only bit if it was not set before.

When home agent receives multicast data for the group a.b.c.d, it first obtains the IPv6 address IPV6_TRASM_ADDRESS prefix:a.b.c.d and then checks to see if at least one mobile node is subscribed to this address from the binding cache and multicast state.

Home agent will then translate IPv4 multicast data packet into an IPv6 multicast data packet. The destination address is IPv6 group address IPV6_TRASM_ADDRESS prefix:a.b.c.d and source address is home agent's IPv6 interface address. The value in Type of Service (TOS) field of IPv4 packet is copied into IPv6 Traffic Class field. IPv4 Protocol and TTL fields are copied into IPv6 Next Header and Hop Limit fields respectively. IPv4 payload is copied into IPv6 payload. UDP checksum is updated which completes the packet translation process [Thesis]. Home agent duplicates the packet for each mobile node member of this group and sends each packet tunneled to the individual mobile node separately.

Any IPv4 fragments sent by the routers must be translated into IPv6 packets with IPv6 Fragment Header. Fragmentation Offset field is copied into the corresponding field in the Fragment Header. 16-bit Identification field is copied into the low-order 16 bits of IPv6 Fragment Header Identification field. The high-order bits of the 32-bit IPv6 Fragment Header Identification field are set to zero. More Fragments (MF) flag is copied to the corresponding field in IPv6 Fragment Header [Thesis].

Multicast translation described in this section is not mobile node agnostic. Home agent gets the join message directly from the mobile node and then updates the membership database which is connected to the binding cache. Home agent has to know all members of each IPv4

group so that it can correctly duplicate the data packets and tunnel to individual mobile nodes.

Source-Specific Multicast (SSM) can also be supported similar to the Any Source Multicast (ASM) described above. In case of SSM, IPv4 multicast addresses use 232.0.0.0/8 prefix and IPv6 multicast addresses use FF3X::/96 prefix. A unique SSM prefix can be configured such as FF3E::/96. This prefix is referred to as IPV6_TRSSM_ADDRESS prefix. Since SSM translation requires a unique address for each IPv4 multicast source, an IPv6 unicast prefix must be configured to the translator to represent IPv4 sources. This prefix is prepended to IPv4 source addresses in translated packets. Also this prefix must be routed towards the translator on the IPv6 network, to enable reverse path forwarding for multicast, and to inform other PIM routers about the correct destination for PIM (S,G) Join messages [Thesis].

6. Handover, Route Optimization and Return Routability

The mobile node moves to a foreign network and sends DNS request locally and the request goes to the DNS server of the foreign network that is configured with a different Pref64. This creates a prefix mismatch problem. Mobile node gets a different synthetic AAAA RR with a different IPv6 address of the destination. MN reverse tunnels its IPv6 packet destined to IPv4-only server to the home agent.

Home agent checks the source address (mobile node's home address) of the inner header in the binding cache for any entry with IPv6-only flag set. Next destination address' prefix is checked in the binding cache. In case the prefix does not match, HA checks the prefix table for a match with the destination address' prefix. In case of a match, a new binding cache entry is added with the new Pref64. HA is responsible for routing the MN's packet with the new Pref64. The packet may take a longer path or the packet may not even reach the destination due to a non existing roaming agreement with the foreign network. If the prefix does not match, home agent forwards the packet since this packet should be going to another IPv6 destination host.

If IPv6-only flag is not set and the prefix matches then this is the first packet sent to a new IPv4-only server. Home agent processes this packet as described in Section 4.

The effect of handover on multicast translation described in Section 5 depends on how IPV6_TRASM_ADDRESS prefix is configured. Mobile node may get a different IPV6_TRASM_ADDRESS prefix locally after moving to a foreign network. Mobile node sends a join request

(Multicast Listener Discovery Report message) with a new multicast group address to the home agent in a tunnel. Home agent adds this group address to its membership database. Home agent MUST add the new IPV6_TRASM_ADDRESS prefix to the multicast prefix table. Home agent MUST set IPv6-only flag in the binding cache for this mobile node.

Route optimization (RO) in DSMIPv6 is used to avoid triangular route every packet to the corresponding node takes by enabling the mobile node to directly send the packets to the correspondent node [RFC3775]. RO is established using control signaling involving the home agent, mobile node and correspondent node. After RO is established mobile node sends its packets directly to the correspondent node. The source address of these packets is the care-of address and MN home address is included in an extension header called home address option. All RO packets involve extension headers.

