PacketCable™ 2.0

Electronic Surveillance Intra-Network Specification

PKT-SP-ES-INF-I02-061013

ISSUED

Notice

This PacketCable specification is a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. (CableLabs®) for the benefit of the cable industry. Neither CableLabs, nor any other entity participating in the creation of this document, is responsible for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document by any party. This document is furnished on an AS-IS basis and neither CableLabs, nor other participating entity, provides any representation or warranty, express or implied, regarding its accuracy, completeness, or fitness for a particular purpose.

Document Status Sheet

Document Control Number:	PKT-SP-ES-INF-I02-061013			
Document Title:	Electronic Surveillance Intra-Network Specification			
Revision History:	I01 – Released 04/06/2006 I02 – Released 10/13/2006			
Date:	October 13, 2006			
Status:	~~Work in Progress~~	~~Draft~~	Issued	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	C~~L/ Member/ Vendor~~	Public

Key to Document Status Codes:

Work in Progress An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.

Draft A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.

Issued A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.

Closed A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks:

Contents

Figures

Tables

This page left intentionally blank.

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to specify the lawfully authorized electronic surveillance requirements for the components internal to the PacketCable architecture that support SIP based PacketCable clients.

1.2 Scope

The scope is limited to interfaces between components internal to the PacketCable network that support SIP based PacketCable clients and the Delivery Function (DF). Requirements for the interface to the law enforcement agency Collection Function are the out of the scope of this document.

1.3 Assumptions

The following assumptions have been made:

- The IMS Charging ID, ICID, will be passed between PacketCable components (P-Charging-Vector) so that it can be used as a correlation identifier by the DF for Event Messages that it receives. When the call involves an Application Server (AS), there are two cases to consider:
    - The Serving-Call Session Control Function (S-CSCF) passes the INVITE to an AS over the ISC interface based on origination or termination filter criteria. In this case, the S-CSCF insures that the ICID is propagated after the INVITE returns from the AS.
    - The call terminates on the AS and the AS originates a new call as a result such that the AS is the only component that knows that the two call legs are related. In this case, the assumption is that the AS propagates the ICID across the call legs.

1.4 Organization of document

Section 2 lists the references. Sections 3 and 4 contain acronyms, terms and definitions. Section 5 is an informative section that contains the technical overview while Section 6 contains the detailed requirements. Section 7 contains informative call flows. Annex A and Annex B contain interface definitions for the Diameter interface for Call Identifying information (CII) and the content tapping interface TAP-Management Information Base (TAP-MIB) respectively.

1.5 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST" This word means that the item is an absolute requirement of this specification.

"MUST NOT" This phrase means that the item is an absolute prohibition of this specification.

"SHOULD" This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

"SHOULD NOT" This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

"MAY" This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[CMSS] PacketCable 1.5 CMS to CMS Signaling Specification, PKT-SP-CMSS1.5-I02-05812, August 12, 2005, Cable Television Laboratories, Inc.

[CPD] PacketCable Control Point Discovery Specification, PKT-SP-CPD-I02-061013, October 13, 2006, Cable Television Laboratories, Inc.

[MIB-CLABDEF] CableLabs Definition MIB Specification, CL-SP-MIB-CLABDEF-I05-050408, April 8, 2005, Cable Television Laboratories, Inc.

[RFC 3455] IETF RFC 3455, Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP), January 2003.

[RFC 3414] IETF RFC 3414/STD0062, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), December 2002.

[RFC 3415] IETF RFC 3415/STD0062, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3588] IETF RFC 3588, Diameter Base Protocol, September 2003.

[RFC 3826] IETF RFC 3826 The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model.

[TS 32.299] 3GPP TS 32.299 v6.6.6, Diameter charging applications, March 2006.

2.2 Informative References

This specification uses the following informative references.

[ACCT] PacketCable Accounting Specification, PKT-SP-ACCT-I02-061013, October 13, 2006, Cable Television Laboratories, Inc.

[DQOS] PacketCable 1.5 Dynamic Quality of Service Specification, PKT-SP-DQOS1.5-I02-050812, August 12, 2005, Cable Television Laboratories, Inc.

[EM] PacketCable 1.5 Event Message Specification, PKT-SP-EM1.5-I02-050812, August 12, 2005, Cable Television Laboratories, Inc.

[NFT TR] PacketCable NAT and Firewall Traversal Specification, PKT-TR-NFT-V02-061013, October 13, 2006, Cable Television Laboratories, Inc.

[RSTF] PacketCable Residential SIP Telephony Feature Specification, PKT-SP-RSTF-I01-060927, September 27, 2006, Cable Television Laboratories, Inc.

[SEC TR] PacketCable Security Technical Report, PKT-TR-SEC-V02-061013, October 13, 2006, Cable Television Laboratories, Inc.

[TGCP] PacketCable 1.5 PSTN Gateway Call Signaling Protocol Specification, PKT-SP-TGCP1.5-I02-050812, August 12, 2005, Cable Television Laboratories, Inc.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027;
  Phone 303-661-9100; Fax 303-661-9199; Internet: http://www.cablelabs.com/
- Internet Engineering Task Force (IETF), Internet: http://www.ietf.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Control Point Discovery (CPD) This term is defined in [CPD]. The PacketCable intercept architecture uses CPD to discover call content intercept access points.

Delivery Function (DF) The Delivery Function provides the interface with the Law Enforcement Agency. It acts as a mediation function in that it takes the information supplied by the internal components and formats the information as required by the interface to the Law Enforcement Agency.

Collection Function (CF) The Collection Function collects and decodes the information provided by the DF and provides the necessary presentation and storage functionality as required by the Law Enforcement Agency. The Collection Function is typically provided by and contained within the Law Enforcement Agency and as such is out of scope with respect to this document.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations.

B2BUA	Back To Back User Agent
BCID	Billing Correlation Identity
BGCF	Breakout Gateway Control Function
CII	Call Identifying Information
CMTS	Cable Modem Termination System
CPD	Control Point Discovery
CSCF	Call Session Control Function
DSCP	Diffserv Code Point
DF	Delivery Function
DQOS	Dynamic Quality of Service
FMC	Fixed Mobile Convergence
HLR	Home Location Register
IAP	Intercept Access Point
I-CSCF	Interrogating CSCF
ICID	IMS Charging Identity
LAES	Lawfully-Authorized Electronics Surveillance
LEA	Law Enforcement Agency
LI	Lawful Intercept
MG	Media Gateway
MGC	Media Gateway Controller
MIB	Management Information Base
P-CSCF	Proxy CSCF
S-CSCF	Serving CSCF
SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
TGCP	Trunking Gateway Control Protocol
UE	User Equipment
USM	User-based Security Model
VACM	View-based Access Control Model

5 TECHNICAL OVERVIEW

5.1 PacketCable Functional Components for Electronic Surveillance

This section provides an overview of the PacketCable Lawfully Authorized Electronic Surveillance (LAES) architecture. The PacketCable LAES architecture is designed to provide law enforcement a full set of call related data records and call content for services defined by the PacketCable specifications. Functional components and interfaces within the PacketCable network that support electronic surveillance are described. Interfaces to external network entities, the law enforcement agency (LEA) Collection Function, are out side the scope for this document.

5.2 PacketCable LAES Architecture

The PacketCable LAES architecture is illustrated in Figure 1 below. Intercept access points on network elements for call related data and call content are identified. The interfaces shown between these elements support dynamic provisioning, call related data reporting and call content reporting.

As illustrated in Figure 1, the PacketCable Proxy-CSCF (P-CSCF) and S-CSCF are responsible for reporting call related events from subscribers assigned to these proxies. The Media Gateway Controller (MGC) also provides call related data for PSTN destination and call forwarding scenarios. The Home Location Register (HLR) is an intercept point for cellular roaming events when the HLR is present in the PacketCable network. The interface between the HLR and the Delivery Function is not defined by PacketCable. The Interrogating-CSCF (I-CSCF) and the Breakout Gateway Control Function (BGCF) provide off network routing and are potentially additional points for reporting call related data events. Application servers for PacketCable defined applications may also report call related events. Events from these intercept access points are reported to the Delivery Function. The Delivery Function correlates the set of events associated with a call, maps the events to a standard set of Collection Function messages , and sends the resulting messages to the LEA Collection Function. A single logical Delivery Function per operator is defined for this present release of the PacketCable intercept architecture.

The Cable Modem Termination System (CMTS) and Media Gateway (MG) are identified as intercept access points (IAPs) for call content. Media servers or aggregation routers in front of media servers may also be IAPs for content intercept. The DF discovers the content IAPs and then provisions the IAPs for reporting. The call content IAPs then report intercepted content back to the DF. The DF mediates and sends the content to the law enforcement agency Collection Function.

