

Honours Year Project Report

Integrating PPP and Mobile-IP

By

Lim Lip Yeow

Department of Information Systems

and

Computer Science

National University of Singapore

1997/1998

Project Number: 452
Project Advisor: Dr. Tay Yong Chiang & Dr. Shum Kam Hong
Deliverables: Report: 1 Volume
Program: 1 Diskette

Abstract

The Mobile Internet Protocol(MIP) is an extension of the Internet Protocol to support mobility of host while maintaining the same IP address. The Point-to-Point Protocol (PPP) is a protocol for networking over serial (possibly telephone) lines. Due to wide availability of compact computers especially notebook computers and modems, computers are now more mobile and network access using PPP over telephone lines are cheap and widespread. However, interaction between this 2 protocols are underspecified and results in certain problems. One of such problems is IP address assignment: MIP may not require an address assigned via PPP Internet Protocol Control Protocol . This project addresses this problem and implements the draft “Mobile-IPv4 Configuration Option for PPP IPCP ” to solve it. This report discusses the problem(s), the draft and some implementation details.

Subject Descriptors:

- C.2.2 Network Protocols
- C.2.1 Network Architecture and Design
- D.4.4 Communication Management

Keywords:

Mobility support, Point-to-point serial link.

Implementation, Software and Hardware:

IBM PC/XT, Linux Slackware 3.2, Linux Kernel Source 2.0.31,

PPP 2.2.0, Gnu C Compiler.

Contents

- 1 Introduction** **1**
 - 1.1 Background 1
 - 1.2 Objective of this Project 2
 - 1.3 Topology of this Report 3

- 2 Problem Analysis** **4**
 - 2.1 The Mobile Internet Protocol (MIP) 5
 - 2.1.1 Protocol Overview 5
 - 2.1.2 Terminology 6
 - 2.1.3 Foreign Agent Care-of address Mode 9
 - 2.1.4 Co-located Care-of address Mode 11
 - 2.2 The Point-to-Point Protocol 12
 - 2.2.1 Protocol Overview 12
 - 2.2.2 The PPP Internet Protocol Control Protocol 15
 - 2.3 The Problem 17

2.3.1	Terminology	17
2.3.2	IP Address Assignment	18
2.4	The MIPv4 Configuration Option for PPP IPCP	21
2.4.1	Protocol Overview	22
2.4.2	Examples of Protocol Operation	23
3	Implementation of the Mobile-IPv4 Configuration Option for PPP IPCP	28
3.1	Program Structure of Linux PPP	29
3.2	The MIPv4 Configuration Option Patch	31
3.2.1	Requirements	31
3.2.2	Commandline User Interface	32
3.2.3	Overview	33
3.2.4	Major Patches	34
4	Future Work	41
4.1	Virtual Private Networks (VPN)	41
4.2	Unavailability of Foreign Agent PPP Servers	43
4.3	Security Consideration	45
5	Conclusion	48

List of Figures

2.1	IP within IP encapsulation (tunneling).	8
2.2	Operation of MIP in foreign agent care-of address mode.	10
2.3	Operation of MIP in co-located care-of address mode.	12
2.4	Frame formats for PPP, LCP and IPCP.	14
2.5	IPCP <i>IP Address</i> option negotiation.	16
2.6	Typical scenario of PPP interoperating with MIP.	18
2.7	The mobile node prefers a foreign agent and the PPP server is a foreign agent.	24
2.8	The mobile node prefers a co-located IP address and the PPP server is a foreign agent.	25
2.9	The mobile node prefers a co-located IP address and the PPP server is at home.	26
4.1	A example of using the Point-to-Point Tunneling Protocol with MIP.	47

List of Tables

2.1	Abbreviations used in the description of Mobile-IPv4 Configuration Option for PPP IPCP	23
3.1	Values that code field can take.	30
3.2	Modification to function <code>ipcp_addci</code>	35
3.3	Modification to function <code>ipcp_nakci</code>	36
3.4	Modification to function <code>ipcp_rejci</code>	37
3.5	Modification to function <code>ipcp_reqci</code> for PPP server.	40

Chapter 1

Introduction

1.1 Background

Computers and the Internet are fast becoming an integral part of everyone's life. The wide availability of cheaper and smaller computers such as notebooks and palmtops is changing the way users attach themselves to networks. Traditionally, users work on a desktop or terminal attached directly to their private LAN's or the Internet. This no longer suffice. Users typically want to be able to work from home or when they are abroad. Extension of the traditional TCP/IP protocols are therefore needed to support:

- Mobility of host computers, that is, host computers are now allowed to attach onto networks other than its home network.

- Dialup via telephone lines, this means that host computers can attach to a network over a telephone line. This network can be an Internet Service Provider which will provide Internet access to the host.

Two protocols were developed independently to address each of these requirements: the Mobile Internet Protocol [Perkins, 1996b] and the Point to Point Protocol [Simpson, 1994].

The Mobile Internet Protocol (MIP) was designed to support the mobility of host computers. Using MIP, a mobile host computer can change its point of network attachment from one network to another without any of its network applications losing connection. Moreover, the mobile host will seem to its user that it is connected to its home network even when it is roaming across networks. This is especially useful for wireless mobile host.

The Point-to-Point Protocol (PPP) enables networking over telephone lines. PPP is basically a way of carrying different protocol packets over a serial line which in most cases is a telephone line.

1.2 Objective of this Project

The objective of this project is to integrate PPP and MIP. In particular, this project implements the internet draft *Mobile-IPv4 Configuration Option for PPP IPCP* [Solomon and Glass, 1998].

By ‘integrating PPP and MIP’, the main scenario of concern is that of the MIP mobile host or node attaching to a network via a PPP link.

1.3 Topology of this Report

The next chapter(chapter 2) will describe the MIP and PPP protocols in greater detail, introduce the problem of IP address assignment and explain the solution given in the internet draft by Solomon and Glass. Chapter 3 will describe the implementation of the internet draft. Other improvements and proposals to other problems of interoperability will be discussed in chapter 4. Chapter 5 will conclude this report.

Chapter 2

Problem Analysis

Contents

2.1	The Mobile Internet Protocol (MIP)	5
2.1.1	Protocol Overview	5
2.1.2	Terminology	6
2.1.3	Foreign Agent Care-of address Mode	9
2.1.4	Co-located Care-of address Mode	11
2.2	The Point-to-Point Protocol	12
2.2.1	Protocol Overview	12
2.2.2	The PPP Internet Protocol Control Protocol	15
2.3	The Problem	17
2.3.1	Terminology	17

2.3.2	IP Address Assignment	18
2.4	The MIPv4 Configuration Option for PPP IPCP	21
2.4.1	Protocol Overview	22
2.4.2	Examples of Protocol Operation	23

This chapter describes the MIP and PPP protocols, their individual requirements, the particular problem of IP address assignment and a solution to this problem. The reader is assumed to be familiar with the Internet Protocol [Postel, 1981] and (inter)networking with the TCP/IP suite. The book *TCP/IP Illustrated, Volume 1: The Protocols* [Stevens, 1994] should be consulted should difficulty arises. The reader who is familiar with MIP and PPP may wish to skip section 2.1 and section 2.2 respectively

2.1 The Mobile Internet Protocol (MIP)

2.1.1 Protocol Overview

The Mobile Internet Protocol is an attempt to enable mobility of computers in an internet of IP-networks. The Internet Protocol [Postel, 1981] requires that every network be assigned a network address or ID and every host on that network to be assigned an IP address with the same network prefix. Routing is done by examining the network prefix of the destination IP address

of the datagram and sending the datagram via a series of routers to a router located on the destination network which sends it to the destined host.

This architecture clearly does not allow a mobile host to use its home network IP address when it is not attached to its home network, because routers will always send datagrams destined for the mobile host to its home network.

MIP solves this problem by having the mobile host to inform a particular computer on the home network of its whereabouts (registration) and having this particular computer to intercept and forward datagrams destined for the mobile host to its current location (tunneling).

The Mobile Internet Protocol specifies 2 modes of operation when the mobile host is in a foreign network:

1. the co-located care-of address mode and
2. the foreign agent care-of address mode.

When the mobile host is at home it can operate normally as its IP address is topologically correct.

2.1.2 Terminology

Some definitions and terminologies are now described before proceeding to more technical discussion.

Mobile Host/Node A host computer which can change its point of network attachment from one network to another without changing its IP address.

Home Agent The host or router on the home network that intercepts and forwards (using a tunnel) datagrams destined to a mobile node when the mobile node is not at home.

Foreign Agent A router on a foreign network that provides routing services to a mobile node.

Home Address The IP address assigned to the mobile node which remains unchanged even when the mobile node is not at home.

Foreign Network Any network whose network address or ID is different from a mobile node's home network's.

