International Telecommunication Union

ITU-T G.7714.1/Y.1705.1

(09/2010)

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 AND NEXT-GENERATION NETWORKS

Internet protocol aspects – Operation, administration and maintenance

Protocol for automatic discovery in SDH and OTN 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 SDH and OTN networks

Summary

Recommendation ITU-T G.7714.1/Y.1705.1 describes the methods, procedures and transport plane mechanisms for discovering layer adjacency for automatically switched optical networks (ASON) according to the requirements of Recommendation ITU-T G.7714/Y.1705 and architecture of Recommendation ITU-T G.8080/Y.1304. Layer adjacency discovery describes the process of discovering the 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 and transport entity capability exchange, etc., will be addressed in future Recommendations.

This 2010 revision includes: Appendix VI (Usage of the different discovery mechanisms) from ITU-T G.7714.1/Y.1705.1 (2003) Amendment 1 (2006); clause 5.1 (CP-CP connectivity relationships determination); clause 9.3 (Interoperable solution when using the ECC-based mechanism) and the inclusion of ODU TCM layer discovery in clause 6.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.7714.1/Y.1705.1	2003-04-22	15
1.1	ITU-T G.7714.1/Y.1705.1 (2003) Amend. 1	2006-02-17	15
2.0	ITU-T G.7714.1/Y.1705.1	2010-09-06	15

Keywords

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

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/.

CONTENTS

		Page
1	Scope	1
2	References	1
3	Definitions	2
	3.1 Terms defined elsewhere	2
	3.2 Terms defined in this Recommendation	3
4	Abbreviations and acronyms	3
5	Discovery methodology	4
	5.1 CP-CP connectivity relationships determination	5
6	Mechanisms for layer adjacency discovery	5
7	Attributes used in layer adjacency	6
8	Layer adjacency based on trail trace string	7
	8.1 Discovery message formats	9
9	Layer adjacency based on embedded control channel messages	11
	9.1 LAPD-based mechanism	11
	9.2 PPP-based mechanism	12
	9.3 Interoperable solution when using the ECC-based mechanism	12
10	Procedures	13
11	Discovery response message	13
	11.1 Miswiring detection	14
	11.2 Misconnection detection	14
Appendix I – Implementation example of discovery process		15
	I.1 Layer adjacency discovery information flow	15
Appendix II – Miswiring detection		17
	II.1 Auto discovery procedures	17
	II.2 Example: Interaction between two DAs using different discovery message formats	21
Appendix III – Example of discovery response message using generalized MPLS-based mechanism		23
Appendix IV – Layer adjacency discovery implementation examples		25
Appendix V – In-band message encoding example		26
Appendix VI – Usage of the different discovery mechanisms		28
	VI.1 Introduction	28
	VI.2 Categories of Type 1 layer adjacency discovery use cases	28
	VI.3 Use cases and scenarios	29
	VI.4 Guidelines for mechanisms and procedures	31
Bibliography		35

Recommendation ITU-T G.7714.1/Y.1705.1

Protocol for automatic discovery in SDH and OTN networks

1 Scope

This Recommendation describes the methods, procedures and transport plane mechanisms for discovering layer adjacency for automatically switched optical networks (ASON) according to the requirements of [ITU-T G.7714] and architecture of [ITU-T G.8080]. Layer adjacency discovery describes the process of discovering the link connection end-point relationships and verifying their connectivity. The term "discovery" is used throughout this 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 for obtaining physical media adjacency discovery and 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 equipments 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.707] Recommendation ITU-T G.707/Y.1322 (2000), *Network node interface for the synchronous digital hierarchy (SDH)*.

[ITU-T G.709] Recommendation ITU-T G.709/Y.1331 (2009), *Interfaces for the Optical Transport Network (OTN)*.

[ITU-T G.774] Recommendation ITU-T G.774 (2001), *Synchronous digital hierarchy (SDH) − Management information model for the network element view*.

[ITU-T G.784] Recommendation ITU-T G.784 (1999), *Synchronous digital hierarchy (SDH) management*.

[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.872] Recommendation ITU-T G.872 (2001), *Architecture of optical transport networks*.

[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.8080] Recommendation ITU-T G.8080/Y.1304 (2006), *Architecture for the automatically switched optical network (ASON)*, plus Amendment 2 (2010).

[ITU-T G.8081] Recommendation ITU-T G.8081/Y.1353 (2010), *Terms and definitions for Automatically Switched Optical Networks (ASON)*.

[ITU-T M.3000] Recommendation ITU-T M.3000 (2000), *Overview of TMN Recommendations*.

[ITU-T M.3010] Recommendation ITU-T M.3010 (2000), *Principles for a telecommunications management network*.

[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*.

[IETF RFC 1570] IETF RFC 1570 (1994), *PPP LCP Extensions*. <http://www.ietf.org/rfc/rfc1570.txt?number=1570>

[IETF RFC 1661] IETF RFC 1661 (1994), *The Point-to-Point Protocol*. <http://www.ietf.org/rfc/rfc1661.txt?number=1661>

[IETF RFC 1662] IETF RFC 1662 (1994), *PPP in HDLC-like Framing*. <http://www.ietf.org/rfc/rfc1662.txt?number=1662>

[IETF RFC 2045] IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. <http://www.ietf.org/rfc/rfc2045.txt?number=2045>

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 The following terms are defined in [ITU-T G.805]:

– adaptation;
– link;
– link connection;
– subnetwork connection (SNC);
– trail.

3.1.2 The following term is defined in [ITU-T G.7714]:

– transport entity capability exchange (TCE)

3.1.3 The following terms are defined in [ITU-T G.8081]:

– discovery agent;
– local TCP-ID;
– local CP-ID;
– policy;
– termination and adaptation performer.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AITS	Acknowledged Information Transfer Service
API	Access Point Identifier
ASON	Automatically Switched Optical Network
CP	Connection Point
DA	Discovery Agent (Also known in [ITU-T G.7714] as a type of control entity)
DA ID	Discovery Agent Identifier
DCC	Data Communications Channel (ECC in SDH)
DCN	Data Communications Network
DM	Discovery Message
ECC	Embedded Control Channel (See [ITU-T G.7712])
GCC	General Communications Channel (ECC in OTN)
GMPLS	Generalized Multi-Protocol Label Signalling
HDLC	High-level Data Link Control (in SDH)
HOVC	Higher Order Virtual Container (in SDH)
LAD	Layer Adjacency Discovery
LAPD	Link Access Procedure D-channel
LCP	Link Control Protocol
LLCF	Link Layer Convergence Function
LOVC	Lower Order Virtual Container (in SDH)
LRM	Link Resource Manager
MS	Multiplex Section
ODUk	Optical Channel Data Unit-k (in OTN)
ODUkT	Optical Channel Data Unit-k Tandem connection monitoring level (in OTN)
OTN	Optical Transport Network
OTUk	completely standardized Optical Channel Transport Unit-k (in OTN)
PC	Protocol Controller
PM	OTN Path Monitoring byte
PPP	Point-to-Point Protocol
RS	Regenerator Section (in SDH)
RSA	Regenerator Section Adaptation
RST	Regenerator Section Termination
MSA	Multiplex Section Adaptation

MST	Multiplex Section Termination
SAPI	Source Access Point Identifier
SDH	Synchronous Digital Hierarchy
SM	Section Monitoring (in OTN)
SNC	Subnetwork Connection
TAP	Termination and Adaptation Performer
TCM	Tandem Connection Monitoring
TCP	Termination Connection Point
TCP-ID	Termination Connection Point Identifier
TEI	Terminal Endpoint Identifier
TT	Trail Termination
TTI	Trail Trace Identifier
UITS	Unacknowledged Information Transfer Service

5 Discovery methodology

The discovery methodology uses the processes defined in the following clauses to determine the TCP-to-TCP relationship. Once the TCP-to-TCP relationship is determined, the CP-to-CP connectivity relationships are derived using local information. The following two discovery methods are defined:

a) In-service discovery process

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. 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 function.

b) Out-of-service discovery process

In this process a test signal is used 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 in-service discovery process, this approach can only be used if the link connection is not carrying any client traffic.