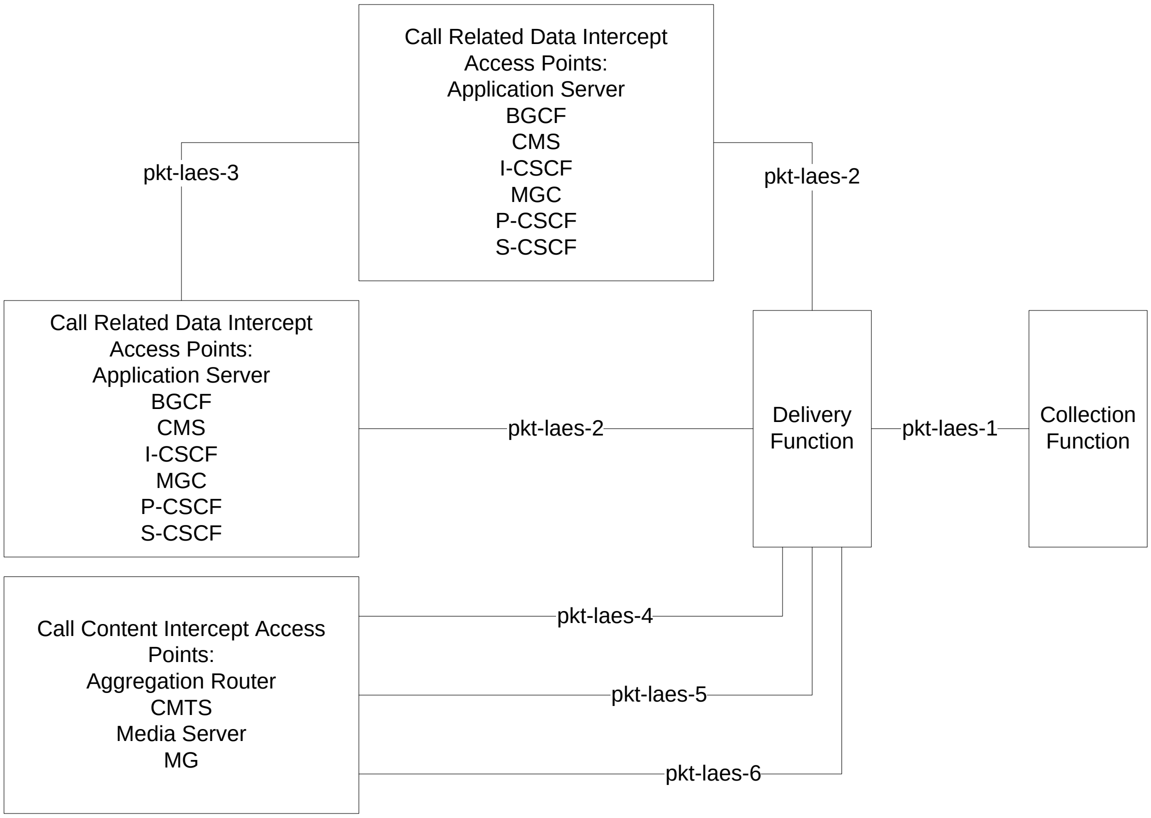

Figure 1 - PacketCable LAES Architecture

Table 1 indicates the protocol used on each of the intercept interfaces within the PacketCable network. Note that the HLR to delivery function interface is outside the scope of this specification.

Table 1 - PacketCable Internal LAES Interfaces

Reference Point	**PacketCable Network Elements**	**Reference Point Description**
PKT-LAES-1	DF - CF	Correlated call related data and call content are reported to the law enforcement agency Collection Function.
PKT-LAES-2	Session Control Element – DF	Intercept call related events are reported to the DF. This reference point is DIAMETER based.
PKT-LAES-3	Session Control Element – Session Control Element	Allows session control elements to dynamically provision intercept in peer elements for calls where the targeted subject's assigned control elements are no longer involved in the call. Call redirect is one example. This reference point is Session Initiation Protocol (SIP) based.
PKT-LAES-4	DF to Content Access Points	The DF dynamically provisions content intercept points. This reference point is SNMPv3 based.
PKT-LAES-5	Content Access Point to DF	Intercepted call content is reported to the DF. This reference point is media over UDP based.
PKT-LAES-6	DF to Content Access Points	The DF uses the Control Point Discovery Protocol to determine the appropriate intercept access point in the network for call content. This reference point is based on [CPD].

Table 2 lists the network elements that are potential call related data and call content intercept access points in the network.

Table 2 - Intercept Access Points

Intercept Access Points	PacketCable Network Elements
Call related data intercept access points	P-CSCF, S-CSCF, I-CSCF, MGC, PacketCable specified application servers, HLR
Call content intercept access points	CMTS, MG, media servers, aggregation routers

5.3 Electronic Surveillance Interfaces to CMS Systems

The PacketCable architecture is designed to support mixed CSCF and CMS based networks as a configurable option for the cable operator. Interception of a single call may span these networks under session transfer or redirect scenarios. Therefore, interception interoperability between Cuscus and CMSs is required. Interoperability requirements are placed on PacketCable call related data intercept access points.

PacketCable call related data intercept access points support content parameters in the P-DCS-LAES header format in order to allow for interoperability with CMS components. The format of the P-DCS-LAES header is specified in [CMSS]. In addition, PacketCable components will provide correlation information to the DF when a BCID appears in that header. Details on the format and use of this header are available in Section 6.

Interoperability requirements are also placed on CMS components. These requirements are specified in CMS related specifications.

5.4 PacketCable Defined Features

PacketCable includes feature capability sets that interact with lawful interception:

- Residential SIP Telephony (RST) feature set as specified in [RSTF].
- Wireless and cellular integration feature set.

The RST features are executed on the UE and RST specific application servers. The following RST servers are identified as intercept access points:

- Announcement or media server or aggregation router preceding the announcement server.
- Voice mail server or aggregation router preceding voice mail server.

Specialized network elements and feature servers execute the PacketCable cellular mobility features on behalf of the PacketCable mobile subscriber. The cellular HLR reports mobility events to the delivery function. Note that the PacketCable home network may execute features for subscribers who have roamed onto other visited networks. Call content may not be not present in the PacketCable home network in these roaming scenarios.

It should be noted that the PacketCable accounting architecture is described in [ACCT].

5.5 PacketCable Interface to Law Enforcement Agency Collection Function

The interface from the PacketCable Delivery Function to the law enforcement agency Collection Function is not within the scope of this document.

6 PACKETCABLE REQUIREMENTS

6.1 Interception of Call Related Data

6.1.1 Event Message Requirements

The target's P-CSCF and S-CSCF are provisioned to report call related data to the DF in the form of Intercept Event Messages. Additional network elements may need to be dynamically provisioned to report call related data on a per call basis for call scenarios when the target's S-CSCF is no longer managing the intercepted call. Examples include call redirection and third party call control scenarios. In these cases, the target S-CSCF inserts a P-DCS-LAES header into SIP messaging to dynamically provision other network elements to report call related data. Specific requirements for dynamic provisioning are specified below. The format of the P-DCS-LAES header is specified in [CMSS].

Intercept Event Messages (IEMs) MUST be sent to the DF address that was specified when the tap was provisioned. The requirements on message contents are contained here with details on the format of the actual DIAMETER messages contained in Annex A.

Each IEM MUST have a correlation ID, a timestamp, an identifier for the element type (e.g., P-CSCF, S-CSCF, I-CSCF, BGCF, MGC, Media Server), and an element identifier. The correlation ID is used to correlate messages associated with a given call.

Three IEMs are defined:

1. A "Report" message that contains the entire SIP message. In addition to the SIP message, there is an attribute that indicates:
2. Message sent directly from the target.
3. Message sent directly to the target.

This will help the DF sort out messages that relate to the "punch list" items that require reporting signaling directly to and from the target.

Note that "directly" in the above implies all those SIP messages reported by the target's P-CSCF or S-CSCF that went to or came from the target.

1. A "Correlate" message. This is to help in providing additional correlation information. e.g., when:
   - Initial SIP message has been reported by target's P-CSCF or S-CSCF.
   - A Back-to-Back User Agent (B2BUA) is encountered (i.e., relating to different Call-IDs).
   - There are multiple targets along a signaling path (i.e., where one set of event messages may be associated with multiple targets).
   - There is a mixed network (CMS and SIP CSCFs) and a Billing Correlation ID (BCID) was received in the P-DCS-LAES header.
   - A related call is spawned by an application, with Fixed-Mobile Convergence (FMC) call transfer being a special case.

   Parameters that may be included in the Correlate message include:

   - Tap-ID: this is a logical identifier that is provisioned by the DF; it may have a one-to-one or one-to-many relationship with a case identifier (i.e., the same Tap-ID may apply to multiple Case IDs, for example if the same target is being tapped by multiple LEAs)
   - PacketCable 1.5 BCID
   - New Dialog Parameters (Call-ID, from-tag, to-tag) Note that in some cases the to-tag may not be populated since it may not be known at the time the Correlate message is sent.
   - Previous Dialog Parameters.

Reason for sending this Correlate message include:

- Initial INVITE Message reported.
- B2BUA encountered.
- Additional Target encountered (i.e., for the case in which there are multiple targets along the signaling path).
- Hand-off occurred (e.g., hand-off to/from a wireless network).
- New origination from an Application Server (e.g., a call from the target terminates on an Application Server, which originates a new call as a result).
- BCID received in a P-DCS-LAES header. Note that in the case where some other reason already exists for sending a correlate message, the message will be sent with that reason but with the BCID parameter included.

2. A "Carrier-Info" Message for cases where a call terminates on the PSTN or on an I-CSCF or BGCF. In the case of a PSTN call, the information within the Carrier-Info message MUST include the Carrier_Identification_Code and Trunk_Group_ID.

