

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 3, 2009

P. Champagne  
W. Dec  
Cisco Systems  
S. Wadhwa  
Juniper Networks  
S. De Cnodder  
Alcatel-Lucent  
R. Maglione  
Telecom Italia  
July 2, 2008

**Multicast Control Extensions for ANCP**  
**draft-ancp-mc-extensions-00.txt**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on January 3, 2009.

Abstract

This draft is aimed at describing the ANCP protocol extensions required to support the NAS initiated ANCP Multicast Control use case described in ANCP framework draft. It proposes the definition of new ANCP message types, along with well known TLVs.

## Table of Contents

<a href="#">1.</a>	Conventions . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">3.</a>	ANCP Header . . . . .	<a href="#">3</a>
<a href="#">4.</a>	ANCP Multicast Control Messages . . . . .	<a href="#">5</a>
<a href="#">4.1.</a>	Multicast Replication Control Request Message . . . . .	<a href="#">5</a>
<a href="#">5.</a>	ANCP Multicast Response Messages . . . . .	<a href="#">12</a>
<a href="#">5.1.</a>	Multicast Status Response Message . . . . .	<a href="#">13</a>
<a href="#">6.</a>	Security Considerations . . . . .	<a href="#">15</a>
<a href="#">7.</a>	IANA Considerations . . . . .	<a href="#">15</a>
<a href="#">8.</a>	Normative References . . . . .	<a href="#">15</a>
	Authors' Addresses . . . . .	<a href="#">16</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">18</a>



## 1. Conventions

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

## 2. Introduction

This draft is aimed at describing the ANCP protocol extensions required to support the ANCP Multicast Control use case described in ANCP framework draft [\[ANCPFW\]](#) [Section 3.4](#). This proposal aims at collecting feedback from this community in order to work towards consensus and the text derived from this document would be intended to be incorporated in "ANCP Protocol I-D" [\[ANCPPR\]](#).

## 3. ANCP Header

The format of the ANCP multicast control messages starts with the common GSMP header as in the case of the existing ANCP implementation. Following is the format of this header:

```

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 (0x88-0C)           |           Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Vers | Sub | Message Type | Result |           Code           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Partition ID |           Transaction Identifier           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|I|           SubMessage Number           |           Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                           |
~                               Message Payload                               ~
|                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The Result field derived from GSMP [\[RFC3292\]](#) has the following codes:

Nack:

Res = 0x1 - Renamed result code indicating that no response is expected to the message other than in cases of failuire caused during the processing of the message contents or that of the contained directive(s).



**AckAll:**

Res = 0x2 - Result code indicating that a response to the message is requested in all cases. It is specifically intended to be used in some cases for Request messages only, and is not to be used in Event messages.

**Success:**

Res = 0x3 - Allowed to be set in Response Messages only to communicate successful execution of all directives in a previous Request.

**Failure:**

Res = 0x4 - Allowed to be set in Response Messages Only to communicate failure of execution of one directives in a previous Request.

The Transaction Identifier field is used to distinguish between Request messages and to associate a Response message to a Request. Applications that require such response correlation MUST set the Transaction Identifier to a value in the range (1,  $2^{24} - 1$ ). When used in this manner, the Transaction Identifier sequencing MUST be maintained independently for each ANCP adjacency and per message type. Furthermore, it SHOULD be incremented linearly for each new message of the given type, cycling back to 1 after running the full range. The transport protocol used by ANCP (TCP currently) guarantees sequenced in order delivery, and thus the reception of a message that does not fit into the expected transaction id sequence can signify a problem with the sender. When such an event is detected, a "Transaction-id Out of Sequence" error message response MUST be sent to the requester, and the Request ignored until sequencing resumes. Message types not requiring response messages to be correlated to requests, such as those conveying events, SHOULD set the Transaction Id field to 0x0.

In the event of an ANCP transport protocol failure, all pending ANCP messages destined to the disconnected recipient can be discarded until the transport is re-established following which the Transaction Identifier is re-initialized.

The value of the Transaction Identifier in a Response message MUST be set to that of the respective Request message. This allows the Requester to correlate the Response to the original Request. The



Transaction Identifier is not used in ANCP adjacency messages. Also, other ANCP applications not requiring it SHOULD set the Transaction Identifier to 0x0 in their messages.

All TLVs within the ANCP message have to be 32 bit aligned, and when necessary padded with 0s to the 32 bit boundary. The padding is not reflected in the message length field.

#### 4. ANCP Multicast Control Messages

##### 4.1. Multicast Replication Control Request Message

The Replication Control Request Message Type 0x90 (TBC) is sent by the ANCP Server (NAS) to the ANCP Client (AN) with a directive to either add (join) or delete (leave) one or more multicast flows on a target object identified in the content of the message. An ANCP Client will use a Replication Control Response message when conveying the outcome of the directive, and this message type is covered in [Section 5.1](#).

The sender of a request message MUST set the Result field to either "AckAll" or "NAck", and SHOULD use "NAck" by default. Furthermore it SHOULD use the same Result field code for all Replication Control Messages sent, i.e. Result field changes SHOULD be avoided. The sender MUST populate the ANCP Transaction Identifier field with a distinct non-zero, linearly incrementing value for each Request per adjacency, as described in [Section 3](#).

The ANCP Multicast Replication Control message payload contains the following TLVs:

```

 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 = Target-Type TLV      |      Length of Target-Info      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|
|                               Value = Target-Info                               |
|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      Type = Command TLV      |      Length of Command Info      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|
|                               Value = Command Info                               |
|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Target-Type:





See [Section 4.1.1](#). The Target-Ttype TLV (0xTBD) can only feature once in a replication Control Request Message, and it MUST be the first TLV in the message. Only one such TLV is allowed in this message type.

Length of Target-Info:

See [Section 4.1.1](#)

Target Info:

See [Section 4.1.1](#)

Command TLV:

The Command TLV (0xTBD) contains the multicast flow directive(s) for the target and any additional parameters passed via sub-TLVs. It MUST immediately follow the Target-type-TLV. See [Section 4.1.2](#)

Length of Command Info:

Includes sub-TLVs. See [Section 4.1.2](#)

Command Info:

Command information as defined below in section [Section 4.1.2](#).

The contents of the Command TLV for the Multicast Control Request Message are defined to be as follows:

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								
Command Code										R O M    Flags										Command Length																			
Addr Family										Encoding Type										Multicast Source Address																			
Addr Family										Encoding Type										Multicast Flow Address																			

Command Code:

Command directive: 0x01 - Add; 0x02 - Delete; 0x03 - Delete All.

Command Length:



Length in bytes of each Command including multicast flow address, but excluding the Command Code header and flags.

Flags:

8 bit General purpose Flag field. Currently the following flags are defined:

R -

Resource Admitted Flag. Set to 1 in an add command to indicate that the flow resources have been reserved by admission control, 0 otherwise. Not used in delete command.

O -

Flow Accounting. When set in add command indicates that byte accounting for the flow is to commence.

M -

When set indicates that multicast flow is SSM and the address-family-element set MUST specify the source and group addresses. When not set indicates that multicast flow is ASM and address-family-element MUST specify the group address, and the Source Address field is to be omitted. Note: Details regarding the interpretation of the Add and Delete command in view of SSM filter modes are to be addressed for future revisionins of this document.

Address Family:

The address family used

The unicast source address/mask follows the format defined in [\[IANAAEA\]](#)



Encoded-Unicast-address: Takes the following format:

```

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
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Addr Family   | Encoding Type |      Unicast Address           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Encoding Type:

The type of encoding used within a specific Address Family.  
 The value '0' is reserved for this field, and represents  
 the native encoding of the Address Family.  
 The address as represented by the given Address Family and  
 Encoding Type.

Address:

The address as represented by the given Address Family and  
 Encoding Type.

The padding will be done after both addresses so that it is 32-bit  
 aligned. As an example for IPv4:

```

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
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| CmdCode=0x01 | 0 0 1   Flags |      Command Length           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| AddrFamily 0x1| Enc Type  0x0 |   Src Address first 2 bytes   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Src Address last 2 bytes      | AddrFamily 0x1| Enc Type 0x0 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                               Multicast Address (4 bytes)      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

In the above example, no padding is required.

A received Request Message containing an unrecognized Target-Type TLV  
 MUST be communicated to the sender as an error in a Response Message  
 indicating the "Unrecognised port Type - 0x04" error. The reception  
 of a Request Message, or any ANCP message, that is found to have  
 mandatory TLVs missing is to be addressed as part of a ANCP base  
 protocol mechanism yet to be defined.