Topologically correct/routable An IP address is said to be topologically correct or routable if the host or router using that address is physically attached to the network to which that IP address belongs. For example, the IP address 137.132.21.3 with netmask 255.255.255.0 is topologically correct or routable if the host using it is physically attached to the 137.132.21.0 network. This IP address is topologically incorrect if the host is physically attached to the 137.132.88.0 network.

Tunnel A method of transporting topologically non-routable IP datagrams ¹

by encapsulating that datagram in another topologically routable datagram. See figure 2.1. In this document, IP within IP encapsulation [Perkins, 1996a] is assumed; however, other encapsulation schemes such as Generic Routing Encapsulation (GRE) [Hanks et al., 1994] are possible.

Care-of Address The IP address of the foreign endpoint of the tunnel used to forward packets from the home agent to the mobile node.

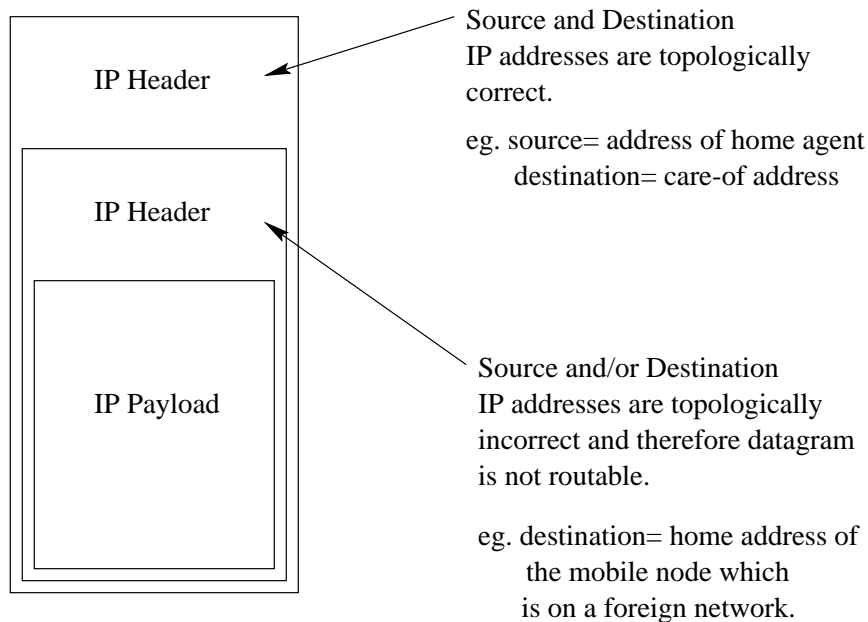


Figure 2.1: IP within IP encapsulation (tunneling).

¹In this document, an IP datagram includes the IP header and the payload

2.1.3 Foreign Agent Care-of address Mode

A foreign agent is a MIP entity on a foreign network. It broadcasts or multicasts *agent advertisements* at periodic intervals to alert any mobile node attaching onto its network of its presence. This agent advertisement may also be used by the mobile node to determine if it has changed its point of network attachment.

In this mode of operation, the mobile node's network interface on the foreign network will retain its home address which is not topologically routable; hence it registers via the foreign agent which forwards the registration request to the home agent. Upon receiving a valid registration request, the home agent will know the mobile node's current location or care-of address. In this case, the care-of address is the foreign agent's address, since the foreign agent will route packets 'on behalf' of the mobile node.

To illustrate, consider the mobile node running a telnet session with a host X which is neither on the home network nor on the foreign network. Datagrams from the mobile node will be routed to X via normal routing mechanisms. Datagrams from X to the mobile node will be routed to the mobile node's home network. The mobile node's home agent will perform proxyarp for the mobile node and intercept datagrams destined for it. These datagrams need to be forwarded to the mobile node which is on a foreign

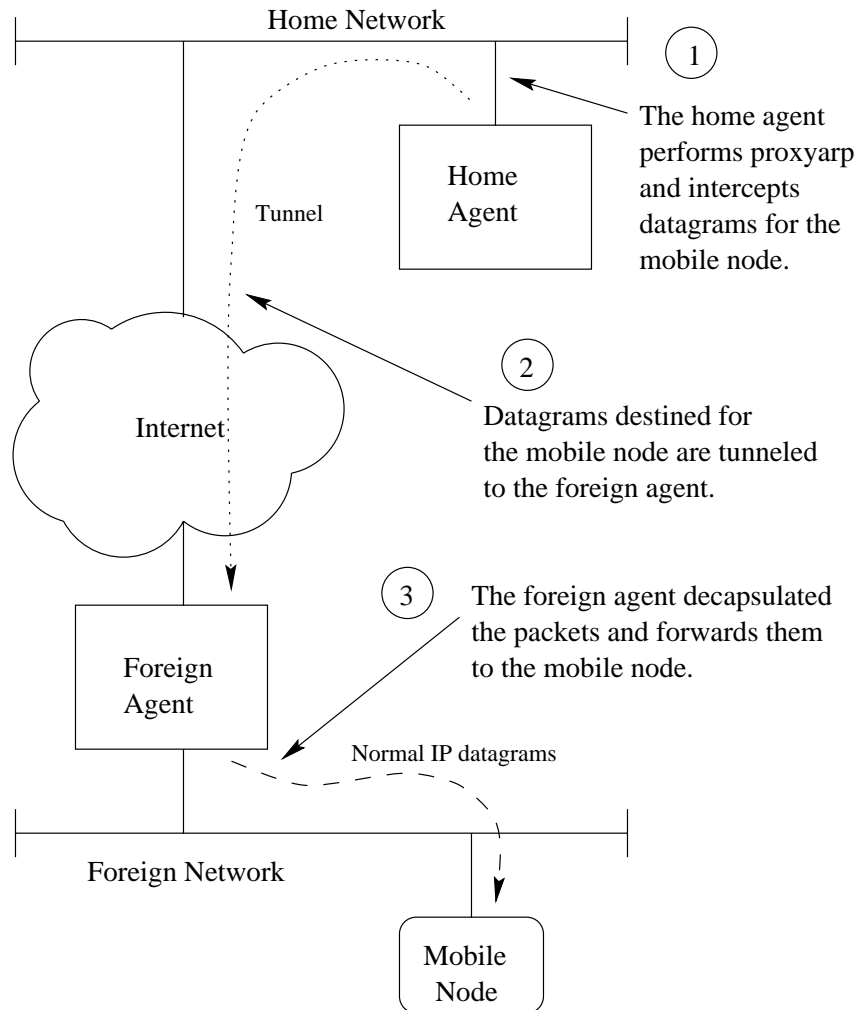


Figure 2.2: Operation of MIP in foreign agent care-of address mode.

network. The home agent does this by encapsulating the entire IP datagram in another IP datagram whose source address is the home agent's IP address and the destination address is the care-of address (in this case the foreign agent's). Once the foreign agent receives the encapsulated (tunneled) packets, it will decapsulate them and forward them to the mobile node's current link-layer address. See figure 2.2

The foreign agent care-of address mode is preferred because:

- it does not require additional IP addresses and
- many mobile nodes can use the same foreign agent as their care-of address

These are clearly advantages in view of the already limited IPv4 address space.

2.1.4 Co-located Care-of address Mode

The co-located care-of address mode caters to the scenario that the mobile node is able to be assigned a temporary, but topologically routable IP address for its network interface on the foreign network in the absence of a foreign agent. This topologically routable IP address can be assigned by a Dynamic Host Configuration Protocol (DHCP) server [Droms, 1993] or the PPP Internet Protocol Control Protocol (IPCP)[McGregor, 1992].

