SIF-004-1996
(SIF Document #SIF-TA-9604-034-R3)

SIF Approved Document

TITLE: TARP/500: The TARP and X.500 Directory Services Interworking Specification.

SOURCE: TARP/X.500 Interworking Subgroup

Editors: John MacAuley, NORTEL
Jeff Learman, ONE

Authors: SIF TARP/X.500 Interworking Subgroup

Contributors:

Paul Doolan	OASys Group
Mike Handler	Applied Innovation
Chris Hunt	AT&T Bell Labs
Cypryan Klish	Alcatel
Jeff Learman	ONE
John MacAuley	NORTEL
Susan Marvin	OASys Group
Kaushik Patel	NORTEL
Ron Roman	Bellcore

Contact: Chris Hunt
Lucent Technologies – Bell Labs
Room 3D-328
101 Crawfords Corner Road
Holmdel, NJ 07733

Telephone: 732-949-3554
Fax: 732-949-6092
christopherhunt@lucent.com

DATE: February 17, 1998

ABSTRACT

TL1-based TMN applications require a mechanism that translates between Target Identifiers (TIDs) and Network Service Access Point (NSAP) addresses. Currently there are two existing standardized but incompatible solutions for this translation problem: TARP, The TID Address Resolution Protocol, and the OSI Directory. Since these two mechanisms are incompatible, a means of interworking them is necessary if networks consisting of a mixture of TARP-based NEs and Directory-based NEs are to be managed.

This document has received the approval of the SONET Interoperability Forum (SIF).

April 25, 1996

TABLE OF CONTENTS

1.

1. Introduction

1.1 Scope and Purpose

This specification defines functionality that can facilitate the interoperability of the TID Address Resolution Protocol (TARP) [BEL95], and Directory Services for Telecommunications Management Network (TMN) and Synchronous Optical Network (SONET) [ANS95]. This functionality can be used in a variety of different network topologies.

This recommendation is based on both a [BEL95] compliant TARP protocol, and a [ANS95] compliant X.500 Directory Service.

1.2 Revision History

Revision	Date	Revision Reason
R1	April 25, 1996	SIF approved version of the specification.
R2	June 25, 1996	Changed some incorrect cross-references.
R3	February 17, 1998	Converted document format from Frame to Word

1.3 References

The following standards contain provisions which, through reference in this text, constitute provisions of this SONET Interoperability Forum (SIF) recommendation. At the time of completion, the editions indicated were valid. All standards are subject to revision.

[ANS95] ANSI T1.245-1995. *Directory Service for Telecommunications Management Network (TMN) and Synchronous Optical Network (SONET)*. American National Standard for Telecommunications.

[BEL93] Bellcore TR-NWT-831 (1993). *Operations Technology Generic Requirements: Operation Application Messages - Language for Operations Application Messages.* July, 1993.

[BEL95] Bellcore GR-253-CORE (1995). *Generic Requirements: Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria*, Issue 2. December, 1995.

[ISO87] ISO 8348:1987 (E). *Information processing systems - Data communications - Network service definition*, Addendum 1: Connectionless-mode transmission.

[ISO88] ISO 8348:1988 (E). *Information processing systems - Data communications - Network service definition*, Addendum 2: Network Layer Addressing.

[ISO92] ISO 10589:1992 (E). *Information technology - Telecommunications and information exchange between systems - Intermediate system to Intermediate system intra-domain* *routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service* (ISO 8473).

[ITU93a] ITU-T Rec. X.500 (1993 E) | ISO/IEC 9594-1:1993. *Information technology - Open Systems Interconnection - The Directory - Overview of Concepts, Models and Services.*

[ITU93b] ITU-T Rec. X.501 (1993 E) | ISO/IEC 9594-2:1993. *Information technology - Open Systems Interconnection - The Directory - Models.*

[ITU93c] ITU-T Rec. X.511 (1993 E) | ISO/IEC 9594-3:1993. *Information technology - Open Systems Interconnection - The Directory - Abstract Service Definition.*

[ITU93d] ITU-T Rec. X.521 (1993 E) | ISO/IEC 9594-7:1993. *Information technology - Open Systems Interconnection - The Directory - Selected Object Classes.*

[ITU94] ITU-T Recommendation M.3100 (1994, Revised). *Generic Network Information Model.*

[SIF96a] SIF Contribution SIF-TA-9602-012. *TARP Cleanup #3*. February 20, 1996.

[SIF96b] SIF Contribution SIF-TA-0496-042. *Specification of the DIB Structure to Support OSI Routing Domains*. April 25, 1996.

1.4 Definitions

The following terminology will be used throughout this recommendation to describe different system functionality.

the Directory

A collection of open systems cooperating to provide directory services.

Directory Information Base (DIB)

The set of information managed by the Directory.

Directory Information Tree (DIT)

The structure of information in the DIB. Entries in the DIB are arranged in the form of a tree, where the vertices represent the entries.

Directory System Agent (DSA)

An OSI application process which accesses the Directory on behalf of the user.

Directory User Agent (DUA)

An OSI application process which represents a user in accessing the Directory.

End Network Element

An NE that supports the ES role of the ES-IS routing protocol, and only handles its own traffic.

Gateway Network Element (GNE)

An NE that supports Level 1 and Level 2 IS-IS routing functions, the IS role of the ES-IS protocol, and mapping of TL1 over X.25 connections to TL1 over 7-layer OSI associations.

Intermediate Network Element (INE)

A NE that supports the IS-IS routing protocol and the IS role of the ES-IS protocol. This NE has one or more subtending systems and performs routing for tandem traffic.

Local Naming Context (LNC)

A distinguished name composed of naming components provided to the system's Registration Agent via the namePrefix field of the RRP, or by some locally defined commissioning mechanism. The Local Naming Context identifies the direct superior entry of the system in the DIT.

Network Entity Title (NET)

An OSI Network Address that unambiguously identifies an OSI Network Layer Entity.

Proxy Agent (PA) or Mediation Device (MD)

An agent which provides an application (e.g., management communication) on behalf of another system(s).

Registration Agent (RA)

An application process that receives an RRP message from the Registration Manager signaling it to start automatic registration. The result is that the system's DUA adds the system's administrative information (e.g., name, make, location, etc.) into the DIB.

Registration Manager (RM)

An RM is an application process responsible for contacting the RA using the RRP to start automatic registration of a newly discovered system into the DIB. The RM provides the RA with the appropriate DSA address and LNC.

Registration Request Protocol (RRP)

The RRP is the protocol used between an RM and the RA on each system in the routing area, permitting the systems to add their entries to the DIB.

Routing Area

IS-IS area as defined by IS-IS [ISO92]; often referred to as a Level 1 Routing Area. Each IS-IS Level 1 router maintains detailed routing information about its area, including the System ID field of the NSAP and the SNPA address of each system in the area.

Routing Domain

A set of End Systems and Intermediate Systems that operate according to the same routing procedures and that is wholly contained with a single Administrative Domain; often referred to as a Level 2 Area.

System

An OSI End System (ES) or Intermediate System (IS).

Target Identifier (TID)

The network-wide unique identifier used to specify a target NE. The value of a TID may be any valid simple or compound TL1 identifier or text string up to 20 characters in length.

TARP connectivity

TARP connectivity exists between two systems if TARP requests and responses can be exchanged reliably between them. This requires at least one path through which each system supports TARP PDU propagation or the use of TARP manual adjacencies to bridge gaps where systems route OSI traffic but do not support TARP.

TARP-only System

A system that supports TARP but not X.500.

TARP/500 Protocol

The DIT definitions, functions, and procedures specified in this document, wherein TARP and X.500 services are used to provide TID/NET resolution.

TMN System

An [ANS95] compliant system that typically supports management via CMIP, may contain an MIB, and is assumed to follow the OSI Application Entity Model.

Transaction Language 1 (TL1)

A text-based operations management protocol used between certain Operation Systems and Network Elements.

T5

A system in which both X.500 and TARP services are used for the purpose of TID/address resolution. T5 operates only on behalf of the system itself.

T5GW

The TARP/500 gateway functionality, in which X.500 services are used on behalf of other systems. T5GW acts as a gateway between TARP and X.500 for TID resolution by registering TARP systems in the DIB and resolving TARP requests via X.500.

T5GW Area Cache (TAC)

The list of TID/NET mappings representing the systems supporting TARP that are in the same Level 1 area as the T5GW, and which have TARP connectivity to the T5GW. Entries in this cache must be in the process of being added to the DIB, or must have already been added. These entries can be determined via the IS-IS information provided by the OSI Network Layer, and through the use of TARP.

X.500-only System

A system which contains an X.500 DUA and does not support TARP. Such a system may or may not conform to this specification.

