

Internet Engineering Task Force  
Internet Draft  
Intended status: Standards Track  
Expires: April 20, 2014

Luca Martini  
Samer Salam  
Ali Sajassi  
Cisco

Matthew Bocci  
Alcatel-Lucent

Satoru Matsushima  
Softbank

Thomas Nadeau  
Juniper Networks

October 20, 2013

## **Inter-Chassis Communication Protocol for L2VPN PE Redundancy**

[draft-ietf-pwe3-iccp-12.txt](#)

### Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 20, 2013

### Abstract

This document specifies an inter-chassis communication protocol (ICCP) that enables Provider Edge (PE) device redundancy for Virtual Private Wire Service (VPWS) and Virtual Private LAN Service (VPLS) applications. The protocol runs within a set of two or more PEs, forming a redundancy group, for the purpose of synchronizing data

amongst the systems. It accommodates multi-chassis attachment circuit as well as pseudowire redundancy mechanisms.

## Table of Contents

<a href="#">1</a>	Specification of Requirements .....	<a href="#">5</a>
<a href="#">2</a>	Introduction .....	<a href="#">5</a>
<a href="#">3</a>	ICCP Overview .....	<a href="#">5</a>
<a href="#">3.1</a>	Redundancy Model & Topology .....	<a href="#">5</a>
<a href="#">3.2</a>	ICCP Interconnect Scenarios .....	<a href="#">7</a>
<a href="#">3.2.1</a>	Co-located Dedicated Interconnect .....	<a href="#">7</a>
<a href="#">3.2.2</a>	Co-located Shared Interconnect .....	<a href="#">8</a>
<a href="#">3.2.3</a>	Geo-redundant Dedicated Interconnect .....	<a href="#">8</a>
<a href="#">3.2.4</a>	Geo-redundant Shared Interconnect .....	<a href="#">9</a>
<a href="#">3.3</a>	ICCP Requirements .....	<a href="#">10</a>
<a href="#">4</a>	ICC LDP Protocol Extension Specification .....	<a href="#">12</a>
<a href="#">4.1</a>	LDP ICCP Capability Advertisement .....	<a href="#">12</a>
<a href="#">4.2</a>	RG Membership Management .....	<a href="#">13</a>
<a href="#">4.2.1</a>	ICCP Connection State Machine .....	<a href="#">13</a>
<a href="#">4.3</a>	Redundant Object Identification .....	<a href="#">16</a>
<a href="#">4.4</a>	Application Connection Management .....	<a href="#">16</a>
<a href="#">4.4.1</a>	Application Versioning .....	<a href="#">17</a>
<a href="#">4.4.2</a>	Application Connection State Machine .....	<a href="#">18</a>
<a href="#">4.5</a>	Application Data Transfer .....	<a href="#">21</a>
<a href="#">4.6</a>	Dedicated Redundancy Group LDP session .....	<a href="#">21</a>
<a href="#">5</a>	ICCP PE Node Failure Detection Mechanism .....	<a href="#">22</a>
<a href="#">6</a>	ICCP Message Formats .....	<a href="#">22</a>
<a href="#">6.1</a>	Encoding ICC into LDP Messages .....	<a href="#">23</a>
<a href="#">6.1.1</a>	ICC Header .....	<a href="#">23</a>
<a href="#">6.1.2</a>	Message Encoding .....	<a href="#">25</a>
<a href="#">6.1.3</a>	ROID Encoding .....	<a href="#">26</a>
<a href="#">6.2</a>	RG Connect Message .....	<a href="#">26</a>
<a href="#">6.2.1</a>	ICC Sender Name TLV .....	<a href="#">27</a>
<a href="#">6.3</a>	RG Disconnect Message .....	<a href="#">28</a>
<a href="#">6.4</a>	RG Notification Message .....	<a href="#">30</a>
<a href="#">6.4.1</a>	Notification Message TLVs .....	<a href="#">31</a>
<a href="#">6.5</a>	RG Application Data Message .....	<a href="#">34</a>
<a href="#">7</a>	Application TLVs .....	<a href="#">34</a>
<a href="#">7.1</a>	Pseudowire Redundancy (PW-RED) Application TLVs .....	<a href="#">34</a>
<a href="#">7.1.1</a>	PW-RED Connect TLV .....	<a href="#">34</a>
<a href="#">7.1.2</a>	PW-RED Disconnect TLV .....	<a href="#">36</a>
<a href="#">7.1.2.1</a>	PW-RED Disconnect Cause TLV .....	<a href="#">36</a>
<a href="#">7.1.3</a>	PW-RED Config TLV .....	<a href="#">37</a>
<a href="#">7.1.3.1</a>	Service Name TLV .....	<a href="#">39</a>
<a href="#">7.1.3.2</a>	PW ID TLV .....	<a href="#">40</a>
<a href="#">7.1.3.3</a>	Generalized PW ID TLV .....	<a href="#">41</a>
<a href="#">7.1.4</a>	PW-RED State TLV .....	<a href="#">42</a>
<a href="#">7.1.5</a>	PW-RED Synchronization Request TLV .....	<a href="#">43</a>
<a href="#">7.1.6</a>	PW-RED Synchronization Data TLV .....	<a href="#">45</a>



<a href="#">7.2</a>	Multi-chassis LACP (mLACP) Application TLVs .....	<a href="#">46</a>
<a href="#">7.2.1</a>	mLACP Connect TLV .....	<a href="#">46</a>
<a href="#">7.2.2</a>	mLACP Disconnect TLV .....	<a href="#">47</a>
<a href="#">7.2.2.1</a>	mLACP Disconnect Cause TLV .....	<a href="#">48</a>
<a href="#">7.2.3</a>	mLACP System Config TLV .....	<a href="#">48</a>
<a href="#">7.2.4</a>	mLACP Aggregator Config TLV .....	<a href="#">49</a>
<a href="#">7.2.5</a>	mLACP Port Config TLV .....	<a href="#">51</a>
<a href="#">7.2.6</a>	mLACP Port Priority TLV .....	<a href="#">53</a>
<a href="#">7.2.7</a>	mLACP Port State TLV .....	<a href="#">55</a>
<a href="#">7.2.8</a>	mLACP Aggregator State TLV .....	<a href="#">57</a>
<a href="#">7.2.9</a>	mLACP Synchronization Request TLV .....	<a href="#">59</a>
<a href="#">7.2.10</a>	mLACP Synchronization Data TLV .....	<a href="#">61</a>
<a href="#">8</a>	LDP Capability Negotiation .....	<a href="#">62</a>
<a href="#">9</a>	Client Applications .....	<a href="#">63</a>
<a href="#">9.1</a>	Pseudowire Redundancy Application Procedures .....	<a href="#">63</a>
<a href="#">9.1.1</a>	Initial Setup .....	<a href="#">64</a>
<a href="#">9.1.2</a>	Pseudowire Configuration Synchronization .....	<a href="#">64</a>
<a href="#">9.1.3</a>	Pseudowire Status Synchronization .....	<a href="#">65</a>
<a href="#">9.1.3.1</a>	Independent Mode .....	<a href="#">66</a>
<a href="#">9.1.3.2</a>	Master/Slave Mode .....	<a href="#">67</a>
<a href="#">9.1.4</a>	PE Node Failure .....	<a href="#">67</a>
<a href="#">9.2</a>	Attachment Circuit Redundancy Application Procedures .	<a href="#">68</a>
<a href="#">9.2.1</a>	Common AC Procedures .....	<a href="#">68</a>
<a href="#">9.2.1.1</a>	AC Failure .....	<a href="#">68</a>
<a href="#">9.2.1.2</a>	PE Node Failure .....	<a href="#">68</a>
<a href="#">9.2.1.3</a>	PE Isolation .....	<a href="#">68</a>
<a href="#">9.2.1.4</a>	Determining Pseudowire State .....	<a href="#">68</a>
<a href="#">9.2.2</a>	Multi-chassis LACP (mLACP) Application Procedures ....	<a href="#">69</a>
<a href="#">9.2.2.1</a>	Initial Setup .....	<a href="#">69</a>
<a href="#">9.2.2.2</a>	mLACP Aggregator and Port Configuration .....	<a href="#">71</a>
<a href="#">9.2.2.3</a>	mLACP Aggregator and Port Status Synchronization .....	<a href="#">72</a>
<a href="#">9.2.2.4</a>	Failure and Recovery .....	<a href="#">74</a>
<a href="#">10</a>	Security Considerations .....	<a href="#">75</a>
<a href="#">11</a>	IANA Considerations .....	<a href="#">75</a>
<a href="#">11.1</a>	MESSAGE TYPE NAME SPACE .....	<a href="#">75</a>
<a href="#">11.2</a>	TLV TYPE NAME SPACE .....	<a href="#">75</a>
<a href="#">11.3</a>	ICC RG Parameter Type Space .....	<a href="#">76</a>
<a href="#">11.4</a>	STATUS CODE NAME SPACE .....	<a href="#">77</a>
<a href="#">12</a>	Acknowledgments .....	<a href="#">77</a>
<a href="#">13</a>	Normative References .....	<a href="#">77</a>
<a href="#">14</a>	Informative References .....	<a href="#">78</a>
<a href="#">15</a>	Author's Addresses .....	<a href="#">78</a>



## **1. Specification of Requirements**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#).

## **2. Introduction**

Network availability is a critical metric for service providers as it has a direct bearing on their profitability. Outages translate not only to lost revenue but also to potential penalties mandated by contractual agreements with customers running mission-critical applications that require tight SLAs. This is true for any carrier network, and networks employing Layer 2 Virtual Private Network (L2VPN) technology are no exception. Network high-availability can be achieved by employing intra and inter-chassis redundancy mechanisms. The focus of this document is on the latter. The document defines an Inter-Chassis Communication Protocol (ICCP) that allows synchronization of state and configuration data between a set of two or more PEs forming a Redundancy Group (RG). The protocol supports multi-chassis redundancy mechanisms that can be employed on either the attachment circuit or pseudowire front.

## **3. ICCP Overview**

### **3.1. Redundancy Model & Topology**

The focus of this document is on PE node redundancy. It is assumed that a set of two or more PE nodes are designated by the operator to form a Redundancy Group (RG). Members of a Redundancy Group fall under a single administration (e.g. service provider) and employ a common redundancy mechanism towards the access (attachment circuits or access pseudowires) and/or towards the core (pseudowires) for any given service instance. It is possible, however, for members of an RG to make use of disparate redundancy mechanisms for disjoint services. The PE devices may be offering any type of L2VPN service, i.e. VPWS or VPLS. As a matter of fact, the use of ICCP may even be applicable for Layer 3 service redundancy, but this is considered to be outside the scope of this document.

The PEs in an RG offer multi-homed connectivity to either individual devices (e.g. CE, DSLAM, etc...) or entire networks (e.g. access network). Figure 1 below depicts the model.





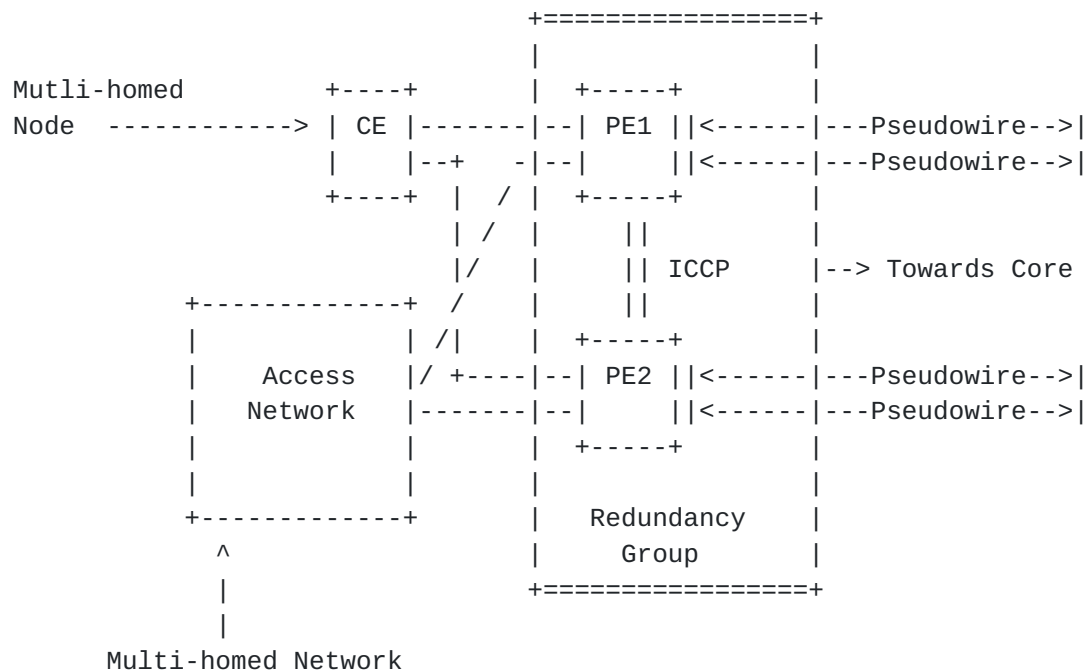


Figure 1: Generic Multi-chassis Redundancy Model

In the topology of Figure 1, the redundancy mechanism employed towards the access node/network can be one of a multitude of technologies, e.g. it could be IEEE 802.1AX Link Aggregation Groups with Link Aggregation Control Protocol (LACP), or SONET APS. The specifics of the mechanism are out of the scope of this document. However, it is assumed that the PEs in the RG are required to communicate amongst each other in order for the access redundancy mechanism to operate correctly. As such, it is required to run an inter-chassis communication protocol among the PEs in the RG in order to synchronize configuration and/or running state data.

Furthermore, the presence of the inter-chassis communication channel allows simplification of the pseudowire redundancy mechanism. This is primarily because it allows the PEs within an RG to run some arbitration algorithm to elect which pseudowire(s) should be in active or standby mode for a given service instance. The PEs can then advertise the outcome of the arbitration to the remote-end PE(s), as opposed to having to embed a hand-shake procedure into the pseudowire redundancy status communication mechanism, and every other possible Layer 2 status communication mechanism.



### 3.2. ICCP Interconnect Scenarios

When referring to 'interconnect' in this section, we are concerned with the links or networks over which Inter-Chassis Communication Protocol messages are transported, and not normal data traffic between PEs. The PEs which are members of an RG may be either physically co-located or geo-redundant. Furthermore, the physical interconnect between the PEs over which ICCP is to run may comprise of either dedicated back-to-back links or a shared connection through the packet switched network (PSN); for e.g., MPLS core network. This gives rise to a matrix of four interconnect scenarios, described next.

