Network Working Group
Request for Comments: 2341
Category: Historic

A. Valencia
M. Littlewood
T. Kolar
Cisco Systems

May 1998

Cisco Layer Two Forwarding (Protocol) "L2F"

Status of Memo

This memo describes a historic protocol for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

Virtual dial-up allows many separate and autonomous protocol domains to share common access infrastructure including modems, Access Servers, and ISDN routers. Previous RFCs have specified protocols for supporting IP dial-up via SLIP [1] and multiprotocol dial-up via PPP [2]. This document describes the Layer Two Forwarding protocol (L2F) which permits the tunneling of the link layer (i.e., HDLC, async HDLC, or SLIP frames) of higher level protocols. Using such tunnels, it is possible to divorce the location of the initial dial-up server from the location at which the dial-up protocol connection is terminated and access to the network provided.

Table of Contents

1.0 Introduction

The traditional dial-up network service on the Internet is for registered IP addresses only. A new class of virtual dial-up application which allows multiple protocols and unregistered IP addresses is also desired on the Internet. Examples of this class of network application are support for privately addressed IP, IPX, and AppleTalk dial-up via SLIP/PPP across existing Internet infrastructure.

The support of these multiprotocol virtual dial-up applications is of significant benefit to end users and Internet Service providers as it allows the sharing of very large investments in access and core infrastructure and allows local calls to be used. It also allows existing investments in non-IP protocol applications to be supported in a secure manner while still leveraging the access infrastructure of the Internet.

It is the purpose of this RFC to identify the issues encountered in integrating multiprotocol dial-up services into an existing Internet Service Provider's Point of Presence (hereafter referred to as ISP and POP, respectively), and to describe the L2F protocol which permits the leveraging of existing access protocols.

1.1. Conventions

The following language conventions are used in the items of specification in this document:

- MUST, SHALL, or MANDATORY -- This item is an absolute requirement of the specification.

- SHOULD or RECOMMEND -- This item should generally be followed for all but exceptional circumstances.

- MAY or OPTIONAL -- This item is truly optional and may be followed or ignored according to the needs of the implementor.

2.0 Problem Space Overview

In this section we describe in high level terms the scope of the problem that will be explored in more detail in later sections.

2.1 Initial Assumptions

We begin by assuming that Internet access is provided by an ISP and that the ISP wishes to offer services other than traditional registered IP address based services to dial-up users of the network.

We also assume that the user of such a service wants all of the security facilities that are available to him in a dedicated dial-up configuration. In particular, the end user requires:

+ End System transparency: Neither the remote end system nor his home site hosts should require any special software to use this service in a secure manner.

+ Authentication as provided via dial-up PPP CHAP or PAP, or through other dialogs as needed for protocols without authentication (e.g., SLIP). This will include TACACS+ and RADIUS solutions as well as support for smart cards and one-time passwords. The authentication should be manageable by the user independently of the ISP.

+ Addressing should be as manageable as dedicated dial-up solutions. The address should be assigned by the home site and not the ISP.

+ Authorization should be managed by the home site as it would in a direct dial-up solution.

+ Accounting should be performed both by the ISP (for billing purposes) and by the user (for charge-back and auditing).

2.2 Topology

Shown below is a generic Internet with Public switched Telephone Network (PSTN) access (i.e., async PPP via modems) and Integrated Services Digital Network (ISDN) access (i.e., synchronous PPP access). Remote users (either async PPP or SLIP, or ISDN) will access the Home LAN as if they were dialed into the Home Gateway, although their physical dial-up is via the ISP Network Access Server.

```
           ...----[L]----+---[L]-------...
                         |
                         |
                        [H]
                         |
                 ________|________________________
                 |                                |
         ________|__                        ______|________
         |         |                        |             |
         |  PSTN  [R]                      [R]  ISDN      |
         |  Cloud  |                        |   Cloud    [N]__[U]
         |         |             Internet   |             |
         |         |                       [R]            |
         [N]______[R]                       |_____________|
          |      |                                |
          |      |                                |
         [U]     |________________________________|
```

```
      [H] = Home Gateway
      [L] = Home LAN(s)
      [R] = Router
      [U] = Remote User
      [N] = ISP Network Access Server ("NAS")
```

2.3 Providing Virtual dial-up Services - a walk-through

To motivate the following discussion, this section walks through an example of what might happen when a Virtual dial-up client initiates access.

The Remote User initiates a PPP connection to an ISP via either the PSTN or ISDN. The Network Access Server (NAS) accepts the connection and the PPP link is established.

The ISP undertakes a partial authentication of the end system/user via CHAP or PAP. Only the username field is interpreted to determine whether the user requires a Virtual dial-up service. It is expected-- but not required--that usernames will be structured (e.g. littlewo@cisco.com). Alternatively, the ISP may maintain a database mapping users to services. In the case of Virtual dial-up, the mapping will name a specific endpoint, the Home Gateway.

If a virtual dial-up service is not required, standard access to the Internet may be provided.

If no tunnel connection currently exists to the desired Home Gateway, one is initiated. L2F is designed to be largely insulated from the details of the media over which the tunnel is established; L2F requires only that the tunnel media provide packet oriented point-to-point connectivity. Obvious examples of such media are UDP, Frame Relay PVC's, or X.25 VC's. Details for L2F operation over UDP are provided in section 5.5. The specification for L2F packet formats is provided in section 4.2, and the message types and semantics starting in section 4.4.

