Internet Engineering Task Force (IETF)
Request for Comments: 6320
Category: Standards Track
ISSN: 2070-1721

S. Wadhwa
Alcatel-Lucent
J. Moisand
Juniper Networks
T. Haag
Deutsche Telekom
N. Voigt
Nokia Siemens Networks
T. Taylor, Ed.
Huawei Technologies
October 2011

Protocol for Access Node Control Mechanism in Broadband Networks

Abstract

This document describes the Access Node Control Protocol (ANCP). ANCP operates between a Network Access Server (NAS) and an Access Node (e.g., a Digital Subscriber Line Access Multiplexer (DSLAM)) in a multi-service reference architecture in order to perform operations related to Quality of Service, service, and subscribers. Use cases for ANCP are documented in RFC 5851. As well as describing the base ANCP protocol, this document specifies capabilities for Digital Subscriber Line (DSL) topology discovery, line configuration, and remote line connectivity testing. The design of ANCP allows for protocol extensions in other documents if they are needed to support other use cases and other access technologies.

ANCP is based on the General Switch Management Protocol version 3 (GSMPv3) described in RFC 3292, but with many modifications and extensions, to the point that the two protocols are not interoperable. For this reason, ANCP was assigned a separate version number to distinguish it.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines a new protocol, the Access Node Control Protocol (ANCP), to realize a control plane between a service-oriented layer 3 edge device (the Network Access Server, NAS) and a layer 2 Access Node (e.g., Digital Subscriber Line Access Multiplexer, DSLAM) in order to perform operations related to quality of service (QoS), services, and subscriptions. The requirements for ANCP and the context within which it operates are described in [RFC5851].

ANCP provides its services to control applications operating in the AN and NAS, respectively. This relationship is shown in Figure 1. Specification of the control applications is beyond the scope of this document, but informative partial descriptions are provided as necessary to give a context for the operation of the protocol.

```
      Access Node                              Network Access Server
 +--------------------+                    +--------------------+
 | +----------------+ |                    | +----------------+ |
 | |   AN Control   | |                    | |  NAS Control   | |
 | |  Application   | |                    | |  Application   | |
 | +----------------+ |                    | +----------------+ |
 | +----------------+ |                    | +----------------+ |
 | |   ANCP Agent   | |    ANCP Messages   | |   ANCP Agent   | |
 | |   (AN side)    |<----------------------->|   (NAS side)   | |
 | +----------------+ |                    | +----------------+ |
 +--------------------+                    +--------------------+
```

Figure 1: Architectural Context for the Access Node Control Protocol

At various points in this document, information flows between the control applications and ANCP are described. The purpose of such descriptions is to clarify the boundary between this specification and, for example, [TR-147]. There is no intention to place limits on the degree to which the control application and the protocol implementation are integrated.

This specification specifies ANCP transport over TCP/IP. TCP encapsulation for ANCP is as defined in Section 3.2.

The organization of this document is as follows:

- Sections 1.2 and 1.3 introduce some terminology that will be useful in understanding the rest of the document.

- Section 2 provides a description of the access networks within which ANCP will typically be deployed.

- o Section 3 specifies generally applicable aspects of ANCP.

- o Section 4 specifies some messages and TLVs intended for use by multiple capabilities spanning multiple technologies.

- o Section 5 and the three following sections describe and specify the ANCP implementation of three capabilities applicable to the control of DSL access technology: topology discovery, line configuration, and remote line connectivity testing.

- o Section 9 is the IANA Considerations section. This section defines a number of new ANCP-specific registries as well as the joint GSMP/ANCP version registry mentioned below.

- o Section 11 addresses security considerations relating to ANCP, beginning with the requirements stated in [RFC5713].

1.1. Historical Note

Initial implementations of the protocol that became ANCP were based on the General Switch Management Protocol version 3 (GSMPv3) [RFC3292]. The ANCP charter required the Working Group to develop its protocol based on these implementations. In the end, ANCP introduced so many extensions and modifications to GSMPv3 that the two protocols are not interoperable. Nevertheless, although this specification has no normative dependencies on [RFC3292], the mark of ANCP’s origins can be seen in the various unused fields within the ANCP message header.

Early in ANCP’s development, the decision was made to use the same TCP port and encapsulation as GSMPv3, and by the time ANCP was finished, it was too late to reverse that decision because of existing implementations. As a result, it is necessary to have a way for an ANCP peer to quickly distinguish ANCP from GSMP during initial adjacency negotiations. This has been provided by a joint registry of GSMP and ANCP version numbers. GSMP has version numbers 1 through 3. ANCP has the initial version number 50.

1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.3. Terminology

This section repeats some definitions from [RFC5851], but it also adds definitions for terms used only in this document.

Access Node (AN): [RFC5851] Network device, usually located at a service provider central office or street cabinet that terminates access (local) loop connections from subscribers. In case the access loop is a Digital Subscriber Line (DSL), the Access Node provides DSL signal termination and is referred to as a DSL Access Multiplexer (DSLAM).

Network Access Server (NAS): [RFC5851] Network element that aggregates subscriber traffic from a number of Access Nodes. The NAS is an enforcement point for policy management and IP QoS in the access network. It is also referred to as a Broadband Network Gateway (BNG) or Broadband Remote Access Server (BRAS).

Home Gateway (HGW): Network element that connects subscriber devices to the Access Node and the access network. In the case of DSL, the Home Gateway is a DSL network termination that may operate either as a layer 2 bridge or as a layer 3 router. In the latter case, such a device is also referred to as a Routing Gateway (RG).

ANCP agent: A logical entity that implements ANCP in the Access Node (AN-side) or NAS (NAS-side).

Access Node control adjacency: (modified from [RFC5851]) The relationship between the AN-side ANCP agent and the NAS-side ANCP agent for the purpose of exchanging Access Node Control Protocol messages. The adjacency may be either up or down, depending on the result of the Access Node Control adjacency protocol operation.

ANCP capability: A specific set of ANCP messages, message content, and procedures required to implement a specific use case or set of use cases. Some ANCP capabilities are applicable to just one access technology while others are technology independent. The capabilities applicable to a given ANCP adjacency are negotiated during adjacency startup.

Type-Length-Value (TLV): A data structure consisting of a 16-bit type field, a sixteen-bit length field, and a variable-length value field padded to the nearest 32-bit word boundary, as described in Section 3.6.2. The value field of a TLV can contain other TLVs. An IANA registry is maintained for values of the ANCP TLV Type field.

Net data rate: [RFC5851] Defined by ITU-T G.993.2 [G.993.2], Section 3.39, i.e., the portion of the total data rate that can be used to transmit user information (e.g., ATM cells or Ethernet frames). It excludes overhead that pertains to the physical transmission mechanism (e.g., trellis coding in the case of DSL). It includes

TPS-TC (Transport Protocol Specific - Transmission Convergence) encapsulation; this is zero for ATM encapsulation and non-zero for 64/65 encapsulation.

Line rate: [RFC5851] Defined by ITU-T G.993.2. It contains the complete overhead including Reed-Solomon and trellis coding.

DSL multi-pair bonding: Method for bonding (or aggregating) multiple xDSL access lines into a single bidirectional logical link, henceforth referred to in this document as "DSL bonded circuit". DSL "multi-pair" bonding allows an operator to combine the data rates on two or more copper pairs, and deliver the aggregate data rate to a single customer. ITU-T recommendations G.998.1 [G.998.1] and G.998.2 [G.998.2], respectively, describe ATM- and Ethernet-based multi-pair bonding.

2. Broadband Access Aggregation

2.1. ATM-Based Broadband Aggregation

The end-to-end DSL network consists of network service provider (NSP) and application service provider (ASP) networks, regional/access network, and customer premises network. Figure 2 shows ATM broadband access network components.

The regional/access network consists of the regional network, Network Access Server (NAS), and the access network as shown in Figure 2. Its primary function is to provide end-to-end transport between the customer premises and the NSP or ASP.

The Access Node terminates the DSL signal. It may be in the form of a DSLAM in the central office, a remote DSLAM, or a Remote Access Multiplexer (RAM). The Access Node is the first point in the network where traffic on multiple DSL access lines will be aggregated onto a single network.

The NAS performs multiple functions in the network. The NAS is the aggregation point for subscriber traffic. It provides aggregation capabilities (e.g., IP, PPP, ATM) between the Regional/Access Network and the NSP or ASP. These include traditional ATM-based offerings and newer, more native IP-based services. This includes support for Point-to-Point Protocol over ATM (PPPoA) and PPP over Ethernet (PPPoE), as well as direct IP services encapsulated over an appropriate layer 2 transport.

Beyond aggregation, the NAS is also the enforcement point for policy management and IP QoS in the regional/access networks. To allow IP QoS support over an existing non-IP-aware layer 2 access network without using multiple layer 2 QoS classes, a mechanism based on hierarchical scheduling is used. This mechanism, defined in [TR-059], preserves IP QoS over the ATM network between the NAS and the Routing Gateway (RG) at the edge of the subscriber network, by carefully controlling downstream traffic in the NAS, so that significant queuing and congestion do not occur farther down the ATM network. This is achieved by using a Diffserv-aware hierarchical scheduler in the NAS that will account for downstream trunk bandwidths and DSL synchronization rates.

[RFC5851] provides detailed definitions of the functions of each network element in the broadband reference architecture.

```
                              Access                   Customer
                       <--- Aggregation -->  <------- Premises ------->
                              Network                   Network

                       +------------------+ +-------------------------+
   +---------+   +---+ | +-----+ +------+ | |+-----+ +---+ +---------+ |
NSP|         | +-|NAS|-| |ATM  |-|Access| --||DSL  |-|HGW|-|Subscriber||
---+ Regional| | +---+ | +-----+ | Node | | ||Modem| +---+ |Devices   ||
   |Broadband| | +---+ |         +------+ | |+-----+       +----------+|
ASP|Network  |-+-|NAS| +--------------|---+ +-------------------------+
---+         | | +---+                |     +-------------------------+
   |         | | +---+                |     |+-----+ +---+ +----------+|
   +---------+ +-|NAS|                +-----|| DSL |-|HGW|-|Subscriber||
                 +---+                      ||Modem| +---+ |Devices   ||
                                            |+-----+       +----------+|
                                            +-------------------------+
 HGW: Home Gateway
 NAS: Network Access Server
```

Figure 2: ATM Broadband Aggregation Topology

2.2. Ethernet-Based Broadband Aggregation

The Ethernet aggregation network architecture builds on the Ethernet bridging/switching concepts defined in IEEE 802. The Ethernet aggregation network provides traffic aggregation, class of service distinction, and customer separation and traceability. VLAN tagging, defined in [IEEE802.1Q] and enhanced by [IEEE802.1ad], is used as the standard virtualization mechanism in the Ethernet aggregation network. The aggregation devices are "provider edge bridges" defined in [IEEE802.1ad].

Stacked VLAN tags provide one possible way to create an equivalent of "virtual paths" and "virtual circuits" in the aggregation network. The "outer" VLAN can be used to create a form of "virtual path" between a given DSLAM and a given NAS. "Inner" VLAN tags create a form of "virtual circuit" on a per-DSL-line basis. This is the 1:1 VLAN allocation model. An alternative model is to bridge sessions from multiple subscribers behind a DSLAM into a single VLAN in the aggregation network. This is the N:1 VLAN allocation model. Section 1.6 of [TR-101] provides brief definitions of these two models, while Section 2.5.1 describes them in more detail.

3. Access Node Control Protocol -- General Aspects

This section specifies aspects of the Access Node Control Protocol (ANCP) that are generally applicable.

3.1. Protocol Version

ANCP messages contain an 8-bit protocol version field. For the protocol version specified in this document, the value of that field MUST be set to 50.

3.2. ANCP Transport

This document specifies the use of TCP / IPsec+IKEv2 / IP for transport of ANCP messages. For further discussion of the use of IPsec and IKEv2, see Section 11. The present section deals with the TCP aspects. Other specifications may introduce additional transports in the future.

> In the case of ATM access, a separate permanent virtual circuit (PVC) that is a control channel and is capable of transporting IP MAY be configured between the NAS and the AN for ANCP messages.
>
> In the case of an Ethernet access/aggregation network, a typical practice is to send the Access Node Control Protocol messages over a dedicated Ethernet virtual LAN (VLAN) using a separate VLAN identifier (VLAN ID).

When transported over TCP, ANCP messages MUST use an encapsulation consisting of a 4-byte header field prepended to the ANCP message as shown in Figure 3.

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    Identifier (0x880C)        |           Length              |
      |-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      ~                         ANCP Message                          ~
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Encapsulation of ANCP Messages over TCP/IP

The fields of the encapsulating header are as follows:

Identifier (16 bits): This identifies a GSMP or ANCP message. It MUST be set to 0x880C.

Length (16 bits): Total length of the ANCP message in bytes, not including the 4-byte encapsulating header.

The Access Node MUST initiate the TCP session to the NAS, using destination port 6068.

> This is necessary to avoid static address provisioning on the NAS for all the ANs that are being served by the NAS. It is easier to configure a given AN with the single IP address of the NAS that serves the AN.

The NAS MUST listen on port 6068 for incoming connections from the Access Nodes.

In the event of an ANCP transport protocol failure, all pending ANCP messages destined to the disconnected recipient SHOULD be discarded until the transport connection is re-established.