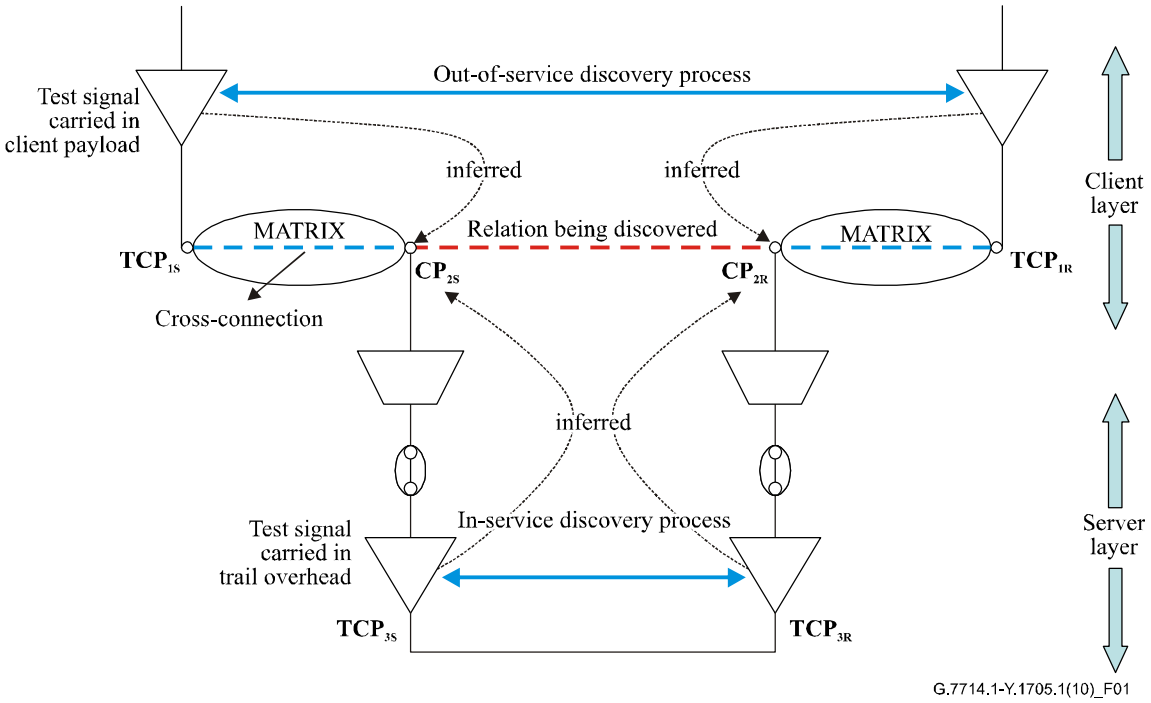

Figure 1 – Entities in the in-service and out-of-service discovery processes

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

5.1 CP-CP connectivity relationships determination

The goal of the layer adjacency discovery process is to discover the relationship between the 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, the CP-to-CP connectivity must be inferred/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 OTN name binding relationships is for further study.

6 Mechanisms for layer adjacency discovery

The mechanisms defined to support the layer adjacency discovery 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 communications channels for the particular layer. The following mechanisms are applicable to SDH layer networks:

- RS layer: Within the RS layer, the J0 section trace and section DCC may be used to support discovery of the RS TCP-to-TCP adjacency.
- MS layer: Within the MS layer, the multiplex section DCC may be used to support discovery of the MS TCP-to-TCP adjacency.

- HOVC layer: Within the 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 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 networks:

- OTUk layer: Within the OTUk layer the SM section monitoring byte and the GCC0 may be used to support discovery of the OTUk adjacency. Specifically, the SAPI subfield within the SM is used to carry the discovery message.
- ODUk layer: Within the ODUk layer the PM path monitoring 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 discovery message.
- ODUkT layer: Within the ODUkT sub-layers, the TTI field may be used to support discovery of the ODUk adjacency. By default, the ODU TCM sub-layer 6 (TCM6) is used for discovery. Specifically, the SAPI subfield within the TTI field is used to carry the discovery message.

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

– **Distinguishing character**

This character "+" is used as the distinguishing character, and its purpose is to avoid the format of SONET/SDH/OTN trail-trace string being confused with some other optional format, e.g., the one specified in Appendix I of [ITU-T G.831].

– **Discovery agent identifier**

The DA ID must be unique within the context of the link being discovered. Two different representations of the DA ID exist: a DA Address and a DA Name.

– **Discovery agent address**

Two attributes are defined to support the DA address:

- **DCN context ID**

  This represents an assigned number (a globally assigned number would be desirable). This attribute may be used in conjunction with the 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.

– **TCP-ID**

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

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 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 defined in [ITU-T G.831]), which specifies two different formats:

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

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

All characters are alphanumeric characters from the T.50 7-bit International Reference Alphabet set (with trailing NULs or SPACEs). Consequently, the second byte is currently 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]-defined formats by placing a non-numerical and non-alphabetic character in the second byte of the message[1]. The remaining 14 bytes are used for carrying 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 discovery message when required. This yields 3 nibbles or 12 bits for every 2 printable characters.

Figure 2 shows the overall J0/J1/J2 or SAPI 16-byte format and depicts how the discovery message (DM) is formatted as compared to the [ITU-T G.831] access point identifier (API).

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

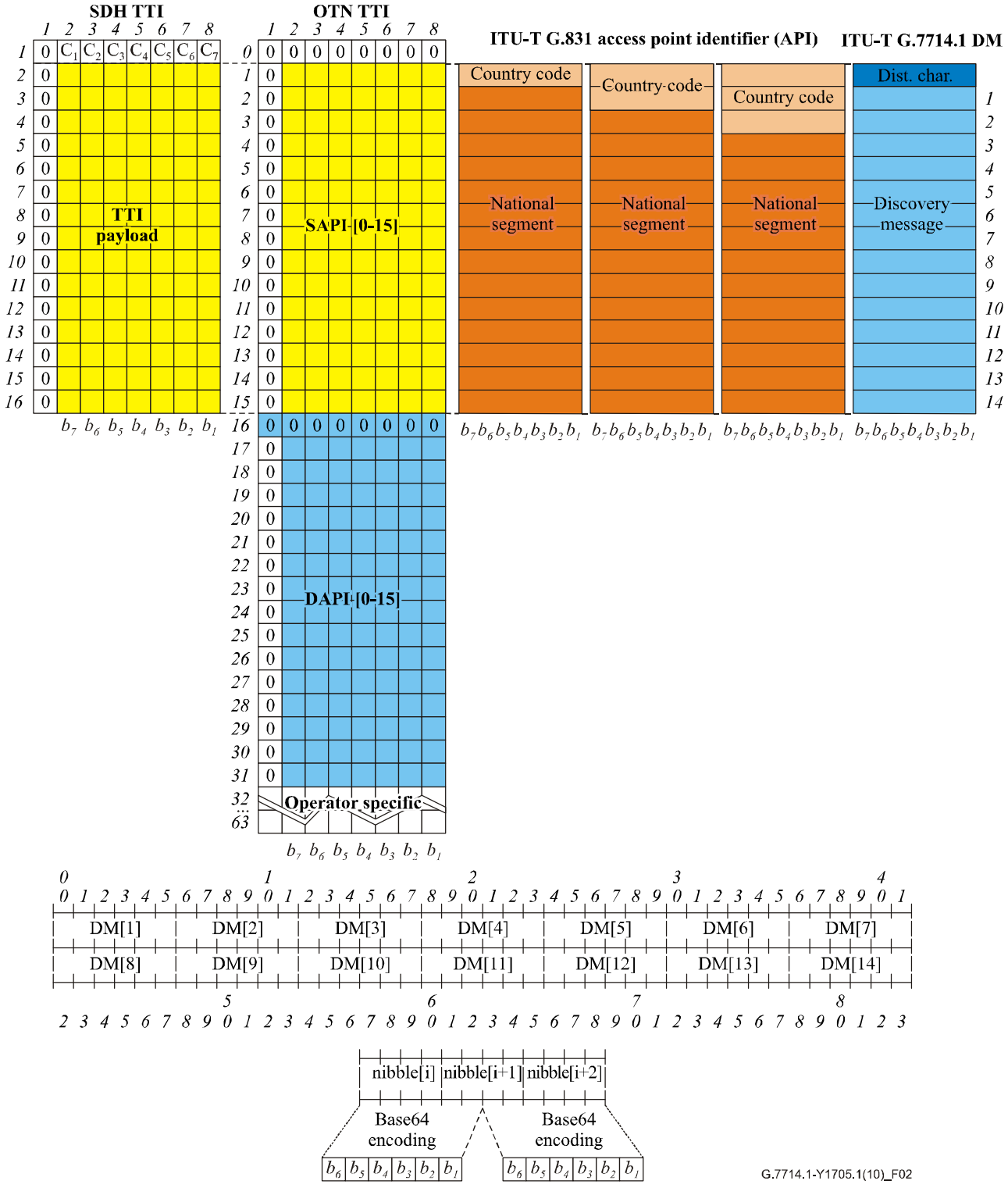

Figure 2 – Auto discovery message (DM) format within the trail trace format

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 discovery message (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 discovery messages as:

- discovery agent 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 discovery message 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.