Once the tunnel exists, an unused Multiplex ID (hereafter, "MID") is allocated, and a connect indication is sent to notify the Home Gateway of this new dial-up session. The Home Gateway either accepts the connection, or rejects. Rejection may include a reason indication, which may be displayed to the dial-up user, after which the call should be disconnected.

The initial setup notification may include the authentication information required to allow the Home Gateway to authenticate the user and decide to accept or decline the connection. In the case of CHAP, the set-up packet includes the challenge, username and raw response. For PAP or text dialog (i.e., for SLIP users), it includes username and clear text password. The Home Gateway may choose to use this information to complete its authentication, avoiding an additional cycle of authentication.

For PPP, the initial setup notification may also include a copy of the the LCP CONFACKs sent in each direction which completed LCP negotiation. The Home Gateway may use this information to initialize its own PPP state (thus avoiding an additional LCP negotiation), or it may choose to initiate a new LCP CONFREQ exchange.

If the Home Gateway accepts the connection, it creates a "virtual interface" for SLIP or PPP in a manner analogous to what it would use for a direct-dialed connection. With this "virtual interface" in place, link layer frames may now pass over this tunnel in both directions. Frames from the remote user are received at the POP, stripped of any link framing or transparency bytes, encapsulated in L2F, and forwarded over the appropriate tunnel.

The Home Gateway accepts these frames, strips L2F, and processes them as normal incoming frames for the appropriate interface and protocol. The "virtual interface" behaves very much like a hardware interface, with the exception that the hardware in this case is physically located at the ISP POP. The other direction behaves analogously, with the Home Gateway encapsulating the packet in L2F, and the POP stripping L2F before transmitting it out the physical interface to the remote user.

At this point, the connectivity is a point-to-point PPP or SLIP connection whose endpoints are the remote user’s networking application on one end and the termination of this connectivity into the Home Gateway’s SLIP or PPP support on the other. Because the remote user has become simply another dial-up client of the Home Gateway access server, client connectivity can now be managed using traditional mechanisms with respect to further authorization, protocol access, and filtering.

Accounting can be performed at both the NAS as well as the Home Gateway. This document illustrates some Accounting techniques which are possible using L2F, but the policies surrounding such Accounting are outside the scope of this specification.

Because L2F connect notifications for PPP clients contain sufficient information for a Home Gateway to authenticate and initialize its LCP state machine, it is not required that the remote user be queried a second time for CHAP authentication, nor that the client undergo multiple rounds of LCP negotiation and convergence. These techniques are intended to optimize connection setup, and are not intended to deprecate any functions required by the PPP specification.

3.0 Service Model Issues

There are several significant differences between the standard Internet access service and the Virtual dial-up service with respect to authentication, address allocation, authorization and accounting. The details of the differences between these services and the problems presented by these differences are described below. The mechanisms used for Virtual Dial-up service are intended to coexist with more traditional mechanisms; it is intended that an ISP’s POP can simultaneously service ISP clients as well as Virtual dial-up clients.

3.1 Security

For the Virtual dial-up service, the ISP pursues authentication only to the extent required to discover the user’s apparent identity (and by implication, their desired Home Gateway). As soon as this is determined, a connection to the Home Gateway is initiated with the authentication information gathered by the ISP. The Home Gateway completes the authentication by either accepting the connection, or rejecting it.

The Home Gateway must also protect against attempts by third parties to establish tunnels to the Home Gateway. Tunnel establishment involves an ISP-to-Home Gateway authentication phase to protect against such attacks.

3.2 Address Allocation

For an Internet service, the user accepts that the IP address may be allocated dynamically from a pool of Service provider addresses. This model often means that the remote user has little or no access to their home network’s resources, due to firewalls and other security policies applied by the home network to accesses from external IP addresses.

For the Virtual dial-up service, the Home Gateway can exist behind the home firewall, allocating addresses which are internal (and, in fact, can be RFC1597 addresses, or non-IP addresses). Because L2F tunnels exclusively at the frame layer, the actual policies of such address management are irrelevant to correct Virtual dial-up service; for all purposes of PPP or SLIP protocol handling, the dial-in user appears to have connected at the Home Gateway.

3.3 Authentication

The authentication of the user occurs in three phases; the first at the ISP, and the second and optional third at the Home gateway.

The ISP uses the username to determine that a Virtual dial-up service is required and initiate the tunnel connection to the appropriate Home Gateway. Once a tunnel is established, a new MID is allocated and a session initiated by forwarding the gathered authentication information.

The Home Gateway undertakes the second phase by deciding whether or not to accept the connection. The connection indication may include CHAP, PAP, or textual authentication information. Based on this information, the Home Gateway may accept the connection, or may reject it (for instance, it was a PAP request and the username/password are found to be incorrect).

Once the connection is accepted, the Home Gateway is free to pursue a third phase of authentication at the PPP or SLIP layer. These activities are outside the scope of this specification, but might include an additional cycle of LCP authentication, proprietary PPP extensions, or textual challenges carried via a TCP/IP telnet session.

3.4 Accounting

It is a requirement that both the Access gateway and the Home Gateway can provide accounting data and hence both may count packets, octets and connection start and stop times.

Since Virtual dial-up is an access service, accounting of connection attempts (in particular, failed connection attempts) is of significant interest. The Home Gateway can reject new connections based on the authentication information gathered by the ISP, with corresponding logging. For cases where the Home Gateway accepts the connection and then continues with further authentication, the Home Gateway might subsequently disconnect the client. For such scenarios, the disconnection indication back to the ISP may also include a reason.

Because the Home Gateway can decline a connection based on the authentication information collected by the ISP, accounting can easily draw a distinction between a series of failed connection attempts and a series of brief successful connections. Lacking this facility, the Home Gateway must always accept connection requests, and would need to exchange a number of PPP packets with the remote system.