Once this IP address is obtained, the mobile node registers with the home agent using this address as the care-of address and subsequent forwarding of packets from home agent to mobile node will be tunneled to this address. Whereas in the foreign agent care-of address mode, the care-of address and the mobile node's home address are located on two different entities, in the

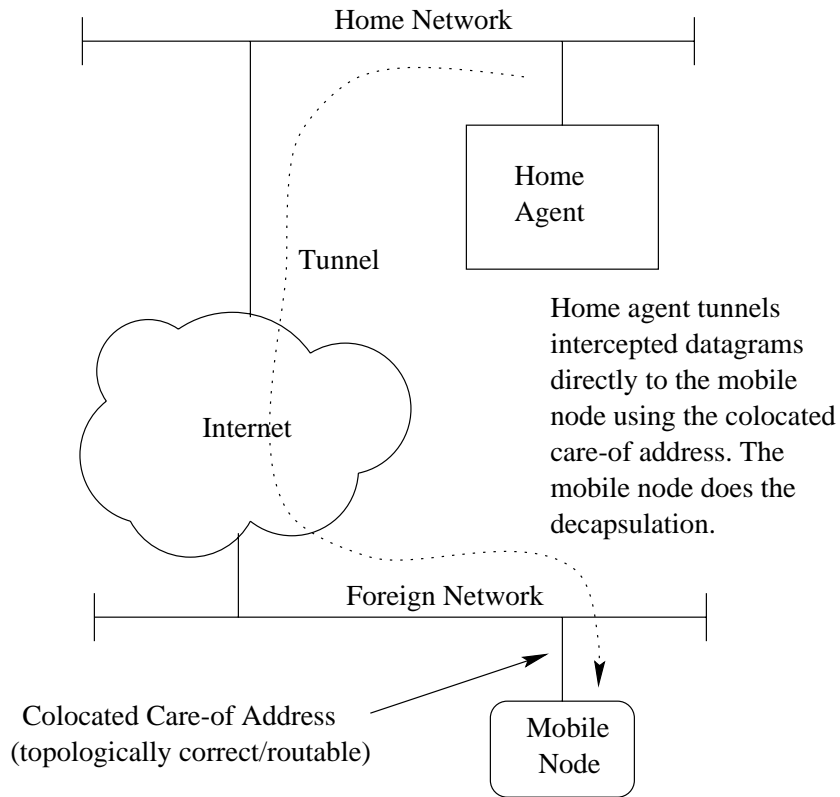


Figure 2.3: Operation of MIP in co-located care-of address mode.

co-located mode the care-of address and the home address are ‘co-located’ on the mobile node. See figure 2.3

2.2 The Point-to-Point Protocol

2.2.1 Protocol Overview

The Point-to-Point Protocol (PPP) is a method for transporting multi-protocol datagrams over point-to-point serial links. It provides 3 functionalities:

1. A method of encapsulating datagrams over serial links based on the *High-Level Data Link Control* (HDLC) protocol. All frames sent through the links are thus encapsulated.
2. A *Link Control Protocol* (LCP) to establish, configure and test the data-link connection.
3. A family of *Network Control Protocols* (NCPs) for establishing and configuring different network-layer protocols.

These 3 functionalities operate together to establish communications over a point-to-point link as follows:

- The originating PPP first sends LCP frames to configure and (optionally) test the data link.
- After the link has been established and optional facilities have been negotiated as needed by the LCP, the originating PPP sends NCP frames to choose and configure one or more network-layer protocols.
- When each of the chosen network-layer protocols has been configured, packets from each network-layer protocol can be sent over the link.
- The link will remain configured for communications until explicit LCP or NCP frames close the link, or until some external event occurs (for example, an inactivity timer expires or a user intervenes).

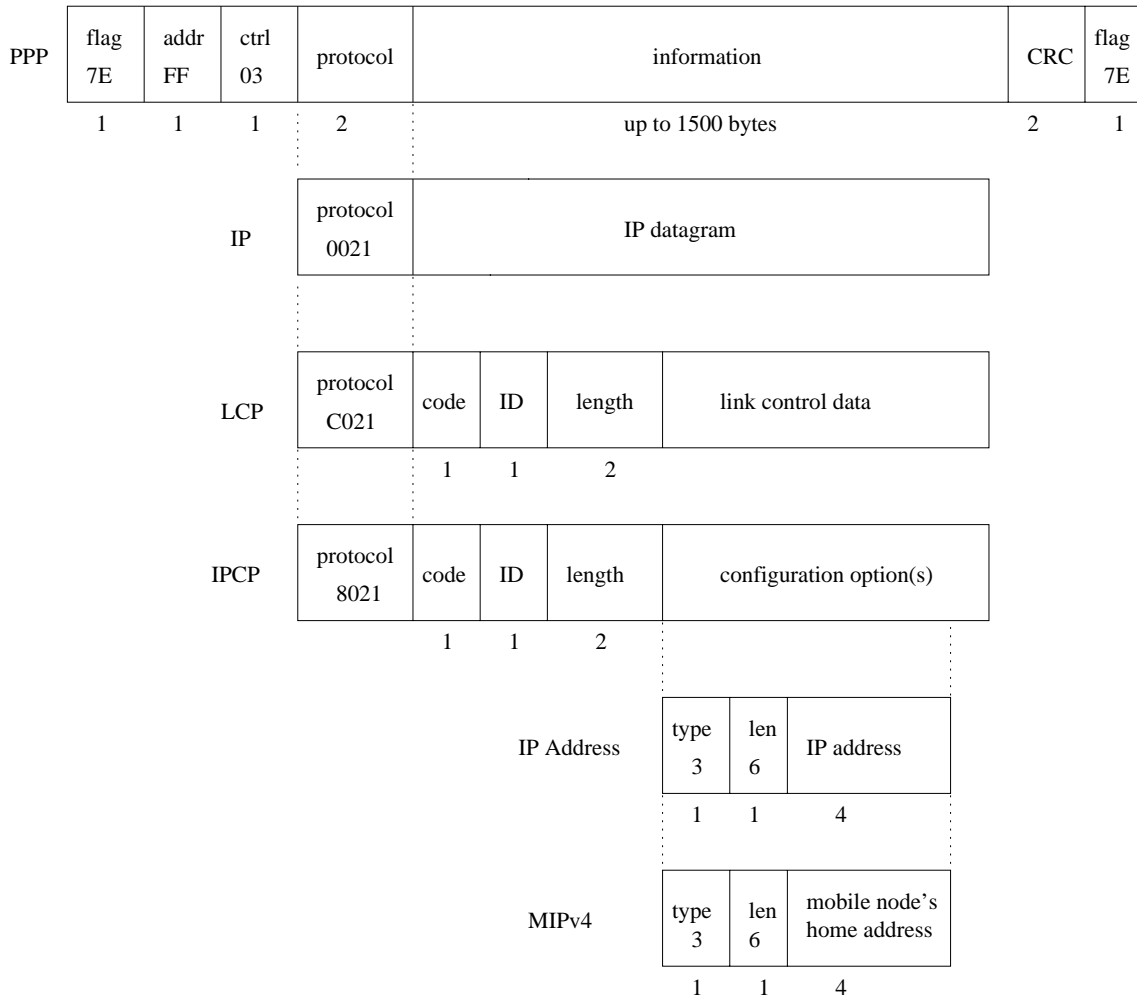


Figure 2.4: Frame formats for PPP, LCP and IPCP.

PPP frame format provides a *protocol field* for multiplexing and demultiplexing different protocol data units carried in the information field. See figure 2.4. Some of these protocols are IP, LCP and IPCP. The *code* field in the LCP and IPCP frames specifies whether that protocol data unit is a configure-request, configure-ack etc.

As the Link Control Protocol is not particularly relevant to this project, it will not be described here. The reader should consult RFC 1661 [Simpson, 1994] if more information is required.

2.2.2 The PPP Internet Protocol Control Protocol

The Internet Protocol Control Protocol [McGregor, 1992] is the network control protocol for configuring a PPP link to support the IP network layer and carry IP datagrams. Basically, it negotiates 2 parameters:

IP address each endpoint of the PPP link must have an IP address to support the IP network layer.

Compression whether compression should be performed on parts of the IP datagram prior to transmission.

The protocol operation and frame format of IPCP is the same as the Link Control Protocol (LCP) except that the option field now carries IPCP configuration option packets.

There are 2 configuration options for negotiating IP addresses: the obsolete *IP Addresses* option and the *IP Address*² option. The frame format for the *IP Address* option is shown in figure 2.4. By exchanging such frames,

²*IP Address* in slanted font denotes the IPCP configuration option and not an address in the Internet Protocol.

the 2 ends of the PPP link (henceforth known as ‘peers’) are able to discover and negotiate the IP addresses for each end of the link.