2. Overview

TARP is an OSI network layer service user that provides a means to resolve the OSI network address corresponding to a TL1 TID (Target Identifier). TARP is required by GR-253-CORE [BEL95] to be supported on any NE that uses TL1 over the NE-NE interface as the management protocol for OAM&P.

NEs that are not managed via TL1 are not required to support TARP. However, for TARP to work properly, all systems between the system that needs the address for the TID and the system to be managed must support TARP, or else TARP manual adjacencies must be used to bridge gaps where systems route traffic but do not support TARP. This means that systems that require TARP cannot be arbitrarily intermixed with systems that do not support TARP without topology-specific manual provisioning.

Furthermore, some equipment manufacturers and service providers view TL1 as a management protocol that has a limited life expectancy in view of the evolving TMN standards. With this in mind, they have elected to implement the OSI (ISO) X.500 Directory Service [ITU93a] as the solution for address resolution instead of TARP. Their intent is to use solutions based on open standards for address resolution and to make use of X.500 for additional services into the future. X.500 is used in conjunction with the Registration Request Protocol (RRP) [ANS95] so that information for system addresses is automatically entered into the Directory, therefore providing the same kind of service offered by TARP.

This document solves this problem by specifying a Directory Information Tree structure for storing TID/NSAP information, and a gateway function to provide TID/NSAP resolution between TARP-only and X.500-only systems.

The gateway function, T5GW, permits intermixing of TARP-only and X.500-only systems in a network. Using information obtained by the OSI IS-IS protocol, T5GW discovers TARP systems in its area and creates DIB entries for them as needed. It uses both X.500 Directory Access Protocol (DAP) and TARP in response to TARP request PDUs. It also alleviates the need for TARP connectivity between OSI routing areas because it effectively uses the DIB as a repository for storing TID/NSAP information. T5GW could be supported as a feature on any kind of ordinary INE, or it could be implemented as a system whose only purpose was to act as a gateway between X.500 and TARP for TID/NSAP resolution.

T5, another possibility defined in this document, is intended for use in an ordinary NE such as a digital cross connect or an add-drop multiplexor. T5 systems use both TARP and X.500 services, in any order, to resolve requests from the local service user. It is a solution for NE vendors that wish to provide systems to service providers regardless of the address resolution method used by the latter. T5 systems can be deployed in TARP-only networks, X.500-only networks, or mixed networks, however, they do not actively assist mixing of TARP-only systems with X.500-only systems.

3. Conformance

The following are classes of conformance to TARP/500. An implementation may claim conformance to any of the following clauses.

3.1 Conformance by DSAs

A DSA shall:

a) Conform to [ANS95], clause 5.2, Conformance by DSAs.
b) Conform to [SIF96b][1], Section 2, OSI Routing Domains.

[1]. Currently, functionality for determining OSI Routing Domains in the DIB which is contained in [SIF96b] is being proposed for inclusion in issue two of [ANS95]. If accepted, or similar functionality is included, the contents of issue two of [ASN95] will supersede any information in [SIF96b].

3.2 Conformance by DUAs

A DUA implementation shall adhere to the requirements of the following:

a) [ANS95] Section 5.1, Conformance by DUAs.
b) [SIF96b], Section 2, OSI Routing Domains.
c) Section 6.1, DUA Procedures.
d) Section 7.2, Representing TID Information.

3.3 T5 Conformance

A T5 implementation shall adhere to the requirements of the following:

a) [ANS95] Section 5.4, Conformance by RAs
b) [BEL95] Section 8, TARP, as amended by [SIF96a]
c) Section 3.2, Conformance by DUAs
d) Section 6.2, T5
e) Section 6.2.1, DSA Bindings

3.4 T5GW Conformance

A T5GW implementation shall adhere to the requirements of the following:

a) Section 3.3, T5 Conformance
b) Section 6.3, T5GW
c) Section 6.3.7, T5GW Provisionable Items
d) Section 6.3.8.1, T5GW Elements of Procedure as amendments to GR-253-CORE

4. Requirements

4.1 User Requirements

The following user requirements have been identified as essential for the TARP and X.500 Interworking Specification.

1. TL1-based TMN applications require a mechanism that translates between TIDs and NSAP addresses. Currently there are two existing standardized but incompatible solutions for name/address resolution: TARP and the OSI Directory. Since these two are incompatible, a means of interworking them is necessary if networks consisting of a mixture of TARP-based NEs and Directory-based NEs are to be managed.

2. [BEL95] requirement R8-108 states: "When a SONET NE supports TL1/OSI on the NE-NE interface (DCC or LAN), the NE shall also support TARP on the NE-NE interface according to the requirements of Sections 8.7 and C.8." Because this requirement may not be adhered to or may be relaxed, there is a need for TARP NEs to be able discover the name or address for non-TARP NEs. For example, some service providers may plan to migrate from TL1 management to CMIP management. During this migration, they may want to deploy NEs that are TL1 manageable (in order to use TL1 management tools that are already in place) but that do not support TARP/500.
3. When X.500 is deployed, the requirements of [ANS95] shall be adhered to.
4. The user may require the X.500 DIB to store name/address information about SONET NEs that support only TARP, but do not support X.500. As a result of the above requirement, a solution is sought to the problem of how to populate the DIB with entries of NEs that support TARP but are not capable of adding and deleting their own entries within the DIB.
5. Addition of TARP information to the X.500 DIB should not violate existing tmnNE entries that comply with [ANS95].
6. The solution must support architectures where one or more mediation devices are used to provide the TL1 and/or other OAM&P applications on behalf of a SONET NE, which may also support other OAM&P applications at its NSAP.
7. There is a requirement to minimize the amount of changes to existing equipment and to minimize the number of new features and/or interworking units that would need to be deployed to solve the TARP/X.500 co-existence problem.

4.2 Requirements on Underlying Services

TARP/500 functions impose the following requirements on underlying services.

4.2.1 Network Layer Services and Information Required

T5 procedures impose no requirements on the Network layer services or management information obtainable that are not already required by TARP. These requirements are:

a) The ability to send and receive TARP PDUs as CLNP DT PDUs using an N-Selector different than that of Transport. The services defined in [ISO87] are sufficient.
b) The ability to determine the NETs of neighboring systems. This information can be provided by the ES-IS protocol and can be provisioned.

T5GW imposes the following additional requirements:

c) The current set of local area addresses must be obtainable. Note that this set can change at runtime as a function of the IS-IS protocol.

d) The set of System IDs for all systems in the area must be obtainable [BEL95, sec. 8.3.3.2, fig. 8-7, pp. 8-19]. Note that this set can change as systems start up and shut down, or as IS-IS manualAreaAddress values are changed by management (thus splitting or joining areas.)

The information specified in (c) and (d) is available in the IS-IS Link State Database (LSDB) and in the IS-IS Forwarding Database (FDB).

4.2.2 X.500 Services Required

The following X.500 services [ITU93c] are required: Read, Search, Modify, ModifyDN, RemoveEntry, and AddEntry. Optionally, the Abandon service may be used.

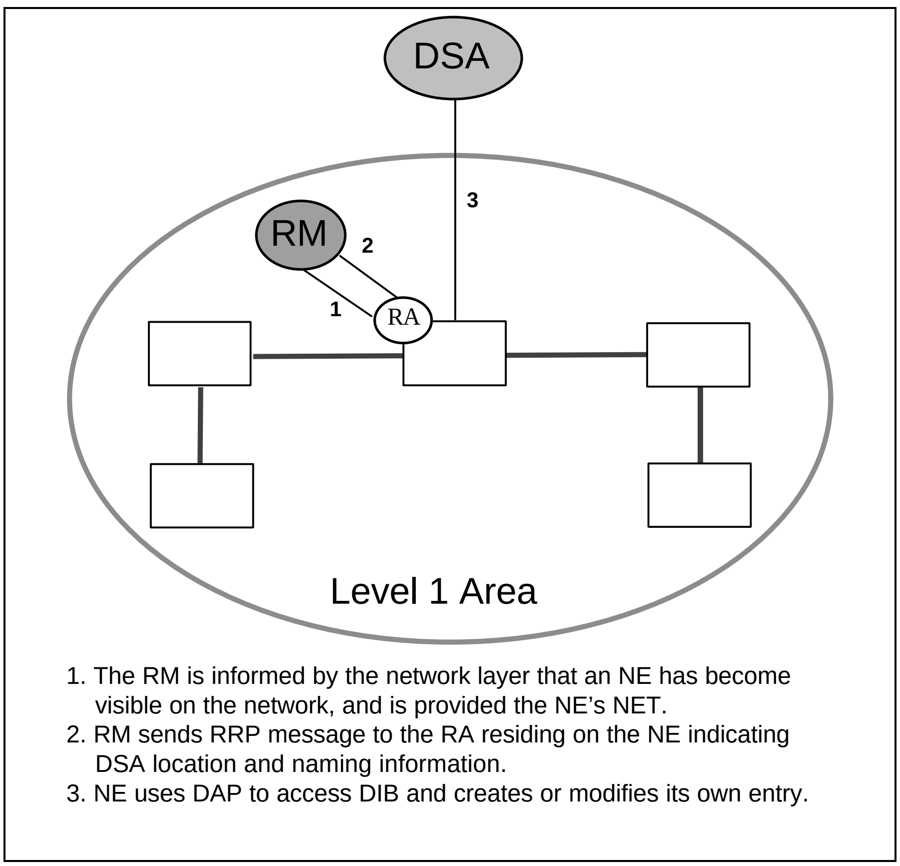

Figure 1 - Registration Request Protocol (RRP).