4.0 Protocol Definition

The protocol definition for Virtual dial-up services requires two areas of standardization:

+ Encapsulation of PPP packets within L2F. The ISP NAS and the Home gateway require a common understanding of the encapsulation protocol so that SLIP/PPP packets can be successfully transmitted and received across the Internet.

+ Connection management of L2F and MIDs. The tunnel must be initiated and terminated, as must MIDs within the tunnel. Termination includes diagnostic codes to assist in the diagnosis of problems and to support accounting.

While providing these services, the protocol must address the following required attributes:

+ Low overhead. The protocol must impose a minimal additional overhead. This requires a compact encapsulation, and a structure for omitting some portions of the encapsulation where their function is not required.

+ Efficiency. The protocol must be efficient to encapsulate and deencapsulate.

+ Protocol independence. The protocol must make very few assumptions about the substrate over which L2F packets are carried.

- + Simple deployment. The protocol must not rely on additional telecommunication support (for instance, unique called numbers, or caller ID) to operate.

4.1 Encapsulation within L2F

4.1.1 Encapsulation of PPP within L2F

The PPP packets may be encapsulated within L2F. The packet encapsulated is the packet as it would be transmitted over a physical link. The following are NOT present in the packet:

- + Flags
- + Transparency data (ACCM for async, bit stuffing for sync)
- + CRC

The following ARE still present:

- + Address and control flags (unless negotiated away by LCP)
- + Protocol value

4.1.2 Encapsulation of SLIP within L2F

SLIP is encapsulated within L2F in much the same way as PPP. The transparency characters are removed before encapsulating within L2F, as is the framing.

4.2 L2F Packet Format

4.2.1 Overall Packet Format

The entire encapsulated packet has the form:

```
                 ---------------------------------
                 |                               |
                 |          L2F Header           |
                 |                               |
                 ---------------------------------
                 |                               |
                 |  Payload packet (SLIP/PPP)    |
                 |                               |
                 ---------------------------------
                 |                               |
                 |    L2F Checksum (optional)    |
                 |                               |
                 ---------------------------------
```

4.2.2 Packet Format

An L2F packet has the form:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|K|P|S|0|0|0|0|0|0|0|0|C| Ver |    Protocol   |Sequence (opt)|\
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
|          Multiplex ID         |           Client ID           | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | L2F
|             Length            |       Offset (opt)            | |Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
|                         Key (opt)                             | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/
+                                 (payload)                     |
+                             .....                             |
+                             .....                             |
+                             .....                             |
+                                 (payload)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   L2F Checksum (optional)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.2.3 Version field

The Ver ("Version") field represents the major version of the L2F software creating the packet. It MUST contain the value 001.

If Ver holds a value other than 1, or any bits are non-zero after bit S but before bit C, this corresponds to a packet containing extensions not understood by the receiving end. The packet is handled as an invalid packet as defined in 4.4.1.

4.2.4 Protocol field

The Protocol specifies the protocol carried within the L2F packet. Legal values (represented here in hexadecimal) are:

Value	Type	Description
0x00	L2F_ILLEGAL	Illegal
0x01	L2F_PROTO	L2F management packets
0x02	L2F_PPP	PPP tunneled inside L2F
0x03	L2F_SLIP	SLIP tunneled inside L2F

If a packet is received with a Protocol of L2F_ILLEGAL or any other unrecognized value, it MUST be treated as an illegal packet as defined in 4.4.1.

4.2.5 Sequence Number

The Sequence number is present if the S bit in the L2F header is set to 1. This bit MUST be 1 for all L2F management packets. It MAY be set to 1 for non-L2F management packets. If a non-L2F management packet is received with the S bit set, all future L2F packets sent for that MID MUST have the S bit set (and, by implication, be sent using sequence numbers). For instance, the Home Gateway might choose to force sequenced packet delivery if it detects an NCP opening for a protocol which can not operate with out-of-sequence packets.

The Sequence number starts at 0 for the first sequenced L2F packet. Each subsequent packet is sent with the next increment of the sequence number. The sequence number is thus a free running counter represented modulo 256. There is distinct Sequence number state (i.e., counter) for each distinct MID value.

For packets with S bit and sequence number, the sequence number is used to protect against duplication of packets, as follows:

The receiving side of the tunnel records the sequence number of each valid L2F packet it receives. If a received packet appears to have a value less than or equal to the last received value, the packet MUST be silently discarded. Otherwise, the packet is accepted and the sequence number in the packet recorded as the latest value last received.

For purposes of detecting duplication, a received sequence value is considered less than or equal to the last received value if its value lies in the range of the last value and its 127 successor values. For example, if the last received sequence number was 15, then packets with sequence numbers 0 through 15, as well as 144 through 255, would be considered less than or equal to, and would be silently discarded. Otherwise it would be accepted.

4.2.6 Packet Multiplex ID

The Multiplex ID ("MID") identifies a particular connection within the tunnel. Each new connection is assigned a MID currently unused within the tunnel. It is recommended that the MID cycle through the entire 16-bit namespace, to reduce aliasing between previous and current sessions. A MID value which has been previously used within a tunnel, has been closed, and will now be used again, must be considered as an entirely new MID, and initialised as such.

The MID with value 0 is special; it is used to communicate the state of the tunnel itself, as distinct from any connection within the tunnel. Only L2F_PROTO packets may be sent using an MID of 0; if any other type is sent on MID 0, the packet is illegal and MUST be processed as defined in 4.4.1.

4.2.7 Client ID

The Client ID ("CLID") is used to assist endpoints in demultiplexing tunnels when the underlying point-to-point substrate lacks an efficient or dependable technique for doing so directly. Using the CLID, it is possible to demultiplex multiple tunnels whose packets arrive over the point-to-point media interleaved, without requiring media-specific semantics.

When transmitting the L2F_CONF message (described below), the peer's CLID must be communicated via the Assigned_CLID field. This MUST be a unique non-zero value on the sender's side, which is to be expected in the Home Gateway's L2F_CONF response, as well as all future non-L2F_CONF packets received.

The CLID value from the last valid L2F_CONF message received MUST be recorded and used as the CLID field value for all subsequent packets sent to the peer.

Packets with an unknown Client ID MUST be silently discarded.

For the initial packet sent during tunnel establishment, where no L2F_CONF has yet been received, the CLID field MUST be set to 0.

Thus, during L2F_CONF each side is told its CLID value. All later packets sent, tagged with this CLID value, serve as a tag which uniquely identifies this peer.

4.2.8 Length

Length is the size in octets of the entire packet, including header, all fields present, and payload. Length does not reflect the addition of the checksum, if one is present. The packet should be silently discarded if the received packet is shorter than the indicated length. Additional bytes present in the packet beyond the indicated length MUST be silently ignored.

4.2.9 Packet Checksum

The Checksum is present if the C bit is present in the header flags. It is a 16-bit CRC as used by PPP/HDLC (specifically, FCS-16 [3]). Is is applied over the entire packet starting with the first byte of L2F flags, through the last byte of payload data. The checksum is then added as two bytes immediately following the last byte of payload data.

4.2.10 Payload Offset

The Offset is present if the F bit is set in the header flags. This field specifies the number of bytes past the L2F header at which the payload data is expected to start. If it is 0, or the F bit is not set, the first byte following the last byte of L2F header is the first byte of payload data.

It is recommended that data skipped due to the payload offset be initialized to 0's.

For architectures where it is more efficient to have the payload start at an aligned 32-bit boundary with respect to the L2F header, it is recommended that the F bit be set, and an offset of 0 be used.

4.2.11 Packet Key

The Key field is present if the K bit is set in the L2F header. The Key is based on the authentication response last given to the peer during tunnel creation (the details of tunnel creation are provided in the next section). It serves as a key during the life of a session to resist attacks based on spoofing. If a packet is received in which the Key does not match the expected value, the packet MUST be silently discarded. Such handling takes precedence over 4.4.1.

The Key value is generated by taking the 128-bit authentication response from the peer, interpreting it as four adjacent 32-bit words in network byte order, XOR'ing these words together, and using the resulting 32-bit value as the Key.

4.2.12 Packet priority

If the P bit in the L2F header is set, this packet is a "priority" packet. When possible for an implementation, a packet received with the P bit should be processed in preference to previously received unprocessed packets without the P bit.

The P bit may be set by an implementation based on criteria beyond the scope of this specification. However, it is recommended that PPP keepalive traffic, if any, be sent with this bit set.

4.3 L2F Tunnel Establishment

When the point-to-point link is first initiated between the NAS and the Home Gateway, the endpoints communicate on MID 0 prior to providing general L2F services to clients. This communication is used to verify the presence of L2F on the remote end, and to permit any needed authentication.

The protocol for such negotiation is always 1, indicating L2F management. The message itself is structured as a sequence of single octets indicating an option, followed by zero or more further octets formatted as needed for the option.

4.3.1 Normal Tunnel Negotiation Sequence

The establishment sequence is best illustrated by a "typical" connection sequence. Detailed description of each functions follows, along with descriptions of the handling of exceptional conditions.

Each packet is described as a source->destination on one line, a description of the L2F packet field contents on the next, and the contents of the packet's body on following lines. The exact encoding of octets will be described later.

Note that this example uses the Key option, but does not use the Offset and Checksum options. The Length field would be present, reflecting the actual length of the packets as encoded as an octet stream.

```
   1. NAS->GW:
       Proto=L2F, Seq=0, MID=0, CLID=0, Key=0
       L2F_CONF
           Name: NAS_name
           Challenge: Rnd
           Assigned_CLID: 22
```

The NAS decides that a tunnel must be initiated from the NAS to the GW. An L2F packet is sent with the Proto field indicating an L2F management message is contained.

Because the tunnel is being initiated, Key is set to 0. The sequence number starts at 0; the MID is 0 to reflect the establishment of the tunnel itself. Since the NAS has not yet received an L2F_CONF, the CLID is set to 0.

The body of the packet specifies the claimed name of the NAS, and a challenge random number which GW will use in authenticating itself as a valid tunnel endpoint. Assigned_CLID is generated to be a value not currently assigned out to any other tunnel to any other Home Gateway.

```
      2. GW->NAS:
          Proto=L2F, Seq=0, MID=0, CLID=22, Key=0
          L2F_CONF
              Name: GW_name
              Challenge: Rnd2
              Assigned_CLID: 73
```

The Home Gateway has processed the previous packet, and sends a response. The protocol continues to be L2F, with a sequence number 0 (each side maintains its own sequence number for transmissions). MID continues to be 0 to reflect tunnel establishment. CLID reflects the Assigned_CLID field of the L2F_CONF received. The Key continues to be 0 during this phase of tunnel establishment.

The body contains the Home Gateway’s name, its own random number challenge, and its own Assigned_CLID for the NAS to place in the CLID field of future packets. The CLID is generated in an analogous manner to that of the NAS. After this, all packets received from the NAS must be tagged with a CLID field containing 73, and all packets sent to the NAS must be tagged with a CLID field containing 22.

```
      3. NAS->GW
          Proto=L2F, Seq=1, MID=0, CLID=73, Key=C(Rnd2)
          L2F_OPEN
              Response: C(Rnd2)
```

The NAS responds with its Key now set to reflect the shared secret. The Key is a CHAP-style hash of the random number received; each packet hereafter will reflect this calculated value, which serves as a key for the life of the tunnel. Both the Home Gateway and the NAS use such Keys for the life of the tunnel. The Key is a 32-bit representation of the MD5 digest resulting from encrypting the shared secret; the full MD5 digest is included in the L2F_OPEN response, in the "response" field.

```
      4. GW->NAS
          Proto=L2F, Seq=1, MID=0, CLID=22, Key=C(Rnd)
          L2F_OPEN
              Response: C(Rnd)
```

The Home Gateway provides closure of the key from the NAS, reflected in both the Key field as well as the "response" field. The tunnel is now available for clients to be established.

4.3.2 Normal Client Negotiation Sequence

This section describes the establishment of a Virtual dial-up client on a NAS into a Home Gateway. It assumes a tunnel has been created in the way described in 4.3.1. The client for this example is a PPP client configured for CHAP.

Treatment of Checksum, Length, and Offset are as in 4.3.1.

```
      1. NAS->GW
          Proto=L2F, Seq=2, MID=1, CLID=73, Key=C(Rnd2)
          L2F_OPEN
              Type: CHAP
              Name: CHAP-name
              Challenge: Rnd3
              Response: <Value received, presumably C(Rnd3)>
              ID: <ID used in challenge>
```

The NAS has received a call, tried CHAP with a challenge value of Rnd3, and found that the client responded. The claimed name lead the NAS to believe it was a Virtual dial-up client hosted by the Home Gateway. The next free MID is allocated, and the information associated with the CHAP challenge/response is included in the connect notification.

```
   2. GW->NAS
       Proto=L2F, Seq=2, MID=1, CLID=22, Key=C(Rnd)
       L2F_OPEN
```

The Home Gateway, by sending back the L2F_OPEN, accepts the client.

```
   3. NAS->GW
       Proto=PPP, Seq=0, MID=1, CLID=73, Key=C(Rnd2)
       <Frame follows>

   4. GW->NAS
       Proto=PPP, Seq=0, MID=1, CLID=22, Key=C(Rnd)
       <Frame follows>
```

Traffic is now free to flow in either direction as sent by the remote client or the home site. The contents is uninterpreted data, HDLC in this case. Data traffic, since it is not the L2F protocol, does not usually use the Seq field, which is set to 0 in non-L2F messages (see the S bit in section 4.2.5 for details on an exception to this).

4.4 L2F management message types

When an L2F packet's Proto field specifies L2F management, the body of the packet is encoded as zero or more options. An option is a single octet "message type", followed by zero or more sub-options. Each sub-option is a single byte sub-option value, and further bytes as appropriate for the sub-option.

Options in L2F are:

Hex Value	Abbreviation	Description
0x00	Invalid	Invalid message
0x01	L2F_CONF	Request configuration
0x02	L2F_CONF_NAME	Name of peer sending L2F_CONF
0x03	L2F_CONF_CHAL	Random number peer challenges with
0x04	L2F_CONF_CLID	Assigned_CLID for peer to use
0x02	L2F_OPEN	Accept configuration
0x01	L2F_OPEN_NAME	Name received from client
0x02	L2F_OPEN_CHAL	Challenge client received
0x03	L2F_OPEN_RESP	Challenge response from client
0x04	L2F_ACK_LCP1	LCP CONFACK accepted from client
0x05	L2F_ACK_LCP2	LCP CONFACK sent to client
0x06	L2F_OPEN_TYPE	Type of authentication used
0x07	L2F_OPEN_ID	ID associated with authentication
0x08	L2F_REQ_LCP0	First LCP CONFREQ from client
0x03	L2F_CLOSE	Request disconnect
0x01	L2F_CLOSE_WHY	Reason code for close
0x02	L2F_CLOSE_STR	ASCII string description
0x04	L2F_ECHO	Verify presence of peer
0x05	L2F_ECHO_RESP	Respond to L2F_ECHO

4.4.1 L2F message type: Invalid

If a message is received with this value, or any value higher than the last recognized option value, or if an illegal packet as defined by other parts of this specification is received, the packet is considered invalid. The packet MUST be discarded, and an L2F_CLOSE of the entire tunnel MUST be requested. Upon receipt of an L2F_CLOSE, the tunnel itself may be closed. All other received message MUST be discarded. An implementation MAY close the tunnel after an interval of time appropriate to the characteristics of the tunnel.

Note that packets with an invalid Key are discarded, but disconnect is not initiated. This prevents denial-of-service attacks. Invalid option types within a message MUST be treated as if the entire message type was invalid.

4.4.2 L2F_CONF

The L2F message type is used to establish the tunnel between the NAS and the Home Gateway. MID is always set to 0. The body of such a message starts with the octet 0x01 (L2F_CONF), followed by all three of the sub-options below.

The L2F_CONF_NAME sub-option MUST be present. It is encoded as the octet value 0x02, followed by an octet specifying a non-zero length, followed by the indicated number of bytes, which are interpreted as the sender's ASCII name.

The L2F_CONF_CHAL sub-option MUST be present. It is encoded as the octet value 0x03, followed by a non-zero octet, followed by a number of bytes specified by this non-zero octet.

The challenge value should be generated using whatever techniques provide the highest quality of random numbers available to a given implementation.

The L2F_CONF_CLID sub-option MUST be present. It is encoded as the octet 0x04, followed by four bytes of Assigned_CLID value. The Assigned_CLID value is generated as a non-zero 16-bit integer value unique across all tunnels which exist on the sending system. The least significant two octets of Assigned_CLID are set to this value, and the most significant two octets MUST be set to 0.

The CLID field is sent as 0 in the initial L2F_CONF packet from NAS to Home Gateway, and otherwise MUST be sent containing the value specified in the Assigned_CLID field of the last L2F_CONF message received.

Key MUST be set to 0 in all L2F_CONF packets, and no key field is included in the packet.

When sent from a NAS to a Home Gateway, the L2F_CONF is the initial packet in the conversation.

When sent from the Home Gateway to the NAS, an L2F_CONF indicates the Home Gateway's recognition of the tunnel creation request. The Home Gateway MUST provide its name and its own challenge in the message body.

In all packets following the L2F_CONF, the Key MUST be set to the CHAP-style hash of the received challenge bytes. The CHAP-style hash is done over the concatenation of the low 8 bits of the assigned CLID, the secret, and the challenge value. Generation of the 32-bit key value is discussed in section 4.2.11.

4.4.3 L2F_OPEN, tunnel establishment

The L2F_OPEN message is used to provide tunnel setup closure (for a MID of 0) or to establish a client connection within a tunnel previously established by L2F_CONF and L2F_OPEN messages (MID not equal to 0). This section describes tunnel establishment; section 4.4.4 following describes clients established within the tunnel.

An L2F_OPEN for tunnel establishment MUST contain only the sub-option 0x03, L2F_OPEN_RESP. This option MUST be followed by the octet 0x10, specifying the size of the 128-bit MD5 digest resulting from encrypting the challenge value in the L2F_CONF, along with the low byte of the Assigned_CLID. After this byte MUST be the sixteen bytes of the generated MD5 digest.

If during tunnel establishment an L2F_OPEN is received with an incorrect L2F_OPEN_RESP, the packet MUST be silently discarded. It is recommended that such an event generate a log event as well.

4.4.4 L2F_OPEN, client establishment

An L2F_OPEN (with non-zero MID) sent from the NAS to the Home Gateway indicates the presence of a new dial-in client. When sent back from the Home Gateway to the NAS, it indicates acceptance of the client. This message starts with the octet 0x02. When sent from the NAS, it may contain further sub-options. When sent from the Home Gateway, it may not contain any sub-options. All further discussion of sub-options in this section apply only to the NAS to Home Gateway direction.

The L2F_OPEN_TYPE sub-option MUST be present. It is encoded as the octet 0x06, followed by a single byte describing the type of authentication the NAS exchanged with the client in detecting the client’s claimed identification. Implicit in the authentication type is the encapsulation to be carried over the life of the session. The authentication types are:

0x01	Textual username/password exchange for SLIP
0x02	PPP CHAP
0x03	PPP PAP
0x04	PPP no authentication
0x05	SLIP no authentication

The L2F_OPEN_NAME sub-option is encoded as the octet 0x01, followed by an octet specifying the length of the name, followed by the indicated number of bytes of the name. This field MUST be present for any authentication type except 0x04 (None). It MUST contain the name specified in the client’s authentication response.

The L2F_OPEN_CHAL sub-option is encoded as the octet 0x02, followed by an octet specifying the length of the challenge sent, followed by the challenge itself. This field is only present for CHAP, and MUST contain the challenge value sent to the client by the NAS.

The L2F_OPEN_RESP sub-option is encoded as the octet 0x03, followed by an octet specifying the length of the response received, followed by the client’s response to the challenge. For CHAP, this field contains the response value received by the NAS. For PAP or textual authentication, it contains the clear text password received from the client by the NAS. This field is absent for authentication 0x04 "None".

The L2F_ACK_LCP1 and L2F_ACK_LCP2 sub-options are encoded as the octets 0x04 and 0x05 respectively, followed in either case by two octets in network byte order specifying the length of the LCP CONFACK last received from or sent to the client. Following these octets is an exact copy of the CONFACK packet. L2F_ACK_LCP1 specifies a copy of the closing CONFACK received from the client, and L2F_ACK_LCP2 specifies a copy of the closing CONFACK sent to the client by the NAS.

The L2F_REQ_LCP0 sub-option is encoded as the octet 0x08, followed by two octets in network byte order specifying the length of the LCP CONFREQ initially received from the client. This may be used by the Home Gateway to detect capabilities of the client which were negotiated away while starting LCP with the NAS. Detection of such options may be used by the Home Gateway to decide to renegotiate LCP.

The L2F_OPEN_ID sub-option is encoded as the octet 0x06, followed by a single octet. This sub-option is only present for CHAP; the single octet contains the CHAP Identifier value sent to the client during the CHAP challenge.

The Home Gateway may choose to ignore any sub-option of the L2F_OPEN, and accept the connection anyway. The Home Gateway would then have to undertake its own LCP negotiations and authentication. To maximize the transparency of the L2F tunnel, it is recommended that extra negotiations and authentication be avoided if possible.

4.4.5 L2F_CLOSE

This message is encoded as the byte 0x03. An L2F_CLOSE may be sent by either side of the tunnel at any time. When sent with MID of 0, it indicates the desire to terminate the entire tunnel and all clients within the tunnel. When sent from the Home Gateway in response to an L2F_OPEN, it indicates that the Home Gateway has declined the connection. When sent with a non-zero MID, it indicates the termination of that client within the tunnel.

The L2F_CLOSE_WHY sub-option is encoded as the byte 0x01 followed four bytes in network byte order specifying a bit mask of reasons for the disconnection. The bits are encoded as:

0x00000001	Authentication failed
0x00000002	Out of resources
0x00000004	Administrative intervention
0x00000008	User quota exceeded
0x00000010	Protocol error
0x00000020	Unknown user
0x00000040	Incorrect password
0x00000080	PPP configuration incompatible
0x00000100	Wrong multilink PPP destination

Bits in the mask 0xFF000000 are reserved for per-vendor interpretation.

An implementation can choose to not provide status bits even if it detects a condition described by one of these bits. For instance, an implementation may choose to not use 0x00000020 due to security considerations, as it can be used to probe user name space.

The L2F_CLOSE_STR sub-option is encoded as the byte 0x02, followed by a two-byte length in network byte order, followed by the indicated number of bytes, which are interpreted as descriptive ASCII text associated with the disconnection. This string may be ignored, but could be recorded in a log to provide detailed or auxiliary information associated with the L2F_CLOSE.

4.4.6 L2F_ECHO

Transmission of L2F_ECHO messages is optional. If an implementation transmits L2F_ECHO messages, it MUST not transmit more than one such request each second. The payload size MUST be 64 bytes or less in length. It is recommended that at least 5 L2F_ECHO messages be sent without response before an implementation assumes that its peer has terminated.

The L2F_ECHO message is encoded as the single byte 0x04. It may be sent by either side once the tunnel is established. MID MUST be 0. An L2F_ECHO_RESP (documented below) MUST be sent back in response.

4.4.7 L2F_ECHO_RESP

All implementations MUST respond to L2F_ECHO, using L2F_ECHO_RESP. The received packet MUST be sent back verbatim, except that the CLID, sequence number, and checksum (if any) MUST be updated, and the L2F_ECHO message type changed to an L2F_ECHO_RESP. Payload data following the 0x04 octet, if any, MUST be preserved in the response.

When an L2F_ECHO_RESP is received, the payload data may be used to associate this response with a previously sent L2F_ECHO, or the packet may be silently discarded.

4.5 L2F Message Delivery

L2F is designed to operate over point-to-point unreliable links. It is not designed to provide flow control of the data traffic, nor does it provide reliable delivery of this traffic; each protocol tunnel carried via L2F is expected to manage flow control and retry itself. Thus, it is only L2F control messages which must be retransmitted; this process is described in this section.

4.5.1 Sequenced delivery

All L2F control messages (i.e., those L2F packets with a protocol type of 0x01) are transmitted with a sequence number. The sequence number is a per-L2F tunnel free running counter which is incremented (modulo 256) after each packet is transmitted. It is used to permit the receiving end to detect duplicated or out-of-order packets, and to discard such packets. Section 4.2.5 describes the process in detail.

4.5.2 Flow control

L2F control messages are expected to be exchanged lock-step. Thus, per-client activities can not occur until tunnel setup is complete. Neither can one client be serviced until the L2F message exchange is complete for a previous client. Thus, it is expected that rarely--if ever--should a flow control action be required. If the input queue of L2F control messages reaches an objectionable level for an implementation, the implementation may silently discard all messages in the queue to stabilize the situation.

4.5.3 Tunnel State table

The following enumerates the handling of L2F messages for tunnel creation in state table format. Events name an L2F_ message type (the L2F_ portion of the named message is omitted to permit a more compact table). A start ("*") matches any event not otherwise matched for the named state.

A NAS starts at initial state Start0, sending a packet before waiting for its first event. A Home Gateway starts at Start1, waiting for an initial packet to start service.

If an event is not matched for a given state, the packet associated with that event is silently discarded.

Tunnel establishment (MID == 0), NAS side.

State	Event	Action	New State
Start0		Send CONF	Start1
Start1	CONF	Send OPEN	Start2
Start1	timeout 1-3	Send CONF	Start1
Start1	timeout 4	Clean up tunnel	(done)
Start2	OPEN	(initiate 1st client)	Open1
Start2	timeout 1-3	Send OPEN	Start2
Start2	timeout 4	Clean up tunnel	(done)
Open1	OPEN	Send OPEN	Open1
Open1	CLOSE	Send CLOSE	Close1
Open1	no MIDs open	Send CLOSE	Close2
Close1	CLOSE	Send CLOSE	Close1
Close1	timeout 4	Clean up tunnel	(done)
Close2	CLOSE	Clean up tunnel	(done)
Close2	timeout 1-3	Send CLOSE	Close2
Close2	timeout 4	Clean up tunnel	(done)

Tunnel establishment (MID == 0), Home Gateway side.

State	Event	Action	New State
Start0	CONF	Send CONF	Start1
Start1	CONF	Send CONF	Start1
Start1	OPEN	Send OPEN	Open1
Start1	timeout 4	Clean up tunnel	(done)
Open1	OPEN	Send OPEN	Open1
Open1	OPEN (MID > 0)	(1st client, below)	Open2
Open1	CLOSE	Send CLOSE	Close1
Open1	timeout 4	Clean up tunnel	(done)

Open2	OPEN (MID > 0)	(below)	Open2
Open2	CLOSE	Send CLOSE	Close1
Close1	CLOSE	Send CLOSE	Close1
Close1	timeout 4	Clean up tunnel	(done)

4.5.4 Client State table

This table is similar to the previous one, but enumerates the states for a client connection within a tunnel in the opened state from 4.5.3. As this sequence addresses clients, MID will be non-zero.