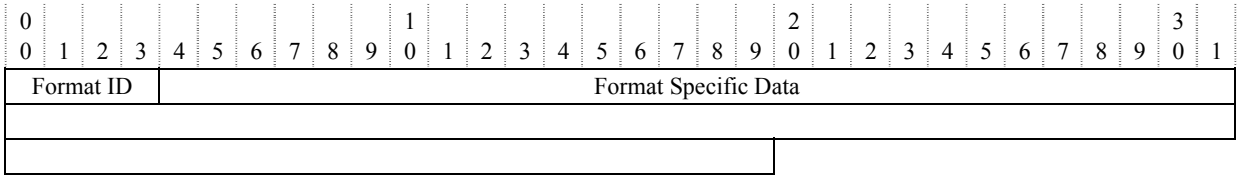

Figure 3 – General discovery message format

This Recommendation defines formats 1, 2 and 3. Additional formats may be provided in the future. If a discovery message 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 which 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 amongst 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 discovery message.

The discovery message 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.

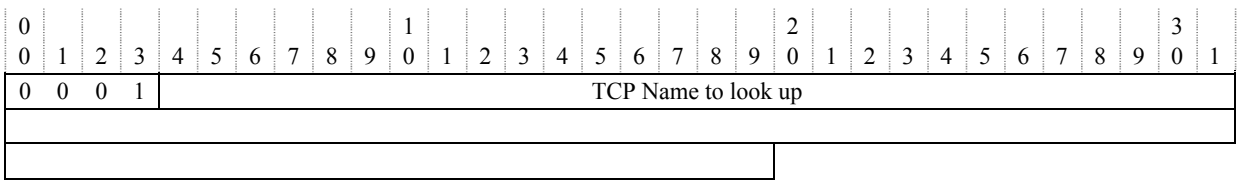

Figure 4 – TCP-ID name message format

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 discovery message. This is called the Local TCP-ID.

The discovery message 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.

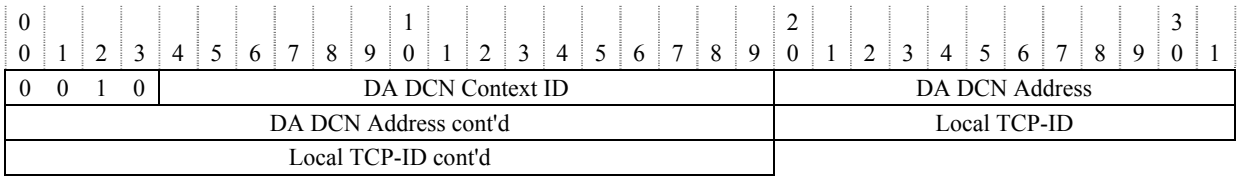

Figure 5 – DA DCN address message format

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

8.1.3 DA DCN name format

Similar to the DCN address format, the DA DCN name format also contains the discovery agent ID and the TCP-ID value. However, unlike the DCN address format, the discovery agent ID 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 which 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 amongst 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 discovery message.

The discovery message 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 address, and 32 bits of TCP-ID.

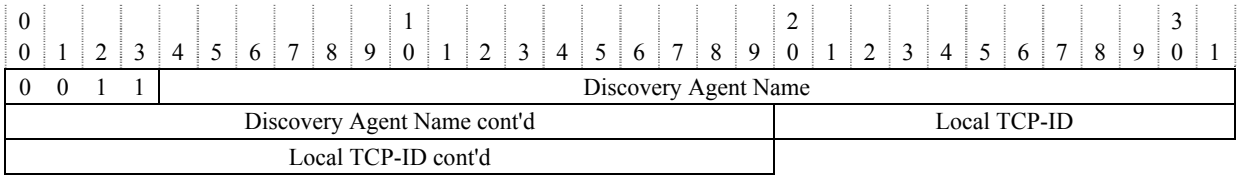

Figure 6 – DA DCN name message format

---

[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 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 discovery message 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.

9 Layer adjacency based on embedded control channel messages

There are two functions required to realize ECC-based layer adjacency discovery: the ECC link layer convergence function and the layer adjacency discovery protocol control function. These messages are applied to the specific layer adjacency which is being discovered. Note that the ECC is provided by a technology-specific mechanism as specified in clause 6.

Figure 7 illustrates the header and data information included by each layer.

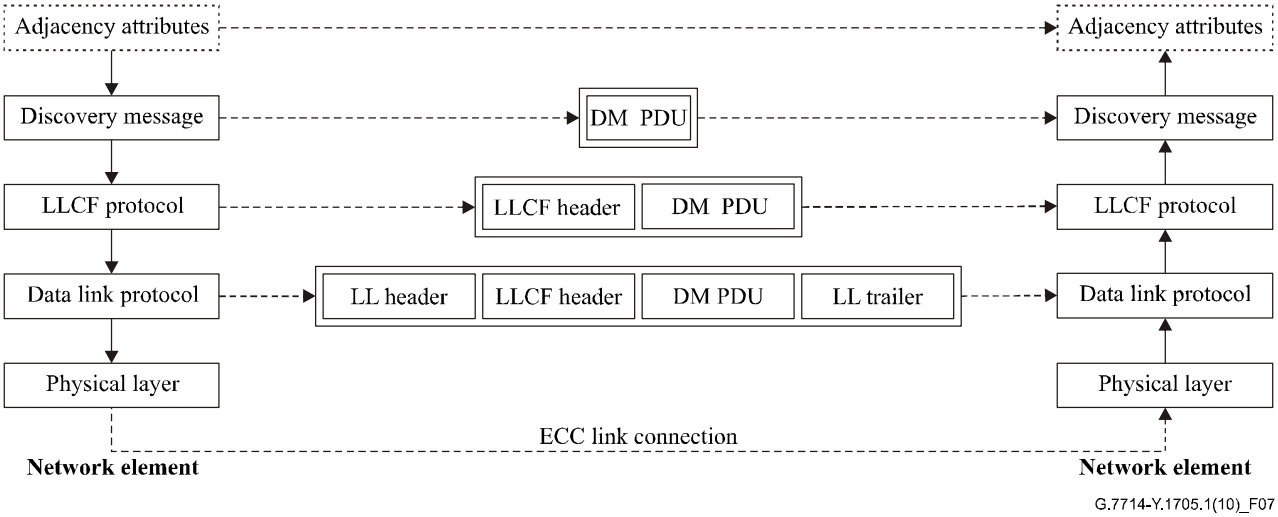

Figure 7 – Layer adjacency discovery functions of importance for ECC messages

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

9.1 LAPD-based mechanism

[ITU-T G.784] requires both AITS and UITS modes to be supported by every network element, so they can be utilized simultaneously over a single ECC channel. The LLCF uses LAPD UITS for transport of layer adjacency discovery 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 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 discovery messages can be sent even in the cases where only unidirectional links exist or there is 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] ([IETF RFC 1661] and [IETF RFC 1662]) using the identification message (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 the ECC-based mechanism

When utilizing HDLC 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 network element 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 network element.

On transmission, the trivial LAPD link layer would place the local network element'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/TEI of 61/0 shall be used for discovery messages. This is different than the 62/0 used by OSI/LAPD over the DCC, as the UI frames for discovery could be mistaken for OSI PDUs.

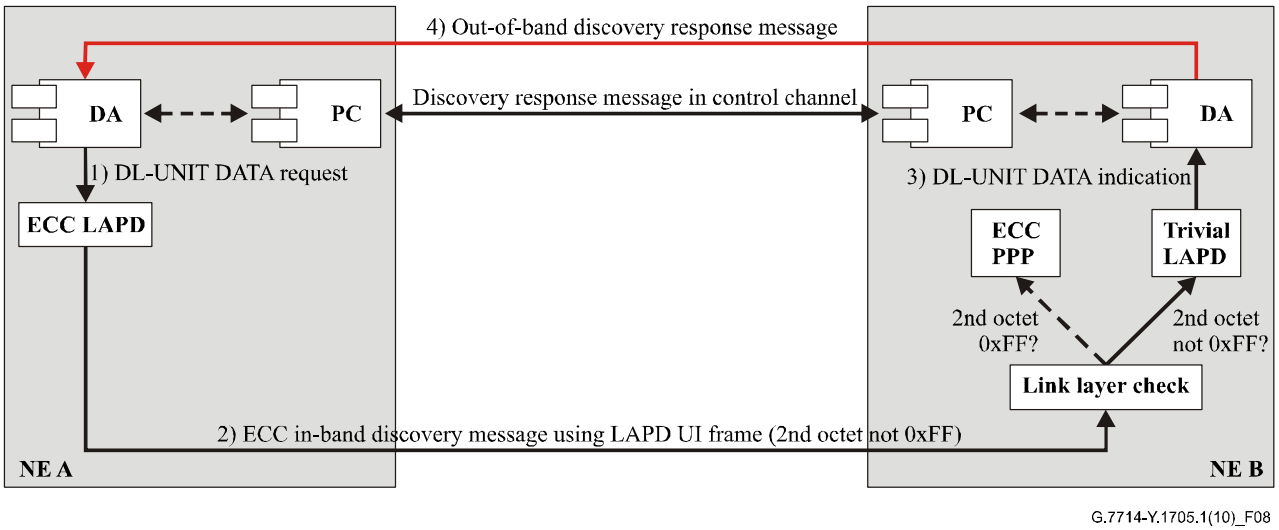

Figure 8 – ECC-based layer adjacency discovery using LAPD

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

10 Procedures

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

1) The initiating discovery agent transmits the discovery message, populating the attributes using one of the formats defined in clause 8.1.