5. Background

5.1 Use of RRP for SONET

The Registration Request Protocol (RRP) is a relatively simple protocol that involves a Registration Manager application (RM) per Level 1 IS area, and a Registration Agent (RA) on each system.

The RM detects when systems become reachable in its area, using the same means as T5GW. When a system becomes reachable, the RM uses the Remote Operations Service Element (ROSE) to inform the RA on the system of the Presentation Address and Application Entity Title of the managing primary DSA and any alternative DSAs that may be available. The RM also uses the RRP to provide a distinguished name (DN) prefix to the RA, which the system then uses as its Local Naming Context (LCN) when creating local DIB entries. Typically, the system will name its entry by appending its relative distinguished name (RDN) to the LNC. The LNC prefix is typically associated with the Level 1 area. The use of RRP alleviates the need for provisioning this information for each system, although it may be provisioned for use in case the RM is not operating.

Figure 1 illustrates the use of the RRP protocol for the registration of a system.

Each system, when contacted by the RM, saves the information provided and uses it to update its own entry in the DIB as described below. This is done even if the system has already done so and the information has not changed, as this is a means to force a refresh of the DIB, or to quickly rebuild it in case of loss of DIB contents for the area.

5.2 TARP Caches

TARP defines two data caches, the TDC[2], which is applicable to all TARP systems, and the Level 2 TDC, which is applicable only to Level 2 INEs. Both are optional in [BEL95].

The TDC contains only entries normally created by TARP, which are for systems from whom this system has received Type 3 PDUs, as modified by Type 4 PDUs. As a side effect of the T5GW discovery procedure, the TDC will have an entry for each TARP system in the area, because the system will receive a TARP Type 3 PDU from each TARP system in the area.

No entries learned from the DIB are entered into this cache. The reason for this is to let the DIB handle its own caching; the traffic and DSA burden associated with use of the read service is not significant enough to warrant additional caching.

2. There is no mention of a "Level 1 TDC" in [BEL95], and because the non-Level-2 TDC may be used on all systems (ENEs, Level 1 INEs, and Level 2 INEs), it is called the "TDC" or "normal TDC".

The Level 2 TDC contains entries for systems in other areas. Entries are created only upon receipt of TARP Type 2 PDUs from systems in other areas. Presumably these entries are modified on receipt of TARP Type 4 PDUs, although this is not explicit in [BEL95].

6. TARP/500 Elements of Procedure

6.1 DUA Procedures

The following procedures shall be used to access and store information in the DIB. They are intended to permit automatic, unattended storage and retrieval of information in the DIB. They are also to be executable concurrently for different DIB entries. On the other hand, operations on any given DIB entry must be sequenced as necessary to produce meaningful results, as noted below.

6.1.1 Summary of TID/NET representation in the DIB

TID/NET modelling for the DIB is detailed in Section 7, DIB/DIT Structure for TID/NET Information. It is also introduced here so that it can be referred to in the sections below.

In order to support systems which have TID/NET mappings within the Directory Information Base (DIB) as defined within [ANS95], a modelling for this information must be specified. The base object class of an entry in the Directory Information Model is the *tmnNE* which consists of a *commonName, managedElementId,* and an *entityAddress*. The *commonName* attribute's distinguished value is used to uniquely identify the *tmnNE* among sibling entries in the Directory Information Tree (DIT). The *managedElementId* attribute replicates the attribute of the same name in the MIB object *managedElement*. The *entityAddress* (or Network Address) attribute may contain the address of the system, such as the NSAP. Although the *commonName* attribute could be used to represent a TID, a different *commonName* maybe used by the NE. In the case of X.500-only systems, or systems which have both X.500 and TARP functionality, or X.500-only systems which contain a TL1 interface, the information stored in the *tmnNE* object cannot be restricted to only a TID and NET, which would be sufficient for TARP.

The only application of TARP as defined in [BEL95] is for the resolution of TID/NETs used to locate a [BEL95] conformant TL1 application over a seven-layer OSI stack. We can therefore assume that TARP provides mapping information for the support of TL1 applications. This allows for representation in the DIB of a TARP TID/NET pair as an OSI application, or more specifically, as a TL1 *applicationEntity* [ITU93d]. The attributes of the *applicationEntity* object class used for representing TL1 application consist of the mandatory attributes *commonName* and *presentationAddress,* as well as the optional attribute, *supportedApplicationContext*. The *commonName* attribute contains the TID value of the system containing the TL1 application, while the TL1 application's presentation address is held in the *presentationAddress* attribute. The *supportedApplicationContext* attribute holds the TL1 application context object identifier.

In order to support situations where a proxy TL1 process may reside on a mediation or adapter device, the *apaeAlias* object class is used to create an alias object entry referencing the proxy TL1 *applicationEntity*. An alias object of this type contains the *commonName* attribute with the value of the TID corresponding to the system represented, as well as the *aliasedEntryName* attribute which contains the DN of the *applicationEntity* which represents the proxy process.

For those X.500 systems that already have a *tmnNE* object entry in the DIB, all that is required is the addition of a subordinate TL1 *applicationEntity* object entry. For those systems that do not have a *tmnNE* object entry in the DIB, one must be created by using the available TARP information. More details on the DIB structure for TARP systems can be found in Section 7.

6.1.2 DibUpdateOwnEntry Procedure

The following procedure shall be used to add or modify their own entries to the DIB.

It may be implemented by a read followed by an add or modify as necessary, or by an add followed by a modify if necessary. The modifyDN service shall be used if the object's naming attribute has changed and the previous value is known.

1. The NE shall add or modify a *tmnNE* object entry anchored under the LNC as described in section 7.2.1. If the DibSearchDomainForTidToGetNet Procedure or the DibSearchDomain-ForNetToGetTid Procedure is supported by the NE, additional procedures specified in [SIF96b] need to be followed.
2. The system shall add/modify a TL1 *applicationEntity* object entry as specified in section 7.2.2, and anchored as a subordinate of this *tmnNE* object entry. If a mediation device is being used to provide proxy TL1 services, an *apaeAlias* object entry may be used instead of the *applicationEntity* object as described in section 7.2.2.
3. Additional *applicationProcess*, *applicationEntity*, and *apaeAlias* object entries for other supported applications may be added or modified as subordinates of this *tmnNE* object.

6.1.3 DibSearchAreaForTidToGetNet Procedure

This procedure shall be used to get the NET associated with a TID of a system in the local area. The X.500 Search operation shall be used, using the LNC as the base object, to find the object whose *commonName* is the TID, whose *applicationContext* is *tl1PeerComm* [BEL95, sec.8.3.7.5, R8-80, pp. 8-24], and whose *objectClass* is *applicationEntity*. The *searchAliases* argument shall be omitted or specified as TRUE.

6.1.4 DibSearchDomainForTidToGetNet Procedure

The X.500 Search operation shall be used to search the DIB for an entry with matching TID, and the entry's NET is retrieved. The search shall be rooted at the DN (or DNs) representing the OSI Routing Domain to find the object whose *commonName* is the TID, whose *applicationContext* is *tl1PeerComm*, and whose *objectClass* is *applicationEntity*. The *searchAliases* argument shall be omitted or specified as TRUE. The mechanism used to obtain the root for the search is described in [SIF96b].

6.1.5 DibSearchDomainForNetToGetTid Procedure

The X.500 Search operation shall be used to search the DIB for an entry with matching NET, and the entry's TID is retrieved. The search shall be rooted at the DN (or DNs) representing the OSI Routing Domain. The search shall target an object whose *applicationContext* is *tl1PeerComm*, whose *objectClass* is *applicationEntity,* and whose *presentationAddress* attribute contains an *nAddress* component set to the NSAP generated from the target NET along with *tSelector*, *sSelector*, and *pSelector* components set to the TL1 application selectors configured in the area. The *searchAliases* argument shall be omitted or specified as TRUE. The mechanism used to obtain the root for the search is described in [SIF96b].

