NETEXT WG
Internet-Draft
Intended status: Informational
Expires: September 9, 2010

H. Yokota
KDDI Lab
S. Gundavelli
Cisco
T. Tran
Y. Hong
ETRI
K. Leung
Cisco

March 8, 2010

Virtual Interface Support for IP Hosts
draft-yokota-netlmm-pmipv6-mn-itho-support-03.txt

Abstract

A Virtual Interface is a software semantic internal to the host operating system. This semantic is widely available in all popular operating systems and is used in various protocol implementations. The Virtual Interface support is also required on the mobile node operating in a Proxy Mobile IPv6 domain, for leveraging various mobility features such as inter-technology handoffs, multihoming and flow mobility support. This document explains the operational details of Virtual Interface construct and the inter-working with the network elements in the Proxy Mobile IPv6 domain for supporting various network-based mobility management features.

This Internet-Draft will expire on September 9, 2010.

Copyright Notice

Table of Contents

1. Introduction

Proxy Mobile IPv6 [RFC5213] is a network-based mobility protocol. Some of the key goals of the protocol include support for multihoming, inter-technology handoffs and flow mobility support. The network elements in the Proxy Mobile IPv6 domain allow the mobile node to attach to the network using multiple interfaces, or perform handoff between different interfaces of the mobile node. However, for supporting these features, the mobile node is required to be activated with specific software configuration that allows the mobile node to either perform inter-technology handoffs between different interfaces or attach to the network using multiple interfaces. This document analyses from the mobile node's perspective a specific approach that allows the mobile node to leverage these mobility features. Specifically, it explores the use of Virtual Interface support, a semantic available on all operating systems, for this purpose.

A Virtual Interface is a software semantic internal to the operating system. This semantic is widely available in all popular operating systems. Many applications such as Mobile IP client [RFC3775], IPsec VPN client [RFC4301] and L2TP client [RFC3931] all rely on this semantic for their protocol implementation and the same semantic can also be useful in this context. Specifically, the mobile node in the Proxy Mobile IPv6 domain [RFC5213] can use virtual interface configuration for leveraging multihoming, inter-technology handoffs and flow mobility features provided by the Proxy Mobile IPv6 domain. The rest of the document provides the operational and implementation details of Virtual Interface on the mobile node and the inter-working between the mobile node using virtual interface and network elements in the Proxy Mobile IPv6 domain for supporting various network-based mobility management features.

2. Requirements Language

In this document, the key words "MAY", "MUST, "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC2119].

3. Virtual Interface Operation

On most operating systems, a network interface is associated with a physical device that provides the capability for transmitting and receiving network packets. In some cases a network interface can also be implemented as a logical interface which does not feature any packet transmission or receive capabilities, but relies on other network interfaces for such capabilities. This logical interface is also known as a virtual interface.