2) Upon receiving an appropriately formatted discovery message, the responding DA checks to determine the applicability of the message, using the distinguishing character to validate the discovery message.

3) After determining that the received message is a discovery message, the responding discovery agent then determines whether the values received are unique with respect to already discovered neighbours.

   If Format ID = 1, a name-server is needed to determine the DA DCN address and TCP-ID.

   If Format ID = 2, then no further address translation is needed.

   If Format ID = 3, then address translation is needed for the DA DCN name.

4) Generate a discovery response message.

11 Discovery response message

When the discovery agent receives the discovery message for the first time, it may notify the originating discovery agent that the message was received on a trail termination associated with a particular TCP. This TCP, called the discovery Sync 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.

Table 1 – Discovery response message attributes

<Received DA DCN ID>	DA DCN ID contained in the received discovery message
<Received TCP-ID>	TCP Identifier contained in the received discovery message
<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 discovery message was received

The Received DA DCN Identifier field shall be included in the discovery response if the following condition is met: the received discovery message includes a DA DCN identifier. If the DA DCN identifier is a DCN name, the name must be 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 identifier was not included in the received discovery message (i.e., the TCP-ID format is in use).

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

The received TCP-ID is the TCP identifier received in the discovery message. The format of the TCP-ID is determined by the format of the discovery message that was received.

The sent Tx TCP-ID is the TCP identifier currently being sent in discovery messages on the discovery Sync TCP. The format of this identifier is determined by the format of the discovery message being sent.

The sent Rx TCP-ID is the TCP identifier for the receive side of the discovery Sync 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 discovery message. If the format of the discovery message 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 discovery message 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 ID. However, if the DA DCN ID received contains a DCN address, then the DCN address may be used directly.

11.1 Miswiring detection

Once a discovery message 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 if a bidirectional link exists. If the TCP-ID corresponding to the remote endpoint of the link connection 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 have been properly wired. This is described in greater detail in Appendix II.

11.2 Misconnection detection

Once a bidirectional link has been discovered, it should be checked against the management-provided policy to determine if correct TCP-link connection endpoints have been correctly connected. If the policy states that the TCP-link connection 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 implementation examples intended to validate the protocol design choices made and specified in the Recommendation.

I.1 Layer adjacency discovery information flow

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

– Layer adjacency discovery;

– transport entity capability exchange.

Completing the layer adjacency discovery process requires a number of functions to interact with one another to identify the TCP link connection. Further, the relationship between the LAD process and the transport entity capability exchange mechanism needs to be described. A sequence diagram detailing the interactions is shown in Figure I.1.

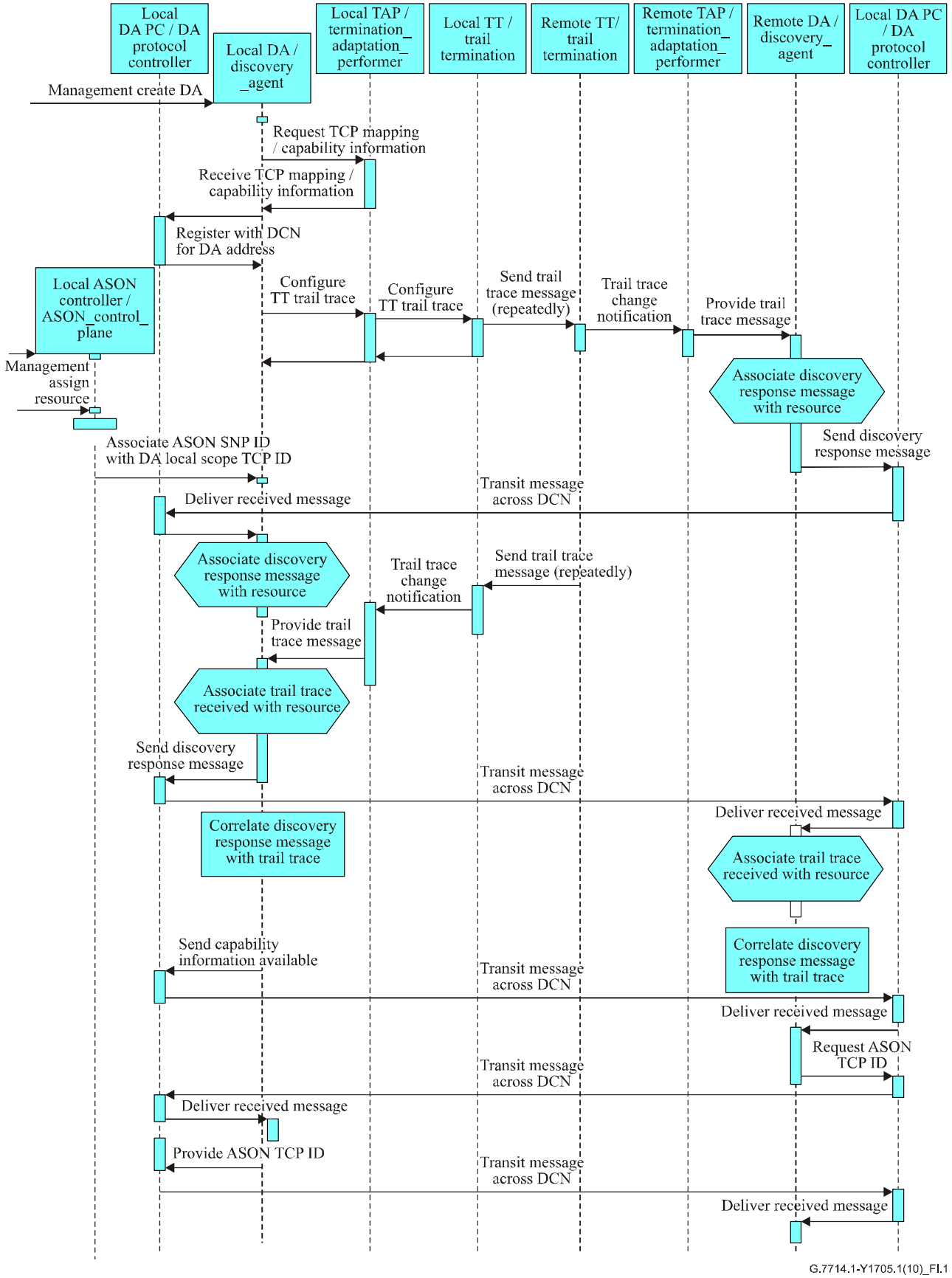

Figure I.1 – Sequence diagram

Appendix II

Miswiring detection

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

This appendix describes how the layer adjacency discovery procedure can detect that the interfaces between two network elements 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 discovery message. This does not, however, preclude other message formats from being used.

II.1 Auto discovery procedures

To automatically discover a layer adjacency between two network elements (e.g., NE A and NE B), both network elements 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 the following two figures (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 discovery message 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 discovery message from NE A;

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

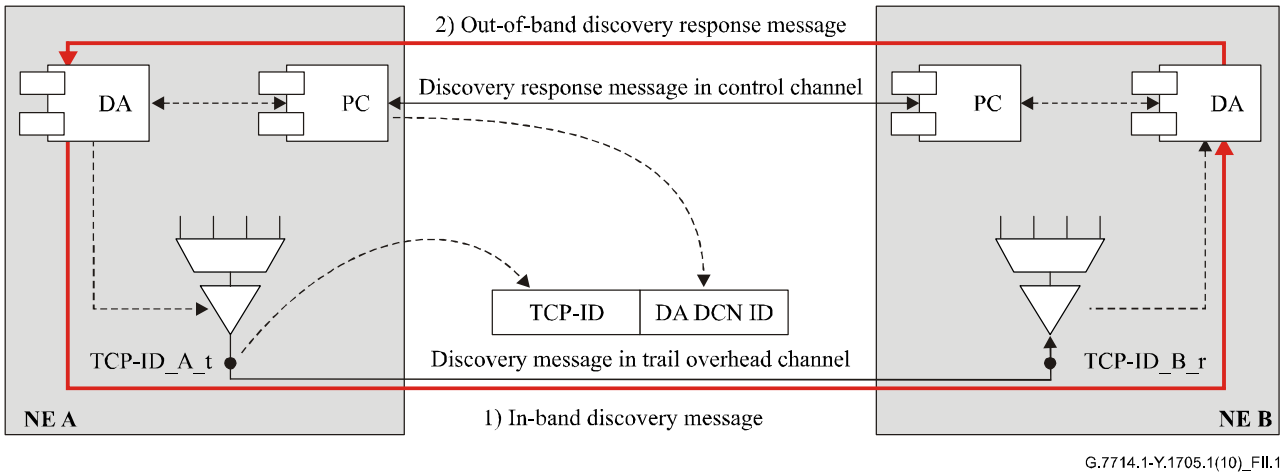

Figure II.1 – Layer adjacency discovery procedure initiated by NE A

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 discovery message 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 discovery message from NE B;