#### 3.2.1. Co-located Dedicated Interconnect

In this scenario, the PEs within an RG are co-located in the same physical location (POP, CO). Furthermore, dedicated links provide the interconnect for ICCP among the PEs.

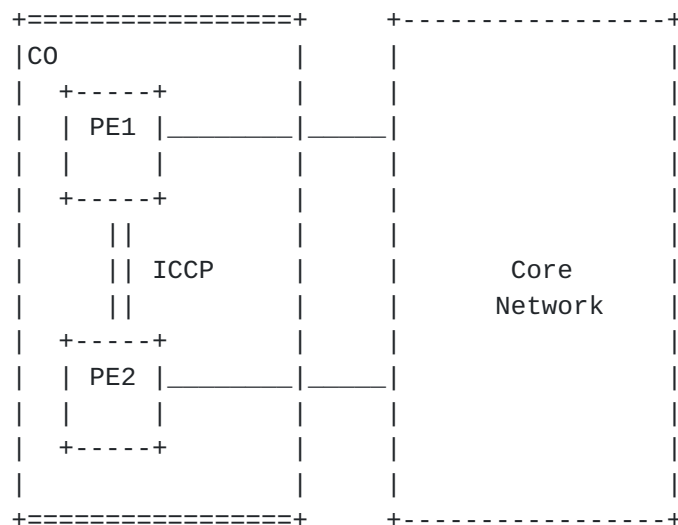


Figure 2: ICCP Co-located PEs Dedicated Interconnect Scenario

Given that the PEs are connected back-to-back in this case, it is possible to rely on Layer 2 redundancy mechanisms to guarantee the robustness of the ICCP interconnect. For example, if the interconnect comprises of IEEE 802.3 Ethernet links, it is possible to provide link redundancy by means of IEEE 802.1AX Link Aggregation Groups.



### 3.2.2. Co-located Shared Interconnect

In this scenario, the PEs within an RG are co-located in the same physical location (POP, CO). However, unlike the previous scenario, there are no dedicated links between the PEs. The interconnect for ICCP is provided through the core network to which the PEs are connected. Figure 3 depicts this model.

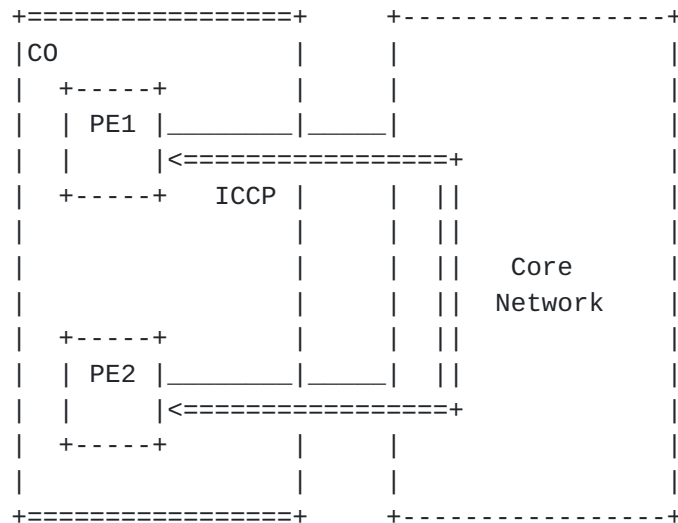


Figure 3: ICCP Co-located PEs Shared Interconnect Scenario

Given that the PEs in the RG are connected over the packet switched network (PSN), then PSN Layer mechanisms can be leveraged to ensure the resiliency of the interconnect against connectivity failures. For example, it is possible to employ RSVP LSPs with Fast Re-Route (FRR) and/or end-to-end backup LSPs.

### 3.2.3. Geo-redundant Dedicated Interconnect

In this variation, the PEs within a Redundancy Group are located in different physical locations to provide geographic redundancy. This may be desirable, for example, to protect against natural disasters or the like. A dedicated interconnect is provided to link the PEs, which is a costly option, especially when considering the possibility of providing multiple such links for interconnect robustness. The resiliency mechanisms for the interconnect are similar to those highlighted in the co-located interconnect counterpart.



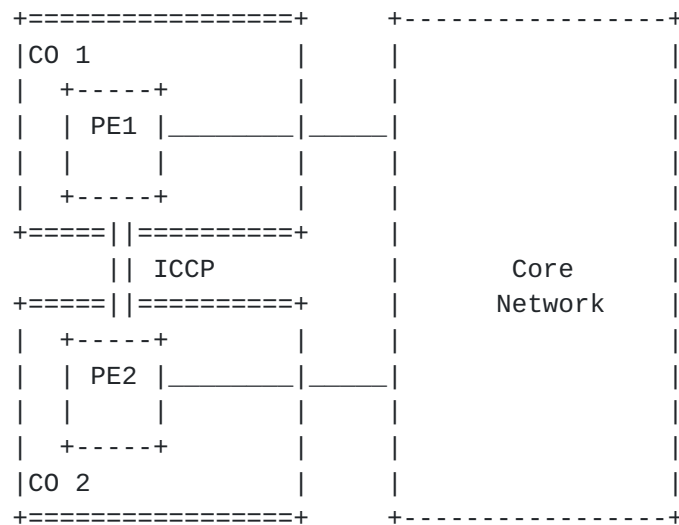


Figure 4: ICCP Geo-redundant PEs Dedicated Interconnect Scenario

#### **3.2.4. Geo-redundant Shared Interconnect**

In this scenario, the PEs of an RG are located in different physical locations and the interconnect for ICCP is provided over the PSN network to which the PEs are connected. This interconnect option is more likely to be the one used for geo-redundancy as it is more economically appealing compared to the geo-redundant dedicated interconnect. The resiliency mechanisms that can be employed to guarantee the robustness of the ICCP transport are PSN Layer mechanisms as has been described in the "Co-located Shared Interconnect" section above.





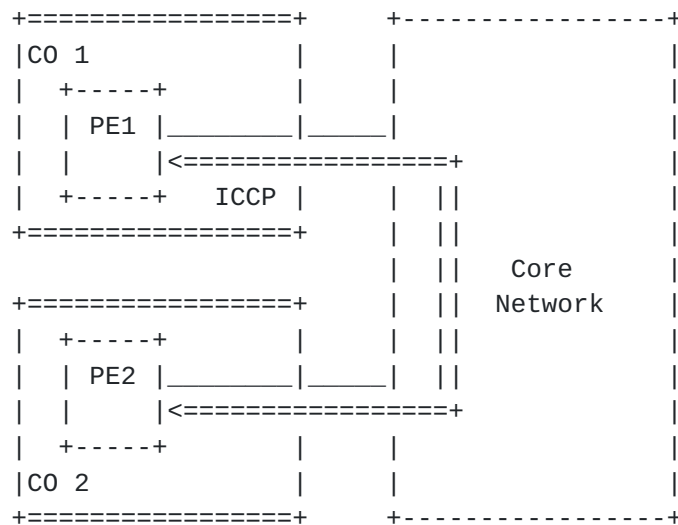


Figure 5: ICCP Geo-redundant PEs Shared Interconnect Scenario

### 3.3. ICCP Requirements

The Inter-chassis Communication Protocol SHOULD satisfy the following requirements:

- i. Provide a control channel for communication between PEs in a Redundancy Group (RG). Nodes may be co-located or remote (refer to "Interconnect Scenarios" section above). It is expected that client applications which make use of ICCP services will only use this channel to communicate control information and not data-traffic. As such the protocol should cater for relatively low bandwidth, low-delay and highly reliable message transfer.
- ii. Accommodate multiple client applications (e.g. multi-chassis LACP, PW redundancy, SONET APS, etc...). This implies that the messages should be extensible (e.g. TLV-based) and the protocol should provide a robust application registration and versioning scheme.
- iii. Provide reliable message transport and in-order delivery between nodes in a RG with secure authentication mechanisms built into the protocol. The redundancy applications that are clients of ICCP expect reliable message transfer, and as such will assume that the protocol takes care of flow-control and retransmissions. Furthermore, given that the applications will rely on ICCP to communicate data used to synchronize state-machines on disparate nodes, it is



critical that ICCP guarantees in-order message delivery. Loss of messages or out-of-sequence messages would have adverse side-effects to the operation of the client applications.

- iv. Provide a common mechanism to actively monitor the health of PEs in an RG. This mechanism will be used to detect PE node failure and inform the client applications. The applications require this to trigger failover according to the procedures of the employed redundancy protocol on the AC and PW. It is desired to achieve sub-second detection of loss of remote node (~ 50 - 150 msec) in order to give the client applications (redundancy mechanisms) enough reaction time to achieve sub-second service restoration time.
- v. Provide asynchronous event-driven state update, independent of periodic messages, for immediate notification of client applications' state changes. In other words, the transmission of messages carrying application data should be on-demand rather than timer-based to minimize inter-chassis state synchronization delay.
- vi. Accommodate multi-link and multi-hop interconnect between nodes. When the devices within an RG are located in different physical locations, the physical interconnect between them will comprise of a network rather than a link. As such, ICCP should accommodate the case where the interconnect involves multiple hops. Furthermore, it is possible to have multiple (redundant) paths or interconnects between a given pair of devices. This is true for both the co-located and geo-redundant scenarios. ICCP should handle this as well.
- vii. Ensure transport security between devices in an RG. This is especially important in the scenario where the members of an RG are located in different physical locations and connected over a shared network (e.g. PSN).
- viii. Must allow operator to statically configure members of RG. Auto-discovery may be considered in the future.
- ix. Allow for flexible RG membership. It is expected that only two nodes per an RG will cover most of the redundancy applications for common deployments. ICCP should not preclude supporting more than two nodes in an RG by virtue of design. Furthermore, it is required to allow a single node to be member of multiple RGs simultaneously.



#### **4. ICC LDP Protocol Extension Specification**

To address the requirements identified in the previous section, ICCP is modeled to comprise of three layers:

- i. Application Layer: This provides the interface to the various redundancy applications that make use of the services of ICCP. ICCP is concerned with defining common connection management procedures and the formats of the messages exchanged at this layer; however, beyond that, it does not impose any restrictions on the procedures or state-machines of the clients, as these are deemed application-specific and lie outside the scope of ICCP. This guarantees implementation inter-operability without placing any unnecessary constraints on internal design specifics.
- ii. Inter Chassis Communication (ICC) Layer: This layer implements the common set of services which ICCP offers to the client applications. It handles protocol versioning, RG membership, Redundant Object identification, PE node identification and ICCP connection management.
- iii. Transport Layer: This layer provides the actual ICCP message transport. It is responsible for addressing, route resolution, flow-control, reliable and in-order message delivery, connectivity resiliency/redundancy and finally PE node failure detection. The Transport layer may differ depending on the Physical Layer of the inter-connect.

##### **4.1. LDP ICCP Capability Advertisement**

When an RG is enabled on a particular PE, the capability of supporting ICCP must be advertised to all LDP peers in that RG. This is achieved by using the methods in [[RFC5561](#)] and advertising the ICCP LDP capability TLV. If an LDP peer supports the dynamic capability advertisement, this can be done by sending a new capability message with the S bit set for the ICCP capability TLV when the first RG is enabled on the PE. If the peer does not support dynamic capability advertisement, then the ICCP TLV MUST be included in the LDP initialization procedures in the capability parameter [[RFC5561](#)].



## **4.2. RG Membership Management**

ICCP defines a mechanism that enables PE nodes to manage their RG membership. When a PE is configured to be a member of an RG, it will first advertise the ICCP capability to its peers. Subsequently, the PE sends an RG Connect message to the peers that have also advertised ICCP capability. The PE then waits for the peers to send their own RG Connect messages, if they haven't done so already. For a given RG, the ICCP connection between two devices is considered to be operational only when both have sent and received ICCP RG Connect messages for that RG.

If a PE that has sent a particular RG Connect message doesn't receive a corresponding RG Connect (or a Notification message with NAK) from a destination, it will remain in a state expecting the corresponding RG Connect message (or Notification message). The RG will not become operational until the corresponding RG Connect Message has been received. If a PE that has sent an RG Connect message receives a Notification message with a NAK, it will stop attempting to bring up the ICCP connection immediately. The PE MUST resume bringing up the connection after it receives an RG Connect message from the peer PE for the RG in question. This is achieved by responding to the incoming RG Connect message with an appropriate RG Connect.

A device MUST send a NAK for an RG Connect message if at least one of the following conditions is satisfied:

- i. the PE is not a member of the RG;
- ii. the maximum number of simultaneous ICCP connections that the PE can handle is exceeded.

A PE sends an RG Disconnect message to tear down the ICCP connection for a given RG. This is a unilateral operation and doesn't require any acknowledgement from the other PEs. Note that the ICCP connection for an RG MUST be operational before any client application can make use of ICCP services in that RG.

### **4.2.1. ICCP Connection State Machine**

The ICCP Connection state machine is defined to have six states as depicted in the state transition table and state transition diagram that follow.