6.2 T5

When resolving TIDs or NETs, T5 systems shall attempt use of both TARP and X.500 before an attempt is considered a failure.[3] When their TID or NSAP changes, T5 systems shall advertise their TID and NSAP via TARP and they shall also update their entries in the DIB accordingly. They also create or update their entries in the DIB whenever they are contacted by the RM, as is required by [ANS95].

6.2.1 DSA Bindings

A T5 system may choose to use a single DSA binding for all requests, it may choose to use a separate binding for each request, or it may open and close DSA bindings as desired to optimize time or use of resources on the DSA. This is a local matter. If the DSA bind fails a T5 system should return to TARP-only operation while trying to bind again.

6.3 T5GW

T5GW systems perform all the functions of T5 systems as described above. In addition, T5GW systems perform special functions on behalf of other systems to permit interoperability between

3. For T5, the order in which TARP and X.500 are attempted is a local matter.

TARP-only and X.500-only systems. They do so in two ways. The first is by registering TARP systems in the DIB, permitting X.500-only systems to discover TARP-only systems and permitting TARP-only systems to discover other TARP-only systems that could not be discovered due to the lack of TARP connectivity. The second is by use of X.500 to perform TARP TID/NET resolution, permitting TARP-only systems to discover X.500-only systems. In effect, T5GW connects all TARP-only systems to the DIB.

T5GW systems are normally Level 2 INEs, although T5GW can be run on a Level 1 INE when necessary, either to avoid upgrading a Level 2 INE that is already in place, or to connect TARP "islands" within a Level 1 area. When T5GW is run on a Level 1 INE, it and behaves just like a level two T5GW system, except that it does not perform TARP Level 2 propagation.

For registration, a T5GW system must discover all TARP systems in its area, and make sure they are registered in the DIB. The systems are discovered by use of information from the IS-IS protocol, and corresponding DIB entries are added or modified as necessary. When contacted by RRP, T5GW systems create or update their entries in the DIB, along with entries for all TARP systems discovered in the area.

For TID/NET resolution, TARP PDUs are handled normally, including propagation, except that the T5GW system also interrogates the DIB, and if the information is found in the DIB, a response is generated.

T5GW does not provide a gateway function to find the TID that matches a NET when the requesting system is a TARP-only system and the target system is an X.500-only system. This is because the TARP system sends its Type 5 PDU directly to the NET. The X.500-only system will drop the PDU because it has no user registered at the N-Selector. The principal use of this function is to populate the DIB with entries for TARP-only systems, as is done by T5GW, however, the X.500-only system will have already added its TID/NET pair to the DIB if such a mapping exists, so support for this feature on the X.500-only system would be redundant.

6.3.1 T5GW Area Cache

The T5GW Area Cache (TAC) is the table of TID/NET mappings representing the systems supporting TARP that are in the same Level 1 area as the T5GW to which the T5GW has TARP connectivity. Entries residing in this cache are either in the process of being added to the DIB or have already been added.

A TAC entry consists of two components: the TID of the system and the NET corresponding to that TID. Entries are added to the TAC as a result of the T5GW Discovery Procedure, described in Section 6.3.2.

The T5GW shall not respond to a TARP type 1 or 2 request if the target TID is present in the TAC, since the T5GW can assume that the target system itself will respond to the query. This reduces network traffic between the system and managing DSA.

TAC entries for which TARP no longer responds should be flushed from the TAC. How this is done is a local matter.

6.3.2 T5GW Discovery Procedure

T5GW is responsible for creating DIB entries for all TARP-only systems in its Level 1 area to which it has TARP connectivity. Since it has no means of distinguishing systems that use both TARP and X.500 from TARP-only systems, it creates DIB entries for all TARP systems, whether they support X.500 or not. To do this, T5GW systems shall perform the following procedure.

> For each new System ID in the area (other than that of the system itself), construct the set of potential NSAPs for TARP on the system whose System ID it is by prepending each area address of this area[4] to the System ID, and appending the TARP NSEL to this. A "new" System ID can be distinguished by its absence from the TAC. Only TAC entries whose tar-pro field specifies ISO addressing and whose *tar-por* field matches a local area address can be considered a match. Discovery procedures for other possible addressing mechanisms are beyond the scope of this recommendation.
>
> For each NSAP so constructed, originate a TARP Type 5 PDU. Between sending these PDUs, wait for a small interval as necessary to avoid flooding. Any TARP Type 3 PDUs sent in response are processed the same as any other received Type 3 PDUs, causing the information to be updated in the DIB and entered into the TAC. The small interval is a pseudorandom number between 0 and a provisionable value, the discovery procedure anti-flooding timer.
>
> For each entry in the TAC, if the NSAP no longer matches any local area address, delete the entry.

Systems shall also be discovered (entered into the TAC) upon receipt of their TARP Type 4 PDUs.

The discovery procedure shall be run at initialization time. It can optionally be run as new System IDs appear in the area or if the set of local area addresses changes. Immediately after the TAC or TDC is flushed by user request, the TAC shall be emptied and the procedure shall be run.

---

4. A single routing area may contain multiple area addresses.

It shall also run periodically for robustness, in case the TARP Type 4 PDU from a system is lost and either the Type 5 PDU sent to the system or its Type 3 response is lost. This also alleviates the need to get notifications from IS-IS that a System ID has become reachable or that the list of area addresses for this area has changed.

6.3.3 DibUpdateOthersEntry Procedure

This procedure shall be used by T5GW systems to add or modify DIB entries on behalf of other NEs. The modifyDN service shall be used if the NE's TID has changed and the previous value is known. Otherwise, the DibSearchAreaForTidToGetNet Procedure shall be performed first to determine whether AddEntry or ModifyEntry operations are required. If the entry is present and the *nAddress* component in the *presentationAddress* does not match the other NE's NET, or any of the Selector values have changed, then the *presentationAddress* attribute of the *applicationEntity* is modified to contain the proper values. If the entry is not present, then the following procedure shall be used:

1. The T5GW system shall add a *tmnNE* object entry anchored under the LNC, as described in section 7.2.1, with the TID/NET information for the system detected via TARP. **The id-neType-createdByT5GW** object identifier is specified as the value of the *neType* attribute of this *tmnNE* object entry.
2. The T5GW system shall add or modify a TL1 *applicationEntity* object entry populated with the information specified in section 7.2.2, and anchored as a subordinate of this *tmnNE* object entry.

This procedure shall not be invoked for a given TID if it is already executing for the same TID.

6.3.4 T5GW Handling of Directory Responses

T5GW systems abandon any subsequent TARP operations related to a Directory Services request whenever the request succeeds. Also, on a successful response to a Directory Search issued with a TARP Type 1, 2 or 5 PDU, the information is returned to the application the timer associated with the TARP PDU shall be stopped.

On a successful response to a Directory Search request issued by the DibSearchAreaForTidToGetNet Procedure upon receipt of a TARP Type 1 PDU, a proxy TARP 3 response shall be issued as described in [BEL95, sec. 8.7.5.6.3, item 6, pp. 8-47].

6.3.5 T5GW Procedures for User Requests

To find the NET that matches a TID, if no matching TAC entry is found, the DibSearchAreaForTidToGetNet Procedure shall be used in addition to sending the TARP Type 1

PDU. The first response (TARP Type 3 PDU or affirmative response from the Directory) shall be used to fulfill the user's request. If a TARP time-out occurs, the DibSearchDomainForTidToGetNet Procedure shall be used in addition to sending the TARP Type 2 PDU, and the first response is used to fulfill the user's request.

To find the TID that matches a NET, the DibSearchDomainForNetToGetTid Procedure shall be used in addition to sending the TARP Type 5 PDU.

To send notification of TID or protocol address change, the system shall use the DibUpdateOwnEntry Procedure in addition to sending its TARP Type 4 PDU.

6.3.6 T5GW Procedures for Reception of TARP PDUs

The following procedures shall be performed by both Level 1 and Level 2 intermediate systems.

When propagating TARP Type 1 PDUs[5], T5GW systems shall invoke the DibSearchAreaForTidToGetNet Procedure if and only if the target TID (*tar-ttg*) in the TARP Type 1 is not present in the TAC. (If the target TID is in the TAC, then the system identified by the TID is in our Level 1 area and can respond to the TARP request itself.) If a matching DIB entry is found, the T5GW shall issue a TARP Type 3 response containing the requested information.

When propagating Type 2 PDUs, if the *tar-por* matches a local area address, the DibSearchDomainForTidToGetNet Procedure shall be invoked.

On receipt of TARP Type 3 PDUs that are valid responses to Type 1, Type 2, or 5 PDUs, if the *tar-por* field matches a local area address, the (TID, NSAP) pair shall be added to the TAC and the DibUpdateOthersEntry Procedure shall be invoked.

When propagating Type 4 PDUs, if the *tar-por* field of the TARP PDU matches a local area address, the DibUpdateOthersEntry Procedure shall be invoked.

6.3.7 T5GW Provisionable Items

Timer A: Local NET Search Timer (default 30 seconds)

> This timer is the duration of the longest reasonable search of the domain for a given NET in order to determine the corresponding TID, used in the DibSearchDomainForTidToGetNet Procedure. For T5GW, this should be long enough to produce results when the NSAP is present, but should be shorter than the system's TARP Timer T3, because any

5. For Level 2 intermediate systems the Level 2 TDC could potentially be consulted for a match, however, the Level 2 cache is considered unreliable and must not be used with the T5GW functionality.

results returned after this time would be ignored. This timer is used only for requests from the local application. The range is 1 to 3600 seconds.

Timer B: Local TID Search Timer (default 40 seconds)

This timer is the duration of the longest reasonable search of the domain for a given TID in order to determine the corresponding NET, used in the DibSearchDomainForTidToGetNet Procedure. For T5GW, it should be long enough to produce results when the NET is present, but should be shorter than the sum of the system's TARP Timers T2 and T4, because any results returned after this time would be ignored. This timer is used only for requests from the local application. The range is 1 to 3600 seconds.

Timer C: Gateway NET Search Timer (default 30 seconds)

This timer is the same as *Timer A*, but is used only for X.500 requests made on behalf of other systems. It should be shorter than the largest value of TARP Timer T3 for any system in the area, because any results returned after this time would be ignored. The range is 1 to 3600 seconds.

Timer D: Gateway TID Search Timer (default 40 seconds)

This timer is the same as *Timer B*, but is used only for X.500 requests made on behalf of other systems. It should be shorter than the largest sum of the values of TARP Timers T2 and T4 for any system in the area, because any results returned after this time would be ignored. The range is 1 to 3600 seconds.

Timer E: Discovery Timer (default 900 seconds)

This timer represents the frequency at which the T5GW should invoke the discovery procedure as described in Section 6.3.2. If the T5GW implementation is capable of receiving IS-IS notifications when a system in the local Level 1 area becomes reachable, then this periodic timer can be configured to run less frequently. It’s function would then be as a consistency audit, instead of a mechanism for detecting systems. For cases where IS-IS notifications are not provided, this periodic timer can be configured to run more frequently, allowing for newly reachable systems to be added more quickly. The range is 1 to 3600 seconds.

Timer F: Discovery Anti-Flooding Timer (default 1000 milliseconds)

This timer is used to avoid flooding the network with TARP Type 5 requests and their TARP Type 3 responses that are generated during the T5GW discovery procedure described in Section 6.3.2. The range is 10 to 60,000 milliseconds.

6.3.8 T5GW Requirements as Amendments to GR-253-CORE

The following excerpts are taken from [BEL95] and updated to support T5GW. Changes from [BEL95] are shown in ***[bold italics]***.

Note that a Level 1 IS T5GW is permitted to issue proxy TARP Type 3 PDUs for TID/NET pairs resolved from information contained in the DIB.[6] This relaxes the topological constraints of deployment.

In the pseudo-code the terms "in the local area" and "not in the local area" are used. This refers to whether the system that initiated the TARP request is in the same Level 1 area. A system is "in the local area" is if the area portion of the NET address provided in the *tar-por* field of the TARP request message matches an area address of the local Level 1 area. (This list of area addresses is obtained by the IS-IS protocol and may change dynamically, although very infrequently.)

As a result of the discovery procedure, T5GWs issue Type 5 PDUs for all systems in its local area and receives Type 3 responses, and these responses are entered into the TAC. Therefore, on a T5GW the TAC will be a reflection of all the local-area systems participating in TARP and with which the T5GW has TARP connectivity. This is true unless the TAC has recently been flushed and is in the process of being rebuilt.

6.3.8.1 T5GW Elements of Procedure as amendments to GR-253-CORE

FROM PAGE 8-42 ...

8.7.4.1 Find NET That Matches TID

When the NE has a TID and needs to find the matching NET, the procedure is the following:

The TARP processor first checks its TDC for the match. If a match is found, the TARP processor would return the result to the requesting application. If no match is found, a TARP Type 1 PDU is originated***[, and If T5GW, invoke the DibSearch-AreaForTidToGetNet Procedure]***. If Timer T1 (see Table 8-3) expires, a TARP Type 2 PDU is originated, status information is passed back to the requesting application indicating that the TARP Type 1 request has failed and that a TARP Type 2 request is being initiated***[, and If T5GW, invoke the DibSearchDomain-ForTidToGetNet Procedure]***. If Timer T2 expires, then Timer T4 is started, an error recovery routine is initiated, and status information is passed back to the requesting application indicating that error recovery is being initiated.

---

6. Proxy TARP Type 3 PDUs would normally only be issued by a Level 2 IS system using the Level 2 TDC.

The error recovery routine is as follows...

FROM PAGE 8-43...

8.7.4.2 Find TID That Matches NET

When the NE has a NET and needs to find the matching TID, the following procedure takes place:

> ***[If T5GW, invoke the DibSearchDomainForNetToGetTid Procedure.]*** A TARP Type 5 PDU is originated. Timer T3 (see Table 8-3) is used; however, if this timer expires, no error recovery procedure occurs, and a status message is provided to indicate that the TID could not be found.

A scenario in which this may occur is one in which a Directory Server NE (DSNE) may want to populate its database. A DSNE would typically know which NETs it could communicate with and could then use TARP to learn the TIDs that correspond to those NETs. ***[T5GW makes use of this facility.]***

8.7.4.3 Send Notification of TID or Protocol Address Change

When the NE needs to notify other NEs of a TID or Protocol Address change, the procedure is the following:

> ***[If T5GW, invoke the DibUpdateOwnEntry Procedure.]*** The TARP Processor originates a TARP Type 4 PDU in which the tar-ttg field contains the NE's TID value that existed prior to the change of TID or Protocol Address.

Note that there is no confirmation that other NEs have successfully received the address change information sent in the TARP Type 4 PDU.

FROM PAGE 8-46...

8.7.5.6 Receipt of a TARP PDU

The following steps are taken by the TARP processor upon receipt of an incoming TARP PDU.

1. Check if tar-lif = 0; if so, discard TARP PDU.
2. Check tar-pro to see if the Protocol Address Type is supported; if not supported, then discard the TARP PDU.
3. Check tar-seq and perform the Loop Detection Procedure (only when NE is an IS, see Section 8.7.5.7***[. If the TARP Type Code is 3 or 5, only update the LDB if tar-seq is zero, otherwise do not perform Loop Detection.]***).
4. The next step depends on the TARP Type Code value and whether the NE is an ES or IS.

8.7.5.6.1 End Systems (ESs)

5. If the TARP Type Code is 1 or 2, check tar-ttg. If tar-ttg matches the ES's TID, then originate a TARP Type 3 PDU response.

6. If the TARP Type Code is 3, update TDC and pass response to the requesting application, unless the response is unsolicited and/or a duplicate response in which case the response should be discarded.

7. If the TARP Type Code is 4, then check to see if tar-ttg matches with TDC data. If there is a match, update TDC with the new information.

8. If the TARP Type Code is 5, originate a TARP Type 3 PDU response.

9. If the TARP Type Code is a value that is not supported by the NE, discard the PDU.

8.7.5.6.2 Level 1 Intermediate Systems (ISs)

5. If the TARP Type Code is 1 or 2, check tar-ttg. If tar-ttg matches the IS's TID, originate a TARP Type 3 PDU response. If the tar-ttg does not match the IS's TID, perform Level 1 Propagation (see Section 8.7.5.8***)[, and If T5GW and tar-ttg is not in TAC and tar-por is in local area, invoke the DibSearchAreaForTidToGetNet Procedure for Type 1 or DibSearch-DomainForTidToGetNet Procedure for Type 2, for the tar-ttg]***.

6. If the TARP Type Code is 3, update the TDC and pass response to requesting the application, unless the response is unsolicited and/or a duplicate response in which case the response should be discarded. ***[If T5GW and tar-por in local area and a valid response from a Type 5 PDU, update TAC with new information and invoke the DibUpdateOthersEntry Procedure. If T5GW and X.500 search operation pending, optionally abandon search operation.]***

7. If the TARP Type Code is 4, check to see if tar-ttg matches with TDC data. If there is a match, update TDC with the new information***. [If T5GW and tar-por in local area, update TAC new information and invoke the DibUpdateOthersEntry Procedure.]*** In either case, perform Level 1 Propagation (see Section 8.7.5.8).

8. If the TARP Type Code is 5, originate a TARP Type 3 PDU response.

9. If the TARP Type Code is a value that is not supported by the NE, perform Level 1 Propagation (see Section 8.7.5.8).

8.7.5.6.3 Level 2 Intermediate Systems (ISs)

5. If the TARP Type Code is 1, check tar-ttg. If tar-ttg matches the IS's TID, originate a TARP Type 3 PDU response. If tar-ttg does not match the IS's

TID, check Level 2 TDC for a match[[7]]. If a match, originate a TARP Type 3 PDU response.[8] If no match, perform Level 1 Propagation (see Section 8.7.5.8)***[, and If T5GW and tar-ttg is not in TAC and tar-por is in local area, invoke the DibSearchAreaForTidToGetNet Procedure for the tar-ttg]***.

6. If the TARP Type Code is 2, check tar-ttg. If tar-ttg matches the IS's TID, originate a TARP Type 3 PDU response. If tar-ttg does not match the IS's TID, check the area part of the address found in the tar-por field. If the area part of the address does not match the IS's area, update Level 2 TDC. ***[If T5GW and tar-ttg is not in TAC and tar-por is in local area, invoke the DibSearchDomainForTidToGetNet Procedure for the tar-ttg.]*** In either case, perform Level 1 and 2 Propagation (see Section 8.7.5.8).

7. If the TARP Type Code is 3, update TDC ***[If T5GW and tar-por in local area and a response from Type 5 PDU, update TAC with new information and invoke the DibUpdateOthersEntry Procedure. If T5GW and X.500 search operation pending, optionally abandon search operation.]*** and pass response to the requesting application, unless the response is unsolicited and/or a duplicate response in which case the response should be discarded.

8. If the TARP Type Code is 4, check to see if tar-ttg matches with TDC data. If there is a match, update TDC with the new information. ***[If T5GW and tar-por is in local area, update TAC with new information and invoke the DibUpdateOthersEntry Procedure.]*** In either case, perform Level 1 and 2 Propagation (see Section 8.7.5.8).

9. If the TARP Type Code is 5, originate a TARP Type 3 PDU response.

10. If the TARP Type Code is a value that is not supported by the NE, perform Level 1 and 2 Propagation (see Section 8.7.5.8).

6.3.8.2 Pseudo-code

FROM PAGE 8-52

8.7.10 TARP Pseudo-code

---

[7]. It is suggested that the Level 2 TDC be disabled for the T5GW functionality since any information that would be maintained in the Level 2 TDC can be retrieved from the X.500 Directory.

[8]. In this scenario, the Level 2 IS is acting as a DS-NE. When the TARP Type 3 response PDU is being constructed, the tar-tor and tar-por fields contain the TID and Protocol Address corresponding to the requested TID address, instead of containing the TID and Protocol Address of the Level 2 IS originating the TARP Type 3 PDU.

This section provides pseudo-code as an aid to the reader in understanding the steps taken by the TARP processor upon receipt of an incoming TARP PDU. The normative description of these steps is provided in Section 8.7.5.6.

The Discovery procedure is executed at initialization time (may be started by Registration Agent side of RRP protocol) and periodically in accordance with the Discovery Timer. Notice that on T5GW systems the OSI stack routing information is used for periodic audits of the local Level 1 area as systems are added or removed The T5GW TAC is an up-to-date copy of all local area Level 1 TARP systems that have TARP connectivity with the T5GW.

BEGIN Pseudocode ALL

```
[All Systems
Procedure Discovery()
  BEGIN
      [If T5GW BEGIN
          Bind to Directory Server.
          invoke DibUpdateOwnEntry Procedure.
      END]
      [If T5GW BEGIN
          Consult routing information base for all systems in Level 1 area.
          For all new systems in area:
          BEGIN
              Construct Type 5 Find TID that matches NET
              (do not issue read from Directory)
              Hand-off to Forwarding Process.
          END
          For all systems no longer found in area:
          BEGIN
              Delete TDC entry.
          END
      END]
      Construct Type 4 Address Change
      Hand-off to Forwarding process.
  END
END Discovery;]
```

<u>End System</u>

```
Procedure EndSystem()
  BEGIN

  Check the TARP Lifetime
      IF Lifetime has expired THEN Discard packet END PROC
```

ELSE (*Check to see if the protocol type is supported*)
IF Protocol Type is not supported **THEN** Discard packet **END PROC**
ELSE (*Perform Type Analysis*)
CASE

TARP type is 1 and Target TID is my TID:
Construct Response (Type 3) hand-off to Forward Process **END PROC**

TARP type is 2 and Target TID is my TID:
Construct Response (Type 3) hand-off to Forward Process **END PROC**

TARP type is 3:
Add triplet to my cache and pass response to requesting application
Discard duplicate responses **END PROC**

TARP type is 4:
IF My cache has a TID match **THEN**
Update my cache with new information
Discard packet **END PROC**
ELSE My cache does not have a match
Discard packet **END PROC**

TARP type is 5:
Construct Response (Type 3) hand-off to
Forward Process **END PROC**

TARP type is not supported or Target TID is not my TID:
Discard packet **END PROC**

END;
END EndSystem;

<u>Level 1 NE</u>
Procedure Propagation(**LEVEL**)
LEVEL: (Level1, Level2, ALL)
BEGIN
Construct adjacency and routing database for adjacency information.
Decrement TARP packet lifetime by 1 hop.
IF tar-lif > 0 **THEN**
Construct packet(s) with new destination and source addresses (PDU Header only, all adjacencies except adjacencies to NE from which packet was received) and hand-off to Forwarding process.
END;
END Propagation()

```
Procedure Loop Detection()
   BEGIN (*Loop Check*)
        IF tar-por is a match THEN
             BEGIN
             [IF TARP Type is 3 or 5 THEN
                  RETURN END PROC]
             IF pdu.tar-seq is non-zero and # ldb.tar-seq
             OR pdu.tar-seq = 0 and LDB Entry Timer is running THEN discard pdu
             ELSE IF pdu.tar-seq = 0 and LDB Entry Timer is not running THEN
             ldb.tar-seq = 0
             start LDB Entry Timer
             ELSE pdu.tar-seq > ldb.tar-seq
             ldb.tar-seq = pdu.tar-seq
             END
        ELSE tar-por is not a match.
             Add couplet (tar-por, tar-seq) to LDB.
             IF tar-seq = 0 THEN start LDB Entry Timer
        END;
   END;
END Loop Detection()

Procedure Intermediate System1()
   BEGIN
   (*Check the TARP Lifetime and Run (Loop Detection Procedure)*)

   IF Lifetime has expired and or LDB is a match THEN Discard packet
        END PROC

   ELSE (*Check to see if the protocol type is supported*)
        IF Protocol Type is not support THEN Discard packet END PROC
        ELSE (*Perform Type Analysis*)
        CASE

        TARP type is 1:
        IF Target TID is my TID THEN
             Construct Response (Type 3) hand-off to Forward Process END PROC
        [ELSE Target TID is not my TID
             IF T5GW and tar-ttg is not in TAC and tar-por in local area THEN
                  invokeDibSearchAreaForTidToGetNet Procedure
             BEGIN Propagation (Level1)
             END Propagation
        END PROC]
```

TARP type is 2:
IF Target TID is my TID **THEN**
Construct Response (Type 3) hand-off to Forward Process **END PROC**
[ELSE Target TID is not my TID
IF T5GW and tar-ttg is not in TAC and tar-por in local area THEN
invoke DibSearchDomainForTidToGetNet Procedure
BEGIN Propagation (Level1)
END Propagation
END PROC]

TARP type is 3:
Add triplet to my cache and pass response to requesting application.
[IF T5GW THEN
IF tar-por in local area and a response from Type 5 PDU THEN
BEGIN
Update TAC with new information
Invoke DibUpdateOthersEntry Procedure
END
IF T5GW and X.500 search operation pending THEN
Abandon search operation]
Discard duplicate responses **END PROC**

TARP type is 4:
IF My cache has a match **THEN**
Update my cache with new information
ELSE My cache does not have a match
Do not update cache
[IF T5GW and tar-por in local area THEN
BEGIN
Update TAC with new information
Invoke DibUpdateOthersEntry Procedure
END]
BEGIN Propagation(All)
END Propagation **END PROC**

TARP type is 5:
Construct Response (Type 3)
Hand-off to Forward Process **END PROC**

TARP type is not supported:
BEGIN Propagation (Level1)
END Propagation **END PROC**
END;
END IntermediateSystem1;

<u>Level 2 NEs</u>

```
Procedure IntermediateSystem2()
  BEGIN
      Check the TARP Lifetime and Run (Loop Detection Procedure)
      IF Lifetime has expired and or LDB is a match THEN
          Discard packet END PROC
      ELSE (*Check to see if the protocol type is supported*)
          IF Protocol Type is not supported THEN Discard Packet END PROC
          ELSE (*Perform Type Analysis*)
          CASE

          TARP type is 1:
          IF TID is my TID THEN
              Construct Response (Type 3)
              Hand-off to Forward Process END PROC
          ELSE (*Check my level 2 cache for a match*)
              IF target TID has a match in my level 2 cache THEN
                  Construct Response (Type 3)
                  Hand-off to Forward Process END PROC
              ELSE Target TID does not match
                  [IF T5GW and tar-ttg is not in TAC and tar-por is in local area
                      THEN invoke DibSearchAreaForTidToGetNet Procedure for
                          the tar-ttg]
                  BEGIN Propagation (Level1)
                  END Propagation END PROC
          END;

          TARP type is 2:
          IF TID is my TID THEN
              Construct Response (Type 3)
              Hand-off to Forward Process END PROC
          ELSE TID is not My TID
              (*Check area address in data portion of packet.  If the area address
              does not match my area address, add triplet to my level 2 cache*)
              [IF T5GW and tar-ttg is not in TAC and tar-por is in local area THEN
                  DibSearchDomainForTidToGetNet Procedure for the tar-ttg]
              BEGIN Propagation(All)
              END Propagation END PROC
          END;

          TARP type is 3:
              Add triplet to my cache and pass response to requesting application
              [IF T5GW and  tar-por in local area and a response from Type 5 PDU
```

THEN Update TAC with new information and invoke the DibUpdateOthersEntry Procedure
IF T5GW and X.500 search operation pending THEN
Abandon search operation]
Discard duplicate responses **END PROC**

TARP type is 4:
IF My cache has a TID match **THEN**
Update my cache with new information
ELSE My cache does not have a TID match
Do not update cache
[IF T5GW and tar-por is in local area THEN
Update TAC with new information and invoke the
DibUpdateOthersEntry Procedure]
BEGIN Propagation(All)
END Propagation **END PROC**
END;

TARP type is 5:
Construct Response (Type 3) hand-off to Forward Process **END PROC**

TARP type is not supported:
BEGIN Propagation(All)
END Propagation **END PROC**

END;
END;
END IntermediateSystem2;

END Pseudocode **ALL**

7. DIB/DIT Structure for TID/NET Information

There are three DIB registration scenarios that can exist for SONET systems in a network that uses the TARP/X.500 functionality:

- TARP-only systems registered via the T5GW functionality;
- T5 systems;
- X.500-only systems.

Section 7.1 provides a basic description of how TMN and SONET systems are represented in a X.500 DIB conformant to [ANS95]. Section 7.2 defines the DIT naming structures, as well as the Directory Schema additions required to support the TARP/X.500 functionality.

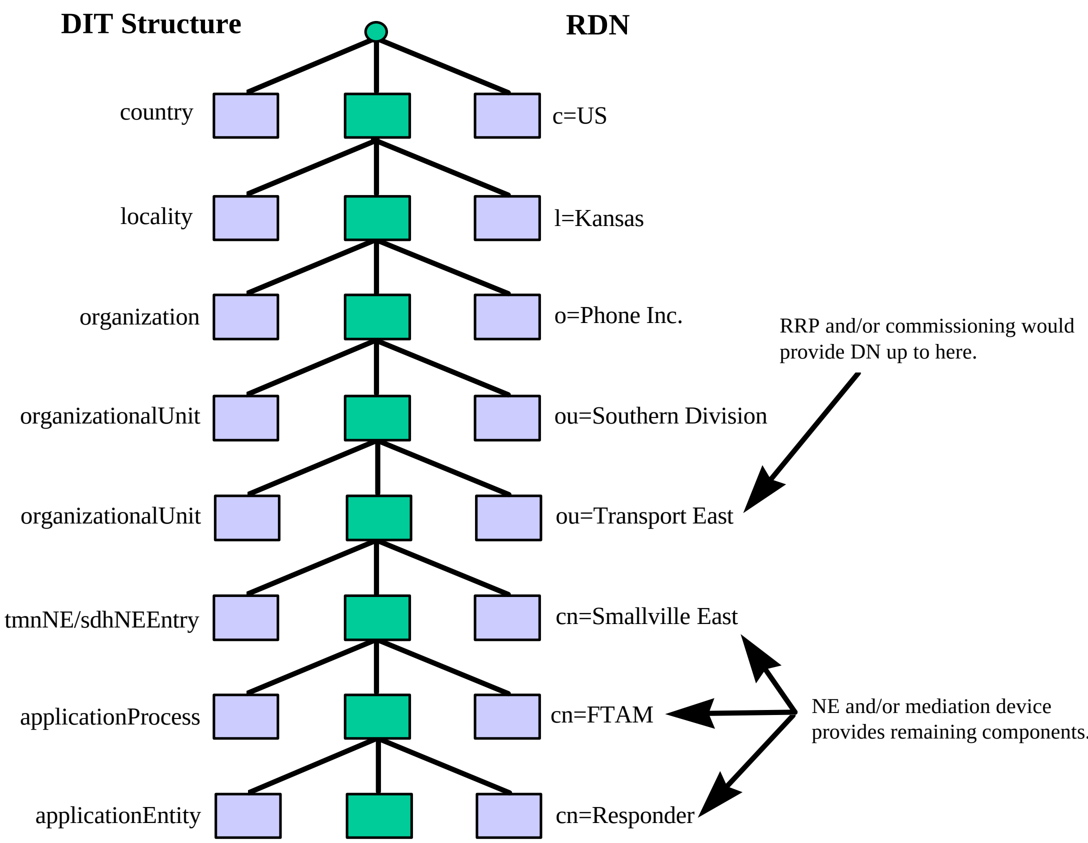

Figure 2 - Example naming structure for an FTAM Responder application entity.

7.1 Background: Naming of X.500-based TMN and SONET Systems

Use of Directory Services for TMN and SONET systems is specified in [ANS95], which defines a number of object classes and attribute types that are to be used by X.500 in a SONET management network, and specifies the RRP protocol, which permits systems to create their own entries in the DIB with minimal provisioning.

According to [ANS95], the location of any TMN or SONET system entry in the DIT is dictated by the information provided to the system's Registration Agent via the *namePrefix* field of the RRP as described in Section 2.1. This prefix, which is a DN, is used as the Local Naming Context (LNC) under which a system creates its entry. Figure 2 provides an example naming structure for a hypothetical FTAM Responder application entity, with Figure 2 in [ANS95] defining the actual DIT naming structure schema for TMN and SONET systems.

In the [ANS95] Information Model, a system has a base object entry of the class *tmnNE* in which basic system information is stored. A system's entry can then be expanded to include the *sdhNEEntry* class for the ability to use even more descriptive attribute types. Once the base *tmnNE* entry has been created, *applicationProcess* and *applicationEntity* objects [ITU93d] can optionally be populated underneath this entry to represent the different OSI services it supports. For the case where a mediation/adapter device is used to provide services on behalf of a system, an *apaeAlias* object located under the system's *tmnNE* entry can be used to reference the object for the desired proxy service, which would be an *applicationProcess* or *applicationEntity* entry located under the entry for the mediation/adapter device. Figure 3 illustrates such a scenario.

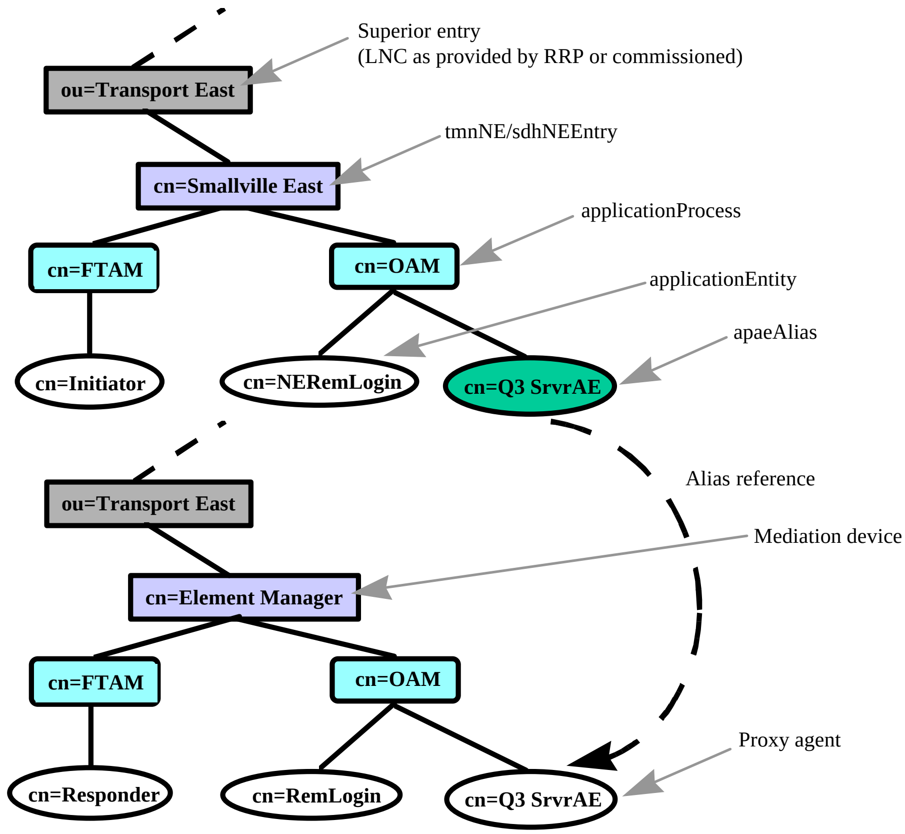

Figure 3 - Example DIT entries for TMN and SONET systems.

