This chapter covers the following topics:

- Label Distribution Protocol (LDP)
- AToM operations

CHAPTER 6

Understanding Any Transport over MPLS

To provide Layer 2 VPN services over an IP/Multiprotocol Label Switching (MPLS) network infrastructure, the Internet Engineering Task Force (IETF) developed a series of solution and protocol specifications for various Layer 2 VPN applications, including pseudowire emulation. Based on the pseudowire emulation specifications, Any Transport over MPLS (AToM) is implemented as part of the Cisco Unified VPN Suite Solution. The Cisco solution also includes alternative pseudowire emulation using Layer 2 Tunnel Protocol Version 3 (L2TPv3). Chapter 3, "Layer 2 VPN Architectures," outlines the benefits and implications of using each technology and highlights some important factors that help network planners and operators determine the appropriate technology.

This chapter starts with an overview of LDP used by pseudowire emulation over MPLS, followed by an explanation of the protocol specifications and operations of AToM. You learn the general properties of the pseudowire emulation over MPLS networks specified in IETF documents. Additional features that AToM supports are also highlighted in this chapter.

Introducing the Label Distribution Protocol

One of the fundamental tasks in the MPLS architecture is to exchange labels between label switch routers (LSR) and define the semantics of these labels. LSRs follow a set of procedures, known as label distribution protocol, to accomplish this task. A label distribution protocol can be an existing protocol with MPLS label extensions or a new protocol that is specifically designed for this purpose. Although the MPLS architecture allows different label distribution protocols, only LDP is used as the signaling protocol for AToM.

NOTE In most MPLS literature, it is common to refer to *label distribution protocol* in lowercase when referring to any protocol that performs label distribution procedures and reserve the *abbreviation LDP* for the specific protocol Label Distribution Protocol, as defined in RFC 3036, "LDP Specification."

The next few sections review some fundamental LDP specifications and operations that are relevant to AToM:

- LDP protocol components
- Discovery mechanisms
- Session establishment
- Label distribution and management
- LDP security

LDP Protocol Components

To have a firm understanding of the protocol operations of LDP, you need to be familiar with the key terminology and protocol entities that are defined in LDP.

LDP peers are two LSRs that use LDP to exchange label information. An LSR might have more than one LDP peer, and it establishes an *LDP session* with each LDP peer. An LDP session is always bidirectional, which allows both LDP peers to exchange label information. However, using a bidirectional signaling session does not make the label-switched path (LSP) bidirectional. As described in Chapter 3, an LSP is unidirectional, and a pseudowire consists of two LSPs of the opposite directions. Besides directly connected LSRs, LDP sessions can be established between non-directly connected LSRs, which are further explained in the later section titled "LDP Extended Discovery."

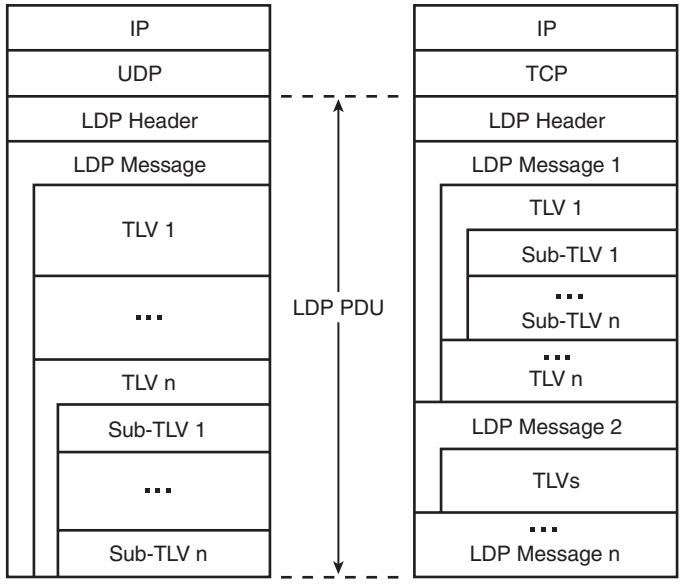
Label space specifies the label assignment. The two types of label space are as follows:

- **Per-interface label space**—Assigns labels from an interface-specific pool of labels. This space typically uses interface resources for labels. For example, a label-controlled ATM interface uses virtual path identifiers (VPI) and virtual circuit identifiers (VCI) as labels.
- **Per-platform label space**—Assigns labels from a platform-wide pool of labels and typically uses resources that are shared across the platform. Hop-by-hop best-effort IP/MPLS forwarding is an example of using the per-platform label space.

In Chapter 3, the AToM overview explains the use of label stacking. To recap, the label stack of AToM typically consists of two labels: tunnel label and pseudowire label. Tunnel labels can be from either per-interface label space or per-platform label space depending on whether the LSRs perform IP/MPLS forwarding in cell mode or frame mode. Pseudowire labels are always allocated from the general-purpose per-platform label space.

LDP uses User Datagram Protocol (UDP) and TCP to transport the protocol data unit (PDU) that carries LDP messages. Figure 6-1 illustrates the structure of an LDP packet. Each LDP PDU is an LDP header followed by one or more LDP messages. All LDP messages have a common LDP message header followed by one or more structured parameters that use a type, length, value (TLV) encoding scheme. The Value field of a TLV might consist of one or more sub-TLVs.

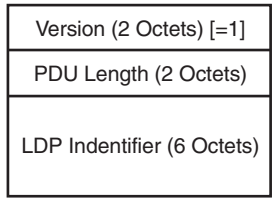
Figure 6-1 *LDP Packet Structure*



The LDP header consists of the following fields, as depicted in Figure 6-2:

- **Version**—The Version field is 2 octets containing the version number of the protocol. The current LDP version is version 1.
- **PDU Length**—The PDU Length field is a 2-octet integer specifying the total length of this PDU in octets, excluding the Version and PDU Length fields. The maximum PDU Length can be negotiated during LDP session initialization.
- **LDP Identifier**—An LDP Identifier consists of 6 octets and identifies an LSR label space. The first 4 octets are a globally unique value that identifies the LSR. The globally unique value is usually the 32-bit router ID of the LSR. The last 2 octets identify a specific label space within the LSR. A zero value of the last 2 octets represents the platform-wide label space. When an LSR uses LDP to advertise more than one label space to another LSR, it creates a separate LDP session for each label space.