Because all route optimization packets (signaling and data) contain extension headers the translation algorithm [I-D.ietf-behave-v6v4-xlate] used in NAT64 would simply ignore the data included in these headers. As a result, route optimization can not even be initiated. IPv6 only mobile nodes involved in communication with IPv4-only servers MUST NOT use route optimization. This ensures that all traffic between the mobile node and corresponding node goes through the home agent and correct IPv6-IPv4 packet translation can be conducted.

7. **Extensions to** Dual Stack Mobile IPv6

Binding cache entry contains the following new entry:

A flag indicating whether or not this mobile node is IPv6-only node and Pref64, the prefix used to route NAT64 traffic to NAT64 server.

IPv6-only flag is set after receiving the first IPv6 packet containing a synthetic IPv6 address. This flag is used to connect the binding cache with the NAT table.

Home agent is configured with a table of NAT64 prefixes, Pref64's that are supported in Dual Stack Mobile IPv6 home domain and its roaming partners. For each Pref64, home agent keeps a 32-bit suffix which is concatenated to the prefix. The resulting 96-bit value is concatenated with IPv4 address of the destination IPv4-only server to obtain the synthesized IPv6 address.

If the Well-Known Prefix is used this table contains 64:FF9B::/96.

In this case there is no associated suffix.

IPv6-only mobile nodes MUST avoid initiating return routability procedure described in Section 5.2.5 of [RFC3775]. When the home agent receives a Home Test Init message, it checks the source address (mobile node's home address) in the binding cache. If the corresponding binding cache entry has its IPv6-only flag set home agent drops the Home Test Init message.

7.1. Multicast Extensions

Multicast anchor at the home agent MUST support at least one IPV6_TRASM_ADDRESS prefix. Multicast anchor at the home agent MUST support IGMP on its IPv4 interface.

Home agent has a table of IPV6_TRASM_ADDRESS prefixes. This table normally contains a single entry, i.e. the local prefix value. It may be populated by more entries in case of handover as described in Section 6. The entries are kept as soft-state and removed after a period of no activity.

Multicast anchor at the home agent MUST support at least one IPV6_TRSSM_ADDRESS prefix. Multicast anchor at the home agent MUST support IGMPv3 on its IPv4 interface as source filtering needed for SSM is supported only by IGMPv3.

8. Protocol Constants

K 110 seconds (as defined in [RFC3519].

9. Security Considerations

For IPv4-only or dual stack mobile nodes security considerations stated in [RFC5555] apply. This document specifies procedures for MIPv6 [RFC3775] for the case of IPv6-only mobile nodes which are not covered in [RFC5555]. Security considerations for IPv4 interface of the home agent is similar to [I-D.ietf-behave-v6v4-xlate-stateful] and the considerations stated there apply.

10. IANA Considerations

TBD.

11. Acknowledgements

The authors are grateful to Marcelo Bagnulo for his comments that helped improve the document.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

[RFC3519] Levkowetz, H. and S. Vaarala, "Mobile IP Traversal of Network Address Translation (NAT) Devices", RFC 3519, April 2003.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[I-D.ietf-behave-v6v4-xlate-stateful] Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 2010.

[I-D.ietf-behave-dns64] Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers",

draft-ietf-behave-dns64-11 (work in progress), October 2010.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[I-D.ietf-behave-v6v4-xlate] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", draft-ietf-behave-v6v4-xlate-23 (work in progress), September 2010.

[RFC5555] Soliman, H., "Mobile IPv6 Support for Dual Stack Hosts and Routers", RFC 5555, June 2009.

[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

12.2. Informative references

[I-D.haddad-mext-nat64-mobility-harmful] Haddad, W. and C. Perkins, "A Note on NAT64 Interaction with Mobile IPv6", draft-haddad-mext-nat64-mobility-harmful-01 (work in progress), April 2010.

[I-D.venaas-behave-mcast46] Venaas, S., Asaeda, H., SUZUKI, S., and T. Fujisaki, "An IPv4 - IPv6 multicast translator", draft-venaas-behave-mcast46-02 (work in progress), December 2010.

[I-D.korhonen-behave-nat64-learn-analysis] Korhonen, J. and T. Savolainen, "Analysis of solution proposals for hosts to learn NAT64 prefix", draft-korhonen-behave-nat64-learn-analysis-01 (work in progress), January 2011.

[Thesis] Teemu Kiviniemi, Helsinki University of Technology, Master's Thesis, "Implementation of an IPv4 to IPv6 Multicast Translator", October 2009.

Authors' Addresses

Behcet Sarikaya
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075

Phone: +1 972-509-5599
Email: sarikaya@ieee.org

Frank Xia
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075

Phone: +1 972-509-5599
Email: xiayangsong@huawei.com