These messages are summarized in the following tables. Table 3 contains common attributes/parameters that are included in all messages while the three tables following that provide the specific attributes in each of the three messages. Details for Diameter messages are contained in Annex A. Messages reported by network elements to the Delivery Function MUST include the common attributes identified in Table 3 below.

Table 3 - Common Attributes

Attribute	Comment
Correlation-ID	Present in all messages related to a given call. This is the ICID related to the IMS charging indicator.
Timestamp	-
Element-Type	Identifies the type of element sending the message (P-CSCF, S-CSCF, MGC, BE, Conf. Server).
Element-ID	Along with the Element-Type – uniquely identifies the Network Element.

Table 4 lists the attributes used in the "Report" IEM. A Report IEM MUST include the attributes listed in Table 4 below.

Table 4 - Report Message

Attribute	Required or Conditional	Comment
(Common Attributes – see Table 3)	R	
SIP-message	R	The SIP-message attribute MUST include the encapsulated SIP Messages that triggered the Report.
Direction	C	Indicates whether sent "to" or "from" the target. The attribute DIRECTION MUST be included by the target UE's P-CSCF or S-CSCF when they send the Report message. Other elements that send the Report message will exclude the Direction attribute.
Direct-Message	R	The attribute Direct-Message is a Boolean that MUST be set to "True" by the target UE's P-CSCF or S-CSCF if the message is going to or from the UE. If the target UE does not receive the

Attribute	Required or Conditional	Comment
		message or the message has been sent by some network element other than the target's P-CSCF or S-CSCF, then the Direct-Message attribute MUST NOT be set to "False".

Table 5 lists the attributes used in the "Correlate" IEM. A Correleate IEM MUST include the attributes listed in Table 5 below.

Table 5 - Correlate Message

Attribute	Required or Conditional	Comment
(Common Attributes – see Table 3)	R	-
TAP-ID	C	Target identifier. Network elements that MUST populate the TAP-ID attribute include the P-CSCF communicating directly with the target and the target UE's S-CSCF. All other network elements that send the Correlate message will not populate the TAP-ID attribute.
BCID	C	PacketCable 1.5 BCID. The BCID is used to help correlate in mixed network situations. The BCID attribute MUST be populated when a BCID is received in the P-DCS-LAES header (see section 6.3 for details).
Dialog Parameters	R	Dialog Parameters MUST be included in the Correlate message and include the SIP "Call-ID" and "from-tag". The "to-tag" MUST be included in the Dialog Parameters included if available. In the case of a B2BUA, this attribute will contain the old dialog parameters.
New Dialog Parameters	C	This MUST be included by an S-CSCF if an Application Server that is a B2BUA is encountered.
Reason	R	The Reason attribute MUST be included in the Correlate message to indicate the reason the Correlate message was sent.

Table 6 lists the attributes used in the "Carrier-Info" IEM. The Carrier-Info IEM MUST include the attributes listed in Table 6 below.

Table 6 - Carrier-Info Message

Attribute	Required or Conditional	Comment
(Common Attributes – see Table 3)	R	Common attributes per Table 3] MUST be included in the Carrier-Info Message.
Carrier-Identification-Code	C	The Carrier-Identification-Code MUST be included in the Carrier-info Message if available.
Trunk-Group-ID	C	MGCs MUST include the Trunk-Group-ID attribute in the Carrier-info Message.

6.1.2 Procedures and Requirements on Network Elements

6.1.2.1 P-CSCF

P-CSCFs that communicate directly with the target are provisioned to know about the intercept. This provisioning will be updated when changes are made to the target's service (e.g., provisioning of a new destination number) subsequent to initiation of the intercept so that all warranted communications continues to be intercepted.

On an origination attempt, the P-CSCF at the origination side MUST report all SIP messages to and from the subject to the DF. It MUST also send a Correlate event message (EM) with reason Initial INVITE message reported.

P-CSCFs MUST report SUBCRIBEs and NOTIFYs between the network and the target to the DF. Some examples are: NOTIFY for message-waiting indicator, subscriptions to the dialog event package and the corresponding notifications, etc.

If a P-CSCF receives an INVITE with a P-DCS-LAES header from any network element other than a User Equipment (UE), it MUST report the INVITE that is received to the DF.

The P-CSCF MUST remove the P-DCS-LAES header from the INVITE before passing it to the UE.

6.1.2.2 S-CSCF

The target's S-CSCF will be provisioned to know about the intercept. This provisioning will be updated when changes are made to the target's service (e.g., provisioning of a new destination number) subsequent to initiation of the intercept so that all warranted communications continues to be intercepted. The target’s S-CSCF reports IEMs to the DF. The target’s S-CSCF also inserts the P-DCS-LAES header to dynamically provision the call for intercept should other CSCFs or ASs need to report IEMs to the DF. The target's S-CSCF MUST report all SIP messages to and from the target. This includes REGISTER requests and responses.

On an origination attempt, the S-CSCF at the origination side MUST report all SIP messages to and from the subject to the DF. The S-CSCF MUST also send a Correlate message on receipt of the initial INVITE. The correlate message can be used by the DF to link target IDs to call IDs, and to link separate call IDs from third party call control ASs. Note that the DF may receive two Correlate messages with the same reason, since the P-CSCF will also send this message in the case of an origination attempt.

The target's S-CSCF MUST include the P-DCS-LAES header with an initial INVITE. This is true whether the session is an origination or termination attempt.

When the S-CSCF either generates or receives a P-DCS-LAES header (for either an origination or termination attempt), and the INVITE is sent to an Application Server (AS) it has to determine whether or not to include the P-DCS-LAES header when sending the INVITE to the AS. There are two cases to consider:

1. In the case of normal AS processing (the call does not terminate on the AS), the S-CSCF MAY remove the LAES header if the S-CSCF is aware that the AS is not involved intercept. This method shields the AS from the LAES header. The S-CSCF MUST re-insert the LAES header into messages received back from an AS when it had previously removed the header prior to routing to the AS. Otherwise, if the S-CSCF passes the header to the AS, the S-CSCF MUST insure that the header is intact after Application Server processing that may include a B2BUA (i.e., if the AS removes the P-DCS-LAES header, the S-CSCF MUST re-insert the header before forwarding it elsewhere). This allows providing support for Lawful Intercept (LI) without needing to intercept at the AS.
2. In the case where the call terminates on an AS, the S-CSCF MUST include the P-DCS-LAES header when forwarding the request to the AS to dynamically provision the AS for intercept. This is to allow for an AS that (unbeknownst to the S-CSCF) originates a new call as a result of the termination attempt on the AS. Such an AS would have to support LI in order to forward the P-DCS-LAES header on the new call leg and to correlate the new call leg with the termination attempt on the AS (i.e., by sending a Correlate message).

Note that only one P-DCS-LAES header is ever included in a SIP message. If an S-CSCF encounters an existing P-DCS-LAES header on a termination attempt for a new target, the S-CSCF MUST forward the existing P-DCS-LAES header (rather than add or replace the existing header), and it MUST also send a Correlate message to the DF with the new TAP-ID as well as the existing dialog id parameters in order to inform the DF that a single set of event messages are for this new target as well as the previous target.

If the target's S-CSCF receives a REFER from the target, it MUST add the P-DCS-LAES header to the REFER. This dynamically provisions network elements handling the subsequent call triggered by the REFER for intercept.

On a termination attempt to the target, the target's S-CSCF MUST report all SIP messages to the DF for that session, including mid-dialog messages as well as termination attempts and responses that never reach the target (e.g., an attempt that failed after being sent to an Application Server). This is true whether or not the call is re-directed.

An S-CSCF sends a correlate EM under any of the following conditions:

- An S-CSCF MUST send a Correlate EM to the DF if it is the target's S-CSCF and it receives an INVITE that is an origination or termination attempt for the target.
- An S-CSCF MUST send a Correlate EM to the DF if it encounters an Application Server that is a B2BUA.
- An S-CSCF MUST send a Correlate EM to the DF if it receives an INVITE with a P-DCS-LAES header destined for a new target (i.e., multiple targets along the path). REQ13183 In that case it MUST include the existing P-DCS-LAES header with the INVITE rather than forwarding or adding a new one.
- An S-CSCF MUST send a Correlate EM to the DF if it receives a P-DCS-LAES header with a BCID.

Note that an S-CSCF may come across situations where more than one Correlate reason exists. When more that one correlation reason exists, the S-CSCF MUST send Correlate message(s) to describe all known reasons. The S-CSCF MAY either send a single Correlate message with multiple reason AVP's or it MAY send multiple Correlate messages.

Note that if the S-CSCF receives an INVITE without a P-DCS-LAES header and it detects a target (in the case of either an origination or termination attempt) then it MUST send a Correlate IEM with reason "Initial SIP message reported". However, if the S-CSCF receives an INVITE with a P-DCS-LAES header and it detects a target, it MUST send a Correlate IEM with reason "Additional Target Encountered". Note also however, that this can occur in case where an AS has forwarded the P-DCS-LAES header as a result of a spawned call and as such this may not be a new target. It is up to the DF to check the tap-id in the Correlate message and verify whether this is really a new target or an existing one.

If an S-CSCF receives an INVITE with a P-DCS-LAES header it MUST report the INVITE to the DF.