Each Multicast Control Request Message may contain one or more  
 command directives, each encapsulated by their own Command TLVs. The  
 sender MUST use separate Command TLVs for each distinct multicast  
 flow. When successive commands relate to a given target port and  
 flow, the state of features controlled or affected by flags as well



as by optional attributes received in the Request message, are to be interpreted as replacing any such previous state for that port and flow. As an example, successive Request messages containing add commands for a given port and flow, but differing in the accounting flag setting should be interpreted as affecting the state of the accounting feature.

The recipient of a Request message is to run an implicit directive numbering across the multiple directives in the message. The numbering is to start from 0x01 for each directive in a given ANCP Request message, and be restarted for subsequent messages. The recipient MUST process the directives in the order of reception (i.e. The first command after the target type TLV info is to be executed first), and use the derived directive number in any response messages, besides the Transaction ID.

The processing/execution of multiple directives contained in a single Multicast Control Request message MUST be interrupted at the first error, and the remaining commands in the Request message discarded. In such a case a multicast control response message MUST be sent indicating the command number that resulted in the error along with the error code.

When the strict sequenced processing of the directives in a single Multicast Control Request message is not required the directives MUST be distributed across separate Request messages.

Each command directive is equipped with an 8-bit Flags field that allows specification of Multicast ASM or SSM modes of operation, and an indication of other features or conditions attached to this command (e.g. To enable accounting for the flow, etc). Unassigned flags are reserved for future use, and could in the future be subject of the capability negotiation. When receiving a Multicast request Message containing an unrecognized Flag set (to 1), a recipient MUST interpret it as an error, and generate an Response message indicating the error.

The multicast flow subject to the command is specified by means of one or two well known Address Family designators ([\[IANAAEA\]](#)), the IPv4-Address-Family (0x01) and the IPv6-Address-Family (0x02). When the M flag is set the two Address-Family tuples MUST be present in the strict order specifying the multicast flow source and group respectively. When the M flag is cleared only one Address-Family is allowed, specifying the multicast flow.

For future extensibility, each command may also have additional TLVs appended following the command and multicast flow information (referred to as "TLVs" in the message format above). Unrecognized





TLVs SHOULD be silently discarded. The figure below is an example of a Request message that would result in a swap from multicast SSM flows 10.1.1.1, 239.23.2.2, to 10.2.1.1, 239.55.3.2 on the target port identified by the "Access Loop Circuit ID":

```

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 (0x88-0C)          |          Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Vers | Sub | MessageType=90 | 0x02 |          Code          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Partition ID |          Transaction Identifier = 0001          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|I|          SubMessage Number          |          Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Type = Target-type 0x1000  |          Target TLV Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Access-Loop-Circuit-ID 0x0001 |          Circuit-ID Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |
~                               Access Loop Circuit ID                               ~
|                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Type = Command TLV          |          Command-TLV Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Cmd Code=0x02 | 0 0 1          |          Command Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| AddrFamily 01 | EncType 0x0  | Mcast Source: 10.1.1.1          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| AddrFamily 01 | EncType 0x0  | Mcast Flow : 239.23.2.2          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Type = Command-TLV          |          Command-TLV Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Cmd Code=0x01 | 0 0 1          |          Command Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| AddrFamily 01 | EncType 0x0  | Mcast Source: 10.2.1.1          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| AddrFamily 01 | EncType 0x0  | Mcast Flow: 239.55.3.2          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

#### 4.1.1.1. Target Type TLV

The Target-type-TLV (TBD) is intended to be a general well known TLV allowing the representation of different types of objects. Its use is not restricted to any specific Message Type.



```

 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 = Target-type       |   Target-TLV Length       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
~                               Target Info                       ~
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

#### Target-type TLV

TLV (0xTBD) indicating the type of target being addressed.  
Numbers TBC. Tentative 0x1000 for single Access-Port.

#### Target-TLV Length:

Length in bytes of Target Info. Excludes TLV header

#### Target Info:

Target information as defined for each the given target  
type. The field can consist of sub-TLVs.

In its simplest form, when targeting a single access line the Target-Type-TLV will be set to a value of (0xTBD), and carry in its payload one or more sub-TLVs identifying the target. The following example illustrates the message format for a single port identified by an Access-Loop-Circuit-ID TLV (0x0001) that could be derived from a Port-UP message:

```

 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 = Target-type       |   Target-TLV Length       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Access-Loop-Circuit-ID=0x0001 |   Circuit-ID Length   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
~                               Access Loop Circuit ID           ~
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

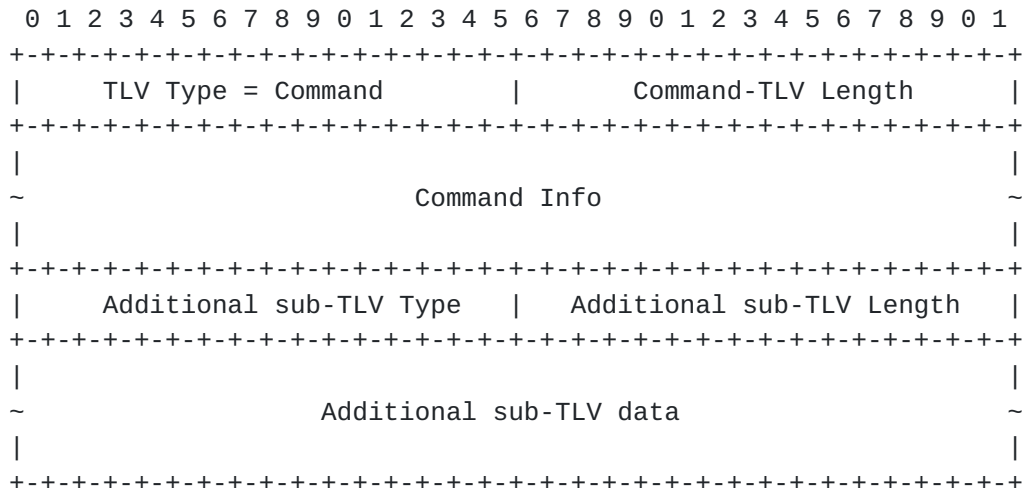
```

#### [4.1.2.](#) Command TLV

The Command TLV (TBD) is intended to be a general well known TLV allowing the encapsulation of one or more command directives in a TLV oriented message. The semantics of the command are allowed to be specified for each message type, ie different message types that choose to carry the Command TLV are expected to define the meaning of



the content of the payload, which could be re-used from those already defined elsewhere if appropriate.



Command TLV:

TLV (0xTBD) indicating the contents to be one or more command directives.

Command TLV Length:

Combined length in bytes of the data in Command Info and sub-TLV. Excludes the Command TLV header

Command-Info:

Command information as defined for each message type. The field can consist of sub-TLVs.

Additional sub-TLV:

Additional sub-TLVs can be present in a command TLV. Any such sub-TLVs must directly follow each command.

Additional sub-TLV Length:

Number of actual bytes contained in the value portion of each additional sub-TLV

## 5. ANCP Multicast Response Messages



### **5.1. Multicast Status Response Message**

The Multicast Status Response Message (Message Type 0x91 - TBC) is sent by the ANCP Client (AN) to the ANCP Server (NAS) in response to a Replication Control Request Message and its command directives. A Response message MUST use the same ANCP Transaction ID as that in the original Request Message. The Success or Failure status is reported in the Result field of the ANCP header as described in [Section 3](#).

A Response Message indicating Success SHOULD simply consist only of the base ANCP header with no body, however the message MAY contain one or more TLVs that are meant to communicate any relevant information to an application. The payload of a Response Message indicating Failure MUST contain an Status-Info TLV (0xTBD), as defined in [Section 5.1.1](#), as its first TLV and SHOULD be followed by the Target-Type-TLV and Port-info. Other TLVs MAY be present. A Response message indicating Failure MUST be sent whenever a Multicast Control Request cannot be fulfilled or results in an execution error. The Cmnd Nmbr parameter in the Status-Info TLV contained by the Response Message is to indicate the number of the command in the Request that resulted in an error.