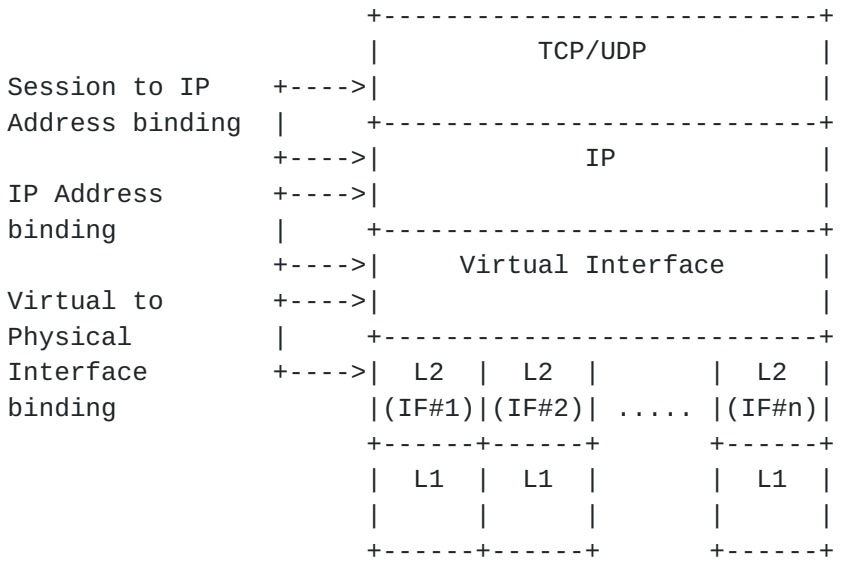

Figure 1: Virtual Interface Implementation

From the perspective of the IP stack and the applications, a virtual interface is just another interface. A host does not see any difference between a virtual and a physical interface. All interfaces are represented as software objects to which IP address configuration is bound. However, the virtual interface have some special properties which are essential for enabling various features. Following are those properties:

- Virtual interface is a logical interface that appears to the host stack as any other interface. IP address configuration can be bound to this interface by configuring one or more IPv4 and/or IPv6 addresses to this interface.

- Virtual interface has a relation to a set of physical interfaces on the host. These physical interfaces in the context of virtual interface are known as sub-interfaces. These sub-interfaces provide transmit and receive functions for sending and receiving packets over physical links. A virtual interface can receive packets sent to any of its sub-interfaces.

- The link-layer identifier of the virtual interface is used in the link-layer header of the IP packets sent through this interface, and the link-layer address of the physical interface will not be used.

- The send/receive vectors of a virtual interface are managed dynamically and are tied to the sub-interfaces. The mapping between this virtual interface and the sub-interfaces can change dynamically and this change will not be visible to the applications. The side effect of this is the ability for the application bound to the address configuration on the virtual interface, to survive across inter-technology handoffs. Applications will survive across the mapping change between a virtual interface and its sub interfaces.

- An IPv6 link as seen by the applications that the virtual interface is being part of through specific sub interface(s), when changed to be as part of through a different set of sub interface(s), will not trigger session loss, address loss, as long as the IPv6 prefix is valid, and the host continues to receives Router Advertisements [RFC4861] from the IP routers to the virtual interface over the sub-interface(s).

- The host has the path awareness of an IPv6 link, through a sub-interface and is driven by the host routing table, which uses the sub-interfaces for packet forwarding. Addresses from Prefix P1, P2 tied to the virtual interface, may have two different link paths, Prefix P1 over E0, Prefix P2 over E1, and this mapping may be reversed, without applications being aware of, and with the needed path changes on the network side.

4. Virtual Interface Use-cases in Proxy Mobile IPv6

This section explains how the virtual interface support on the mobile node can be used for enabling some of the Proxy Mobile IPv6 protocol features.

4.1. Multihoming Support

A mobile node in the Proxy Mobile IPv6 domain can potentially attach to the Proxy Mobile IPv6 domain, simultaneously through multiple interfaces. Each of the attachment links are assigned a unique set of IPv6 prefixes. If the host is configured to use virtual interface over the physical interface through which it is attached, following are the related considerations.