The History-Info in call forwarding cases may not capture all changes of identity. In order to capture all termination attempts and translations along the way, the S-CSCF SHOULD report INVITEs where an AS either changed or could change the identity (Request-URI or P-Asserted-Identity). Therefore, if an S-CSCF either receives an INVITE with a P-DCS-LAES header or is otherwise aware that the INVITE is associated with a target, it MUST report all such INVITEs that may have an change in identity (e.g., possible change in Request-URI, PAID or History-Info) and MAY do so by reporting all SIP INVITE for targets that it forwards on behalf of ASs. The DF is expected to check for identity changes and report those to the LEA as required.

Note that the following guidelines are to be followed in selecting whether the SIP message received or the SIP message forwarded by the S-CSCF should be selected as the one that is reported:

- If an INVITE is received with the P-DCS-LAES header, the S-CSCF MUST report any message that may indicate a change in identity.

- Furthermore, if an INVITE is received with the P-DCS-LAES header the S-CSCF SHOULD report message on the target side of the S-CSCF (e.g., in the case of an origination attempt from the target, the target's S-CSCF would report message received from the target and responses sent to the target).

If the target's S-CSCF receives a 3XX from the target, it will do the following:

- If the target's S-CSCF receives a 3XX from the target and if it takes on the responsibility of sending INVITEs as a result of the 3XX, the S-CSCF MUST report the INVITE and any other SIP messages associated with that session.
- If the target's S-CSCF receives a 3XX from the target and if it takes on the responsibility of sending INVITEs as a result of the 3XX, the S-CSCF MUST also insert the P-DCS-LAES header so that the final destination of the call can be traced.
- If the target's S-CSCF receives a 3XX from the target and the S-CSCF returns the 3XX along the signaling path (i.e., does not handle it), then the S-CSCF MUST include the P-DCS-LAES header in the 3XX response.

If an S-CSCF receives a 3XX with a P-DCS-LAES header and if it sends an INVITE as a result of receiving the 3XX, then it MUST take on the responsibility of reporting all of the SIP messages associated with the INVITE. The S-CSCF MUST also include the P-DCS-LAES header in the INVITE.

If an S-CSCF receives a REFER with a P-DCS-LAES header and if it sends an INVITE as a result of receiving the REFER, then it MUST take on the responsibility of reporting all of the SIP messages associated with the INVITE. The S-CSCF MUST also include the P-DCS-LAES header in the INVITE.

Certain peering relationships between home and visited networks allow a home S-CSCF to route to a visited I-CSCF. In this case the S-CSCFis the last element before the transit network and therefore needs to reports a Carrier-Info message. When a S-CSCF routes a call directly to a visited network I-CSCF, it MUST report a Carrier-Info message to the DF.

6.1.2.3 MGC

If a Media Gateway Controller receives the P-DCS-LAES header, it MUST report the INVITE to the DF and send a Carrier-Info EM.

6.1.2.4 I-CSCF or BGCF

If an I-CSCF or BGCF receives an INVITE with the P-DCS-LAES header it MUST report the INVITE. If an I-CSCF or BGCF is sending the INVITE to another carrier via an IP interconnect, it MUST also send the Carrier-Info EM. The I-CSCF or BGCF MUST remove the P-DCS-LAES header before forwarding the SIP message to a peer network not under the control of the network operator.

6.1.2.5 Application Server

Application servers may need to report IEMs if they generate subsequent calls related to a target service. They also may need to insert the LAES header to dynamically provision calls for intercept. Therefore, requirements are placed on the AS related to the P-DCS-LAES header.

If an Application Server (AS) receives an INVITE with the P-DCS-LAES header and originates a new session or causes a call transfer to occur and the AS is acting as a UAC, it MUST send a Correlate EM. The application server MUST include the P-DCS-LAES header in the INVITE for the new session.

If an Application Server receives a 3XX with a P-DCS-LAES header and if it sends an INVITE as a result of receiving the 3XX, then it MUST take on the responsibility of reporting all of the SIP messages associated with the INVITE. It MUST also include the P-DCS-LAES header in the INVITE.

If Application Server receives a REFER with a P-DCS-LAES header and if it sends an INVITE as a result of receiving the REFER, then it MUST take on the responsibility of reporting all of the SIP messages associated with the INVITE. It MUST also include the P-DCS-LAES header in the INVITE.

6.1.2.6 Home Location Register

The Home Location Register reports visited network registration events from roaming UEs on visited cellular networks. The communications between the HLR and DF is out of scope of this specification.

6.1.3 Dialed Digit Extraction (DDE)

The DF MUST provide mid-call dialed digits that originate from the target subject when authorized. The approach recommended here is for the DF to intercept the content stream at the content IAP. This content stream will be passed to DTMF receivers within the DF which will then extract the dialed digits and pass them to the LEA.

6.2 Interception of Call Content

6.2.1 Invocation of call content intercept: Dynamic Discovery of Intercept Access Point

Discovery of the Content Intercept Access Point (IAP) is done using the discovery mechanism described in [CPD]. In this approach as illustrated in

Figure 2, the Delivery Function sends a Control Point Discovery request message to the destination IP address of the media endpoint. The Content IAP responds with the IP address used to request the TAP-MIB for content tapping along with an identifier that indicates which protocol to use.

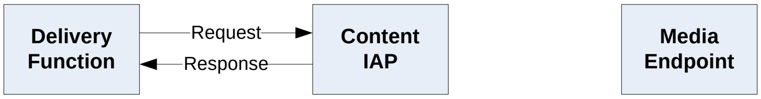

Figure 2 - Content IAP Discovery

Table 7 lists potential Content IAPs for capturing media traffic sent to or from media endpoints in a PacketCable network. The CMTS intercepts all call content sent to and received from a target UE when the target UE is located within the cable network, regardless of the features applied to the media at the UE.

Table 7 - Content IAPs

Media Endpoint	Content IAP	Example Scenario
UE	CMTS	
Media Server including Application Servers that provide media services (e.g., voice-mail).	Aggregation router or switch in front of the media server	Call from another domain that gets forwarded to Voicemail.
Media Gateway	Media Gateway	PSTN call from off-net that gets forwarded off-net.

6.2.1.1 Procedures and Requirements on Network Elements

6.2.1.1.1 Delivery Function

As soon as the Delivery Function obtains the SDP, it SHOULD send a Control Point Discovery (CPD) message per [CPD] towards the media endpoint in order to obtain the IP address and protocol needed to complete the content tap. The DF MUST send a CPD message per [CPD] upon SDP receipt if it does not know the Intercept Access Point (IAP) that can be used to intercept the content. In the case where ICE is used [NFT TR], the DF SHOULD send the message to all of the candidates. In a PacketCable network with a CPE NAT, the DF will typically only get a response from the STUN candidate, since the TURN server will ignore (and not forward) the message and the local candidate will not be reachable.

Alternatively the DF MAY keep a table of addresses for TURN servers. This allows it to avoid sending messages to TURN candidates. It could also check to see if the default value in the "c=" line of the SDP is a

TURN candidate. If it is, then it can try other candidates, otherwise it can send the Control Point Discovery message only to the address in the "c=" lines.

Note that in the case where there are multiple aggregation routers in front of a media services endpoints (e.g., voicemail or media server endpoints), the DF will receive a single CPD response back but may have to have the provisioned IP address of the alternative and install the content tap on both.

For CPD messages that are sent towards peer networks not controlled by the operator, border routers MUST have ACLs installed so that the CPD message will be dropped and an ICMP port unreachable is returned to the DF.

Note that a DF should receive either a CPD Response or an ICMP unreachable for every CPD Request sent out. If it does not receive one of these, this is an indication of either an error or a lost UDP packet. Multiple retries with no response SHOULD be reported as an alarm. Random testing for leakage by the DF is also suggested. This can be done by sending CPD Requests to randomly selected IP addresses.

The Delivery Function MUST support the PKTC-LAES-6 interface per the [CPD] specification as a Requestor.

There are multiple control relationships based on the role of the device. The Requestor (i.e., the DF), MAY send the CPD Request with a wild card value ("0"), so that the DF can determine the role of the control point.

The DF MUST set the "Forward if not supported" flag to 0 when sending a CPD Request towards a client device. This is to avoid the possibility of a CPD message arriving at a client and the user detecting that the call is being tapped. Normally there is only a single Control Point between the Requestor and the Media endpoint so that all CPD Request messages for LI are usually sent with the "Forward if not supported" flag to 0. In the case of an aggregation router/switch that supports content tapping in front of a number of media servers, where one of those media servers is a conference server that also supports content tapping, the Delivery Function (which is the Requestor in this case) MUST make a request with CR ID set to 5 to identify the conference server as the TAP point and with the "Forward if not supported" flag set to 1. The aggregation router switch that also supports the LI CR TYPE but for a different CR ID (CR ID = 3), will forward the message to the conference server so that it can respond.

The Delivery Function (DF) can test to ensure that the destination address of the CPD Request message is in fact a media server endpoint and not a client device by first sending a message with the "Forward if not supported" flag set to 0 and getting a response back from the aggregation device (CR ID = 3), indicating its role as being in front of a media server endpoint.

In some cases, endpoints that are providing media services may not be single homed. In that case, the DF maintains a list of alternate control points, i.e., it sends the CPD Request to the media endpoint and if it gets a response with the address of one Control Point, it looks up the alternatives and provides a content tap on all.

The CMTS MUST support the control point discovery interface specification [CPD] as a Control Point for Lawful Intercept. Components that are used as aggregation routers or switches in front of media servers SHOULD also support this control point discovery interface specification [CPD] as a Control Point.

