Neighbor Discovery

OIF-ND-IA-01.0

August 19, 2015

Implementation Agreement created and approved
by the Optical Internetworking Forum
www.oiforum.com

The OIF is an international non profit organization with over 90 member companies, including the world's leading carriers and vendors. Being an industry group uniting representatives of the data and optical worlds, OIF's purpose is to accelerate the deployment of interoperable, cost-effective and robust optical internetworks and their associated technologies. Optical internetworks are data networks composed of routers and data switches interconnected by optical networking elements.

With the goal of promoting worldwide compatibility of optical internetworking products, the OIF actively supports and extends the work of national and international standards bodies. Working relationships or formal liaisons have been established with IEEE 802.1, IEEE 802.3ba, IETF, IP-MPLS Forum, IPv6 Forum, ITU-T SG13, ITU-T SG15, MEF, ATIS-OPTXS,ATIS-TMOC, TMF and the XFP MSA Group.

For additional information contact:
The Optical Internetworking Forum, 48377 Fremont Blvd.,
Suite 117, Fremont, CA 94538
510-492-4040 ☞ info@oiforum.com

www.oiforum.com

Working Group: **Networking & Operations**

TITLE: **Neighbor Discovery**

SOURCE: **TECHNICAL EDITORS**

Jonathan Sadler
Coriant
1415 W. Diehl Rd
Naperville, IL 60563
Phone: +1 630 798 6182
jonathan.sadler@coriant.com

WORKING GROUP CHAIR

Evelyne Roch
Huawei
evelyne.roch@huawei.com

ABSTRACT: This implementation agreement specifies the functional requirements and protocol to perform Neighbor Discovery.

Table of Contents

List of Figures

List of Tables

1 Introduction

Successful setup of services across a network requires understanding the network topology, equipment capability, current link availability as well as capacity. Determining this information from a previously unknown state is commonly called network or topology discovery. A number of different mechanisms can be used to perform network discovery, including self-publication of topology information into a distributed database or topology development through polling a list of network elements.

A network topology is comprised of Network Elements and the bidirectional facilities that interconnect them. Each network element has different capabilities including the ability to switch traffic between links as well as the ability to encapsulate/decapsulate traffic. When the ports on two network elements interconnected by a facility have common capability (e.g. the support of SONET OC48; the support of 100Mb Ethernet PHY), a link exists.

1.1 Problem Statement

Service providers desire the ability to determine the existence of a link dynamically, removing dependency on manually developed records. Additionally, the service providers desire the ability to limit the amount of manual configuration required to use a link. Automating the process of identifying a link and exchanging the configuration data is known as neighbor discovery.

As a result of neighbor discovery, the following information can be developed:

- Identity of the network element port connected to the far end of the link
- Data-plane capability of the network element ports on each link ends
- Management-plane details for each link end
- Control-plane details for each link end

1.2 Scope

The scope of this implementation agreement is to define the requirements for Neighbor Discovery, including architecture, procedure and use cases.

1.2.1 Items not in scope

The following areas are NOT covered within this document:

- Requirements for inter-carrier interfaces. The extensions in this document were defined within the framework of intra-carrier protocol requirements for ASON.

1.3 Relationship to Other Standards Bodies

This document, to the maximum extent possible, uses standards and specifications already available from other organizations. Specifically,

- The overall discovery process is based on ITU-T specification [G.7714] [G.7714.1]
- The SDH/SONET overheads used for discovery are from ITU-T specification [G.707] and ANSI specification [T1.105].
- The OTN (ODUk) overheads used for discovery is from ITU-T specification [G.709Ed4].
- The Ethernet protocol used for discovery is from IEEE specification [IEEE802.1AB].

This version of the implementation agreement also documents private extensions, codepoints and formats of these extensions based on the OIF E-NNI implementation agreements [OIF-E-NNI-Sig-02.0] [OIF-E-NNI-OSPF-02.0].

It is the intent of OIF to develop E-NNI protocols in close alignment with ITU-T Recommendations, and foundation IETF RFCs. As such, the OIF has aligned formats with IETF and ITU-T standard specifications where possible and will continue to pursue alignment with standards in its future work. As additional standard specifications become available that address functions included in this Implementation Agreement, additional revisions for further alignment with these standards will be considered.

1.4 Merits to OIF

The E-NNI Neighbor Discovery implementation agreement is a key step towards the implementation of an open inter-domain interface that allows offering dynamic setup and release of various services. This activity supports the overall mission of the OIF.

1.5 Working Groups

Networking & Operations Working Group

Carrier Working Group

Interoperability Working Group

1.6 Document Organization

This document is organized as follows:

- Section 1: Introduction and Scope of the Document
- Section 2: Terminology and Abbreviations
- Section 3: Functional Elements and Communication Channels
- Section 4: Discovery Process
- Section 5: Use Cases
- Section 6: Requirements
- Section 7: Discovery Protocol
- Section 9: Discovery Trigger and ITU-T recommendations
- Section 8: References

1.7 Keywords

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 [RFC2119].

2 Terminology and Abbreviations

2.1 Definitions

The following terms are used in this implementation agreement:

Control Domain
: This terminology is adopted from ITU-T [G.8080]. A type of transport domain where the criterion for membership is the scope of a control plane component responsible for the transport resources within the transport domain.

Inter-domain Link
: A link with endpoints in two different Routing Areas at a particular level of the routing hierarchy.

Intra-domain Link
: A link with both endpoints within the same Routing Area at a particular level of the routing hierarchy.

Layer
: This terminology is adopted from ITU-T [G.805]. A layer (network) is a ”topological component“ that represents the complete set of access groups of the same type which may be associated for the purpose of transferring information.

Level
: This terminology is adopted from ITU-T [G.8080]. A routing hierarchy describes the relationships between an RA and a containing RA or contained RAs. RAs at the same depth within the routing hierarchy are considered to be at the same routing level.

Node ID
: This terminology is adopted from ITU-T [G.7715.1]. The Node ID identifies a node in the transport topology graph. A node may represent either an abstraction of a Routing Area or a subnetwork.

Protocol Controller
: This terminology is adopted from ITU-T [G.8080]. The Protocol Controller provides the function of mapping the parameters of the abstract interfaces of the control components into messages carried by a protocol to support interconnection via an interface.

Signaling Control Network (SCN)
: The packet network that carries control plane messages between Protocol Controllers

Signaling Controller
: Signaling Controller (see [G.8080])

TE Link
: This definition is per [RFC4203], which defines a TE link as a “logical“ link that has TE properties. The TE link is logical in a sense that it represents a way to group or map the information about certain physical resources (and their properties) into the information used by Constrained SPF for path computation.

2.2 Abbreviations

The following abbreviations are used in this implementation agreement.

ASON	Automatically Switched Optical Networks
CC	Connection Controller
E-NNI	External Network-Network Interface
IETF	Internet Engineering Task Force
I-NNI	Internal Network-Network Interface
IP	Internet Protocol
ITU-T	International Telecommunications Union - Telecommunications
LRM	Link Resource Manager
LSA	Link State Advertisement
NNI	Network-Network Interface
OSPF	Open Shortest Path First
PC	Protocol Controller
RA	Routing Area
RSVP	ReSource reserVation Protocol
SCN	Signaling Communications Network

3 Functional Elements and Communication Channels

The primary function that performs neighbor discovery is known as the Discovery Agent (DA). The DA does not need to be integrated into a Network Element – it can be located on a management system or an external controller of the network element. Furthermore, the relationship between Discovery Agent and Network Element is m:n, enabling more than one Network Element to be managed by a Discovery Agent as well as for more than one Discovery Agent to manage separate ports on the Network Element. There is a limit of a 1:1 relationship between a Discovery Agent and a layer termination point located within the Network Element.

The majority of the communication between Discovery Agents may be performed using in-band or out-of-band channels, but will require in-band channels to perform the discovery trigger. This communication is facilitated through two "Protocol Controllers" that understand the detail of the channel.

More about how these channels are used is described in the following section.

4 Discovery Process

Developing the detailed information about a link is done by three high-level phases, as defined by ITU-T G.7714:

- Discovery Trigger
- Layer Adjacency Discovery (LADJ)
- Transport Capability Exchange (TCE)

4.1 Discovery Trigger

The Discovery Trigger is an in-band indication sent across a link. The specifics defining a Discovery Trigger is technology dependent, but it always contains the identity of the network element port transmitting the discovery trigger. The receiver of a discovery trigger message processes the message recognizing the network element port where the message was received. Between these two pieces of information, it is possible to identify the existence of a unidirectional link between the far network element port and near network element port.

The Discovery Trigger is performed using signal formats defined for a specific technology. Since it is carried in-band in one direction, it enables a unidirectional link to be discovered within a specific layer. It is expected the network operator will only connect a single transmitter to a single receiver.

4.2 Layer Adjacency Discovery (LADJ)

While a unidirectional link may exist between two ports, it is possible that a bidirectional link does not. This can occur in two cases: 1) the reflective unidirectional link has not been installed or 2) the reflective unidirectional link is incorrectly connected. Figure 1 shows a correctly connected link (in blue) as well as a link that falls into case 1 (in red). Figure 2 shows two different scenarios for case 2: a link where the reflective unidirectional link is connected to a different port on the same system (in blue) and a link where the reflective unidirectional link is connected to a different port on a different system (in red).

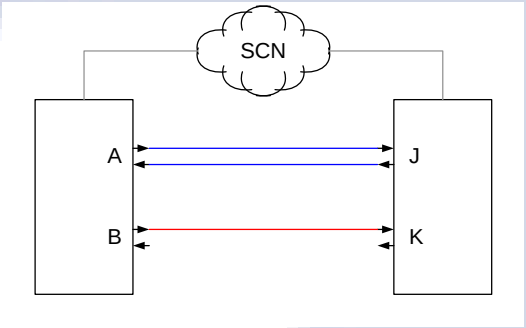


Figure 1. Correctly connected bidirectional link and incorrectly connected unidirectional link

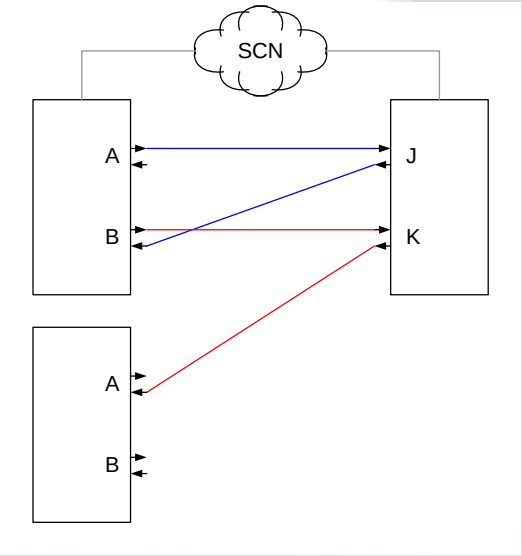


Figure 2. Incorrectly connected links on different port and different system

Identifying a bidirectional link requires correlating two unidirectional links with reflective near/far port identity information. Therefore, the existence of the unidirectional links needs to be known by a common actor. This actor may be located anywhere, including in a management system or directly in the network element.

In many cases, two domains connected by a bi-directional link will have separate discovery processes at each end. To support this, adjacency discovery allows for separate actors to exist for each end of the bidirectional link. G.7714 defines a set of abstract messages used by a pair of actors to perform Layer Adjacency Discovery.

4.3 Transport Capability Exchange

Once a bidirectional link has been identified in a layer, the discovery process can continue with identifying how the link may be used. Different network elements will have different capabilities – some will be able to switch a signal from this link to another, others will have the ability to generate/terminate a signal and still others have the ability to be flexibly configured to do either switching or terminating. This exchange allows both ends of the links to match-up their capabilities and deduce the existence additional bidirectional links.

Transport Capability Exchange may also used to provide additional configuration information for the identified link end, including details of specific data-plane variations in use (e.g. SDH vs SONET overhead processing), the management plane configuration and control plane configuration associated with the link end.

5 Example use cases for the neighbor discovery process

The sections below describe the specific use cases for neighbor discovery that need to be supported. These use cases are described in their "positive" form, with no description of mis-wiring or DCN failure cases.

The use cases described require support for one or more of the protocols supporting the G.7714 Neighbor Discovery phases. Table 1 shows the relationship to the different phases.

Section	DT	LADJ	TCE
5.1.1 Single bidirectional link between two NEs	X		
5.1.2 Single bidirectional link between two NEs under control of a common discovery agent	X		
5.1.3 Single bidirectional link between two NEs with internal discovery agents	X	X	
5.1.4 Single bidirectional link between two NEs with external discovery agents	X	X	
5.2.1 Single bidirectional link between two NEs – TCE Protocol and Adjacency Maintenance	X	X	X
5.2.2 Single bidirectional link between two NEs – Authentication/Authorization	X	X	X
5.2.3 Single bidirectional link between two NEs - Link-end Capability Exchange	X	X	X
5.2.4 Single bidirectional link between two NEs - Mgmt Plane Config Exchange	X	X	X
5.2.5 Single bidirectional link between two NEs - Control Plane Config Exchange	X	X	X
5.2.6 Multiple bidirectional links between two NEs with control plane configuration exchange	X	X	X
5.3 Renegotiation	X	X	X
5.4 Rediscovery	X	X	X
5.5 Recovery of DA Adjacency	X	X	X

Table 1. Use Case relationship to G.7714 Phases

5.1 Overall Use-Cases

5.1.1 Single bidirectional link between two NEs

This example use case is focused on the overall message exchange between discovery agents and does not assume where the discovery agent is located.

The Discovery Agents have two communications methods available for the discovery process: an in-band trail trace that is used as the discovery trigger and an in-band or out-of-band mechanism to carry LADJ and TCE messages. To isolate the Discovery Agent from the specific details of how the communication methods are realized, this use case defines two separate transmit/receive handlers. The first is specific to the in-band DT exchange while the second is for the in or out-of-band LADJ/TCE exchange.

The high level message sequence (based on G.7714.1, Figure I.1) is shown below:

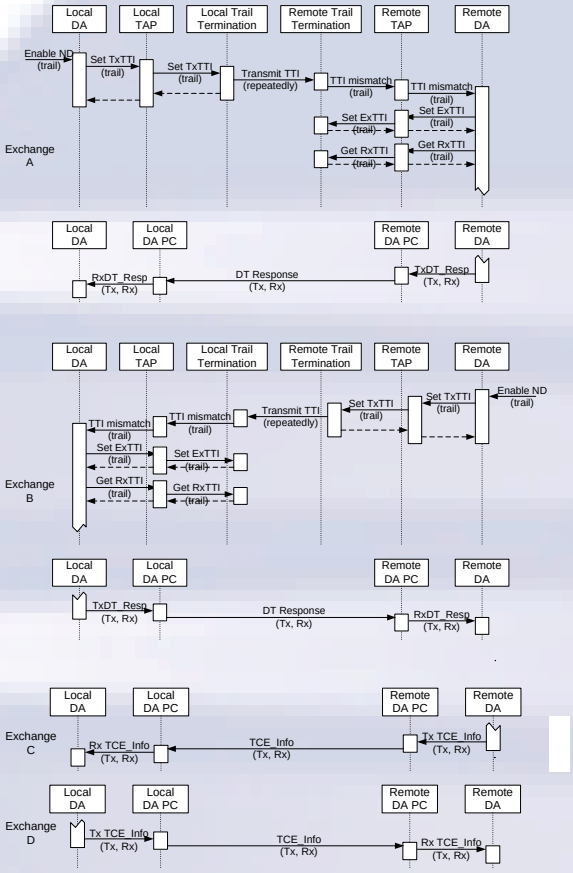


Figure 3. Message Sequence Diagram for DT/LAD/TCE

The MSD details four distinct message exchanges. Exchanges A & B show the LADJ phase while Exchanges C & D shows the TCE phase.

The Discovery Trigger and LADJ phases are initiated whenever a layer termination point[1] is given to a Discovery Agent or the layer termination point returns to in-service from an out-of-service condition. This enables the Discovery

[1] A termination point may be a port or it may be a TTP located in a higher layer above a point of flexibility. The termination point supports the access point named by an ifIndex. A G.800 Topological Link (TL) places an ifIndex into a node (which, by extension, is within a network) and gives the ifIndex a network wide address.

Agent to learn about any changes to the remote end of a link resulting from changes to the underlying connection and trail.

Within the LADJ phase, the message exchanges for a link end operate independently. When each end has successfully correlated the unidirectional link provided via the LADJ message exchange into a bidirectional link, it will progress to TCE. If the unidirectional link is not successfully correlated due to mis-wiring, the LADJ phase will repeat until the successful correlation occurs. When both-ends have successfully performed correlation, the bi-directional link is considered to be discovered.

Within the TCE phase, the message exchanges performed on behalf of a link end also operate independently. Each exchange is used by one end of the link to report the neighbor its configuration, and for the neighbor to request modification of the configuration. While many configurable items can be modified (e.g. negotiation of the Area a link will be in), some items will be immutable (e.g. the ability to terminate a specific layer) or may not be supported by the neighbor (e.g. the specific variant of signaling protocol to use for on link). This is ok – the neighbor can accept the detail provided in the configuration report message as "advisory only".

When an endpoint is "upgraded" and adds new capability, it is necessary to be able to renegotiate the configuration for the link without affecting connections already active on the link.

Since the DA Adjacency is out-of-band, there needs to be a mechanism to identify if the far end DA has lost state during a DCN failure. This is accomplished by placing a random number known as a session identifier in each of the Configuration messages sent in a TCE exchange. This session identifier exists separately for each TCE phase but remains the same for the lifetime of the DA.

The TCE phase decomposes into a number of sub-phases. These are:

- Authentication/Authorization. This optional phase is performed to determine the non-refutable identity of the far endpoint and validate it is authorized to connect to this port.
- Link-end capability exchange. This phase is performed to identify links that exist in higher layers than the layer where DT/LADJ was performed. As an example, a SONET/SDH link operating over an OTUk interface will perform DT/LADJ in the OTUk layer and will have links in the ODUk as well as VC3/VC4/VC4-nc layers. This phase is critical to help identify additional misconnection scenarios, including the connection of a SONET/SDH over OTUk interface to an Ethernet over OTUk interface.
- Management Plane configuration exchange. This phase allows for Management information to be exchanged. It is described further in Section 5.2.4 below.
- Control Plane configuration exchange. This multi-phase exchange allows for Control Plane information to be exchanged. It is further described in Section 5.2.5 below.

Three sub-use cases exist, based on the location of the Discovery Agent. These are described in the following sections.

5.1.2 Single bidirectional link between two NEs under control of a common discovery agent

This example use case builds on the example use case defined in Section 5.1. As in the previous use case the location of the discovery agent is not specified. This use case recognizes the special case where both endpoints of a link are under the management of a common discovery agent. In an SDN environment, two switches may be under the control of a common SDN controller; in that case the controller may function as a common discovery agent for both endpoints of the links between the switches. This applies whether the switches are in the same domain or are in different domains, provided that both domains are under a common SDN controller.

In this use case, many of the phases of the discovery process become internal to the discovery agent and do not require standardization. The impact on discovery is the following:

1. Discovery Trigger processing in Exchange A and B still applies
2. Discovery Trigger response in Exchange A and B is not required
3. TCE exchanges in Exchange C and D is not required

5.1.3 Single bidirectional link between two NEs with internal discovery agents

This example use case builds on the example use case defined in Section 5.1.1. In the earlier use case, the location of the discovery agent is not specified. In this use case, a discovery agent on the near end is located on the NE, and is operating on the behalf of all ports on the NE.

With the discovery agent on the NE, all of the interfaces between functional entities (DA, DA PC, TAP) associated with a specific link end are internal APIs and do not need to be externally specified.

Because the Discovery Agents associated with the link are on different NEs, there is a need for a protocol operating over a data-network to support the exchange of LADJ and TCE information.

5.1.4 Single bidirectional link between two NEs with external discovery agents

This example use case builds on the one defined in Section 5.1.1. In the earlier use case, the location of the discovery agent is not specified. In this use case, a discovery agent on the near end is located off of the NE (for example on a management system), and is operating on the behalf of a set of ports on the NE.

With the discovery agent external to the NE, the interfaces between the DA and DA PC are located on a different platform than the TAP. As a result, the interface between the DA and TAP needs to be externally specified.

5.2 TCE Exchange Use Cases

The following sections provide specific use cases for each of the TCE Exchanges. These are written for a single bi-directional link. Section 5.2.6 extends the use cases to handle multiple bidirectional links.

5.2.1 Single bidirectional link between two NEs – TCE Protocol and Adjacency Maintenance

This example use case provides detail on the overall TCE protocol and Adjacency Maintenance.

The TCE protocol continuously operates, from the time the LADJ exchange is complete until the link is declared down (either by fault or by administrative request). This continuous operation is necessary to allow for 1) renegotiation of capabilities for a link and 2) to allow for graceful shutdown of a link.

In order to validate DCN connectivity for the TCE Protocol, the protocol must include a heartbeat/echo exchange mechanism. Heartbeat/echo exchange messages are expected to be received periodically, with the specific period for transmission configurable by the network operator, as well as the specific period of time that can pass without receiving a message. When the TCE Protocol Adjacency is lost, it is reported as a minor alarm for the link until the Adjacency is restored.

5.2.2 Single bidirectional link between two NEs – Authentication/Authorization

This example use case provides detail on the exchange of identity information, allowing for authorization of the remote link end for connection to this port.

Prior to the TCE Protocol, the Discovery Agent only has access to course mechanisms to determine if the link being discovered is acceptable under the policies of the network operator. These mechanisms allow for rogue NEs masquerading as another NE to be connected. The Authentication/Authorization phase allows for identity validation mechanisms to be used, including non-refutable cryptographic techniques.

The IETF has a rich history of providing advanced cryptographic mechanisms for identity validation, including simple challenge protocols such as CHAP and extensible authentications protocols such as EAP. The Authentication/Authorization mechanism utilizes these protocols, with initial support for CHAP and future support for EAP-TLS and EAP-TTLS.

Authentication Exchange is performed as two separate unidirectional exchanges, with the near end validating the far end and the far end (optionally) validates the near end. Both ends are required to have support for cryptographic mechanisms. With advanced protocols such as EAP-TLS and EAP-TTLS, the near end and (optionally) the far end utilize digital certificates in order to digitally sign the cryptographic messages.

As a part of the authentication process, identity verification and authorization is performed. This can be done on the NE using a pre-stored/pre-shared set of peer entity information, or it can be done using a RADIUS server.

A Network Element may not need to support an Authentication/Authorization mechanism. However, for forward compatibility, it does need to include the ability to negotiate the Authentication/Authorization mechanism, facilitating interoperability.

5.2.3 Single bidirectional link between two NEs - Link-end Capability Exchange

This example use case provides detail specific to the Link-end Capability exchange defined in Section 5.1. This use case describes the information exchanged to determine what potential client layer links exist above the layer link discovered by DT/LADJ.

Each link-end has a series of layer termination, client to server layer adaptation and layer switching functions, starting with the discovered link. A viable client layer link requires a set of these capabilities, starting with the discovered link and going upward though adaptation to client layer link to match.

To determine if a match exists, each endpoint will provide to the other endpoint its capabilities. The other endpoint then performs the match and determines what links exist.

An example of this match is shown in Figure 4.

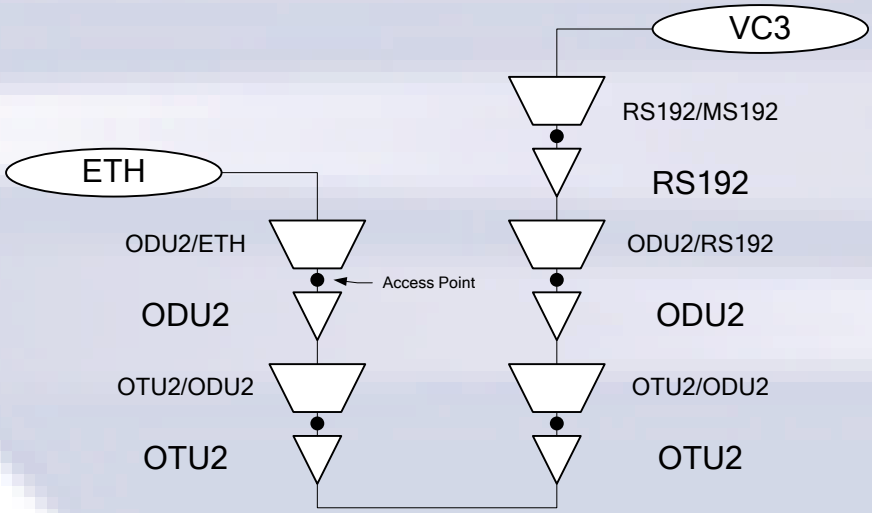


Figure 4. Example of incompatible endpoint capabilities

5.2.4 Single bidirectional link between two NEs - Mgmt Plane Config Exchange

This example use case provides detail specific to the Management Plane configuration exchange defined in Section 5.1. This use case describes the information exchanged in support of the management plane.

After establishing what client layer links exist, the Management system has the opportunity to pass configuration information about the link-ends. This information facilitates links between two NEs managed by different OSSes as well as for OSSes that don't have awareness of TCP ID to port name mapping. This information can be:

- OSS System Name (Optional)
- NE Name (e.g. TL1 TID)
- Discovered Link-end name (e.g. TL1 AID)

Since links in each layer have unique endpoints and therefore potentially different link-end names (e.g. an OTU2 link-end has a separate name from an ODU2 link-end), each discovered link (i.e. per layer) may separately provide Discovered Link-end name.

5.2.5 Single bidirectional link between two NEs - Control Plane Config Exchange

This example use case builds on the example use case defined in Section 5.1. This use case describes the information exchanged to configure the control plane to use the discovered link. The Control Plane Config Exchange consists of two different sets of information: Endpoint name and Control Plane adjacency configuration.

Endpoint names are dependent on understanding how a link fits into the overall network structure. This is due to the way that Node IDs are scoped to an Area. Without determining the Area of the link, it is impossible to know which Node ID should be exchanged. For this reason, the Control Plane Config Exchange is done as a two-phase exchange. The first phase establishes the Area of the link while the second phase provides the specific configuration detail.

The Area of the link is determined by both endpoints providing their area hierarchy as an ordered list, starting with leaf and going to root. This information is examined by each link end to identify the lowest (i.e. closest to leaf) area in common between the two link-ends. If no areas are in common, then control plane config exchange cannot proceed. When this occurs, a TCE failure condition will be raised for the link.

As an example, if two NEs exchange the following ordered lists:

NE A: (0.1.1.0, 0.1.2.0, 0.1.3.0)

NE B: (0.1.1.5, 0.1.2.0, 0.1.3.0)

The lowest area id that is in common is 0.1.2.0. This is the area of the link.

Once the area is determined, the second phase exchange will pass the specific NodeID and IfIndex for the link as well as attribute information for the link (e.g. Cost, Resource Class, SRG), the parameters required for the signaling adjacency and (optionally) the parameters required for any routing adjacencies.

Attribute information for the link may either be separately configured on each end of the link and validated with the other end of the link, or configured on one end of the link and provided to the other end. Single ended configuration is accomplished using PPP's convention of sending a value of ZERO or NULL for a parameter when the local end doesn't have any configuration detail, and the remote end sending a ConfNAK, with the suggested value to use instead. This can be used for Cost and SRG as they both require non-zero and non-null values. However, this cannot be used for Resource Class, as Resource Class allows a value of ZERO. Instead, Resource Class will need to have an "unconfigured" flag added to the value exchanged by TCE. This allows an unconfigured NE to notify its peer it is looking for configuration information.

5.2.6 Multiple bidirectional links between two NEs with control plane configuration exchange

This use case discusses what occurs when more than one bidirectional link exists between two NEs. For most of the TCE Exchanges, there is no difference. However, for Control Plane, there is additional information that may be exchanged to facilitate configuration of bundled links.

This example use case builds on the example use case defined in Section 5.2.5. In the earlier use case, the specific configuration information exchanged for a single control plane link specified. In this use case, the information exchanged when there are multiple parallel links is described.

It is quite common for multiple parallel links to exist between two NEs. This is done to provide additional capacity as well as to remove single points of failure. The control plane uses Bundled links to aggregate multiple links into a single resource pool that can be advertised by routing.

Bundled links require the links being aggregated to have common attributes, making them equivalent for routing purposes. This means as links in the same layer are discovered by TCE with the same Area, Peer Node ID, SRG, Resource Class and Cost they may be placed in a bundled link. The resulting bundled link has a different link-end identifier from the component links, but all other link attributes are the same. A bundled link does not have its own signaling or routing adjacency. Signaling is done using the adjacency established for the specific component link a connection is being established on.

It should be noted, there is no requirement that all links support the same client layer links (e.g. an OC48 link and an OC192 link may be bundled. Additionally two OC48 links, one supporting STS-48c while the other does not, may also be bundled). When common client layer links are supported by links in a bundle, their capacity will be aggregated in routing announcements.

Once a link end has identified that a bundled link is possible, it will invoke a third phase exchange to negotiate the creation of a bundle as well as the configuration information for the bundle.

5.3 Renegotiation

The network elements at the ends of a link may be hardware capable for many features, but not software capable. As a result, a new software release may enable additional features not previously supported. This means a link end for which the DA previously performed TCE may have new capabilities to be conveyed. To handle this, the TCE protocol should allow for renegotiation.

G.7714 defines a state machine in Figure I-2 of Appendix I that extends the LCP state machine found in RFC 1661. This utilizes the same messages as already defined for LCP, but allows for renegotiation without resetting LCP to the closed state.

The state machine in Figure I-2 of G.7714 and the RFC1661 LCP state machine are interoperable, enabling an implementation to initially use LCP and transition in the future to use of the G.7714 specified state machine.

5.4 Rediscovery

When a link fails, the link will be rediscovered when the link is repaired. Until the rediscovery occurs, the previously discovered information will be retained until there is a reason to invalidate that information. Maintaining the previously discovered information is necessary to allow routing over failed links to be accommodated.

Rediscovery is performed in two stages: Dataplane verification and TCE verification.

Dataplane verification is performed using the DT as described in Section 4.1. If the far endpoint data-plane identifier (i.e. TCP ID) is found to have change, the link will not be brought back into service. Instead a mismatch alarm will be raised, allowing the operator to address the change. If the change is allowed, the link will be reset, with all active connections failed.

TCE Verification is performed by having each xCP (starting with LCP, then Authentication, etc.) send a CONFREQ with the last negotiated parameters sent by the local NE including a Session identifier. If any parameter, including the session identifier, is different than what was previously received, a CONFREJ will be sent indicating the Session identifier as the offending option, the xCP will be declared down to higher-layer protocols and negotiation will start from scratch.

5.5 Recovery of DA Adjacency

Since the DA Adjacency is out-of-band, a failure of link isn't expected when the DA Adjacency fails. As a result, there needs to be a mechanism to trigger rediscovery. When the DA Adjacency alarm has been cleared, the second stage of rediscovery (TCE Verification) will be performed. As stated in Section 5.4, any difference what was previously received will cause a CONFREJ will be sent indicating the Session identifier as the offending option, the xCP will be declared down to higher-layer protocols and negotiation will start from scratch.

6 Requirements

The following requirements were considered in the development of this Implementation Agreement. The requirements are subdivided to correspond to the use cases provided in Section 5.1.

6.1 All use cases

The requirements in this section apply to all use cases described in Section 5.1.

6.1.1 General Requirements

Req#	Status	Description
ND-1	Req	Neighbor Discovery shall provide a mechanism to automatically configure links connected between two termination points.

6.1.1.1 Functional Component Requirements

Req#	Status	Description
ND-1.1	Req	Neighbor Discovery shall be provided using one or two Discovery Agents (DA).
ND-1.2	Req	A Discovery Agent shall provide an interface to enable/disable, configure global parameters and retrieve the configuration of global parameters.

6.1.2 Data Plane Related Requirements[2]

Req#	Status	Description
ND-2.1	Req	Neighbor Discovery shall support operation on termination points associated with links.
ND-2.1.1	Req	Neighbor Discovery shall support TDM links.

[2] This section states requirements the Neighbor Discovery *protocol* definition needs to meet, not requirements for a Neighbor Discovery *implementation*. It does not mandate an *implementation* support all technologies listed. For example, while ND-2.1.4 requires the Neighbor Discovery *protocol* support Ethernet links, a Neighbor Discovery *implementation* may be scoped to a different set of layers (e.g. OTN) and therefore does not need to comply.

Req#	Status	Description
ND-2.1.2	Req	Neighbor Discovery shall support operation via Section/RS termination points on SONET/SDH physical links, including: OC3/STM-1, OC12/STM-4, OC48/STM-16, OC192/STM-64.
ND-2.1.3	Req	Neighbor Discovery shall support operation via OTU termination points on OTU physical links operating using white-light optics, including: OTU1, OTU2, OTU3.
ND-2.1.4	Req	Neighbor Discovery shall support Ethernet links.
ND-2.2	Req	Neighbor Discovery shall support operation on Termination Points associated with a logical resource (i.e. non-physical links).
ND-2.2.1	Req	Neighbor Discovery shall support operation via SONET/SDH termination points on logical resources, including STS1/VC3, STS3c/VC4, STS12c/VC4-4c, STS48c/VC4-16c and STS192c/VC4-64c.
ND.2.2.2	Req	Neighbor Discovery shall support operation via Section/RS termination points on SONET/SDH logical resources operating over OTN, including RS48 (over ODU1) and RS192 (over ODU2).
ND-2.2.3	Req	Neighbor Discovery shall support operation via ODU termination points on logical resources, including ODU1, ODU2, ODU3 and ODU4.
ND-2.2.4	Req	Neighbor Discovery shall support operation via Ethernet termination points on logical resources operating over OTN, including ETH over ODU1 and ETH over ODU2.
ND-2.3	Req	Neighbor Discovery shall provide an interface to enable/disable, configure and retrieve configuration associated with a termination point.
ND-2.4	Req	Neighbor Discovery shall provide specific status indications and conditions per termination point.
ND-2.5	Req	Neighbor Discovery shall provide management system interfaces to retrieve discovery status indications and conditions per termination point.
ND-2.6	Req	Neighbor Discovery shall provide management system interfaces to autonomously report error conditions and the clearing of error conditions per termination point.

6.1.3 Discovery Trigger Requirements

Req#	Status	Description
ND-3	Req	The DA shall discover the existence of links between termination points on two NEs.
ND-3.1	Req	The DA shall support discovery of uni-directional links.
ND-3.2	Req	The DA shall support discovery of bi-directional links. Note: There are two different types of bi-directional links: co-routed and associated. This distinction describes how the underlying transmit and receive components are routed between the common termination endpoints. This routing is invisible to the discovery process, making the distinction of no consequence to discovery. Hence, no distinction is made in the requirements.
ND-3.2.1	Req	The DA shall automatically determine if the transmit and receive components of a bidirectional link that are connected to a local termination point are connected to the same remote termination point. If they are not, the bidirectional link is “miswired”.
ND-3.2.2	Req	The DA shall report an MISWIRED condition on a link that is discovered to be miswired.
ND-3.2.3	Req	The DA shall clear the MISWIRED condition when the link is determined by the DA to no longer be miswired.
ND-3.3	Req	The DA shall support the Discovery Trigger mechanism as described in G.7714.1.
ND-3.3.1	Req	The DA shall seize the in-band channel and initiate sending Discovery Trigger messages when a link initially has discovery enabled.
ND-3.3.2	Req	The DA shall seize the in-band channel and initiate sending Discovery Trigger messages when a link with discovery enabled goes returns to in-service.

Req#	Status	Description
ND-3.3.3	Req	The DA shall yield the in-band channel and stop sending Discovery Trigger messages when a link completes LADJ. Note: The TTI overhead bytes may be shared by DT transmission and other applications. The DA will yield the TTI overhead bytes, causing management configured TTI values to be exchanged when the DT/LADJ process has completed. When a common DA is supporting both ends of a discovered trail, LADJ is an internal process. When LADJ messaging is in use, this requires the receipt of a LADJ message for this port which contains the same TCP ID as the TCP ID received in the DT message received on this port (i.e. is not misconnected/miswired).
ND-3.3.4	Req	The DA shall use an in-band channel for Discovery Trigger Messages.
ND-3.3.4.1	Req	The DA shall support use of the TTI overhead as the in-band channel carrying Discovery Trigger Messages.
ND-3.3.4.1.1	Req	The DA shall utilize the RS TTI as the in-band channel for SONET/SDH physical ports.
ND-3.3.4.1.2	Req	The DA shall utilize the OTU TTI as the in-band channel for OTU physical ports.
ND-3.3.4.1.4	Req	The DA shall configure the Sent_TTI.
ND-3.3.4.1.5	Req	The DA shall allow the user to configure the TTI value to be sent when DT is not in progress.
ND-3.3.4.1.5.1	Req	The DA shall prevent the user from configuring a TTI value that starts with a plus (+) character while ND is enabled on an endpoint.
ND-3.3.4.1.6	Req	The DA shall automatically switch from using the User-configured TTI value to the DT when rediscovery is performed.

Req#	Status	Description
ND-3.3.4.1.7	Req	The DA shall automatically switch from using the DT to the User-configured TTI value when initial discovery or rediscovery is completed. Note: Initial discovery and rediscovery are considered completed when the DA has determined this port is not mis-wired. When a common DA is supporting both ends of a discovered trail, this is an internal process. When LADJ messaging is in use, this requires the receipt of a LADJ message for this port which contains the same TCP ID as the TCP ID received in the DT message received on this port.
ND-3.3.4.2	Req	The DA shall support use of the ECC overhead as the in-band channel carrying Discovery Trigger Messages.
ND-3.3.4.2.1	Req	The DA shall utilize the OTU GCC0 as the in-band channel for OTU physical ports.
ND-3.3.4.2.2	Req	The DA shall utilize the RS DCC as the in-band channel for SONET/SDH physical ports.
ND-3.3.4.3	Req	The DA shall support use of in-band link-discovery protocols for carrying Discovery Trigger Messages.
ND-3.3.5	Req	The DA shall transmit a termination point's identity over an in-band channel associated with the termination point to a remote system.
ND-3.3.6	Req	The DA shall receive from the in-band channel the identity of the connected, far termination point while the DT/LADJ phase is in progress.
ND-3.3.6.1	Req	The DA shall configure an Expected_TTI receive value. Note: This will cause the NE to generate an error condition when the received TTI value does not match the received value.
ND-3.3.6.2	Req	The DA shall configure TTI-mismatch-AIS-insertion as DISABLED while the DT/LADJ phase is in-progress.
ND-3.3.6.3	Req	The DA shall receive TTI mismatch indications from the NE for resources that have enabled Neighbor Discovery while the DT/LADJ phase is in progress.
ND-3.3.6.4	Req	The DA shall retrieve the Received_TTI upon receiving a TTI mismatch indication.
ND-3.3.6.5	Req	The DA shall configure the Expected_TTI value with the Received_TTI value to clear the mismatch condition while the DT/LADJ phase is in progress.

Req#	Status	Description
ND-3.4	Req	The DA shall support the Layer Adjacency Discovery mechanism as described in G.7714.1. Note: When a common DA is supporting both ends of a discovered trail, this is an internal process and does not require the LADJ protocol.
ND-3.4.1	Req	The DA shall determine the far DA associated with a far termination point identified over an in-band channel observed at a local termination point. Note: This method used to determine the far DA is dependent on the identification format provided in the in-band channel.

6.1.4 Policy Requirements

ND-4	Req	The DA shall utilize a policy mechanism to determine if the link connected neighbor is in policy. If it is not, the bidirectional link is considered “misconnected”. Note: This policy mechanism allows the links to be validated before being taken into service. The policy may be seeded by the Planning Tool but may also be updated by the user. The policy mechanism shall operate on a minimum of the following information: <nearEndPoint, farEndPoint>
ND-4.1	Req	The DA shall place misconnected links operationally out-of-service.

6.2 Multiple Discovery Agent cases

The requirements in this section apply to use cases where multiple discovery agents are involved in discovering links. When a common DA supports both ends of a discovered trail, the LADJ and TCE phases occur completely within the DA. Therefore a protocol is not required.

6.2.1 Discovery Trigger Response

ND-5.1	Req	The DA shall support the Layer Adjacency Discovery mechanism as described in G.7714.1.
ND-5.1.2	Req	The DA shall periodically notify the far DA of the far termination point identity observed in an in-band channel at a local termination point, and the identity of the local termination point where it was observed.
ND-5.1.2.1	Req	The DA shall send the periodic notification using an in-band mechanism where possible. Note: An in-band mechanism is available for OTN and Ethernet physical ports. It is not available for SONET/SDH physical ports.
ND-5.1.2.2	Req	The DA shall send the periodic notification using an out-of-band mechanism.
ND-5.1.2.3	Req	The DA shall stop sending periodic notification when a link fails, when TCE has been initiated on a link, when Neighbor Discovery has been disabled on a link, or when the Discovery Agent has been disabled.

6.2.2 TCE Requirements

Req#	Status	Description

Req#	Status	Description
ND-6	Req	The DA shall support the Transport Capability Exchange (TCE) mechanism described in G.7714.1

6.2.3 TCE Transport Protocol Requirements

Req#	Status	Description
ND-6.1	Req	The DA shall utilize a protocol compliant with G.7714.1 for TCE.

6.2.4 TCE Negotiation Requirements

Req#	Status	Description
ND-6.2	Req	The DA shall support an extensible multi-phase exchange of Capability information, including Security method negotiation, adjacency maintenance, and Transport Capability

6.2.4.1 TCE Security Method Negotiation Requirements

Req#	Status	Description
ND-6.2.1	Req	The DA shall perform negotiation of security method.
ND-6.2.1.1	Req	The DA shall support negotiation of a link without Security Exchange.
ND-6.2.1.2	Req	The DA shall support Security Exchange using clear-text username and passwords.

6.2.4.2 TCE Adjacency Maintenance

Req#	Status	Description
ND-6.2.2	Req	The DA shall perform DA adjacency maintenance.
ND-6.2.2.1	Req	The DA shall send Echo Response messages for each Echo Request message received, copying the payload from the Echo Request message to the Echo Response sent.
ND-6.2.2.2	Req	The DA shall periodically send Echo Request messages to validate the DA Adjacency.
ND-6.2.2.3	Req	The DA shall place a unique sequence number in the payload of the Echo Request message. The Sequence number shall start at 0 when the DA adjacency is first established and monotonically increase for a period of 2^32-1. When 2^32-1 is achieved, the sequence number shall automatically wrap-around to 0.

Req#	Status	Description
ND-6.2.2.4	Req	The DA shall support configuration in seconds the time period between Echo messages transmitted per termination point. The provisioning range shall be 1-30 seconds. The default time period shall be 15 seconds.
ND-6.2.2.5	Req	The DA shall verify an Echo Response message is received for each Echo Request sent to determine the state of the DA Adjacency.
ND-6.2.2.6	Req	The DA shall support configuration of the number of consecutive missing Echo Response messages before declaring a Loss of DA Adjacency condition per termination point. The provisioning range shall be 1-30 messages. The default shall be 3 messages.
ND-6.2.2.7	Req	The DA shall clear a Loss of DA Adjacency condition upon receipt of an Echo Response on a DA Adjacency.
ND-6.2.2.8	Req	The DA shall utilize a Session Identifier to identify when a DA does not have the previously negotiated configuration after recovering from an SCN failure.
NS-6.2.2.8.1	Req	The DA shall utilize a 32-bit unsigned integer for the Session Identifier.
ND-6.2.2.8.2	Req	The DA shall create a session identifier using a random number when Neighbor Discovery is enabled on an endpoint or on DA restart when the persisted Discovery information for an endpoint with Discovery Enabled is not valid.
ND-6.2.2.8.3	Req	The DA shall persist the session identifier sent. This session identifier shall be used when sending capability exchange messages after restart.
ND-6.2.2.8.4	Req	The DA shall persist the session identifier received.
ND-6.2.2.8.5	Req	The DA shall validate the session identifier received in a TCE message to determine if the same identifier was received previously for the link identified by the local/remote endpoints.

6.2.4.3 TCE Security Negotiation Requirements

Req#	Status	Description
ND-6.2.3	Req	The DA shall exchange security credentials using the negotiated security protocol after completing adjacency negotiation and adjacency maintenance enters the Open state.

Req#	Status	Description
ND-6.2.3.1	Req	The DA shall support configuration of the security credentials to be sent per termination point.
ND-6.2.3.2	Req	The DA shall support use of a RADIUS Server to validate credentials received.
ND-6.2.3.3	Req	The DA shall support a local database to validate credentials received.
ND-6.2.3.4	Req	The DA shall support configuration of credentials into the local database.
ND-6.2.3.5	Req	The DA shall support raising an AuthenticationFailing condition on a termination point when multiple attempts to authenticate fail in a period of time.
ND-6.2.3.5.1	Req	The DA shall support configuration of the number of failed authentication attempts and the period of time to be validated. The default shall be 5 failed authentication attempts within the last 30 seconds.
ND-6.2.3.5.2	Req	The DA shall clear the AuthenticationFailing condition when the termination point successfully authenticates.

6.2.4.4 TCE Dataplane Capability Negotiation Requirements

Req#	Status	Description
ND-6.2.4	Req	The DA shall perform Transport Endpoint Capability (TEC) negotiation after completing security exchange or after TCE adjacency negotiation if no authentication protocol was negotiated by TCE.
ND-6.2.4.1	Req	The DA shall utilize the same messages and state machines of the TCE protocol, but with a different protocol number to perform Transport Endpoint Capability negotiation.
ND-6.2.4.2	Req	The DA shall utilize the G.7714 state machine for the TEC protocol. Note: This will allow the TCE process to renegotiate the capabilities of an active link.
ND-6.2.4.3	Req	The DA shall describe the layers supporting switching functions and applications.
ND-6.2.4.4	Req	The DA shall utilize a TEC capability option TLV to notify the peer endpoint of each stack of potential layer termination and adaptation functions supported by a link endpoint to access a switching function or application.

Req#	Status	Description
ND-6.2.4.5	Req	The DA shall place one or more stack of potential layer termination and adaptation functions into each TEC capability option TLV.
ND-6.2.4.6	Req	The DA shall always use the layer of the discovered link for the server layer described in a TEC capability option TLV.
ND-4.2.4.7	Req	The DA shall place any intermediate layers along with the client layer supporting switching or an application in Client Layer SubTLV.
ND-4.2.4.8	Req	The DA shall ACKnowledge all TEC Capability option TLVs received.
ND-4.2.4.10	Req	The DA shall use the TEC information learned from a peer entity to update the list of potential link connections supported on a local termination point. Note: The list of potential link connections supported is the intersection of the Transport Endpoint Capabilities supported by each link end. This list is provided to routing for advertisement purposes and to Link Resource Management for bookkeeping purposes.

6.3 Management System Information Exchange

The requirements in this section apply to use cases where the discovery agents support the exchange of management information for the link end. This is performed by an additional TCE negotiation.

6.3.1 TCE Management Plane Negotiation Requirements

Req#	Status	Description
ND-4.2.5	Req	The DA shall perform Management Plane configuration negotiation after TEC negotiation completes and TEC enters the Open state.
ND-4.2.5.1	Req	The DA shall utilize the same messages and state machines of the TEC protocol, but with a different protocol number to perform Management Plane configuration negotiation.
ND-6.2.5.2	Req	The DA shall utilize the G.7714 state machine for the Management Plane Negotiation protocol. Note: This will allow the TCE process to renegotiate the capabilities of an active link.
ND-6.2.5.3	Req	The DA shall exchange Management Domain Name, Node Name and Link End Name in Management Plane configuration messages.

Req#	Status	Description
ND-6.2.5.3.1	Req	The DA shall support variable length Management Domain Names up to 32 printable ASCII characters in length.
ND-6.2.5.3.2	Req	The DA shall support variable length Node Names up to 32 printable ASCII characters in length.
ND-6.2.5.3.3	Req	The DA shall support variable length Link End Names up to 32 printable ASCII characters in length.
ND-6.2.5.4	Req	The DA shall configure the Management Domain Name sent on the DA entity. Note: The Management Domain Name is a display name. The Name is validated as printable ASCII characters and for length only.
ND-6.2.5.5	Req	The DA shall use the NEName/SystemID/TID for the Node Name sent.
ND-6.2.5.6	Req	The DA shall use the AID of the termination point for the Link End Name sent.
ND-6.2.5.7	Req	The DA shall support retrieval of the Management Domain Name, Node Name and Link End Name received from a peer.

6.4 Control Plane Information Exchange

The requirements in this section apply to use cases where the discovery agents support the exchange of control plane information for the link end. This is performed by an additional TCE negotiation.

6.4.1 TCE Control Plane Hierarchy Negotiation

Req#	Status	Description
ND-6.2.6	Req	The DA shall perform negotiation of control plane area hierarchy after TEC negotiation completes and TEC enters the Open state.
ND-6.2.6.1	Req	The DA shall utilize the same messages and state machines as the TEC protocol, but with a different protocol number to perform Area Hierarchy negotiation.
ND-6.2.6.2	Req	The DA shall utilize the G.7714 state machine for the Control Plane Hierarchy Negotiation protocol. Note: This will allow the TCE process to renegotiate the capabilities of an active link.
ND-6.2.6.3	Req	The DA shall exchange the area hierarchy as an ordered list of AreaIDs from Root to Leaf.

Req#	Status	Description
ND-6.2.6.4	Req	The DA shall allow for single-ended UNI configuration by NAKing an empty Area Hierarchy TLV (i.e. Length = 0), providing an Area Hierarchy TLV containing the lowest AreaID of the local node.
ND-6.2.6.5	Req	The DA shall ACKnowledge any non-empty Area Hierarchy TLV received with data that validated correctly.
ND-6.2.6.5.1	Req	The DA shall NAK any Area Hierarchy TLV that does not contain at least one AreaID in common with the local NE.
ND-6.2.6.5.2	Req	The DA shall NAK any Area Hierarchy TLV that has two or more AreaIDs in common with the local NE, but in opposite hierarchy order. Example: If the local system has a hierarchy of three areas (A, B and C) in order ABC, but the received order is CBA, the Area Hierarchy would be NAKed.
ND-6.2.6.7	Req	The DA shall determine the area of the discovered link by comparing the received Area Hierarchy with the Area Hierarchy of the local NE.
ND-6.2.6.7.1	Req	The DA shall declare the area of the link to be the lowest AreaID in common.
ND-6.2.6.7.2	Req	The DA shall raise a NoCommonAreaID condition when no AreaIDs are in common.
ND-6.2.6.8	Req	The DA shall not enter the Open state for Hierarchy Exchange when no AreaIDs are in common.

6.4.2 TCE Control Plane Configuration Negotiation Requirements

Req#	Status	Description
ND-6.2.7	Req	The DA shall perform negotiation of Control Plane Configuration after Area Hierarchy negotiation completes.
ND-6.2.7.1	Req	The DA shall utilize the same messages and state machines of the TEC protocol, but with a different protocol number to perform Control Plane Configuration negotiation.
ND-6.2.7.2	Req	The DA shall utilize the G.7714 state machine for the CP Configuration Negotiation protocol . Note: This will allow the TCE process to renegotiate the capabilities of an active link.
ND-6.2.7.3	Req	The DA shall exchange AreaID, NodeID and ifIndex, using a CP Endpoint Address TLV.

Req#	Status	Description
ND-6.2.7.4	Req	The DA shall exchange Signaling PC ID and Signaling PC SCN Address, using a Signaling PC TLV.
ND-6.2.7.4.1	Req	The DA shall indicate the Signaling Protocol Type accommodating the following values: IETF GMPLS OIF UNI 1.0 OIF UNI 1.0r2 OIF UNI 2.0 OIF UNI 2.0r2 OIF E-NNI 1.0 OIF E-NNI 2.0
ND-6.2.7.4.2	Req	The DA shall exchange the SC PC ID when required by the Signaling Protocol.
ND-6.2.7.4.3	Req	The DA shall send a SC PC ID of 0x00000000 when not required by the Signaling Protocol.
ND-6.2.7.4.4	Req	The DA shall send the Signaling PC SCN address with Format ID and Data formats defined for the DT message.
ND-6.2.7.4.5	Req	The DA shall support the DA DCN Address format to carry Signaling PC SCN information.
ND-6.2.7.4.6	Req	The DA shall support the DA DCN Name format to carry Signaling PC SCN information.
ND-6.2.7.4.7	Req	The DA shall support the TCP ID Address format to carry Signaling PC SCN information.
ND-6.2.7.4.8	Req	The DA shall NAK a received Signaling PC if the SC Proto Type field contains an SC Protocol Type inconsistent with the signaling protocol configured for the termination point.
ND-6.2.7.4.9	Req	The DA shall raise a SignalingProtocolMismatch condition when the SC Protocol Type received is inconsistent with the signaling protocol configured for the termination point.
ND-6.2.7.4.10	Req	The DA shall clear a SignalingProtocolMismatch condition when the SC Protocol Type received is consistent with the signaling protocol configured for the termination point.
ND-6.2.7.5	Req	The DA shall optionally exchange Routing Link Configuration, including: Cost/Metric, Resource Class, and SRG.
ND-6.2.7.5.4	Req	The DA shall allow for single-ended UNI configuration by NAKing a Metric/Cost, Resource Class or SRG TLV. The NAK shall contain a suggested Metric, Resource Class and/or SRG TLV.

Req#	Status	Description
ND-6.2.7.5.5	Req	The DA shall not exchange a Cost/Metric TLV when a Link is configured with Routing disabled.

7 Discovery Protocol

The requirements above specify a number of protocols to support Neighbor discovery. They are described in the following sections.
Some of the use cases do not require every protocol. The applicability of a protocol to a use case is described in each section.

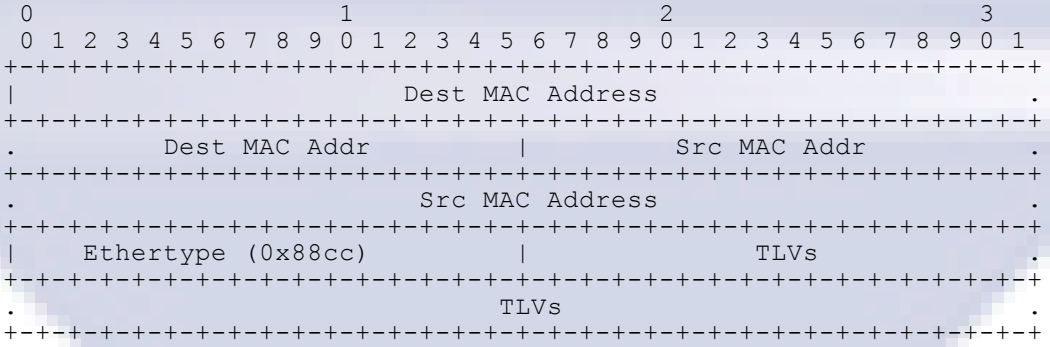
7.1 Discovery Trigger

Discovery Trigger protocols have been defined for transport technologies. The specific format and parameter information are dependent on the technology in use. The following sections provide specific definitions for IEEE (Ethernet) and ITU (SONET/SDH and OTN) defined technologies.
Support for the Discovery Trigger is required for all Neighbor Discovery use cases.

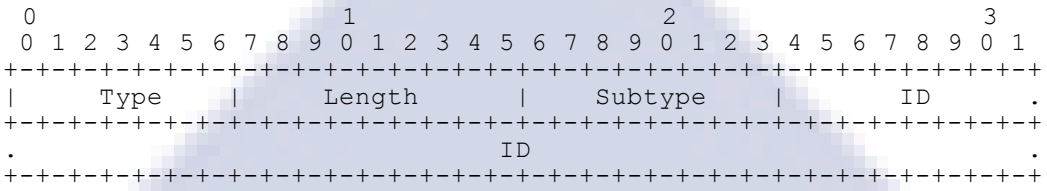
7.1.1 Ethernet

The IEEE has defined the Link Layer Discovery Protocol (IEEE 802.1AB) for use in Ethernet layer networks. This protocol uses a non-forwarded multicast address to announce the identity of a system connected to a link or a repeater segment. Since LLDP is not forwarded, a separate protocol is required to exchange topology information across all NEs in a network.

The LLDP protocol announces a string of TLVs to convey identity information. Two TLVs are mandatory in an announcement – the Chassis TLV and the Port TLV. Other TLVs may be included to announce capability information or vendor specific data. The format of an LLDP PDU is as follows:



The Chassis TLV and Port TLV share a common TLV format, as follows:



The Chassis TLV and Port TLV support the use of identifiers from a number of different namespaces. The namespace is identified by the subtype field. While the format of the TLV is the same, the definition of subtype is different.

In G.7714.1 (2015), the ITU-T has specified the following Chassis ID TLV subtype definition for use as a discovery trigger on Ethernet links:

Subtype	ID Scope	Reference
4	Global	MAC Address (IEEE 802-2001)

In G.7714.1 (2015), the ITU-T has specified the following Port ID TLV subtype definition for use as a discovery trigger on Ethernet links:

Subtype	ID Scope	Reference
7	Chassis	Local Port Identifier (IEEE 802-1AB) – SMIv2 ifIndex

While the LLDP specification meets the requirements for Discovery Trigger, it does not meet the requirements for Discovery Response and TCE. Additional protocols are necessary to meet these requirements.

7.1.2 SONET/SDH and OTN

The ITU has defined Link Discovery (G.7714, G.7714.1) for use in SONET/SDH and OTN networks. This standard uses a trail's overhead to announce the system's identity to neighboring NEs.

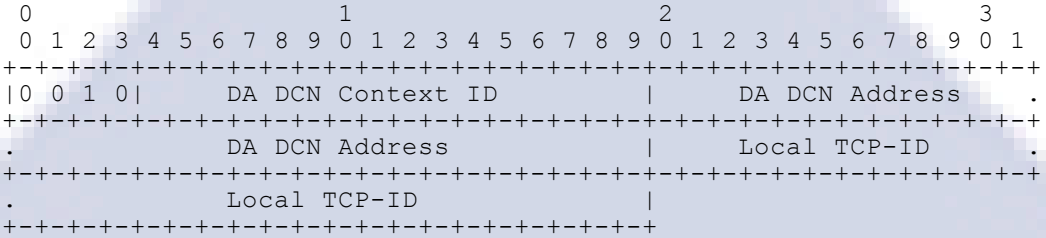
The G.7714.1 specification defines discovery trigger messages so the port identity fits into an ASCII, repeating short message (under 16 bytes) channel. A number of different identifier formats are supported depending on the use case (e.g. management vs control plane use) and DCN environments.

The message formats are defined as binary messages to make the most efficient use of the trail overhead. However the overhead channels are limited to using ASCII printable characters. The binary message is converted into ASCII printable characters using base64 encoding as defined in RFC2045. This encoding expands the message by a factor of 6/8 (i.e. an 24bit message will require 4 characters, or 32 bits). This limits a 14 byte ASCII string to 84 bits.

The use cases, formats used and data carried are as follows:

7.1.2.1 IPv4 DCN

The OIF Neighbor Discovery Implementation Agreement uses the DA DCN Format in IPv4 networks. This format matches the control plane deployment use case, with a network-wide DCN IP Address on each NE. The format of this message is as follows:

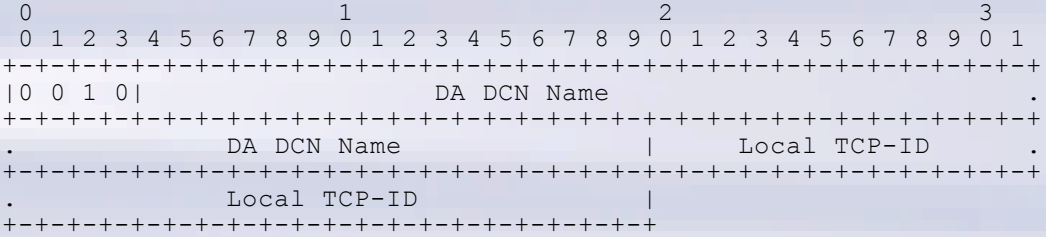


For the OIF Neighbor Discovery Implementation Agreement:

1) the DA DCN Context ID is a provisioned field with a default of 0x0000,
2) the DA DCN Address carries the IPv4 address used for discovery communications on the local NE, and
3) the Local TCP-ID carries the ifIndex associated with the trail termination where the discovery message is being announced.

7.1.2.2 IPv6 DCN

The OIF Neighbor Discovery Implementation Agreement uses the DA Name Format in IPv6 networks. The format of this message is as follows:



For the OIF Neighbor Discovery Implementation Agreement:

1) the DA DCN Name carries a name for the Discovery Agent, and
2) the Local TCP-ID carries the ifIndex associated with the trail termination where the discovery message is being announced.

While the DA DCN format best matches the control plane deployment use case, an IPv6 address cannot fit within the constrained message. As a result, a nameserver is needed to translate the DA DCN Name into the DA's IPv6 address. The protocols used to access the nameserver are not covered in this implementation agreement.

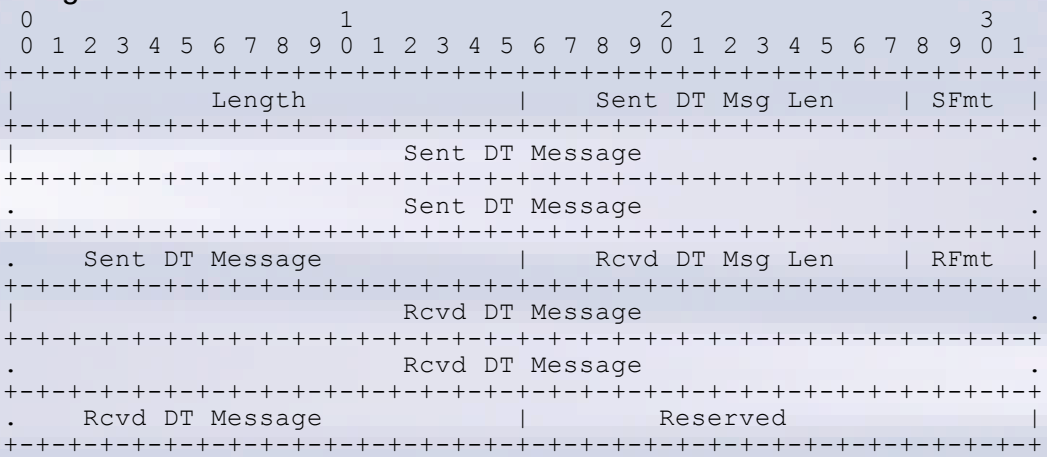
7.2 Layer Adjacency Discovery - Discovery Response

Layer Adjacency Discovery is made possible by the receiving Discovery Agent sending a Discovery Response to the sending Discovery Agent identified in a

Discovery Trigger message. This response contains the received Discovery Trigger message along with the Discovery Trigger message being sent on the port where the message was received.

Support for the Discovery Response is only required for Neighbor Discovery use cases where separate discovery agents are responsible for each link end. It is not required when a common discovery agent handles both link ends.Unlike the Discovery Trigger message, the Discovery Response does not need to be carried in-band. This removes a number of restrictions (e.g. size) to the message format.

The Discovery Response message is defined as a UDP message with the following format:



The Discovery Response message is sent to UDP port 7714.

7.3 Transport Capability Exchange

The Transport Capability Exchange protocol is used to negotiate configuration information for the link. Negotiation is necessary as the specific configuration used is dependent on the relationship between the two trail endpoints.

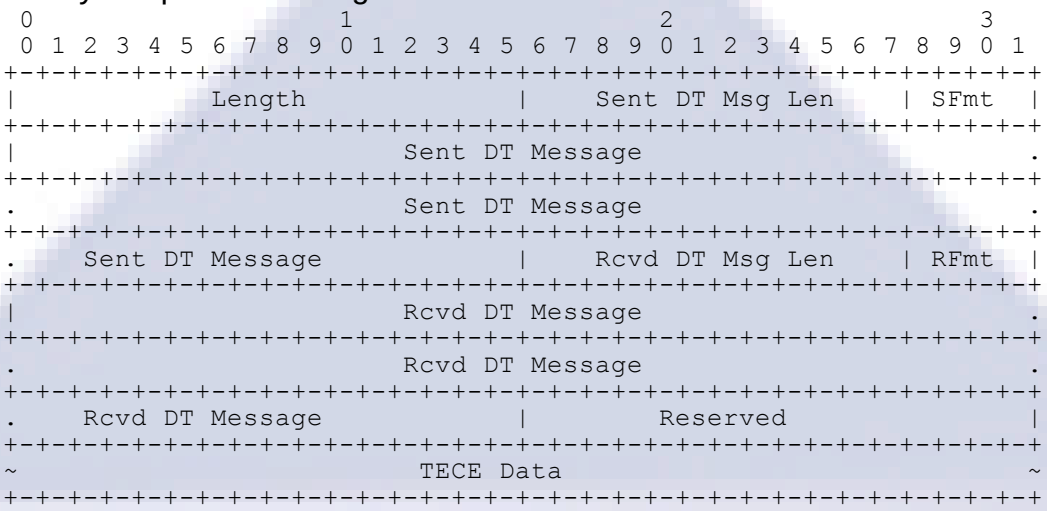
Support for Transport Capability Exchange is only required for Neighbor Discovery use cases where separate discovery agents are responsible for each link end.

ITU G.7714 defines a state machine to perform configuration negotiation, but does not provide a protocol to carry configuration messages. Appendix II of ITU G.7714 identifies the state machine is derived from the IETF's Point-to-Point Protocol (PPP) Link Control Protocol (LCP) state machine. It is therefore possible to use PPP to perform the configuration negotiation.

Note: The IETF has another protocol for exchanging link information: the Link Management Protocol. Unlike PPP, LMP does not support negotiation. For this reason, this implementation agreement specifies the use of PPP.

Using PPP requires a message format that encapsulates TECE messages and carries them over the DCN network. Since this message is being carried out-of-band, the message needs to include identifiers for the ends of the link being

described. The Capability Exchange message is defined as an extension of the Discovery Response message. The format is as follows:



The Capability Exchange message uses the same UDP port as the Discovery Response message.

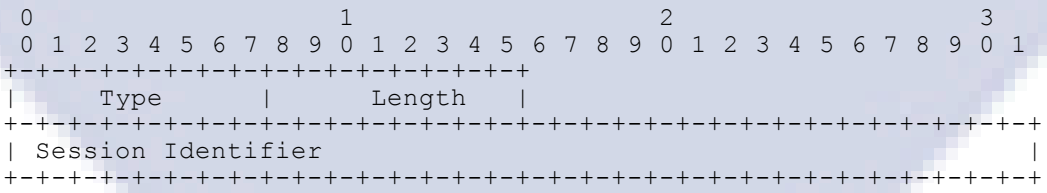
Within the PPP protocol, separate negotiation sessions are used for exchanging different configuration data. The following sections describe the negotiation options exchanged.

7.3.1 Security Exchange

The PPP protocol suite includes authentication exchange as a part of link negotiation. The OIF Neighbor Discovery Implementation Agreement maintains the use of these protocols, including authentication protocol negotiation exchange as a part of LCP. In addition to clear-text authentication, it is recommended that implementations also support the Challenge Authentication Protocol (CHAP) as defined in RFC 1994.

7.3.2 Restart Session Identifier

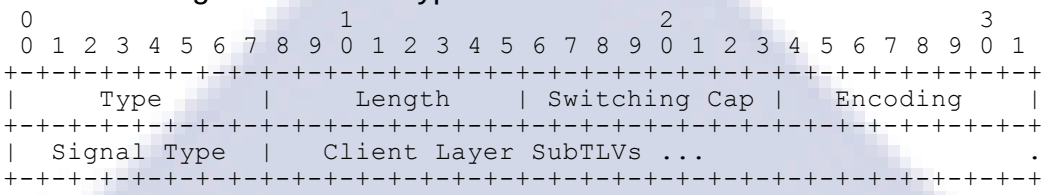
The OIF Neighbor Discovery Implementation Agreement uses an LCP option to exchange a restart session identifier. This option uses the following format and a type of 0x81:



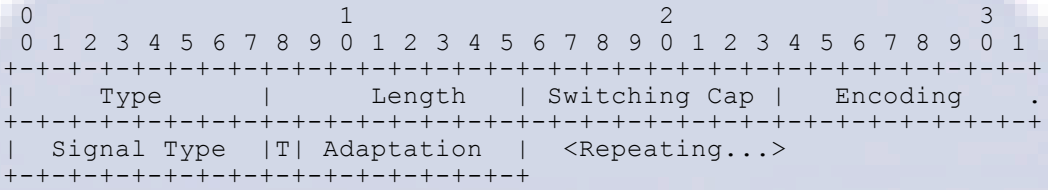
7.3.3 Transport Endpoint Capability Information

Transport Endpoint Capability is represented using the same {Switching Capability, Encoding Type, Signal Type} tuple and Adaptation Type information

defined in the layer stack used by the OIF's Multilayer Amendment. The option uses the following format and a type of 0x01 :



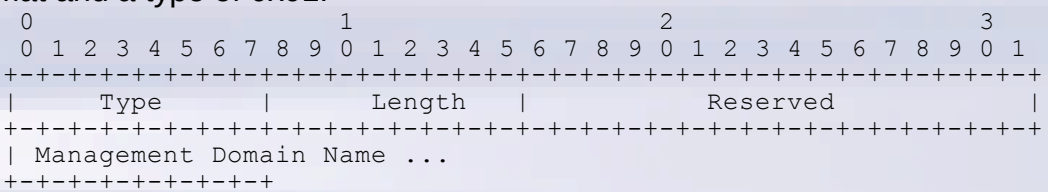
The Client Layer TLVs carried within the option are as follows, with a type of 0x02:



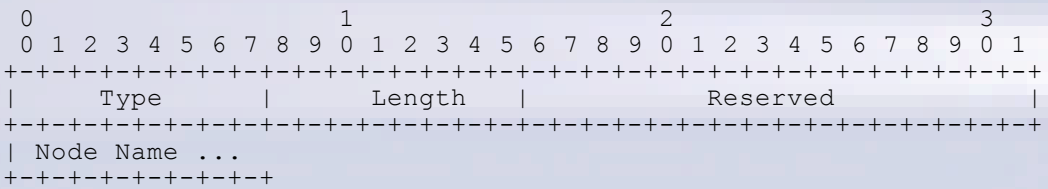
The interpretation of fields is as defined in OIF-ENNI-OSPF-02.1.

7.3.4 Management Domain Names

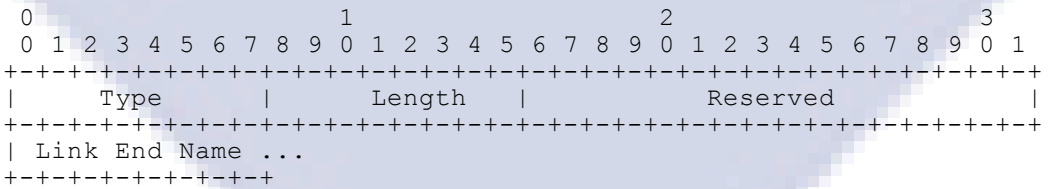
Support for exchanging Management domain names is required for all Neighbor Discovery use cases where separate discovery agents are responsible for each link end and automated management system information sharing is enabled. The Management Domain Name is exchanged by a TLV with the following format and a type of 0x01:



The Node Name is exchanged by a TLV with the following format and a type of 0x02:



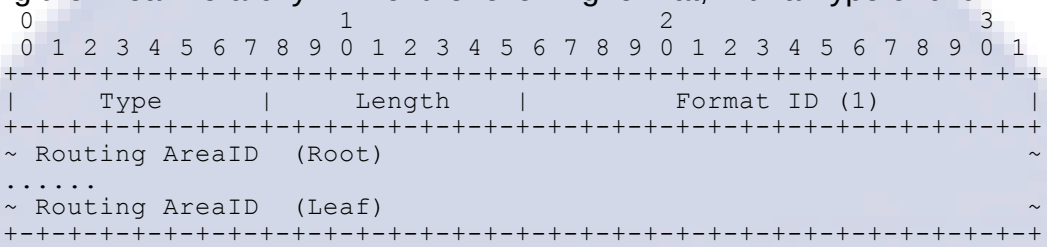
The Link End Name is exchanged by a TLV with the following format and a type of 0x03:



7.3.5 Control Plane Area Hierarchy

Support for exchanging Control Plane Area Hierarchy is required for all Neighbor Discovery use cases where separate discovery agents are responsible for each link end, the control plane is allowed to control the link and automated configuration of control plane signaling characteristics is enabled.

The area hierarchy is exchanged as an ordered list of AreaIDs from Root to Leaf using the Area Hierarchy TLV of the following format, with a Type of 0x01:

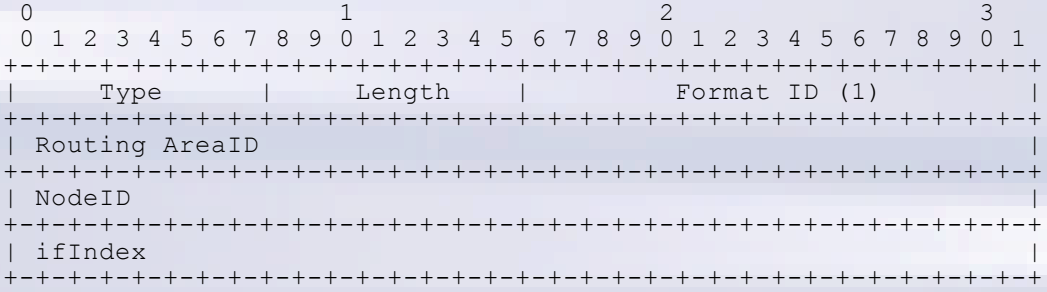


7.3.6 Control Plane Signaling

Support for exchanging Control Plane Signaling configuration is required for all Neighbor Discovery use cases where separate discovery agents are responsible for each link end, the control plane is allowed to control the link and automated configuration of control plane signaling characteristics is enabled.

7.3.6.1 Link End Name

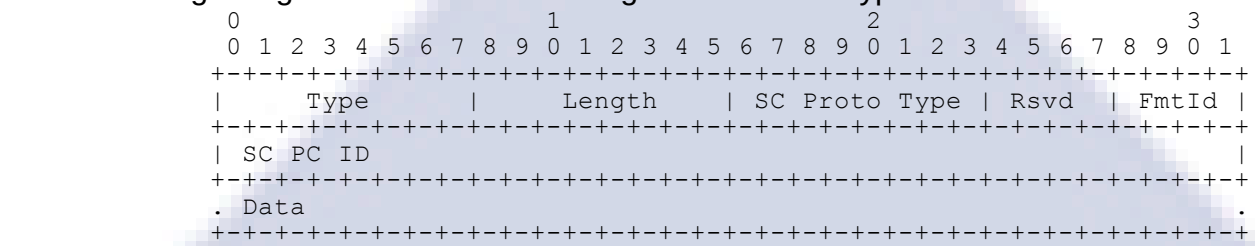
The DA shall exchange AreaID, NodeID and ifIndex, using a CP Endpoint Address TLV of the following format with a Type of 0x01:



Note: The format ID specifies the overall format of the TLV (e.g. 32-bit Routing AreaID, 32-bit NodeID and 32-bit ifIndex). Only one format may be used on a link. Only one format is defined at this time, with additional formats defined in the future as needed.

7.3.6.2 CP Signaling Adjacency

The DA shall exchange Signaling PC ID and Signaling PC SCN Address, using a Signaling PC TLV of the following format with a Type of 0x02:



The Signaling Protocol Type is indicated using the SC Protocol Type field. The values used in this field are as follows:

0x00 = IETF GMPLS
0x10 = OIF UNI 1.0
0x11 = OIF UNI 1.0r2
0x12 = OIF UNI 2.0
0x13 = OIF UNI 2.0r2
0x20 = OIF E-NNI 1.0
0x21 = OIF E-NNI 2.0

8 References

8.1 ITU-T

[G.707] ITU-T Recommendation G.707/Y.1322 (01/2007), Network Node Interface for the Synchronous Digital Hierarchy (SDH)

[G.709Ed2] ITU-T Recommendation G.709/Y1331 (03/2003), *Interfaces for the Optical Transport Network (OTN)*

[G.709Ed4] ITU-T Recommendation G.709/Y.1331 (02/2012), *Interfaces for the Optical Transport Network (OTN)*

[G.783] ITU-T Recommendation G.783 (03/2006), Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks

[G.784] ITU-T Recommendation G.784 (03/2008), Management aspects of synchronous digital hierarchy (SDH) transport network elements

[G.800] ITU-T Recommendation G.800 (02/2012), Unified functional architecture of transport networks

[G.805] ITU-T Recommendation G.805 (03/2000), Generic Functional Architecture of Transport Networks

[G.874] ITU-T Recommendation G.874 (08/2013), Management aspects of optical transport network elements

[G.7712] ITU-T Recommendation G.7712/Y.1703 (09/2010), Architecture and Specification of Data Communication Network

[G.7714] ITU-T Recommendation G.7714/Y.1705 (08/2005), Generalized Automatic Discovery for Transport Entities

[G.7714.1] ITU-T Recommendation G.7714.1/Y.1705.1 (01/2015), Architecture and Requirements for Routing in the Automatically Switched Optical Network

[G.8080] ITU-T Recommendation G.8080/Y.1304 (02/2012), Architecture for the Automatically Switched Optical Network (ASON)

8.2 OIF

[OIF-UNI-02.0] OIF Implementation Agreement, “User Network Interface (UNI) 2.0 Signaling Specification Common Part,” OIF-UNI-02.0-Common, February 2008.

[OIF-E-NNI-SIG-02.0] OIF Implementation Agreement, “OIF E-NNI Signaling Specification”, OIF-E-NNI-Sig-02.0, April 2009

[OIF-E-NNI-OSPF-01.0] OIF Implementation Agreement, “External Network-Network Interface (E-NNI) OSPF-based Routing - 1.0 (Intra-Carrier) Implementation Agreement”, OIF-ENNI-OSPF-01.0, January 2007

[OIF-E-NNI-OSPF-02.0] OIF Implementation Agreement, “External Network-Network Interface (E-NNI) OSPF-based Routing - 2.0 (Intra-Carrier) Implementation Agreement”, OIF-ENNI-OSPF-02.0, July 2011

[OIF-ENNI-ML-AM-01.0] OIF Implementation Agreement, “Multi-Layer Amendment to E-NNI 2.0 – Common Part”, OIF-ENNI-ML-AM-01.0, April 2013

[OIF-ENNI-OSPF-02.1] OIF Implementation Agreement, “OIF Multilayer Amendment to E-NNI 2.0 OSPFv2-based Routing”, OIF-ENNI-OSPF-02.1, April 2013

8.3 ANSI

[T1.105] ANSI T1.105: SONET Basic Description including Multiplex Structure, Rates and Formats

8.4 IEEE

[IEEE802.1AB] IEEE 802.1AB-2009: IEEE Standard for Station and Media Access Control Connectivity Discovery

9 Appendix A: Discovery Trigger and ITU-T recommendations

The requirements tagged with identifiers ND-3.3.6.[1-5] discuss manipulation of a number of dataplane configuration parameters defined by ITU-T in G.784 and G.874. The following table describes the mapping into specific MI signals.

ND-3.3.6.[1-5] signal	G.784 MI signal	G.874 MI signal
Expected_TTI	* _MI_ExTI	* _MI_ExSAPI * _MI_ExDAPI
TTI_mismatch-AIS-insertion	* _MI_TIMAISdis	* _MI_TIMActDis
Received_TTI	* _MI_AcTI	* _MI_GetAcTI

10 Appendix B: List of companies belonging to OIF when document is approved

Acacia Communications
ADVA Optical Networking
Alcatel-Lucent
Altera
AMCC
Amphenol Corp.
Analog Devices
Anritsu
Avago Technologies Inc.
Broadcom
Brocade
BRPhotonics
BTI Systems
China Telecom
Ciena Corporation
Cisco Systems
ClariPhy Communications
Coriant R&G GmbH
CPqD
EMC Corporation
Emcore
Ericsson
ETRI
FCI USA LLC
Fiberhome Technologies Group
Finisar Corporation
Fujikura
Fujitsu
Furukawa Electric Japan
Google
Hitachi
Huawei Technologies Co., Ltd.
IBM Corporation
Infinera
Inphi
Intel
Ixia

JDSU
Juniper Networks
Kaiam
Kandou
KDDI R&D Laboratories
Keysight Technologies, Inc.
Luxtera
M/A-COM Technology Solutions
Mellanox Technologies
Microsemi Inc.
Microsoft Corporation
Mitsubishi Electric Corporation
Molex
MoSys, Inc.
MultiPhy Ltd
NEC
NeoPhotonics
NTT Corporation
O-Net Communications (HK) Limited
Oclaro
Orange
PETRA
Picometrix
PMC Sierra
QLogic Corporation
Qorvo
Ranovus
Rockley Photonics
Samtec Inc.
Semtech
Socionext Inc.
Spirent Communications
Sumitomo Electric Industries
Sumitomo Osaka Cement
TE Connectivity
Tektronix
TELUS Communications, Inc.
TeraXion
Texas Instruments
Time Warner Cable

US Conec
Verizon
Xilinx
Yamaichi Electronics Ltd.
ZTE Corporation