ICCP Connection State Transition Table

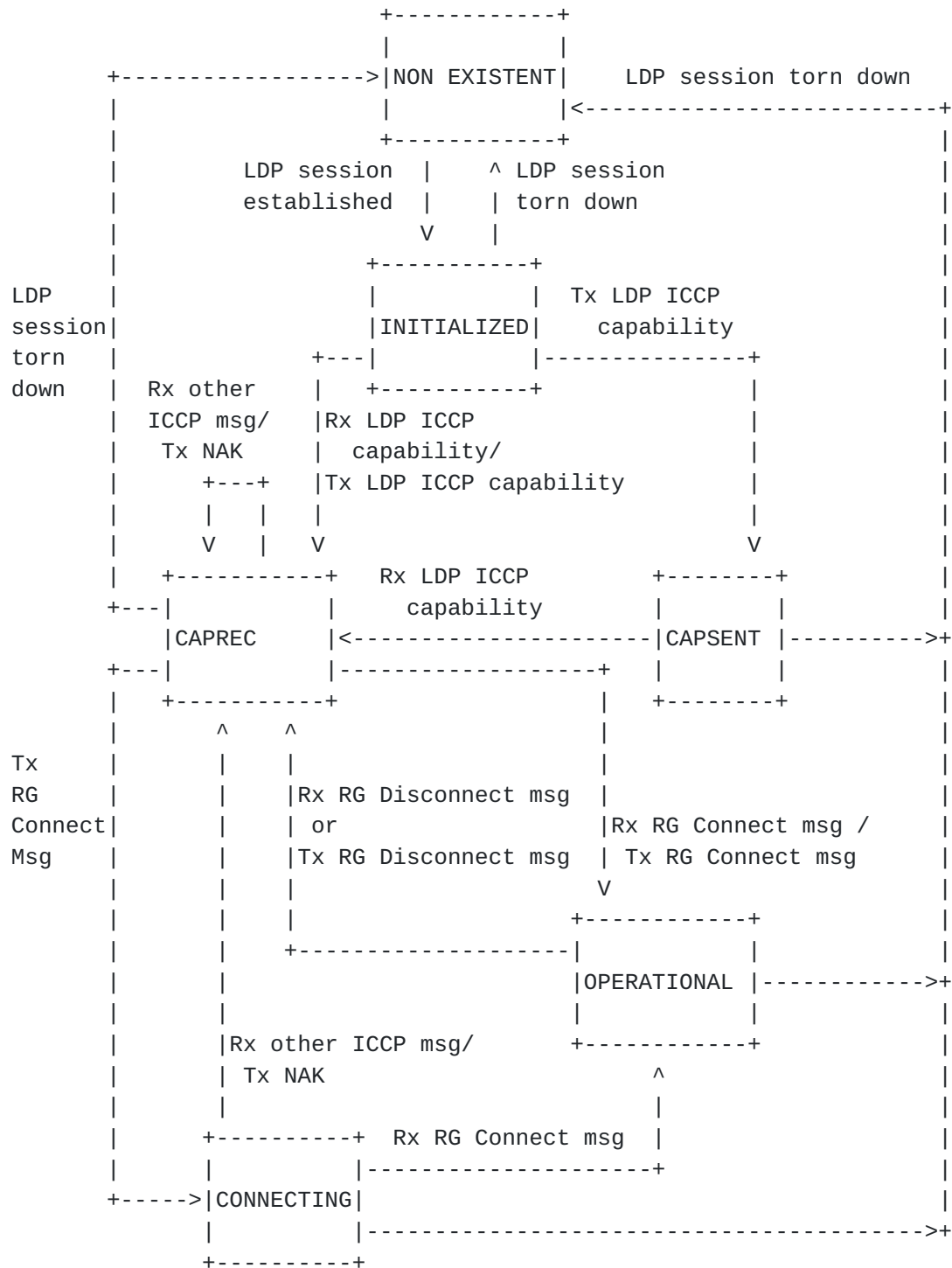




STATE	EVENT	NEW STATE
NON EXISTENT	LDP session established	INITIALIZED
INITIALIZED	Transmit LDP ICCP Capability	CAPSENT
	Receive LDP ICCP Capability Action: Transmit LDP ICCP Capability	CAPREC
	LDP session torn down	NON EXISTENT
CAPSENT	Receive LDP ICCP Capability	CAPREC
	LDP session torn down	NON EXISTENT
CAPREC	Transmit RG Connect Message	CONNECTING
	Receive acceptable RG Connect Message Action: Transmit RG Connect Message	OPERATIONAL
	Receive any other ICCP Message Action: Transmit NAK in RG Notification Message	CAPREC
	LDP session torn down	NON EXISTENT
CONNECTING	Receive acceptable RG Connect Message	OPERATIONAL
	Receive any other ICCP Message Action: Transmit NAK in RG Notification Message	CAPREC
	LDP session torn down	NON EXISTENT
OPERATIONAL	Receive acceptable RG Disconnect Message	CAPREC
	Transmit RG Disconnect Message	CAPREC
	LDP session torn down	NON EXISTENT

ICCP Connection State Transition Diagram







### **4.3. Redundant Object Identification**

ICCP offers its client applications a uniform mechanism for identifying links, ports, forwarding constructs and more generally objects (e.g. interfaces, pseudowires, VLANs, etc...) that are being protected in a redundant setup. These are referred to as Redundant Objects (RO). An example of an RO is a multi-chassis link-aggregation group that spans two PEs. ICCP introduces a 64-bit opaque identifier to uniquely identify ROs in an RG. This identifier, referred to as Redundant Object ID (ROID), MUST match between RG members for the protected object in question. That allows separate systems in an RG to use a common handle to reference the protected entity irrespective of its nature (e.g. physical or virtual) and in a manner that is agnostic to implementation specifics. Client applications that need to synchronize state pertaining to a particular RO SHOULD embed the corresponding ROID in their TLVs.

### **4.4. Application Connection Management**

ICCP provides a common set of procedures by which applications on one PE can connect to their counterparts on another PE, for purpose of inter-chassis communication in the context of a given RG. The prerequisite for establishing an application connection is to have an operational ICCP RG connection between the two endpoints. It is assumed that the association of applications with RGs is known a priori, e.g. by means of device configuration. ICCP then sends an Application-specific Connect TLV (carried in RG Connect message), on behalf of each client application, to each remote PE within the RG. The client may piggyback application-specific information in that Connect TLV, which for example can be used to negotiate parameters or attributes prior to bringing up the actual application connection. The procedures for bringing up the application connection are similar to those of the ICCP connection: An application connection between two nodes is up only when both nodes have sent and received RG Connect Messages with the proper Application-specific Connect TLVs. A PE MUST send a Notification Message to NAK an application connection request if one of the following conditions is encountered:

- i. the application doesn't exist or is not configured for that RG;
- ii. the application connection count exceeds the PE's capabilities.

When a PE receives such a NAK notification, it MUST stop attempting to bring up the application connection until it receives a new



application connection request from the remote PE. This is done by responding to the incoming RG Connect message (carrying an Application-specific Connect TLV) with an appropriate RG Connect message (carrying a corresponding Application-specific Connect TLV).

When an application is stopped on a device or it is no longer associated with an RG, it MUST signal ICCP to trigger sending an Application-specific Disconnect TLV (in the RG Disconnect message). This is a unilateral notification to the other PEs within an RG, and as such doesn't trigger any response.

#### **4.4.1. Application Versioning**

During application connection setup time, a given application on one PE can negotiate with its counterpart on a peer PE the proper application version to use for communication. If no common version is agreed upon, then the application connection is not brought up. This is achieved through the following set of rules:

- If an application receives an Application-specific Connect TLV with a version number that is higher than its own, it MUST send a Notification message with a NAK TLV indicating status code "Incompatible Protocol Version" and supplying the version that is locally supported by the PE.
- If an application receives an Application-specific Connect TLV with a version number that is lower than its own, it MAY respond with an RG Connect that has an Application-specific Connect TLV using the same version that was received. Alternatively, the application MAY respond with a Notification message to NAK the request using the "Incompatible Protocol Version" code, and supplying the version that is supported. The above allows an application to operate in either backwards compatible or incompatible mode.
- If an application receives an Application-specific Connect TLV with a version that is equal to its own, then the application MUST honor or reject the request based on whether the application is configured for the RG in question, and whether or not the application connection count has been exceeded.





#### **4.4.2. Application Connection State Machine**

The Application Connection state machine has six states as described in the state transition table and diagram that follow.

ICCP Application Connection State Transition Table

STATE	EVENT	NEW STATE
NON EXISTENT	ICCP connection established	RESET
RESET	ICCP connection torn down	NON EXISTENT
	Transmit Application Connect TLV	CONNSENT
	Receive Application Connect TLV	CONNREC
	Receive any other Application TLV Action: Transmit NAK TLV	RESET
CONNSENT	Receive NAK TLV	RESET
	Receive Application Connect TLV with A-bit=1 Action: Transmit Application Connect TLV with A-bit=1	OPERATIONAL
	Receive any other Application TLV Action: Transmit NAK TLV	RESET
	ICCP connection torn down	NON EXISTENT
CONNREC	Transmit NAK TLV	RESET
	Transmit Application Connect TLV with A-bit=1	CONNECTING
	Receive any Application TLV except Connect Action: Transmit NAK TLV	RESET
	ICCP connection torn down	NON EXISTENT
CONNECTING	Receive Application Connect TLV with A-bit=1	OPERATIONAL
	Receive any other Application TLV Action: Transmit NAK TLV	RESET
	ICCP connection torn down	NON EXISTENT
OPERATIONAL	Receive Application Disconnect TLV	RESET
	Transmit Applicaton Disconnect TLV	RESET



NON EXISTENT

```

+-----+
|                                     |
+----->|NON EXISTENT|   ICCP connection torn down
|                                     |<-----+
|                                     +-----+
|   ICCP connection|   ^ ICCP connection
|   established    |   | torn down
|                 |   |
|                 V   |
|                 +-----+<-----+ Rx other App TLV/
|                 | RESET |   | Tx NAK
|                 +-----+<-----+
| ICCP connect   Rx App   |   |   |
| torn down     Connect TLV|   |   |
|               +-----+<-----+
|               |   ^   ^   ^   ^   | Tx App
|               |   |   |   |   | Connect TLV|
|               | Tx NAK |   |   |   |   |
|               | or    |   |   |   |   |
|               | Rx non |   |   |   |   |
|               | Connect|   |   |   |   |
|               V      TLV/Tx NAK |   | Rx NAK TLV   V
| +-----+<-----+<-----+<-----+
+-|<-----+<-----+<-----+<-----+
| CONNREC |<-----+<-----+<-----+<-----+
+-|<-----+<-----+<-----+<-----+
| +-----+<-----+<-----+<-----+
| Tx App Connect |<-----+<-----+<-----+<-----+
| TLV (A=1)      |<-----+<-----+<-----+<-----+
|                | Rx App Disconn | Tx App
|                | or              | Connect TLV
|                | Tx App Disconn V (A=1)
|                | +-----+<-----+<-----+<-----+
|                +-----+<-----+<-----+<-----+
|                | Rx other App | OPERATIONAL |<-----+<-----+
|                | TLV / Tx NAK |<-----+<-----+<-----+<-----+
|                +-----+<-----+<-----+<-----+
|                | +-----+<-----+<-----+<-----+
|                | ^ Rx App Connect
|                | TLV (A=1)
|                +-----+<-----+<-----+<-----+
+--->|CONNECTING|<-----+<-----+<-----+<-----+
|<-----+<-----+<-----+<-----+
+-----+

```



#### **4.5. Application Data Transfer**

When an application has information to transfer over ICCP it triggers the transmission of an Application Data message. ICCP guarantees in-order and loss-less delivery of data. An application may NAK a message or a set of one or more TLVs within a message by using the Notification Message with NAK TLV. Furthermore, an application may implement its own ACK mechanism, if deemed required, by defining an application-specific TLV to be transported in an Application Data message.

It is left up to the application to define the procedures to handle the situation where a PE receives a NAK in response to a transmitted Application Data message. Depending on the specifics of the application, it may be favorable to have the PE, which sent the NAK, explicitly request retransmission of data. On the other hand, for certain applications it may be more suitable to have the original sender of the Application Data message handle retransmissions in response to a NAK. ICCP supports both models.

#### **4.6. Dedicated Redundancy Group LDP session**

For certain ICCP applications, it is required to exchange a fairly large amount of RG information in a very short period of time. In order to better distribute the load in a multiple processor system, and to avoid head of line blocking to other LDP applications, it may be required to initiate a separate TCP/IP session between the two LDP speakers.

This procedure is OPTIONAL, and does not change the operation of LDP or ICCP.

A PE that requires a separate LDP session will advertise a separate LDP adjacency with a non-zero label space identifier. This will cause the remote peer to open a separate LDP session for this label space. No labels need to be advertised in this label space, as it is only used for one or a set of ICCP RGs. All relevant LDP and ICCP procedures still apply as described in the relevant documents.



## **5. ICCP PE Node Failure Detection Mechanism**

ICCP provides its client applications a notification when a remote PE that is member of the RG fails. This is used by the client applications to trigger failover according to the procedures of the employed redundancy protocol on the AC and PW. To that end, ICCP does not define its own KeepAlive mechanism for purpose of monitoring the health of remote PE nodes, but rather reuses existing fault detection mechanisms. The following mechanisms may be used by ICCP to detect PE node failure:

- BFD

Run a BFD session [[RFC5880](#)] between the PEs that are members of a given RG, and use that to detect PE node failure. This assumes that resiliency mechanisms are in place to protect connectivity to the remote PE nodes, and hence loss of BFD periodic messages from a given PE node can only mean that the node itself has failed.

- IP Reachability Monitoring

It is possible for a PE to monitor IP layer connectivity to other members of an RG that are participating in IGP/BGP. When connectivity to a given PE is lost, the local PE interprets that to mean loss of the remote PE node. This assumes that resiliency mechanisms are in place to protect the route to the remote PE nodes, and hence loss of IP reachability to a given node can only mean that the node itself has failed.

It is worth noting here that loss of the LDP session with a PE in an RG is not a reliable indicator that the remote PE itself is down. It is possible, for e.g. that the remote PE encounters a local event that leads to resetting the LDP session, while the PE node remains operational for purpose of traffic forwarding.

## **6. ICCP Message Formats**

This section defines the messages exchanged at the Application and ICC layers.





## 6.1. Encoding ICC into LDP Messages

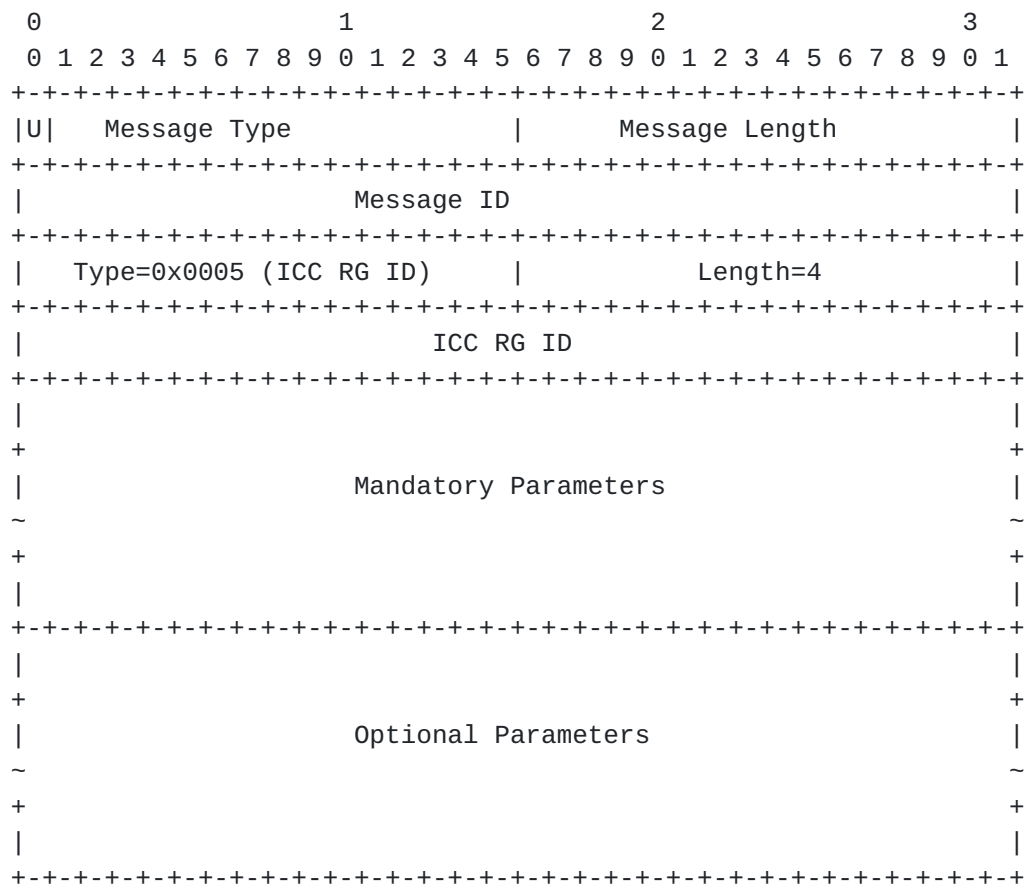
ICCP requires reliable, in-order, state-full message delivery, as well as capability negotiation between PEs. The LDP protocol offers all these features, and is already in wide use in the applications that would also require the ICCP protocol extensions. For these reasons, ICCP takes advantage of the already defined LDP protocol infrastructure.