6.2.1.1.2 Content IAPs

CMTSs, Aggregation routers or switches in front of media servers and Media Gateways MUST support Control Point Discovery for Lawful Intercept.

6.2.2 Call Content Message Requirements

Requirements for the content tapping interface are included in this section. The SNMPv3 TAP-MIB definition is contained in Annex B.

Call content message requirements include:

- The ability to set up the interface with the delivery function: destination address (where to send the call content), format, transport, call content identifier.

- The ability to specify a duration or expiry time for the intercept. If this time is exceeded, all content tapping will stop and all state associated with the intercept will be deleted by the content IAP.
- The ability to set up a layer 3 IP protocol classifier (filter specification) to describe the packets that need to be replicated, encapsulated and transported.
- The ability to delete the intercept.

These requirements are described by means of the API description for Delivery Function Setup, Intercept Request and Intercept Stop in the following subsections.

6.2.2.1 Delivery Function Setup

The Delivery Function Setup is an initial set to prepare for content intercept delivery to the DF, but does not provision a specific target for intercept.

```
Result   <-- SetupDF(Content-ID
                                        ,ExpiryTime
                                        ,DFaddress
                                        ,DSCP
                                        ,Transport)
```

"Result" indicates pass or fail.

Content-ID: The content-ID is a 32 bit number which is added to the content sent to the DF in order to identify it. It is up the DF to insure that this value is uniquely defined in order to be able to identify the received content streams with a particular tap authorization.

ExpiryTime: Expiry time for the intercept. After the time specified has past, the intercept will cease to exist. Because the DF may be the only device that is aware of intercepts, this mechanism ensures that intercepts do not remain in the case where the DF disappears or loses its memory.

DFaddress: IP address and port of the Delivery Function where the Content IAP should send the replicated packets.

DSCP: Diffserv Code Point (DSCP) value for packets in the content stream; the default value is 0x0b100010 hex corresponding to AF41 (Assured Forwarding) Per Hop Behavior.

Transport: Transport and packet encapsulation format for replicate packets sent to the DF. The only format presently supported is the PacketCable call content format.

6.2.2.2 Intercept Request

The Intercept Request command is used to set up an intercept for a specific stream defined by a filter specification.

```
Result  <-- InterceptRequest(Content-ID
                ,FilterSpec)
```

FilterSpec: is used to specify packets to be copied. The "FilterSpec" for layer3/4 IP includes the following parameters:

- Destination IP prefix (address and number of bits)
- Destination port range
- Source IP prefix (address and number of bits)
- Source port range
- Protocol ID

Any of the above parameters can be wild-carded as long as there is sufficient information to specify the stream. For an RTP media stream, the destination IP address and port as well as protocol ID will normally be specified while one or more of the source parameters may not.

6.2.2.3 Intercept Stop

The Intercept Stop function removes replication for the filter specification(s) for the content ID specified.

```
        Result <---- InterceptStop(Content-ID)
```

"Result" is an indication of pass or fail.

6.2.3 Invocation of call content intercept: TAP MIB

The procedures for DFs and Content IAPs are described in the following sections.

6.2.3.1 Delivery Function

As soon as the Delivery Function obtains the SDP and uses Control Point Discovery to obtain the IP address and protocol to do the content tap, it MUST use address of the media endpoint in order to set up the filter specification for content tapping per the TAP MIB described in Annex B with the following attributes:

- Receive port from the "m=" line,
- Receive IP address from the "c=" line, and
- UDP for the Protocol ID.

The DF MUST continue to check the MIB and insure that the network element acting as the Content IAP has not re-booted (which will cause the MIB to disappear). If that happens, the DF MUST re-install the TAP.

6.2.3.2 Content IAPs

CMTSs, aggregation routers or switches in front of media servers and Media Gateways acting as Content IAPs MUST support the TAP-MIB Annex B for Lawful Intercept. Note that intercept state SHOULD not withstand re-boots of Content IAP network elements. This, as well as a timeout within the MIB itself, are there on purpose to insure that intercept state does not remain inadvertently as a result of failures such that content taps end up staying beyond their authorization limits.

6.2.4 Delivery of Call Content to Delivery Function

The format of the content delivery between Content IAPs and the DF MUST adhere to the following format:

Table X - Payload of Call Content Connection Datagrams

CCCID (4 bytes)
Intercepted Information (arbitrary length)

Intercepted RTP information will be of the following format:

Table Y - Intercepted Information

Original IP Header (20 bytes)
Original UDP Header (8 bytes)
Original RTP Header (variable length, 12-72 bytes)
Original Payload (arbitrary length)

Note that protocols other than RTP may be intercepted, such as for T.38 fax relay.

6.3 Interoperability with Networks Supporting NCS Clients

PacketCable networks supporting NCS clients via the CMS will use the Event Messages Appendix A of [EM] and will continue performing content tapping over the [TGCP] and Dynamic Quality of Server (DQOS) interfaces [DQOS]. Requirements placed on CMS network elements for interoperability with PacketCable CSCF elements are defined in the CMS specifications. Requirements apply to the CMS, which is upgraded to support interoperability with PacketCable CSCF networks. The MGC and MG should be upgraded to support PacketCable intercept provisioning interfaces defined herein, or they may use existing TGCP and DQOS interfaces. In order to support mixed CMS and CSCF networks the following procedures apply for performing Lawful Intercept. This specification assumes that the CMTS within the PacketCable network is upgraded to support the PacketCable TAP MIB specified in Annex B of this document.

6.3.1 S-CSCF

The target's S-CSCF MUST include the "Laes-content" parameters in the P-DCS-LAES header if the content parameters have been provisioned for that intercept. The "cccid" and "bcid" parameters MUST NOT be populated. However, if the S-CSCF receives a P-DCS-LAES header with "cccid" and/or "bcid" parameters it MUST of course leave those parameter in the header if it passes the header on.

If the S-CSCF receive a P-DCS-LAES header with the "bcid" parameter, it MUST send a Correlate EM with the BCID value along with the dialog parameters.

6.3.2 DF

The DF needs to be able to map a combination of messages defined in [EM] and the present specification to the set of messages delivered to the law agency.

The DF can determine whether a PacketCable component is performing content tapping by via TGCP and DQOS or methods defined in the present document by whether it receives a non-zero "cccid" value in a Media_Report message. If the DF cannot determine if a component is tapping via TGCP and DQOS, it MUST perform the content tapping using the procedures described in Section 6.2 of this document.

6.4 Security Requirements

Given the sensitive nature of lawful intercept - both from the standpoint of the need to protect the intercept data, as well as conceal the identities of the intercept targets, the LI solution MUST have the ability to provide authentication, integrity checking and encryption on all interfaces.

The lawful intercept architecture is illustrated in Figure 1. These LAES interfaces share common protocol stacks as the balance of the PacketCable architecture, except for the use of SNMPv3 on PKT-LAES-4. PKT-LAES-2 carries event messages via DIAMETER. PKT-LAES-3 carries provisioning of call data via SIP. PKT-LAES-4 provisions content intercept access points via SNMPv3. PKT-LAES-5 carries intercepted call content over UDP. PKT-LAES-6 is used to discover content intercept access points. The Zb reference point defined in the PacketCable Security Technical Report [SEC TR] is applied to all these interfaces to provide security as illustrated in Table 8. The Zb reference point supports IPSec and TLS based security mechanisms as described in the Security Technical Report [SEC TR]. TLS does not apply to all intercept interfaces as indicated in the following table.

Table 8 - Zb interface applied to intercept

LAES Interface	Similar PacketCable Interface	Protocol	Zb Security Layer Options
PKT-LAES-2	rf	Diameter	IPSec, TLS
PKT-LAES-3	Mw	SIP	IPSec, TLS
PKT-LAES-4	none	SNMPv3	IPSec
PKT-LAES-5	Media	UDP	IPSec
PKT-LAES-6	PKTC-CPD-1	Control Point Discovery	IPSec

In addition to the Zb security mechanisms, the PKT-LAES-4 interface needs additional security mechanisms to protect access to the intercept provisioning data stored on the TAP MIB. SNMPv3 also supports transport security that can be applied in addition to, or in place of, IPSec as defined in the Zb reference point. The balance of this section describes the additional security requirements placed on this SNMPv3 interface.

SNMPv3 provides an extended User Security Model (USM), which provides data integrity, data origin authentication, protection against disclosure of the message payload, and protection against message delay or replay.

When the SNMP MIB Annex B is used for LI messages, the USM MUST be used. Authentication MUST be enabled. The MD5 authentication algorithm (usmHMACMD5AuthProtocol) MUST be supported. The SHA1 authentication algorithm (usmHMACSHAAuthProtocol) SHOULD be supported.

Privacy MUST be enabled. Pre-shared keys MUST be supported as a minimum (shared between the content IAPs and the DF). The SNMPv3_AES Transform ID MUST be enabled if AES is supported. Otherwise, the SNMPv3_DES Transform ID MUST be enabled if AES is not supported. The SNMPv3_NULL Transform IDs MUST be supported. The DES encryption transform for SNMPv3 is specified in [RFC 3414]. The AES encryption transform for SNMPv3 is specified in [RFC 3826].

The content IAP MUST also support the ability to protect the MIBs from disclosure or control by unauthorized USM users [RFC 3414] by means of view access control [RFC 3415].