Figure 6-2 *LDP Header Format*

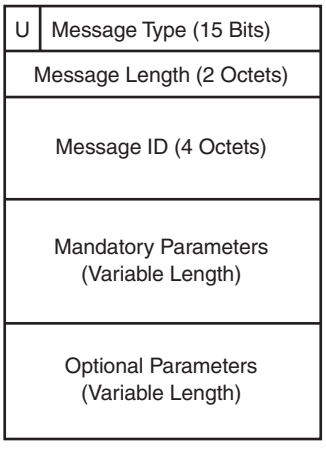


Four categories exist for LDP messages:

- **Discovery messages**—Provide a mechanism in which LSRs indicate their presence in a network by sending Hello messages periodically. Discovery messages include the LDP Link Hello message and the LDP Targeted Hello message. You learn more about discovery messages in the next section "Discovery Mechanisms."
- **Session messages**—Establish, maintain, and disconnect sessions between LDP peers. Session messages are LDP Initialization messages and Keepalive messages. You learn more about session messages in the section "Session Establishment" later in this chapter.
- **Advertisement messages**—Create, update, and delete label mappings. All LDP Address messages and LDP Label messages belong to advertisement messages.
- **Notification messages**—Provide advisory information and signal error information to LDP peers.

Except for discovery messages that use UDP as the underlying transport, LDP messages rely on TCP to ensure reliable and in-order delivery of messages. All LDP messages have the format that is depicted in Figure 6-3.

Figure 6-3 *LDP Message Format*



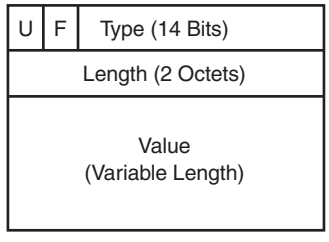
The following fields make up the LDP message format:

- **Unknown Message Bit (U-Bit)**—The U-bit tells the receiver of the message what action to take if he does not understand the message. If the U-bit is set to 0, the receiver needs to respond to the originator of the message with a notification message. Otherwise, the receiver should silently ignore this unknown message.
- **Message Type**—The Message Type field identifies the type of message.
- **Message Length**—The Message Length field specifies the total number of octets of the Message ID, Mandatory Parameters, and Optional Parameters.

- **Message ID**—The Message ID field is a 4-octet value that identifies individual messages.
- **Mandatory Parameters**—The Mandatory Parameters field is a set of required parameters with variable lengths that pertain to this message. Some messages do not have mandatory parameters.
- **Optional Parameters**—The Optional Parameters field is a set of optional parameters that have variable lengths. Many messages do not have optional parameters.

Most information that is carried in an LDP message is encoded in TLVs. TLV provides a generic and extensible encoding scheme for existing and future applications that use LDP signaling. An LDP TLV consists of a 2-bit Flag field, a 14-bit Type field, and a 2-octet Length field, followed by a variable length Value field. Figure 6-4 shows the common TLV encoding scheme.

Figure 6-4 *LDP TLV Encoding*



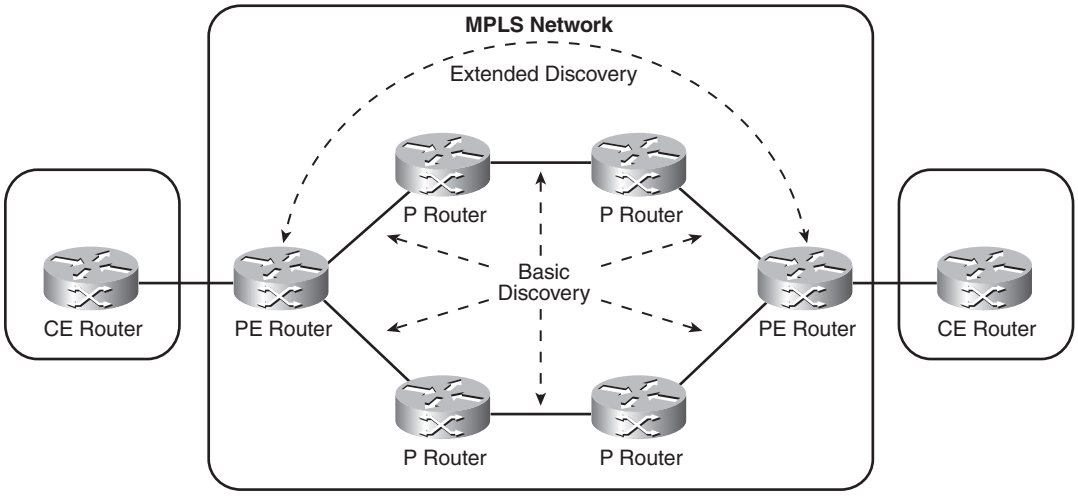
Like the unknown message bit, the unknown TLV bit (U-bit) tells the receiver whether it should send a notification message to the originator if the receiver does not understand the TLV. If the U-bit is set to 0, the receiver must respond with a notification message and discard the entire message. Otherwise, the unknown TLV is silently ignored and the rest of the message is processed as if the unknown TLV does not exist.

The forward unknown TLV bit (F-bit) applies only when the U-bit is set to 1 and the TLV is unknown to the receiver. If the F-bit is set to 0, the unknown TLV is not forwarded. Otherwise, it is forwarded with the containing message.

Discovery Mechanisms

LSRs use LDP discovery procedures to locate possible LDP peers. The basic discovery mechanism identifies directly connected LDP peers. The extended discovery mechanism identifies non-directly connected LDP peers. LSRs discover LDP peers by exchanging LDP Hello messages. As you learned in the previous section, two types of LDP Hello messages exist. LDP Link Hellos are used for LDP basic discovery, and LDP Targeted Hellos are used for LDP extended discovery. Figure 6-5 illustrates where LDP basic discovery and LDP extended discovery occur in an MPLS network.

Figure 6-5 *LDP Basic and Extended Discovery*



LDP Basic Discovery

With LDP basic discovery enabled on an interface, an LSR periodically sends LDP Link Hello messages out the interface. LDP Link Hellos are encapsulated in UDP packets and sent to the well-known LDP discovery port 646 with the destination address set to the multicast group address 224.0.0.2. This multicast address represents all routers on this subnet.

An LDP Link Hello message that an LSR sends carries the LDP identifier for the label space that the LSR intends to use for the interface and other information, such as Hello hold time. When the LSR receives an LDP Link Hello on an interface, it creates a Hello adjacency to keep track of a potential LDP peer reachable at the link level on the interface and learns the label space that the peer intends to use for the interface.