[RFC5036] [Section 3.5](#) defines a generic LDP message structure. A new set of LDP message types is defined to communicate the ICCP information. LDP message types in the range of 0x700 to 0x7ff will be used for ICCP.

Message types are allocated by IANA, and requested in the IANA section below.

### 6.1.1. ICC Header

Every ICCP message comprises of an ICC specific LDP Header followed by message data. The format of the ICC Header is as follows:





- U-bit

Unknown message bit. Upon receipt of an unknown message, if U is clear (=0), a notification is returned to the message originator; if U is set (=1), the unknown message is silently ignored. The following sections which define messages specify a value for the U-bit.

- Message Type

Identifies the type of the ICCP message, must be in the range of 0x0700 to 0x07ff.

- Message Length

Two octet integer specifying the total length of this message in octets, excluding the U-bit, Message Type and Length fields.

- Message ID

Four octet value used to identify this message. Used by the sending PE to facilitate identifying RG Notification messages that may apply to this message. A PE sending an RG Notification message in response to this message SHOULD include this Message ID in the "NAK TLV" of the RG Notification message; see Section "RG Notification Message".

- ICC RG ID TLV

A TLV of type 0x0005, length 4, containing 4 octets unsigned integer designating the Redundancy Group which the sending device is member of. RG ID value 0x00000000 is reserved by the protocol.

- Mandatory Parameters

Variable length set of required message parameters. Some messages have no required parameters.

For messages that have required parameters, the required parameters MUST appear in the order specified by the individual message specifications in the sections that follow.

- Optional Parameters

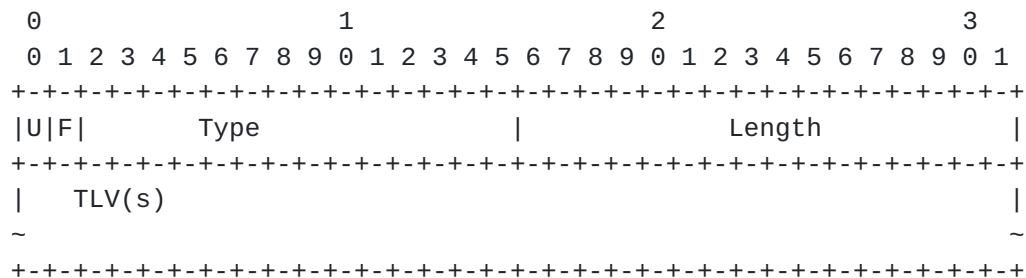
Variable length set of optional message parameters. Many messages have no optional parameters.



For messages that have optional parameters, the optional parameters may appear in any order.

### 6.1.2. Message Encoding

The generic format of an ICC parameter is:



- U-bit

Unknown TLV bit. Upon receipt of an unknown TLV, if U is clear (=0), a notification MUST be returned to the message originator and the entire message MUST be ignored; if U is set (=1), the unknown TLV MUST be silently ignored and the rest of the message processed as if the unknown TLV did not exist. The sections following that define TLVs specify a value for the U-bit.

- F-bit

Forward unknown TLV bit. This bit applies only when the U-bit is set and the LDP message containing the unknown TLV is to be forwarded. If F is clear (=0), the unknown TLV is not forwarded with the containing message; if F is set (=1), the unknown TLV is forwarded with the containing message. The sections following that define TLVs specify a value for the F-bit. By setting both the U- and F-bits, a TLV can be propagated as opaque data through nodes that do not recognize the TLV.

- Type

Fourteen bits indicating the parameter type.

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.



- TLV(s): A set of 0 or more TLVs, that vary according to the message type.

### 6.1.3. R0ID Encoding

The Redundant Object Identifier (R0ID) is a generic opaque handle that uniquely identifies a Redundant Object (e.g. link, bundle, VLAN, etc...) which is being protected in an RG. It is encoded as follows:

```

      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
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                     R0ID                                     |
+                                     +
|                                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

where: R0ID is an 8 octets field encoded as an unsigned integer. The R0ID value of 0 is reserved.

The R0ID is carried within application specific TLVs.

## 6.2. RG Connect Message

The RG Connect Message is used to establish the ICCP RG connection in addition to individual Application connections between PEs in an RG. An RG Connect message with no "Application-specific connect TLV" signals establishment of the ICCP RG connection. Whereas, an RG Connect message with a valid "Application-specific connect TLV" signals the establishment of an Application connection, in addition to the ICCP RG connection if the latter is not already established.

An implementation MAY send a dedicated RG Connect message to set up the ICCP RG connection and a separate RG Connect message per client application. However, all implementations MUST support the receipt of an RG Connect message that triggers the setup of the ICCP RG connection as well as a single Application connection simultaneously.

A PE sends an RG Connect Message to declare its membership in a Redundancy Group. One such message should be sent to each PE that is member of the same RG. The set of PEs to which RG Connect Messages should be transmitted is known via configuration or an auto-discovery mechanism that is outside the scope of this specification. If a device is member of multiple RGs, it MUST send separate RG Connect Messages for each RG even if the receiving device(s) happen to be the





same.

The format of the RG Connect Message is as follows:

- i. ICC header with Message type = "RG Connect Message" (0x0700)
- ii. ICC Sender Name TLV
- iii. Zero or one Application-specific connect TLV

The currently defined Application-specific connect TLVs are:

- PW Redundancy Connect TLV
- mLACP Connect TLV

The details of these TLVs are discussed in the "Application TLVs" section.

The RG Connect message can contain zero or one Application-specific connect TLV, but no application connect TLV can be sent more than once.

#### **6.2.1. ICC Sender Name TLV**

A TLV that carries the hostname of the sender encoded in UTF-8. This is used primarily for purpose of management of the RG and easing network operations. The specific format is shown below:

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|           Type = 0x0001           |   Length   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Sender Name                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
~                                           ~
|           ...                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U=F=0
- Type set to 0x0001 (from ICC parameter name space).
- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.



- Sender Name

Hostname of sending device encoded in UTF-8, and SHOULD NOT exceed 80 characters.

### 6.3. RG Disconnect Message

The RG Disconnect Message serves dual-purpose: to signal that a particular Application connection is being closed within an RG, or that the ICCP RG connection itself is being disconnected because the PE wishes to leave the RG. The format of this message is:

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|  Message Type=0x0701          |      Message Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                Message ID                        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Type=0x0005 (ICC RG ID)      |      Length=4                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                ICC RG ID                         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                Disconnect Code TLV              |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                Optional Application-specific Disconnect TLV
~-----~-----~-----~-----~-----~-----~-----~-----~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                Optional Parameter TLVs           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|-----|-----|-----|-----|-----|-----|-----|-----|
~-----~-----~-----~-----~-----~-----~-----~-----~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U-bit

U=0

- Message Type

The message type for RG Disconnect Message is set to (0x0701)



- Length

Length of the TLV in octets excluding the U-bit, Message Type, and Message Length fields.

- Message ID

Defined in the "ICC Header" section above.

- ICC RG ID

Defined in the "ICC Header" section above.

- Disconnect Code TLV

The format of this TLV is as follows:

```

      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
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|U|F|          Type=0x0004          |   Length   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                     ICCP Status Code                                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

- U,F Bits

both U and F are set to 0.

- Type

set to "Disconnect Code TLV" (0x0004)

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- ICCP Status Code

A status code that reflects the reason for the disconnect message. Allowed values are "ICCP RG Removed" and "ICCP Application Removed from RG".



- Optional Application-specific Disconnect TLV

Zero or one Application-specific Disconnect TLVs which are defined later in the document. If the RG Disconnect message has a status code of "RG Removed", then it MUST NOT contain any Application-specific Disconnect TLVs, as the sending PE is signaling that it has left the RG and, thus, is disconnecting the ICCP RG connection, with all associated client application connections. If the message has a status code of "Application Removed from RG", then it MUST contain exactly one Application-specific Disconnect TLV, as the sending PE is only tearing down the connection for the specified application. Other applications, and the ICCP RG connection are not to be affected.

- Optional Parameter TLVs

None are defined for this message in this document. This is specified to allow for future extensions.

#### **6.4. RG Notification Message**

A PE sends an RG Notification Message to indicate one of the following: to reject an ICCP connection, to reject an application connection, to NAK an entire message or to NAK one or more TLV(s) within a message. The Notification message can only be sent to a PE that is already part of an RG.

The RG Notification Message MUST NOT be used to NAK messages or TLVs corresponding to multiple ICCP applications simultaneously. In other words, there is a limit of at most a single ICCP application per RG Notification Message.

The format of the RG Notification Message is:

- i. ICC header with Message type = "RG Notification Message" (0x0702)
- ii. Notification Message TLVs.

The currently defined Notification message TLVs are:

- i. ICC Sender Name TLV
- ii. NAK TLV.





#### 6.4.1. Notification Message TLVs

The ICC Sender Name TLV uses the same format as in the RG Connect message, and was described above.

The NAK TLV is defined as follows:

[illegible]

- U, F Bits

both U and F are set to 0.

- Type

```
set to "NAK TLV" (0x0002)
```

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- ICCP Status Code

A status code that reflects the reason for the NAK TLV. Allowed values are:

- i. Unknown RG (0x00010001)

This code is used to reject a new incoming ICCP connection for an RG that is not configured on the local PE. When this code is used, the Rejected Message ID field MUST contain the message ID of the rejected "RG Connect" message.



-ii. ICCP Connection Count Exceeded (0x00010002)

This is used to reject a new incoming ICCP connection that would cause the local PE's ICCP connection count to exceed its capabilities. When this code is used, the Rejected Message ID field MUST contain the message ID of the rejected "RG Connect" message.

-iii. Application Connection Count Exceeded (0x00010003)

This is used to reject a new incoming application connection that would cause the local PE's ICCP connection count to exceed its capabilities. When this code is used, the Rejected Message ID field MUST contain the message ID of the rejected "RG Connect" message and the corresponding Application Connect TLV MUST be included in the "Optional TLV".

-iv. Application not in RG (0x00010004)

This is used to reject a new incoming application connection when the local PE doesn't support the application, or the application is not configured in the RG. When this code is used, the Rejected Message ID field MUST contain the message ID of the rejected "RG Connect" message and the corresponding Application Connect TLV MUST be included in the "Optional TLV".

-v. Incompatible Protocol Version (0x00010005)

This is used to reject a new incoming application connection when the local PE has an incompatible version of the application. When this code is used, the Rejected Message ID field MUST contain the message ID of the rejected "RG Connect" message and the corresponding Application Connect TLV MUST be included in the "Optional TLV".

-vi. Rejected Message (0x00010006)

This is used to reject an RG Application Data message, or one or more TLV(s) within the message. When this code is used, the Rejected Message ID field MUST contain the message ID of the rejected "RG Application Data" message.



-vii. ICCP Administratively Disabled (0x00010007)

This is used to reject any ICCP messages from a peer from which the PE is not allowed to exchange ICCP messages due to local administrative policy.

- Rejected Message ID

If non-zero, four octets value that identifies the peer message to which the NAK TLV refers. If zero, no specific peer message is being identified.

- Optional TLV(s)

A set of one or more optional TLVs. If the status code is "Rejected Message" then this field contains the TLV(s) that were rejected. If the entire message is rejected, all its TLVs **MUST** be present in this field; otherwise, the subset of TLVs that were rejected **MUST** be echoed in this field.

If the status code is "Incompatible Protocol Version" then this field contains the original "Application Connect TLV" sent by the peer, in addition to the "Requested Protocol Version TLV" defined below:

[illegible]

- U and F Bits

Both are set to 0.

- Type

```
set to 0x0003 for "Requested Protocol Version TLV"
```

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.



- Connection Reference

This field is set to the Type field of the Application specific Connect TLV that was rejected because of incompatible version.

- Requested Version

The version of the application supported by the transmitting device. For this version of the protocol it is set to 0x0001.

## **6.5. RG Application Data Message**

The RG Application Data Message is used to transport application data between PEs within an RG. A single message can be used to carry data from only one application. Multiple application TLVs are allowed in a single message, as long as all of these TLVs belong to the same application. The format of the Application Data Message is:

- i. ICC header with Message type = "RG Application Data Message" (0x703)
- ii. "Application specific TLVs"

The details of these TLVs are discussed in the "Application TLVs" section. All application specific TLVs in one RG Application Data Message MUST belong to a single application but MAY reference different ROs.

## **7. Application TLVs**

### **7.1. Pseudowire Redundancy (PW-RED) Application TLVs**

This section discusses the ICCP TLVs for the Pseudowire Redundancy application.