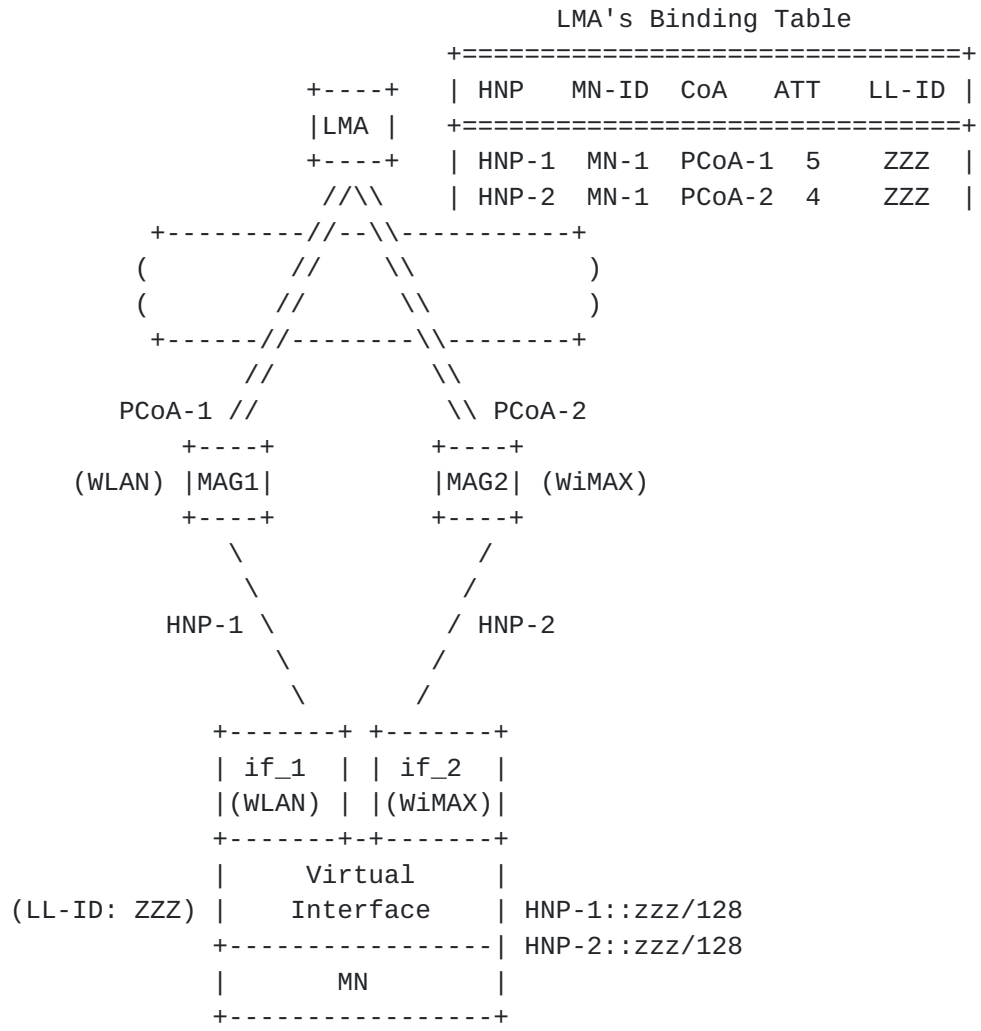

Figure 2: Multihoming Support

- The mobile node detects the advertised prefixes from the MAG1 and MAG2 as the onlink prefixes on the link to which the virtual interface is attached.

- The mobile node can generate address configuration using stateless auto configuration mode from any of those prefixes.

- The applications can be bound to any of the addresses bound to the virtual interface and that is determined based on the source address selection rules.

- The host has path awareness for the hosted prefixes based on the received Router Advertisement messages. Any packets with source address generated using HNP_1 will be routed through the interface if_1 and for packets using source address from HNP_2 will be routed through the interface if_2.

4.2. Inter-Technology Handoff Support

The Proxy Mobile IPv6 protocol enables a mobile node with multiple network interfaces to move between access technologies, but still retaining the same address configuration on its attached interface. The protocol enables a mobile node to achieve address continuity during handoffs. If the host is configured to use virtual interface over the physical interface through which it is attached, following are the related considerations.

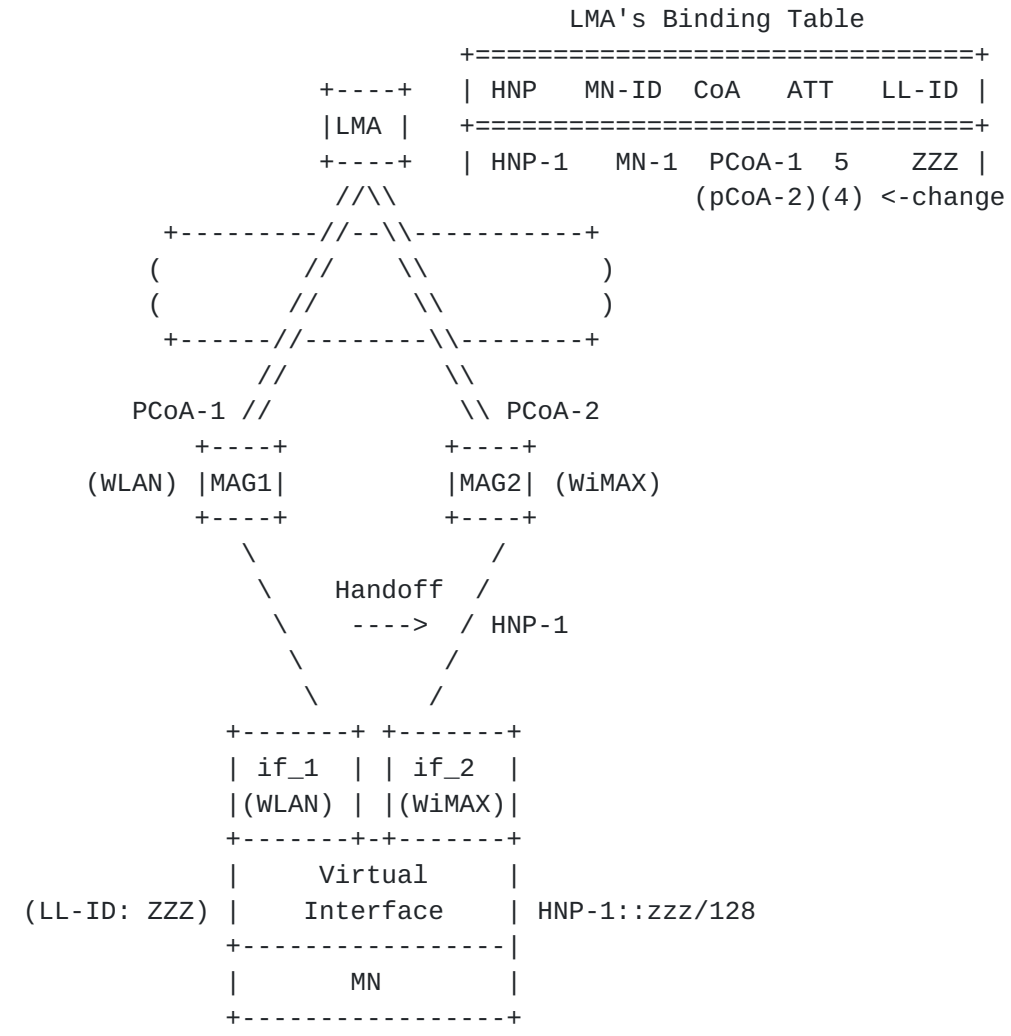

Figure 3: Inter-Technology Handoff Support

- When the mobile node performs an handoff between if_1 and if_2, the change will not be visible to the applications of the mobile node. It will continue to receive Router Advertisements from the network, but from a different sub-interface path.

- The protocol signaling between the network elements will ensure the local mobility anchor will switch the forwarding for the advertised prefix set from MAG1 to MAG2.

- The MAG2 will host the prefix on the attached link and will include the home network prefixes in the Router Advertisements that it sends on the link.

4.3. Flow Mobility Support

For supporting flow mobility support, there is a need to support vertical handoff scenarios such as transferring a subset of prefixes from one interface to another. This scenario is defined in [I-D.jeyatharan-netext-pmip-partial-handoff]. The mobile node can support this scenario by using the virtual interface support. This scenario is similar to the Inter-technology handoff scenario defined in Section 4.2, only a subset of the prefixes are moved between interfaces.

5. IANA Considerations

This specification does not require any IANA Actions.

6. Security Considerations

This specification explains the operational details of virtual interface on an IP host. The Virtual Interface implementation on the host is not visible to the network and does not require any special security considerations.

7. Acknowledgements

The authors would like to acknowledge prior discussions on this topic in NETLMM and NETEXT working groups. The authors would also like to thank Joo-Sang Youn for his comments on the document.