A distinguished *commonName* attribute value is used by the *tmnNE* to uniquely name itself among sibling entries in the DIT. There may be multiple *commonName* attribute values to provide alternative names for searching, however, there is only one distinguished value. Typically, if the *userLabel* conditional package is present in the MIB object for a *tmnNE* [ITU94], the user label is used as the distinguished value for the *commonName* DIB attribute. If the *userLabel* attribute is not present, or does not provide a unique name among its siblings, then an alternative naming value must be defined for that *tmnNE* entry.[9]

[9]. For a SONET system, one such alternative name could be the system's Target Identifier.

7.2 Representing TID Information

For storing TID information in the DIB, it is necessary to employ a scheme that will conform to existing X.500 naming mechanisms, while providing unique naming in the presence of existing [ANS95] entries. It is also important to integrate TID information in such a way as to facilitate the use of reasonably efficient algorithms by TARP/500 systems. More specifically, this will require DIB integration for:

- TID/NET pairs as mapped by TARP;
- OSI Routing Domain information to support properly scoped directory searches corresponding to TARP Type 2 PDUs.

To address these requirements, the existing [ANS95] Directory Schema is enhanced to support this new information.

The TARP protocol supports only one-to-one mappings between TIDs and NSAPs, that is, no more than one TID can be associated with a given NSAP and vice-versa. Since [BEL95] defines TARP to be for the purpose of locating a TL1 application over a seven-layer OSI stack, TID information can be represented in the DIB using a TL1 application entity entry without loss of generality.

For those TMN and SONET systems that already have a *tmnNE* object entry in the DIB, all that is required is the addition of a TL1 application entity as described in section 7.2.2. For those systems that do not have a *tmnNE* object entry in the DIB, section 7.2.1 describes how one must be created, either by a DUA on the system itself, or via a proxy agent such as T5GW.

7.2.1 SONET NE Entries

If TL1 application information for a SONET NE is to be added to the DIB, the system in question must already have a *tmnNE* object entry in the DIB. For those systems that do not have sufficient MIB information to create a *tmnNE* object entry, one will have to be created using available TARP information, either by the system itself, or via a proxy agent such as T5GW. In these cases, the *tmnNE* object entry shall contain the following attribute value assignments for its mandatory attributes:

objectClass

The *tmnNE* object class object identifier **id-oc-tmnNE** as defined in [ANS95];

commonName

The TID of the system as would have been returned by TARP;[10]

---

10. An implementation should not assume that the common name for all tmnNE objects is a TID.

entityAddress

The NSAP of the system which is created from the NET as would have been returned by TARP;

managedElementId

An empty *GraphicString* for the *pString* member.

Any additional attributes or attribute values of this *tmnNE* object may be assigned as deemed appropriate.

For *tmnNE* object entries created by T5GW on behalf of others, the *tmnNE* optional attribute *neType* shall be present and shall have the object identifier **id-neType-createdByT5GW** as one of its values.

id-neType-createdByT5GW OBJECT IDENTIFIER ::= ***{<TBD>}***

7.2.2 TL1 Application Entities

A TMN or SONET system that contains a [BEL95] conformant TL1 application entity shall have an *applicationEntity* or *apaeAlias* object entry populated as a subordinate (but not necessarily a direct subordinate) of its *tmnNE* object entry. This is done either by the system itself, or on behalf of the system by a proxy agent such as T5GW.

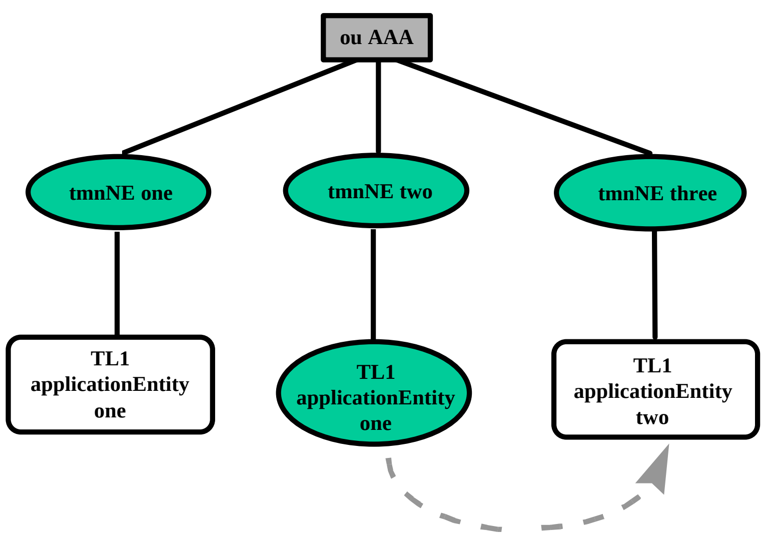

Figure 4- Example TL1 application entities.

When the *applicationEntity* object entry is used it shall contain the following attribute value assignments:

objectClass

The *applicationEntity* object class object identifier **id-oc-applicationEntity** as defined in [ITU93d];

commonName

The TID corresponding to the TL1 application as would have been mapped by TARP;

presentationAddress

The presentation address of the TL1 application, which is created by building the *nAddress* component from the NET as mapped by TARP, and by using the default TL1 selector values as defined in [BEL95, sec. 8.3, pp. 8-20], or as commissioned in the local routing area;

supportedApplicationContext

The object identifier *tl1PeerComm*, as defined in [BEL95, sec. 8.3.7.5, R8-80, pp. 8-24], representing the application context for peer NE/NE TL1 communications.

Any additional attributes or attribute values of this *applicationEntity* object may be assigned as deemed appropriate.

When the *apaeAlias* object entry is used it shall contain the following attribute value assignments:

objectClass

The *apaeAlias* object class object identifier **id-oc-apaeAlias** as defined in [ANS95];

commonName

The TID corresponding to the TL1 application as would have been returned by TARP;

aliasedEntryName

The DN of the *applicationEntity* object entry representing the proxy TL1 application entity.

Figure 4 provides an example DIT structure for *tmnNE* object entries containing TL1 application entities.

8. TID/Address Resolvability Rules

In order for TID/address resolution to work, the following conditions must hold. Note that these conditions are not to be interpreted as conformance requirements but as ramifications of TARP, TARP/500, and RRP as specified.

1. TARP connectivity is required between any system that might request the address for a TID and any system whose TID might be requested, unless T5GW is used. Note that TARP connectivity can be provided across non-TARP systems by provisioning TARP adjacencies.

2. If full TARP connectivity is not provided, a T5GW system is required in each "TARP island" within each IS-IS Level 1 routing area. A TARP island is the maximal set of systems that support TARP and enjoy mutual TARP-connectivity. If a TARP island spans multiple routing areas, there must be a T5GW system in each area. If a routing area contains multiple TARP islands or portions of islands, a T5GW system is required in each island-portion within the area. This is because a T5GW system only serves TARP systems in its area with which it can exchange propagated TARP PDUs. Of course, TARP islands or island-portions that do not contain any systems (OSs or NEs) that require TARP services do not require T5GW systems.

3. A Registration Manager (RM) is required in each Level 1 routing area containing a T5, T5GW, or TMN system. Also, a DSA must be reachable via OSI. (These requirements are inherent to the use of [ANS95].) In addition, DSAs must support DIT extensions specified in this document.

Notes:

A. TARP connectivity is supported by TARP-only, T5, and T5GW systems, or by TARP adjacencies between such systems for routes across non-TARP systems.

B. A T5 system can be deployed at any point in an operating network without affecting the address TID/address resolution of the preexisting systems.