#### **7.1.1. PW-RED Connect TLV**

This TLV is included in the RG Connect message to signal the establishment of PW-RED application connection.





```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0010                               |   Length   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Protocol Version                       |A|   Reserved   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Optional Sub-TLVs                               |
~                                                                 ~
|                                                                 |
+                                     +---+---+---+---+---+---+---+---+
|                                     ...                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0010 for "PW-RED Connect TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Protocol Version

The version of this particular protocol for the purposes of ICCP. This is set to 0x0001.

- A bit

Acknowledgement Bit. Set to 1 if the sender has received a PW-RED Connect TLV from the recipient. Otherwise, set to 0.

- Reserved

Reserved for future use.

- Optional Sub-TLVs

There are no optional Sub-TLVs defined for this version of the protocol.



**7.1.2. PW-RED Disconnect TLV**

This TLV is used in an RG Disconnect Message to indicate that the connection for the PW-RED application is to be terminated.

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0011           |   Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Optional Sub-TLVs   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0011 for "PW-RED Disconnect TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Optional Sub-TLVs

The only optional Sub-TLV defined for this version of the protocol is the "PW-RED Disconnect Cause" TLV defined next:

**7.1.2.1. PW-RED Disconnect Cause TLV**

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0019           |   Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Disconnect Cause String   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                                                                 ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```



- U and F Bits

Both are set to 0.

- Type

set to 0x0019 for "PW-RED Disconnect Cause TLV"

- Length

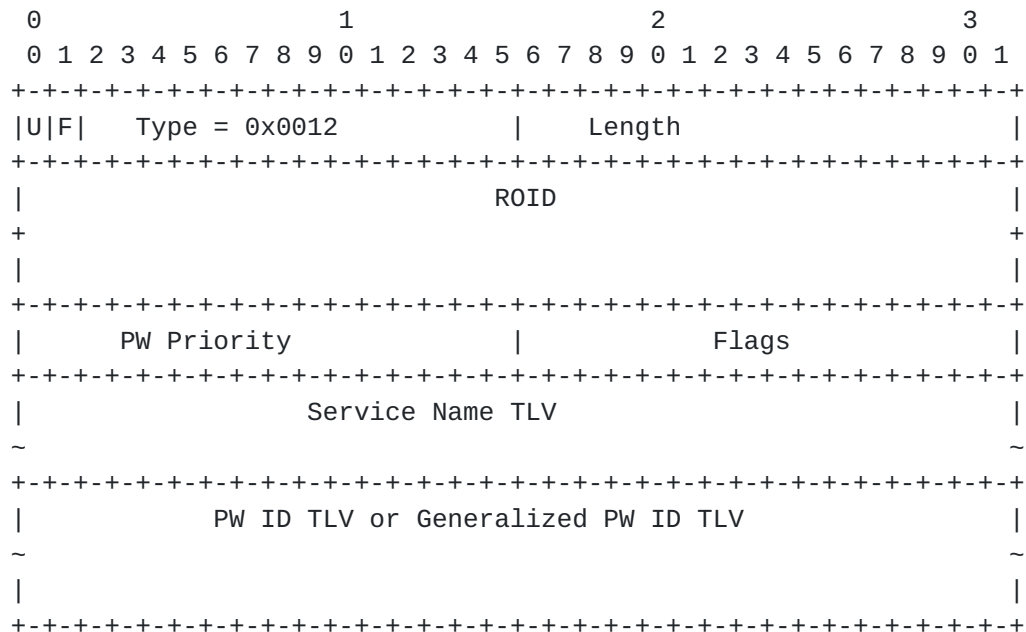
Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Disconnect Cause String

Variable length string specifying the reason for the disconnect.  
Used for network management.

#### **7.1.3. PW-RED Config TLV**

The PW-RED Config TLV is used in the RG Application Data message and has the following format:





- U and F Bits

Both are set to 0.

- Type

set to 0x0012 for "PW-RED Config TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- ROID

As defined in the ROID section above.

- PW Priority

Two octets Pseudowire Priority. Used to indicate which PW has better priority to go into Active state. Numerically lower numbers are better priority. In case of a tie, the PE with the numerically lower identifier (i.e. IP Address) has better priority.

- Flags

Valid values are:

- i. Synchronized (0x01)

Indicates that the sender has concluded transmitting all pseudowire configuration for a given service.

- ii. Purge Configuration (0x02)

Indicates that the pseudowire is no longer configured for PW-RED operation.

- iii. Independent Mode (0x04)

Indicates that the pseudowire is configured for redundancy using the Independent Mode of operation, per section 5.1 of [[RED-BIT](#)].





-iv. Independent Mode with Request Switchover (0x08)

Indicates that the pseudowire is configured for redundancy using the Independent Mode of operation with the use of the "Request Switchover" bit, per section 6.3 of [\[RED-BIT\]](#).

-v. Master Mode (0x10)

Indicates that the pseudowire is configured for redundancy using the Master/Slave Mode of operation, with the advertising PE acting as Master, per [section 5.2](#) of [\[RED-BIT\]](#).

-vi. Slave Mode (0x20)

Indicates that the pseudowire is configured for redundancy using the Master/Slave Mode of operation, with the advertising PE acting as Slave, per section 5.2 of [\[RED-BIT\]](#).

- Sub-TLVs

The "PW-RED Config TLV" includes the following two sub-TLVs:

-i. Service Name TLV

-ii. PW ID TLV or Generalized PW ID TLV

The format of the sub-TLVs is as follows:

#### [7.1.3.1. Service Name TLV](#)

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|  Type                               |  Length                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Service Name                               |
~                                                                           ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```



- U and F Bits

Both are set to 0.

- Type

set to 0x0013 for "Service Name TLV"

- Length

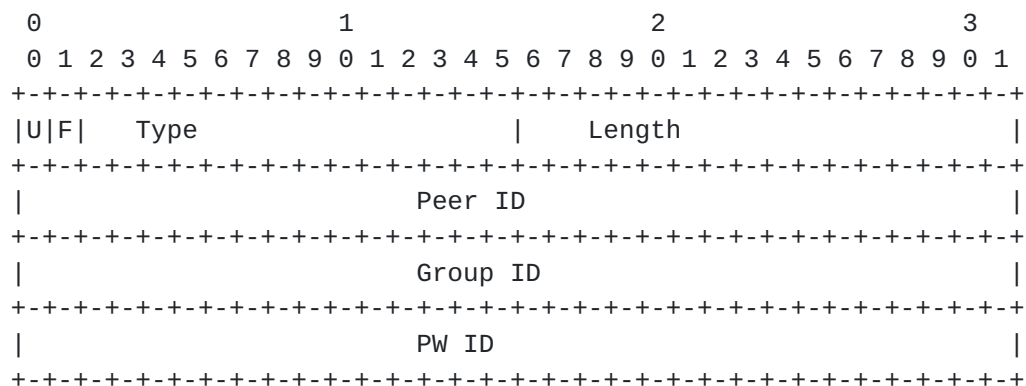
Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Service Name

The name of the L2VPN service instance encoded in UTF-8 format and up to 80 character in length.

#### **7.1.3.2. PW ID TLV**

This TLV is used to communicate the configuration of PWs for VPWS.



- U and F Bits

Both are set to 0.

- Type

set to 0x0014 for "PW ID TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.



- Peer ID

Four octet LDP Router ID of the peer at the far end of the PW.

- Group ID

Same as Group ID in [\[RFC4447\] section 5.2](#).

- PW ID

Same as PW ID in [\[RFC4447\] section 5.2](#).

### **7.1.3.3. Generalized PW ID TLV**

This TLV is used to communicate the configuration of PWs for VPLS.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type = 0x0015           |   Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  AGI Type   |   Length   |   Value   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~               AGI Value (contd.)               ~
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  AII Type   |   Length   |   Value   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~               SAII Value (contd.)               ~
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  AII Type   |   Length   |   Value   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~               TAIL Value (contd.)               ~
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F bits

both set to 0.

- Type

set to 0x0015



- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- AGI, AII, SAII and TAII

defined in [\[RFC4447\] section 5.3.2](#).

#### **7.1.4. PW-RED State TLV**

The PW-RED State TLV is used in the RG Application Data Message. This TLV is used by a device to report its PW status to other members in the RG.

The format of this TLV is as follows:

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|  Type=0x0016          |  Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               ROID                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Local PW State                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Remote PW State                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0016 for PW-RED State TLV.

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.





- R0ID

As defined in the R0ID section above.

- Local PW State

The status of the PW as determined by the sending PE, encoded in the same format as the "Status Code" field of the "PW Status TLV" defined in [\[RFC4447\]](#) and extended in [\[RED-BIT\]](#).

- Remote PW State

The status of the PW as determined by the remote peer of the sending PE. Encoded in the same format as the "Status Code" field of the "PW Status TLV" defined in [\[RFC4447\]](#) and extended in [\[RED-BIT\]](#).

#### **[7.1.5. PW-RED Synchronization Request TLV](#)**

The PW-RED Synchronization Request TLV is used in the RG Application Data message. This TLV is used by a device to request from its peer to retransmit configuration or operational state. The following information can be requested:

- configuration and/or state for one or more pseudowires
- configuration and/or state for all pseudowires
- configuration and/or state for all pseudowires in a given service

The format of the TLV is as follows:

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|  Type=0x0017                               |  Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Request Number          |C|S|  Request Type          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Optional Sub-TLVs            |
~                                                                    ~
|                                                                    |
+                               +---+---+---+---+---+---+---+---+
|                               ...                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```



- U and F Bits

Both are set to 0.

- Type

set to 0x0017 for "PW-RED Synchronization Request TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Request Number

2 octets. Unsigned integer uniquely identifying the request. Used to match the request with a response. The value of 0 is reserved for unsolicited synchronization, and MUST NOT be used in the PW-RED Synchronization Request TLV.

- C Bit

Set to 1 if request is for configuration data. Otherwise, set to 0.

- S Bit

Set to 1 if request is for running state data. Otherwise, set to 0.

- Request Type

14-bits specifying the request type, encoded as follows:

0x00	Request Data for specified pseudowire(s)
0x01	Request Data for all pseudowires in specified service(s)
0x3FFF	Request All Data

- Optional Sub-TLVs

A set of zero or more TLVs, as follows:

If the Request Type field is set to (0x00), then this field contains one or more PW ID TLV(s) or Generalized PW ID TLV(s). If the Request Type field is set to (0x01), then this field contains one or more Service Name TLV(s). If the Request Type field is set to (0x3FFF), then this field MUST be empty.



**7.1.6. PW-RED Synchronization Data TLV**

The PW-RED Synchronization Data TLV is used in the RG Application Data message. A pair of these TLVs is used by a device to delimit a set of TLVs that are sent in response to a PW-RED Synchronization Request TLV. The delimiting TLVs signal the start and end of the synchronization data, and associate the response with its corresponding request via the Request Number field.

The PW-RED Synchronization Data TLVs are also used for unsolicited advertisements of complete PW-RED configuration and operational state data. In this case, the Request Number field **MUST** be set to 0.

This TLV has the following format:

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|  Type=0x0018                |  Length                |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Request Number                |  Flags                |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0018 for "PW-RED Synchronization Data TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Request Number

2 octets. Unsigned integer identifying the Request Number from the "PW-RED Synchronization Request TLV" which solicited this synchronization data response.

- Flags

2 octets, response flags encoded as follows:



0x00 Synchronization Data Start

0x01 Synchronization Data End

## 7.2. Multi-chassis LACP (mLACP) Application TLVs

This section discusses the ICCP TLVs for Ethernet attachment circuit redundancy using the multi-chassis LACP (mLACP) application.

### 7.2.1. mLACP Connect TLV

This TLV is included in the RG Connect message to signal the establishment of mLACP application connection.

```

      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
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|U|F|  Type=0x0030                |  Length                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      Protocol Version          |A|      Reserved          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                Optional Sub-TLVs            |
~                                                                    ~
|                                                                |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                ...                                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

#### - U and F Bits

Both are set to 0.

#### - Type

set to 0x0030 for "mLACP Connect TLV"

#### - Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

#### - Protocol Version

The version of this particular protocol for the purposes of ICCP. This is set to 0x0001.





- A Bit

Acknowledgement Bit. Set to 1 if the sender has received an mLACP Connect TLV from the recipient. Otherwise, set to 0.

- Reserved

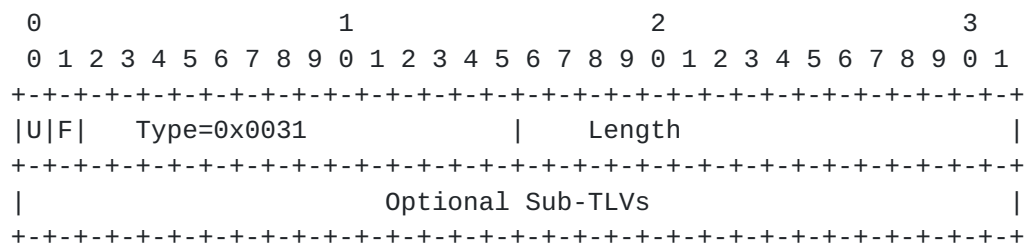
Reserved for future use.

- Optional Sub-TLVs

There are no optional Sub-TLVs defined for this version of the protocol.

### 7.2.2. mLACP Disconnect TLV

This TLV is used in an RG Disconnect Message to indicate that the connection for the mLACP application is to be terminated.



- U and F Bits

Both are set to 0.

- Type

```
set to 0x0031 for "mLACP Disconnect TLV"
```

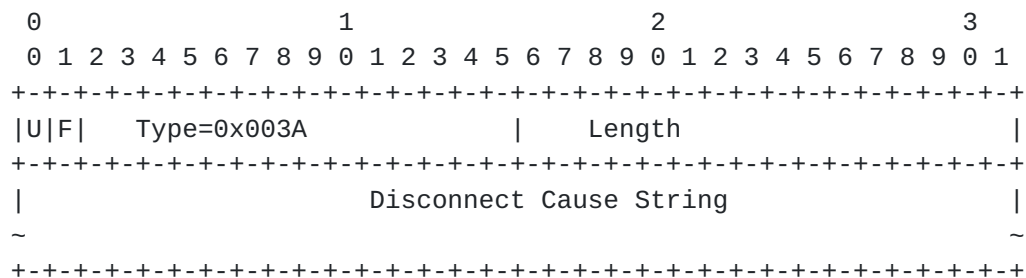
- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Optional Sub-TLVs

The only optional Sub-TLV defined for this version of the protocol is the "mLACP Disconnect Cause" TLV defined next:



**7.2.2.1. mLACP Disconnect Cause TLV**

- U and F Bits

Both are set to 0.

- Type

set to 0x003A for "mLACP Disconnect Cause TLV"

- Length

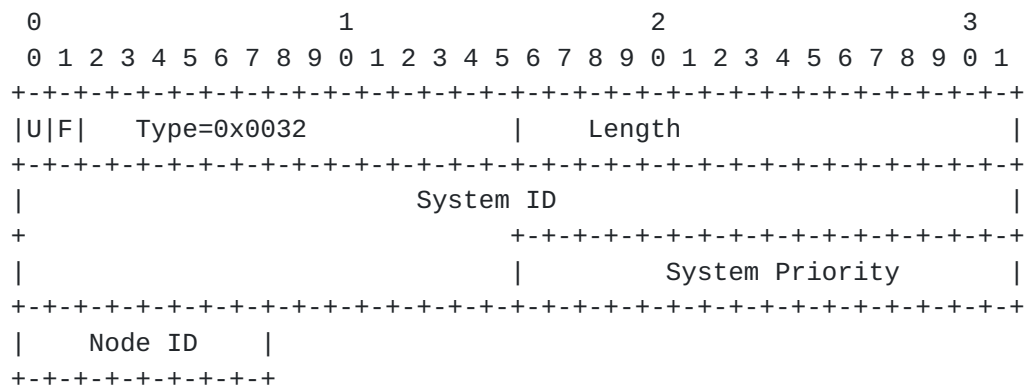
Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Disconnect Cause String

Variable length string specifying the reason for the disconnect. Used for network management.