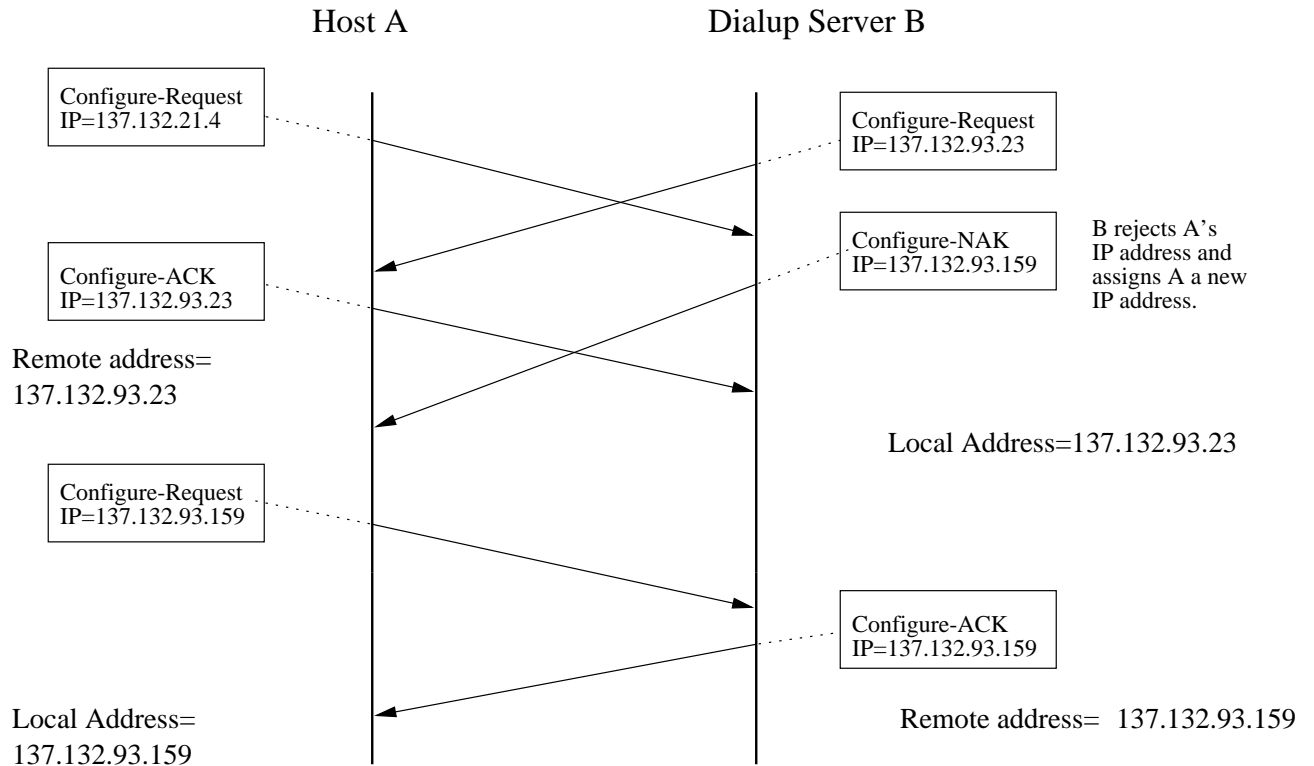


Figure 2.5: IPCP *IP Address* option negotiation.

An example of this IPCP *IP Address* option negotiation is shown in figure 2.5. In this scenario, host A can be any desktop computer that has dialed into either an Internet Service Provider (ISP) or perhaps our ISCS dialup server using a modem and a telephone line. Both ends propose their IP addresses, but the PPP on dialup server B is typically configured to reject the peer’s idea of its IP address. It sends host A a configure-NAK (negative acknowledgement) and assigns host A a topologically routable IP address.

Host A accepts and sends a configure-request to indicate that it would like to use this IP address. The dialup server B acknowledges and the negotiation completes. Note that both sides must converge on the correct 2 local and remote addresses.

2.3 The Problem

This section discusses the problem of IP address assignment that arises when MIP is used over a PPP link. The typical scenario is shown in figure 2.6 Other problems are discussed in chapter 4.

2.3.1 Terminology

For ease of discussion, the following terminologies are defined in addition to those already introduced in the previous sections.

PPP client Although there is no distinction of client and server in the PPP protocol, this artificially imposed distinction will greatly clarify the following discussion. The PPP client will loosely refer to the peer which requires its PPP peer to provide network access. The PPP client will usually be the mobile node also.

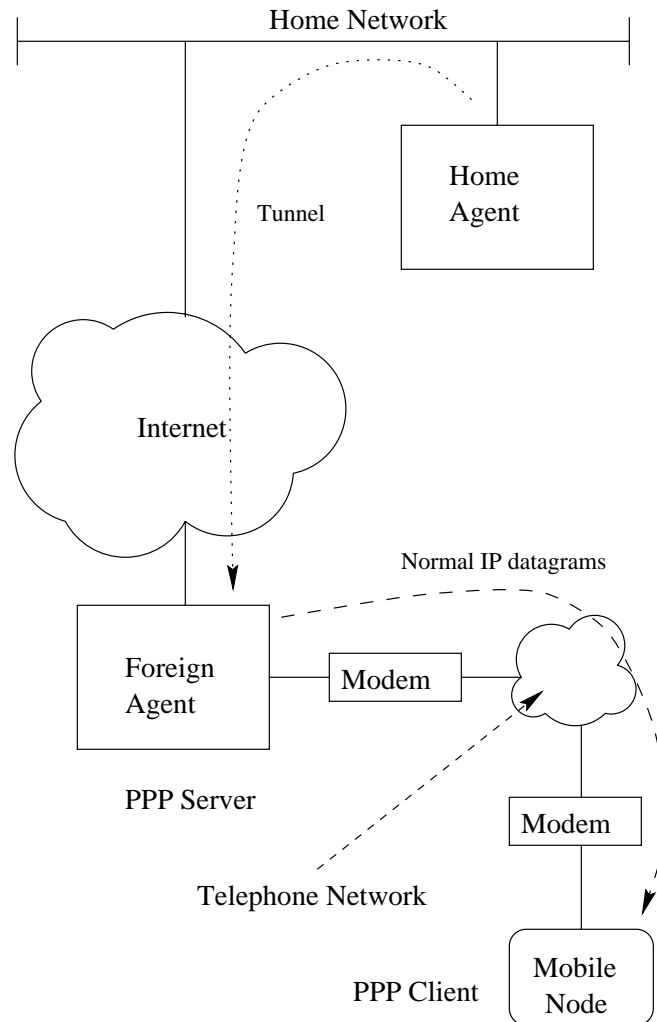


Figure 2.6: Typical scenario of PPP interoperating with MIP.

PPP server The PPP server will usually refer to the PPP peer capable of providing network access to its PPP peer. The home agent and foreign agent are usually PPP servers.

2.3.2 IP Address Assignment

We note some requirements of MIP with regards to IP addresses:

- In the co-located care-off address mode, the mobile node's network interface is assigned with a temporary, topologically correct IP address (section 2.1.4).
- In the foreign agent care-of address mode, the mobile node's network interface should be configured to its home address (section 2.1.3).
- When the mobile node is at home, its network interface address should be its home address.

Note also that the PPP IPCP's *IP Address* configuration option does not specify which IP address should be used under what circumstances and that current practice is often similar to the scenario in figure 2.5 where the PPP server dictates the assignment of a topologically correct address regardless of the circumstances. This means that when the mobile node connects to a foreign network via PPP, MIP can only operate in the co-located care-of address mode. Section 2.1.3 has already mentioned that this is *not* the preferred mode.

The problem is that interaction between PPP IPCP and MIP is underspecified and therefore ambiguous.

From a protocol point of view, there is currently no way for the mobile node to use a foreign agent care-of address, without first being assigned a unique IP address, even if the PPP server also supports foreign agent

functionality. The reason for this can be seen by walking through the IPCP negotiation:

1. A mobile node (PPP client) connects to a PPP server via PPP and proposes its home address in an IPCP Configure-Request containing the IP-Address option. In this scenario, we assume that the mobile node is connecting to some foreign link.
2. The PPP server has no way of knowing whether this Configure-Request was received from:
 - (a) a mobile node proposing its home address or
 - (b) a conventional node proposing some topologically non-routable address.

In such situation, the PPP server must (conservatively) send a Configure-Nak of the IP-Address option supplying a topologically appropriate address for use by the PPP client at the other end of the PPP link.

3. The mobile node, in turn, has no way of knowing whether this Configure-Nak was received because the PPP server is a foreign agent being conservative, or because the PPP server does not implement Mobile IP at all. Therefore, the mobile node must (conservatively) assume that the PPP server does not implement Mobile IP and continue the negotiation

of an IP address in IPCP, after which point the mobile node can use the assigned address as a co-located care-of address.

Here we observe that, even if the mobile node's peer is a foreign agent and sends an Agent Advertisement to the mobile node after IPCP reaches the Opened state, the mobile node will still have negotiated a routable address in step 3, which it is likely already using as a co-located care-of address. This defeats the purpose of foreign agent care-of addresses, which are designed to be shared by multiple mobile nodes and to eliminate the need to assign a unique address to each mobile node.

2.4 The MIPv4 Configuration Option for PPP IPCP

The Mobile-IPv4 Configuration Option for PPP IPCP is an extension to the IPCP that tries to solve the problem of IP address assignment discussed in the previous section (section 2.3.2). The internet draft specifying the Mobile-IPv4 Configuration Option for PPP IPCP has become an Internet Engineering Task Force (IETF) Proposed Standard as of 18 February 1998. This section describes the Mobile-IPv4 Configuration Option for PPP IPCP and illustrates its operation with a few examples.

2.4.1 Protocol Overview

The Mobile-IPv4 Configuration Option for PPP IPCP provides a way for the PPP server and client to negotiate which IP address to use based on what functionality each is capable of or which MIP mode is going to be used. It also detects if the PPP client is connecting to its home network and assigns the link the PPP client's home address accordingly.