3.3. Encoding of Text Fields

In ANCP, all text fields use UTF-8 encoding [RFC3629]. Note that US-ASCII characters have the same representation when coded as UTF-8 as they do when coded according to [US_ASCII].

When extracting text fields from a message, the ANCP agent MUST NOT assume that the fields are zero-terminated.

3.4. Treatment of Reserved and Unused Fields

ANCP messages contain a number of fields that are unused or reserved. Some fields are always unused (typically because they were inherited from GSMPv3 but are not useful in the ANCP context). Others are reserved in the current specification, but are provided for flexibility in future extensions to ANCP. Both reserved and unused fields MUST be set to zeroes by the sender and MUST be ignored by the receiver.

Unused bits in a flag field are shown in figures as ’x’. The above requirement (sender set to zero, receiver ignore) applies to such unused bits.

3.5. The ANCP Adjacency Protocol

ANCP uses the adjacency protocol to synchronize the NAS and Access Nodes and maintain the ANCP session. After the TCP connection is established, adjacency protocol messages MUST be exchanged as specified in this section. ANCP messages other than adjacency protocol messages MUST NOT be sent until the adjacency protocol has achieved synchronization.

3.5.1. ANCP Adjacency Message Format

The ANCP adjacency message format is shown in Figure 4 below.

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |     Version   | Message Type  |     Timer     |M|     Code    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                          Sender Name                          |
      +                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                               |                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
      |                         Receiver Name                         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                          Sender Port                          |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Receiver Port                         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | PType |P Flag |               Sender Instance                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Partition ID  |              Receiver Instance                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Reserved      | # of Caps     | Total Length                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      ~                   Capability Fields                           ~
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: ANCP Adjacency Message Format

The fields of the ANCP adjacency message are as follows:

Version (8 bits): ANCP version, which is subject to negotiation. This is the key parameter by means of which ANCP messages can be distinguished from GSMP messages received over the same port.

Message Type (8 bits): Always has value 10 (adjacency protocol).

Timer (8 bits): The Timer field is used to negotiate the timer value used in the adjacency protocol with the peer. The timer specifies the nominal time between periodic adjacency protocol messages. It is a constant for the duration of an ANCP session. The Timer field is specified in units of 100 ms, with a default value of 250 (i.e., 25 seconds).

M flag (1 bit): Used in the SYN message to prevent the NAS from synchronizing with another NAS and the AN from synchronizing with another AN. In the SYN message, it is always set to 1 by the NAS and to 0 by the AN. In other adjacency message types, it is always set to 0 by the sender and ignored by the receiver.

Code (7 bits): The adjacency protocol message type. It MUST have one of the following values:

Code = 1: SYN;

Code = 2: SYNACK;

Code = 3: ACK;

Code = 4: RSTACK.

Sender Name (48 bits): For the SYN, SYNACK, and ACK messages, is the identifier of the entity sending the message. The Sender Name is a 48-bit quantity that is unique within the operational context of the device. A 48-bit IEEE 802 Media Access Control (MAC) address, if available, may be used for the Sender Name. If the Ethernet encapsulation is used, the Sender Name MUST be the Source Address from the MAC header. For the RSTACK message, the Sender Name field is set to the value of the Receiver Name field from the incoming message that caused the RSTACK message to be generated.

Receiver Name (48 bits) For the SYN, SYNACK, and ACK messages, is the name of the entity that the sender of the message believes is at the far end of the link. If the sender of the message does not know the name of the entity at the far end of the link, this field SHOULD be set to zero. For the RSTACK message, the Receiver Name field is set to the value of the Sender Name field from the incoming message that caused the RSTACK message to be generated.

Sender Port (32 bits): For the SYN, SYNACK, and ACK messages, is the local port number of the link across which the message is being sent. For the RSTACK message, the Sender Port field is set to the value of the Receiver Port field from the incoming message that caused the RSTACK message to be generated.

Receiver Port (32 bits): For the SYN, SYNACK, and ACK messages, is what the sender believes is the local port number for the link, allocated by the entity at the far end of the link. If the sender of the message does not know the port number at the far end of the link, this field SHOULD be set to zero. For the RSTACK message, the Receiver Port field is set to the value of the Sender Port field from the incoming message that caused the RSTACK message to be generated.

PType (4 bits): PType is used to specify if partitions are used and how the Partition ID is negotiated.

Type of partition being requested:

0 - no partition;

1 - fixed partition request;

2 - fixed partition assigned.

P Flag (4 bits): Used to indicate the type of partition request.

1 - new adjacency;

2 - recovered adjacency.

In case of a conflict between the peers’ views of the value of the P Flag, the lower value is used.

Sender Instance (24 bits): For the SYN, SYNACK, and ACK messages, is the sender’s instance number for the link to the peer. It is used to detect when the link comes back up after going down or when the identity of the entity at the other end of the link changes. The instance number is a 24-bit number that is guaranteed to be unique within the recent past and to change when the link or node comes back up after going down. Zero is not a valid instance number. For the RSTACK message, the Sender Instance field is set to the value of the Receiver Instance field from the incoming message that caused the RSTACK message to be generated.

Partition ID (8 bits): Field used to associate the message with a specific partition of the AN. The value of this field is negotiated during the adjacency procedure. The AN makes the final decision, but will consider a request from the NAS. If the AN does not support partitions, the value of this field MUST be 0. Otherwise, it MUST be non-zero.

Receiver Instance (24 bits): For the SYN, SYNACK, and ACK messages, is what the sender believes is the current instance number for the link, allocated by the entity at the far end of the link. If the sender of the message does not know the current instance number at the far end of the link, this field SHOULD be set to zero. For the RSTACK message, the Receiver Instance field is set to the value of the Sender Instance field from the incoming message that caused the RSTACK message to be generated.

Reserved (8 bits): Reserved for use by a future version of this specification.

# of Caps (8 bits): Indicates the number of Capability fields that follow.

Total Length (16 bits): Indicates the total number of bytes occupied by the Capability fields that follow.

Capability Fields: Each Capability field indicates one ANCP capability supported by the sender of the adjacency message. Negotiation of a common set of capabilities to be supported within the ANCP session is described below. The detailed format of a Capability field is shown in Figure 5 and described below.

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |     Capability Type           |   Capability Length           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      ~                                                               ~
      ~                    Capability Data                            ~
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Capability Field

The sub-fields of this structure are as follows:

Capability Type (16 bits): Indicates the specific capability supported. An IANA registry exists for values of this sub-field. The values specified by this document are listed below.

Capability Length (16 bits): The number of bytes of data contained in the Capability Data sub-field, excluding padding. If the definition of a particular capability includes no capability data, the value of the Capability Length sub-field is zero.

Capability Data (as indicated by Capability Length): Contains data associated with the capability as specified for that capability. If the definition of a particular capability includes no capability data, the Capability Data sub-field is absent (has zero length). Otherwise, the Capability Data sub-field MUST be padded with zeroes as required to terminate on a 4-byte word boundary. The possibility of specifying capability data provides the flexibility to advertise more than the mere presence or absence of a capability if needed.

The following capabilities are defined for ANCP as applied to DSL access:

- Capability Type: DSL Topology Discovery = 0x01

  Access technology: DSL

  Length (in bytes): 0

  Capability Data: NULL

  For the detailed protocol specification of this capability, see Section 6.

- Capability Type: DSL Line Configuration = 0x02

  Access technology: DSL

  Length (in bytes): 0

  Capability Data: NULL

  For the detailed protocol specification of this capability, see Section 7.

- Capability Type: DSL Remote Line Connectivity Testing = 0x04

  Access technology: DSL

  Length (in bytes): 0

  Capability Data: NULL

  For the detailed protocol specification of this capability, see Section 8.

In addition to the adjacency messages whose format is shown in Figure 6, ANCP adjacency procedures use the Adjacency Update message (Figure 6) to inform other NASs controlling the same AN partition when a particular NAS joins or loses an adjacency with that partition.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Version    | Message Type  | Result|       Code            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Partition ID  |           Transaction Identifier              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |I|      SubMessage Number      |           Length              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: The Adjacency Update Message

The Adjacency Update message is identical to the general ANCP message header described in Section 3.6, but the field settings are in part specific to the Adjacency Update message. The fields in this message are as follows:

Version (8 bits): The ANCP version negotiated and running in this adjacency.

Message Type (8 bits): Always 85.

Result (4 bits): Set to Ignore (0).

Code (12 bits): Set to the total number of adjacencies currently established on this partition, from the point of view of the AN.

Partition ID (8 bits): The partition identifier of the partition for which this notification is being sent.

Transaction Identifier (24 bits): MUST be set to 0.

I (1 bit), SubMessage number (15 bits): Set as described in Section 3.6.1.7.

Length (16 bits): Set as described in Section 3.6.1.8.

3.5.2. ANCP Adjacency Procedures

3.5.2.1. Overview

The ANCP adjacency protocol operates symmetrically between the NAS and the AN. In the absence of errors or race conditions, each peer sends a SYN message, receives a SYNACK message in acknowledgement, and completes the establishment of the adjacency by sending an ACK message. Through this exchange, each peer learns the values of the Name, Port, and Instance parameters identifying the other peer, and the two peers negotiate the values of the Version, Timer, P Flag, and Partition ID parameters and the set of capabilities that the adjacency will support.

Once the adjacency has been established, its liveness is periodically tested. The peers engage in an ACK message exchange at a frequency determined by the negotiated value of the Timer field.

If an inconsistency, loss of contact, or protocol violation is detected, the detecting peer can force a restart of the synchronization process by sending an RSTACK message to the other end.

Once an adjacency has been established, if more than one NAS has established an adjacency to the same partition, then the AN sends an Adjacency Update message to each such NAS to let it know how many established adjacencies the partition currently supports. Similarly, if an adjacency is lost, the AN sends an Adjacency Update message to each of the remaining adjacent NASs to let them know about the change in status.

3.5.2.2. Adjacency Protocol State Machine

The adjacency protocol is described by the following rules and state tables. It begins with the sending of a SYN by each end as soon as the transport connection has been established. If at any point the operations A, B, C, or "Verify Adjacent State" defined below detect a mismatch, a log SHOULD be generated, identifying the fields concerned and the expected and received values for each.

The rules and state tables use the following operations:

- The "Record Adjacency State" operation is defined in Section 3.5.2.3.2.

- The "Verify Adjacency State" operation consists of verifying that the contents of the incoming SYNACK message match the adjacency state values previously recorded.

- The procedure "Reset the link" is defined as:

  1. Generate a new instance number for the link.

  2. Delete the peer verifier (set to zero the values of Sender Instance, Sender Port, and Sender Name previously stored by the "Record Adjacency State" operation).

  3. Send a SYN message (Section 3.5.2.3.1).

4. Enter the SYNSENT state.

- The state tables use the following Boolean terms and operators.

  A. The Sender Instance in the incoming message matches the value stored from a previous message by the "Record Adjacency State" operation.

  B. The Sender Instance, Sender Port, Sender Name, and Partition ID fields in the incoming message match the values stored from a previous message by the "Record Adjacency State" operation.

  C. The Receiver Instance, Receiver Port, Receiver Name, and Partition ID fields in the incoming message match the values of the Sender Instance, Sender Port, Sender Name, and Partition ID currently sent in outgoing SYN, SYNACK, and ACK messages, except that the NAS always accepts the Partition ID value presented to it in a SYN or SYNACK message.

  "&&" Represents the logical AND operation.

  "||" Represents the logical OR operation.

  "!" Represents the logical negation (NOT) operation.

- A timer is required for the periodic generation of SYN, SYNACK, and ACK messages. The value of the timer is negotiated in the Timer field. The period of the timer is unspecified, but a value of 25 seconds is suggested. Note that since ANCP uses a reliable transport protocol, the timer is unlikely to expire in any state other than ESTAB.

  There are two independent events: the timer expires, and a packet arrives. The processing rules for these events are:

  Timer Expires: Reset Timer

  If state = SYNSENT Send SYN

  If state = SYNRCVD Send SYNACK

  If state = ESTAB Send ACK

```
      Packet Arrives:

         If incoming message is an RSTACK:

            If (A && C && !SYNSENT) Reset the link

            Else discard the message.

         If incoming message is a SYN, SYNACK, or ACK:

            Response defined by the following state tables.

         If incoming message is any other ANCP message and state !=
         ESTAB:

            Discard incoming message.

            If state = SYNSENT Send SYN (Note 1)

            If state = SYNRCVD Send SYNACK (Note 1)

      Note 1: No more than two SYN or SYNACK messages should be sent
      within any time period of length defined by the timer.
```

- State synchronization across a link is considered to be achieved when the protocol reaches the ESTAB state. All ANCP messages, other than adjacency protocol messages, that are received before synchronization is achieved will be discarded.

3.5.2.2.1. State Tables

State: SYNSENT

Condition	Action	New State
SYNACK && C	Update Peer Verifier; Send ACK	ESTAB
SYNACK && !C	Send RSTACK	SYNSENT
SYN	Update Peer Verifier; Send SYNACK	SYNRCVD
ACK	Send RSTACK	SYNSENT

State: SYNRCVD

Condition	Action	New State
SYNACK && C	Verify Adjacency State; Send ACK	ESTAB
SYNACK && !C	Send RSTACK	SYNRCVD
SYN	Record Adjacency State; Send SYNACK	SYNRCVD
ACK && B && C	Send ACK	ESTAB
ACK && !(B && C)	Send RSTACK	SYNRCVD

State: ESTAB

Condition	Action	New State
SYN || SYNACK	Send ACK (Note 2)	ESTAB
ACK && B && C	Send ACK (Note 3)	ESTAB
ACK && !(B && C)	Send RSTACK	ESTAB

Note 2: No more than two ACKs should be sent within any time period of length defined by the timer. Thus, one ACK MUST be sent every time the timer expires. In addition, one further ACK may be sent between timer expirations if the incoming message is a SYN or SYNACK. This additional ACK allows the adjacency protocol to reach synchronization more quickly.

Note 3: No more than one ACK should be sent within any time period of length defined by the timer.

3.5.2.3. The Adjacency Protocol SYN Message

3.5.2.3.1. Action by the Sender

The SYN message is sent in accordance with the state tables just described. The sender sets the individual fields as follows:

Version: SHOULD be set to the highest version of ANCP that the sender supports.

Message Type: MUST be set to 10.

Timer: SHOULD be set to the value configured in the AN or NAS sending the message.

M Flag: MUST be set to 1 by the NAS, and 0 by the AN.

Code: MUST be set to 1 (SYN).

Sender Name: Set as described in Section 3.5.1.

Receiver Name: SHOULD be set to 0.

Sender Port: Set as described in Section 3.5.1.

Receiver Port: SHOULD be set to 0.

PType: Set according to the following rules:

Settings by the AN:

0 - the AN does not support partitions;

2 - the value of Partition ID contained in this message is assigned to the current partition.

Settings by the NAS:

0 - the NAS leaves the decision on partitioning to the AN (RECOMMENDED setting);

1 - the NAS requests that the AN use the value of Partition ID contained in this message for the current partition. The NAS MAY use this setting even if it has already received a SYN message from the AN, provided that the AN has indicated support for partitions. The NAS MUST be prepared to use whatever value it receives in a subsequent SYN or SYNACK message, even if this differs from the requested value.

P Flag: Set to the mode of adjacency setup (new adjacency vs. recovered adjacency) requested by the sender. Warning: setting P Flag=1 runs the risk of state mismatch because ANCP does not provide the means for the NAS to audit the current state of the AN.

Sender Instance: Set as described in Section 3.5.1.

Partition ID: MUST be set to 0 if PType=0; otherwise, set to the assigned or requested partition identifier value.

Receiver Instance: SHOULD be set to 0.

# of Caps: MUST be set to the number of Capability fields that follow.

Total Length: MUST be set to the total number of bytes in the Capability fields that follow.

Capability Fields: One Capability field MUST be present for each ANCP capability for which the sender wishes to advertise support.

3.5.2.3.2. Action by the Receiver

Upon receiving a validly formed SYN message, the receiver first checks the value of the Version field. If this value is not within the range of ANCP versions that the receiver supports, the message MUST be silently ignored. Similarly, the message is silently ignored if the M flag is 0 and the receiver is an AN or if the M flag is 1 and the receiver is a NAS. If these checks are passed and the receiver is in ESTAB state, it returns an ACK (as indicated by the ESTAB state table in Section 3.5.2.2.1). The contents of the ACK MUST reflect the adjacency state as previously recorded by the receiver.

Otherwise, the receiver MUST perform the "Record Adjacency State" operation by recording the following fields:

Version: The supported Version value received in the SYN message. This value MUST be used for all subsequent ANCP messages sent during the life of the adjacency.

Timer: The larger of the Timer value received in the SYN message and the value with which the receiver is configured.

Sender Name: The value of the Sender Name field in the SYN message just received.

Receiver Name: The value used by the receiver in the Sender Name field of SYN, SYNACK, and ACK messages it sends in this adjacency.

Sender Port: The value of the Sender Port field in the SYN message just received.

Receiver Port: The value used by the receiver in the Sender Port field of SYN, SYNACK, and ACK messages it sends in this adjacency.

Sender Instance: The value of the Sender Instance field in the SYN message just received.

P Flag: The lesser of the value determined by local policy and the value received in the SYN message. That is, preference is given to "0 - New adjacency" if there is a conflict.

Partition ID: If the SYN receiver is the AN, this is set to 0 if the AN does not support partitions or to the non-zero value of the partition identifier it chooses to assign otherwise. If the SYN receiver is the NAS, this is set to the value of the Partition ID field copied from the SYN.

Receiver Instance: The value used by the receiver in the Sender Instance field of SYN, SYNACK, and ACK messages it sends in this adjacency.

Capabilities: The set of ANCP capabilities that were offered in the SYN and are supported by the receiver.

3.5.2.4. The Adjacency Protocol SYNACK Message

3.5.2.4.1. Action by the Sender

The SYNACK is sent in response to a successfully received SYN message, as indicated by the state tables. The Version, Timer, P Flag, and Partition ID fields MUST be populated with the values recorded as part of adjacency state. The # of Caps, Total Length, and Capability fields MUST also be populated in accordance with the Capabilities recorded as part of adjacency state. The remaining fields of the SYNACK message MUST be populated as follows:

Message Type: MUST be 10.

M flag: MUST be set to 0.

Code: MUST be 2 (SYNACK).

PType: MUST be 0 if the Partition ID value is 0 or 2 if the Partition ID value is non-zero.

Sender Name: MUST be set to the Receiver Name value recorded as part of adjacency state.

Receiver Name: MUST be set to the Sender Name value recorded as part of adjacency state.

Sender Port: MUST be set to the Receiver Port value recorded as part of adjacency state.

Receiver Port: MUST be set to the Sender Port value recorded as part of adjacency state.

Sender Instance: MUST be set to the Receiver Instance value recorded as part of adjacency state.

Receiver Instance: MUST be set to the Sender Instance value recorded as part of adjacency state.

If the set of capabilities recorded in the adjacency state is empty, then after sending the SYNACK the sender MUST raise an alarm to management, halt the adjacency procedure, and tear down the TCP session if it is not being used by another adjacency. The sender MAY also terminate the IPsec security association if no other adjacency is using it.

3.5.2.4.2. Action by the Receiver

As indicated by the state tables, the receiver of a SYNACK first checks that the Receiver Name, Receiver Port, and Receiver Instance values match the Sender Name, Sender Port, and Sender Instance values it sent in SYN message that is being acknowledged. The AN also checks that the PType and Partition ID match. If any of these checks fail, the receiver sends an RSTACK as described in Section 3.5.2.6.1.

The receiver next checks whether the set of capabilities provided in the SYNACK is empty. If so, the receiver MUST raise an alarm to management and halt the adjacency procedure.

Assuming that the SYNACK passes these checks, two cases arise. The first possibility is that the receiver has already recorded adjacency state. This will occur if the SYNACK is received while the receiver is in SYNRCVD state. In this case, the Version, Timer, Sender Name, Sender Port, Sender Instance, P Flag, and capability-related fields in the SYNACK MUST match those recorded as part of adjacency state. If a mismatch is detected, the receiver sends an RSTACK. This is the "Verify Adjacency State" procedure shown in the SYNRCVD state table.

If, on the other hand, the SYNACK is received while the receiver is in SYNSENT state, the receiver MUST record session state as described in Section 3.5.2.3.2.

In either case, if the receiver is the NAS, it MUST accept the Partition ID value provided in the SYNACK, updating its recorded adjacency state if necessary.

3.5.2.5. The Adjacency Protocol ACK Message

3.5.2.5.1. Actions by the Sender

As indicated by the state tables, the ACK message is sent in a number of different circumstances. The main-line usages are as a response to SYNACK, leading directly to the ESTAB state, and as a periodic test of liveness once the ESTAB state has been reached.

The sender MUST populate the ACK from recorded adjacency state, exactly as described in Section 3.5.2.4.1. The only difference is that Code MUST be set to 3 (ACK).

3.5.2.5.2. Actions by the Receiver

The required actions by the receiver are specified by the state tables. In addition to the checks B and C, the receiver SHOULD verify that the remaining contents of the ACK match the recorded adjacency state at the receiver. If that check fails, the receiver MUST send an RSTACK as described in Section 3.5.2.6.1.

Once the adjacency has been established, either peer can initiate the ACK exchange that tests for liveness. To meet the restrictions on ACK frequency laid down in the notes to the state tables, it is desirable that only one such exchange occur during any one interval. Hence, if a peer receives an ACK when in ESTAB state, it MUST reply to that ACK as directed by the state tables, but SHOULD NOT initiate another ACK exchange in the same interval. To meet this objective, the receiver MUST reset its timer when it receives an ACK while in ESTAB state.

> It is, of course, possible that two exchanges happen because of race conditions.

3.5.2.6. The Adjacency Protocol RSTACK Message

3.5.2.6.1. Action by the Sender

The RSTACK is sent in response to various error conditions as indicated by the state tables. In general, it leads to a restart of adjacency negotiations (although this takes a few steps when the original sender of the RSTACK is in ESTAB state).

As indicated in Section 3.5.1, the Sender Name, Port, and Instance fields in the RSTACK MUST be copied from the Receiver, Name, Port, and Instance fields in the message that caused the RSTACK to be sent. Similarly, the Receiver identifier fields in the RSTACK MUST be copied from the corresponding Sender identifier fields in the message that triggered the RSTACK.

If the sender has recorded adjacency state, the Version, Timer, PType, P Flag, Partition ID, and capability-related fields SHOULD be set based on the recorded adjacency state. Otherwise, they SHOULD be the same as the sender would send in a SYN message. The Message Type MUST be 10, the M flag MUST be 0, and Code MUST be 4 (RSTACK).

3.5.2.6.2. Action by the Receiver

The receiver of an RSTACK MAY attempt to diagnose the problem that caused the RSTACK to be generated by comparing its own adjacency state with the contents of the RSTACK. However, the primary purpose of the RSTACK is to trigger action as prescribed by Section 3.5.2.2.

3.5.2.7. Loss of Synchronization

Loss of synchronization MAY be declared if after synchronization is achieved:

o no valid ANCP messages are received in any period of time in excess of three times the value of the Timer field negotiated in the adjacency protocol messages, or

o a mismatch in adjacency state is detected.

In either case, the peer detecting the condition MUST send an RSTACK to the other peer, as directed in Section 3.5.2.6.1, in order to initiate resynchronization.

While re-establishing synchronization with a controller, a switch SHOULD maintain its connection state, deferring the decision about resetting the state until after synchronization is re-established.

Once synchronization is re-established, the decision about resetting the connection state SHOULD be made based on the negotiated value of the P Flag.

3.6. ANCP General Message Formats

This section describes the general format of ANCP messages other than the adjacency messages. See Figure 7.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Version    | Message Type  | Result|      Result Code      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Partition ID  |            Transaction Identifier             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |I|      SubMessage Number      |           Length              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                         Message Payload                       ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: ANCP General Message Format

3.6.1. The ANCP Message Header

A complete explanation of the ANCP general message header fields follows.

3.6.1.1. Version Field (8 bits)

This field carries the version of ANCP that was agreed upon for the session during adjacency negotiation.

3.6.1.2. Message Type Field (8 bits)

This field indicates the ANCP message type. Message type values are registered in an IANA registry.

3.6.1.3. Result Field (4 bits)

In request messages, the Result field indicates the circumstances under which a response is required. ANCP specifies what Result value each request message type should have. In responses, the Result field indicates either Success (0x3) or Failure (0x4), as the case may be.

Ignore: Res = 0x0 - Treat this field as a "no operation" and follow the response procedures specified for the received message type.

Nack: Res = 0x1 - Result value indicating that a response is expected to the request only in cases of failure caused during the processing of the message contents or of the contained directive(s).

AckAll: Res = 0x2 - Result value indicating that a response to the message is requested in all cases.

Success: Res = 0x3 - Result value indicating that this is a response and that the request was executed successfully. The Result Code field for a successful result is typically 0, but it MAY take on other values as specified for particular message types.

Failure: Res = 0x4 - Result value indicating that this is a response and that the request was not executed successfully. The receiver of the response SHOULD take further action as indicated by the Result Code value and any diagnostic data contained in a Status-Info TLV included in the response.

3.6.1.4. Result Code Field (12 bits)

This field gives further information concerning the result in a response message. It is mostly used to pass an error code in a failure response, but it can also be used to give further information in a success response message or an event message. In a request message, the Result Code field is not used and MUST be set to 0x0 (No result).

A number of Result Code values are specified below. Specification of additional Result Code values in extensions or updates to this document MUST include the following information:

- Result Code value;

- One-line description;

- Where condition detected (control application or ANCP agent);

- Further description (if any);

- Required additional information in the response message;

- Target (control application or ANCP agent at the peer that sent the original request);

- Action RECOMMENDED for the receiving ANCP agent.

In addition to any suggested action in the text that follows, a count of the number of times a given non-zero Result Code value was received SHOULD be provided for management. Where an action includes the re-sending of a request, a given request SHOULD NOT be re-sent more than once.

This document specifies the following Result Code values.

Result Code value: 0x2

* One-line description: Invalid request message

* Where condition detected: ANCP agent

* Further description: The request was a properly formed message that violates the protocol through its timing or direction of transmission. The most likely reason for this outcome in the field will be a race condition.

* Required additional information in the response message: None, if the response message is of the same type as the request. As specified in Section 4.2, if the response message is a Generic Response message.

* Target: ANCP agent at the peer that sent the original request

* Action RECOMMENDED for the receiving ANCP agent: The original request MAY be re-sent once only after a short delay. Inform the control application with appropriate identification of the failed transaction if the second attempt fails or no second attempt is made.

Result Code value: 0x6

* One-line description: One or more of the specified ports are down

* Where condition detected: Control application

* Further description (if any): This Result Code value indicates a state mismatch between the NAS and AN control applications, possibly due to a race condition.

- Required additional information in the response message: If the request identified multiple access lines or the response is a Generic Response message, then the response MUST contain a Status-Info TLV encapsulating TLV(s) containing the line identifier(s) of the access lines that are not operational.

- Target: Control application at the peer that sent the original request

- Action RECOMMENDED for the receiving ANCP agent: Indicate the error and forward the line identifier(s) to the control application.

Result Code value: 0x13

- One-line description: Out of resources

- Where condition detected: ANCP protocol layer or control application

- Further description (e.g., memory exhausted): This Result Code value MUST be reported only by the AN, and indicates a condition that is probably unrelated to specific access lines (although it may be related to the specific request).

- Required additional information in the response message: None, if the response message is of the same type as the request. As specified in Section 4.2, if the response message is a Generic Response message.

- Target: ANCP agent at the peer that sent the original request

- Action RECOMMENDED for the receiving ANCP agent: If the NAS receives this Result Code value from multiple requests for the same AN in a short interval, it SHOULD reduce the rate at which it sends requests in proportion to the rate at which requests are failing with Result Code = 19. It MAY retry individual requests. If only a specific request is failing with Result Code = 19, the ANCP agent in the NAS MAY request the control application to decompose the request into simpler components if this is possible.

Result Code value: 0x51

- One-line description: Request message type not implemented

- Where condition detected: ANCP agent

- Further description: This could indicate a mismatch in protocol version or capability state. It is also possible that support of a specific message is optional within some ANCP capability.

- Required additional information in the response message: None, if the response message is of the same type as the request. As specified in Section 4.2, if the response message is a Generic Response message.

- Target: ANCP agent at the peer that sent the original request

- Action RECOMMENDED for the receiving ANCP agent: If the receiver of this Result Code value expects that support of the message type concerned is mandatory according to the capabilities negotiated for the session, it MAY re-send the message in case the message was corrupted in transit the first time. If that fails, and use of the message type cannot be avoided, the ANCP agent MAY reset the adjacency by sending an RSTACK adjacency message as described in Section 3.5.2.6.1, where Sender and Receiver Name, Port, and Instance are taken from recorded adjacency state. If a reset does not eliminate the problem, the receiving ANCP agent SHOULD raise an alarm to management and then cease to operate.

Result Code value: 0x53

- One-line description: Malformed message

- Where condition detected: ANCP agent

- Further description: This could be the result of corruption in transit, or an error in implementation at one end or the other.

- Required additional information in the response message: None, if the response message is of the same type as the request. As specified in Section 4.2, if the response message is a Generic Response message.

- Target: ANCP agent at the peer that sent the original request

- Action RECOMMENDED for the receiving ANCP agent: The request SHOULD be re-sent once to eliminate the possibility of in-transit corruption.

Result Code value: 0x54

- One-line description: Mandatory TLV missing

- Where condition detected: ANCP agent

- Further description: None

- Required additional information in the response message: The response message MUST contain a Status-Info message that encapsulates an instance of each missing mandatory TLV, where the length is set to zero and the value field is empty (i.e., only the 4-byte TLV header is present).

- Target: ANCP agent at the peer that sent the original request

- Action RECOMMENDED for the receiving ANCP agent: Re-send the message with the missing TLV(s), if possible. Otherwise, report the error to the control application with an indication of the missing information required to construct the missing TLV(s).

Result Code value: 0x55

- One-line description: Invalid TLV contents

- Where condition detected: ANCP agent

- Further description: The contents of one or more TLVs in the request do not match the specifications provided for the those TLVs.

- Required additional information in the response message: The response MUST contain a Status-Info TLV encapsulating the erroneous TLVs copied from the original request.

- Target: ANCP agent at the peer that sent the original request

- Action RECOMMENDED for the receiving ANCP agent: Correct the error and re-send the request, if possible. Otherwise, report the error to the control application with an indication of the erroneous information associated with the invalid TLV(s).

Result Code value: 0x500

- One-line description: One or more of the specified ports do not exist

- Where condition detected: Control application

- Further description (if any): This may indicate a configuration mismatch between the AN and the NAS or Authentication, Authorization, and Accounting (AAA).

- Required additional information in the response message: If the request identified multiple access lines or the response is a Generic Response message, then the response MUST contain a Status-Info TLV encapsulating TLV(s) containing the rejected line identifier(s).

- Target: Control application at the peer that sent the original request

- Action RECOMMENDED for the receiving ANCP agent: Indicate the error and forward the line identifiers to the control application.

3.6.1.5. Partition ID (8 bits)

The Partition ID field MUST contain the value that was negotiated for Partition ID during the adjacency procedure as described above.

3.6.1.6. Transaction ID (24 bits)

The Transaction ID is set by the sender of a request message to associate a response message with the original request message. Unless otherwise specified for a given message type, the Transaction ID in request messages MUST be set to a value in the range (1, 2^24 - 1). When used in this manner, the Transaction ID sequencing MUST be maintained independently for each message type within each ANCP adjacency. Furthermore, it SHOULD be incremented by 1 for each new message of the given type, cycling back to 1 after running the full range. For event messages, the Transaction ID SHOULD be set to zero.

Unless otherwise specified, the default behavior for all ANCP responses is that the value of the Transaction ID MUST be copied from the corresponding request message.

3.6.1.7. I Flag and SubMessage Number (1 + 15 bits)

In GSMPv3, these provide a mechanism for message fragmentation. Because ANCP uses TCP transport, this mechanism is unnecessary. An ANCP agent MUST set the I Flag and subMessage Number fields to 1 to signify "no fragmentation".

3.6.1.8. Length (16 bits)

This field MUST be set to the length of the ANCP message in bytes, including its header fields and message body but excluding the 4-byte encapsulating header defined in Section 3.2.

3.6.2. The ANCP Message Body

The detailed contents of the message payload portion of a given ANCP message can vary with the capability in the context of which it is being used. However, the general format consists of zero or more fixed fields, followed by a variable amount of data in the form of Type-Length-Value (TLV) data structures.

The general format of a TLV is shown in Figure 8:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type (IANA registered)    |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                             Value                             ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: General TLV Format

The fields of a TLV are defined as follows:

Type (16 bits): The TLV Type is an unsigned value identifying the TLV type and nature of its contents. An IANA registry has been established for ANCP TLV Type codes.

Length (16 bits): The number of bytes of data in the Value field of the TLV, excluding any padding required to bring this TLV to a 4-byte word boundary (see "Value" below). If a TLV contains other TLVs, any padding in the contained TLVs MUST be included in the value of Length. Depending on the specification of the TLV, the value of Length can be zero, a constant for all instances of the TLV, or a varying quantity.

Value (variable): The actual data carried by the TLV, if any. The Value field in each TLV MUST be padded with zeroes as required to align with a 4-byte word boundary. The Value field of a TLV MAY include fixed fields and/or other TLVs.

Unless otherwise specified, TLVs MAY be added to a message in any order. If the recipient of a message does not understand a particular TLV, it MUST silently ignore it.

A number of TLVs are specified in the remainder of this document.

3.7. General Principles for the Design of ANCP Messages

ANCP allows for two messaging constructs to support request/response interaction:

a. The same message type is used for both the request message and the response message. The Result and Result Code field settings are used to differentiate between request and response messages.

b. The request and response messages use two different message types.

The first approach is illustrated by the protocol specifications in Section 8.4, the second by specifications in Section 6.4. The purpose of this section is to provide more details about the second approach in order to allow the use of this messaging construct for the development of additional ANCP extensions.

As Section 3.6 indicated, all ANCP messages other than adjacency messages share a common header format. When the response message type is different from that of the request, the specification of the request message will typically indicate that the Result field is set to Ignore (0x0) and provide procedures indicating explicitly when the receiver should generate a response and what message type it should use.

The Transaction ID field is used to distinguish between multiple request messages of the same type and to associate a response message to a request. Specifications of ANCP messages for applications not requiring response correlation SHOULD indicate that the Transaction ID MUST be set to zero in requests. Applications that require response correlation SHOULD refer to the Transaction ID behavior described in Section 3.6.1.

The specification for a response message SHOULD indicate in all cases that the value of the Transaction Identifier MUST be set to that of the corresponding request message. This allows the requester to establish whether or not correlation is needed (by setting a non-zero or zero value for the Transaction ID).

4. Generally Useful ANCP Messages and TLVs

This section defines two messages and a number of TLVs that could be useful in multiple capabilities. In some cases, the content is under-specified, with the intention that particular capabilities spell out the remaining details.

4.1. Provisioning Message

The Provisioning message is sent by the NAS to the AN to provision information of global scope (i.e., not associated with specific access lines) on the AN. The Provisioning message has the format shown in Figure 9. Support of the Provisioning message is OPTIONAL unless the ANCP agent claims support for a capability that requires its use.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           TCP/IP Encapsulating Header (Section 3.2)           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                ANCP General Message Header                    |
   +                      (Section 3.6.1)                          +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                             TLVs                              ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Format of the Provisioning Message

The message header field settings given below are REQUIRED in the Provisioning message. The remaining message header fields MUST be set as specified in Section 3.6.1. Which TLVs to carry in the Provisioning message is specified as part of the specification of the capabilities that use that message. The Provisioning message MAY be used to carry data relating to more than one capability at once, assuming that the capabilities concerned can coexist and have all been negotiated during adjacency establishment.

Message Type: MUST be set to 93.

Result: MUST be set to 0x0 (Ignore).

Result Code: MUST be set to zero.

Transaction ID: MUST be populated with a non-zero value chosen in the manner described in Section 3.6.1.6.

If the AN can process the message successfully and accept all the provisioning directives contained in it, the AN MUST NOT send any response.

Unless otherwise specified for a particular capability, if the AN fails to process the message successfully it MUST send a Generic Response message (Section 4.2) indicating failure and providing appropriate diagnostic information.

4.2. Generic Response Message

This section defines the Generic Response message. The Generic Response message MAY be specified as the appropriate response to a message defined in an extension to ANCP, instead of a more specific response message. As a general guideline, specification of the Generic Response message as a response is appropriate where no data needs to be returned to the peer other than a result (success or failure), plus, in the case of a failure, a code indicating the reason for failure and a limited amount of diagnostic data. Depending on the particular use case, the Generic Response message MAY be sent by either the NAS or the AN.

Support of the Generic Response message, both as sender and as receiver, is REQUIRED for all ANCP agents, regardless of what capabilities they support.

The AN or NAS MAY send a Generic Response message indicating a failure condition independently of a specific request before closing the adjacency as a consequence of that failure condition. In this case, the sender MUST set the Transaction ID field in the header and the Message Type field within the Status-Info TLV to zeroes. The receiver MAY record the information contained in the Status-Info TLV for management use.

The format of the Generic Response message is shown in Figure 10.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           TCP/IP Encapsulating Header (Section 3.2)           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                ANCP General Message Header                    |
   +                      (Section 3.6.1)                          +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                Access line identifying TLV(s)                 |
   +                (copied from original request)                 +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Status-Info TLV                            |
   ~                     (Section 4.5)                             ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NOTE: TLVs MAY be in a different order from what is shown in this figure.

Figure 10: Structure of the Generic Response Message

This document specifies the following header fields. The remaining fields in the ANCP general message header MUST be set as specified in Section 3.6.1.

Message Type: MUST be set to 91.

Result: MUST be set to 0x3 (Success) or 0x4 (Failure).

Result Code: MUST be set to zero for success or an appropriate non-zero value for failure.

Transaction ID: MUST be copied from the message to which this message is a response.

If the original request applied to a specific access line or set of lines, the TLVs identifying the line(s) and possibly the user MUST be copied into the Generic Response message at the top level.

The Status-Info TLV MAY be present in a success response, to provide a warning as defined for a specific request message type. It MUST be present in a failure response. See Section 4.5 for a detailed description of the Status-Info TLV. The actual contents will depend on the request message type this message is responding to and the value of the Result Code field.

To prevent an infinite loop of error responses, if the Generic Response message is itself in error, the receiver MUST NOT generate an error response in return.

4.3. Target TLV

Type: 0x1000 to 0x1020 depending on the specific content. Only 0x1000 has been assigned in this specification (see below). Support of any specific variant of the Target TLV is OPTIONAL unless the ANCP agent claims support for a capability that requires its use.

Description: The Target TLV (0x1000 - 0x1020) is intended to be a general means to represent different types of objects.

Length: Variable, depending on the specific object type.

Value: Target information as defined for each object type. The Value field MAY consist of sub-TLVs.

TLV Type 0x1000 is assigned to a variant of the Target TLV representing a single access line and encapsulating one or more sub-TLVs identifying the target. Figure 11 is an example illustrating the TLV format for a single port identified by an Access-Loop-Circuit-ID TLV (0x0001) (Section 5.1.2.1).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    TLV Type = 0x1000          |Length = Circuit-ID Length + 4 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Access-Loop-Circuit-ID=0x0001 |       Circuit-ID Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                    Access Loop Circuit ID                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Example of Target TLV for Single Access Line

4.4. Command TLV

Type: 0x0011

Description: The Command TLV (0x0011) is intended to be a general means of encapsulating one or more command directives in a TLV-oriented message. The semantics of the command can be specified for each message type using it. That is, the specification of each message type that can carry the Command TLV is expected to define the meaning of the content of the payload, although re-use of specifications is, of course, permissible when appropriate. Support of any specific variant of the Command TLV is OPTIONAL unless the ANCP agent claims support for a capability that requires its use.

Length: Variable, depending on the specific contents.

Value: Command information as defined for each message type. The field MAY include sub-TLVs. The contents of this TLV MUST be specified as one "command" or alternatively a sequence of one or more "commands", each beginning with a 1-byte Command Code and possibly including other data following the Command Code. An IANA registry has been established for Command Code values. This document reserves the Command Code value 0 as an initial entry in the registry.

4.5. Status-Info TLV

Name: Status-Info

Type: 0x0106

Description: The Status-Info-TLV is intended to be a general container for warning or error diagnostics relating to commands and/or requests. It is a supplement to the Result Code field in the ANCP general header. The specifications for individual message types MAY indicate the use of this TLV as part of responses, particularly for failures. As mentioned above, the Generic Response message will usually include an instance of the Status-Info TLV. Support of the Status-Info TLV, both as sender and as receiver, is REQUIRED for all ANCP agents, regardless of what capabilities they support.

Length: Variable, depending on the specific contents.

Value: The following fixed fields. In addition, sub-TLVs MAY be appended to provide further diagnostic information.

Reserved (8 bits): See Section 3.4 for handling of reserved fields.

Msg Type (8 bits): Message Type of the request for which this TLV is providing diagnostics.

Error Message Length (16 bits): Number of bytes in the error message, excluding padding, but including the language tag and delimiter. This MAY be zero if no error message is provided.

Error Message: Human-readable string providing information about the warning or error condition. The initial characters of the string MUST be a language tag as described in [RFC5646], terminated by a colon (":"). The actual text string follows the delimiter. The field is padded at the end with zeroes as necessary to extend it to a 4-byte word boundary.

Section 3.6.1.4 provides recommendations for what TLVs to add in the Status-Info TLV for particular values of the message header Result Code field.

Figure 12 illustrates the Status-Info TLV.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    TLV Type = 0x0106          |              Length           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Reserved   |  Msg Type     |      Error Message Length     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        Error Message (padded to 4-byte boundary)              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           optional sub-TLVs...                                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: The Status-Info TLV

5. Introduction to ANCP Capabilities for Digital Subscriber Lines (DSLs)

DSL is a widely deployed access technology for Broadband Access for Next Generation Networks. Specifications such as [TR-059], [TR-058], and [TR-092] describe possible architectures for these access networks. The scope of these specifications includes the delivery of voice, video, and data services.

The next three sections of this document specify basic ANCP capabilities for use specifically in controlling Access Nodes serving DSL access (Tech Type = 0x05). The same ANs could be serving other access technologies (e.g., Metro-Ethernet, Passive Optical Networking, WiMax), in which case the AN will also have to support the corresponding other-technology-specific capabilities. Those additional capabilities are outside the scope of the present document.

5.1. DSL Access Line Identification

Most ANCP messages involve actions relating to a specific access line. Thus, it is necessary to describe how access lines are identified within those messages. This section defines four TLVs for that purpose and provides an informative description of how they are used.

5.1.1. Control Context (Informative)

Three types of identification are described in [TR-101] and provided for in the TLVs defined in this section:

- identification of an access line by its logical appearance on the user side of the Access Node;

- identification of an access line by its logical appearance on the NAS side of the Access Node; and

- identification down to the user or host level as a supplement to access line identification in one of the other two forms.

All of these identifiers originate with the AN control application, during the process of DSL topology discovery. The control application chooses which identifiers to use and the values to place into them on a line-by-line basis, based on AN configuration and deployment considerations.

Aside from its use in ANCP signalling, access line identification is also used in DHCP ([RFC2131], [RFC3315]) transactions involving hosts served by DSL. Either the AN or the NAS can serve as a DHCP relay node. [TR-101] requires the AN or NAS in this role to add access line identification in Option 82 (Information) ([RFC3046], with its IPv6 equivalent in [RFC4649]) to each DHCP request it forwards to the DHCP server. It is desirable for efficiency that the identification used in this signalling should be the same as the identification used in ANCP messages.

From the point of view of ANCP itself, the identifiers are opaque. From the point of view of the AN control application, the syntax for the user-side access line identifier is the same as specified in Section 3.9.3 of [TR-101] for DHCP Option 82. The syntax for the ASCII form of the NAS-side access line identifier will be similar.

Access line identification by logical appearance on the user side of the Access Node will always identify a DSL access line uniquely. Identification by the logical appearance on the NAS side of the Access Node is unique only if there is a one-to-one mapping between the appearances on the two sides and no identity-modifying aggregation between the AN and the NAS. In other cases, and in particular in the case of Ethernet aggregation using the N:1 VLAN model, the user-side access line identification is necessary, but the NAS-side identification is potentially useful information allowing the NAS to build up a picture of the aggregation network topology.

Additional identification down to the user or host level is intended to supplement rather than replace either of the other two forms of identification.

> Sections 3.8 and 3.9 of [TR-101] are contradictory on this point. It is assumed here that Section 3.9 is meant to be authoritative.

The user-level identification takes the form of an administered string that again is opaque at the ANCP level.

The NAS control application will use the identifying information it receives from the AN directly for some purposes. For examples, see the introductory part of Section 3.9 of [TR-101]. For other purposes, the NAS will build a mapping between the unique access line identification provided by the AN, the additional identification of the user or host (where provided), and the IP interface on a particular host. For access lines with static IP address assignment, that mapping could be configured instead.

5.1.2. TLVs for DSL Access Line Identification

This section provides a normative specification of the TLVs that ANCP provides to carry the types of identification just described. The Access-Loop-Circuit-ID TLV identifies an access line by its logical appearance on the user side of the Access Node. Two alternatives, the Access-Aggregation-Circuit-ID-ASCII TLV and the Access-Aggregation-Circuit-ID-Binary TLV, identify an access line by its logical appearance on the NAS side of the Access Node. It is unlikely that a given AN uses both of these TLVs, either for the same line or for different lines, since they carry equivalent information. Finally, the Access-Loop-Remote-ID TLV contains an operator-configured string that uniquely identifies the user on the associated access line, as described in Sections 3.9.1 and 3.9.2 of [TR-101].

ANCP agents conforming to this section MUST satisfy the following requirements:

- ANCP agents MUST be able to build and send the Access-Loop-Circuit-ID TLV, the Access-Loop-Remote-ID TLV, and either the Access-Aggregation-Circuit-ID-ASCII TLV or the Access-Aggregation-Circuit-ID-Binary TLV (implementation choice), when passed the associated information from the AN control application.

- ANCP agents MUST be able to receive all four TLV types, extract the relevant information, and pass it to the control application.

- If the Access-Loop-Remote-ID TLV is present in a message, it MUST be accompanied by an Access-Loop-Circuit-ID TLV and/or an Access-Aggregation-Circuit-ID-ASCII TLV or Access-Aggregation-Circuit-ID-Binary TLV with two VLAN identifiers.

  The Access-Loop-Remote-ID TLV is not enough to identify an access line uniquely on its own. As indicated above, an Access-Aggregation-Circuit-ID-ASCII TLV or Access-Aggregation-Circuit-ID-Binary TLV with two VLAN identifiers may or may not identify an access line uniquely, but this is up to the control application to decide.

- If the Access-Aggregation-Circuit-ID-ASCII TLV or Access-Aggregation-Circuit-ID-Binary TLV is present in a message with just one VLAN identifier, it MUST be accompanied by an Access-Loop-Circuit-ID TLV.

5.1.2.1. Access-Loop-Circuit-ID TLV

Type: 0x0001

Description: A locally administered human-readable string generated by or configured on the Access Node, identifying the corresponding access loop logical port on the user side of the Access Node.

Length: Up to 63 bytes

Value: ASCII string

5.1.2.2. Access-Loop-Remote-ID TLV

Type: 0x0002

Description: An operator-configured string that uniquely identifies the user on the associated access line, as described in Sections 3.9.1 and 3.9.2 of [TR-101].

Length: Up to 63 bytes

Value: ASCII string

5.1.2.3. Access-Aggregation-Circuit-ID-Binary TLV

Type: 0x0006

Description: This TLV identifies or partially identifies a specific access line by means of its logical circuit identifier on the NAS side of the Access Node.

For Ethernet access aggregation, where a per-subscriber (stacked) VLAN can be applied (1:1 model as defined in [TR-101]), the TLV contains two value fields. Each field carries a 12-bit VLAN identifier (which is part of the VLAN tag defined by [IEEE802.1Q]). The first field MUST carry the inner VLAN identifier, while the second field MUST carry the outer VLAN identifier.

When the N:1 VLAN model is used, only one VLAN tag is available. For the N:1 model, the Access-Aggregation-Circuit-ID-Binary TLV contains a single value field, which MUST carry the 12-bit VLAN identifier derived from the single available VLAN tag.

In the case of an ATM aggregation network, where the DSLAM is directly connected to the NAS (without an intermediate ATM switch), the Virtual Path Identifier (VPI) and Virtual Circuit Identifier (VCI) on the DSLAM uplink correspond uniquely to the DSL access line on the DSLAM. The Access-Aggregation-Circuit-ID-Binary TLV MAY be used to carry the VPI and VCI. The first value field of the TLV MUST carry the VCI, while the second value field MUST carry the VPI.

Each identifier MUST be placed in the low-order bits of its respective 32-bit field, with the higher-order bits set to zero. The ordering of the bits of the identifier MUST be the same as when the identifier is transmitted on the wire to identify an Ethernet frame or ATM cell.

The Access-Aggregation-Circuit-ID-Binary is illustrated in Figure 13.

Length: 4 or 8 bytes

Value: One or two 32-bit binary fields.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    TLV Type = 0x0006          |        Length = 4 or 8        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Single VLAN Identifier, inner VLAN identifier, or VCI        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                Outer VLAN identifier or VPI                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: The Access-Aggregation-Circuit-ID-Binary TLV

5.1.2.4. Access-Aggregation-Circuit-ID-ASCII TLV

Type: 0x0003

Description: This TLV transmits the ASCII equivalent of the Access-Aggregation-Circuit-ID-Binary TLV. As mentioned in the previous section, the AN control application will use a format similar to that specified in Section 3.9.3 of [TR-101] for the format of the "circuit-id".

As an extension to the present document, the Access Node could convey to the NAS the characteristics (e.g., bandwidth) of the uplink on the Access Node. This TLV or the binary equivalent defined above then serves the purpose of uniquely identifying the uplink whose characteristics are being defined. The present document does not specify the TLVs needed to convey the uplink characteristics.

Length: Up to 63 bytes

Value: ASCII string

6. ANCP-Based DSL Topology Discovery

Section 3.1 of [RFC5851] describes the requirements for the DSL Topology Discovery capability.

6.1. Control Context (Informative)

The AN control application in the DSLAM requests ANCP to send a DSL-specific Port Up message to the NAS under the following circumstances:

- o when a new adjacency with the NAS is established, for each DSL loop that is synchronized at that time;

- subsequent to that, whenever a DSL access line resynchronizes; and
- whenever the AN control application wishes to signal that a line attribute has changed.

The AN control application in the DSLAM requests ANCP to send a DSL-specific Port Down message to the NAS under the following circumstances:

- when a new adjacency with the NAS is established, for each DSL loop that is provisioned but not synchronized at that time;
- whenever a DSL access line that is equipped in an AN but administratively disabled is signaled as "IDLE"; and
- subsequent to that, whenever a DSL access line loses synchronization.

The AN control application passes information to identify the DSL loop to ANCP to include in the Port Up or Port Down message, along with information relating to DSL access line attributes.

In the case of bonded copper loops to the customer premise (as per DSL multi-pair bonding described by [G.998.1] and [G.998.2]), the AN control application requests that ANCP send DSL-specific Port Up and Port Down messages for the aggregate "DSL bonded circuit" (represented as a single logical port) as well as the individual DSL access lines of which it is comprised. The information relating to DSL access line attributes that is passed by the AN control application is aggregate information.

ANCP generates the DSL-specific Port Up or Port Down message and transfers it to the NAS. ANCP on the NAS side passes an indication to the NAS control application that a DSL Port Up or Port Down message has been received along with the information contained in the message.

The NAS control application updates its view of the DSL access line state, performs any required accounting operations, and uses any included line attributes to adjust the operation of its queuing/scheduling mechanisms as they apply to data passing to and from that DSL access line.

Figure 14 summarizes the interaction.

```
   1.   Home            Access                          NAS
       Gateway           Node

             ----------->     -------------------------->
                  DSL          Port Up (Event message)
                 Signal       (default line parameters)

   2.   Home            Access                          NAS
       Gateway           Node

             ----------->     -------------------------->
                  DSL           Port Up (Event message)
                Resynch       (updated line parameters)

   3.   Home            Access                          NAS
       Gateway           Node

             ----------->     -------------------------->
             Loss of          Port Down (Event message)
             DSL Signal       (selected line parameters)
```

Figure 14: ANCP Message Flow for DSL Topology Discovery

6.2. Protocol Requirements

The DSL topology discovery capability is assigned capability type 0x0001. No capability data is associated with this capability.

6.2.1. Protocol Requirements on the AN Side

The AN-side ANCP agent MUST be able to create DSL-specific Port Up and Port Down messages according to the format specified in Section 6.3.

The AN-side ANCP agent MUST conform to the normative requirements of Section 5.1.2.

The AN-side ANCP agent MUST follow the AN-side procedures associated with DSL-specific Port Up and Port Down messages as they are specified in Section 6.4.

6.2.2. Protocol Requirements on the NAS Side

The NAS-side ANCP agent MUST be able to receive and validate DSL-specific Port Up and Port Down messages according to the format specified in Section 6.3.

The NAS-side ANCP agent MUST conform to the normative requirements of Section 5.1.2.

The NAS-side ANCP agent MUST follow the NAS-side procedures associated with DSL-specific Port Up and Port Down messages as they are specified in Section 6.4.

6.3. ANCP Port Up and Port Down Event Message Descriptions

The format of the ANCP Port Up and Port Down Event messages is shown in Figure 15.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           TCP/IP Encapsulating Header (Section 3.2)           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                ANCP General Message Header                    |
   +                      (Section 3.6.1)                          +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                    Unused (20 bytes)                          ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |x|x|x|x|x|x|x|x| Message Type  |   Tech Type   |  Reserved     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     # of TLVs                 | Extension Block length (bytes)|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                 Access line identifying TLV(s)                ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                DSL-Line-Attributes TLV                        |
   ~        (MANDATORY in Port Up, OPTIONAL in Port Down)          ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NOTE: TLVs MAY be in a different order from what is shown in this figure.

Figure 15: Format of the ANCP Port Up and Port Down Event Messages for DSL Topology Discovery

See Section 3.6.1 for a description of the ANCP general message header. The Message Type field MUST be set to 80 for Port Up, 81 for Port Down. The 4-bit Result field MUST be set to zero (signifying Ignore). The 12-bit Result Code field and the 24-bit Transaction

Identifier field MUST also be set to zeroes. Other fields in the general header MUST be set a as described in Section 3.6.

The five-word Unused field is a historical leftover. The handling of unused/reserved fields is described in Section 3.4.

The remaining message fields belong to the "extension block", and are described as follows:

Extension Flags (8 bits): The flag bits denoted by 'x' are currently unspecified and reserved.

Message Type (8 bits): Message Type has the same value as in the general header (i.e., 80 or 81).

Tech Type (8 bits): MUST be set to 0x05 (DSL).

Reserved (8 bits): set as described in Section 3.4.

# of TLVs (16 bits): The number of TLVs that follow, not counting TLVs encapsulated within other TLVs.

Extension Block Length (16 bits): The total length of the TLVs carried in the extension block in bytes, including any padding within individual TLVs.

TLVs: One or more TLVs to identify a DSL access line and zero or more TLVs to define its characteristics.

6.4. Procedures

6.4.1. Procedures on the AN Side

The AN-side ANCP agent creates and transmits a DSL-specific Port Up or Port Down message when requested by the AN control application and presented with the information needed to build a valid message. It is RECOMMENDED that the Access Node use a dampening mechanism per DSL access line to control the rate at which state changes are communicated to the NAS.

At the top level, the extension block within a DSL-specific Port Up or Port Down message MUST include TLVs from Section 5.1.2 to identify the DSL access line.

TLVs presenting DSL access line attributes (i.e., the TLVs specified in Section 6.5) MUST be encapsulated within the DSL-Line-Attributes TLV. When the DSL-Line-Attributes TLV is present in a message, it MUST contain at least one such TLV and will generally contain more than one. In the Port Up message, the DSL-Line-Attributes TLV MUST be present. In the Port Down message, the DSL-Line-Attributes TLV MAY be present.

6.4.2. Procedures on the NAS Side

The NAS-side ANCP agent MUST be prepared to receive Port Up and Port Down messages for a given DSL access line or logical port at any time after negotiation of an adjacency has been completed. It is possible for two Port Up messages in succession to be received for the same DSL access line without an intervening Port Down message, and vice versa.

The NAS-side ANCP agent SHOULD validate each message against the specifications given in Section 6.3 and the TLV specifications given in Sections 5.1.2 and 6.5. If it finds an error, it MAY generate a Generic Response message containing an appropriate Result Code value. If it does so, the message MUST contain copies of all of the identifier TLVs from Section 5.1.2 that were present in the Port Up or Port Down message. The message MUST also contain a Status-Info TLV that in turn contains other information appropriate to the message header Result Code value as described in Section 3.6.1.4.

6.5. TLVs for DSL Line Attributes

As specified above, the DSL-Line-Attributes TLV is inserted into the Port Up or Port Down message at the top level. The remaining TLVs defined below are encapsulated within the DSL-Line-Attributes TLV.

6.5.1. DSL-Line-Attributes TLV

Type: 0x0004

Description: This TLV encapsulates attribute values for a DSL access line serving a subscriber.

Length: Variable (up to 1023 bytes)

Value: One or more encapsulated TLVs corresponding to DSL access line attributes. The DSL-Line-Attributes TLV MUST contain at least one TLV when it is present in a Port Up or Port Down message. The actual contents are determined by the AN control application.

6.5.2. DSL-Type TLV

Type: 0x0091

Description: Indicates the type of transmission system in use.

Length: 4 bytes

Value: 32-bit unsigned integer

ADSL1 = 1

ADSL2 = 2

ADSL2+ = 3

VDSL1 = 4

VDSL2 = 5

SDSL = 6

OTHER = 0

6.5.3. Actual-Net-Data-Rate-Upstream TLV

Type: 0x0081

Description: Actual upstream net data rate on a DSL access line.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.4. Actual-Net-Data-Rate-Downstream TLV

Type: 0x0082

Description: Actual downstream net data rate on a DSL access line.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.5. Minimum-Net-Data-Rate-Upstream TLV

Type: 0x0083

Description: Minimum upstream net data rate desired by the operator.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.6. Minimum-Net-Data-Rate-Downstream TLV

Type: 0x0084

Description: Minimum downstream net data rate desired by the operator.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.7. Attainable-Net-Data-Rate-Upstream TLV

Type: 0x0085

Description: Maximum net upstream rate that can be attained on the DSL access line.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.8. Attainable-Net-Data-Rate-Downstream TLV

Type: 0x0086

Description: Maximum net downstream rate that can be attained on the DSL access line.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.9. Maximum-Net-Data-Rate-Upstream TLV

Type: 0x0087

Description: Maximum net upstream data rate desired by the operator.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.10. Maximum-Net-Data-Rate-Downstream TLV

Type: 0x0088

Description: Maximum net downstream data rate desired by the operator.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.11. Minimum-Net-Low-Power-Data-Rate-Upstream TLV

Type: 0x0089

Description: Minimum net upstream data rate desired by the operator in low power state.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.12. Minimum-Net-Low-Power-Data-Rate-Downstream TLV

Type: 0x008A

Description: Minimum net downstream data rate desired by the operator in low power state.

Length: 4 bytes

Value: Rate in kbits/s as a 32-bit unsigned integer

6.5.13. Maximum-Interleaving-Delay-Upstream TLV

Type: 0x008B

Description: Maximum one-way interleaving delay.

Length: 4 bytes

Value: Time in ms as a 32-bit unsigned integer

6.5.14. Actual-Interleaving-Delay-Upstream TLV

Type: 0x008C

Description: Value corresponding to the interleaver setting.

Length: 4 bytes

Value: Time in ms as a 32-bit unsigned integer

6.5.15. Maximum-Interleaving-Delay-Downstream TLV

Type: 0x008D

Description: Maximum one-way interleaving delay.

Length: 4 bytes

Value: Time in ms as a 32-bit unsigned integer

6.5.16. Actual-Interleaving-Delay-Downstream

Type: 0x008E

Description: Value corresponding to the interleaver setting.

Length: 4 bytes

Value: Time in ms as a 32-bit unsigned integer

6.5.17. DSL-Line-State TLV

Type: 0x008F

Description: The state of the DSL access line.

Length: 4 bytes

Value: 32-bit unsigned integer

SHOWTIME = 1

IDLE = 2

SILENT = 3

6.5.18. Access-Loop-Encapsulation TLV

Type: 0x0090

Description: The data link protocol and, optionally, the encapsulation overhead on the access loop. When this TLV is present, at least the data link protocol MUST be indicated. The encapsulation overhead MAY be indicated. The Access Node MAY choose to not convey the encapsulation on the access loop by specifying values of 0 (NA) for the two encapsulation fields.

Length: 3 bytes

Value: The 3 bytes (most to least significant) and valid set of values for each byte are defined as follows:

Byte 1: Data Link

ATM AAL5 = 0

ETHERNET = 1

Byte 2: Encapsulation 1

NA = 0

Untagged Ethernet = 1

Single-tagged Ethernet = 2

Double-tagged Ethernet = 3

Byte 3: Encapsulation 2

NA = 0

PPPoA LLC = 1

PPPoA Null = 2

IPoA LLC = 3

IPoA Null = 4

Ethernet over AAL5 LLC with FCS = 5

Ethernet over AAL5 LLC without FCS = 6

Ethernet over AAL5 NULL with FCS = 7

Ethernet over AAL5 NULL without FCS = 8

The Access-Loop-Encapsulation TLV is illustrated in Figure 16.

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |    TLV Type = 0x0090          |        Length = 3             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Data link     |    Encaps 1   |    Encaps 2   | Padding (=0)  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: The Access-Loop-Encapsulation TLV

7. ANCP-Based DSL Line Configuration

The use case for ANCP-based DSL Line Configuration is described in Section 3.2 of [RFC5851].

7.1. Control Context (Informative)

Triggered by topology information reporting a new DSL access line or triggered by a subsequent user session establishment (via PPP or DHCP), RADIUS/AAA sends service parameters to the NAS control application for configuration on the access line. The NAS control application passes the request on to the NAS-side agent, which sends the information to the AN by means of a Port Management (line configuration) message. The AN-side agent passes this information up to the AN control application, which applies it to the line. Figure 17 summarizes the interaction.

```
     Home            Access               NAS             RADIUS/AAA
    Gateway           Node                              Policy Server

          ----------->     --------------->
             DSL          Port Up message)
            Signal       (line parameters)

          -------------------------------->   -------------->
                  PPP/DHCP Session            Authentication &
                                              authorization

                          <----------------
                            Port Management message
                            (line configuration)
```

Figure 17: Message Flow - ANCP Mapping for Initial Line Configuration

The NAS could update the line configuration as a result of a subscriber service change (e.g., triggered by the policy server). Figure 18 summarizes the interaction.

```
   User     Home            Access         NAS
           Gateway           Node

                --------------------------->
                  PPP/DHCP Session

      -----------------------------------------------------> Web portal,
                  Service on demand                            OSS, etc.
                                                                  |
                                             <-----------  RADIUS/AAA
                                             Change of     Policy Server
                                           authorization

                             <------------
                              Port Management
                                  message
                              (new profile)
```

OSS: Operations Support System

Figure 18: Message Flow - ANCP Mapping for Updated Line Configuration

7.2. Protocol Requirements

The DSL access line configuration capability is assigned capability type 0x0002. No capability data is associated with this capability.

7.2.1. Protocol Requirements on the NAS Side

The NAS-side ANCP agent MUST be able to create DSL-specific Port Management (line configuration) messages according to the format specified in Section 7.3.

The NAS-side ANCP agent MUST conform to the normative requirements of Section 5.1.2.

The NAS-side ANCP agent MUST follow the NAS-side procedures associated with DSL-specific Port Management (line configuration) messages as they are specified in Section 7.4.

7.2.2. Protocol Requirements on the AN Side

The AN-side ANCP agent MUST conform to the normative requirements of Section 5.1.2.

The AN-side ANCP agent MUST be able to receive and validate DSL-specific Port Management (line configuration) messages according to the format specified in Section 7.3.

The AN-side ANCP agent MUST follow the AN-side procedures associated with DSL-specific Port Management (line configuration) messages as specified in Section 7.4.

7.3. ANCP Port Management (Line Configuration) Message Format

The ANCP Port Management message for DSL access line configuration has the format shown in Figure 19.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           TCP/IP Encapsulating Header (Section 3.2)           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                ANCP General Message Header                    |
   +                      (Section 3.6.1)                          +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                       Unused (12 bytes)                       ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        Unused (2 bytes)       |  Function=8   | X-Function=0  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Unused (4 bytes)                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |x|x|x|x|x|x|x|x| Message Type  |            Reserved           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     # of TLVs                 | Extension Block length (bytes)|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                 Access line identifying TLV(s)                ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                Line configuration TLV(s)                      ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NOTE: TLVs MAY be in a different order from what is shown in this figure.

Figure 19: Port Management Message for DSL Line Configuration

See Section 3.6 for a description of the ANCP general message header. The Message Type field MUST be set to 32. The 12-bit Result Code field MUST be set to 0x0. The 4-bit Result field MUST be set to either 0x1 (Nack) or 0x2 (AckAll), as determined by policy on the NAS. The 24-bit Transaction Identifier field MUST be set to a positive value. Other fields in the general header MUST be set as described in Section 3.6.

The handling of the various unused/reserved fields is described in Section 3.4.

The remaining message fields are described as follows:

Function (8 bits): Action to be performed. For line configuration, Function MUST be set to 8 (Configure Connection Service Data). This action type requests the Access Node (i.e., DSLAM) to apply service configuration data contained in the line configuration TLVs to the DSL access line designated by the access line identifying TLVs.

X-Function (8 bits): Qualifies the action set by Function. For DSL access line configuration, this field MUST be set to 0.

Extension Flags (8 bits): The flag bits denoted by ’x’ before the Message Type field are reserved for future use.

Message Type (8 bits): Message Type has the same value as in the general header (i.e., 32).

Reserved (16 bits): Reserved for future use.

# of TLVs (16 bits): The number of TLVs that follow, not counting TLVs encapsulated within other TLVs.

Extension Block Length (16 bits): The total length of the TLVs carried in the extension block in bytes, including any padding within individual TLVs.

TLVs: Two or more TLVs to identify a DSL access line and configure its service data.

Other ANCP capabilities, either specific to DSL or technology-independent, MAY reuse the Port Management message for service configuration. If the settings of the fixed fields are compatible with the settings just described, the same Port Management message that is used for DSL access line configuration MAY be used to carry TLVs relating to the other capabilities that apply to the same DSL access line.

Use of the Port Management message for configuration MAY also be generalized to other access technologies, if the respective capabilities specify use of access line identifiers appropriate to those technologies in place of the identifiers defined in Section 5.1.2.

7.4. Procedures

Service configuration MAY be performed on an access line regardless of its current state.

7.4.1. Procedures on the NAS Side

When requested by the NAS control application and presented with the necessary information to do so, the NAS-side agent MUST create and send a Port Management message with the fixed fields set as described in the previous section. The message MUST contain one or more TLVs to identify an access line according the requirements of Section 5.1.2. The NAS MUST include one or more TLVs to configure line service parameters for that line. Section 7.5 currently identifies only one such TLV, Service-Profile-Name, but other TLVs MAY be added by extensions to ANCP.

7.4.2. Procedures on the AN Side

The AN-side ANCP agent MUST be prepared to receive Port Management (line configuration) messages for a given DSL access line or logical port at any time after negotiation of an adjacency has been completed.

The AN-side ANCP agent SHOULD validate each message against the specifications given in Section 7.3 and the TLV specifications given in Sections 5.1.2 and 7.5. If it finds an error it MUST return a Port Management response message that copies the Port Management request as it was received, but has the Result header field set to 0x04 (Failure) and the Result Code field set to the appropriate value. The AN-side agent MAY add a Status-Info TLV (Section 4.5) to provide further information on the error, particularly if this is recommended in Section 3.6.1.4 for the given Result Code value. If it does so, the various length fields and the # of TLVs field within the message MUST be adjusted accordingly.

7.5. TLVs for DSL Line Configuration

Currently, only the following TLV is specified for DSL access line configuration. More TLVs may be defined in a future version of this specification or in ANCP extensions for individual service attributes of a DSL access line (e.g., rates, interleaving delay, multicast channel entitlement access-list).

7.5.1. Service-Profile-Name TLV

Type: 0x0005

Description: Reference to a pre-configured profile on the DSLAM that contains service-specific data for the subscriber.

Length: Up to 64 bytes

Value: ASCII string containing the profile name (which the NAS learns from a policy server after a subscriber is authorized).

8. ANCP-Based DSL Remote Line Connectivity Testing

The use case and requirements for ANCP-Based DSL remote line connectivity testing are specified in Section 3.3 of [RFC5851].

8.1. Control Context (Informative)

The NAS control application initiates a request for remote connectivity testing for a given access line. The NAS control application can provide loop count and timeout test parameters and opaque data for its own use with the request. The loop count parameter indicates the number of test messages or cells to be used. The timeout parameter indicates the longest that the NAS control application will wait for a result.

The request is passed in a Port Management (Operations, Administration, and Maintenance, OAM) message. If the NAS control application has supplied test parameters, they are used; otherwise, the AN control application uses default test parameters. If a loop count parameter provided by the NAS is outside the valid range, the AN does not execute the test, but returns a result indicating that the test has failed due to an invalid parameter. If the test takes longer than the timeout value (default or provided by the NAS), the AN control application can return a failure result indicating timeout or else can send no response. The AN control application can provide a human-readable string describing the test results, for both failures and successes. If provided, this string is included in the response. Responses always include the opaque data, if any, provided by the NAS control application.

Figure 20 summarizes the interaction.

```
   +-------------+    +-----+      +-------+          +----------------+
   |Radius/AAA   |----|NAS  |------| DSLAM |----------|    CPE         |
   |Policy Server|    +-----+      +-------+          | (DSL Modem +   |
   +-------------+                                    |Routing Gateway)|
                                                      +----------------+
                    Port Management Message
                    (Remote Loopback          ATM loopback
                     Trigger Request)         or EFM Loopback
                  1.  ---------------->    2. -------->
                                               <-------+
                         3. <---------------
                         Port Management Message
                    (Remote Loopback Test Response)
```

CPE: Customer Premises Equipment
EFM: Ethernet First Mile

Figure 20: Message Flow for ANCP-Based OAM

8.2. Protocol Requirements

The DSL remote line connectivity testing capability is assigned capability type 0x0004. No capability data is associated with this capability.

8.2.1. Protocol Requirements on the NAS Side

The NAS-side ANCP agent MUST be able to create DSL-specific Port Management (OAM) messages according to the format specified in Section 8.3.

The NAS-side ANCP agent MUST conform to the normative requirements of Section 5.1.2.

The NAS-side ANCP agent MUST follow the NAS-side procedures associated with DSL-specific Port Management (OAM) messages as they are specified in Section 8.4.

8.2.2. Protocol Requirements on the AN Side

The AN-side ANCP agent MUST conform to the normative requirements of Section 5.1.2.

The AN-side ANCP agent MUST be able to receive and validate DSL-specific Port Management (OAM) messages according to the format specified in Section 8.3.

The AN-side ANCP agent MUST follow the AN-side procedures associated with DSL-specific Port Management (OAM) messages as specified in Section 8.4.

8.3. Port Management (OAM) Message Format

The Port Management message for DSL access line testing has the same format as for DSL access line configuration (see Section 7.3), with the following differences:

- The Result field in the request SHOULD be set to AckAll (0x2), to allow the NAS to receive the information contained in a successful test response.

- The Function field MUST be set to 9 (Remote Loopback). (The X-Function field continues to be 0.)

- The appended TLVs in the extension value field include testing-related TLVs rather than subscriber service information.

The Port Management (OAM) message is illustrated in Figure 21.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           TCP/IP Encapsulating Header (Section 3.2)           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                ANCP General Message Header                    |
   +                      (Section 3.6.1)                          +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                       Unused (12 bytes)                       ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        Unused (2 bytes)       |  Function=9   | X-Function=0  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Unused (4 bytes)                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |x|x|x|x|x|x|x|x| Message Type  |           Reserved            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     # of TLVs                 | Extension Block length (bytes)|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                 Access line identifying TLV(s)                ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                   Testing-related TLVs                        ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NOTE: TLVs MAY be in a different order from what is shown in this figure.

Figure 21: Port Management Message for
DSL Line Remote Connectivity Testing

8.4. Procedures

From the point of view of ANCP, it is permissible to attempt line connectivity testing regardless of the state of the line. However, testing could fail in some states due to technology limitations.

8.4.1. NAS-Side Procedures

When requested by the NAS control application and presented with the necessary information to do so, the NAS-side agent creates and sends a Port Management (OAM) request with the fixed fields set as described in the previous section. The message MUST contain one or more TLVs to identify an access line according the requirements of Section 5.1.2. The NAS MAY include the Opaque-Data TLV and/or the OAM-Loopback-Test-Parameters TLV (defined in Section 8.5) to configure the loopback test for that line.

8.4.2. AN-Side Procedures

The AN-side ANCP agent SHOULD validate each message against the specifications given in Section 8.3 and the TLV specifications given in Sections 5.1.2 and 8.5. If it finds an error it MUST return a Port Management response message that copies the Port Management request as it was received, but has the Result header field set to 0x04 (Failure) and the Result Code field set to the appropriate value. Result Code value 0x509 as described below MAY apply, as well as the other Result Code values documented in Section 3.6.1.4. Result Code value 0x509 SHOULD be used if the OAM-Loopback-Test-Parameters TLV is present with an invalid value of the Count field. The AN-side agent MAY add a Status-Info TLV (Section 4.5) to provide further information on the error, particularly if this is recommended in Section 3.6.1.4 for the given Result Code value. If it does so, the various length fields and the # of TLVs field within the message MUST be adjusted accordingly.

If the received message passes validation, the AN-side ANCP agent extracts the information from the TLVs contained in the message and presents that information to the AN control application. It MUST NOT generate an immediate response to the request, but it MUST instead wait for the AN control application to indicate that the response should be sent.

When requested by the AN control application and presented with the necessary information to do so, the AN-side agent creates and sends a Port Management (OAM) response to the original request. The Result field MUST be set to Success (0x3) or Failure (0x4), and the Result Code field SHOULD be set to one of the following values, as indicated by the AN control application.

0x500: Specified access line does not exist. See the documentation of Result Code 0x500 in Section 3.6.1.4 for more information. The Result header field MUST be set to Failure (0x4).

0x501: Loopback test timed out. The Result header field MUST be set to Failure (0x4).

0x503: DSL access line status showtime

0x504: DSL access line status idle

0x505: DSL access line status silent

0x506: DSL access line status training

0x507: DSL access line integrity error

0x508: DSLAM resource not available. The Result header field MUST be set to Failure (0x04).

0x509: Invalid test parameter. The Result header field MUST be set to Failure (0x4).

All other fields of the request including the TLVs MUST be copied into the response unchanged, except that in a successful response the OAM-Loopback-Test-Parameters TLV MUST NOT appear. If the AN control application has provided the necessary information, the AN-side agent MUST also include an instance of the OAM-Loopback-Test-Response-String TLV in the response.

8.5. TLVs for the DSL Line Remote Connectivity Testing Capability

The following TLVs have been defined for use with the DSL access line testing capability.

8.5.1. OAM-Loopback-Test-Parameters TLV

Type: 0x0007

Description: Parameters intended to override the default values for this loopback test.

Length: 2 bytes

Value: Two unsigned 1-byte fields described below (listed in order of most to least significant).

Byte 1: Count. Number of loopback cells/messages that should be generated on the local loop as part of the loopback test. The Count value SHOULD be greater than 0 and less than or equal to 32.

Byte 2: Timeout. Upper bound on the time in seconds that the NAS will wait for a response from the DSLAM. The value 0 MAY be used to indicate that the DSLAM MUST use a locally determined value for the timeout.

The OAM-Loopback-Test-Parameters TLV is illustrated in Figure 22.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    TLV Type = 0x0007          |        Length = 2             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Count       |  Timeout      |         Padding (=0)          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 22: The OAM-Loopback-Test-Parameters TLV

8.5.2. Opaque-Data TLV

Type: 0x0008

Description: An 8-byte opaque field used by the NAS control application for its own purposes (e.g., response correlation). The procedures in Section 8.4.2 ensure that if it is present in the request it is copied unchanged to the response.

Length: 8 bytes

Value: Two 32-bit unsigned integers.

8.5.3. OAM-Loopback-Test-Response-String TLV

Type: 0x0009

Description: Suitably formatted string containing useful details about the test that the NAS will display for the operator, exactly as received from the DSLAM (no manipulation or interpretation by the NAS).

Length: Up to 128 bytes

Value: UTF-8 encoded string of text.

9. IANA Considerations

This section documents the following IANA actions:

- establishment of the following new ANCP registries:

  ANCP Message Types;

  ANCP Result Codes;

ANCP Port Management Functions;

ANCP Technology Types;

ANCP Command Codes;

ANCP TLV Types;

ANCP Capabilities.

- o establishment of a new joint GSMP/ANCP version registry;
- o addition of ANCP as another user of TCP port 6068 in the port number registry available from http://www.iana.org. The current user is GSMP.

All of these actions are described in detail below except for the port registration, for which the final point above provides sufficient information.

10. IANA Actions

10.1. ANCP Message Type Registry

IANA has created a new registry, ANCP Message Types. Additions to that registry are permitted by Standards Action, as defined by [RFC5226]. The values for Message Type MAY range from 0 to 255, but new Message Types SHOULD be assigned values sequentially from 90 onwards (noting that 91 and 93 are already assigned). The initial contents of the ANCP Message Types registry are as follows:

Message Type	Message Name	Reference
10	Adjacency Protocol	RFC 6320
32	Port Management	RFC 6320
80	Port Up	RFC 6320
81	Port Down	RFC 6320
85	Adjacency Update	RFC 6320
91	Generic Response	RFC 6320
93	Provisioning	RFC 6320

10.2. ANCP Result Code Registry

IANA has created a new registry, ANCP Result Codes. The documentation of new Result Codes MUST include the following information:

- Result Code value (as assigned by IANA);
- One-line description;
- Where condition detected (control application or ANCP agent);
- Further description (if any);
- Required additional information in the response message;
- Target (control application or ANCP agent at the peer that sent the original request);
- Action RECOMMENDED for the receiving ANCP agent.

The values for Result Code are expressed in hexadecimal and MAY range from 0x0 to 0xFFFFFF. The range 0x0 to 0xFFF is allocated by the criterion of IETF Review, as defined by [RFC5226]. IANA SHOULD allocate new Result Code values from this range sequentially beginning at 0x100. The range 0x1000 onwards is allocated by the criterion of Specification Required, as defined by [RFC5226]. IANA SHOULD allocate new Result Code values from this range sequentially beginning at 0x1000. The initial contents of the ANCP Message Types registry are as follows:

Result Code	One-line description	Reference
0x0	No result	RFC 6320
0x2	Invalid request message	RFC 6320
0x6	One or more of the specified ports are down	RFC 6320
0x13	Out of resources	RFC 6320
0x51	Request message type not implemented	RFC 6320
0x53	Malformed message	RFC 6320
0x54	Mandatory TLV missing	RFC 6320
0x55	Invalid TLV contents	RFC 6320
0x500	One or more of the specified ports do not exist	RFC 6320
0x501	Loopback test timed out	RFC 6320
0x502	Reserved	RFC 6320
0x503	DSL access line status showtime	RFC 6320
0x504	DSL access line status idle	RFC 6320
0x505	DSL access line status silent	RFC 6320
0x506	DSL access line status training	RFC 6320
0x507	DSL access line integrity error	RFC 6320
0x508	DSLAM resource not available	RFC 6320
0x509	Invalid test parameter	RFC 6320

10.3. ANCP Port Management Function Registry

IANA has created a new ANCP Port Management Function registry, with the following initial entries. Additions to this registry will be by Standards Action, as defined by [RFC5226]. Values may range from 0 to 255. IANA SHOULD assign values sequentially beginning with 1, taking account of the values already assigned below.

> NOTE: Future extensions of ANCP may need to establish sub-registries of permitted X-Function values for specific values of Function.

Function Value	Function Name	Reference
0	Reserved	RFC 6320
8	Configure Connection Service Data	RFC 6320
9	Remote Loopback	RFC 6320

10.4. ANCP Technology Type Registry

IANA has created a new ANCP Technology Type registry, with additions by Expert Review, as defined by [RFC5226]. The Technology Type MUST designate a distinct access transport technology. Values may range from 0 to 255. IANA SHOULD assign new values sequentially beginning at 2, taking into account of the values already assigned below. The initial entries are as follows:

Tech Type Value	Tech Type Name	Reference
0	Not technology dependent	RFC 6320
1	Passive Optical Network (PON)	RFC 6320
5	Digital Subscriber Line (DSL)	RFC 6320
255	Reserved	RFC 6320

10.5. ANCP Command Code Registry

IANA has created a new ANCP Command Code registry, with additions by Standards Action, as defined by [RFC5226]. Values may range from 0 to 255. IANA SHOULD assign new values sequentially beginning with 1. The initial entry is as follows:

Command Code Value	Command Code Directive Name	Reference
0	Reserved	RFC 6320

10.6. ANCP TLV Type Registry

IANA has created a new ANCP TLV Type registry. Values are expressed in hexadecimal and may range from 0x0000 to 0xFFFF. Additions in the range 0x0000 to 0x1FFF are by IETF Review, as defined by [RFC5226]. IANA SHOULD assign new values in this range sequentially beginning at 0x100, taking account of the assignments already made below. Additions in the range 0x2000 to 0xFFFF are by Specification Required, again as defined by [RFC5226]. IANA SHOULD assign new values in this range sequentially beginning at 0x2000. In both cases, the documentation of the TLV MUST provide:

- a TLV name following the convention used for the initial entries (capitalized words separated by hyphens);

- a brief description of the intended use;

- a precise description of the contents of each fixed field, including its length, type, and units (if applicable);
- identification of any mandatory encapsulated TLVs;
- an indication of whether optional TLVs may be encapsulated, with whatever information is available on their identity (could range from a general class of information to specific TLV names, depending on the nature of the TLV being defined).

The initial entries are as follows:

Type Code	TLV Name	Reference
0x0000	Reserved	RFC 6320
0x0001	Access-Loop-Circuit-ID	RFC 6320
0x0002	Access-Loop-Remote-ID	RFC 6320
0x0003	Access-Aggregation-Circuit-ID-ASCII	RFC 6320
0x0004	DSL-Line-Attributes	RFC 6320
0x0005	Service-Profile-Name	RFC 6320
0x0006	Access-Aggregation-Circuit-ID-Binary	RFC 6320
0x0007	OAM-Loopback-Test-Parameters	RFC 6320
0x0008	Opaque-Data	RFC 6320
0x0009	OAM-Loopback-Test-Response-String	RFC 6320
0x0011	Command	RFC 6320
0x0081	Actual-Net-Data-Rate-Upstream	RFC 6320
0x0082	Actual-Net-Data-Rate-Downstream	RFC 6320
0x0083	Minimum-Net-Data-Rate-Upstream	RFC 6320
0x0084	Minimum-Net-Data-Rate-Downstream	RFC 6320
0x0085	Attainable-Net-Data-Rate-Upstream	RFC 6320
0x0086	Attainable-Net-Data-Rate-Downstream	RFC 6320
0x0087	Maximum-Net-Data-Rate-Upstream	RFC 6320
0x0088	Maximum-Net-Data-Rate-Downstream	RFC 6320
0x0089	Minimum-Net-Low-Power-Data-Rate-Upstream	RFC 6320
0x008A	Minimum-Net-Low-Power-Data-Rate-Downstream	RFC 6320
0x008B	Maximum-Interleaving-Delay-Upstream	RFC 6320
0x008C	Actual-Interleaving-Delay-Upstream	RFC 6320
0x008D	Maximum-Interleaving-Delay-Downstream	RFC 6320
0x008E	Actual-Interleaving-Delay-Downstream	RFC 6320
0x008F	DSL-Line-State	RFC 6320
0x0090	Access-Loop-Encapsulation	RFC 6320
0x0091	DSL-Type	RFC 6320
0x0106	Status-Info	RFC 6320
0x1000	Target (single access line variant)	RFC 6320
0x1001 - 0x1020	Reserved for Target variants	RFC 6320

10.7. ANCP Capability Type Registry

IANA has created a new ANCP Capability Type registry, with additions by Standards Action as defined by [RFC5226]. Values may range from 0 to 255. IANA SHOULD assign values sequentially beginning at 5. The specification for a given capability MUST indicate the Technology Type value with which it is associated. The specification MUST further indicate whether the capability is associated with any capability data. Normally, a capability is expected to be defined in the same document that specifies the implementation of that capability in protocol terms. The initial entries in the ANCP capability registry are as follows:

Value	Capability Type Name	Tech Type	Capability Data?	Reference
0	Reserved			RFC 6320
1	DSL Topology Discovery	5	No	RFC 6320
2	DSL Line Configuration	5	No	RFC 6320
3	Reserved			RFC 6320
4	DSL Line Testing	5	No	RFC 6320

10.8. Joint GSMP / ANCP Version Registry

IANA has created a new joint GSMP / ANCP Version registry. Additions to this registry are by Standards Action as defined by [RFC5226]. Values may range from 0 to 255. Values for the General Switch Management Protocol (GSMP) MUST be assigned sequentially beginning with 4 for the next version. Values for the Access Network Control Protocol (ANCP) MUST be assigned sequentially beginning with 50 for the present version. The initial entries are as follows:

Version	Description	Reference
1	GSMP Version 1	RFC 1987
2	GSMP Version 2	RFC 2297
3	GSMP Version 3	RFC 3292
50	ANCP Version 1	RFC 6320

11. Security Considerations

Security of ANCP is discussed in [RFC5713]. A number of security requirements on ANCP are stated in Section 8 of that document. Those applicable to ANCP itself are copied to the present document:

- The protocol solution MUST offer authentication of the AN to the NAS.

- The protocol solution MUST offer authentication of the NAS to the AN.

- The protocol solution MUST allow authorization to take place at the NAS and the AN.

- The protocol solution MUST offer replay protection.

- The protocol solution MUST provide data-origin authentication.

- The protocol solution MUST be robust against denial-of-service (DoS) attacks. In this context, the protocol solution MUST consider a specific mechanism for the DoS that the user might create by sending many IGMP messages.

- The protocol solution SHOULD offer confidentiality protection.

- The protocol solution SHOULD ensure that operations in default configuration guarantee a low number of AN/NAS protocol interactions.

Most of these requirements relate to secure transport of ANCP. Robustness against denial-of-service attacks partly depends on transport and partly on protocol design. Ensuring a low number of AN/NAS protocol interactions in default mode is purely a matter of protocol design.

For secure transport, either the combination of IPsec with IKEv2 (references below) or the use of TLS [RFC5246] will meet the requirements listed above. However, the use of TLS has been rejected. The deciding point is a detail of protocol design that was unavailable when [RFC5713] was written. The ANCP adjacency is a major point of vulnerability for denial-of-service attacks. If the adjacency can be shut down, either the AN clears its state pending reestablishment of the adjacency, or the possibility of mismatches between the AN’s and NAS’s view of state on the AN is opened up. Two ways to cause an adjacency to be taken down are to modify messages so that the ANCP agents conclude that they are no longer synchronized, or to attack the underlying TCP session. TLS will protect message contents but not the TCP connection. One has to use either IPsec or the TCP authentication option [RFC5925] for that. Hence, the conclusion that ANCP MUST run over IPsec with IKEv2 for authentication and key management.

In greater detail: the ANCP stack MUST include IPsec [RFC4301] running in transport mode, since the AN and NAS are the endpoints of the path. The Encapsulating Security Payload (ESP) [RFC4303] MUST be used, in order to satisfy the requirement for data confidentiality. ESP MUST be configured for the combination of confidentiality, integrity, and anti-replay capability. The traffic flow confidentiality service of ESP is unnecessary and, in fact, unworkable in the case of ANCP.

IKEv2 [RFC5996] is also REQUIRED, to meet the requirements for mutual authentication and authorization. Since the NAS and AN MAY be in different trust domains, the use of certificates for mutual authentication could be the most practical approach. However, this is up to the operator(s) concerned.

The AN MUST play the role of initiator of the IKEv2 conversation.

12. Contributors

Swami Subramanian was an early member of the authors’ team. The ANCP Working Group is grateful to Roberta Maglione, who served as design team member and primary editor of this document for two years before stepping down.

13. Acknowledgements

The authors would like to thank everyone who provided comments or inputs to this document. The authors acknowledge the inputs provided by Wojciech Dec, Peter Arberg, Josef Froehler, Derek Harkness, Kim Hyldgaard, Sandy Ng, Robert Peschi, and Michel Platnic, and the further comments provided by Mykyta Yevstifeyev, Brian Carter, Ben Campbell, Alexey Melnikov, Adrian Farrel, Robert Sparks, Peter St. Andre, Sean Turner, Dan Romascanu, Brian Carter, and Michael Scott.

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3292] Doria, A., Hellstrand, F., Sundell, K., and T. Worster, "General Switch Management Protocol (GSMP) V3", RFC 3292, June 2002.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

14.2. Informative References

[G.993.2] "ITU-T Recommendation G.993.2, Very high speed digital subscriber line transceivers 2 (VDSL2)", 2006.

[G.998.1] "ITU-T Recommendation G.998.1, ATM-based multi-pair bonding", 2005.

[G.998.2] "ITU-T Recommendation G.998.2, Ethernet-based multi-pair bonding,", 2005.

[IEEE802.1Q] IEEE, "IEEE 802.1Q-2005, IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks - Revision", 2005.

[IEEE802.1ad] IEEE, "IEEE 802.1ad-2005, Amendment to IEEE 802.1Q-2005. IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks - Revision - Amendment 4: Provider Bridges", 2005.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC4649] Volz, B., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option", RFC 4649, August 2006.

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5713] Moustafa, H., Tschofenig, H., and S. De Cnodder, "Security Threats and Security Requirements for the Access Node Control Protocol (ANCP)", RFC 5713, January 2010.

[RFC5851] Ooghe, S., Voigt, N., Platnic, M., Haag, T., and S. Wadhwa, "Framework and Requirements for an Access Node Control Mechanism in Broadband Multi-Service Networks", RFC 5851, May 2010.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

[TR-058] Broadband Forum, "TR-058, Multi-Service Architecture & Framework Requirements", September 2003.

[TR-059] Broadband Forum, "TR-059, DSL Evolution - Architecture Requirements for the Support of QoS-Enabled IP Services", September 2003.

[TR-092] Broadband Forum, "TR-092, Broadband Remote access server requirements document", 2005.

[TR-101] Broadband Forum, "TR-101, Architecture & Transport: Migration to Ethernet Based DSL Aggregation", 2005.

[TR-147] Broadband Forum, "TR-147, Layer 2 Control Mechanism For Broadband Multi-Service Architectures", 2008.

[US_ASCII] American National Standards Institute, "Coded Character Set - 7-bit American Standard Code for Information Interchange", ANSI X.34, 1986.

Authors' Addresses

Sanjay Wadhwa
Alcatel-Lucent
701 E Middlefield Rd
Mountain View, CA 94043-4079
USA

EMail: sanjay.wadhwa@alcatel-lucent.com

Jerome Moisand
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

EMail: jmoisand@juniper.net

Thomas Haag
Deutsche Telekom
Heinrich-Hertz-Strasse 3-7
Darmstadt 64295
Germany

EMail: haagt@telekom.de

Norbert Voigt
Nokia Siemens Networks
Siemensallee 1
Greifswald 17489
Germany

EMail: norbert.voigt@nsn.com

Tom Taylor (editor)
Huawei Technologies
1852 Lorraine Ave
Ottawa
Canada

EMail: tom111.taylor@bell.net