- 0x00 - Success
- 0x01 - Malformed message
- 0x02 - Command not supported
- 0x03 - Flag set but not supported
- 0x04 - Unrecognized Target Type
- 0x05 - Unsupported Address Family
- 0x06 - Malformed flow address
- 0x07 - No resources
- 0x08 - Unknown Port
- 0x09 - Target port down
- 0x0a - Configuration error (such as Port not enabled for multicast)
- 0x0b - Multicast flow does not exist
- 0x0c - Unsupported address encoding
- 0x0d - Additional info needed to execute command (payload MAY contain an indication of the expected info)
- 0x0e - Multicast flow count exceeded
- 0x0f - M Flag set, but no IP Source address provided
- 0x10 - Transaction-id out of sequence

An example of a failure message for an invalid address, including the Target-Type TLV for a 4 byte "Access Loop Circuit ID", followed by TLV padding, is as follows:





```

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 (0x88-0C)      |      Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Vers | Sub | MessageType=91 | 0x4 |      Code      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Partition ID |      Transaction Identifier = 0001      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| I | SubMessage Number |      Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Status-info-TLV=TBD      |      Status-TLV-Length=14      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0x06 | 0x01 |      Target-type-TLV=0x100      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Target-TLV-Length      |      0x0001      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Client-ID Length      |      Access Loop Circuit ID      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      ID cont...      | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

#### 5.1.1.1. Status-Info TLV

The Status-info-TLV is intended to be a general well known TLV used to convey the status code regarding commands and/or requests. The format of the Status-Info-TLV (TBD) is shown below.

```

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 = Status-info      |      Status TLV Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Result Code | Cmnd Nmbr |      Error Message Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Error Message (aligned to 4 bytes length)      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      TLVs...      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Status-info TLV:

TLV (0xTBD) conveying the status or error response of a command

Status TLV Length:

Specifies the length in bytes of the Status Info TLV payload. Excludes the TLV header



**Result Code:**

Conveys the result code for the command or message, as defined by the application.

**Cmnd Nmbr:**

Contains the command number copied from the Request message. The value of 0 is used whenever the error is not specific to a command.

**Error Message Length:**

Contains the length of an optional error message or 0 if none.

**TLVs:**

This field is of indeterminate length, and contains zero or more of the TLVs associated with the Status-info-TLV.

## **6. Security Considerations**

Security of the ANCP protocol is discussed in [[ANCPSEC](#)]

## **7. IANA Considerations**

This proposal defines several new ANCP message types and well known mandatory TLVs. As such an application for an IANA assignment will need to be made.

## **8. Normative References**

- [ANCPFW] 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", [draft-ietf-ancp-framework-06.txt](#), work in progress, October 2007.
- [ANCPPR] Wadhwa, S., Moisand, J., Subramanian, S., Haag, T., and N. Voigt, "Protocol for Access Node Control Mechanism in Broadband Networks", [draft-ietf-ancp-protocol-02.txt](#) work in progress, March 2006.
- [ANCPSEC] Moustafa, H., Tschofenig, T., and S. De Cnodder, "Security



Threats and Security Requirements for the Access Node  
Control Protocol (ANCP)",  
[draft-ietf-ancp-security-threats-05.txt](#) work in progress,  
March 2008.

[IANAAEA] "http://www.iana.org/assignments/address-family-numbers",  
2005.

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

[RFC3292] Doria, A. and et al, "General Switch Management Protocol  
(GSMP) V3", June 2002.

#### Authors' Addresses

Philippe Champagne  
Cisco Systems  
7025-4 Kit Creek Rd.  
Research Triangle Park, NC 27709  
USA

Phone:  
Email: [pchamp@cisco.com](mailto:pchamp@cisco.com)

Wojciech Dec  
Cisco Systems  
Haarlerbergweg 13-19  
1101 CH Amsterdam  
The Netherlands

Phone:  
Email: [wdec@cisco.com](mailto:wdec@cisco.com)

Sanjay Wadhwa  
Juniper Networks  
10 Technology Park Drive  
Westford, MA 01886  
USA

Phone: phone  
Email: [swadhwa@juniper.net](mailto:swadhwa@juniper.net)



Stefaan De Cnodder  
Alcatel-Lucent  
Copernicuslaan 50  
B-2018 Antwerp  
Belgium

Phone: +32 3 240 85 15  
Email: [stefaan.de\\_cnodder@alcatel-lucent.be](mailto:stefaan.de_cnodder@alcatel-lucent.be)

Roberta Maglione  
Telecom Italia  
via Reiss Romoli 274  
Torino  
Italy

Phone: +390112285007  
Email: [roberta.maglione@telecomitalia.it](mailto:roberta.maglione@telecomitalia.it)





## Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