The frame/option format of Mobile-IPv4 Configuration Option for PPP IPCP is already shown in figure 2.4. This MIPv4 option is used together with the *IP Address* option in the protocol, i. e. , the protocol exchanges configuration messages containing a combination of this 2 options in its operation.

Note that whereas in the original IPCP *IP Address* option protocol, the negotiation is quite symmetrical and therefore the two peers are quite indistinguishable, the Mobile-IPv4 Configuration Option for PPP IPCP protocol imposes a distinction between 'PPP client' and 'PPP server'. This is not surprising and no longer artificial because the protocol specifies interaction between a mobile node and a network access provider such as a foreign agent or a home agent.

To illustrate the protocol operation, we give a few examples in the next section (section 2.4.2). This form protocol description is chosen as it gives

PPP Message Types	
ConfReq	Configure-Request
ConfRej	Configure-Reject
ConfAck	Configure-Ack
ConfNak	Configure-Nak
IPCP Configuration Option Types	
MIPv4	Mobile-IPv4 Configuration Option
IP	<i>IP Address</i>
IP addresses	
a.b.c.d	some non-zero IP address
w.x.y.z	some non-zero IP address
home	a mobile node's home IP address
PPP server	any address that the PPP server uses for its end of the link
coa	an IP care-of address
0	the all zeros IP address (0.0.0.0)

Table 2.1: Abbreviations used in the description of Mobile-IPv4 Configuration Option for PPP IPCP

the reader a better understanding of how the aims of the protocol are accomplished. Section 3.2 will give a complete description of the protocol.

2.4.2 Examples of Protocol Operation

The abbreviations used in the following examples are described in table 2.1. The following examples also assume that both the PPP server and client implement the MIPv4 configuration option. The protocol is compatible with PPP peers that do not implement the MIPv4 option; those cases will be discussed in Section 3.2, because they are not directly relevant to the problem we are solving now. Note also that the protocol provides for more scenarios than are represented in these three examples.

Example 1 The mobile node prefers the foreign agent mode and the PPP server is an foreign agent. The mobile node proposes that it prefers the foreign agent mode by sending ConfReq(MIPv4=home). The PPP server accepts and acknowledges with ConfAck(MIPv4=home). This establishes the IP address at the PPP client's end of the link to be the home address of the mobile node. Figure 2.7 shows the entire negotiation.

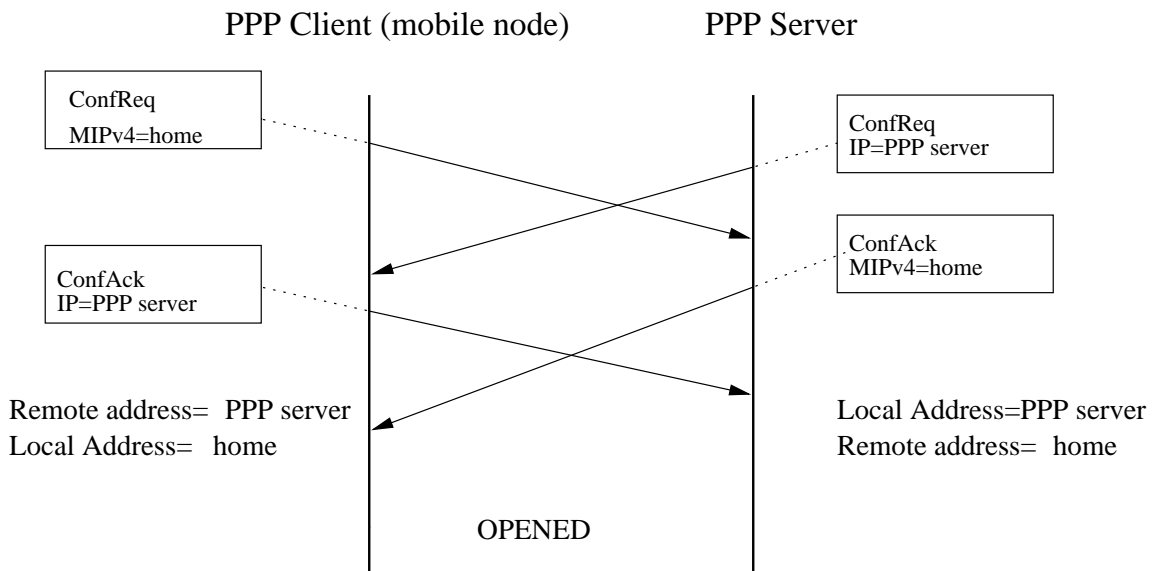


Figure 2.7: The mobile node prefers a foreign agent and the PPP server is a foreign agent.

Example 2 The mobile node prefers the co-located coa mode and the PPP server is a foreign agent which does not assign a co-locate care-of address. The mobile node proposes that it prefers the co-located coa mode by sending ConfReq(IP=0,MIPv4=home). The PPP server tells

the client that it cannot assign a co-located coa, but it can provide foreign agent functionality by sending ConfRej(IP=0) (Figure 2.8). The mobile node then proceeds to negotiate as in Example 1.

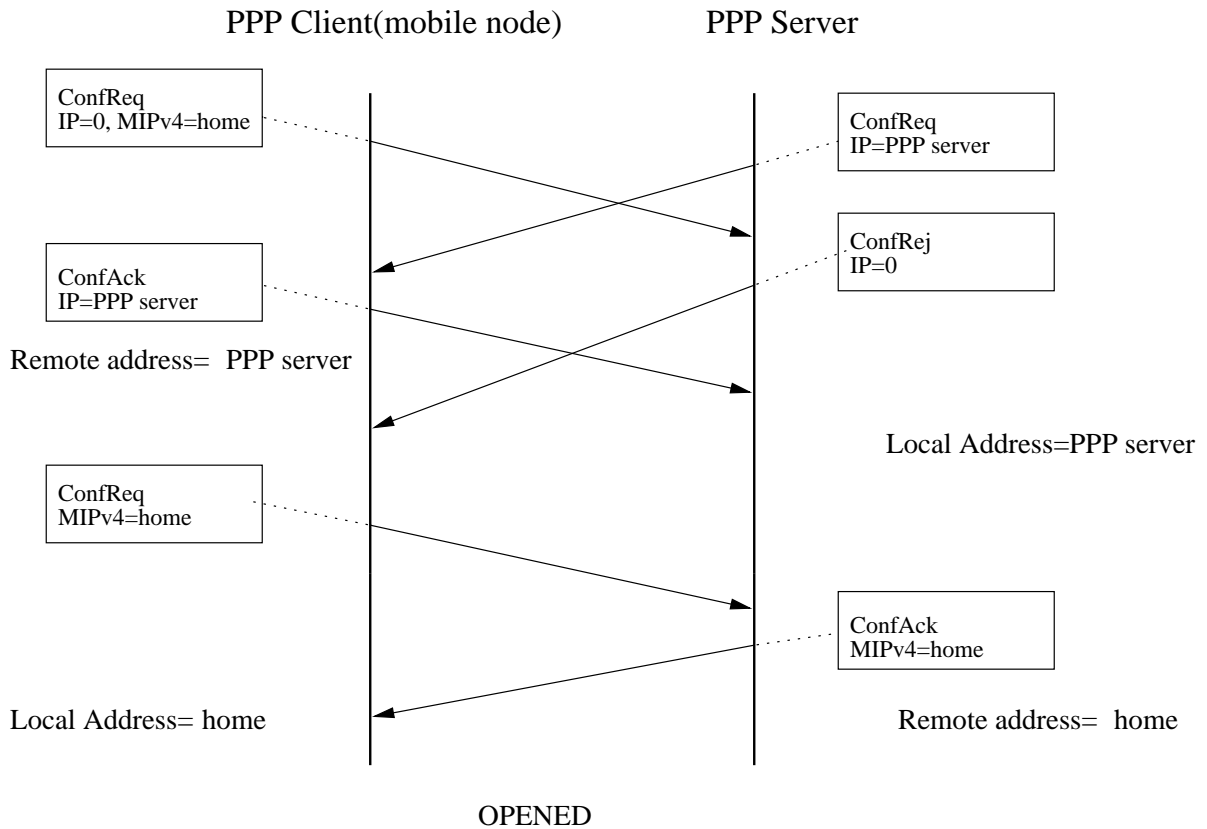


Figure 2.8: The mobile node prefers a co-located IP address and the PPP server is a foreign agent.

Example 3 The mobile node prefers the co-located coa mode and the PPP

server is located on the home network. The mobile node proposes that

it prefers the co-located coa mode by sending ConfReq(IP=0, MIPv4=home).

The PPP server tells the client that it is at home by sending Conf-

Nak(IP=home) (Figure 2.9). The mobile node double checks if it is at home by sending ConfReq(IP=home,MIPv4=home). PPP server replies with ConfAck(IP=home,MIPv4=home) telling mobile node that it is at home.