HEADING: USM Requirements

The usmUserTable MUST be configured with the following entries:

usmUserEngineID - the SNMP local engine id

usmUserName - LAES-TAP-Prov--<unique network element identifier>

usmUserSecurityName - LAES-TAP-Prov--<unique network element identifier>

usmUserCloneFrom – 0.0

usmUserAuthProtocol - usmHMACMD5AuthProtocol or

usmHMACSHAAuthProtocol

usmUserAuthKeyChange

usmUserOwnAuthKeyChange

usmUserPrivProtocol – usmDESPrivProtocol

UsmUserPrivKeyChange

UsmUserOwnPrivKeyChange

usmUserPublic

usmUserStorageType - permanent

usmUserStatus – active

New users MAY be created by cloning as defined in SNMPv3. This MAY be done through the config file, or later through SNMP Set operations.

6.4.1 VACM Requirements

The following VACM entries MUST be defined for PacketCable Electronic Surveillance. Other table entries MAY be implemented at vendor or operator discretion.

VACM views MUST be defined as described below.

HEADING2: VacmSecurityToGroup Table

The following configuration of the vacmSecurityToGroup table provides a read/write/create view.

vacmSecurityModel - USM

vacmSecurityName - '' LAES-TAP-Prov --<unique network element identifier>

vacmGroupName - 'LAES-TAP-ProvFullAccess'

vacmSecurityToGroupStorageType - permanent

vacmSecurityToGroupStatus – active

HEADING2: vacmAccessTable

The vacmAccessTable MUST be configured with the following entries. Other table entries MAY be implemented at vendor or operator discretion.

This configuration allows for read/write access of all Electronic Surveillance modules in the content IAPs and notifications as defined in the PacketCable MIB modules [MIB-CLABDEF] (see NotifyViewName below):

vacmGroupName – LAES-TAP-ProvFullAccess

vacmAccessContextPrefix

vacmAccessSecurityModel - USM

vacmAccessSecurityLevel – authPriv

vacmAccessContextMatch – exact

vacmAccessReadViewName – ReadOnlyView

vacmAccessWriteViewName – FullAccessView

vacmAccess NotifyViewName – NotifyView

vacmAccessStorageType – permanent

vacmAccessStatus - active

HEADING2: MIB View Requirements

The FullAccessView MUST consist of the MIB2 system group, the IFMIB, and all PacketCable defined MIB modules. It MAY include vendor defined MIBs, VACM, USM, and Notifications MIB. The following lists the required OIDs.

1.3.6.1.4.1.4491.2.2 /* PacketCable Project MIB tree */

The ReadOnlyView MUST consist of the entire MIB tree contained in the electronic surveillance intercept access point for call content, including PacketCable defined MIB modules, and vendor defined MIB modules for PacketCable Electronic Surveillance.

1.3.6.1 /* Full Internet MIB Tree*/

The NotifyView MUST consist of the MTA MIB tree, MIB-2 System MIB tree and the snmpTrapOID MIB. It MAY include vendor defined MIB modules.

1.3.6.1.4.1.4491.2.2.1 /*PacketCable Project MIB tree*/

1.3.6.1.2.1.1 /* MIB-2 system mib tree */

1.3.6.1.6.3.1.1.4.1.0 /* snmpTrapOID mib*/

7 PACKETCABLE SAMPLE ELECTRONIC SURVEILLANCE CALL FLOWS (INFORMATIVE)

7.1 Origination from Target

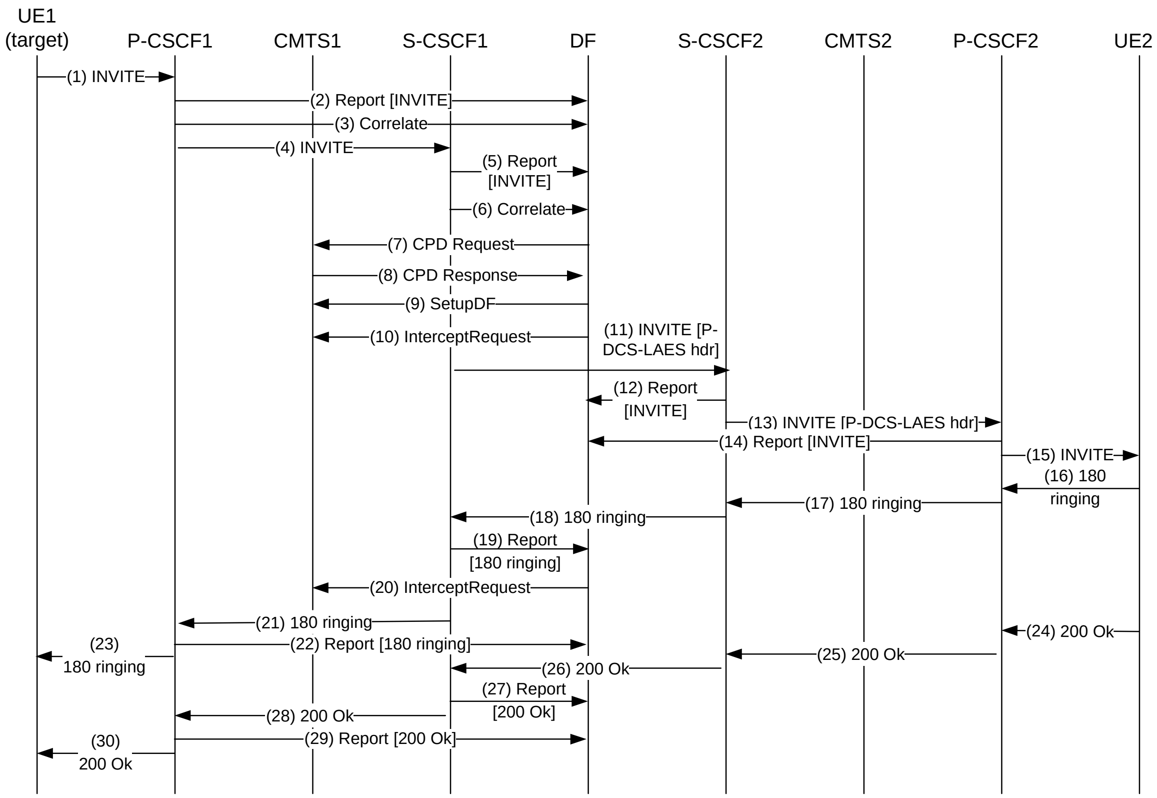

Figure 3 - Origination From Target

1-6: SIP UE UE1 (the target) sends an INVITE to P-CSCF1 which forwards it to (the target's) S-CSCF1. Both P-CSCF1 and S-CSCF1 report the INVITE as well as Correlate. The Correlate EM includes the TAP-ID and dialog parameters with Reason: "initial SIP message reported".

7-8: Control Point Discovery (CPD) request sent towards UE1 (based on the IP address of the "c=" line of the SDP in the INVITE. CMTS1 responds with the IP address of its TAP-MIB.

9-10: the DF does a SetupDF and InterceptRequest for the TAP-MIB. The classifier supplied in the InterceptRequest is based on the IP address and port supplied in the SDP within the INVITE.

11: S-CSCF1 adds the P-DCS-LAES header and forwards the INVITE to S-CSCF2. Note that this can be done in parallel with 4-9.

Note: S-CSCF1 to S-CSCF2 may go through an I-CSCF (not shown here).

12: S-CSCF2 reports the INVITE.

13-14: S-CSCF2 passes the INVITE to P-CSCF2 which reports the INVITE.

15: P-CSCF2 then strips the header before passing the INVITE to the SIP UE UE2.

16-18: "180 ringing" passed back.

19: S-CSCF1 reports the "180 ringing".

20: InterceptRequest for the flow specified in the SDP of the 180 ringing.

21-23: "180 ringing" passed back and reported by P-CSCF1.

24-30: "200 Ok" passed back and reported by S-CSCF1 and P-CSCF1.

7.2 Termination on the Target

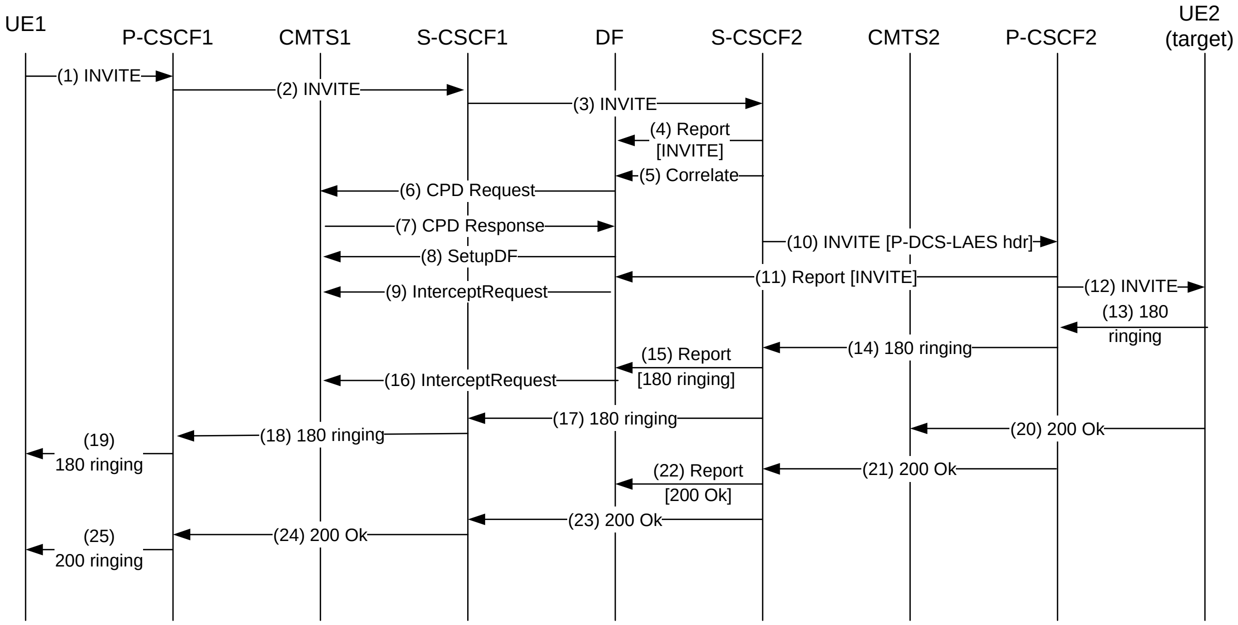

Figure 4 - Termination on the Target

1-3: INVITE arrives at (target's) S-CSCF2.