8. References

8.1. Normative References

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

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.

8.2. Informative References

[HP-PNAT] "HP ProLiant Network Adapter Teaming", <http://bizsupport.austin.hp.com/bc/docs/support/SupportManual/c01415139/c01415139.pdf>.

[I-D.jeyatharan-netext-pmip-partial-handoff] Jeyatharan, M., Gundavelli, S., and V. Devarapalli, "Partial Handoff Support in PMIPv6, draft-jeyatharan-netext-pmip-partial-handoff-01 (work in progress)", March 2009.

[Intel-PROSet] "Intel Advanced Networking Services With Ethernet Teaming", <http://www.intel.com/support/network/sb/cs-009747.htm>.

[Linux-Bonding-Driver] "Linux Bonding Driver", <http://www.linuxfoundation.org/collaborate/workgroups/networking/bonding>.

[Net:Bridge] "Net:Bridge", <http://www.linuxfoundation.org/en/Net:Bridge>.

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

[RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

Appendix A. Virtual Interface Implementation Guidelines

Most of the current operating systems support tools for setting up virtual interface at the mobile node such as NET:Bridge and interface bonding drivers, which are available on Linux operating system. Some network equipment venders such as Intel, HP, also provide tools for setting up the virtual interface.

The flowing sections are guidelines of some specific implementations.

A.1. Linux Bonding Driver

The Linux bonding driver provides a method for aggregating multiple network interfaces into a single logical bonded interface. The behavior of the bonded interfaces depends upon the modes such as high availability or maximum throughput mode.

There are two methods for configuring bonding: with support from the distro's network initialization scripts, and without. Distros generally use one of two packages for the network initialization scripts: initscripts or sysconfig. Recent versions of these packages have support for bonding, while older versions do not. /etc/net has built-in support for interface bonding.

The detail implementation of the Linux Bonding Driver can be found in [Linux-Bonding-Driver]

A.2. Net:Bridge

The Linux bridging implementation provides a bridge that can connect two or more physical NICs together to form one logical interface. Packets are forwarded based on NIC's address, rather than IP address. Since forwarding is done at Layer 2, all protocols can go transparently through a bridge.

The Linux bridging code has been integrated into 2.4 and 2.6 kernel series. The detail implementation of the Linux Bonding Driver can be found in [Net:Bridge]

A.3. Intel Advanced Networking Services With Ethernet Teaming

Intel provides a graphical user interface tool, called PROSet, for managing Intel's network products and Advanced Networking Services (ANS). PROSet is designed to run in on Linux, Microsoft(R) Windows(R) 2000, Windows Server 2003, Windows Server 2008 and Windows 7. PROSet is used to perform diagnostics, configure load balancing and fault tolerance teaming, and VLANs. In addition, it displays the MAC address, driver version, and status information. On Linux the

PROSet executable is known as xprocfg. Xprocfg can be used in custom initialization scripts.

Operating System	Configuration Tool
Windows (All versions)	Intel PROSet
NetWare 5/6	Autoexec.ncf; iAns.lan, Inetcfg
Linux	Xprocfg

Table 1: Operating System Configuration Tools

The detail implementation of the Intel's PROSet can be found in [Intel-PROSet]

Appendix-A

Authors' Addresses

Hidetoshi Yokota
KDDI Lab
2-1-15 Ohara, Fujimino
Saitama, 356-8502
JP

Email: yokota@kddilabs.jp

Sri Gundavelli
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Email: sgundave@cisco.com

Tran Minh Trung
ETRI
161 Gajeong-Dong Yuseung-Gu
Daejeon,   305-700
Korea

Phone: +82 42 860 1132
Email: trungtm2909@gmail.com

Yong-Geun Hong
ETRI
161 Gajeong-Dong Yuseung-Gu
Daejeon,   305-700
Korea

Phone: +82 42 860 6557
Email: yonggeun.hong@gmail.com

Kent Leung
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Email: kleung@cisco.com