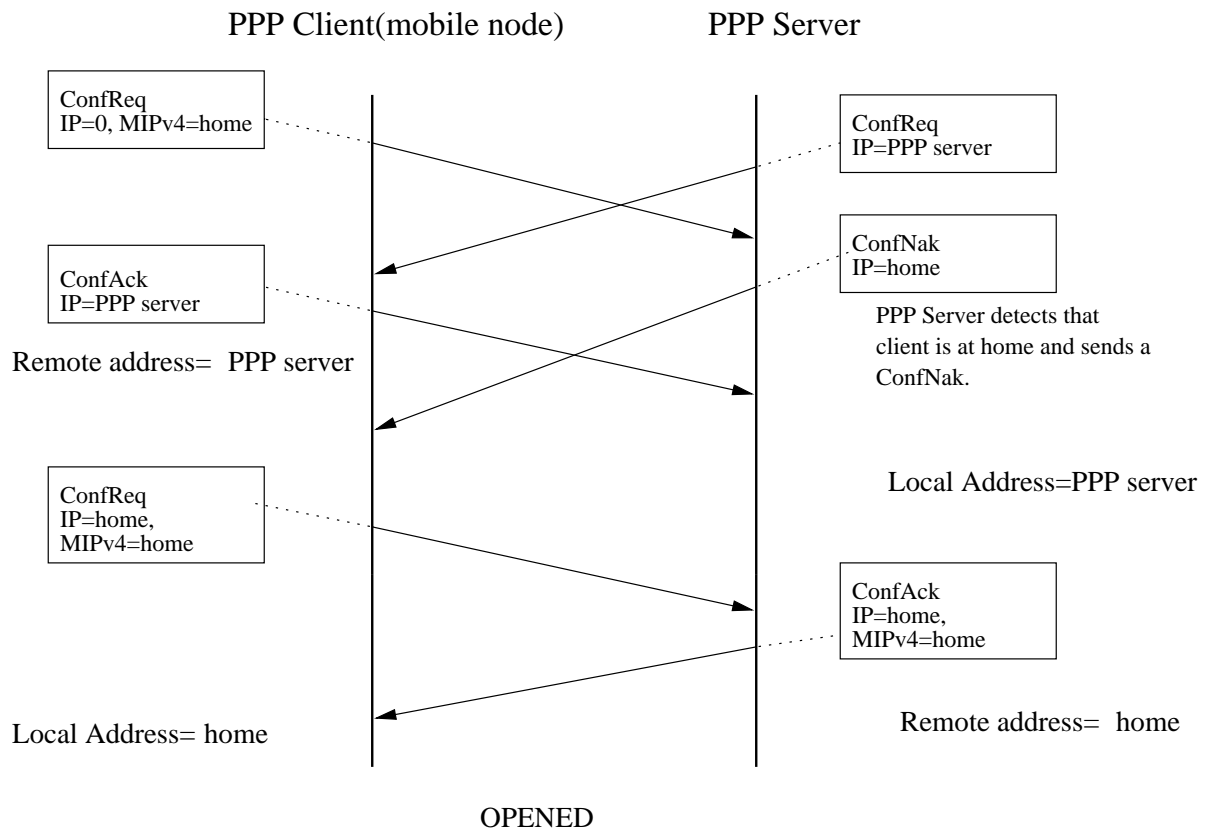


Figure 2.9: The mobile node prefers a co-located IP address and the PPP server is at home.

These three examples show that the MIPv4 configuration option protocol introduces some “intelligence” into PPP IPCP for determining what functionality the PPP server supports and whether the mobile node is at home.

The IP address of the PPP client is then fixed accordingly. No IP address is assigned by the PPP server unnecessarily.

Chapter 3

Implementation of the Mobile-IPv4 Configuration Option for PPP IPCP

Contents

3.1	Program Structure of Linux PPP	29
3.2	The MIPv4 Configuration Option Patch	31
3.2.1	Requirements	31
3.2.2	Commandline User Interface	32
3.2.3	Overview	33
3.2.4	Major Patches	34

This section describes how the Mobile-IPv4 Configuration Option for PPP IPCP is implemented on the Linux platform. Linux was chosen as the platform because its PPP source codes were widely available and free of charge.

The program structure of the Linux PPP will be described before our implementation. This is because our implementation patches the Linux PPP codes.

3.1 Program Structure of Linux PPP

Linux PPP is divided into two main components:

The PPP kernel driver manages the physical link and handles the low level HDLC framing. This module resides in the kernel.

The PPP daemon implements most of the required protocols such as LCP, IPCP, Compression Control Protocol (CCP), authentication protocols etc. It resides in user space and is typically invoked by the user with various commandline parameters.

The PPP kernel driver is of no relevance to this project and will not be described here.

The backbone of the PPP daemon (pppd) is a finite state machine(FSM) module which is shared by the Link Control Protocol (LCP) and many of

Code Field Semantics	
1	Configure-Request
2	Configure-Ack
3	Configure-Nak
4	Configure-Reject
5	Terminate-Request
6	Terminate-Ack
7	Code-Reject
8	Protocol-Reject
9	Echo-Request
10	Echo-Reply
11	Discard-Request

Table 3.1: Values that code field can take.

the network control protocols including IPCP. This FSM can be viewed as an “event dispatcher” with timeout and retransmission capabilities. Each network control protocol maintains one copy of the FSM for each link it manages. The FSM reads an incoming packet from the link via the PPP kernel driver, demultiplexes it according to the code field and dispatches it to the relevant FSM event handling function. See figure 2.4 for the frame formats and table 3.1 for the values of code field.

This FSM event handling function will further dispatch the packet to the protocol specific event handling function whose pointer is stored in a data-structure initialized by the specific protocol. A buffer for storing outgoing packets is also passed as parameter (the pointer is passed) to the protocol specific event handling function so that any reply packets can be communicated to the FSM which sends it out onto the link.

3.2 The MIPv4 Configuration Option Patch

Our implementation of the MIPv4 configuration option is based on the Linux PPP version 2.2.0 code. Patching was chosen over a rewrite of the entire PPP daemon because the current PPP daemon is excellent and there is no necessity to write codes that do not directly relate to this project.

The description of our implementation will begin with the requirements, the commandline user interface, an overview of the modification and finally the important functions that is patched. As far as possible, insignificant coding details will not be described.

3.2.1 Requirements

We list here a few important requirements that our implementation must meet in order to be compliant to the draft [Solomon and Glass, 1998]:

1. Compatibility with non MIPv4 option enabled PPP. Naturally, only co-located mode will be supported in such situations.
2. ConfNak of the MIPv4 option is undefined and must not be sent.
3. A unique IP address must never be assigned unless absolutely necessary.
4. Non-MIP-aware peers and non-mobile nodes must never send a ConfReq with a MIPv4 configuration option.

5. PPP servers not performing MIP functionality should send a ConfRej in response to any MIPv4 option request.
6. The *IP Addresses* configuration option must never be used with the MIPv4 option.

3.2.2 Commandline User Interface

Three new commandline options are added:

“mip” The presence of this commandline option indicates that the host implements MIP. IPCP will process IP address negotiation with the MIPv4 configuration option. The next two options are meaningful only if this option is set.

“agent” Indicates that the current host is either a home agent or foreign agent. When not set, the host is assumed to be a mobile node by default.

“colocated” Indicates that the current host prefers the co-located care-of address mode. When not set, it defaults to the foreign agent care-of address mode.

Two existing commandline options have been overloaded with new semantics:

<local IP address>:<remote IP address> The local IP address portion of this option is used to supply the protocol with the mobile node's home address.

netmask n The netmask n is used to compute the network prefix of the PPP server's and client's proposed IP address to detect connection to home network.

3.2.3 Overview

We modify the IPCP module(`ipcp.c`) so that the PPP client will:

1. initiate negotiation with a ConfReq containing a MIPv4 configuration option;
2. send the appropriate ConfReq's at the appropriate time;
3. respond to the different replies ConfRej, ConfAck, ConfNak appropriately by changing state;

and the PPP server will:

1. never send a ConfReq containing a MIPv4 configuration option;
2. respond to the appropriate ConfReq's by sending appropriate ConfRej, ConfAck, ConfNak etc.

Note that the server is more passive and need not maintain any state information whereas the client needs to maintain state information so that it knows which ConfReq to send next should previous ConfReq fail to be accepted by the server. The static global 1-byte variable `MipV4state` is used to store the client's state information. We cannot incorporate these new states into the FSM because the FSM is shared by other protocols; hence the new states are added to the IPCP event handling functions.

3.2.4 Major Patches

The modifications or additions to important functions in `ipcp.c` are described here. Many tedious details such as error checking, cleanup, packet preprocessing, global variable maintainence etc. are omitted.