Note that S-CSCF1 to S-CSCF2 may route via an I-CSCF (not shown).

4-5: S-CSCF2 reports the INVITE and sends a Correlate message to the DF.

6-9: The DF sends a CPD message to find the address of the TAP MIB. In then sets up the interface to the DF ("SetupDF") and initiates an "InterceptRequest" to the TAP-MIB based on the classifier in the SDP of the INVITE.

10: S-CSCF2 adds the P-DCS-LAES header before forwarding on the INVITE. Note that this could be done in parallel with 4-9.

11-12: P-CSCF2 reports the INVITE and strips the P-DCS-LAES header before sending the invite to the UE UE2.

13-14: 180 ringing.

15: 180 ringing reported.

16: InterceptRequest for the flow specified in the SDP of the 180 ringing.

17-19: 180 ringing passed to UE1.

20-21: 200 Ok.

22: 200 Ok reported.

23-25: 200 Ok passed to UE1.

Note that an alternative to this call flow is for the DF to do the content tap on CMTS2. In that case, it would have to wait to receive the report with the SDP in the "180 ringing" from UE2 before initiating the content tap.

7.3 Call to the PSTN

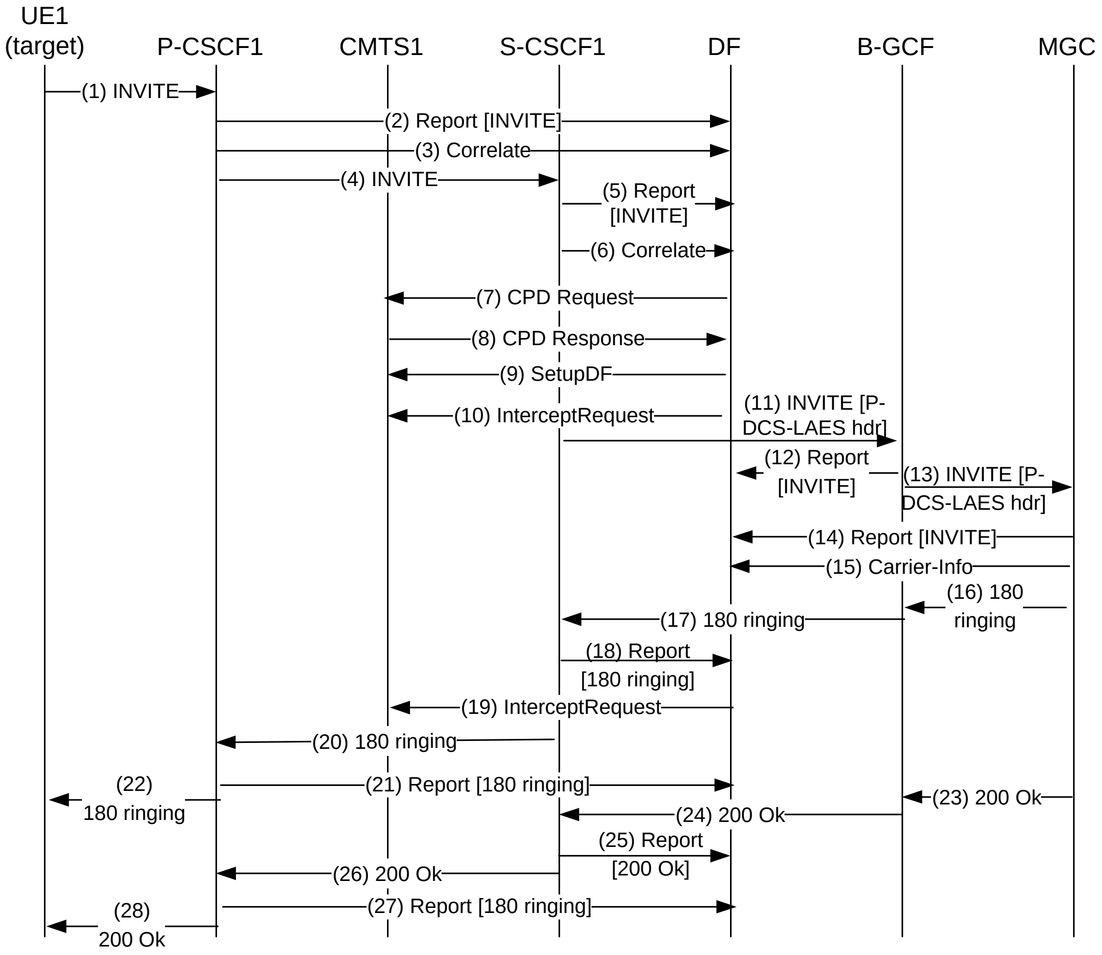

Figure 5 - Call to the PSTN

1-6: INVITE from the target UE is sent to P-CSCF1 and the S-CSCF1. Both report the INVITE to the DF as well as a Correlate message with Reason: "initial SIP message reported".

7-10: The DF sends a CPD message to find the address of the TAP MIB. In then sets up the interface to the DF ("SetupDF") and initiates an "InterceptRequest" to the TAP-MIB based on the classifier in the SDP of the INVITE.

11: S-CSCF1 adds the P-DCS-LAES header before forwarding on the INVITE. Note that this could be done in parallel with 4-9.

12-15: BGCF forwards the INVITE to the MGC. Both BGCF and MGC report the INVITE. The MGC also sends "Carrier-Info".

16-17: 180 ringing.

18: S-CSCF1 reports "180 ringing".

19: InterceptRequest for the flow specified in the SDP of the 180 ringing.

20-22: "180 ringing" passed back and reported by the P-CSCF1.

23-28: "200 Ok" passed back and is reported by S-CSCF1 and P-CSCF1.

Note that the above assumes that the MGC is within the same network. If the MGC is provided by an inter-connect carrier, then it is up to the BGCF to send the "Carrier-Info". It will also strip of the P-DCS-LAES header before forwarding the INVITE. MGC (in the other carrier network) will not send any reports to the DF.

7.4 Call from other domain Re-directed to Voice-mail

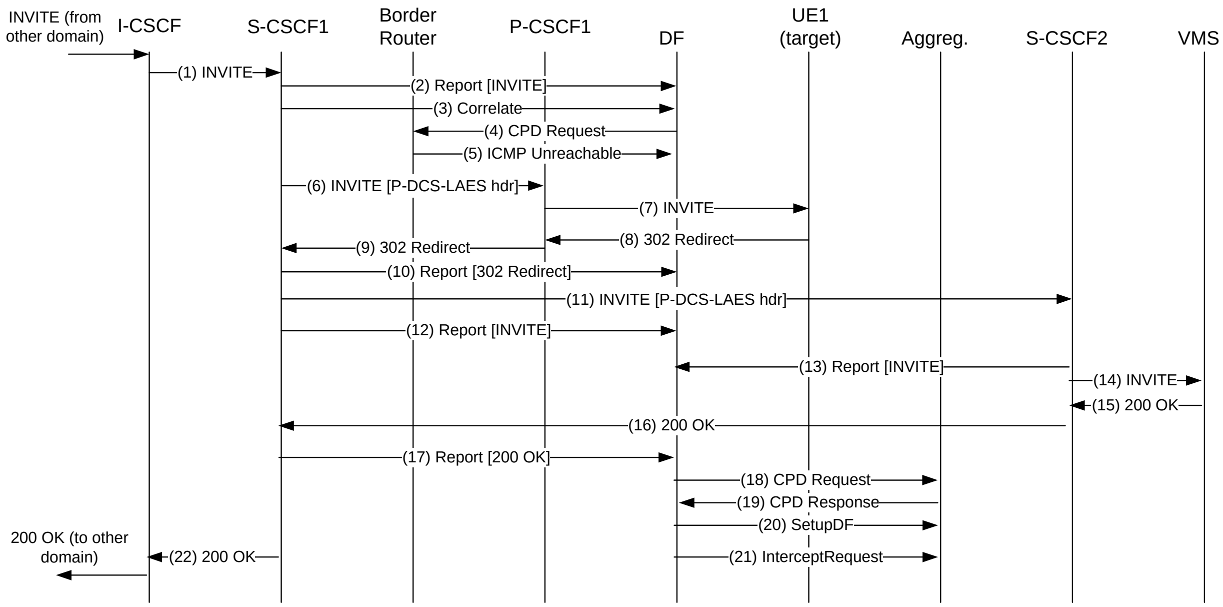

Figure 6 - Call from other domain Re-directed to Voicemail

1-3: INVITE from other domain arrives at targets S-CSCF (S-CSCF1) which reports the INVITE and sends a Correlate message.