**7.2.3. mLACP System Config TLV**

The mLACP System Config TLV is sent in the RG Application Data message. This TLV announces the local node's LACP System Parameters to the RG peers.





- U and F Bits

Both are set to 0.

- Type

set to 0x0032 for "mLACP System Config TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- System ID

6 octets field encoding the System ID used by LACP as specified in [[IEEE-802.1AX](#)] [section 5.3.2](#).

- System Priority

2 octets encoding the LACP System Priority as defined in [[IEEE-802.1AX](#)] [section 5.3.2](#).

- Node ID

One octet, LACP node ID. Used to ensure that the LACP Port Numbers are unique across all devices in an RG. Valid values are in the range 0 - 7. Uniqueness of the LACP Port Numbers across RG members is ensured by encoding the Port Numbers as follows:

- Most significant bit always set to 1
- The next 3 most significant bits set to Node ID
- Remaining 12 bits freely assigned by the system

#### **[7.2.4. mLACP Aggregator Config TLV](#)**

The mLACP Aggregator Config TLV is sent in the RG Application Data message. This TLV is used to notify RG peers about the local configuration state of an aggregator.



```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0036               |   Length               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               R0ID                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |   MAC Address             |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Actor Key                   |   Member Ports Priority   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Flags       | Agg Name Len |   Aggregator Name         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                                                         ~
|                               ...                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0036 for "mLACP Aggregator Config TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- R0ID

Defined in the 'R0ID Encoding' section above.

- Aggregator ID

Two octets, LACP Aggregator Identifier as specified in [IEEE-802.1AX] [section 5.4.6](#)

- MAC Address

Six octets encoding the Aggregator MAC address.





- Actor Key

Two octets, LACP Actor Key for the corresponding Aggregator, as specified in [[IEEE-802.1AX](#)] [section 5.3.5](#).

- Member Ports Priority

Two octets, LACP administrative port priority associated with all interfaces bound to the Aggregator. This field is valid only when the "Flags" field has "Priority Set" asserted.

- Flags

Valid values are:

- i. Synchronized (0x01)

Indicates that the sender has concluded transmitting all Aggregator configuration information.

- ii. Purge Configuration (0x02)

Indicates that the Aggregator is no longer configured for mLACP operation.

- iii. Priority Set (0x04)

Indicates that the "Member Ports Priority" field is valid.

- Agg Name Len

One octet, length of the "Aggregator Name" field in octets.

- Aggregator Name

Aggregator name encoded in UTF-8 format, up to a maximum of 20 characters. Used for ease of management.

#### [7.2.5. mLACP Port Config TLV](#)

The mLACP Port Config TLV is sent in the RG Application Data message. This TLV is used to notify RG peers about the local configuration state of a port.



```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0033           |   Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Port Number           |   MAC Address           |
+-----+-----+
|
+-----+-----+
|   Actor Key           |   Port Priority           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Port Speed           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Flags   | Port Name Len |   Port Name           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~
|
|   ...
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0033 for "mLACP Port Config TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Port Number

Two octets, LACP Port Number for the corresponding interface as specified in [[IEEE-802.1AX](#)] [section 5.3.4](#). The Port Number MUST be encoded with the Node ID as was discussed above.

- MAC Address

Six octets encoding the port MAC address.

- Actor Key

Two octets, LACP Actor Key for the corresponding interface, as specified in [[IEEE-802.1AX](#)] [section 5.3.5](#).



- Port Priority

Two octets, LACP administrative port priority for the corresponding interface, as specified in [[IEEE-802.1AX](#)] [section 5.3.4](#). This field is valid only when the "Flags" field has "Priority Set" asserted.

- Port Speed

Four octets integer encoding the port's current bandwidth in units of 1,000,000 bits per second. This field corresponds to the ifHighSpeed object of IF-MIB [[RFC2863](#)].

- Flags

Valid values are:

- i. Synchronized (0x01)

Indicates that the sender has concluded transmitting all member link port configurations for a given Aggregator.

- ii. Purge Configuration (0x02)

Indicates that the port is no longer configured for mLACP operation.

- iii. Priority Set (0x04)

Indicates that the "Port Priority" field is valid.

- Port Name Len

One octet, length of the "Port Name" field in octets.

- Port Name

Port (interface) name encoded in UTF-8 format, up to a maximum of 20 characters.

#### [7.2.6](#). mLACP Port Priority TLV

The mLACP Port Priority TLV is sent in the RG Application Data message. This TLV is used by a device to either advertise its operational Port Priority to other members in the RG, or to authoritatively request that a particular member of an RG change its port priority.



```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0034           |   Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           OpCode           |   Port Number       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Aggregator ID    |   Last Port Priority   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Current Port Priority    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0034 for "mLACP Port Priority TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- OpCode

Two octets identifying the operational code-point for the TLV, encoded as follows:

```

0x00 Local Priority Change Notification
0x01 Remote Request for Priority Change

```

- Port Number

2 octets field representing the LACP Port Number as specified in [IEEE-802.1AX] [section 5.3.4](#). When the value of this field is 0, it denotes all ports bound to the Aggregator specified in the "Aggregator ID" field. When non-zero, the Port Number MUST be encoded with the Node ID as was discussed above.

- Aggregator ID

Two octets, LACP Aggregator Identifier as specified in [IEEE-802.1AX] [section 5.4.6](#)





#### - Last Port Priority

Two octets, LACP port priority for the corresponding interface, as specified in [IEEE-802.1AX] [section 5.3.4](#). For local ports, this field encodes the previous operational value of port priority. For remote ports, this field encodes the operational port priority last known to the PE via notifications received from its peers in the RG.

#### - Current Port Priority

Two octets, LACP port priority for the corresponding interface, as specified in [IEEE-802.1AX] [section 5.3.4](#). For local ports, this field encodes the new operational value of port priority being advertised by the PE. For remote ports, this field specifies the new port priority being requested by the PE.

### [7.2.7. mLACP Port State TLV](#)

The mLACP Port State TLV is used in the RG Application Data message. This TLV is used by a device to report its LACP port status to other members in the RG.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0035           |   Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Partner System ID    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Partner System Priority |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Partner Port Number       |   Partner Port Priority   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Partner Key               | Partner State | Actor State |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Actor Port Number         |   Actor Key              |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Selected   | Port State   |   Aggregator ID         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

#### - U and F Bits

Both are set to 0.



- Type

set to 0x0035 for "mLACP Port State TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Partner System ID

6 octets, the LACP Partner System ID for the corresponding interface, encoded as a MAC address as specified in [IEEE-802.1AX] [section 5.4.2.2](#) item r.

- Partner System Priority

2 octets field specifying the LACP Partner System Priority as specified in [IEEE-802.1AX] [section 5.4.2.2](#) item q.

- Partner Port Number

2 octets encoding the LACP Partner Port Number as specified in [IEEE-802.1AX] [section 5.4.2.2](#) item u. The Port Number MUST be encoded with the Node ID as was discussed above.

- Partner Port Priority

2 octets field encoding the LACP Partner Port Priority as specified in [IEEE-802.1AX] [section 5.4.2.2](#) item t.

- Partner Key

2 octets field representing the LACP Partner Key as defined in [IEEE-802.1AX] [section 5.4.2.2](#) item s.

- Partner State

1 octet field encoding the LACP Partner State Variable as defined in [IEEE-802.1AX] [section 5.4.2.2](#) item v.

- Actor State

1 octet encoding the LACP Actor's State Variable for the port as specified in [IEEE-802.1AX] [section 5.4.2.2](#) item m.



- Actor Port Number

2 octets field representing the LACP Actor Port Number as specified in [[IEEE-802.1AX](#)] [section 5.3.4](#). The Port Number MUST be encoded with the Node ID as was discussed above.

- Actor Key

2 octet field encoding the LACP Actor Operational Key as specified in [[IEEE-802.1AX](#)] [section 5.3.5](#).

- Selected

1 octet encoding the LACP 'Selected' variable, defined in [[IEEE-802.1AX](#)] [section 5.4.8](#), as follows:

0x00 SELECTED  
0x01 UNSELECTED  
0x02 STANDBY

- Port State

1 octet encoding the operational state of the port as follows:

0x00 Up  
0x01 Down  
0x02 Administrative Down  
0x03 Test (e.g. IEEE 802.3ah OAM Intrusive Loopback mode)

- Aggregator ID

Two octets, LACP Aggregator Identifier to which this port is bound based on the outcome of the LACP selection logic.

#### **[7.2.8. mLACP Aggregator State TLV](#)**

The mLACP Aggregator State TLV is used in the RG Application Data message. This TLV is used by a device to report its Aggregator status to other members in the RG.



```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0037                               |   Length   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Partner System ID |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Partner System Priority |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|       Partner Key                               |       Aggregator ID       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|       Actor Key                               |       Agg State       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0037 for "mLACP Aggregator State TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Partner System ID

6 octets, the LACP Partner System ID for the corresponding interface, encoded as a MAC address as specified in [IEEE-802.1AX] [section 5.4.2.2](#) item r.

- Partner System Priority

2 octets field specifying the LACP Partner System Priority as specified in [IEEE-802.1AX] [section 5.4.2.2](#) item q.

- Partner Key

2 octets field representing the LACP Partner Key as defined in [IEEE-802.1AX] [section 5.4.2.2](#) item s.

- Aggregator ID

Two octets, LACP Aggregator Identifier as specified in [IEEE-802.1AX] [section 5.4.6](#)





- Actor Key

2 octet field encoding the LACP Actor Operational Key as specified in [[IEEE-802.1AX](#)] [section 5.3.5](#).

- Agg State

1 octet encoding the operational state of the Aggregator as follows:

0x00 Up

0x01 Down

0x02 Administrative Down

0x03 Test (e.g. IEEE 802.3ah OAM Intrusive Loopback mode)

#### **[7.2.9](#). mLACP Synchronization Request TLV**

The mLACP Synchronization Request TLV is used in the RG Application Data message. This TLV is used by a device to request from its peer to re-transmit configuration or operational state. The following information can be requested:

- system configuration and/or state
- configuration and/or state for a specific port
- configuration and/or state for all ports with a specific LACP key
- configuration and/or state for all mLACP ports
- configuration and/or state for a specific aggregator
- configuration and/or state for all aggregators with a specific LACP key
- configuration and/or state for all mLACP aggregators

The format of the TLV is as follows:



```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0038                               |   Length   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Request Number                               |C|S|   Request Type   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Port Number / Aggregator ID   |               Actor Key           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0038 for "mLACP Synchronization Request TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Request Number

2 octets. Unsigned integer uniquely identifying the request. Used to match the request with a response. The value of 0 is reserved for unsolicited synchronization, and MUST NOT be used in the mLACP Synchronization Request TLV.

- C Bit

Set to 1 if request is for configuration data. Otherwise, set to 0.

- S Bit

Set to 1 if request is for running state data. Otherwise, set to 0.

- Request Type

14-bits specifying the request type, encoded as follows:



0x00 Request System Data  
0x01 Request Aggregator Data  
0x02 Request Port Data  
0x3FFF Request All Data

- Port Number / Aggregator ID

2 octets. When Request Type field is set to 'Request Port Data', this field encodes the LACP Port Number for the requested port. When the Request Type field is set to 'Request Aggregator Data', this field encodes the Aggregator ID of the requested Aggregator. When the value of this field is 0, it denotes that all ports (or Aggregators), whose LACP Key is specified in the "Actor Key" field, are being requested.

- Actor Key

Two octets, LACP Actor key for the corresponding port or Aggregator. When the value of this field is 0 (and the Port Number/Aggregator ID field is 0 as well), it denotes that information for all ports or Aggregators in the system is being requested.

#### **7.2.10. mLACP Synchronization Data TLV**

The mLACP Synchronization Data TLV is used in the RG Application Data message. A pair of these TLVs is used by a device to delimit a set of TLVs that are being transmitted in response to an mLACP Synchronization Request TLV. The delimiting TLVs signal the start and end of the synchronization data, and associate the response with its corresponding request via the 'Request Number' field.

The mLACP Synchronization Data TLVs are also used for unsolicited advertisements of complete mLACP configuration and operational state data. The 'Request Number' field MUST be set to 0 in this case. For such unsolicited synchronization, the PE MUST advertise all system, Aggregator and port information as done during the initialization sequence.

This TLV has the following format:



```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F|   Type=0x0039               |   Length               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Request Number               |   Flags                 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- U and F Bits

Both are set to 0.

- Type

set to 0x0039 for "mLACP Synchronization Data TLV"

- Length

Length of the TLV in octets excluding the U-bit, F-bit, Type, and Length fields.

- Request Number

2 octets. Unsigned integer identifying the Request Number from the "mLACP Synchronization Request TLV" which solicited this synchronization data response.

- Flags

2 octets, response flags encoded as follows:

0x00 Synchronization Data Start

0x01 Synchronization Data End

## 8. LDP Capability Negotiation

As required in [\[RFC5561\]](#) the following TLV is defined to indicate the ICCP capability:

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F| TLV Code Point=0x700       |   Length               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|S| Reserved   |   Reserved   | VER/Maj   |   Ver/Min   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```





where:

- U-bit

SHOULD be 1 (ignore if not understood).

- F-bit

SHOULD be 0 (don't forward if not understood).

- TLV Code Point

The TLV type, which identifies a specific capability. The ICCP code point is requested in the IANA allocation section below.

- S-bit The State Bit indicates whether the sender is advertising or withdrawing the ICCP capability. The State bit is used as follows:

- 1 - The TLV is advertising the capability specified by the TLV Code Point.

- 0 - The TLV is withdrawing the capability specified by the TLV Code Point.

- Ver/Maj

The major version revision of the ICCP protocol, this document specifies 1.0. This field is then set to 1

- Ver/Min

The minor version revision of the ICCP protocol, this document specifies 1.0. This field is then set to 0

ICCP capability is advertised to a LDP peer if there is at least one RG enabled on the local PE.

## **9. Client Applications**

### **9.1. Pseudowire Redundancy Application Procedures**

This section defines the procedures for the Pseudowire Redundancy (PW-RED) Application.

It should be noted that the PW-RED application SHOULD NOT be enabled together with an AC Redundancy application for the same service instance. This simplifies the operation of the multi-chassis



redundancy solution (Figure 1) and eliminates the possibility of deadlock conditions between the AC and PW redundancy mechanisms.

#### **9.1.1. Initial Setup**

When an RG is configured on a system and multi-chassis pseudowire redundancy is enabled in that RG, the PW-RED application MUST send an "RG Connect" message with "PW-RED Connect TLV" to each PE that is a member of the same RG. The sending PE MUST set the A bit to 1 if it has already received a "PW-RED Connect TLV" from its peer; otherwise, the PE MUST set the A bit to 0. If a PE, that has sent the TLV with the A bit set to 0, receives a "PW-RED Connect TLV" from a peer, it MUST repeat its advertisement with the A bit set to 1. The PW-RED application connection is considered to be operational when both PEs have sent and received "PW-RED Connect TLVs" with the A bit set to 1. Once the application connection becomes operational, the two devices can start exchanging "RG Application Data" messages for the PW-RED application.

If a system receives an "RG Connect" message with "PW-RED Connect TLV" that has a differing Protocol Version, it must follow the procedures outlined in the "Application Versioning" section above.

When the PW-RED application is disabled on the device, or is unconfigured for the RG in question, the system MUST send an "RG Disconnect" message with "PW-RED Disconnect TLV".

#### **9.1.2. Pseudowire Configuration Synchronization**

A system MUST advertise its local PW configuration to other PEs that are members of the same RG. This allows the PEs to build a view of the redundant nodes and pseudowires that are protecting the same service instances. The advertisement MUST be initiated when the PW-RED application connection first comes up. To that end, the system sends "RG Application Data" messages with "PW-RED Config TLVs" as part of an unsolicited synchronization. A PE MUST use a pair of "PW-RED Synchronization Data TLVs" to delimit the set of TLVs that are being sent as part of this unsolicited advertisement.

In the case of a configuration change, a PE MUST re-advertise the most up to date information for the affected pseudowires.

As part of the configuration synchronization, a PE advertises the ROID associated with the pseudowire. This is used to correlate the pseudowires that are protecting each other on different PEs. As well,



a PE advertises the configured PW redundancy mode. This can be one of the following four options: Master Mode, Slave Mode, Independent Mode or Independent Mode with Request Switchover. If the received redundancy mode does not match the locally configured mode for the same ROID, then the PE MUST respond with an "RG Notification Message" to NAK the "PW-RED Config TLV". The PE MUST disable the associated local pseudowire until a satisfactory "PW-RED Config TLV" is received from the peer. This guarantees that device mis-configuration does not lead to network wide problems (e.g. by creating forwarding loops). The PE SHOULD also raise an alarm to alert the operator. If a PE receives a NAK for an advertised "PW-RED Config TLV", it MUST disable the associated pseudowire and SHOULD raise an alarm to alert the operator.

Furthermore, a PE advertises in its "PW-RED Config TLVs" a priority value that is used to determine the precedence of a given pseudowire to assume the Active role in a redundant setup. A PE also advertises a Service Name that is global in the context of an RG and is used to identify which pseudowires belong to the same service. Finally, a PE also advertises the pseudowire identifier as part of this synchronization.

#### **9.1.3. Pseudowire Status Synchronization**

PEs, that are member of an RG, synchronize pseudowire status for the purpose of identifying, on a per ROID basis, which pseudowire will be actively used for forwarding and which pseudowire(s) will be placed in standby state.

Synchronization of pseudowire status is done by sending the "PW-RED State TLV" whenever the pseudowire state changes on a PE. This includes changes to the local end as well as the remote end of the pseudowire.

A PE may request that its peer retransmit previously advertised PW-RED state. This is useful for instance when the PE is recovering from a soft failure. To request such retransmission, a PE MUST send a set of one or more "PW-RED Synchronization Request TLVs".

A PE MUST respond to a "PW-RED Synchronization Request TLV" by sending the requested data in a set of one or more PW-RED TLVs delimited by a pair of "PW-RED Synchronization Data TLVs". The TLVs comprising the response MUST be ordered such that the Synchronization Response TLV with the "Synchronization Data Start" flag precedes the various other PW-RED TLVs encoding the requested data. These, in turn, MUST precede the Synchronization Data TLV with the "Synchronization Data End" flag. It is worth noting that the response



may span across multiple RG Application Data messages; however, the above TLV ordering MUST be retained across messages, and only a single pair of Synchronization Data TLVs must be used to delimit the response across all Application Data Messages.

A PE MAY re-advertise its PW-RED state in an unsolicited manner. This is done by sending the appropriate config and state TLVs delimited by a pair of "PW-RED Synchronization Data TLVs" and using a 'Request Number' of 0.

While a PE has a pending synchronization request for a pseudowire or a service, it SHOULD silently ignore all TLVs for said pseudowire or service that are received prior to the synchronization response and which carry the same type of information being requested. This saves the system from the burden of updating state that will ultimately be overwritten by the synchronization response. Note that TLVs pertaining to other pseudowires or services are to continue to be processed per normal in the interim.

If a PE receives a synchronization request for a pseudowire or service that doesn't exist or is not known to the PE, then it MUST trigger an unsolicited synchronization of all pseudowire information (i.e. replay the initialization sequence).

In the subsections that follow, we describe the details of pseudowire status synchronization for each of the PW redundancy modes defined in [\[RED-BIT\]](#).

#### **[9.1.3.1](#). Independent Mode**

This section covers the operation in Independent Mode with or without Request Switchover capability.

In this mode, the operator must ensure that for a given RO, the PW Priority values configured for all associated pseudowires on a given PE are collectively higher (or lower) than those configured on other PEs in the same RG. If this condition is not satisfied after the PEs have exchanged "PW-RED State TLVs", a PE MUST disable the associated pseudowire(s) and SHOULD raise an alarm to alert the operator. Note that the PW Priority MAY be the same as the PW Precedence defined in [\[RED-BIT\]](#).

For a given RO, after the all the PEs in an RG have exchanged their "PW-RED State TLVs", the PE with the best PW Priority (i.e. least numeric value) advertises Active preferential forwarding status in LDP on all its associated pseudowires. Whereas, all other PEs in the RG advertise Standby preferential forwarding status in LDP on their





associated pseudowires.

If the service is VPWS, then only a single pseudowire per service will be selected for forwarding. This is the pseudowire that is independently advertised with Active preferential forwarding status on both endpoints, as described in [\[RED-BIT\]](#).

If the service is VPLS, then one or multiple pseudowires per service will be selected for forwarding. These are the pseudowires that are independently advertised with Active preferential forwarding status on both PW endpoints, as described in [\[RED-BIT\]](#).

#### **[9.1.3.2. Master/Slave Mode](#)**

In this mode, the operator must ensure that for a given R0, the PW Priority values configured for all associated pseudowires on a given PE are collectively higher (or lower) than those configured on other PEs in the same RG. If this condition is not satisfied after the PEs have exchanged "PW-RED State TLVs", a PE MUST disable the associated pseudowire(s) and SHOULD raise an alarm to alert the operator. Note that the PW Priority MAY be the same as the PW Precedence defined in [\[RED-BIT\]](#). In addition, the operator must ensure that, for a given R0, all the PEs in the RG are consistently configured as Master or Slave.

In the context of a given R0, if the PEs in the RG are acting as Master, then the PE with the best PW Priority (i.e. least numeric value) advertises Active preferential forwarding status in LDP on only a single pseudowire, following the procedures in sections [5.2](#) and [6.2](#) of [\[RED-BIT\]](#). Whereas, all the other pseudowires on other PEs in the RG are advertised with Standby preferential forwarding status in LDP.

#### **[9.1.4. PE Node Failure](#)**

When a PE node detects that a remote PE, that is member of the same RG, has gone down, the local PE examines if it has redundant PWs for the affected services. If the local PE has the highest priority (after the failed PE) then it becomes the active node for the services in question, and subsequently activates its associated PW(s).



## **9.2. Attachment Circuit Redundancy Application Procedures**

### **9.2.1. Common AC Procedures**

This section describes generic procedures for AC Redundancy applications, independent of the type of the AC (ATM, FR or Ethernet).

#### **9.2.1.1. AC Failure**

When the AC Redundancy mechanism on the Active PE detects a failure of the AC, it should send an ICCP Application Data message to inform the redundant PEs of the need to take over. The AC failures can be categorized into the following scenarios:

- Failure of CE interface connecting to PE
- Failure of CE uplink to PE
- Failure of PE interface connecting to CE

#### **9.2.1.2. PE Node Failure**

When a PE node detects that a remote PE, that is member of the same RG, has gone down, the local PE examines if it has redundant ACs for the affected services. If the local PE has the highest priority (after the failed PE) then it becomes the active node for the services in question, and subsequently activates its associated ACs.

#### **9.2.1.3. PE Isolation**

When a PE node detects that it has been isolated from the core network (i.e. all core facing interfaces/links are not operational), then it should ensure that its AC Redundancy mechanism will change the status of any active ACs to Standby. The AC Redundancy application should then send ICCP Application Data messages in order to trigger failover to a standby PE.

#### **9.2.1.4. Determining Pseudowire State**

If the PEs in an RG are running an AC Redundancy application over ICCP, then the Independent Mode of PW Redundancy, as defined in [\[RED-BIT\]](#), MUST be used. On a given PE, the Preferential Forwarding status of the PW (Active or Standby) is derived from the state of the



associated AC(s). This simplifies the operation of the multi-chassis redundancy solution (Figure 1) and eliminates the possibility of deadlock conditions between the AC and PW redundancy mechanisms. The rules by which the PW status is derived from the AC status are as follows:

- VPWS

For VPWS, there's a single AC per service instance. If the AC is Active, then the PW status should be Active. If the AC is Standby, then the PW status should be Standby.

- VPLS

For VPLS, there could be multiple ACs per service instance (i.e. VFI). If AT LEAST ONE AC is Active, then the PW status should be Active. If ALL ACs are Standby, then the PW status should be Standby.

In this case, the PW-RED application is not used to synchronize PW status between PEs. Rather, the AC Redundancy application should synchronize AC status between PE, in order to establish which AC (and subsequently which PE) is Active or Standby for a given service. When that is determined, each PE will then derive its local PWs state according to the rules described above. The Preferential Forwarding status bit, described in [[RED-BIT](#)], is used to advertise PW status to the remote peers.

### **[9.2.2. Multi-chassis LACP \(mLACP\) Application Procedures](#)**

This section defines the procedures that are specific to the multi-chassis LACP (mLACP) application, which is applicable for Ethernet ACs.

#### **[9.2.2.1. Initial Setup](#)**

When an RG is configured on a system and mLACP is enabled in that RG, the mLACP application MUST send an "RG Connect" message with "mLACP Connect TLV" to each PE that is member of the same RG. The sending PE MUST set the A bit to 1 in the said TLV if it has received a corresponding "mLACP Connect TLV" from its peer PE; otherwise, the sending PE MUST set the A bit to 0. If a PE receives an "mLACP Connect TLV" from its peer after sending the said TLV with the A bit set to 0, it MUST resend the TLV with the A bit set to 1. A system considers the mLACP application connection to be operational when it



has sent and received "mLACP Connect TLVs" with the A bit set to 1. When the mLACP application connection between a pair of PEs is operational, the two devices can start exchanging "RG Application Data" messages for the mLACP application. This involves having each PE advertise its mLACP configuration and operational state in an unsolicited manner. A PE SHOULD subscribe to the following order when advertising its mLACP state upon initial application connection setup:

- Advertise system configuration
- Advertise Aggregator configuration
- Advertise port configuration
- Advertise Aggregator state
- Advertise port state

A PE MUST use a pair of "mLACP Synchronization Data TLVs" to delimit the entire set of TLVs that are being sent as part of this unsolicited advertisement.

If a system receives an "RG Connect" message with "mLACP Connect TLV" that has a differing Protocol Version, it MUST follow the procedures outlined in the "Application Versioning" section above.

After the mLACP application connection has been established, every PE MUST communicate its system level configuration to its peers via the use of "mLACP System Config TLV". This allows every PE to discover the Node ID and the locally configured System ID and System Priority values of its peers.

If a PE receives an "mLACP System Config TLV" from a remote peer advertising the same Node ID value as the local system, then the PE MUST respond with an "RG Notification Message" to NAK the "mLACP System Config TLV". The PE MUST suspend the mLACP application until a satisfactory "mLACP System Config TLV" is received from the peer. It SHOULD also raise an alarm to alert the operator. Furthermore, if a PE receives a NAK for an "mLACP System Config TLV" that it has advertised, the PE MUST suspend the mLACP application and SHOULD raise an alarm to alert the network operator of potential device mis-configuration.

If a PE receives an "mLACP System Config TLV" from a new peer advertising the same Node ID value as another existing peer with which the local system has an established mLACP Application connection, then the PE MUST respond to the new peer with an "RG Notification Message" to NAK the "mLACP System Config TLV" and MUST ignore the offending TLV.

If the Node ID of a particular PE changes due to administrative





configuration action, the PE MUST then inform its peers to purge the configuration of all previously advertised ports and/or aggregators, and MUST replay the initialization sequence by sending an unsolicited synchronization of: the system configuration, Aggregator configuration, port configuration, Aggregator state and port state.

It is necessary for all PEs in an RG to agree upon the System ID and System Priority values to be used ubiquitously. To achieve this, every PE MUST use the values for the two parameters that are supplied by the PE with the numerically lowest value (among RG members) of System Aggregation Priority. This guarantees that the PEs always agree on uniform values, which yield the highest System Priority.

When the mLACP application is disabled on the device, or is unconfigured for the RG in question, the system MUST send an "RG Disconnect" message with "mLACP Disconnect TLV".

#### **9.2.2.2. mLACP Aggregator and Port Configuration**

A system MUST synchronize the configuration of its mLACP enabled Aggregators and ports with other RG members. This is achieved via the use of "mLACP Aggregator Config TLVs" and "mLACP Port Config TLVs", respectively. An implementation MUST advertise the configuration of Aggregators prior to advertising the configuration of any of their associated member ports.

The PEs in an RG MUST all agree on the MAC address to be associated with a given Aggregator. It is possible to achieve this via consistent configuration on member PEs. However, in order to protect against possible misconfiguration, a system MUST use, for any given Aggregator, the MAC address supplied by the PE with the numerically lowest System Aggregation Priority in the RG.

A system that receives an "mLACP Aggregator Config TLV" with an ROID to Key association that is different from its local association MUST NAK the corresponding TLV and disable the Aggregator with the same ROID. Furthermore, it SHOULD raise an alarm to alert the operator. Similarly, a system that receives a NAK in response to a transmitted "mLACP Aggregator Config TLV" MUST disable the associated Aggregator and SHOULD raise an alarm to alert the network operator.