`ipcp_addci(f, ucp, lenp)` This function is called by FSM to add configuration information (“ci”) to ConfReq packets. Code to send the appropriate MIPv4 ConfReq according to the state are added. Table 3.2.

State	Action
MIPV4INITIAL	<i>if</i> commandline option “colocated” is set, <i>then</i> send ConfReq(IP=0, MIPv4=home) and change state to MIPV4REQ1SENT

State	Action
	<i>else</i> send ConfReq(MIPv4=home).
MIPv4USEFA	Send ConfReq(MIPv4=home).
MIPv4USECOA	Send ConfReq(IP=coa, MIPv4=home) and change state to MIPv4REQ1SENT
MIPv4ATHOME	Send ConfReq(IP=home, MIPv4=home)
MIPv4USEREQ5	Send ConfReq(IP=0) and change state to MIPv4REQ5SENT
MIPv4USEREQ6	Send ConfReq(IP=a.b.c.d), where a.b.c.d may be determined by previous exchanges.
MIPv4USEDEFAULT	Send ConfReq(IP=home).

Table 3.2: Modification to function `ipcp_addci`.

Note that `MIPv4USEREQ5`, `MIPv4USEREQ6`, `MIPv4USEDEFAULT` are for backward compatibility. Numbering in state values follow numbering of cases in the original draft [Solomon and Glass, 1998].

`ipcp_nakci(f, ucp, lenp)` This function is called by FSM when it receives a ConfNak. Code to change state so that `ipcp_addci` will send the appropriate ConfReq are added. Table 3.3.

Packet Contents	Action
ConfNak(IP=coa)	Change state to MIPV4USECOA and set IP address to try to coa.
ConfNak(IP=home)	Change state to MIPV4ATHOME.

Table 3.3: Modification to function `ipcp_nakci`.

ipcp_rejci(f, ucp, lenp) This function is called by FSM when it receives a ConfRej. Code to change state so that `ipcp_addci` will send the appropriate ConfReq are added. Table 3.4.

Packet Contents and State	Action
ConfRej(IP=0),MIPV4REQ1SENT	Change state to MIPV4USEFA.
ConfRej(IP=0),MIPV4REQ5SENT	Change state to MIPV4USEDEFAULT.
ConfRej(IP=home)	Change state to MIPV4USEFA.
ConfRej(IP=coa)	Change state to MIPV4USEFA.
ConfRej(MIPv4=home)	<i>if</i> at home, <i>then</i> change state to MIPV4USEREQ6 <i>else</i> change state to MIPV4USEREQ5.

Packet Contents and State	Action
ConfRej(IP=don't care,MIPv4=home)	Change state to MIPV4USEDEFAULT.

Table 3.4: Modification to function `ipcp_rejci`.

ipcp_ackci(f, ucp, lenp) This function is called by the FSM when it receives a ConfAck. Code is added to check for which option is acknowledged and IP addresses are set accordingly.

ipcp_reqci(f, ucp, lenp) This function is called by FSM when it receives a ConfReq. The PPP client should determine if it is at home from the ConfReq received from the PPP server. The PPP server needs to interpret the different ConfReq's and construct the appropriate replies.

Table 3.5 shows the PPP server's logic.

Packet Contents	Reply
ConfReq(IP=0,MIPv4=home)	ConfNak(IP=coa) Use coa as your co-located address.
	ConfNak(IP=home) You are at home.
	ConfRej(IP=0) I cannot assign an address, use me as a foreign agent.

Packet Contents	Reply
	ConfRej(MIPv4=home) I do not implement the MIPv4 option.
	ConfRej(IP=0,MIPv4=home) Use default.
ConfReq(IP=coa,MIPv4=home)	ConfAck(IP=coa,MIPv4=home) ok, use coa as your co-located care-of address. Opened.
	ConfNak(IP=alternate-coa) no, use alternate-coa as your co-located care-of address.
	ConfNak(IP=home) you are at home.
	ConfRej(IP=coa) coa is not topologically correct and I cannot assign you a correct one; you may use me as a foreign agent
	ConfRej(MIPv4=home) I do not implement the Mobile-IPv4 option.
	ConfRej(IP=coa,MIPv4=home) use the default.
ConfReq(IP=home, MIPv4=home)	ConfAck(IP=home,MIPv4=home) yes, you are at home. Opened.
	ConfNak(IP=coa) you are not at home, use

Packet Contents	Reply
	coa as your co-located care-of address.
	ConfRej(IP=home) you are not at home and I cannot assign a co-located care-of address; you may use me as a foreign agent.
	ConfRej(MIPv4=home) I do not implement the Mobile-IPv4 option.
	ConfRej(IP=home,MIPv4=home) Use the default.
ConfReq(MIPv4=home)	ConfAck(MIPv4=home) ok, wait for an advertisement to figure out where you are. Opened.
	ConfRej(MIPv4=home) I do not implement the Mobile-IPv4option.
ConfReq(IP=0)	ConfNak(IP=a.b.c.d) Use a.b.c.d as your address/co-located-care-of-address” .
	ConfRej(IP=0) I cannot assign an address or I do not implement the IP-Address option.
ConfReq(IP=a.b.c.d)	ConfAck(IP=a.b.c.d) ok, a.b.c.d is your address/home- address/co-located-care-of-

Packet Contents	Reply
	address. Opened.
	ConfNak(IP=w.x.y.z) No, use w.x.y.z as your address/home-address/co-located-care-of-address.
	ConfRej(IP=a.b.c.d) a.b.c.d is not topologically correct, but I cannot give you a good one or I do not implement theIP-Address option.

Table 3.5: Modification to function `ipcp_reqc` for PPP server.

Chapter 4

Future Work

This section continues the thread of discussion started in section 2. Other problems of integrating PPP and MIP that are not solved and implemented by this project will be discussed. In particular, we will discuss the problem of virtual private networks, unavailability of foreign agent PPP servers and security .

4.1 Virtual Private Networks (VPN)

The IP addresses that have been referred to so far are all real public IP addresses. It is already mentioned that the IPv4 public address space is limited and constrained. A natural question to ask would be whether the PPP server can assign private addresses and whether mobile nodes can originate

from virtual private networks and therefore have home addresses that are private.

The first question is not of much interest, since we prefer the foreign agent mode of operation. Even so, the problem can be solved with the PPP server performing Network Address Translation (NAT) or IP masquerading in Linux and issuing private addresses to PPP clients.

The second question is tricky. Will the mobile node be allowed to send datagrams using its home address to a host having a public address? How will this correspondent host route its reply to the mobile node? To fully solve this problem requires additional routing mechanisms, the description of which are beyond the scope of this report. However, a partial solution has been proposed by Kory Hamzeh in RFC 2107 [Hamzeh, 1997] called the *Ascend Tunnel Management Protocol (ATMP)*.

ATMP provides a subset of the functionalities of MIP. It is implemented by Ascend Communications as a commercial product and is also supported by the newer versions of Linux. ATMP like MIP consists of 3 entities: the home agent, the foreign agent and the mobile node. The home agent is a router or a gateway between the private network and the Internet. The foreign agent is any network access provider on the Internet. The mobile node connects to the foreign agent, uses its home address (private) for the network attachment and registers with the home agent via the foreign agent. Once

registration succeeds, a bidirectional tunnel is set up between home agent and foreign agent. The mobile node will have remote virtual connection to its private network. Internet access can be accomplished via a gateway performing network address translation (NAT) on the home network. Like the MIP, if the mobile node connects to the foreign agent via PPP, the MIPv4 configuration option can be used to negotiate the IP addresses of the PPP endpoints.

This protocol is sufficient for remote connection to VPN, but not sufficient to support full mobility of the mobile node. Access to the Internet by the mobile node via ATMP also incurs a high overhead since datagrams need to be routed via the gateway on the home network twice in a round trip. Perhaps future work could be done to refine the protocol or incorporate it into MIP.

4.2 Unavailability of Foreign Agent PPP Servers

Another natural question would be if the PPP server is not a foreign agent, but there is a foreign agent nearby, say less than 2 hops away, can the mobile node PPP client run in foreign agent mode using the ‘nearby’ foreign agent?

The Point-to-Point Tunneling Protocol (PPTP) [Hamzeh et al., 1997] may offer a partial solution. (Note that the PPTP may be superseded by the more

general Layer Two Tunneling Protocol–L2TP [Hamzeh et al., 1998])

Recall from section 2.2 that a PPP peer (particularly the server) performs the following functions:

1. Physical link management.
2. Logical Link Control using LCP.
3. Network Control using NCP.
4. Multiprotocol routing and network access.