Client establishment (MID != 0), NAS side.

State	Event	Action	New State
Start0		Send OPEN	Start1
Start1	OPEN	(enable forwarding)	Open1
Start1	CLOSE	Clean up MID	(MID done)
Start1	timeout 1-3	Send OPEN	Start1
Start1	timeout 4	Clean up MID	(MID done)
Start1	client done	Send CLOSE	Close2
Open1	OPEN	(no change)	Open1
Open1	CLOSE	Send CLOSE	Close1
Open1	client done	Send CLOSE	Close2
Close1	CLOSE	Send CLOSE	Close1
Close1	timeout 4	Clean up MID	(MID done)
Close2	CLOSE	Clean up MID	(MID done)
Close2	timeout 1-3	Send CLOSE	Close2
Close2	timeout 4	Clean up MID	(MID done)

Client establishment (MID != 0), Home Gateway side.

State	Event	Action	New State
Start0	OPEN	Send OPEN	Open1
Start0	OPEN (fail)	Send CLOSE	Close3
Open1	OPEN	Send OPEN	Open1
Open1	CLOSE	Send CLOSE	Close1
Open1	client done	Send CLOSE	Close2
Close1	CLOSE	Send CLOSE	Close1
Close1	timeout 4	Clean up MID	(MID done)
Close2	CLOSE	Clean up MID	(MID done)
Close2	timeout 1-3	Send CLOSE	Close2
Close2	timeout 4	Clean up MID	(MID done)
Close3	OPEN	Send CLOSE	Close3
Close3	timeout 4	Clean up MID	(MID done)

5. Protocol Considerations

Several aspects of operation over L2F, while outside the realm of the protocol description itself, serve to clarify the operation of L2F.

5.1 PPP Features

Because L2F in operation carries uninterpreted frames, it permits operation of features without explicit knowledge of these features. For instance, if a PPP session is carried, L2F is simply transporting HDLC frames. The two PPP endpoints can negotiate higher-level features, such as reliable link, compression, multi-link, or encryption. These features then operate between the two PPP endpoints (the dial-in client on one end, and the Home Gateway on the other), with L2F continuing to simply ship HDLC frames back and forth.

For similar reasons, PPP echo requests, NCP configuration negotiation, and even termination requests, are all simply tunneled HDLC frames.

5.2 Termination

As L2F simply tunnels link-layer frames, it does not detect frames like PPP TERMREQ. L2F termination in these scenarios is driven from a protocol endpoint; for instance, if a Home Gateway receives a TERMREQ, its action will be to "hang up" the PPP session. It is the responsibility of the L2F implementation at the Home Gateway to convert a "hang up" into an L2F_CLOSE action, which will shut down client’s session in the tunnel cleanly. L2F_CLOSE_WHY and L2F_CLOSE_STR may be included to describe the reason for the shutdown.

5.3 Extended Authentication

L2F is compatible with both PAP and CHAP protocols. SLIP does not provide authentication within the protocol itself, and thus requires an ASCII exchange of username and password before SLIP is started. L2F is compatible with this mode of operation as well.

One-time password cards have become very common. To the extent the NAS can capture and forward the one-time password, L2F operation is compatible with password cards. For the most general solution, an arbitrary request/response exchange must be supported. In an L2F environment, the protocol must be structured so that the NAS can detect the apparent identity of the user and establish a tunnel connection to the Home Gateway, where the arbitrary exchange can occur.

5.4 MNP4 and Apple Remote Access Protocol

L2F appears compatible with Apple’s ARAP protocol. Its operation under L2F has not been described simply because this experimental RFC does not have a corresponding implementation of such operation.

5.5 Operation of IP and UDP

L2F tries to be self-describing, operating at a level above the particular media over which it is carried. However, some details of its connection to media are required to permit interoperable implementations. This section describes the issues which have been found when operating L2F over IP and UDP.

L2F uses the well-known UDP port 1701 [4]. The entire L2F packet, including payload and L2F header, is sent within a UDP datagram. The source and destination ports are the same (1701), with demultiplexing being achieved using CLID values. It is legal for the source IP address of a given CLID to change over the life of a connection, as this may correspond to a peer with multiple IP interfaces responding to a network topology change. Responses should reflect the last source IP address for that CLID.

IP fragmentation may occur as the L2F packet travels over the IP substrate. L2F makes no special efforts to optimize this. A NAS implementation MAY cause its LCP to negotiate for a specific MRU, which could optimize for NAS environments in which the MTUs of the path over which the L2F packets are likely to travel have a consistent value.

6.0 Acknowledgments

L2F uses a packet format inspired by GRE [5]. Thanks to Fred Baker for consultation, Dave Carrel for consulting on security aspects, and to Paul Traina for philosophical guidance.

7.0 References

[1] Romkey, J., "A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP", RFC 1055, June 1988.

[2] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[3] Simpson, W., "PPP in HDLC-like Framing", STD 51,, RFC 1662, July 1994.

[4] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.

[5] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 1701, October 1994.

8.0 Security Considerations

Security issues are discussed in Section 3.1.

9.0 Authors' Addresses

Tim Kolar
Cisco Systems
170 West Tasman Drive
San Jose CA 95134-1706

EMail: tkolar@cisco.com

Morgan Littlewood
Cisco Systems
170 West Tasman Drive
San Jose CA 95134-1706

EMail: littlewo@cisco.com

Andy Valencia
Cisco Systems
170 West Tasman Drive
San Jose CA 95134-1706

EMail: valencia@cisco.com

9.0 Full Copyright Statement