LDP Extended Discovery

For some MPLS applications such as AToM, exchanging label information between non-directly connected LSRs is necessary. Before establishing LDP sessions between non-directly connected LSRs, the LSRs engage in LDP extended discovery by periodically sending Targeted Hello messages to a specific address. LDP Targeted Hello messages are encapsulated in UDP packets and sent to the well-known LDP discovery port 646 with a specific unicast address.

An LDP Targeted Hello message that an LSR sends carries the LDP Identifier for the label space that the LSR intends to use and other information. When the receiving LSR receives an LDP Targeted Hello, it creates a Hello adjacency with a potential LDP peer reachable at the network level and learns the label space that the peer intends to use.

When an LSR sends LDP a Targeted Hello to a receiving LSR, the receiving LSR can either accept the Targeted Hello or ignore it. The receiving LSR accepts the Targeted Hello by creating a Hello adjacency with the originating LSR and periodically sending Targeted Hellos to it.

Session Establishment

After two LSRs exchange LDP discovery Hello messages, they start the process of session establishment, which proceeds in two sequential phases:

1. Transport connection establishment
2. Session initialization

The objective of the transport connection establishment phase is to establish a reliable TCP connection between two LDP peers. If both LDP peers initiate an LDP TCP connection, it might result in two concurrent TCP connections. To avoid this situation, an LSR first determines whether it should play the active or passive role in session establishment by comparing its own transport address with the transport address it obtains through the exchange of LDP Hellos. If its address has a higher value, it assumes the active role. Otherwise, it is passive. When an LSR plays the active role, it initiates a TCP connection to the LDP peer on the well-known LDP TCP port 646.

After the LSR establishes the TCP connection, session establishment proceeds to the session initialization phase. In this phase, LDP peers exchange and negotiate session parameters such as the protocol version, label distribution methods, timer values, label ranges, and so on.

If an LSR plays the active role, it starts the negotiation of session parameters by sending an Initialization message to its LDP peer. The Initialization message carries both the LDP Identifier for the label space of the active LSR and the LDP Identifier of the passive LSR. The receiver compares the LDP Identifier with the Hello adjacencies created during LDP discovery. If the receiver finds a match and the session parameters are acceptable, it replies with an Initialization message with its own session parameters and a Keepalive message to acknowledge the sender's parameters. When the sender receives an Initialization message with acceptable session parameters, it responds with a Keepalive message.

When both LDP peers exchange Initialization and Keepalive messages with each other, the session initialization phase is completed successfully and the LDP session is considered operational.

Label Distribution and Management

Label distribution and management consist of different control, retention, and advertisement modes. Even though it is possible to use an arbitrary permutation for an MPLS application, a certain combination of control, retention, and advertisement modes is usually more preferable or appropriate for a particular MPLS application.

The next few sections explain the following aspects in label distribution and management:

- Label binding
- Label advertisement message
- Label advertisement mode
- Label distribution control mode
- Label retention mode

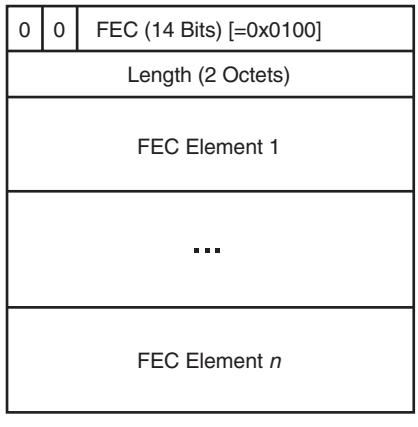
Label Binding

The main focus of an MPLS application is the distribution and management of label bindings. Label bindings are always the centerpiece of information in LDP signaling.

LDP associates a *Forwarding Equivalence Class* (FEC) with each LSP that it creates. An FEC specifies which packets should be forwarded through the associated LSP. Each FEC is defined as a collection of one or more FEC elements. Each FEC element identifies a set of packets that are mapped to the corresponding LSP. For those who are familiar with IP routing, you can consider an FEC as a set of IP routes following a common forwarding path, and an FEC element as a specific IP route prefix.

A label binding is the association between an FEC and a label that represents a specific LSP. The association is created by placing an FEC TLV and a Label TLV in a label advertisement message. Figure 6-6 depicts the FEC TLV encoding.

Figure 6-6 *FEC TLV Encoding*



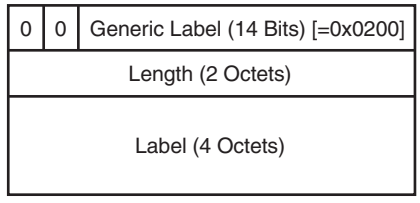
For FEC element 1 to FEC element *n*, the first octet in the FEC element indicates the FEC element type. The encoding scheme of the FEC element varies depending on the FEC element type, such as address prefix and host address. MPLS pseudowire emulation applications such as AToM use the Pseudowire ID FEC element.

Several different types of Label TLV encodings are available, including the following:

- Generic Label TLV
- ATM Label TLV
- Frame Relay Label TLV

Generic Label TLV carries a label from the platform-wide label space and is the most common encoding among MPLS applications (see Figure 6-7).

Figure 6-7 *Generic Label TLV Encoding*



Generic Label TLV has a type of 0x0200. A label is a 20-bit label value in a 4-octet Label field.

LDP Advertisement Message

Label bindings are exchanged through LDP advertisement messages. The advertisement messages that are most relevant to pseudowire emulation over MPLS application are these

- Label Mapping
- Label Request
- Label Withdraw
- Label Release

Label Mapping messages advertise label bindings to LDP peers. A Label Mapping message contains one FEC TLV and one Label TLV. Each FEC TLV might have one or more FEC elements depending on the type of application, and each Label TLV has one label.

When an LSR needs a label binding for a specific FEC but does not already have it, it can explicitly request this label binding from its LDP peer by sending a Label Request message. A Label Request message contains the FEC for which a label is being requested. The receiving LSR then responds to a Label Request message with a Label Mapping message for the requested FEC if it has such a binding. Otherwise, it responds with a Notification message indicating why it cannot satisfy the request.

Whereas Label Mapping messages create the bindings between FECs and labels, Label Withdraw messages break them. An LSR sends a Label Withdraw message to an LDP peer to signal that the peer should not continue to use specified label bindings that the LSR previously advertised. A Label Withdraw message contains the FEC for which the label binding is being

withdrawn and optionally the originally advertised label. If no Label TLV is included in a Label Withdraw message, all labels that are associated with the FEC are to be withdrawn. Otherwise, only the label that is specified in the Label TLV is to be withdrawn.

An LSR that receives a Label Withdraw message must acknowledge it with a Label Release message. The LSR also uses Label Release messages to indicate that it no longer needs specific label bindings previously requested of or advertised by its LDP peer. A Label Release message contains the FEC for which the label binding is being released and optionally the originally advertised label. If no Label TLV is included in a Label Release message, all labels that are associated with the FEC are to be released. Otherwise, only the label that is specified in the Label TLV is to be released.

Label Advertisement Mode

The MPLS architecture specifies two label advertisement modes. If an LSR explicitly requests a label binding for a particular FEC from the next-hop LSR of this FEC, it uses *downstream on-demand* label advertisement mode. If an LSR advertises label bindings to its LDP peers that have not explicitly requested them, it uses *downstream unsolicited* advertisement mode.

Choosing which label advertisement mode to use depends on the characteristics of a particular MPLS implementation and application. Between each pair of LDP peers, they must have the same label advertisement mode.

Label Distribution Control Mode

Label distribution control determines how LSPs are established initially, and it has two modes: *independent* and *ordered* label distribution control.

With independent label distribution control, each LSR advertises label bindings to its peers at any time. It does not wait for the downstream or next-hop LSR to advertise the label binding for the FEC that is being distributed in the upstream direction. A consequence of using independent mode is that an upstream label can be advertised before a downstream label is received.

When an LSR is using ordered label distribution control, it cannot advertise a label binding for an FEC unless it has a label binding for the FEC from the downstream or next-hop LSR. It has to wait for the downstream LSR to advertise the label binding for the FEC that is being distributed in the upstream direction. As a result, ordered control makes the label distribution of a given LSP occur sequentially from the last hop of the LSP toward the first hop of the LSP.

Label Retention Mode

When an LSR receives a label binding for an FEC from a peer that is not the next hop for the FEC, it has the option to either store or discard the label binding based on the label retention mode in use.

Conservative label retention keeps only the label bindings that will be used to forward packets. The main advantage is that only the labels that are required for data forwarding are allocated and maintained. Because downstream on-demand advertisement mode is mainly employed when the label space is limited, it is normally used with the conservation label retention mode.

With *liberal label retention*, an LSR keeps every label binding it receives from its LDP peers regardless of whether the peers are the next-hop LSRs for the advertised label binding. The main advantage is that an LSP can be updated quickly when the label forwarding information is changed. Liberal label retention is mainly used where the label space is considered an inexpensive resource. When it is used with downstream unsolicited advertisement mode, liberal label retention reduces the total number of label advertisement messages required to set up LSPs. If an LSR is using conservative retention mode in this scenario, it has to send Label Request messages to the peer for the label bindings that it has discarded during the initial label advertisement if that peer becomes the next-hop LSR for the FECs that are being requested.

LDP Security

LDP uses TCP for transport of LDP messages. The LDP specification does not provide its own security measures but leverages the existing TCP MD5 authentication mechanism defined in RFC 1321 and also used by BGP in RFC 2385. MD5 authentication uses a message digest to validate the authenticity and integrity of an LDP message.

A message digest is calculated with the MD5 hash algorithm that uses a shared secret key and the contents of the TCP segment. Unlike clear-text passwords, message digest prevents the shared secret from being snooped. In addition to protecting against spoofing, MD5 authentication provides good protection against denial of service (DoS) and man-in-the-middle attacks.

Understanding AToM Operations

In Chapter 3, you learned how AToM achieves a high degree of scalability by using the MPLS encoding method. You also read an overview of LDP in the previous section. Reading through this section, you will develop a further understanding of how MPLS encapsulation, LDP signaling, and pseudowire emulation work together.

The primary tasks of AToM include establishing pseudowires between provider edge (PE) routers and carrying Layer 2 packets over these pseudowires. The next sections cover the operations of AToM from the perspectives of both the control plane and the data plane as follows:

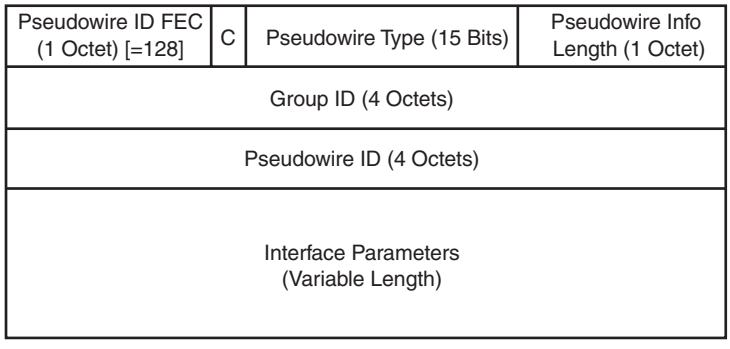
- Pseudowire label binding
- Establishing AToM pseudowires
- Control word negotiation
- Using sequence numbers
- Pseudowire encapsulation

Pseudowire Label Binding

An AToM pseudowire essentially consists of two unidirectional LSPs. Each is represented by a pseudowire label, also known as a VC label. The pseudowire label is part of the label stack encoding that encapsulates Layer 2 packets going over AToM pseudowires. Refer to Chapter 3 for an overview of an AToM packet.

The label distribution procedures that are defined in LDP specifications distribute and manage the pseudowire labels. To associate a pseudowire label with a particular Layer 2 connection, you need a way to represent such a Layer 2 connection. The baseline LDP specification only defines Layer 3 FECs. Therefore, the pseudowire emulation over MPLS application defines a new LDP extension—the Pseudowire ID FEC element—that contains a pseudowire identifier shared by the pseudowire endpoints. Figure 6-8 depicts the Pseudowire ID FEC element encoding.

Figure 6-8 *Pseudowire ID FEC Element*



The Pseudowire ID FEC element has the following components:

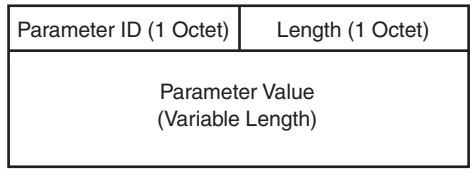
- **Pseudowire ID FEC**—The first octet has a value of 128 that identifies it as a Pseudowire ID FEC element.
- **Control Word Bit (C-Bit)**—The C-bit indicates whether the advertising PE expects the control word to be present for pseudowire packets. A control word is an optional 4-byte field located between the MPLS label stack and the Layer 2 payload in the pseudowire packet. The control word carries generic and Layer 2 payload-specific information. If the C-bit is set to 1, the advertising PE expects the control word to be present in every pseudowire packet on the pseudowire that is being signaled. If the C-bit is set to 0, no control word is expected to be present.
- **Pseudowire Type**—PW Type is a 15-bit field that represents the type of pseudowire. Examples of pseudowire types are shown in Table 6-1.

- **Pseudowire Information Length**—Pseudowire Information Length is the length of the Pseudowire ID field and the interface parameters in octets. When the length is set to 0, this FEC element stands for all pseudowires using the specified Group ID. The Pseudowire ID and Interface Parameters fields are not present.
- **Group ID**—The Group ID field is a 32-bit arbitrary value that is assigned to a group of pseudowires.
- **Pseudowire ID**—The Pseudowire ID, also known as VC ID, is a non-zero, 32-bit identifier that distinguishes one pseudowire from another. To connect two attachment circuits through a pseudowire, you need to associate each one with the same Pseudowire ID.
- **Interface Parameters**—The variable-length Interface Parameters field provides attachment circuit-specific information, such as interface MTU, maximum number of concatenated ATM cells, interface description, and so on. Each interface parameter uses a generic TLV encoding, as shown in Figure 6-9.

Table 6-1 *Pseudowire Types*

Pseudowire Type	Description
0x0001	Frame Relay data-link connection identifier (DLCI)
0x0002	ATM AAL5 service data unit (SDU) virtual channel connection (VCC)
0x0003	ATM Transparent Cell
0x0004	Ethernet VLAN
0x0005	Ethernet
0x0006	High-Level Data Link Control (HDLC)
0x0007	PPP

Figure 6-9 *Interface Parameter Encoding*



Even though LDP allows multiple FEC elements encoded into an FEC TLV, only one FEC element—the Pseudowire ID FEC element—exists in each FEC TLV for the pseudowire emulation over MPLS application.

Establishing AToM Pseudowires

Typically, two types of LDP sessions are involved in establishing AToM pseudowires. They are the nontargeted LDP session and the targeted LDP session.

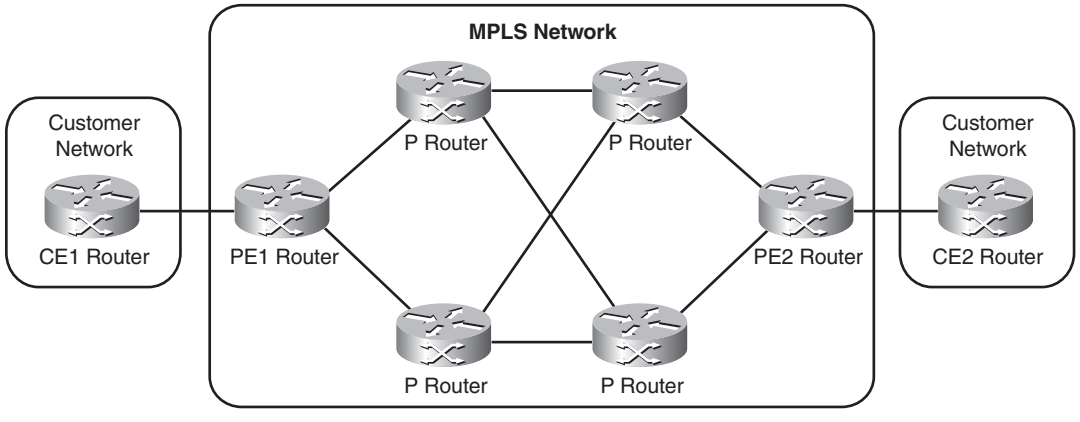
The nontargeted LDP session that is established through LDP basic discovery between a PE router and its directly connected P routers is used to distribute tunnel labels. The label distribution and management of tunnel labels pertains to the deployment model of the underlying MPLS network. It can be some combination of downstream on-demand or unsolicited label advertisement, independent or ordered control, and conservative or liberal label retention. Neither pseudowire emulation nor AToM dictates any particular label distribution and management mode for tunnel labels.

NOTE In some MPLS deployment scenarios, tunnel LSPs are set up through Resource Reservation Protocol Traffic Engineering (RSVP-TE) instead of nontargeted LDP sessions.

The other type of LDP sessions are established through LDP extended discovery between PE routers. These sessions are known as targeted LDP sessions because they send periodic Targeted Hello messages to each other. Targeted LDP sessions in the context of pseudowire emulation distribute pseudowire labels. IETF documents on pseudowire emulation over MPLS specify the use of downstream unsolicited label advertisement. In Cisco IOS Software, AToM uses independent label control and liberal label retention to improve performance and convergence time on pseudowire signaling.

Figure 6-10 illustrates an example of AToM deployment.

Figure 6-10 *AToM Deployment Model*



The following steps explain the procedures of establishing an AToM pseudowire:

1. A pseudowire is provisioned with an attachment circuit on PE1.
2. PE1 initiates a targeted LDP session to PE2 if none already exists. Both PE routers receive LDP Keepalive messages from each other and complete the session establishment. They are ready to exchange pseudowire label bindings.
3. When the attachment circuit state on PE1 transitions to up, PE1 allocates a local pseudowire label corresponding to the pseudowire ID that is provisioned for the pseudowire.
4. PE1 encodes the local pseudowire label into the Label TLV and the pseudowire ID into the FEC TLV. Then it sends this label binding to PE2 in a Label Mapping message.
5. PE1 receives a Label Mapping message from PE2 and decodes the pseudowire label and pseudowire ID from the Label TLV and FEC TLV.
6. PE2 performs Steps 1 through 5 independently.
7. After PE1 and PE2 exchange the pseudowire labels and validate interface parameters for a particular pseudowire ID, the pseudowire with that pseudowire ID is considered established.

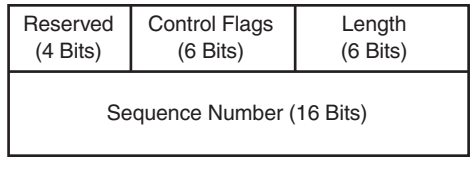
If one attachment circuit on one PE router goes down, a Label Withdraw message is sent to the peering PE router to withdraw the pseudowire label that it previously advertised.

Control Word Negotiation

During pseudowire establishment, Label Mapping messages are sent in both directions. To enable the pseudowire, you need to set some interface parameters to certain values that the peering PE router expects. When a mismatch occurs, fixing the problem requires manual intervention or configuration changes. The protocol cannot correct the mismatch automatically. For example, when the interface MTUs of the peering PE routers are different, the pseudowire is not established.

You can negotiate the presence of the control word through protocol signaling. The control word has 32 bits, as shown in Figure 6-11. If it is present, the control word is encapsulated in every pseudowire packet and carries per-packet information, such as sequence number, padding length, and control flags.

Figure 6-11 *AToM Control Word*



For certain Layer 2 payload types that are carried over pseudowires, such as Frame Relay DLCI and ATM AAL5, the control word must be present in the pseudowire encapsulation. That means you must set the C-bit in the pseudowire ID FEC element to 1 in both Label Mapping messages. When you receive a Label Mapping message that requires the mandatory control word but has a C-bit of 0, a Label Release message is sent with an Illegal C-bit status code. In this case, the pseudowire is not enabled.

For other Layer 2 payload types, the control word is optional. If a PE router cannot send and receive the optional control word, or if it is capable of doing that but prefers not to do so, the C-bit in the Label Mapping message that the PE router sends is set to 0. If a PE router is capable of and prefers sending and receiving the optional control word, the C-bit in the Label Mapping message it sends is set to 1. When two PE routers exchange Label Mapping messages, one of the following scenarios could happen when the control word is optional:

- Both C-bits are set to the same value—that is, either 0 or 1. In this case, the pseudowire establishment is complete. The control word is used if the common C-bit value is 1. Otherwise, the control word is not used.
- A PE router receives a Label Mapping message but has not sent a Label Mapping message for the pseudowire, and the local C-bit setting is different from the remote C-bit setting. If the received Label Mapping message has the C-bit set to 1, in this case, the PE router ignores the received Label Mapping message and continues to wait for the next Label message for the pseudowire. If the received Label Mapping message has the C-bit set to 0, the PE router changes the local C-bit setting to 0 for the Label Mapping message to be sent. If the attachment circuit comes up, the PE router sends a Label Mapping message with the latest local C-bit setting.
- A PE router has already sent a Label Mapping message, and it receives a Label Mapping message from a remote PE router. However, the local C-bit setting is different from the remote C-bit setting. If the received Label Mapping message has the C-bit set to 1, in this case, the PE router ignores the received Label Mapping message and continues to wait for the next label message for the pseudowire. If the received Label Mapping message has the C-bit set to 0, the PE router sends a Label Withdraw message with a Wrong C-bit status code, followed by a Label Mapping message with the C-bit set to 0. The pseudowire establishment is now complete, and the control word is not used.

To summarize the previous two scenarios, when the C-bit settings in the two Label Mapping messages do not match, the PE router that prefers the use of the option control word surrenders to the PE router that does not prefer it, and the control word is not used.

Configuring whether the control word is to be used in an environment with many different platforms is sometimes a tedious process. AToM automates this task by detecting the hardware capability of the PE router. AToM always prefers the presence of the control word and utilizes the control word negotiation procedures to reach a common C-bit value between PE routers.

Using Sequence Numbers

Because Layer 2 packets are normally transported over Layer 1 physical media directly, most Layer 2 protocols assume that the underlying transport ensures in-order packet delivery. These protocols might not function correctly if out-of-order delivery occurs. For instance, if PPP LCP packets are reordered, the end-to-end PPP connection is unable to establish.

To avoid out-of-order packets, the best solution is to engineer a reordering-free packet network. Even though this goal is not always easy to achieve, you should make it a priority because no matter what kind of remedy you might use, network performance suffers significantly from out-of-order delivery.

Sequencing that is defined in pseudowire emulation mainly serves a detection mechanism for network operators to troubleshoot occasional out-of-order delivery problems. Implementations might choose to either discard or reorder out-of-order packets when they are detected. Because the latter requires huge packet buffer space for high-speed links and has significant performance overhead, AToM simply discards out-of-order packets and relies on the upper layer to retransmit these packets.

The first step in using sequencing is to signal the presence of the control word, as described in the previous section. The control word contains a 16-bit Sequence Number field. However, the presence of the control word does not mandate sequencing. When sequencing is not used, Sequence Number value is set to 0.

After negotiating the control word, the sequence number is set to 1 and increments by 1 for each subsequent packet that is being transmitted. If the transmitting sequence number reaches the maximum value 65535, it wraps around to 1 again.

To detect an out-of-order packet, the receiving PE router calculates the expected sequence number for the next packet by using the last receiving sequence number (which has an initial value of 0) plus 1, and then mod (modulus) by 2^{16} ($2^{16} = 65536$). If the result is 0, the expected sequence number is set to 1. A packet that is received over a pseudowire is considered in-order if one of the following conditions is met:

- The receiving sequence number is 0.
- The receiving sequence number is no less than the expected sequence number and the result of the receiving sequence number minus the expected sequence number is less than 32768.
- The receiving sequence number is less than the expected sequence number and the result of the expected sequence number minus the receiving sequence number is no less than 32768.

If none of these conditions is satisfied, the packet is considered out-of-order and is discarded.

Sometimes the sending or the receiving PE router might lose the last transmitting or receiving sequence number because of transient system problems. This router might want to restart the sequence number from the initial value. AToM implements a set of signaling procedures to reliably resynchronize the sequence number. Although the IETF documents do not specify these

procedures, the procedures are interoperable with any standard-compliant implementation. The resynchronization procedures in AToM are as follows:

- If the transmitting PE router needs to reset the transmitting sequence number, it must inform the receiving PE router to reset the receiving sequence number. AToM accomplishes this by letting the transmitting PE router send a Label Release message to the receiving PE router, followed by a Label Request message. Because the receiving PE router interprets this as a pseudowire flapping, it resets the receiving sequence number.
- If the receiving PE router needs to reset the receiving sequence number, it must inform the receiving PE router to reset the transmitting sequence number. AToM does so by letting the receiving PE router send a Label Withdraw message to the transmitting PE router, followed by a Label Mapping message. Because the transmitting PE router perceives this as a pseudowire flapping, it resets the transmitting sequence number.

Pseudowire Encapsulation

To properly emulate Layer 2 protocols over pseudowires, you need to encapsulate each Layer 2 payload in such a way that Layer 2 characteristics are preserved as close to what they are in the native form as possible.