PPTP allows one machine to handle function 1 and possibly 2 and another machine to perform the rest of the functions. The first machine is called a PPTP Access Concentrator (PAC) which provides the physical serial line connection. The second machine is a PPTP Network Server (PNS) and is separated from the PAC by an IP network. L2TP generalizes this basic idea so that the network between the PNS and the PAC need not be restricted to an IP network. PPP frames are then tunneled between PNS and PAC using PPTP or L2TP. Only the PNS and PAC need to implement these protocols which are not ‘visible’ to the PPP client. See figure 4.1

Using PPTP, a mobile node PPP client can use a foreign agent that is a PPTP/L2TP PNS and is a few hops away. The MIP need no additional modifications for this; however, foreign agent discovery capability must be in-

corporated into PPTP/L2TP. This foreign agent discovery capability should try to discover the closest foreign agent PNS within a fixed number of hops. Should this fail, the non-foreign-agent PNS (we assume one is always available) should assign a topologically correct IP address using IPCP and the mobile node should fall back on MIP in co-located mode.

Note also that all these modifications are really complicating the whole IP routing fabric and that perhaps an entirely new approach might yield a cleaner and more elegant solution.

4.3 Security Consideration

In this section, the scenario of concern is that of a mobile node requiring confidentiality between itself, the correspondent host and the home agent. Such scenarios are not contrived: A police officer using a mobile node on a foreign network may be transmitting classified information back to its home network or to a correspondent host in another police department; An employee might be working using a mobile node on a sensitive company project.

The Mobile-IP base protocol already provide some form of authentication. What is required now is confidentiality. Of some importance here is whether a foreign agent can be trusted. If the foreign agent can be trusted, it can perform the computationally expensive encryption. If not, the mobile node

will have to perform it. A protocol may be needed to decide who performs encryption.

The next crucial point is how secure a PPP link is and how to ensure security on the IP network?

The latter is dealt with in RFC 1825 *Security Architecture for the Internet Protocol*[Atkinson, 1995b] and RFC 1827 *IP Encapsulating Security Payload*[Atkinson, 1995a].

The former is also dealt with in another protocol described in the internet draft *Securing L2TP using IPSEC*[Patel and Aboba, 1997].

With respect to MIP and PPP, if there is going to be 2 layers of security measure (one link level security negotiation and encryption-decryption and another network level one at the foreign agent), we could just as well let the mobile node perform the security measures and turn off the ones at the link layer and at the foreign agent. This may result in smaller overheads. Currently this is also underspecified.

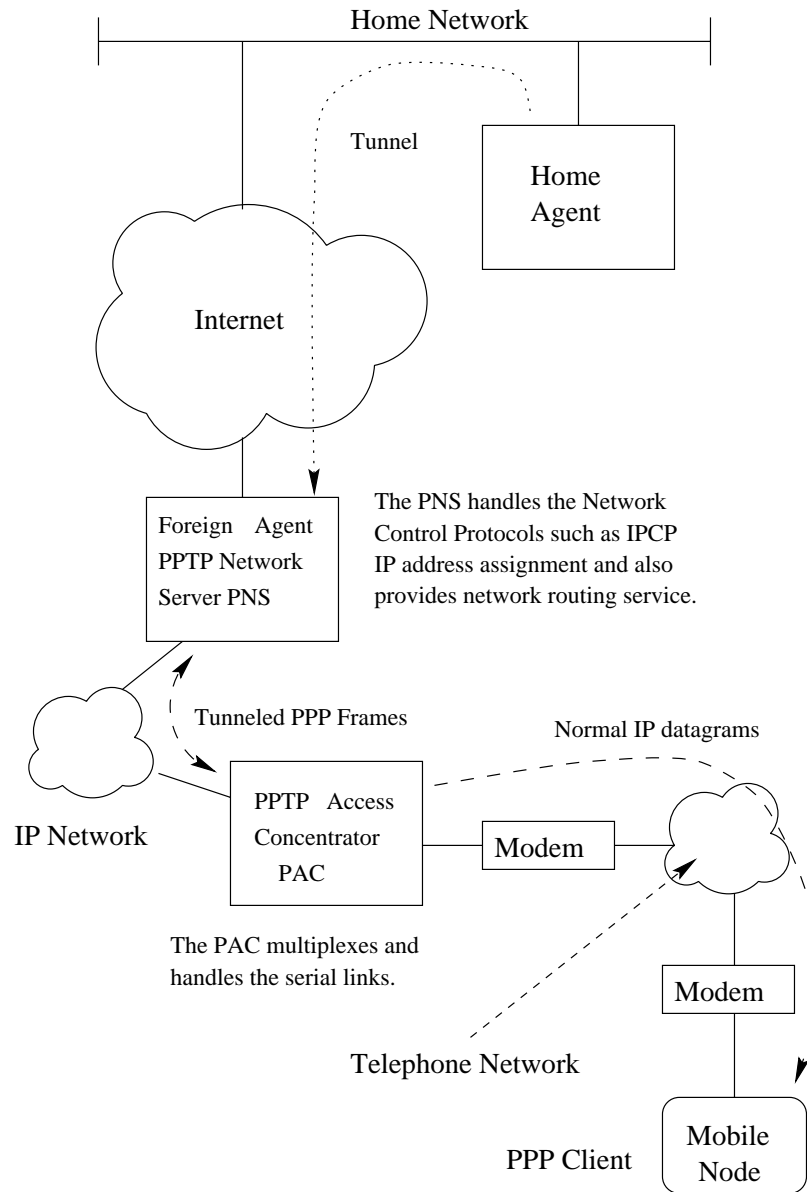


Figure 4.1: A example of using the Point-to-Point Tunneling Protocol with MIP.

Chapter 5

Conclusion

This report has described and explained the Mobile Internet Protocol and the Point-to-Point Protocol. The problems with their interoperability has also been highlighted. In particular, this project deals with the problem of IP address assignment by PPP IPCP. The problem arises because of the lack of specification of the two protocols on their interoperability. The draft *Mobile-IPv4 Configuration Option for PPP IPCP* [Solomon and Glass, 1998] is a specification that resolves this ambiguity between the two protocols and this project provides an implementation of this draft. We have described our implementation in chapter 3 and discussed further problems and proposals in the previous chapter. Many of these problems require further protocol design and modifications. Much work is still needed in this area to make mobile-IP an attractive and viable solution. One important factor is the compatibility of

MIP with other protocols. Our implementation of Mobile-IPv4 Configuration Option for PPP IPCP is a step towards the compatibility of mobile-IP with PPP. Although direct network connections seem to be ousting PPP's niche, PPP over the telephone line remains a cheap and widely available alternative.

Bibliography

[Atkinson, 1995a] Atkinson, R. (1995a). IP Encapsulating Security Payload (ESP). *RFC 1827*.

[Atkinson, 1995b] Atkinson, R. (1995b). Security Architecture for the Internet Protocol. *RFC 1825*.

[Droms, 1993] Droms, R. (1993). Dynamic Host Configuration Protocol. *RFC 1541*.

[Hamzeh, 1997] Hamzeh, K. (1997). Ascend Tunnel Management Protocol - ATMP. *RFC 2107*.

[Hamzeh et al., 1997] Hamzeh, K., Pall, G. S., Verthein, W., Taarud, J., and Little, W. A. (1997). Point-to-Point Tunneling Protocol – PPTP. *Internet Draft*. draft-ietf-pppext-pptp-02.txt.

[Hamzeh et al., 1998] Hamzeh, K., Rubens, A., Kolar, T., Littlewood, M., Palter, B., Valencia, A. J., Townsley, W. M., Pall, G. S., Verthein, W.,

- and Taarud, J. (1998). Layer Two Tunneling Protocol L2TP. *Internet Draft*. draft-ietf-pppext-l2tp-09.txt.
- [Hanks et al., 1994] Hanks, S., Li, T., Farinacci, D., and Traina, P. (1994). Generic Routing Encapsulation (GRE). *RFC 1701*.
- [McGregor, 1992] McGregor, G. (1992). The PPP Internet Protocol Control Protocol (IPCP). *RFC 1332*.
- [Patel and Aboba, 1997] Patel, B. V. and Aboba, B. (1997). Securing L2TP using IPSEC. *Internet Draft*. draft-ietf-pppext-l2tp-security-00.txt.
- [Perkins, 1996a] Perkins, C. (1996a). IP Encapsulation within IP. *RFC 2003*.
- [Perkins, 1996b] Perkins, C. (1996b). IP Mobility Support. *RFC 2002*.
- [Postel, 1981] Postel, J. (1981). Internet Protocol. *STD 5 RFC 791*.
- [Simpson, 1994] Simpson, W. A. (1994). The Point-to-Point Protocol (PPP). *RFC 1661*.
- [Solomon and Glass, 1998] Solomon, J. and Glass, S. (1998). Mobile-IPv4 Configuration Option for PPP IPCP. *Internet Draft*. draft-ietf-pppext-ipcp-mip-03.txt.
- [Stevens, 1994] Stevens, W. R. (1994). *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley.