International Telecommunication Union

ITU-T G.7714.1/Y.1705.1

(08/2017)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Data over Transport – Generic aspects – Transport network control aspects

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Internet protocol aspects – Operation, administration and maintenance

Protocol for automatic discovery in transport networks

Recommendation ITU-T G.7714.1/Y.1705.1

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
General	G.7000–G.7099
Transport network control aspects	**G.7700–G.7799**
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T G.7714.1/Y.1705.1

Protocol for automatic discovery in transport networks

Summary

Recommendation ITU-T G.7714.1/Y.1705.1 describes the methods, procedures and transport plane mechanisms for discovering layer adjacency according to the requirements of Recommendation ITU-T G.7714/Y.1705. Layer adjacency discovery describes the process of discovering link connection end-point relationships and verifying their connectivity. Two alternative methods are described: one using a test set in the client layer, the other using in-band overhead in the server layer. Additional actions that may be required for obtaining physical media adjacency discovery, transport entity capability exchange, etc., will be addressed in future Recommendations.

Recommendation ITU-T G.7714.1/Y.1705.1 (2017) includes:

1) support for transport networks other than synchronous digital hierarchy (SDH) and optical transport network (OTN), specifically Ethernet (ETH); and

2) errata updates.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T G.7714.1/Y.1705.1	2003-04-22	15	11.1002/1000/6291
1.1	ITU-T G.7714.1/Y.1705.1 (2003) Amd. 1	2006-02-17	15	11.1002/1000/8748
2.0	ITU-T G.7714.1/Y.1705.1	2010-09-06	15	11.1002/1000/10896
3.0	ITU-T G.7714.1/Y.1705.1	2015-01-13	15	11.1002/1000/12379
4.0	ITU-T G.7714.1/Y.1705.1	2017-08-13	15	11.1002/1000/13337

Keywords

Auto-discovery, automatic switched optical network, automatic switched transport network, layer adjacency discovery, network resources.

---

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T G.7714.1/Y.1705.1

Protocol for automatic discovery in transport networks

1 Scope

This Recommendation describes the methods, procedures and transport plane mechanisms for discovering layer adjacency according to the requirements of [ITU-T G.7714]. Layer adjacency discovery (LAD) describes the process of discovering link connection (LC) end-point relationships and verifying their connectivity. The term "discovery" is used throughout the Recommendation to refer to both "discovery" and verification. Two alternative methods are described: one using a test set in the client layer; the other using in-band overhead in the server layer. Additional actions that may be required to obtain physical media adjacency discovery, transport entity capability exchange, etc., will be addressed in future Recommendations.

Equipment developed prior to this Recommendation might not interwork with some of the features developed within this Recommendation. Care should be taken where old and new equipment are to interwork.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T G.709] Recommendation ITU-T G.709/Y.1331 (2016) *Interfaces for the optical transport network*.

[ITU-T G.805] Recommendation ITU-T G.805 (2000), *Generic functional architecture of transport networks*.

[ITU-T G.831] Recommendation ITU-T G.831 (2000), *Management capabilities of transport networks based on the synchronous digital hierarchy (SDH)*.

[ITU-T G.7701] Recommendation ITU-T G.7701 (2016), *Common control aspects.*

[ITU-T G.7712] Recommendation ITU-T G.7712/Y.1703 (2010), *Architecture and specification of data communication network*.

[ITU-T G.7714] Recommendation ITU-T G.7714/Y.1705 (2005), *Generalized automatic discovery for transport entities*.

[ITU-T G.8010] Recommendation ITU-T G.8010/Y.1306 (2004), *Architecture of Ethernet layer networks*.

[ITU-T T.50] Recommendation ITU-T T.50 (1992), *International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange*.

[IEEE 802.1AB] IEEE Std 802.1AB-2016, *IEEE Standard for local and metropolitan area networks – Station and media access control connectivity discovery*.

[IEEE 802.3] IEEE Std 802.3-2015, *Standard for Ethernet*.

[IETF RFC 1570] IETF RFC 1570 (1994), *PPP LCP extensions*.

[IETF RFC 1662] IETF RFC 1662 (1994), *PPP in HDLC-like framing*.

[IETF RFC 2045] IETF RFC 2045 (1996), *Multipurpose internet mail extensions (MIME) – Part One: Format of Internet message bodies*.

[IETF RFC 3518] IETF RFC 3518 (2003), *Point-to-point protocol (PPP) bridging control protocol (BCP).*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 [ITU-T G.805]:

a) adaptation

b) link

c) link connection

d) trail

3.1.2 [ITU-T G.7701]:

a) discovery agent

b) policy

c) termination adaptation performer (TAP)

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 local TCP-ID: A termination connection point identifier (TCP-ID) that has local significance to the discovery agent transmitting the discovery messages.

3.2.2 local CP-ID: A connection point identifier (CP-ID) that has local significance to the discovery agent transmitting the discovery messages.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ADM Add-Drop Multiplexer

AIS Alarm Indication Signal

AITS Acknowledged Information Transfer Service

API Access Point Identifier

ASON Automatically Switched Optical Network

BCP Bridging Control Protocol

CO Central Office

CP Connection Point

CP-ID Connection Point Identifier

DA-ID Discovery Agent Identifier

DA Discovery Agent

DA ID	Discovery Agent Identifier
DAPI	Destination Access Point Identifier
DCC	Data Communications Channel
DCN	Data Communications Network
DM	Discovery Message
DXC	Digital cross Connect
ECC	Embedded Communication Channel
ETH	Ethernet
ETY	Ethernet PHY
GCC	General Communications Channel
GMPLS	Generalized Multi-Protocol Label Signalling
HDLC	High-level Data Link Control
HOVC	Higher Order Virtual Container
IP	Internet Protocol
IRV	International Reference Version
LAD	Layer Adjacency Discovery
LAPD	Link Access Procedure D-channel
LC	Link Connection
LCP	Link Control Protocol
LL	Link Layer
LLCF	Link Layer Convergence Function
LLDP	Link Layer Discovery Protocol
LOVC	Lower Order Virtual Container
MAC	Media Access Control
MIB	Management Information Base
MPLS	Multi-Protocol Label Signalling
MS	Multiplex Section
NE	Network Element
NMS	Network Management System
OAM	Operations Administration and Maintenance
ODUk	Optical channel Data Unit-k
OTN	Optical Transport Network
OTUk	completely standardized Optical channel Transport Unit-k
PC	Protocol Controller
PDU	Protocol Data Unit
PM	Path Monitoring
PPP	Point-to-Point Protocol

RS Regenerator Section

SAPI Source Access Point Identifier

SDH Synchronous Digital Hierarchy

SM Section Monitoring

SNP Subnetwork Point

STM Synchronous Transport Module

TAP Termination and Adaptation Performer

TCE Transport entity Capability Exchange

TCM Tandem Connection Monitoring

TCP Termination Connection Point

TCP-ID Termination Connection Point Identifier

TIM Trace Identifier Mismatch

TLV Type-Length-Value

TT Trail Termination

TTI Trail Trace Identifier

UDP User Datagram Protocol

UI Unnumbered Information

UITS Unacknowledged Information Transfer Service

WDM Wavelength Division Multiplexing

5 Discovery methodology

The discovery methodology uses the processes specified in clauses 6 to 12 to determine the termination connection point- (TCP)- to-TCP relationship. Once the TCP-to-TCP relationship is determined, the connection point- (CP)- to-CP connectivity relationships are derived using local information. The following two discovery methods are defined.

a) *Link discovery process using server trail overhead method*

In this process, the server layer trail overhead is used to discover the peer TCPs (e.g., TCP_{3S} to TCP_{3R} in Figure 1). The server layer trail overhead is used to carry the discovery message (DM). The CP-to-CP relationships are inferred from the TCP-to-TCP relationships using local knowledge of the configuration of the adaptation function and its relationship with the trail termination (TT) function. This process does not disrupt the client signal being carried by the link.

b) *Link discovery process using client layer payload method*

In this process, a signal is injected into the client layer payload to discover the peer TCPs (e.g., TCP_{1S} to TCP_{1R} in Figure 1). The CP-to-CP relationship is inferred from the local knowledge of the matrix connection that was previously set up to connect the test signal to the desired CP (shown in Figure 1). In contrast to the link discovery process using the server trail overhead, this approach may impact the client traffic being carried in the LC.

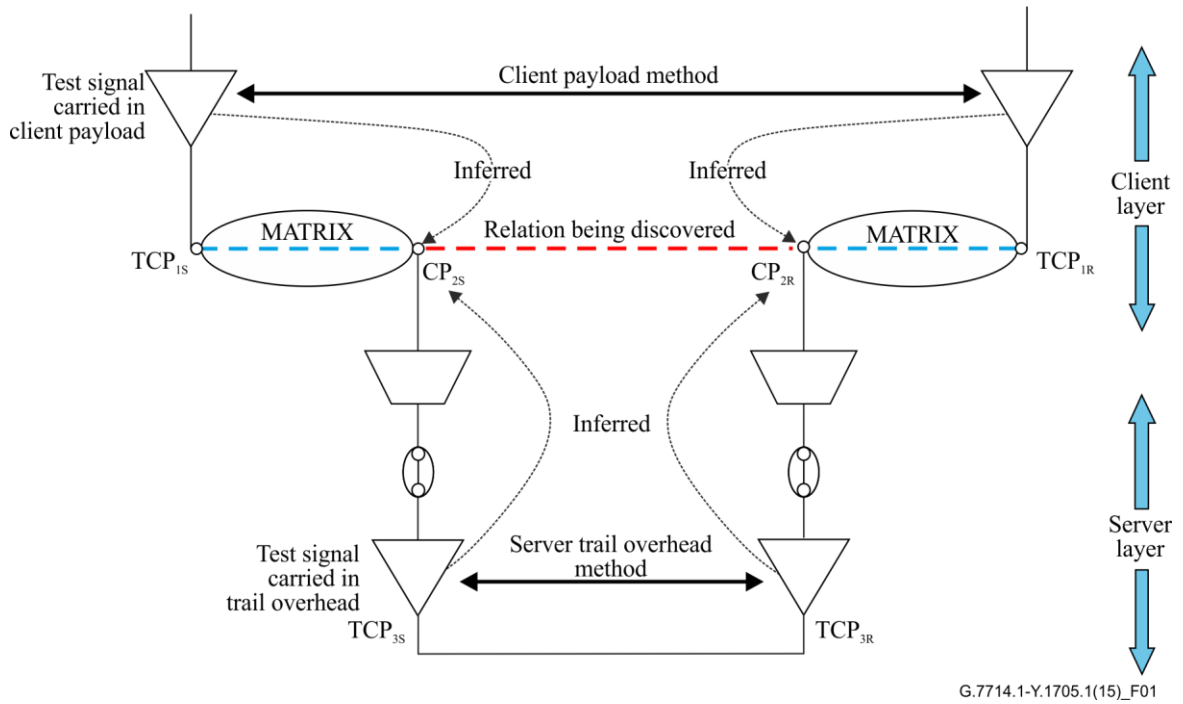

Figure 1 – Entities in the in-service and out-of-service discovery processes

The discovery methodology, namely the information elements, message formats and the transport mechanisms, described in clause 5.1 is identical for both processes.

5.1 CP-CP connectivity relationships determination

The goal of the LAD process is to discover the relationship between connection points CP_{2S} and CP_{2R} as shown in Figure 1. This can be indirectly inferred by means of either:

a) discovering the relationship between TCP_{1S} and TCP_{1R} (using the out-of-service discovery mechanism); or

b) discovering the relationship between TCP_{3S} and TCP_{3R} (using the in-service discovery mechanism).

Once the TCP-to-TCP relationship is determined, CP-to-CP connectivity must be inferred or derived using local information of the CP-TCP bindings. This information (i.e., the name binding between CP_{2S} and $TCP_{1S/3S}$ as well as between CP_{2R} and $TCP_{1R/3R}$) is pre-provisioned and resides in the equipment.

NOTE – A validation method for optical transport network (OTN) name-binding relationships is for further study.

6 Mechanisms for layer adjacency discovery

6.1 Layer adjacency discovery of transport entities

The mechanisms defined to support the LAD process apply on a per layer basis. Within each of the layer networks that support the discovery process, different mechanisms are available. These may reuse the available embedded communication channels (ECCs) for the particular layer. The following mechanisms are applicable to synchronous digital hierarchy (SDH) layer network transport entities:

– RS layer – within the regenerator section (RS) layer, the J0 section trace and section data communications channel (DCC) may be used to support discovery of the RS TCP-to-TCP adjacency;

– MS layer – within the multiplex section (MS) layer, the multiplex section DCC may be used to support discovery of the MS TCP-to-TCP adjacency;

– HOVC layer – within the higher order virtual container (HOVC) layer, the higher order path layer J1 trace may be used to support discovery of the HOVC TCP-to-TCP adjacency;

– LOVC layer – within the lower order virtual container (LOVC) layer, the lower order path layer J2 trace may be used to support discovery of the LOVC TCP-to-TCP adjacency.

The following mechanisms are applicable to the OTN layer network transport entities.

– OTUk layer – within the completely standardized optical channel transport unit-k (OTUk) layer, the section monitoring (SM) byte and the general communications channel- (GCC-) 0 may be used to support discovery of the OTUk adjacency. Specifically, the source access point identifier (SAPI) subfield within the SM is used to carry the DM.

– ODUk layer – within the optical channel data unit-k (ODUk) layer, the path monitoring (PM) byte and the GCC-1 and GCC-2 bytes may be used to support discovery of the ODUk adjacency. Specifically, the SAPI subfield within the PM is used to carry the DM.

– ODUkT layer – within the ODUkT sublayers, the trail trace identifier (TTI) field may be used to support discovery of the ODUk adjacency. By default, the ODU tandem connection monitoring (TCM) sublayer 6 (TCM6) is used for discovery. Specifically, the SAPI subfield within the TTI field is used to carry the DM.

The following mechanisms are applicable to ITU-T G.8010 Ethernet (ETH) layer network transport entities:

– ETY layer – discovery of the Ethernet PHY (ETY) trail is performed using the client payload method. IEEE 802.1AB [link layer discovery protocol (LLDP)] operations administration and maintenance (OAM) messages are inserted at the media access control (MAC)/ETH_A, which are carried across the ETH link. Within the LLDP messages, the chassis ID and port ID type-length-values (TLVs) are used to carry the DM identifiers.

Appendix VI provides clarification of the network scenarios under which the various discovery mechanisms described in this Recommendation may be utilized. This includes guidelines for their usage, as well as potential associated implications.

7 Attributes used in layer adjacency

– **Discovery agent identifier**

The discovery agent identifier (DA ID) must be unique within the context of the link being discovered. Two different representations of the DA ID exist: a discovery agent (DA; also known in [ITU-T G.7714] as a type of control entity) address and a DA name.

– **Discovery agent address**

Two attributes are defined to support the DA address:

– **Data communications network context identifier**

This represents an assigned number (a globally assigned number would be desirable). This attribute may be used in conjunction with the data communications network (DCN) address attribute to guarantee uniqueness for the DA ID. If the sending and receiving discovery agents at each end of the link are within different DCN contexts, but use the same DCN addresses, they may be unable to communicate.

– **DA DCN address**

This represents the address used to identify the discovery agent.

– **Discovery agent name**

This is a name that can be resolved into a DA address. The DA name may be assigned a DCN name.

– **TCP-ID**

The termination connection point identifier (TCP-ID) contains the identifier for the TCP being discovered. This has only local significance within the scope of the DA.

– **DCN Identifier**

The DCN identifier, or DCN ID, is a DCN name or a DCN address. When used for discovery agents, the term DA DCN ID may be used.

8 Layer adjacency based on trail trace string

The trail trace bytes (Jx in SDH or TTI in OTN) provides a mechanism to pass a message that is 16 bytes in length. Each trace byte consists of a message start bit and 7-bits for "payload". The message start bit is set for the first byte in the message and clear for all remaining bytes in the message. The payload of the first trace byte is reserved to carry a 7-bit cyclic redundancy check (CRC) for the message in SDH and is set to all zeroes in OTN. The payload of the second and subsequent bytes is the access point identifier (API), as described in [ITU-T G.831], which specifies two different formats:

a) one-, two- or three-character E.164; and

b) two- or three-character b-ISO 3166 country code, with country-specific extension.

All characters are alphanumeric characters from the ITU-T T.50 7-bit international reference version (IRV) set (with trailing NULLs or SPACEs). As a result, the second byte conforming to the two formats listed above is limited to the following characters:

- A-Z;
- a-z;
- 0-9.

This Recommendation defines a third type of format, which is differentiated from the ITU-T G.831-specified formats by placing a non-numerical and non-alphabetic character in the second byte of the message.[1] The remaining 14 bytes are used to carry the information required by [ITU-T G.7714], namely the DA ID and TCP-ID. These 14 bytes provide 84 bits for the discovery data.

Since the DA ID and TCP-ID are typically numbers, a method for encoding numbers into printable characters is used. Base64 encoding, as defined in [IETF RFC 2045], provides a relatively efficient method to represent 6 bits of information in a printable character, which allows existing provisioning interfaces to be used to provision the DM when required. This yields 3 nibbles or 12 bits for every two printable characters.

Figure 2 shows the overall J0/J1/J2 or SAPI 16-byte format and depicts how the DM is formatted as compared to the ITU-T G.831 API.

[1] See Appendix IV for use cases requiring printable characters.

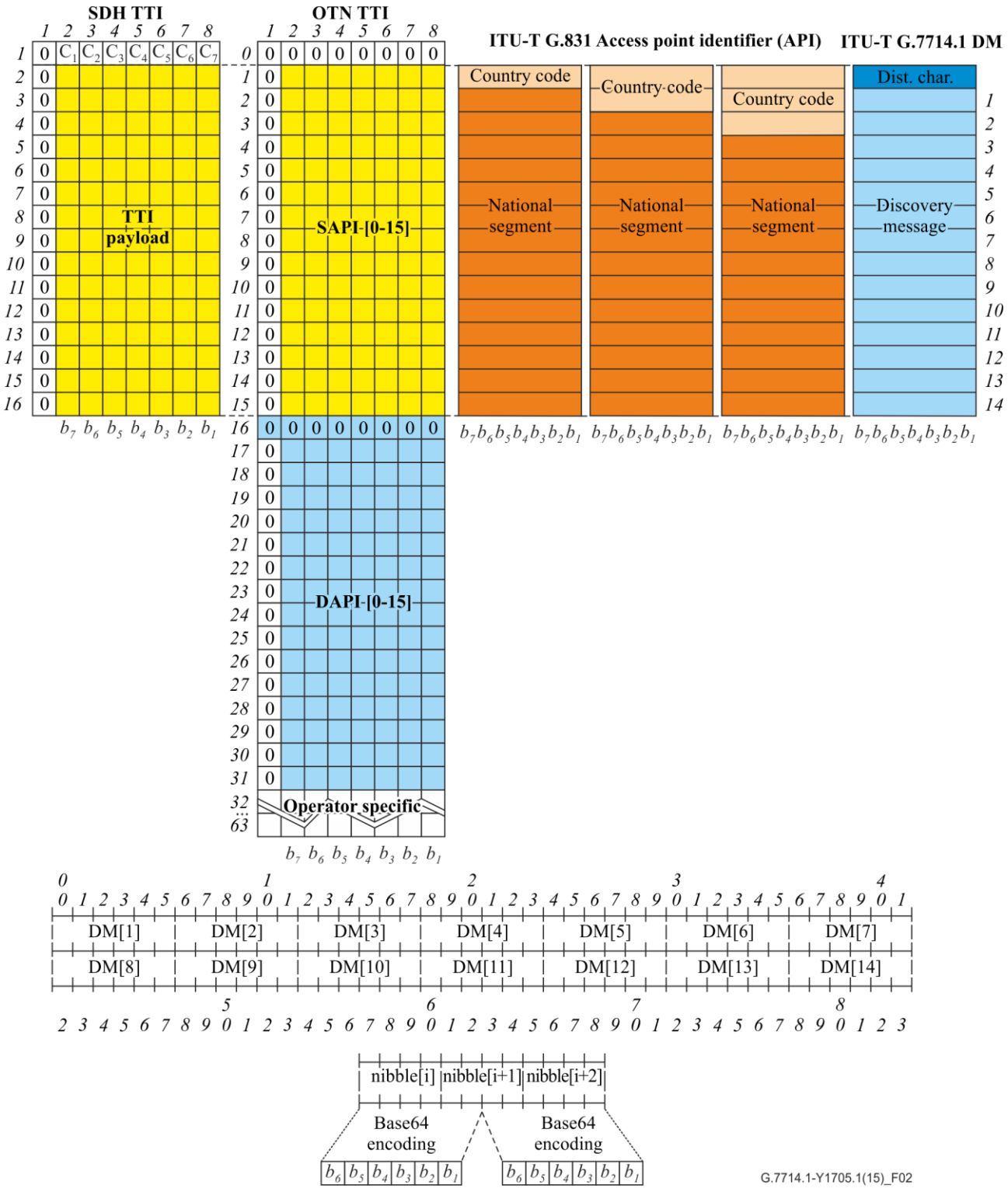

Figure 2 – Auto-discovery message format within the trail trace format

Use of the trail trace bytes for the discovery process does not necessarily mean that an in-service test is being performed. This is a consequence of sharing the trail trace bytes with other functions such as in-service connection monitoring and non-intrusive layer monitoring. If these functions are unable to handle the DM (either due to configuration or software limitation), the functions will need to be disabled while the discovery process is being performed. The interaction between the discovery process and other functions using the trail trace bytes is for further study.

8.1 Discovery message formats

The messages defined in this clause are independent of the mechanism chosen to support them. [ITU-T G.7714] defines the attributes identified through the exchange of DMs as:

– DA ID;

– TCP-ID.

This information can be contained directly in the message or can be derived from the message by an external process, such as a name-server. A number of formats for the DM are therefore necessary.

To facilitate these formats, the general message format shown in Figure 3 is used. This format contains 4 bits of format ID and 80 bits of format specific data.

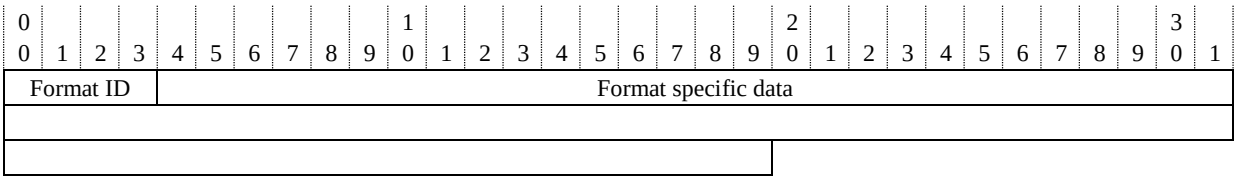

Figure 3 – General discovery message format

This Recommendation defines formats 1, 2, 3 and 4 as defined in clauses 8.1.1 to 8.1.4. Additional formats may be provided in the future. If a DM is received with unknown format IDs, the message should be discarded.

8.1.1 TCP name format

The TCP name format contains a TCP name. The sending and receiving discovery agents are part of a federation that provides a name-server allowing the name to be uniquely resolved into the discovery agent DCN address and TCP-ID. The namespace may be subdivided among different name servers that are responsible for resolving names within the assigned parts of the name space. The format of the name is defined by the context of the name server, and is not specified here.

The sender and receiver are required to have *a priori* knowledge of the common context for the name. The context defines the method to uniquely resolve the name. The method for resolving the received names into the address of the remote discovery agent and the remote TCP-ID is outside the scope of this Recommendation. The address of the name-server that performs the resolution is a well-known attribute that is scoped per trail. This means that the name-server can be different for each trail terminating a DM.

The DM to be used with the TCP-ID name format is shown in Figure 4. This format contains 4 bits of format ID and 80 bits of TCP name.

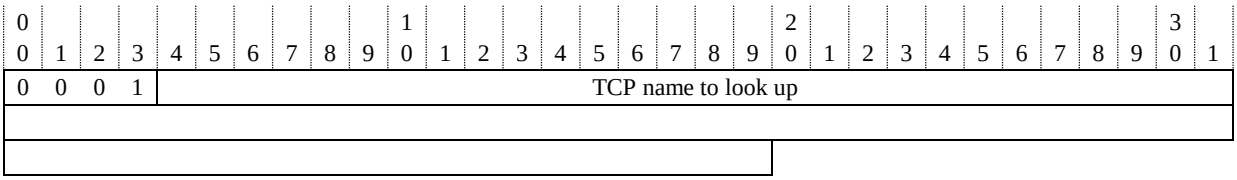

Figure 4 – TCP-ID name message format

This approach allows the internal distribution of discovery agents to be hidden from the receiving discovery agent. It also allows a discovery agent to manage TCP-ID name spaces larger than 32 bits.

8.1.2 DA DCN address format

The DA DCN address format contains the actual discovery agent ID and TCP-ID values. The discovery agent ID consists of a DCN context ID[2] as well as the DCN address of the sending discovery agent. The remainder of the message contains a TCP-ID, which has local significance to the discovery agent transmitting the DM. This is called the local TCP-ID.

The DM to be used with the DA DCN address format is shown in Figure 5. This format contains 4 bits of format ID, 16 bits of DA DCN context ID, 32 bits of DA DCN address and 32-bits of TCP-ID.

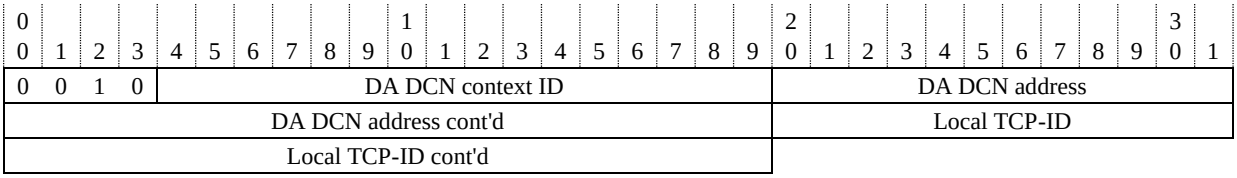

Figure 5 – DA DCN address message format

The use of this format is recommended when the distribution of the discovery agents is not hidden, and the DCN addresses as well as the TCP-IDs used by a discovery agent can fit within 32 bits.

8.1.3 DA DCN name format

Similar to the DA DCN address format, the DA DCN name format also contains the discovery agent name and the TCP-ID value. However, unlike the DCN address format, the discovery agent name is in the form of a DCN name. Consequently, a name-server must be used to translate the DCN name into the DCN address of the discovery agent.

As with the TCP-ID name format, the sending and receiving discovery agents are part of a federation that provides a name-server allowing the name to be uniquely resolved into the discovery agent DCN address and TCP-ID. The namespace may be subdivided among different name-servers that are responsible for resolving names within the assigned parts of the name space. The format of the name is defined by the context of the name-server, and is not specified here.

The remainder of the message contains the local TCP-ID, which has local significance to the discovery agent transmitting the DM.

The DM to be used with the DA DCN name format is shown in Figure 6. This format contains 4 bits of format ID, 48 bits of DA DCN name and 32 bits of TCP-ID.

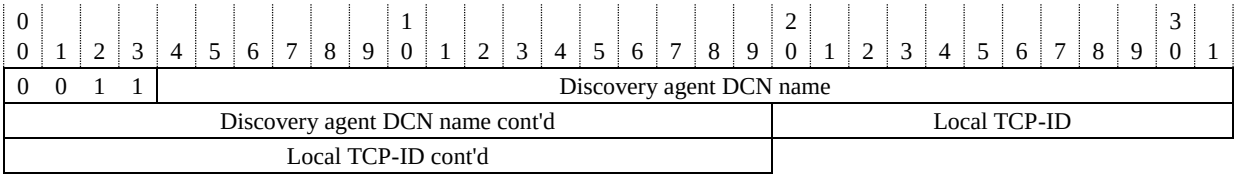

Figure 6 – DA DCN name message format

---

[2] The DCN context ID defines the context of the received DCN address. This value is included in the discovery message to aid in the debugging of the discovery process and is not interpreted by the receiving discovery agent. If the sending and receiving discovery agents at each end of the link are within different DCN contexts, but use the same DCN addresses, they may be unable to communicate. This ID may be, for example, a 2-byte Internet AS-Number as defined in [b-IETF RFC 1930]. If the DCN context ID has not been configured, then the value of 0 is used.

Unlike the TCP-ID name format, the discovery agent responsible for the TCP-ID value provided in this DM format is not hidden. Use of this format is recommended when the TCP-IDs, used by a discovery agent, can fit within 32 bits, but the DA DCN address cannot fit within 32 bits. This format also allows for independent reconfiguration of the DCN addresses used to reach the DA.

8.1.4 ETH MAC address format

Similar to the DCN address format, the ETH MAC name format also contains the discovery agent ID and the TCP-ID value. However, unlike the DCN address format, the discovery agent ID can have different types including: a chassis ID or an Ethernet MAC address. Consequently, a name-server may be needed to translate the DCN name into the DCN address of the discovery agent.

The DM also contains the local TCP-ID, which has local significance to the transmitting discovery agent. As with the chassis ID, the local TCP-ID can have different types, including: an Ethernet MAC address or an interface index.

A DM format has been defined for backward compatibility with SDH and OTN discovery agents making it possible to configure LLDP parameters without specifically supporting the LLDP management information base (MIB). This format, shown in Figure 7, contains 4 bits of format ID, containing a 48-bit Ethernet MAC address to be used for chassis ID and a 32-bit interface-index to be used as the port ID.

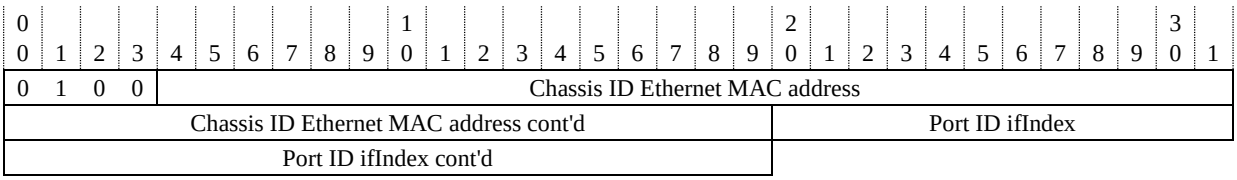

Figure 7 – ETH MAC address message format

This message is never actually sent in an LLDP message. Instead, it is used by a discovery agent to provide a network element (NE) with the chassis ID and port ID values for transmission in LLDP messages.

9 Layer adjacency based on embedded control channel messages

There are two functions required to realize embedded communication channel- (see [ITU-T G.7712]; ECC-) based LAD: the ECC link layer convergence function (LLCF) and LAD protocol control function. These messages are applied to the specific layer adjacency that is being discovered. Note that the ECC is provided by a technology-specific mechanism as specified in clause 6.

Figure 8 illustrates the header and data information included in each layer.

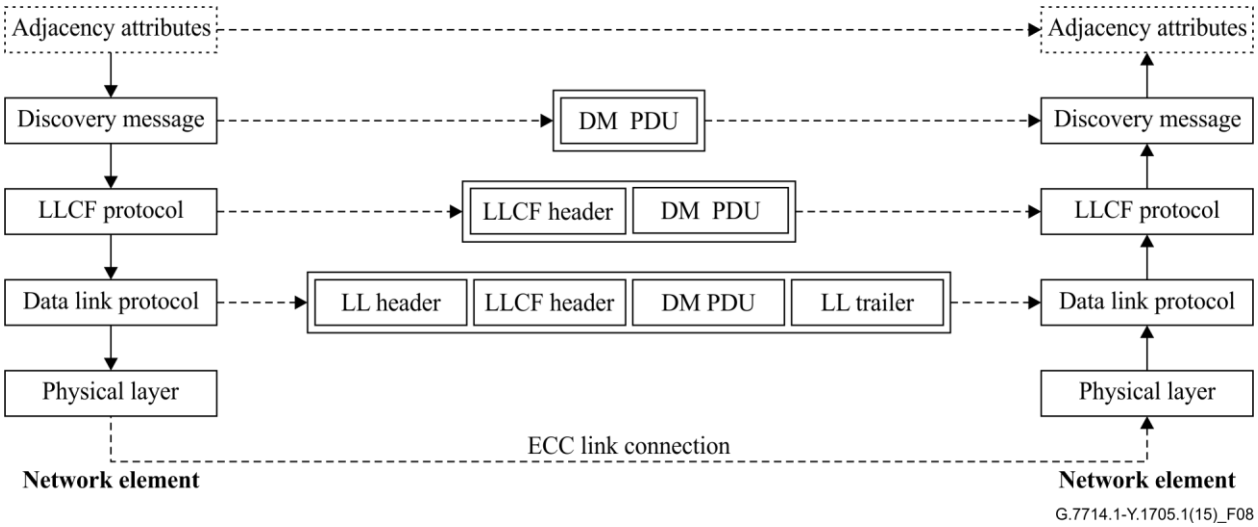

Figure 8 – Layer adjacency discovery functions of importance for ECC messages

Two mechanisms are available at the ECC based on the link layer protocol that is supported.

9.1 LAPD-based mechanism

[b-ITU-T G.784] requires both acknowledged information transfer service (AITS) and unacknowledged information transfer service (UITS) modes to be supported by every NE, so they can be utilized simultaneously over a single ECC channel. The LLCF uses link access procedure D-channel (LAPD) UITS for transport of LAD information. The interface from LLCF to LAPD utilizes DL_UNIT DATA primitives to request the transmission of unnumbered information frames. This discovery information is transferred between peer entities, employing the message used in the point-to-point protocol (PPP) transport.

The sending of DL_UNIT DATA primitives can occur at any time and does not affect the LAPD state machine, permitting the OSI/IP network layer to continue using AITS, if desired. Therefore, the DMs can be sent even in the cases where only unidirectional link exists or miswiring of a bidirectional connection.

The payload of this string shall be as defined for the trace (see clause 8) and shall interwork with PPP receivers.

9.2 PPP-based mechanism

Message exchange over PPP shall conform to [IETF RFC 1570] and [ITU-T G.7712] ([b-IETF RFC 1661] and [IETF RFC 1662]) using the identification message [link control protocol (LCP) code-point 12] defined in [IETF RFC 1570]. The payload of this string shall be as defined for the trace (see clause 8) and shall interwork with LAPD receivers.

9.3 Interoperable solution when using ECC based mechanism

When utilizing high-level data link control (HDLC; in SDH) over the ECC for discovery as described in clauses 9.1 and 9.2, the HDLC address field of the frame shall be used to distinguish between LAPD and PPP link layer frames using the second octet of the frame.

In a PPP frame the address field is set to a fixed value of All-Stations (single octet value 0xff) as specified in [IETF RFC 1662], whereas the address field (second octet) of the LAPD frame can never have a single octet value of 0xff. PPP MUST NOT be permitted to negotiate Address-and-Control-Field-Compression as outlined in clause 3.2 of [IETF RFC 1662] to allow both link layer protocols to function simultaneously over the ECC.

A PPP-only NE acting as a receiver would distinguish the adjacency discovery LAPD frame (based on contents of the HDLC address field), pass it through a trivial LAPD link layer to remove the fixed LAPD header fields, and pass the information field to the discovery agent. No other LAPD frames would need to be supported by a PPP-only NE.

On transmission the trivial LAPD link layer would place the local NE's discovery data in the LAPD type B unnumbered information packet information field with the specific addressing for discovery and pass it to HDLC for sending over the ECC.

When LAPD is used for discovery, a SAPI/terminal endpoint identifier (TEI) of 61/0 shall be used for DMs. This is different than the 62/0 used by OSI/LAPD over the DCC, as the unnumbered information (UI) frames for discovery could be mistaken for OSI protocol data units (PDUs).

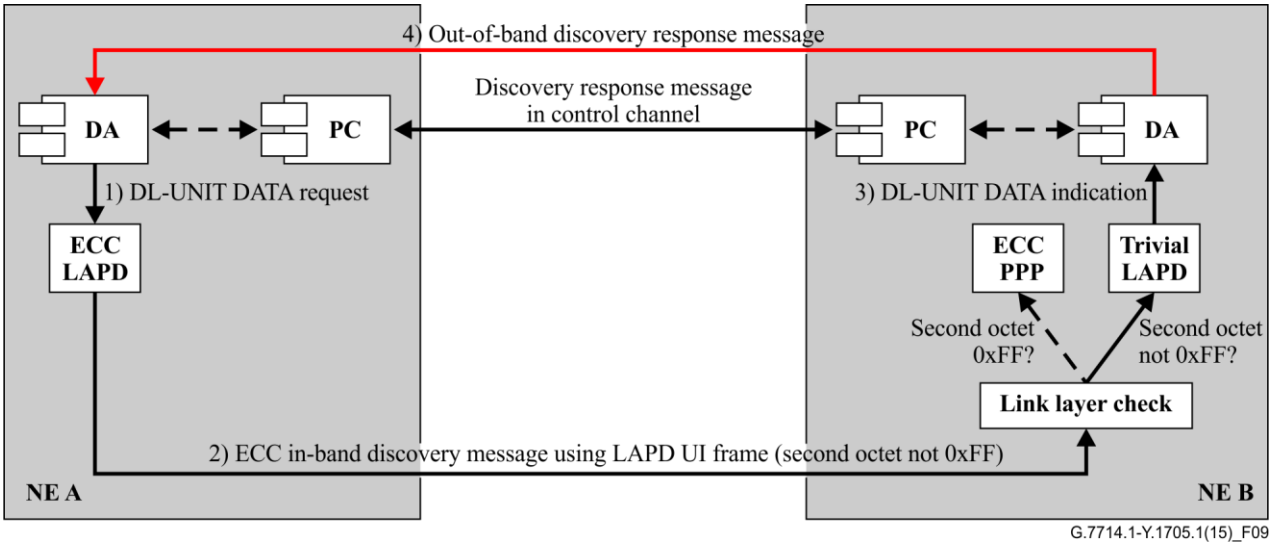

Figure 9 – ECC-based layer adjacency discovery using LAPD

This solution does not require that a PPP-only NE implement the LAPD state machine, given the unnumbered information frame mechanism being utilized passes through the existing LAPD state machine without forcing a state transition.

9.4 PPP-based support in OTN ECC

Message exchange over the OTN ECC can be performed over GCC-0, GCC-1 and GCC-2. These are specified in clauses 15.7.2.2 and 15.8.2.3 of [ITU-T G.709]. When using GCC-1 and GCC-2 for neighbour discovery, NEs on either end of the link must have access to the ODU frame structure. As per clause 7.1.2.2.2 of [ITU-T G.7712], PPP in HDLC framing over GCC is used.

9.4.1 LLDP/PPP

For the discovery trigger and adjacency discovery, the LCP identification message of PPP is used as defined in clause 9.2. The payload of the LCP identification message is an LLDP message as described in clause 10. In the chassis ID TLV of the LLDP message, an IP address is used as the DA DCN name or address, and the port ID field is used to convey the local TCP-IDs.

9.4.2 LLDP/MAC/PPP

LLDP may also be supported over MAC layer as originally specified in [IEEE 802.1AB]. Carrying MAC frames over PPP can be accomplished using the following options in [IETF RFC 3518].

1) After LCP/NCP protocols complete, run bridging control protocol (BCP) with the MAC-Support option (3) with MAC type of 1 for [IEEE 802.3] Ethernet.

2) After BCP reaches opened state, use clause 4.2 of [IETF RFC 3518] to send untagged frames. PPP bridging is specified as 0x0031 in the PPP header. The MAC type of 1 is also specified.

3) To send an LLDP packet in a MAC frame, fill the destination MAC with 01-80-C2-00-00-0E. This specifies the "*Nearest bridge*" Group MAC address for LLDP. The EtherType field is set to 0x88cc.

Discovery information is carried in LLDP as described in clauses 9.4.1 and 10.1.

10 Layer adjacency based on packet OAM frames

10.1 Ethernet link discovery mechanism

As with ECC-based LAD, there are two functions required to realize LLDP OAM frame discovery: the ETH OAM LLCF and LAD protocol control function. These OAM messages are applied to the Ethernet link being discovered.

Figure 10 illustrates the header and data information included by each layer.

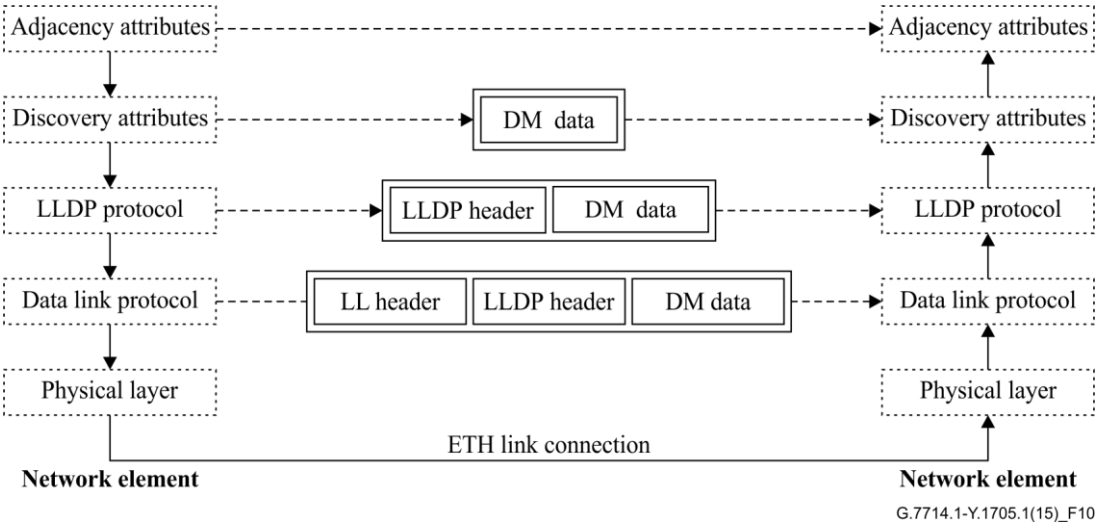

Figure 10 – Layer adjacency discovery functions of importance for ETH link discovery

The encoding of the DM is as defined in [IEEE 802.1AB] (LLDP). The outgoing PDU shall utilize destination and source MAC addresses as defined in clauses 7.1 and 7.2, respectively, of [IEEE 802.1AB] and carry the mandatory chassis ID, port ID, time-to-live and end of LLDP PDU TLVs. The chassis ID field is used to convey the DA DCN name or DA DCN address, and the port ID field is used to convey the local TCP-IDs.

Since LLDP is a one-way, transmit-only protocol, there is no mechanism for active bidirectional link detection or transport entity capability exchange (TCE) negotiation. However, passive TCE negotiation is possible. In this approach, the negotiation process does not actively exchange proposed, accepted and not-accepted configuration parameters. Instead, the process uses a well-defined set of behaviours that all NEs follow to determine configuration given the parameters provided.

As an example of passive negotiation, LLDP has an optional TLV to convey the maximum frame size supported by a port. Instead of specifically stating what size peer to use, with acknowledge or negative-acknowledge (ACK/NAK) exchanges, this option provides the peer with the max frame size supported and it is expected that the peer will observe that limit.

10.2 MPLS-TP transport entity discovery mechanism

For further study.

11 Procedures

The discovery methods and procedures described in this clause are independent of the transport mechanism. The procedure for LAD is as follows:

1) the initiating discovery agent transmits the DM, populating the attributes as required within clause 7;
2) upon receiving an appropriately formatted DM, the responding DA checks to determine the applicability of the message using the distinguishing character to validate the DM;
3) after determining that the received message is a DM, the responding discovery agent then determines whether the values received are unique with respect to already discovered neighbours as follows:
   i) if the DM uses a TCP-ID name format, a name-server is needed to determine the DA DCN address and TCP-ID,
   ii) if the DM uses a format containing the DA DCN Address, then no further address translation is needed,
   iii) if the DM uses a format containing a DA DCN Name, then address translation is needed to convert the DA DCN name into a DA DCN Address;
4) when active discovery is in use, generate a discovery response message.

12 Discovery response message

When the discovery agent receives the DM for the first time, it may notify the originating discovery agent using the discovery response message that the message was received on a TT associated with a particular TCP. This TCP, called the discovery sink TCP, is identified in the response using the discovery information currently being sent on the TCP. Additional optional attributes may be included as a part of an implementation. The discovery response message is sent using the DCN. It may also be sent in-band if the technology is capable of that. See Table 1.

Table 1 – Discovery response message attributes

<Received DA DCN ID>	DA DCN ID contained in the received DM
<Received TCP-ID>	TCP-ID contained in the received DM
<Sent DA DCN ID>	DA DCN ID actively being sent by the responding discovery agent
<Sent Tx TCP-ID>	TCP-ID actively being sent by the responding discovery agent
<Sent Rx TCP-ID>	Identifier for the TCP on which the DM was received

The received DA DCN ID field shall be included in the discovery response if the following condition is met: the received DM includes a DA DCN ID. If the DA DCN ID is a DCN name, the name must copied exactly into the response message and not be translated when being sent in the discovery response. This attribute shall not be included if a DA DCN ID was not included in the received DM (i.e., the TCP-ID format is in use).

The sent DA DCN ID field shall be included in the discovery response if the following condition is met: the format of the DM currently being sent on the discovery Sink TCP includes a DCN identifier. The sent DA DCN ID will contain the same DA DCN ID being sent on the discovery Sink TCP. This attribute shall not be included if the DA DCN ID is not included in the current DM being sent on the discovery Sink TCP.

The received TCP-ID is the TCP Identifier received in the DM. The format of the TCP-ID is determined by the format of the DM that was received.

The sent Tx TCP-ID is the TCP Identifier currently being sent in DMs on the discovery Sink TCP. The format of this identifier is determined by the format of the DM being sent.

The sent Rx TCP-ID is the TCP identifier for the receive side of the discovery Sink TCP. The format of this identifier is the same as for the sent Tx TCP-ID. This shall always be sent with bidirectional links, allowing for different TCP-IDs to be used for the Tx and Rx directions on a Trail. It shall also be sent when the Tx and Rx TCP-IDs are the same. This attribute shall not be sent for a unidirectional TCP-endpoint.

The DCN Address of the discovery agent to which the discovery response message is sent will be determined from the DA DCN ID received in the DM. If the format of the DM received does not include a DA DCN ID, then it is expected that a name-server function has been provided to allow the DCN Address to be looked up given the received TCP-ID.

When the DA DCN ID received in a DM is a name, then it is expected that a name-server function has been provided to allow the DCN address to be looked up given the received DA DCN name. However, if the DA DCN ID received contains a DCN address, then the DCN address may be used directly.

12.1 Miswiring detection

Once a DM has been received on a resource and a discovery response message describing the same resource is received over the DCN, it is possible to correlate the messages and determine whether a bidirectional link exists. If the TCP-ID corresponding to the remote endpoint of the LC is not the same in both messages, then a miswired condition exists. If the TCP-ID is the same, then the Transmit/Receive signal pair has been properly wired. This is described in greater detail in Appendix II.

12.2 Misconnection detection

Once a bidirectional link has been discovered, it should be checked against management-provided policy to determine whether correct TCP-LC endpoints have been correctly connected. If the policy states that the TCP-LC endpoints may not be paired to form a link, then a misconnection condition exists. In absence of this policy, it is not possible to identify a misconnection condition.

Appendix I

Implementation example of discovery process

(This appendix does not form an integral part of this Recommendation.)

This appendix provides an implementation example intended to validate the protocol design choices made and specified in this Recommendation.

I.1 Layer adjacency discovery information flow

As described in [ITU-T G.7714], the discovery process includes the following steps:

– LAD;

– service capability exchange.

Completing the LAD process requires a number of functions to interact to identify the TCP LC. Further, the relationship between the LAD process and the service capability exchange mechanism needs to be described. Figure I.1 is a sequence diagram detailing the interactions.

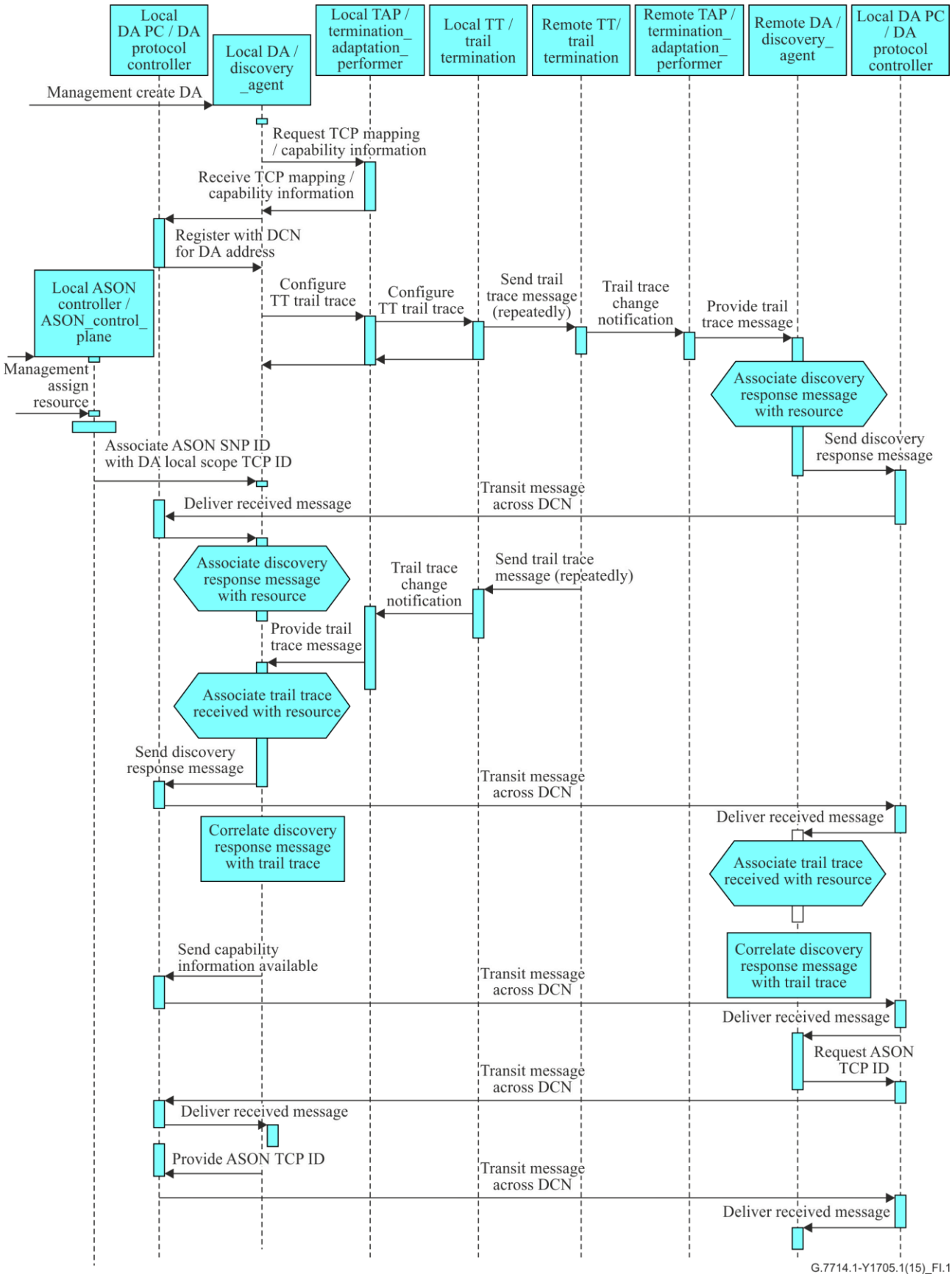

Figure I.1 – Sequence diagram

Appendix II

Miswiring detection

(This appendix does not form an integral part of this Recommendation.)

This appendix describes how the LAD procedure can detect that the interfaces between two NEs are miswired. In the examples of this appendix, the DA DCN address format as defined in clause 8.1.2 is used for the in-band DM. This does not, however, preclude other message formats from being used.

II.1 Auto-discovery procedures

To automatically discover a layer adjacency between two NEs (e.g., NE A and NE B), both NEs have to perform the discovery procedure in order to learn the association between the local TCPs and the remote TCPs. The two discovery processes on the two NEs are executed independently, i.e., there is no specific protocol message exchange that triggers the neighbouring NE to perform the discovery process. This is depicted in Figures II.1 and II.2. Figure II.1 illustrates the discovery process that is initiated by the DA responsible for NE A, whereas Figure II.2 shows the process that is triggered by the DA responsible for NE B. When the discovery process initiated by the DA related to NE A (DA_A) is completed (i.e., DA_A has received the discovery response message), both DA_A and DA_B (DA related to NE B) have the following set of information elements:

< DA-ID_A, TCP-ID_A_t, DA-ID_B, TCP_ID_B_r, [TCP_ID_B_t] >

These information elements have the following meaning:

– DA-ID_A: DCN ID of DA related to NE A;

– TCP-ID_A_t: local TCP-ID of TCP in NE A from which the DM was transmitted;

– DA-ID_B: DCN ID of DA related to NE B;

– TCP_ID_B_r: local TCP_ID of TCP in NE B that received the DM from NE A;

– [TCP_ID_B_t]: local TCP_ID of TCP in NE B (transmit direction) associated with TCP_ID_B_r.

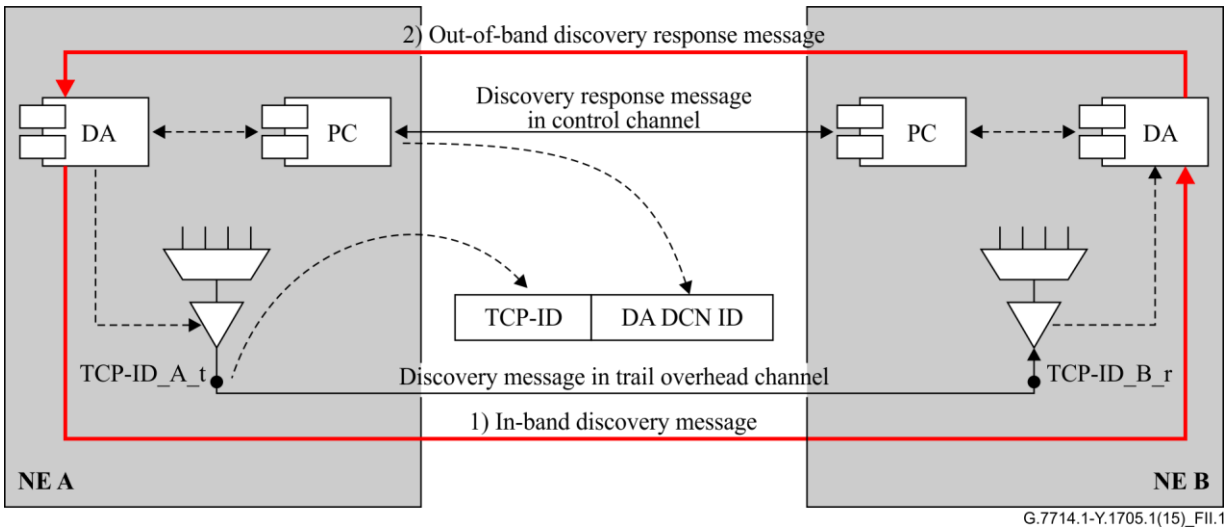

Figure II.1 – Layer adjacency discovery procedure initiated by NE A

When the discovery process initiated by the DA related to NE B (DA_B) is completed (i.e., DA_B has received the discovery response message), both DA_B and DA_A have the following set of information elements:

< DA-ID_B, TCP-ID_B_t, DA-ID_A, TCP_ID_A_r, [TCP_ID_A_t] >

These information elements have the following meaning:

– DA-ID_B: DCN ID of DA related to NE B;

– TCP-ID_B_t: local TCP-ID of TCP in NE B from which the DM was transmitted;

– DA-ID_A: DCN ID of DA related to NE A;

– TCP_ID_A_r: local TCP_ID of TCP in NE A that received the DM from NE B;

– [TCP_ID_A_t]: local TCP_ID of TCP in NE A (transmit direction) associated with TCP_ID_A_r.

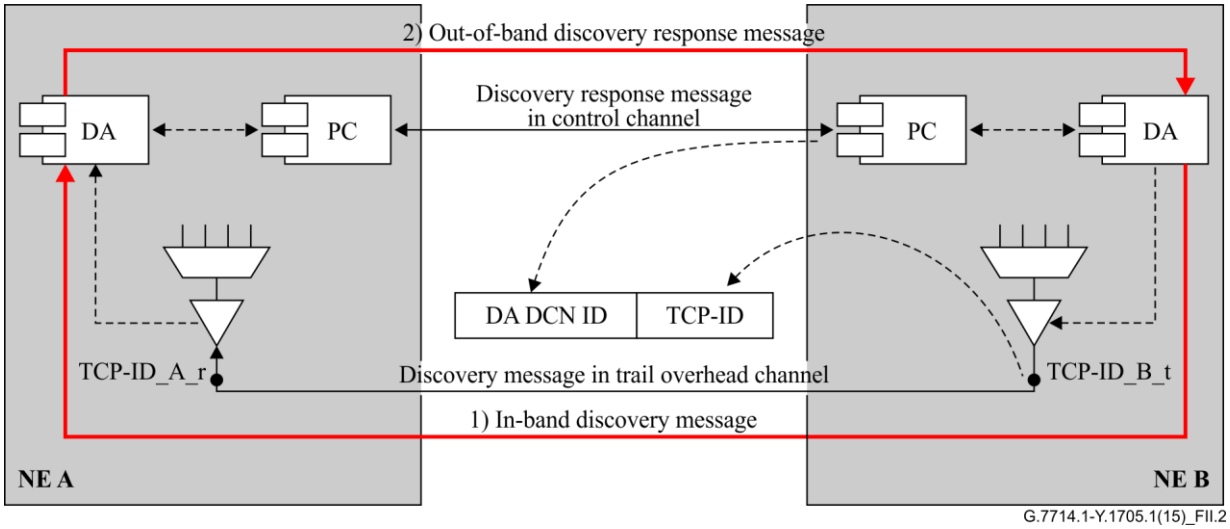

Figure II.2 – Layer adjacency discovery procedure initiated by NE B

In order to perform miswiring detection, it is necessary that both discovery processes on the two neighbouring NEs (NE A and NE B) have completed. Once both DA_A and DA_B have reached this state, they both have the following two sets of information elements that can be correlated for miswiring detection on either side (see Figure II.4):

– < DA-ID_A, TCP-ID_A_t, DA-ID_B, TCP_ID_B_r, [TCP_ID_B_t] > and

– < DA-ID_B, TCP-ID_B_t, DA-ID_A, TCP_ID_A_r, [TCP_ID_A_t] >

From the perspective of DA_A, the two sets of information elements that are bound to the same local pair of TCPs need to be found in a first step. This can be done based on the local TCP-IDs that were locally assigned to the TCPs (TCP_ID_A_t in the transmit direction, i.e., from NE A to NE B and TCP-ID_A_r in the receive direction, i.e., from NE B to NE A). When the two information element sets are identified that are locally bound together, the following consistency checks can be performed:

– Check whether the DA-IDs are the same on both sides.

– Check whether the remote TCP-IDs (TCP-ID_B_t and TCP-ID_B_r) are also bound to the correct TCPs on the remote side.

  Depending on whether the same TCP-ID value is used for the remote TCPs in transmit and receive direction or whether they both have different values, the DA_A needs to know the binding between the two TCP-IDs on the remote side. If the remote TCP-IDs in the transmit and receive direction are the same (TCP-ID_B_t = TCP-ID_B_r) the remote DA (DA_B) does not need to include the TCP-ID in the transmit direction (TCP-ID_B_t) in the discovery

response message. If the remote TCP-IDs are different (TCP-ID_B_t ≠ TCP-ID_B_r), the remote DA (DA_B) must include the optional TCP-ID in the transmit direction (TCP-ID_B_t) in the discovery response message.

The DA-ID check ensures that the same two DAs are involved in the discovery process in both directions (the one initiated by DA_A and the one initiated by DA_B). This also ensures that the scope of the TCP-IDs is the same. It shall be noted that the TCP-IDs only have local significance and are only unique within the scope of a single DA.

When the DA-ID check is passed successfully, the consistency check on the remote TCP-IDs can be performed. It checks whether the pairs of remote TCP-IDs received via the out-of-band discovery response message and the in-band DM from DA_B are consistent.

In the two examples shown in Figure II.3 and Figure II.4, the TCP-IDs in transmit and receive direction on both NE A and NE B are the same. In Figure II.3, the wiring between NE A and NE B is correct. In Figure II.4, the interfaces I/F n and I/F m on NE A and the interfaces I/F k and I/F l on NE B are miswired. Table II.1 and Table II.2 contain the corresponding sets of discovery information DA_A has obtained after the DM exchange.

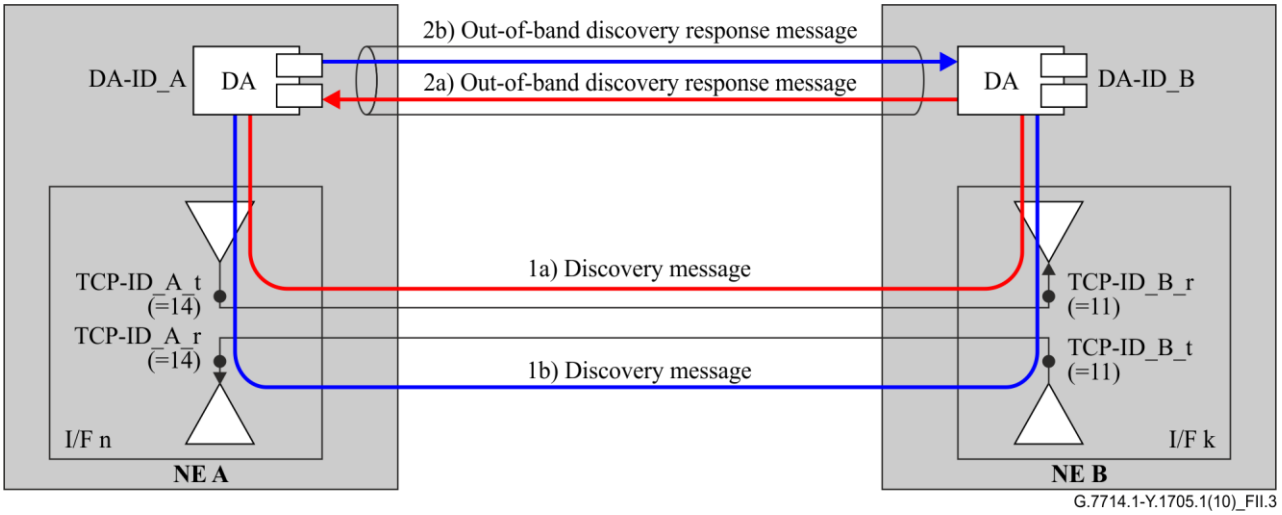

Figure II.3 – Auto-discovery in case of correctly wired interfaces

Table II.1 – Example of the two sets of discovery information from the perspective of DA_A for the correctly wired case depicted in Figure II.3

Process Initiator	<Received DA DCN ID>	Received TCP-ID associated with interface n	<Sent DA DCN ID>	<Sent Tx TCP-ID> associated with interface k	Optional <Sent Tx TCP-ID> associated with interface k/n
DA_A	DA-ID_A	TCP-ID_A_t	DA-ID_B	TCP-ID_B_r	TCP-ID_B_t
Value	1	***14***	2	***11***	11
DA_B	DA-ID_B	TCP-ID_B_t	DA-ID_A	TCP-ID_A_r	TCP-ID_A_t
Value	2	***11***	1	***14***	14

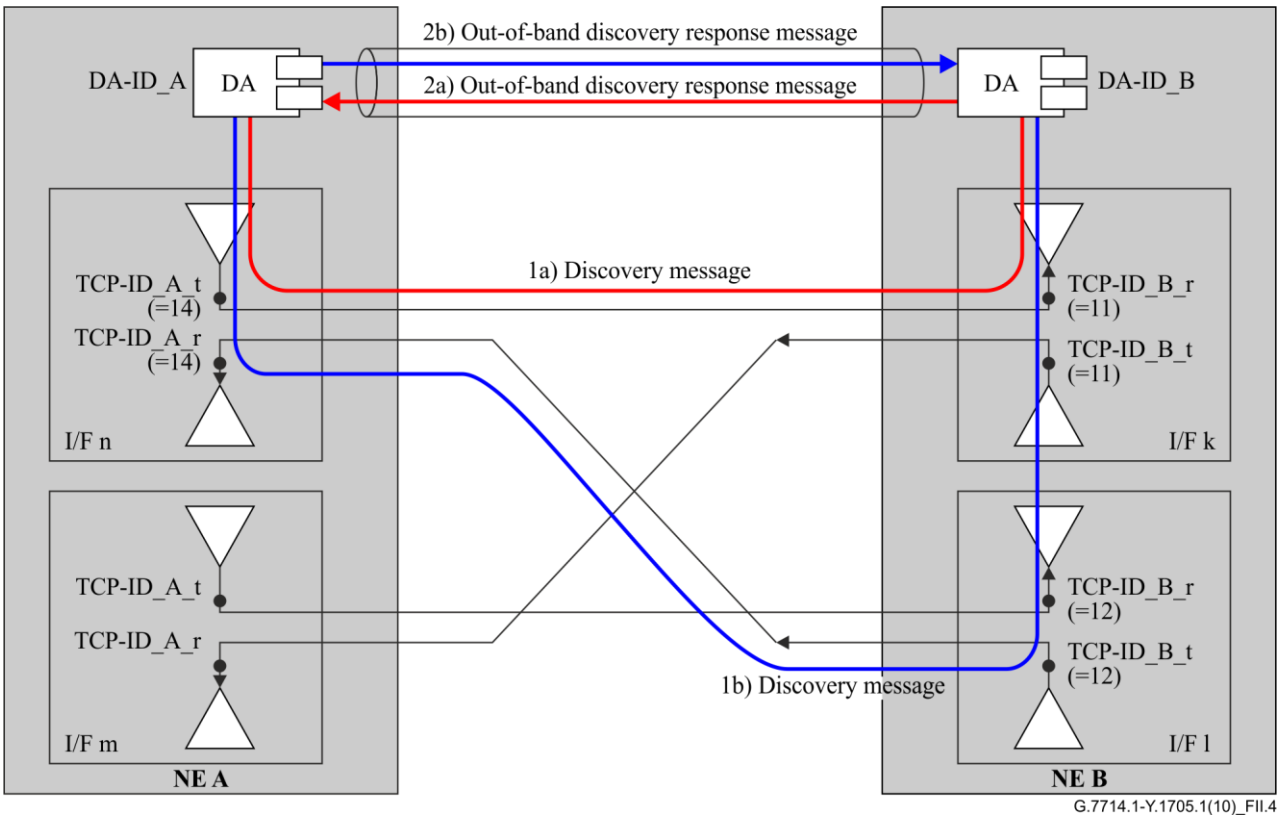

Figure II.4 – Auto-discovery in case of miswired interfaces

Table II.2 – Example of the two sets of discovery information from the perspective of DA_B for the miswired case depicted in Figure II.4

Process Initiator	<Received DA DCN ID>	Received TCP-ID associated with interface n	<Sent DA DCN ID>	<Sent Tx TCP-ID> associated with interface k	Optional <Sent Tx TCP-ID> associated with interface k/n
DA_A	DA-ID_A	TCP-ID_A_t	DA-ID_B	TCP-ID_B_r	TCP-ID_B_t
Value	1	***14***	2	***11***	11
DA_B	DA-ID_B	TCP-ID_B_t	DA-ID_A	TCP-ID_A_r	TCP-ID_A_t
Value	2	***12***	1	***14***	14

The values given in Table II.1 show that the two sets of discovery information from the perspective of DA_A belong together because the local TCP-ID_A_t and TCP-ID_A_r have the same value (TCP-ID_A_t = TCP-ID_A_r = 14) and hence, are related to the same bidirectional TCP. In the next step, the DA-ID consistency check is performed. In the examples given, the information sets are DA-consistent because the sent and received DA DCN IDs (1-2 and 2-1) indicate that the same two DAs are involved in the discovery process. Finally, whether the remote TCP-IDs (from the DA_A perspective) refer to the same remote TCP is checked. In the example, the check leads to a positive result, since TCP-ID_B_r and TCP-ID_B_t are equal and both have the same value 11 in the two discovery information sets.

In the second example, all the checks are passed successfully, as in the previous example, except the final remote TCP consistency check. This final TCP-ID check reveals that the remote TCP-IDs (from the perspective of DA_A) do not refer to the same remote TCP, because TCP-ID_B_r and TCP-ID_B_t have different values (11 in the out-of-band discovery response message versus 12 in the in-band DM). Hence, DA_A can indicate the detected miswiring by raising an appropriate alarm, for example.

II.2 Example: Interaction between two DAs using different DM formats

The procedure also works when two DAs actively engaged in discovering a link are using different DM formats. This example shows one discovery agent using TCP name format, while the other is using DCN DA address format.

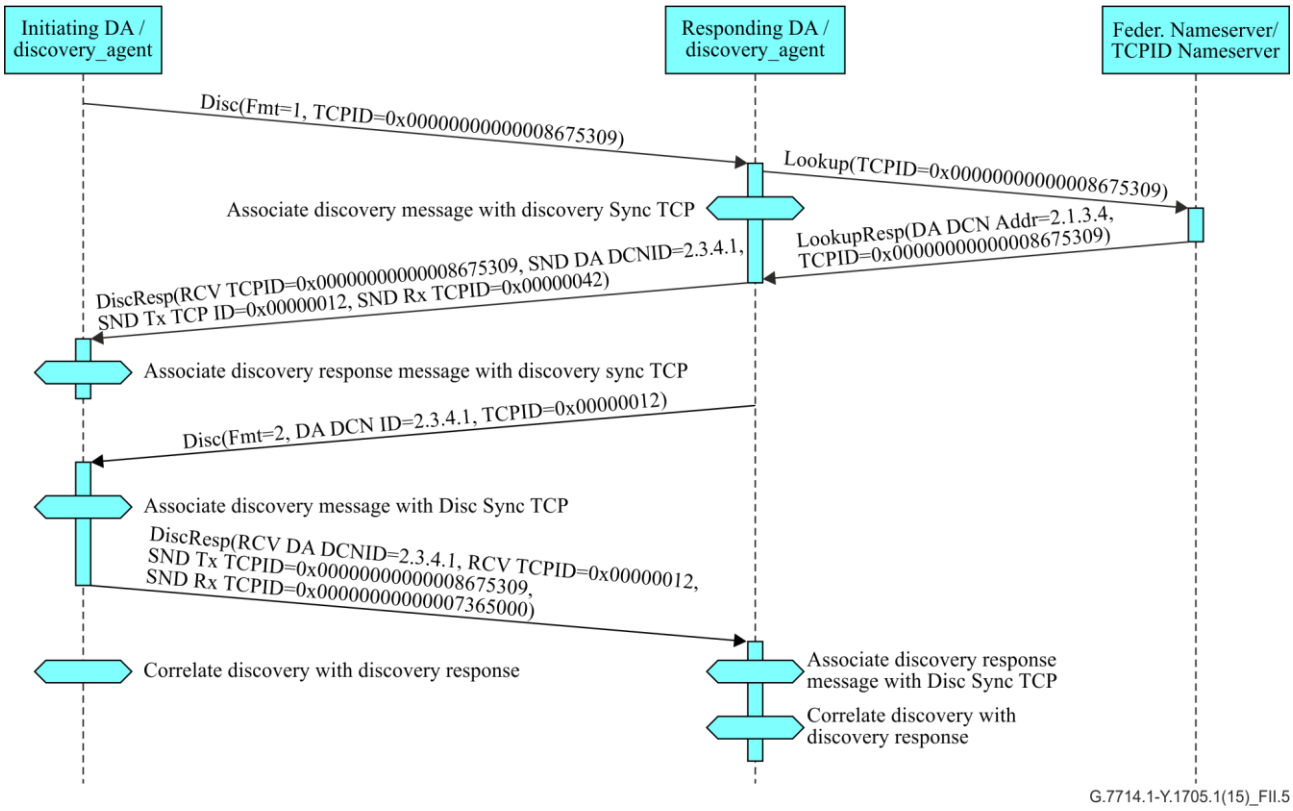

Figure II.5 – Sequence for discovery between two DAs using different message formats

In this example, the initiating discovery agent sends a DM in TCP name format. The DM *DISC(Fmt=1, TCPID=0x0000000000008675309)* is sent in band to the responding discovery agent. When received by the responding discovery agent, the Rx TCP (*0x42)* on which the DM was received is recorded. This is called the Sink TCP.

The TCP name *(0x0000000000008675309)* in the received DM is then translated into the DA DCN address for the initiating discovery agent *(address = 2.1.3.4)* and TCP-ID *(0x0000000000008675309)* using a name-server.

Once the DA DCN address is known, a discovery response message is returned to the initiating discovery agent. The discovery response message includes the attributes in the received DM, the attributes that are currently being sent on the Tx TCP *(Fmt=2, DA DCN Address=2.3.4.1, Tx TCPID=0x0000 0012)* related to the Sink TCP, as well as the TCP-ID for the Sink TCP *(Rx TCPID=0x0000 0042)*. Once received by the initiating discovery agent, a unidirectional LC has been identified.

This process is repeated for the opposite direction. However, since this time the DA DCN address format is being used, the DM *DISC (Fmt=2, DA DCN Address=2.3.4.1, TCPID=0x0000 0012)* sent includes a DA DCN Address and TCP-ID. When the DM is received, the Sink TCP on which it was received is recorded *(0x0000000000007365000).* Since the DM received includes a DCN Address, the Discovery Response can be returned without a name-server lookup.

As before, the response includes the attributes in the DM received, the current attributes being sent on the Tx TCP *(Fmt=1, TCPID=0x0000000000008675309)* related to the Sink TCP, as well as the

TCP-ID of the Sink TCP *(0x0000000000007365000)*. Again, when the discovery response message is received, a unidirectional LC has been identified.

At this time, it is possible for the discovery response messages to be correlated by each of the ends of the LC to determine whether the bidirectional link has been miswired. Specifically, see Table II.3.

Table II.3 – Correlation of discovery response messages

	A ≥ B Tx TCPID	**A ≥ B Rx TCPID**	**B ≥ A DA DCN ID**	**B ≥ A Tx TCPID**	**B ≥ A Rx TCPID**
A ≥ B	00000000000008675309		2.3.4.1	0x12	0x42
B ≥ A	00000000000008675309	00000000000007365000	2.3.4.1	0x12	

Since the A ≥ B Tx TCPID, the B ≥ A DA DCN ID, and B ≥ A Tx TCPID fields match, the link is correctly connected.

The Tx and Rx TCPIDs may now be provided to service capability exchange to determine the capabilities of the link.

Appendix III

Example of discovery response message using a generalized multi-protocol label signalling-based mechanism

(This appendix does not form an integral part of this Recommendation.)

This appendix illustrates one implementation of the LAD as described in this Recommendation, using a generalized multi-protocol label signalling- (GMPLS-) based mechanism. Other possible GMPLS-based implementations are left for further study.

This example assumes the use of the DA DCN-ID (in-band) DM format (as described in clauses 8.1.2 and 8.1.3) and that the bidirectional control channel between involved parties is established and available for exchanging the discovery response message (as described in clause 11). The bidirectional control channel establishment and maintenance mechanisms and related message exchange are outside of the scope of this appendix. In addition, it is assumed that at a given TCP-ID represents both transmitter and receiver, i.e., the identifier of the TCP where the (received) TCP-ID is received corresponds to the sent TCP-ID.

In this context, when using J0, the local/remote TCP-ID is equivalent to an interface index, and referenced as an unnumbered LOCAL/REMOTE INTERFACE_ID, respectively. When using J1/J2, the local/remote TCP-ID is equivalent to an SDH Label (at both end-points) that can be referenced as an unnumbered LOCAL/REMOTE INTERFACE_ID, respectively. The local/Remote DA DCN-ID corresponds to the IPv4 LOCAL_/REMOTE_CONTROL_ ADDRESS of the local/remote discovery agent, respectively.

In Figure III.1, summarizing the DM exchange, Node A is referred to as the remote node, and Node B as the local node.

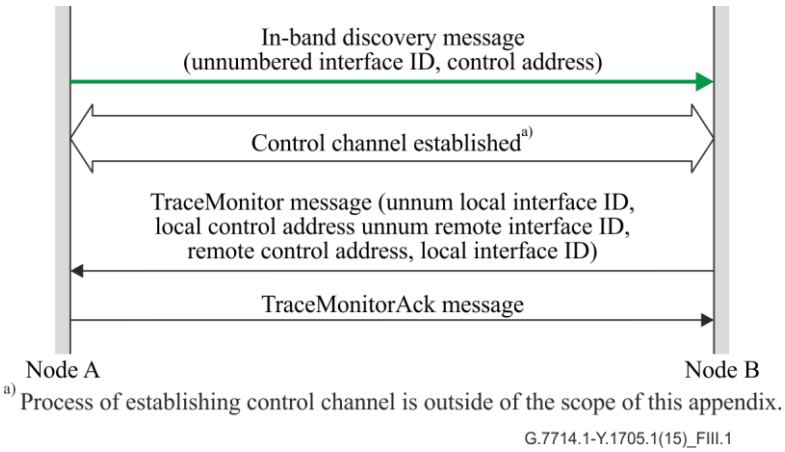

Figure III.1 – Summary of discovery messages used in GMPLS based implementations

Upon reception of the in-band DM from the DA of node A, an out-of-band discovery response message referred to as the (extended) TraceMonitor message is sent toward the DA of node B over the bidirectional control channel using user datagram protocol/Internet protocol (UDP/IP). This message includes the following information elements (i.e., objects):

<TraceMonitor Message> ::= <Common Header> <MESSAGE_ID> <LOCAL_INTERFACE_ID> <TRACE>[<REMOTE_TRACE>]

where:

<TRACE> ::= <Trace Type> <Trace Length> <Trace Message>

<Trace Type> type of the trace byte (i.e., J0, J1 or J2) used by the local in-band discovery message

<Trace Length> length in bytes of the <Trace Message>

<Trace message> contains the <unnumbered LOCAL_INTERFACE_ID> and the <LOCAL_CONTROL_ADDRESS> fields

<REMOTE_TRACE> ::= <Trace Type> <Trace Length> <Trace Message>

<Trace Type> type of the trace byte (i.e., J0, J1 or J2) used by the remote in-band discovery message

<Trace Length> length in bytes of the <Trace Message>

<Trace message> contains the <unnumbered REMOTE_INTERFACE_ID> <REMOTE_CONTROL_ADDRESS> fields

Upon reception of the TraceMonitor message from the DA of node B, a TraceMonitorAck message is sent to the DA of Node A to acknowledge its reception.

<TraceMonitorAck Message>::= <Common Header> <Message_ID_ACK>

NOTE – Subsequent message exchanges are outside of the scope of this appendix.

Appendix IV

Layer adjacency discovery implementation examples

(This appendix does not form an integral part of this Recommendation.)

The use of discovery is independent of the automatically switched optical network (ASON) control plane realization, which may range from fully centralized to fully distributed.

– Example 1: External discovery agent controlling trail trace or ECC to implement LAD

  When the discovery agent is located in an external system, an external interface is used by the NE to provision and receive the trail trace message. As an existing text-oriented man-machine language may be reused to provide this interface, the DM should be limited to printable characters defined by [ITU-T T.50].

– Example 2: Internal discovery agent controlling trail trace or ECC to implement LAD

  When the discovery agent is located on the NE, the interface used to provision and receive the trail trace message is a local implementation matter.

Appendix V

In-band message encoding example

(This appendix does not form an integral part of this Recommendation.)

Given the message formats defined in clause 8.1, the transmission of the TCP-ID, and discovery agent name or address is accomplished by encoding a sequence of six bits as a printable ITU-T T.50 character. The mapping of the bits to printable ITU-T T.50 characters is defined in [IETF RFC 2045]. Figure V.1 shows the relationship of the octet string to be mapped, and the printable string that results from mapping.

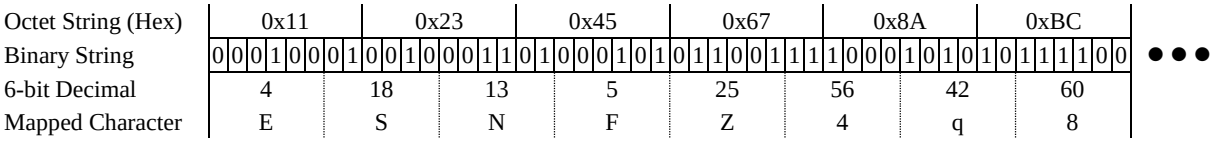

Figure V.1 – Relationship between DM octet string and 6-bit mapped characters

Once the DM has been mapped, the distinguishing character "+" is prepended yielding the discovery string.

Some example encoding for the different formats are as follows:

Format 1: TCP name format

Format type:	0001_2
Name:	0x1234 5678 ABCD EF00 4321
The octet string that will be mapped is:	0x1123 4567 8ABC DEF0 0432 1x[3]
The printable character string after mapping is:	$ESNFZ4q83vAEMh_{64}$
The resulting discovery string is:	+ESNFZ4q83vAEMh

Format 2: DA DCN address format

Format Type:	0010_2
DCN Context ID:	0x0000 (octet string)
DA DCN address:	0x10203040 (octet string)
TCP-ID:	0x12345678 (octet string)
The octet string that will be mapped is:	0x2000 0102 0304 0123 4567 8x[3]
The printable character string after mapping is:	$IAABAgMEASNFZ4_{64}$
The resulting discovery string is:	+IAABAgMEASNFZ4

[3] Since 14 characters are available in the trace message, 84 bits are available for carrying the discovery data. This yields 10 octets, with 4 bits remaining. The last octet shown here contains the 4 remaining bits in the high order nibble, causing the lower order nibble to have no meaning as signified by the "x" used here and is not mapped.

Format 3: DA DCN name format

Format Type:	0011_2
Name:	0x9876 5432 10AA
TCP-ID:	0x12345678 (octet string)
The octet string that will be mapped is:	0x3987 6543 210A A123 4567 $8x^3$
The printable character string after mapping is:	$OYdlQyEKoSNFZ4_{64}$
The resulting discovery string is:	+OYdlQyEKoSNFZ4

Appendix VI

Usage of the different discovery mechanisms

(This appendix does not form an integral part of this Recommendation.)

VI.1 Introduction

This appendix provides clarification of the network scenarios under which the various discovery mechanisms described in the main body of this Recommendation may be utilized, including guidelines for usage of mechanisms and procedures as well as potential associated implications.

VI.2 Categories of Type 1 layer adjacency discovery use cases

The auto-discovery use cases can be subdivided into the categories depicted in Figure VI.1, i.e., pre-service, in-service and out-of-service. Within the context of this Recommendation, the terms pre-service, in-service and out-of-service are defined as follows.

pre-service: The entity that is in a pre-service state is the trail whose associated client LCs have not been allocated. As a consequence, operations will not impact any traffic. Pre-service includes scenarios where discovery is done immediately after a fault has been cleared and before service is considered restored (e.g., during a soaking interval).

in-service: The entity that is in an in-service state is the trail whose associated client LCs have been allocated (one or more).

out-of-service: The entity that is in an out-of-service state is the trail where all allocated client LCs are in a failed or non-usable state.

This appendix only addresses auto-discovery use cases where the applied auto-discovery mechanism may cause some behavioural problems in the network, i.e., in-service cases. The pre-service and out-of-service use cases, drawn with dotted lines in Figure VI.1, are not further discussed. Moreover, a type 2 LAD is also not considered because the LCs cannot be in service (i.e., carry traffic) at the same time as a type 2 LAD is applied (see [ITU-T G.7714] for the definition of type 1 and type 2 LADs).

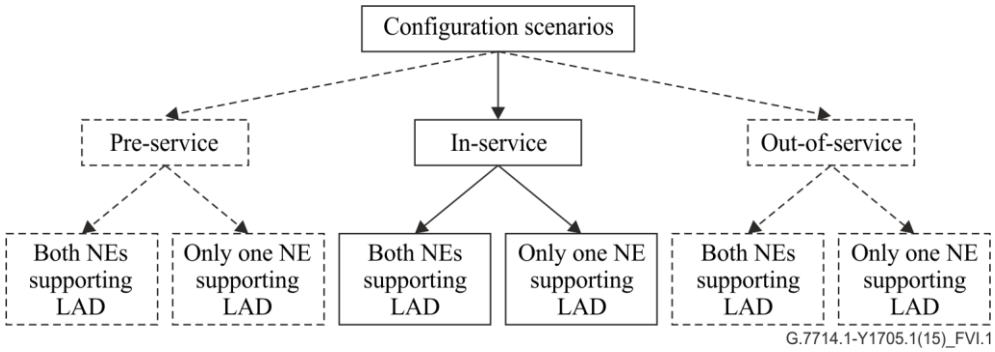

Figure VI.1 – Categorization of discovery scenarios

VI.3 Use cases and scenarios

The various use cases where a type 1 LAD can be applied are described in this clause and guidelines are provided in clause VI.4 that explain how discovery can be accomplished based on the constraints imposed by the different scenarios. As specified in the main body of this Recommendation, it is assumed that there is always congruency between the signal being used for LAD and the entity being discovered. In describing the various scenarios, two cases are broadly distinguished:

a) where all the NEs are auto-discovery capable; and

b) where some of the NEs within the network are not auto-discovery capable.

VI.3.1 All NEs are auto-discovery capable (ubiquitous deployment)

Ubiquitous deployment means that all NEs are auto-discovery capable and it is assumed that all involved NEs support LAD as defined in [ITU-T G.7714] and in the main body of this Recommendation. For this subset of cases, either trail-trace-based or ECC-based DMs can be used, provided all the NEs agree on a specific common mechanism.

VI.3.2 All NEs are not auto-discovery capable

If some of the NEs within the network are assumed to be unable to understand the auto-DMs (e.g., legacy equipment). We consider two scenario classes for the case where auto-discovery is being performed at a particular layer between the two NEs that represent the endpoints of that layer:

– where both NEs are LAD-capable;

– where one of the two NEs does not support LAD.

VI.3.2.1 Auto-discovery between LAD-capable NEs

As described in [ITU-T G.7714], the LAD process requires that two NEs that are performing LAD must be immediate neighbours with respect to the layer where discovery is taking place (e.g., for SDH at RS, MS, HO or LO path layer). It is not possible, for example, to perform LAD based on using the section trail trace (J0), RS DCCs, or MS DCCs when there is a NE between the two LAD-capable NEs that does not support LAD and terminates the RS and MS. Therefore, it is only possible to perform LAD at the path layer for such a configuration and the HOVC path-trace (J1)-based discovery method may have to be used. This is illustrated in Figure VI.2. It is also possible for the network management system (NMS) to run the HO path layer LAD process by proxy for the NEs, as depicted in Figure VI.3.

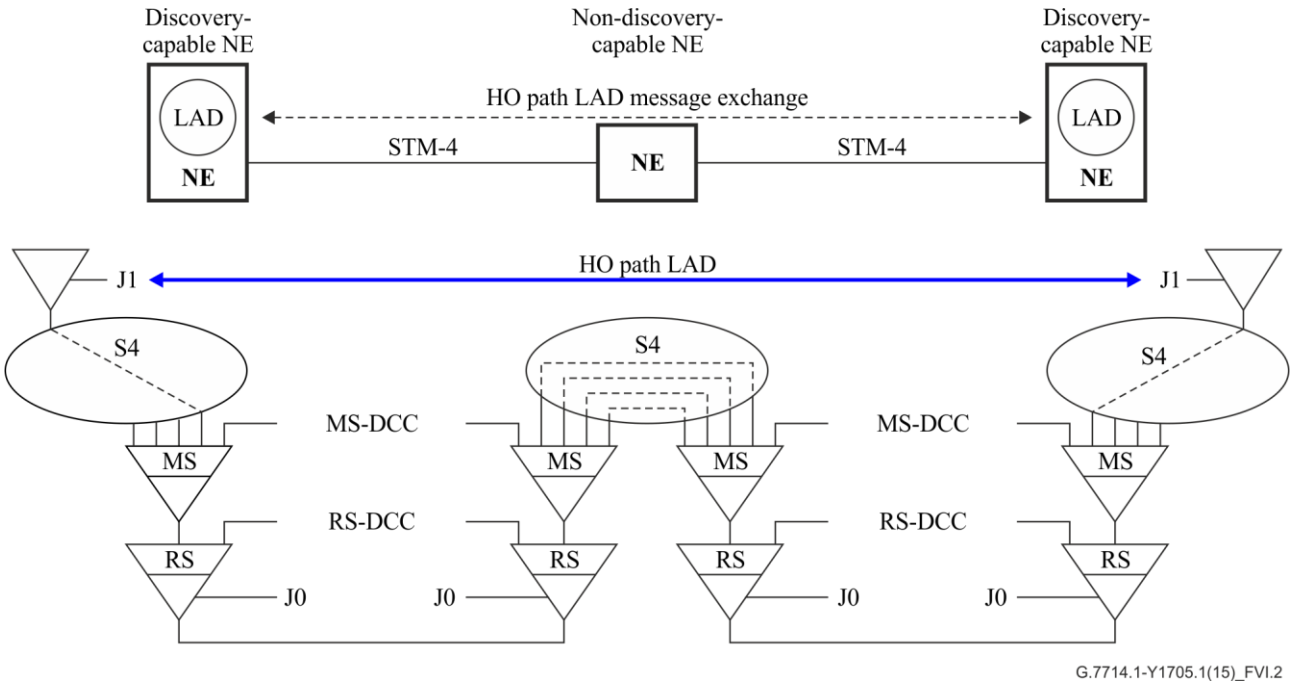

Figure VI.2 – Immediate discovery-capable neighbours at HO path layer – LAD done by NEs

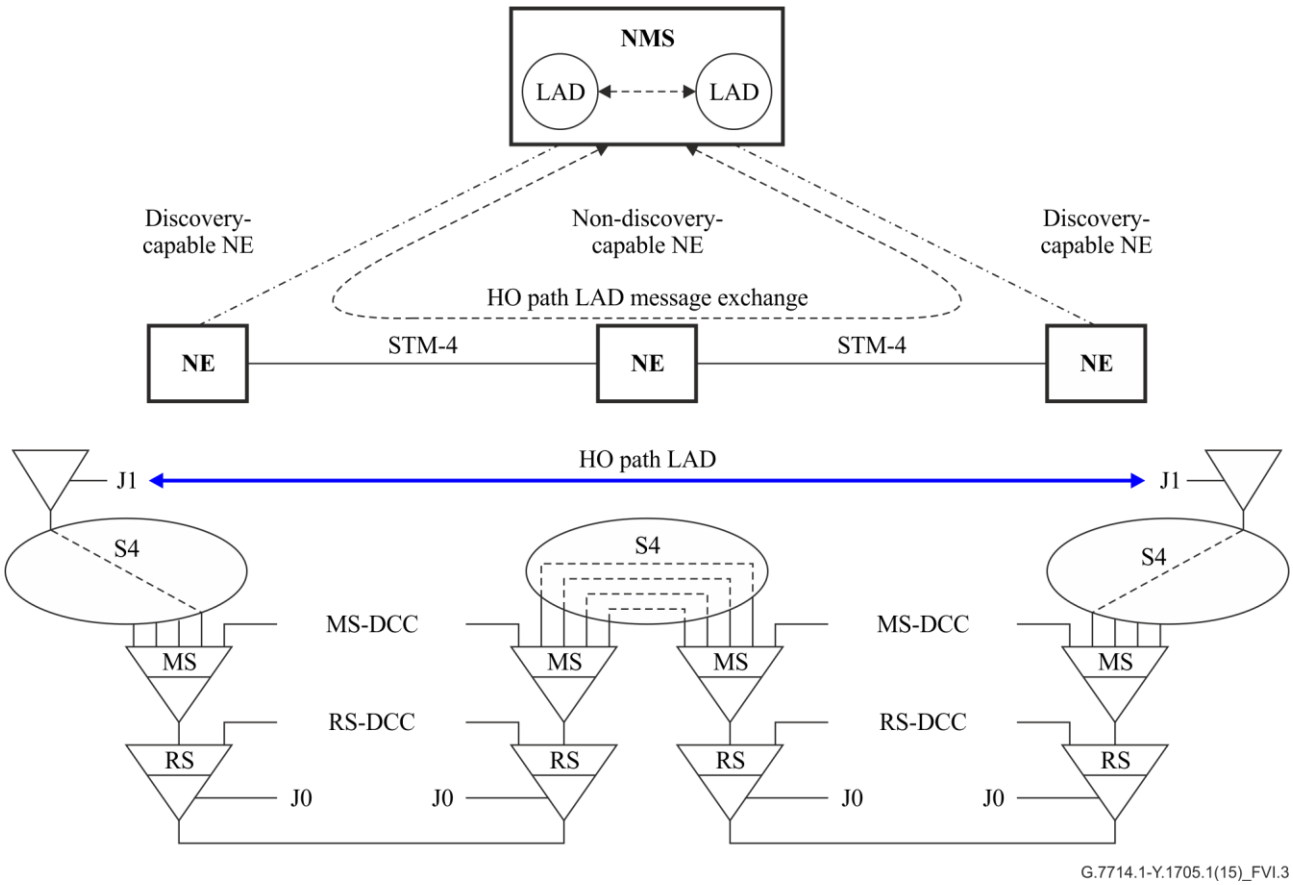

Figure VI.3 – Immediate discovery-capable neighbours at HO path layer – LAD done by NMS

VI.3.2.2 Auto-discovery between a LAD-capable NE and a non-LAD-capable NE

In this case, it is assumed that the non-LAD-capable NE terminates the layer being discovered (see Figure VI.4). In such a case, LAD cannot be performed at that specific layer, since the DMs sent by the LAD-capable NE are not understood by the non-LAD-capable NE. In such a scenario, it is important that the non-LAD-capable NE does not generate alarms and, more important, not perform consequent actions that could unnecessarily disrupt service. One possible means for the network operator to avoid such alarms and consequent actions is to disable the transmission of DMs at the LAD-capable NE or to obey the guidelines described in clause VI.4.

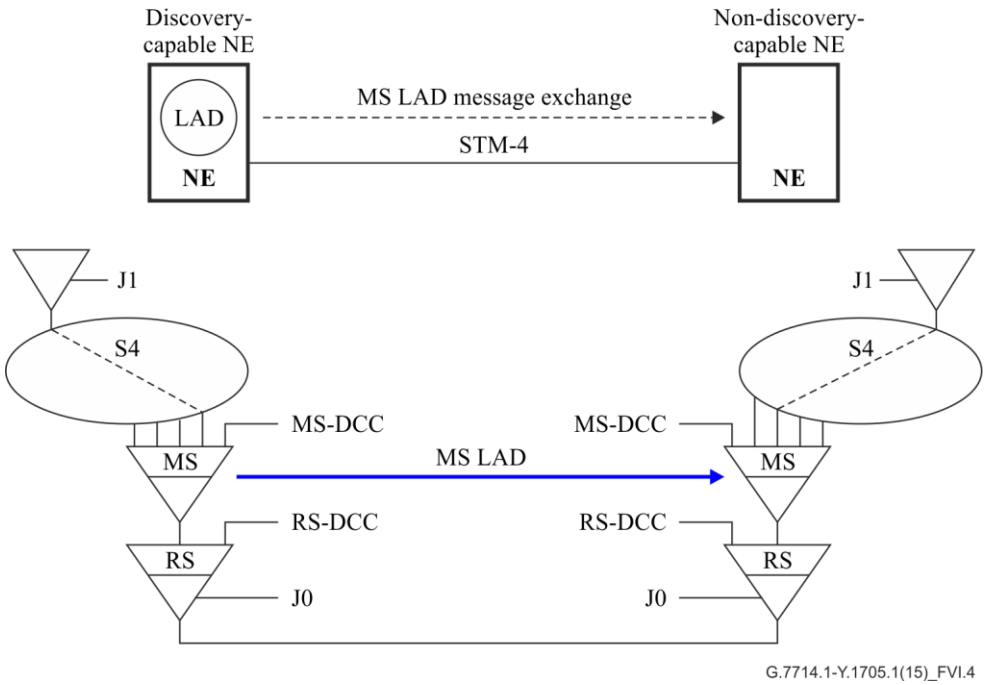

Figure VI.4 – Discovery-capable NE trying to discover a non-discovery-capable NE

VI.4 Guidelines for mechanisms and procedures

This clause provides guidelines on the usage of the trail trace (J0, J1 and J2) and ECC (MS DCC or RS DCC) mechanisms for LAD for the various use cases and scenarios described in clause VI.3.

VI.4.1 ECC-based LAD

Auto-discovery using the DCC is a viable option when the DCC is available on the synchronous transport module-n (STM-n) interface that needs to be discovered. The DCC provides a packet-based interface; its use for LAD is not affected by the service state (in-service, out-of-service, pre-service) of the given STM-n interface it is associated with. The LAD process making use of the DCC does not have any impact on the traffic on the STM-n interface. However, there are a number of use cases where the DCC may not be sufficient for LAD, based on DCC availability given the DCN deployment scenarios described in clause IV.4.1.1.

VI.4.1.1 DCN deployment scenarios impacting the availability of DCCs

There are two scenarios which affect the deployment of DCC-based LAD messages.

a) No DCC connectivity [e.g., central office (CO) LAN supporting the DCN].

In this scenario, there is no DCC connectivity between the add-drop multiplexers (ADMs) and the digital cross connect (DXC) in the CO. Instead, as shown in Figure VI.5, the CO LAN is used to carry the management communication between the NEs in the CO. Although there is connectivity (e.g., STM-n) between the ADMs and the DXC, the management communication does not follow the same topology as these optical connections that contain the DCCs. The DXC could be used to interconnect low-speed optical interfaces between ADMs within a CO – and therefore the DCCs on these low-speed optical interfaces are not available for auto-discovery.

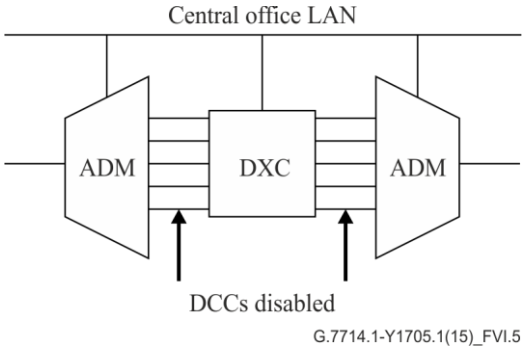

Figure VI.5 – Central office with disabled DCC connectivity

b) Limited DCC availability or DCCs not enabled on all parallel interfaces between two NEs.

In this scenario, as depicted in Figure VI.6, there may be limited or no DCC availability for management communication between NEs – e.g., due to disabling of DCCs, or limited DCC resources. This could occur between multiple carriers, at a customer-to-carrier interface, or where only out-of-band connectivity is available between the NEs – and therefore the DCC is not available for auto-discovery. It is also possible that there are multiple parallel optical interfaces connecting the two NEs. However, the DCCs on only one link or a small subset of links may be enabled. This may be the case for several administrative reasons, e.g.:

– DCC processing not supported for all interfaces;

– configuration decision (e.g., in case of multiple parallel links, the DCCs are only enabled on some of them, since the capacity of a single DCC may be sufficient for management communication between the two NEs);

– policy decisions in the case of connectivity of NEs between different administrative domains.

In all these cases, it may not be possible to perform LAD on every link using DCC, because some of the links may not have the DCC enabled.

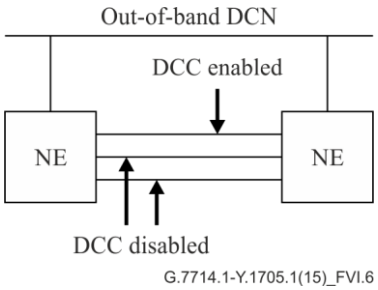

Figure VI.6 – Central office with DCC enabled on only one link or using an out-of-band DCN

VI.4.2 Trail-trace-based LAD

Case A (using J0, J1 and J2 bytes)

The trail-trace bytes can be utilized for Type 1 LAD, which allows the client layer LCs to be inferred from the discovered server layer trail as depicted in Figure 1. Depending on the configuration of the TT functions involved in the LAD process, some behavioural issues could arise. In particular, traffic impact has to be avoided while the interfaces are in the in-service state and are carrying traffic. These scenarios where such behavioural issues might occur are addressed in this clause and are discussed in detail in the following. Moreover, application and configuration guidelines are provided in order to avoid traffic impacts.

Case B (using the TTI field of the TCM sublayer 6)

If intermediate equipment such as wavelength division multiplexing (WDM) transponders terminate the OTU layer, the TTI field of TCM sublayer 6 is used for auto-discovery in OTN networks. This scenario is illustrated in Figure VI.7.

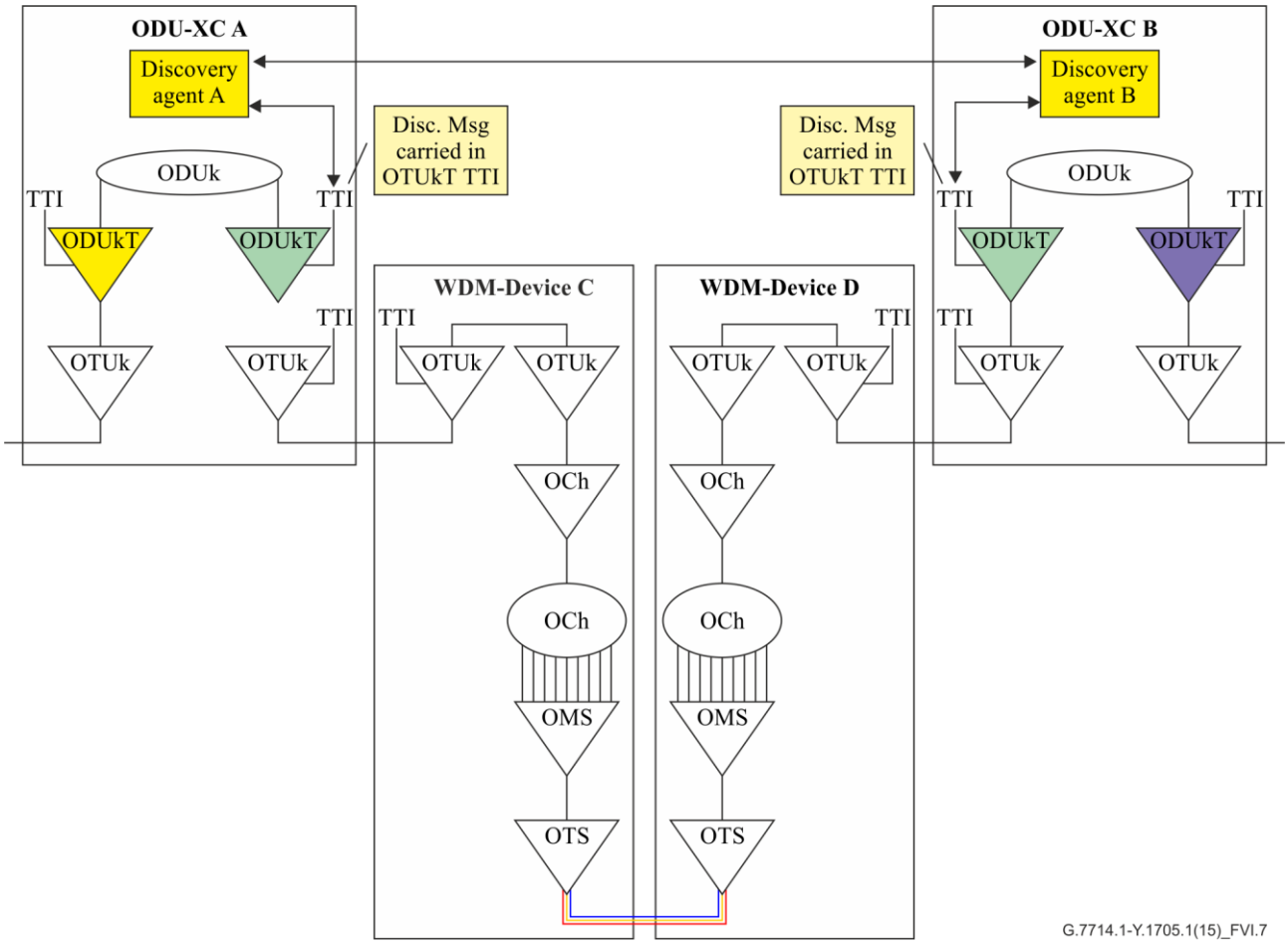

Figure VI.7 – Example of ODUkT TTI (SAPI sub-field)-based LAD for OTN with WDM equipment

VI.4.2.1 Pre-service and out-of-service cases

The use of the trail-trace bytes for LAD does not cause any behavioural issues as long as the interface is in a pre-service or out-of-service state because no traffic is being carried over it.

VI.4.2.2 In-service cases

It should be noted that discovery enabling or disabling capability is provided at each link end at a specific layer independent of the remote end. Note that the discovery process is only permitted to change (provision) the TTI when the discovery process is enabled.

Usage of the trail-trace bytes as defined in [b-ITU-T G.707] allow transmission and reception of APIs so that the receiving terminal can verify its continued connection to the intended transmitter. The formats used for LAD are different to formats commonly used for pre-existing applications. It is expected that new equipment should be able to recognize this usage. In order to avoid undesired trace identifier mismatch (TIM) alarms for some legacy equipment, the discovery-capable NE should not change the TTI (i.e., should disable auto-discovery) at its trail end when there is a non-discovery capable NE at the other end. Discovery should also be disabled when the trail includes monitors that are monitoring the TTI and are unable to distinguish DMs.

Note that the discovery process could be performed in a management system, thereby making an NE discovery capable.

For some existing equipment, use of the trail-trace bytes for discovery may raise alarms, and if the consequent action [alarm indication signal (AIS) insertion] is not disabled, may cause traffic loss. Therefore, TT points that allow trail-trace-based discovery should set TIMAISdis=true to prevent the insertion of an AIS when the TTI does not match. In national networks where TIMAISdis is required to be always false (see [b-ITU-T G.806]), trail-trace-based discovery should not be performed.

If dTIM detection is enabled, the LAD process can use the MI_cTIM as a notification that the trail trace has changed (MI_AcTI).

Non-intrusive monitoring

Non-intrusive monitor functions (see [b-ITU-T G.783]) may observe the trail trace. If the non-intrusive monitor function is not aware of the use of trail-trace for discovery, unexpected changes in TTI will be observed.

VI.4.3 Inter-carrier, user-provider implications

The LAD process can be enabled or disabled on each interface. This allows the network operators to configure the interfaces according to their policy.

Bibliography

[b-ITU-T G.707] Recommendation ITU-T G.707/Y.1322 (2007), *Network node interface for the synchronous digital hierarchy (SDH).*

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

[b-ITU-T G.784] Recommendation ITU-T G.784 (2008), *Management aspects of Synchronous digital hierarchy (SDH) transport network elements*.

[b-ITU-T G.806] Recommendation ITU-T G.806 (2012), *Characteristics of transport equipment – Description methodology and generic functionality*.

[b-ISO 3166] International Standard 3166 (all parts), *Codes for the representation of names of countries and their subdivisions.*

[b-IETF RFC 1661] IETF RFC 1661 (1994), *The point-to-point protocol*.

[b-IETF RFC 1930] IETF RFC 1930 (1996), *Guidelines for creation, selection, and registration of an Autonomous System (AS)*.

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	**Y.1700–Y.1799**
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	Y.3500–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2017