Aside from the MPLS label stack, pseudowire encapsulation also contains payload-specific information that varies on a per-transport and per-packet basis. This section discusses the payload-specific part of the encapsulation, which includes the control word and the Layer 2 payload.

The next few sections explain how the following Layer 2 protocols are encapsulated and processed on PE routers:

- ATM
- Frame Relay
- HDLC
- PPP
- Ethernet

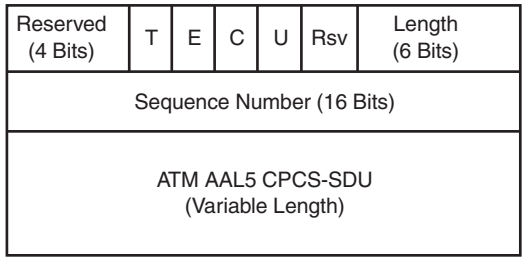
ATM

AToM supports two types of encapsulation for ATM transport: ATM AAL5 common part convergence sublayer service data unit (CPCS-SDU) and ATM Cell.

The ATM AAL5 CPCS-SDU encapsulation includes a mandatory control word. The ATM AAL5 CPCS-SDU encapsulation requires segmentation and reassembly (SAR) on the CE-PE ATM interface. When an ingress PE router receives ATM cells from a CE router, it reassembles them into an AAL5 CPCS-SDU and copies ATM control flags from the cell header into the

control word before sending it over a pseudowire. The AAL5 CPCS-SDU is segmented into ATM cells with proper cell headers on the egress PE router. Figure 6-12 illustrates the AAL5 CPCS-SDU pseudowire encapsulation.

Figure 6-12 *AAL5 CPCS-SDU Pseudowire Encapsulation*

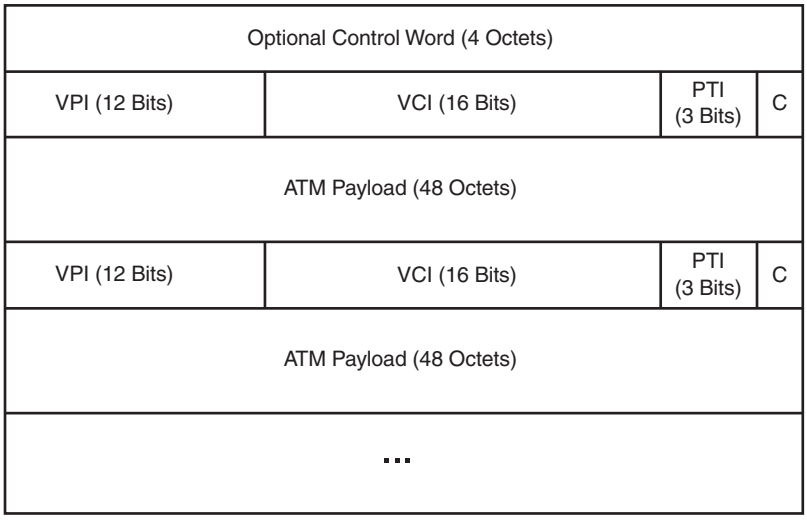


The control flags in the control word are described as follows:

- **Transport Type (T-Bit)**—This bit indicates whether the packet contains an ATM Operation, Administration, and Maintenance (OAM) cell or an AAL5 CPCS-SDU. If T = 1, the packet contains an ATM OAM cell. Otherwise, it contains an AAL5 CPCS-SDU. Being able to transport an ATM OAM cell in the AAL5 mode provides a way to enable administrative functionality over AAL5 VC.
- **EFCI (E-Bit)**—The E-bit stores the value of the EFCI bit of the last cell to be reassembled when the payload contains an AAL5 CPCS-SDU or that of the ATM OAM cell when the payload is an ATM OAM cell on the ingress PE router. The egress PE router then sets the EFCI bit of all cells to the value of the E-bit.
- **CLP (C-Bit)**—This is set to 1 if the CLP bit of any cell is set to 1 regardless of whether the cell is part of an AAL5 CPCS-SDU or is an ATM OAM cell on the ingress PE router. The egress PE router sets the CLP bit of all cells to the value of the C-bit.
- **Command/Response Field (U-Bit)**—When FRF.8.1 Frame Relay/ATM PVC Service Interworking traffic is being transmitted, the CPCS-UU Least Significant Bit of the AAL5 CPCS-SDU might contain the Frame Relay C/R bit. This flag carries that bit from the ingress PE router to the egress PE router.

With the ATM Cell encapsulation, ATM cells are transported individually without SAR. The ATM Cell encapsulation consists of the optional control word and one or more ATM cells. Each ATM cell has a 4-byte ATM cell header and a 48-byte ATM cell payload. Figure 6-13 illustrates the ATM cell pseudowire encapsulation.

Figure 6-13 *ATM Cell Pseudowire Encapsulation*

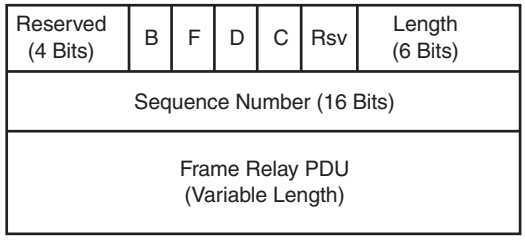


The maximum number of ATM cells that an ingress PE router can fit into a single pseudowire packet is constrained by the network MTU and the number of ATM cells that the egress PE router is willing to receive. This is signaled to the ingress PE router through the interface parameter "maximum number of concatenated ATM cells" in the Label Mapping message.

Frame Relay

Frame Relay DLCIs are locally significant, and it is likely that two Frame Relay attachment circuits that are connected through a pseudowire have different DLCIs. Therefore, you do not need to include DLCI as part of the Frame Relay pseudowire encapsulation. The control word is mandatory. Control flags in the Frame Relay header are mapped to the corresponding flag fields in the control word. Frame Relay payloads that are carried over pseudowires do not include the Frame Relay header or the FCS. Figure 6-14 illustrates the Frame Relay pseudowire encapsulation.