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

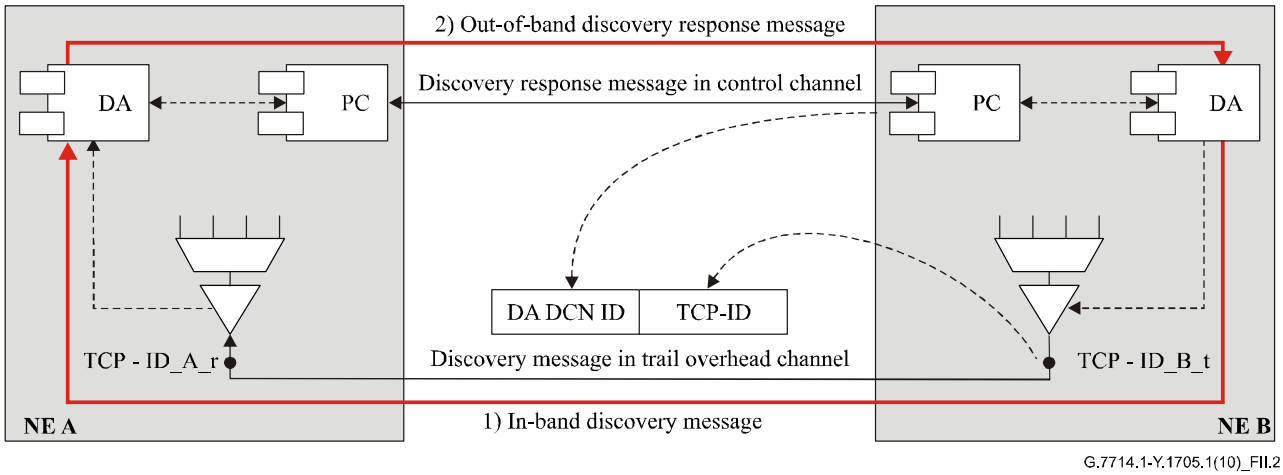

Figure II.2 – Layer adjacency discovery procedure initiated by NE B

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 DA_A's perspective, 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-ID's 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 the transmit and receive directions or whether they both have different values, the DA_A needs to know the binding between the two TCP-IDs on the remote side. In the case where the remote

TCP-IDs in the transmit and receive directions 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. In the case 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 discovery message from DA_B are consistent.

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

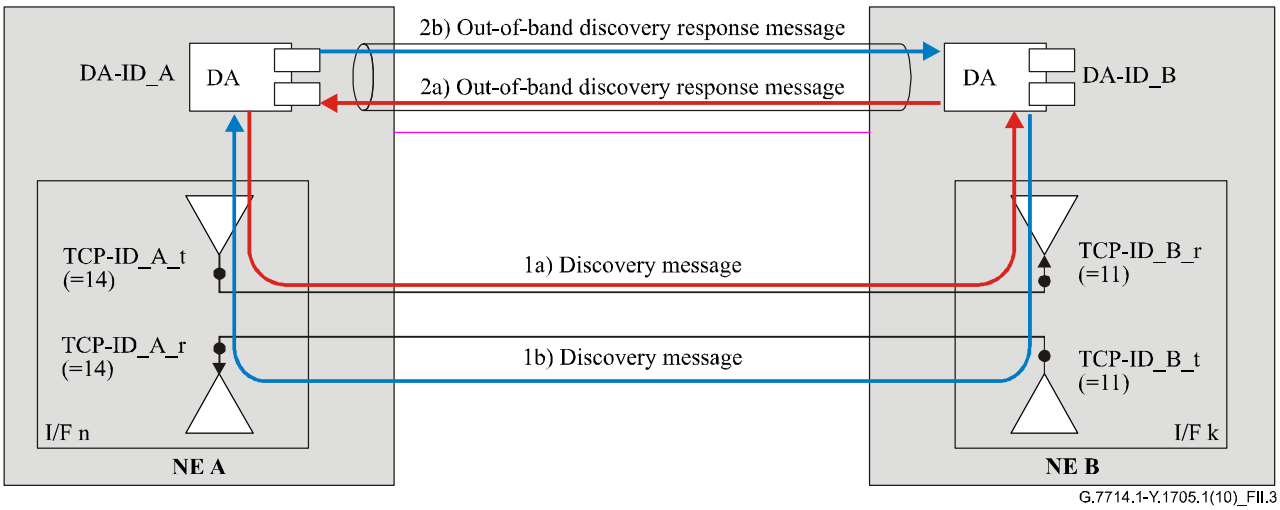

Figure II.3 – Auto discovery in case of correctly wired interfaces

Table II.1 – Example of the two sets of discovery information from DA_A's perspective 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

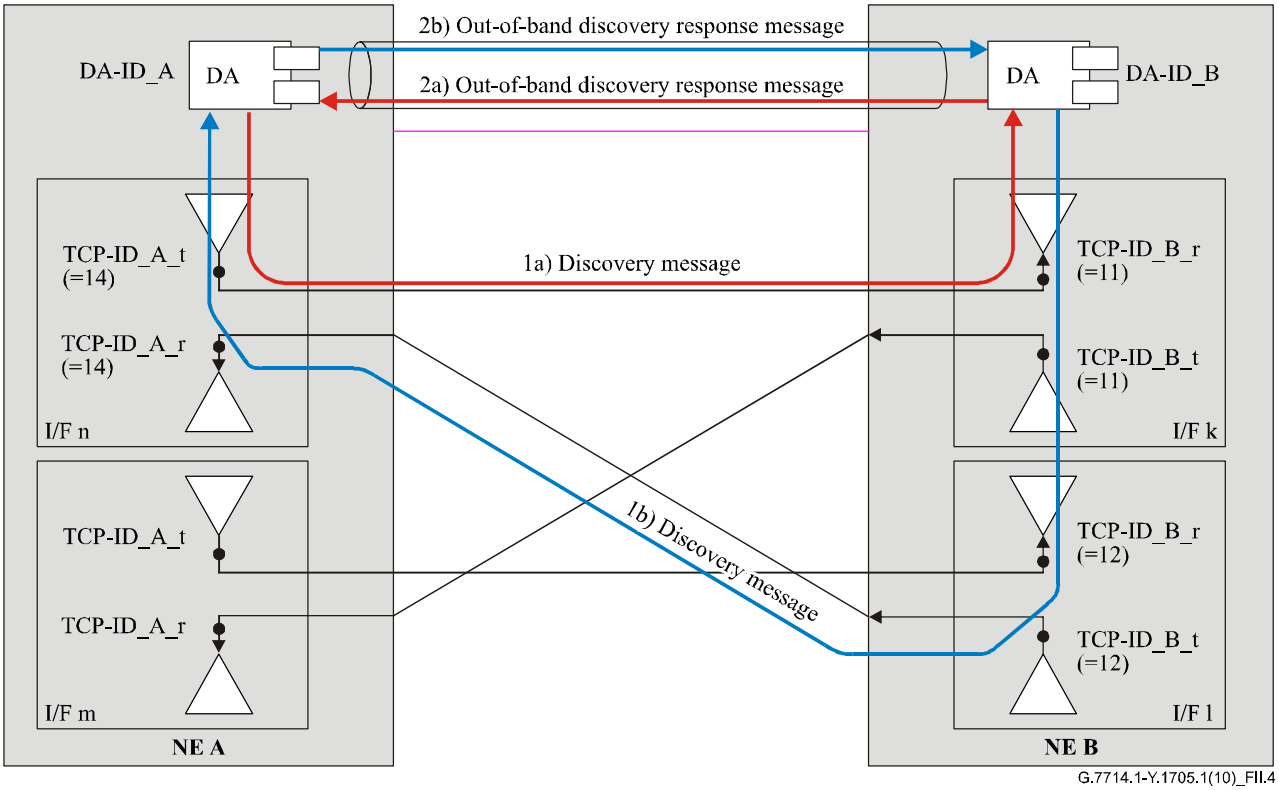

Figure II.4 – Auto discovery in case of miswired interfaces

Table II.2 – Example of the two sets of discovery information from DA_B's perspective 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 are from DA_As perspective belonging 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 are hence related to the same bidirectional TCP. In the next step, the DA-ID consistency check is performed. In the given examples 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, it is checked whether the remote TCP-IDs (from the DA_A perspective) are referring to the same remote TCP. 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 this 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 DA_A perspective) are not referring 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 discovery message). Hence DA_A can indicate the detected miswiring by, e.g., raising and appropriate alarm.

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

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

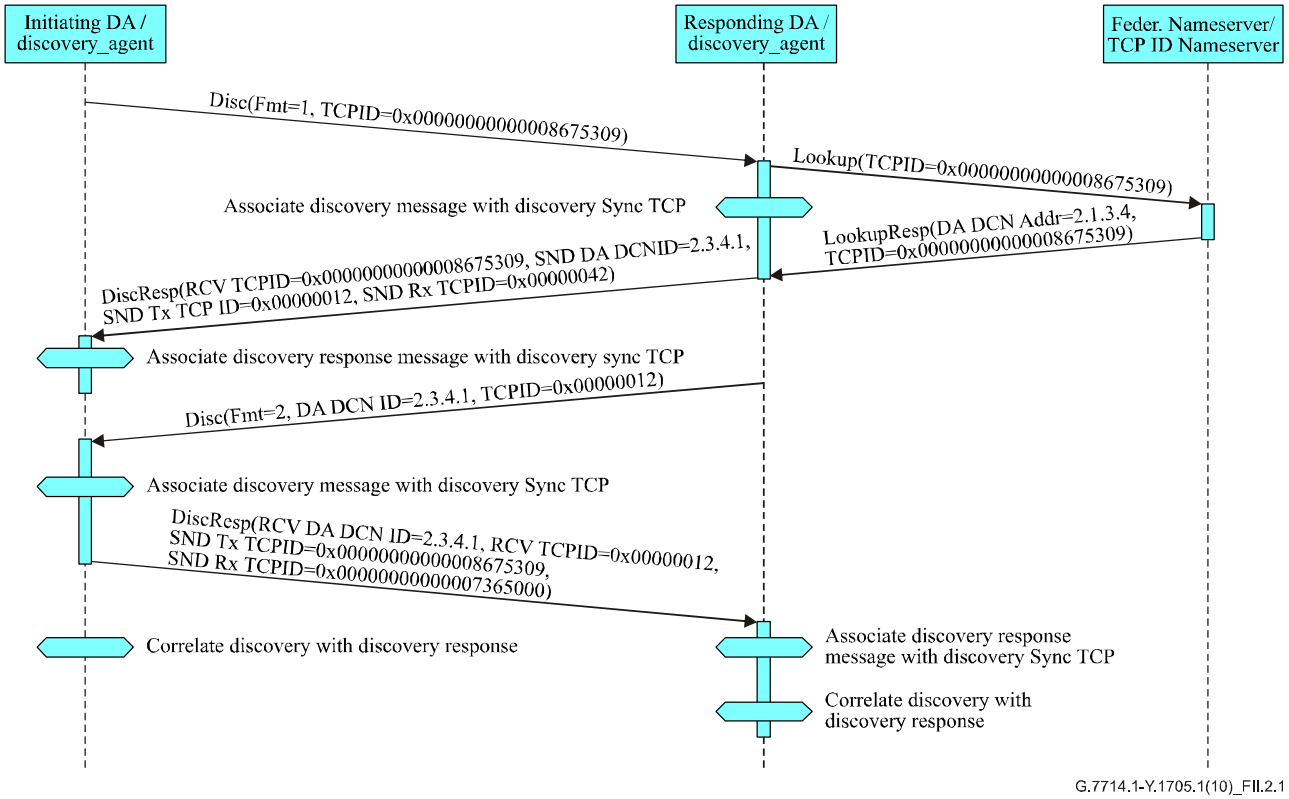

Figure II.5 – Sequence for discovery between two DAs using different message formats

In this example, the initiating discovery agent sends a discovery message in TCP name format. The discovery message *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)* that the discovery message was received on is recorded. This is called the Sync TCP.

The TCP name *(0x0000000000008675309)* in the received discovery message is then translated into the DA DCN address for the initiating discovery agent *(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 discovery message, 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 Sync TCP, as well as the TCP-ID for the Sync TCP *(Rx TCPID=0x0000 0042)*. Once received by the initiating discovery agent, a unidirectional link connection has been identified.

This process is repeated for the opposite direction. However, since this time the DA DCN address format is being used, the discovery message *DISC (Fmt=2, DA DCN Address=2.3.4.1, TCPID=0x0000 0012)* sent includes a DA DCN address and TCP-ID. When the discovery message is received, the Sync TCP it was received on is recorded *(0x0000000000007365000)*. Since the discovery message 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 discovery message received, the current attributes being sent on the Tx TCP *(Fmt=1, TCPID=0x0000000000008675309)* related to the Sync TCP, as well as the TCP-ID of the Sync TCP *(0x0000000000007365000)*. Again, when the discovery response message is received, a unidirectional link connection has been identified.

At this time, it is possible for the discovery response messages to be correlated by each of the ends of the link connection to determine if the bidirectional link has been miswired. Specifically:

	A ≥ B Tx TCPID	**A ≥ B Rx TCPID**	**B ≥ A DA DCN ID**	**B ≥ A Tx TCPID**	**B ≥ A Rx TCPID**
A ≥ B	0000000000008675309		2.3.4.1	0x12	0x42
B ≥ A	0000000000008675309	0000000000007365000	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 transport entity capability exchange to determine the capabilities of the link.

Appendix III

Example of discovery response message using generalized MPLS-based mechanism

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

This appendix illustrates one implementation of layer adjacency discovery as described by this Recommendation, using generalized MPLS (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) Discovery Message" format (as defined 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 defined 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 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 (DA), respectively.

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

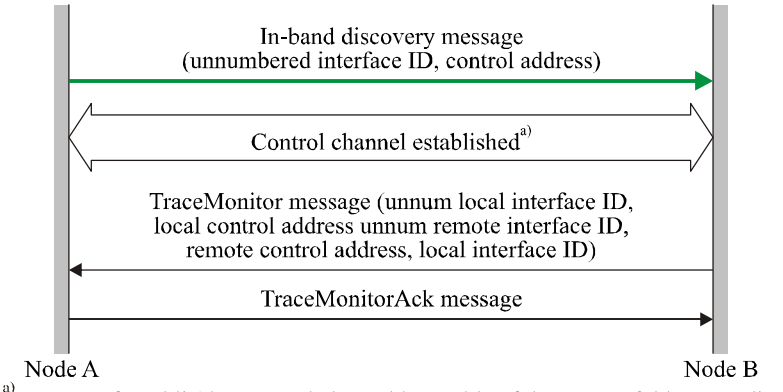

Figure III.1 – Summary of discovery messages used in GMPLS-based implementations

Upon reception of the in-band discovery message from node A's DA, an out-of-band discovery response message, referred to as the (Extended) TraceMonitor message, is sent toward node A's DA over the bidirectional control channel using 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 Node B's DA, a TraceMonitorAck message is sent to Node B's DA 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 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 network element to provision and receive the trail trace message. As an existing text-oriented man-machine language may be reused to provide this interface, the discovery message 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 network element, 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.

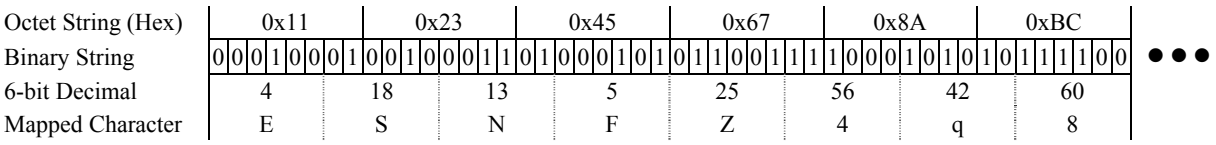

Figure V.1 – Relationship between the discovery message octet string and 6-bit mapped characters

Once the discovery message 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 (see Note)
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 (see Note)
The printable character string after mapping is:	$IAABAgMEASNFZ4_{64}$
The resulting discovery string is:	+IAABAgMEASNFZ4

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 (see Note)
The printable character string after mapping is:	$OYdlQyEKoSNFZ4_{64}$

The resulting discovery string is: +OYdlQyEKoSNFZ4

NOTE – 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.

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 namely 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 link connections 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 soaking interval).

In-service: The entity that is in an in-service state is the trail whose associated client link connections 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 link connections 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., the 'in-service' case. The 'pre-service' and 'out-of-service' use cases, drawn with dotted lines in Figure VI.1 are not further discussed. Moreover, Type 2 LAD is also not considered because the link connections (LCs) cannot be in service (i.e., carry traffic) at the same time Type 2 LAD is applied (see [ITU-T G.7714] for the definition of Type 1 and Type 2 LADs).

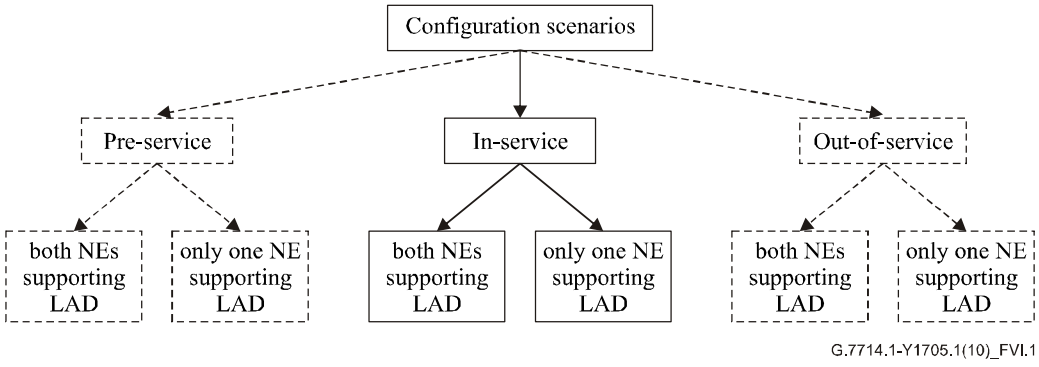

Figure VI.1 – Categorization of discovery scenarios

VI.3 Use cases and scenarios

The various use cases where Type 1 layer adjacency auto-discovery (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 layer adjacency discovery and the entity being discovered. In describing the various scenarios, we broadly distinguish between two cases:

a) the case where all the network elements (NEs) are auto-discovery capable; and

b) the case 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, respectively. For this subset of cases one can use either trail-trace-based or ECC-based discovery messages, provided all the NEs agree on a specific common mechanism.

VI.3.2 All NEs are not auto-discovery capable

In this case some of the NEs within the network are assumed to be unable to understand the auto-discovery messages (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:

- scenarios where both NEs are LAD-capable;
- scenarios 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 layer adjacency discovery 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 regenerator and multiplex sections (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 to run the HO path layer LAD process by proxy for the NEs, as depicted in Figure VI.3.

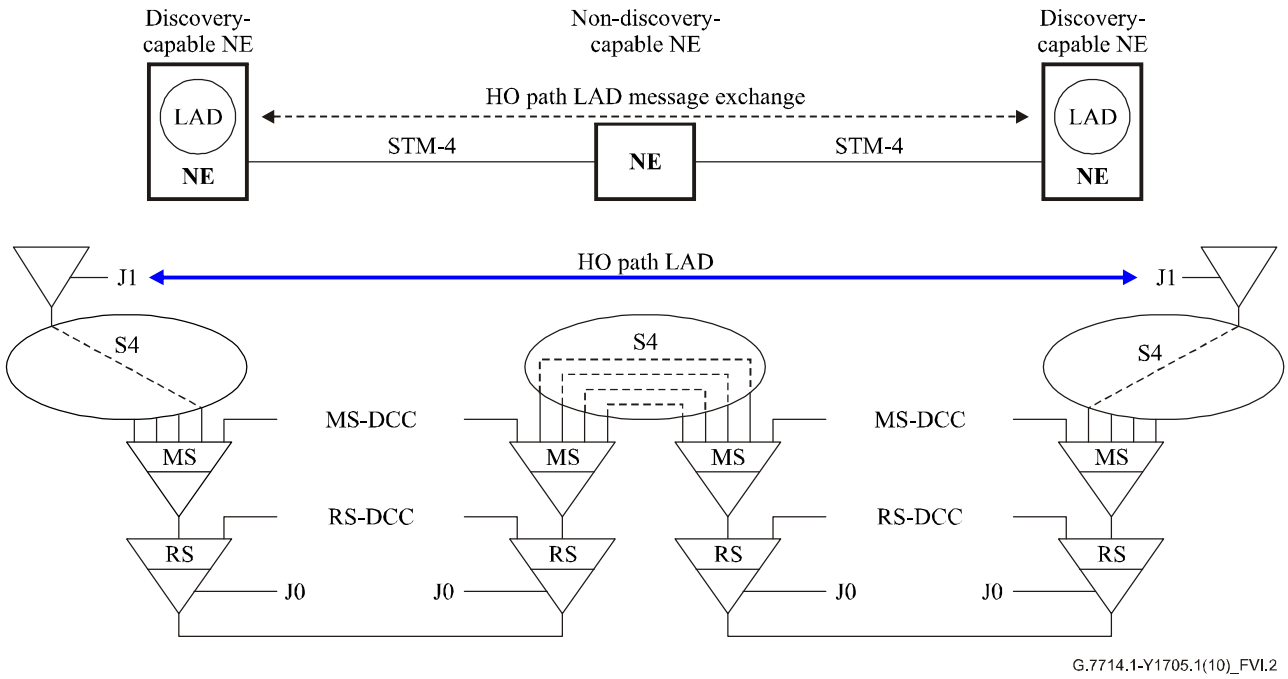

Figure VI.2 – Immediate discovery-capable neighbours at HO path layer – LAD done by NEs

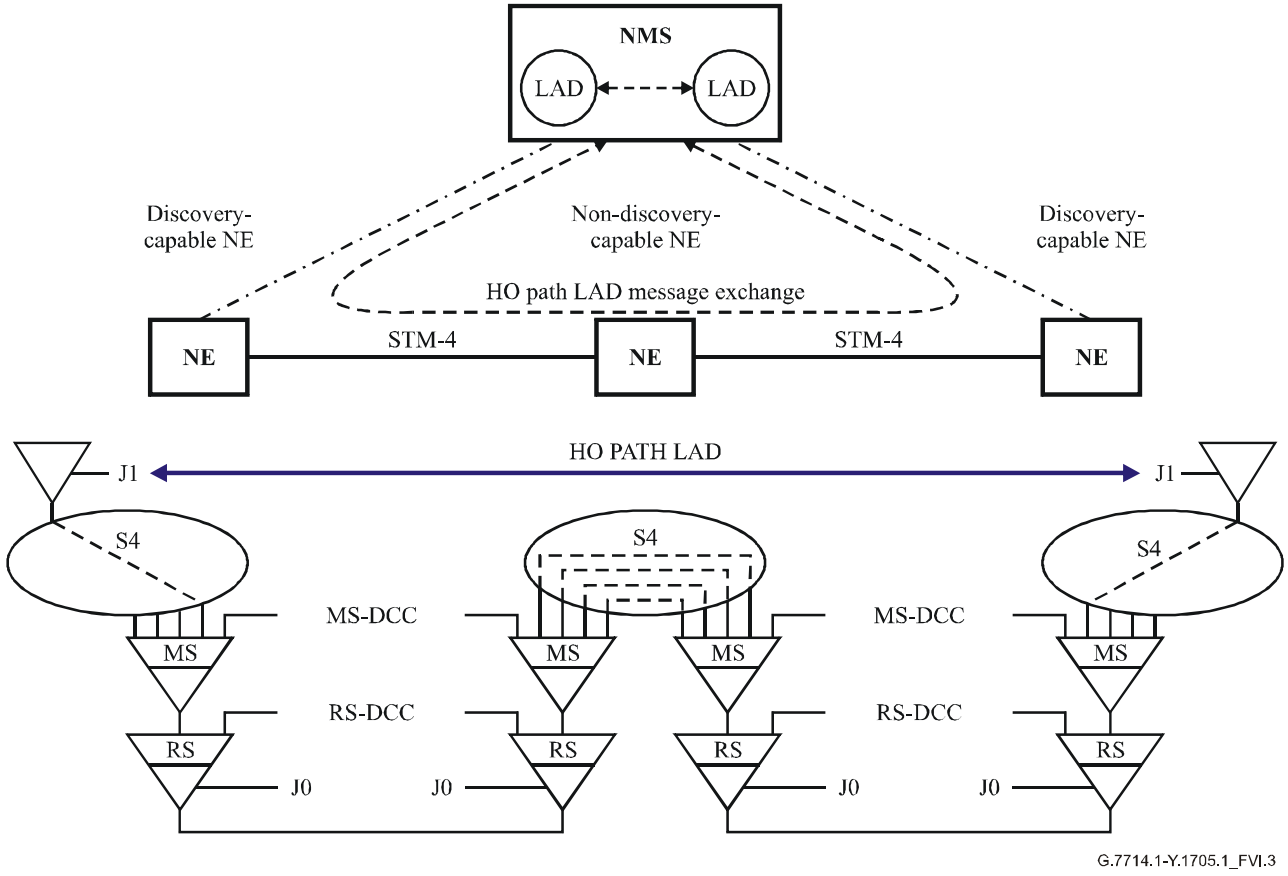

Figure VI.3 – Immediate discovery-capable neighbours at HO path layer – LAD done by the network management system

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

In this case we make the assumption that the non-LAD-capable NE terminates the layer being discovered (see Figure VI.4). In such a case, layer adjacency discovery cannot be performed at that specific layer since the discovery messages 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, does 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 discovery messages at the LAD-capable NE or to obey the guidelines as described in clause VI.4.

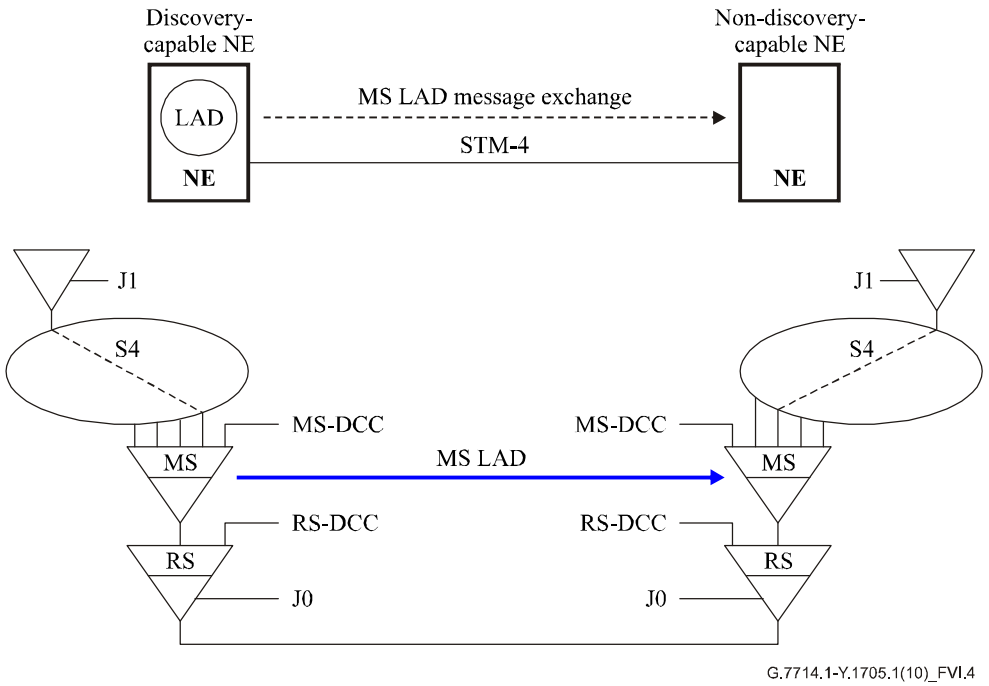

Figure VI.4 – Discovery-capable NE trying to discover a non-discovery-capable NE

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 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 below.

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 LAN supporting the DCN).

In this scenario, there is no DCC connectivity between the add-drop multiplexers (ADMs) and the digital cross connect switch (DXC) in the central office (CO). Instead, as shown in Figure VI.5, the CO LAN is used to carry the management communication between the network elements 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 DCC on these low-speed optical interfaces are not available for auto-discovery.

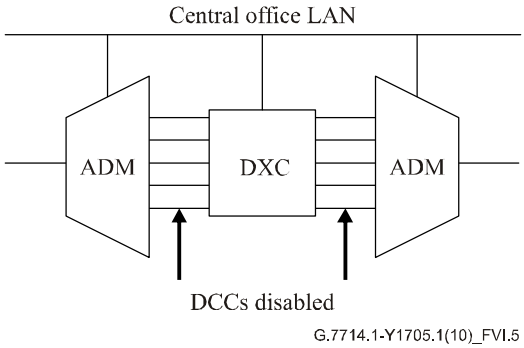

Figure VI.5 – Central office with disabled DCC connectivity

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 network elements – 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.

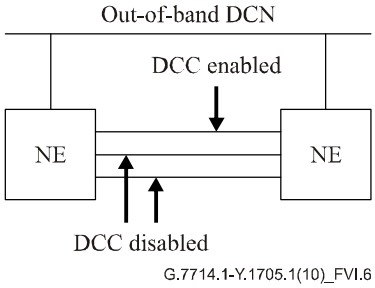

Figure VI.6 – Central office with DCC enabled on only one link or using an out-of-band DCN

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 one to infer the client layer LCs from the discovered server layer trail as depicted in Figure 1. Depending on the configuration of the

trail termination 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 below. Moreover, application and configuration guidelines are provided in order to avoid traffic impacts.

Case B (using the TTI field of the TCM sub-layer 6)

In the cases where intermediate equipment such as WDM transponders terminates the OTU layer, the TTI field of the TCM sub-layer 6 is used for auto-discovery in OTN networks. This scenario is illustrated in Figure VI.7.

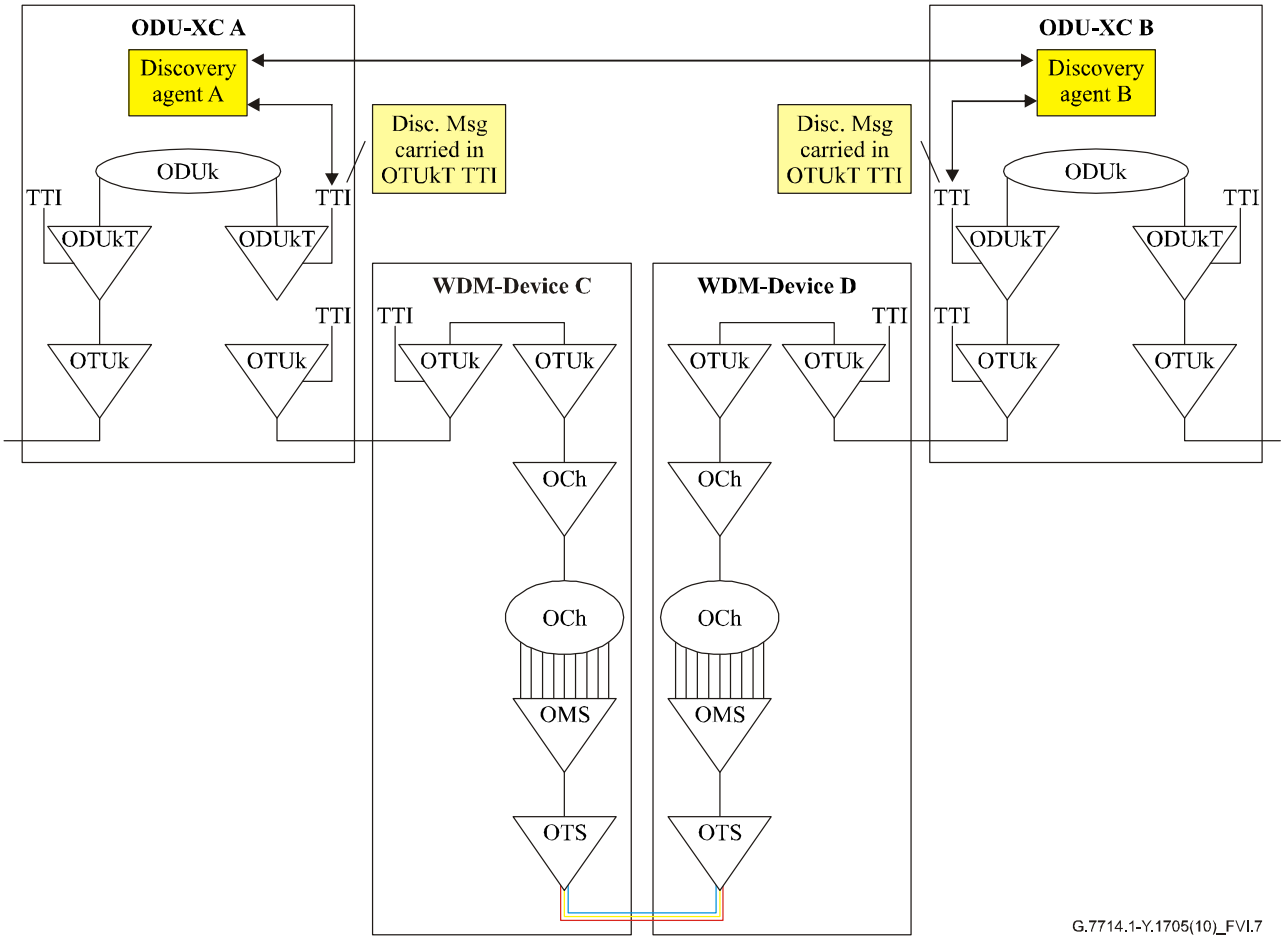

Figure VI.7 – Example of ODUkT TTI (SAPI sub-field)-based LAD for OTN with WDM equipment

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 the discovery enabling/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 [ITU-T G.707] allow transmission and reception of access point identifiers (APIs) so that the receiving terminal can verify its continued connection to the intended transmitter. The formats used for LAD are different from 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 discovery messages.

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 (AIS insertion) is not disabled, may cause traffic loss. Therefore, trail termination points that allow trail-trace-based discovery should set TIMAISdis=true to prevent the insertion of AIS when the trail-trace identifier 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 trail trace information (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.783] Recommendation ITU-T G.783 (2006), *Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks*.

[b-ITU-T G.806] Recommendation ITU-T G.806 (2009), *Characteristics of transport equipment – Description methodology and generic functionality*.

[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 AND NEXT-GENERATION NETWORKS

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
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Future networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999

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	General tariff principles
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	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	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
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 and next-generation networks**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2011