A system MAY enforce a restriction that all ports that are to be bundled together on a given PE share the same Port Priority value. If so, the system MUST advertise this common priority in the "mLACP Aggregator Config TLV" and assert the "Priority Set" flag in such TLV. Furthermore, the system in this case MUST NOT advertise individual Port Priority values in the associated "mLACP Port Config



TLVs" (i.e. the "Priority Set" flag in these TLVs should be 0).

A system MAY support individual Port Priority values to be configured on ports that are to be bundled together on a PE. If so, the system MUST advertise the individual Port Priority values in the appropriate "mLACP Port Config TLVs", and MUST NOT assert the "Priority Set" flag in the corresponding "mLACP Aggregator Config TLV".

When the configurations of all ports for member links associated with a given Aggregator have been sent by a device, it asserts that fact by setting the "Synchronized" flag in the last port's "mLACP Port Config TLV". If an Aggregator doesn't have any candidate member ports configured, this is indicated by asserting the "Synchronized" flag in its "mLACP Aggregator Config TLV".

Furthermore, for a given port/Aggregator, an implementation MUST advertise the port/Aggregator configuration prior to advertising its state (via the "mLACP Port State TLV" or "mLACP Aggregator State TLV"). If a PE receives an "mLACP Port State TLV" or "mLACP Aggregator State TLV" for a port or Aggregator that it had not learned of before via an appropriate Port or Aggregator Config TLV, then the PE MUST request synchronization of the configuration and state of all mLACP ports as well as all mLACP Aggregators from its respective peer. If during a synchronization (solicited or unsolicited), a PE receives a State TLV for a port or Aggregator that it has not learned of before, then the PE MUST send a NAK for the offending TLV. The PE MUST NOT request re-synchronization in this case.

When mLACP is unconfigured on a port/Aggregator, a PE MUST send a "Port/Aggregator Config TLV" with the "Purge Configuration" flag asserted. This allows receiving PEs to purge any state maintained for the decommissioned port/Aggregator. If a PE receives a "Port/Aggregator Config TLV" with the "Purge Configuration" flag asserted, and the PE is not maintaining any state for that port/Aggregator, then it MUST silently discard the TLV.

#### **9.2.2.3. mLACP Aggregator and Port Status Synchronization**

PEs within an RG need to synchronize their state-machines for proper mLACP operation with a multi-homed device. This is achieved by having each system advertise its Aggregators and ports running state in "mLACP Aggregator State TLVs" and "mLACP Port State TLVs", respectively. Whenever any LACP parameter for an Aggregator or a port, whether on the Partner (i.e. multi-homed device) or the Actor (i.e. PE) side, is changed a system MUST transmit an updated TLV for the affected Aggregator and/or port. Moreover, when the



administrative or operational state of an Aggregator or port changes, the system MUST transmit an updated Aggregator or port state TLV to its peers.

If a PE receives an Aggregator or port state TLV where the 'Actor Key' doesn't match what was previously received in a corresponding Aggregator or port config TLV, the PE MUST then request synchronization of the configuration and state of the affected Aggregator or port. If such a mismatch occurs between the config and state TLVs as part of a synchronization (solicited or unsolicited), then the PE MUST send a NAK for the state TLV. Furthermore, if a PE receives a port state TLV with the 'Aggregator ID' set to a value that doesn't map to some Aggregator that the PE had learned of via a previous Aggregator config TLV, then the PE MUST request synchronization of the configuration and state of all Aggregators and ports. If the above anomaly occurs during a synchronization, then the PE MUST send a NAK for the offending port state TLV.

A PE MAY request that its peer retransmit previously advertised state. This is useful for example when the PE is recovering from a soft failure and attempting to relearn state. To request such retransmissions, a PE MUST send a set of one or more "mLACP Synchronization Request TLVs".

A PE MUST respond to an "mLACP Synchronization Request TLV" by sending the requested data in a set of one or more mLACP TLVs delimited by a pair of "mLACP Synchronization Data TLVs". The TLVs comprising the response MUST be ordered in the RG Application Data message(s) such that the Synchronization Response TLV with the "Synchronization Data Start" flag precedes the various other mLACP TLVs encoding the requested data. These, in turn, MUST precede the Synchronization Data TLV with the "Synchronization Data End" flag. Note that the response may span across multiple RG Application Data messages, for example when MTU limits are exceeded; however, the above ordering MUST be retained across messages, and only a single pair of Synchronization Data TLVs MUST be used to delimit the response across all Application Data Messages.

A PE device MAY re-advertise its mLACP state in an unsolicited manner. This is done by sending the appropriate Config and State TLVs delimited by a pair of "mLACP Synchronization Data TLVs" and using a 'Request Number' of 0.

While a PE has a pending synchronization request for a system, Aggregator or port, it SHOULD silently ignore all TLVs for said system, Aggregator or port that are received prior to the synchronization response and which carry the same type of information being requested. This saves the system from the burden of updating



state that will ultimately be overwritten by the synchronization response. Note that TLVs pertaining to other systems, Aggregators or ports are to continue to be processed per normal in this case.

If a PE receives a synchronization request for an Aggregator, port or Key that doesn't exist or is not known to the PE, then it **MUST** trigger an unsolicited synchronization of all system, Aggregator and port information (i.e. replay the initialization sequence).

If a PE learns, as part of a synchronization operation from its peer, that the latter is advertising a Node ID value which is different from the value previously advertised, then the PE **MUST** purge all port/aggregator data previously learnt from that peer prior to the last synchronization.

#### **9.2.2.4. Failure and Recovery**

When a PE that is active for a multi-chassis link aggregation group encounters a core isolation fault, it **SHOULD** attempt to fail-over to a peer PE which hosts the same R0. The default fail-over procedure is to have the failed PE bring down the link(s) towards the multi-homed CE (e.g. by bringing down the line-protocol). This will cause the CE to fail-over to the other member link(s) of the bundle that are connected to the other PE(s) in the RG. Other procedures for triggering fail-over are possible, and are outside the scope of this document.

Upon recovery from a previous fault, a PE **MAY** reclaim active role for a multi-chassis link aggregation group if configured for revertive protection. Otherwise, the recovering PE may assume standby role when configured for non-revertive protection. In the revertive scenario, a PE **SHOULD** assume active role within the RG by sending an "mLACP Port Priority TLV" to the currently active PE, requesting that the latter change its port priority to a value that is lower (i.e. numerically larger) for the Aggregator in question.

If a system is operating in a mode where different ports of a bundle are configured with different Port Priorities, then the system **MUST NOT** advertise or request change of Port Priority values for aggregated ports collectively (i.e. by using a 'Port Number' of 0 in the "mLACP Port Priority TLV"). This is to avoid ambiguity in the interpretation of the 'Last Port Priority' field.

If a PE receives an "mLACP Port Priority TLV" requesting a priority change for a port or Aggregator that is not local to the device, then the PE **MUST** re-advertise the local configuration of the system, as well as the configuration and state of all its mLACP ports and





Aggregators.

If a PE receives an "mLACP Port Priority TLV" in which the remote system is advertising priority change for a port or Aggregator that the local PE had not learned of before via an appropriate Port or Aggregator Config TLV, then the PE MUST request synchronization of the configuration and state of all mLACP ports as well as all mLACP Aggregators from its respective peer.

## **10. Security Considerations**

The security considerations described in [[RFC5036](#)] and [[RFC4447](#)] that apply to the base LDP specification, and to the PW LDP control protocol extensions apply to the capability mechanism described in this document.

The ICCP protocol is not intended to be applicable when the redundancy group spans PE in different administrative domains. Furthermore, implementations SHOULD provide a mechanism to select to which LDP peers the ICCP capability will be advertised, and from which LDP peers the ICCP messages will be accepted.

## **11. IANA Considerations**

### **11.1. MESSAGE TYPE NAME SPACE**

This document uses several new LDP message types, IANA already maintains a registry of name "MESSAGE TYPE NAME SPACE" defined by [[RFC5036](#)]. The following values are suggested for assignment:

Message type	Description
0x0700	RG Connect Message
0x0701	RG Disconnect Message
0x0702	RG Notification Message
0x0703	RG Application Data Message

### **11.2. TLV TYPE NAME SPACE**

This document use a new LDP TLV type, IANA already maintains a registry of name "TLV TYPE NAME SPACE" defined by [[RFC5036](#)]. The following value is suggested for assignment:

TLV Type	Description
0x700	ICCP capability TLV.



### 11.3. ICC RG Parameter Type Space

IANA needs to set up a registry of "ICC RG parameter type". These are 14-bit values. Parameter Type values 1 through 0x000F are specified in this document, Parameter Type values 0x0010 through 0x1FFF are to be assigned by IANA, using the "Expert Review" policy defined in [\[RFC5226\]](#). Parameter Type values 0x2000 through 0x2FFF, 0x3FFF, and 0 are to be allocated using the IETF consensus policy defined in [\[RFC5226\]](#). Parameter Type values 0x3000 through 0x3FFE are reserved for vendor proprietary extensions and are to be assigned by IANA, using the "First Come First Served" policy defined in [\[RFC5226\]](#). A Parameter Type description is required for any assignment from this registry. Additionally, for the vendor proprietary extensions range a citation of a person or company name is also required. A document reference should also be provided.

Initial ICC RG parameter type space value allocations are specified below:

Parameter Type	Description	Reference
-----	-----	-----
0x0001	ICC Sender Name	[RFCxxxx]
0x0002	NAK TLV	[RFCxxxx]
0x0003	Requested Protocol Version TLV	[RFCxxxx]
0x0004	Disconnect Code TLV	[RFCxxxx]
0x0005	ICC RG ID TLV	[RFCxxxx]
0x0010	PW-RED Connect TLV	[RFCxxxx]
0x0011	PW-RED Disconnect TLV	[RFCxxxx]
0x0012	PW-RED Config TLV	[RFCxxxx]
0x0013	Service Name TLV	[RFCxxxx]
0x0014	PW ID TLV	[RFCxxxx]
0x0015	Generalized PW ID TLV	[RFCxxxx]
0x0016	PW-RED State TLV	[RFCxxxx]
0x0017	PW-RED Synchronization Request TLV	[RFCxxxx]
0x0018	PW-RED Synchronization Data TLV	[RFCxxxx]
0x0019	PW-RED Disconnect Cause TLV	[RFCxxxx]
0x0030	mLACP Connect TLV	[RFCxxxx]
0x0031	mLACP Disconnect TLV	[RFCxxxx]
0x0032	mLACP System Config TLV	[RFCxxxx]
0x0033	mLACP Port Config TLV	[RFCxxxx]
0x0034	mLACP Port Priority TLV	[RFCxxxx]
0x0035	mLACP Port State TLV	[RFCxxxx]
0x0036	mLACP Aggregator Config TLV	[RFCxxxx]
0x0037	mLACP Aggregator State TLV	[RFCxxxx]
0x0038	mLACP Synchronization Request TLV	[RFCxxxx]
0x0039	mLACP Synchronization Data TLV	[RFCxxxx]



0x003A

mLACP Disconnect Cause TLV

[RFCxxxx]

#### **11.4. STATUS CODE NAME SPACE**

This document use several new Status codes, IANA already maintains a registry of name "STATUS CODE NAME SPACE" defined by [[RFC5036](#)]. The following values is suggested for assignment: The "E" column is the required setting of the Status Code E-bit.

Range/Value	E	Description	Reference
-----	----	-----	-----
0x00010001	0	Unknown ICCP RG	
0x00010002	0	ICCP Connection Count Exceeded	
0x00010003	0	ICCP Application Connection Count Exceeded	
0x00010004	0	ICCP Application not in RG	
0x00010005	0	Incompatible ICCP Protocol Version	
0x00010006	0	ICCP Rejected Message	
0x00010007	0	ICCP Administratively Disabled	
0x00010010	0	ICCP RG Removed	
0x00010011	0	ICCP Application Removed from RG	

#### **12. Acknowledgments**

The authors wish to acknowledge the important contributions of Dennis Cai, Neil McGill, Amir Maleki, Dan Biagini, Robert Leger, Sami Boutros, Neil Ketley and Mark Christopher Sains.

The authors also thank Daniel Cohn, Lizhong Jin and Ran Chen for the valuable input, discussions and comments.

#### **13. Normative References**

[RFC5036] L. Andersson et al, "LDP Specification", [RFC 5036](#), October 2007.

[RFC5561] "LDP Capabilities", [RFC5561](#), July 2009.

[RFC4447] "Transport of Layer 2 Frames Over MPLS", Martini, L., et al., [rfc4447](#) April 2006.

[IEEE-802.1AX] IEEE Std. 802.1AX-2008, "IEEE Standard for Local and metropolitan area networks- Link Aggregation", IEEE Computer Society, November 2008.



[RFC2863] K. McCloghrie, F. Kastenholz, "The Interfaces Group MIB",  
[rfc2863](#), June 2000.

#### **14. Informative References**

[RED-BIT] Praveen Muley, Mustapha Aissaoui, "Pseudowire  
Preferential Forwarding Status Bit",  
[draft-ietf-pwe3-redundancy-bit-07.txt](#), May 2012,  
Work in progress.

[RFC5880] D. Katz, D. Ward, "Bidirectional Forwarding Detection",  
[RFC5880](#), June 2010

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an  
IANA Considerations section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008

#### **15. Author's Addresses**

Luca Martini  
Cisco Systems, Inc.  
9155 East Nichols Avenue, Suite 400  
Englewood, CO, 80112  
e-mail: [lmartini@cisco.com](mailto:lmartini@cisco.com)

Samer Salam  
Cisco Systems, Inc.  
595 Burrard Street, Suite 2123  
Vancouver, BC V7X 1J1  
Canada  
e-mail: [ssalam@cisco.com](mailto:ssalam@cisco.com)

Ali Sajassi  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134  
e-mail: [sajassi@cisco.com](mailto:sajassi@cisco.com)

Matthew Bocci  
Alcatel-Lucent  
Grove House, Waltham Road Rd  
White Waltham, Berks, UK. SL6 3TN  
e-mail: [matthew.bocci@alcatel-lucent.co.uk](mailto:matthew.bocci@alcatel-lucent.co.uk)





Satoru Matsushima  
Softbank Telecom  
1-9-1, Higashi-Shinbashi, Minato-ku  
Tokyo 105-7313, JAPAN  
e-mail: satoru.matsushima@gmail.com

Thomas Nadeau  
Juniper Networks  
10 Technology Park Drive,  
Westford, MA 01886  
USA  
e-mail: tnadeau@juniper.net

#### Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Expiration Date: April 2014