Figure 6-14 *Frame Relay Pseudowire Encapsulation*



The Frame Relay control flags in the control word are described as follows:

- **Backward Explicit Congestion Notification (B-Bit)**—The ingress PE router copies the BECN field of an incoming Frame Relay packet into the B-bit. The B-bit value is copied to the BECN field of the outgoing Frame Relay packet on the egress PE router.
- **Forward Explicit Congestion Notification (F-Bit)**—The ingress PE router copies the FECN field of an incoming Frame Relay packet into the F-bit. The F-bit value is copied to the FECN field of the outgoing Frame Relay packet on the egress PE router.
- **Discard Eligibility (D-Bit)**—The ingress PE router copies the DE field of an incoming Frame Relay packet into the D-bit. The D-bit value is copied to the DE field of the outgoing Frame Relay packet on the egress PE router.
- **Command/Response (C-Bit)**—The ingress PE router copies the C/R field of an incoming Frame Relay packet into the C-bit. The C-bit value is copied to the C/R field of the outgoing Frame Relay packet on the egress PE router.

HDLC

HDLC mode provides port-to-port transport of HDLC encapsulated frames. The pseudowire HDLC encapsulation consists of the optional control word, HDLC address, control and protocol fields without HDLC flags, and the FCS.

You can also use the HDLC mode to transport Frame Relay User-to-Network Interface (UNI) or Network-to-Network Interface (NNI) traffic port-to-port transparently because they use HDLC framing.

PPP

PPP mode provides port-to-port transport of PPP encapsulated frames. The PPP pseudowire encapsulation consists of the optional control word and the protocol field without media-specific framing information, such as HDLC address and control fields or FCS.

When you enable the Protocol Field Compression (PFC) option in PPP, the Protocol field is compressed from two octets into a single octet. PFC occurs between CE routers and is transparent to PE routers. PE routers transmit the protocol field in its entirety as it is received from CE routers.

If the CE-PE interface uses HDLC-like framing, the ingress PE router always strips off HDLC address and control fields from the PPP frames before transporting them over pseudowires. Perhaps two CE routers negotiate Address and Control Field Compression (ACFC). The egress PE router has no way of knowing that unless it snoops into the PPP LCP negotiation between the CE routers, and that is normally undesirable because of system complexities and performance overhead. Therefore, the egress PE router cannot determine whether it should add HDLC address and control fields for PPP frames that are being sent to the CE router.

In Cisco IOS, AToM uses a simple solution to solve this problem without snooping. Basically, the PPP specification says that a PPP implementation that supports HDLC-like framing must prepare to receive PPP frames with uncompressed address and control fields at all times regardless of ACFC. So with AToM, the egress PE router always adds HDLC address and control fields back to the PPP packet if the egress CE-PE interface uses HDLC-like framing. For interfaces that do not use HDLC-like framing, such as PPP over Ethernet, PPP over Frame Relay, and PPP over ATM AAL5, the egress PE router does not add HDLC address and control fields to the PPP packet.

Ethernet

With the Ethernet pseudowire encapsulation, the preamble and FCS are removed from the Ethernet frames on the ingress PE router before sending them over pseudowires, and they are regenerated on the egress PE router. The control word is optional.

Ethernet pseudowires have two modes of operations:

- **Raw mode**—In raw mode, an Ethernet frame might or might not have an IEEE 802.1q VLAN tag. If the frame does have this tag, the tag is not meaningful to both the ingress and egress PE routers.
- **Tagged mode**—In tagged mode, each frame must contain an IEEE 802.1q VLAN tag. The tag value is meaningful to both the ingress and egress PE routers.

To explain how ingress and egress PE routers process a VLAN tag, it is necessary to define the semantics for the VLAN tag first. For example, when the ingress PE receives an Ethernet frame from a CE router and the frame contains a VLAN tag, there are two possible scenarios:

- The VLAN tag is a service delimiter. The provider uses a *service delimiter* to distinguish one type of customer traffic from another. For example, each service-delimiting VLAN tag can represent a different customer who the provider is serving or a particular network service that the provider wants to offer. Some equipment that the provider operates usually places this VLAN tag onto the Ethernet frame.
- The VLAN tag is not a service delimiter. A CE router or some equipment that the customer operates usually places this VLAN tag. The VLAN tag is not meaningful to the ingress PE router.

If an Ethernet pseudowire operates in raw mode, a service-delimiting VLAN tag, if present, is removed from the Ethernet frame that is received from a CE router before the frame is sent over the pseudowire. If the VLAN tag is not a service delimiter, it is passed across the pseudowire transparently.

If an Ethernet pseudowire operates in tagged mode, each Ethernet frame that is sent over the pseudowire must have a VLAN tag, regardless of whether it is a service-delimiting VLAN tag.

In both modes, the service-delimiting VLAN tags have only local significance. That is, these tags are meaningful only at a particular CE-PE interface. When the egress PE router receives an Ethernet frame from the pseudowire, it references the operation mode and its local configuration to determine how to process this frame before transmitting it to the CE router. If the egress PE is using raw mode, it might add a service-delimiting VLAN tag, but it will not rewrite or remove a VLAN tag that is already present in the frame. If the egress PE is using tagged mode, it can rewrite, remove, or keep the VLAN tag that is present in the frame.

In Metro Ethernet deployment, in which CE routers and PE routers are connected through an Ethernet switched access network, packets that arrive at PE routers can contain two IEEE 802.1q VLAN tags. This type of packet is commonly known as a QinQ packet. When the outer VLAN tag is the service-delimiting VLAN tag, QinQ packets are processed exactly like the ones with a single VLAN tag in both raw mode and tagged mode. When the combination of the outer and inner VLAN tags is used for service-delimiting, it is processed as if it were a single VLAN tag but with an extended range of values.

If you need to take QoS into consideration, the ingress PE router can map the user priority bits in the VLAN header to the MPLS EXP bits in the MPLS label stack. In this way, transit LSRs in the MPLS network can apply QoS policies to the Ethernet frames that are carried over pseudowires.

Summary

This chapter first gave an overview of LDP, including LDP components and operations that are related to pseudowire emulation over MPLS. Then the chapter explained the control signaling and data switching details of AToM.

Despite multiple possible combinations of label distribution and management modes for pseudowire signaling, AToM implements the combination that uses LDP downstream unsolicited advertisement, independent control, and liberal retention modes.

Pseudowire emulation over MPLS introduces new protocol extensions and signaling procedures, such as the Pseudowire ID FEC element in the FEC TLV that is defined to represent a pseudowire that connects two attachment circuits on different PE routers, the negotiation of the control word presence, and sequence number resynchronization.

Pseudowire emulation over MPLS also specifies new encapsulation methods and data switching procedures, such as the control word that is customized for carrying transport-specific information, Layer 2 payload encapsulations, ingress/egress processing optimized for transporting over pseudowires, and the sequence number for detecting out-of-order packets.