4-5: CPD Request is rejected by the inter-domain router (ICMP port unreachable).

6: S-CSCF1 adds the P-DCS-LAES header and forwards the INVITE to P-CSCF1.

7: P-CSCF1 strips off the P-DCS-LAES header and forwards the INVITE to the target (UE1).

8-9: UE1 does a re-direct (302) which is passed back to S-CSCF1.

10: 302 reported.

11-12: S-CSCF1 sends an INVITE to voicemail system (VMS) as a result of receiving the 302 (P-DCS-LAES header included). It reports the INVITE.

Note: an Application Server rather than S-CSCF may recurse the INVITEs as a result of receiving the 302. If that is the case, then it is up to the Application Server to report the SIP messages associated with new INVITE (and include the P-DCS-LAES header in that INVITE).

13: INVITE is reported.

13-14: INVITE forwarded to voice-mail (VMS) and reported.

15-16: 200 OK returned.

17: S-CSCF1 reports the 200 OK.

18-19: DF sends CPD Request based on IP address in "c=" line of SDP in 200 OK. CPD response returned from aggregation router in front of VMS.

20-21: SetupDF and Intercept Request for both flows.

22: 200 OK returned to UE1 (note that this can be done in parallel with 17-21).

Annex A Diameter Event Messages (Normative)

PacketCable call data information MUST be sent from PacketCable network elements to the Delivery Function using Diameter accounting messages from the Diameter Base Protocol specified in [RFC 3588] and [TS 32.299]. Diameter accounting is a client/server protocol that uses the messages:

- Accounting Request (ACR). This message is used to send surveillance call data messages from the network elements to the Delivery Function.
- Accounting Answer (ACA). This message is used to acknowledge an Accounting Request.

All data transported in Diameter messages is in the form of an Attribute-Value Pair (AVP). This section describes the use of AVPs from the Diameter Base Protocol and additional AVPs defined in this specification for surveillance information. In addition AVPs defined in the 3GPP specification [TS 32.299] are also used as described in Section A.3.3. Conditional AVPs SHOULD be omitted if the parameter is not available to prevent empty objects.

The Diameter client resides in the PacketCable network element and the Diameter server resides in the Delivery Function. The Diameter client MUST implement the accounting state machine “CLIENT, ACCOUNTING” described in section 8.2 of [RFC 3588]. The Diameter server MUST implement the accounting state machine "SERVER, STATELESS ACCOUNTING" described in section 8.2 of [RFC 3588].

Diameter messages MUST be transported over TCP as specified in [RFC 3588].

The following symbols from [RFC 3588] are used to indicate the presence of AVPs in the tables below:

- <AVP> indicates a mandatory AVP with a fixed position in the message.
- {AVP} indicates a mandatory AVP in the message.
- [AVP] indicates an optional AVP in the message.
- *AVP indicates that multiple occurrences of an AVP are possible.

A.1 Accounting-Request Message

Table 9 shows the structure of a Diameter Accounting-Request message. Network elements reporting events to the Delivery Function MUST use the message format specified in Table 9 as shown below.

Table 9 - Accounting-Request Message

Diameter base protocol AVPs	
AVP	**Used in ACR**
<Diameter-Header:271,REQ,PXY>	Yes
<Session-Id> -- Diameter Session Id	Yes
{Origin-Host}	Yes
{Origin-Realm}	Yes
{Destination-Realm}	Yes
{Accounting-Record-Type}	Yes
{Accounting-Record-Number}	Yes
[Acct-Application-Id]	Yes
[Vendor-Specific-Application-Id]	No
[Vendor-Id]	No
{ Auth-Application-Id }	No

Diameter base protocol AVPs	
AVP	**Used in ACR**
{ Acct-Application-Id }	No
[User-Name]	No
[Accounting-Sub-Session-Id]	No
[Accounting-RADIUS-Session-Id]	No
[Acct-Multi-Session-Id]	No
[Acct-Interim-Interval]	No
[Accounting-Realtime-Required]	No
[Origin-State-Id]	Yes
[Event-Timestamp]	Yes
*[Proxy-Info]	No
{ Proxy-Host }	No
{ Proxy-State }	No
*[Route-Record]	No
*[AVP]	No
PacketCable Diameter Surveillance AVPs	
{Event-Message-Type}	Yes
{Element-Type}	Yes
{Element_ID}	Yes
[SIP-message]	Yes
[Direction]	Yes
[Direct-Message]	Yes
[Tap-Id]	Yes
[BCID]	Yes
[Dialog-Id]	Yes
[New-Dialog-Id]	Yes
[Correlate-Reason]	Yes
{LI-Information}	Yes
3GPP Diameter Accounting AVPs	
{IMS-Charging-Identifier}	Yes
[Inter-Operator-Identifier]	Yes
[Originating-IOI]	No
[Terminating-IOI]	Yes
[Trunk-Group-Id]	Yes
[Incoming-Trunk-Group-Id]	No
[Outgoing-Trunk-Group-Id]	Yes

The Event-Timestamp AVP MUST be used for the PacketCable Surveillance application.

A.2 Accounting-Answer Message

Table 10 shows the structure of a Diameter Accounting-Answer message. The Delivery Function MUST use the message format specified in Table 10 to acknowledge an event received from a PacketCable network element.

Table 10 - Accounting-Answer Message

Diameter base protocol AVPs	
AVP	**Used in ACA**
<Diameter-Header:271,PXY>	Yes
<Session-Id>	Yes
{Result-Code}	Yes
{Origin-Host}	Yes
{Origin-Realm}	Yes
{Accounting-Record-Type}	Yes
{Accounting-Record-Number}	Yes
[Acct-Application-Id]	Yes
[Vendor-Specific-Application-Id]	No
[Vendor-Id]	No
{ Auth-Application-Id }	No
{ Acct-Application-Id }	No
[User-Name]	No
[Accounting-Sub-Session-Id]	No
[Accounting-RADIUS-Session-Id]	No
[Acct-Multi-Session-Id]	No
[Error-Reporting-Host]	No
[Acct-Interim-Interval]	No
[Accounting-Realtime-Required]	No
[Origin-State-Id]	Yes
[Event-Timestamp]	Yes
*[Proxy-Info]	No
{ Proxy-Host }	No
{ Proxy-State }	No
*[AVP]	No

A.3 Diameter AVPs

The AVP types, i.e., Enumerated, specified in the tables below are defined in [RFC 3588].

A.3.1 Diameter Base AVPs

AVPs defined in the Diameter Base Protocol are not described further in the document except for the AVPs listed below where the PacketCable Surveillance application requires specific values. The IETF DIAMETER AVPs contained within the ACR message defined in Table 9and the ACA message defined in Table 10 MUST conform to the format described in Table 11 below.

Table 11 - Use of IETF Diameter AVPs

AVP Name	AVP Code	Used in		Value Type	AVP Flag rules				
		ACR	ACA		Must	May	Should not	Must not	May Encr.
Accounting-Record-Number	485	M	M	Unsigned32	M	P	-	V	Y
Accounting-Record-Type	480	M	M	Enumerated	M	P	-	V	Y
Destination-Host	293	Oc	Oc	DiamIdent	M	P	-	V	N
Destination-Realm	283	M	-	DiamIdent	M	P	-	V	N
Event-Timestamp	55	M	M	Time	M	P	-	V	N
Origin-Host	264	M	M	DiamIdent	M	P	-	V	N
Origin-Realm	296	M	M	DiamIdent	M	P	-	V	N
Origin-State-Id	278	OC	OC	Unsigned32	M	P	-	V	N
Result-Code	268	-	M	Unsigned32	M	P	-	V	N
Session-Id	263	M	M	UTF8String	M	P	-	V	Y
Acct-Application-Id	259	M	M	Unsigned 32	M	P		V	Y

A.3.1.1 Acct-Application-Id AVP

The Acct-Application-Id AVP (AVP code 259), as part of the Vendor-Specific-Application-Id grouped AVP, MUST contain the value of 3.

A.3.1.2 Accounting-Record-Type AVP

The Accounting-Record-Type AVP (AVP Code 480) MUST be of type Enumerated and contain the type of accounting record being sent. This AVP MUST be set to Event (1) for PacketCable Surveillance messages.

A.3.2 PacketCable Surveillance AVPs

Additional AVPs are defined for the PacketCable Surveillance application. The information is summarized in Table 12.

The 'V' in the AVP Flag Rules column of Table 12 indicates that the Vendor-Id field is present in the AVP per [RFC 3588]. The 'V' bit, known as the Vendor-Specific bit, indicates whether the optional Vendor-ID field is present in the AVP header. When set the AVP Code belongs to the specific vendor code address space.

The PacketCable Diameter Surveillance AVPs contained within the ACR message defined in Table 9 and the ACA message defined in Table 10 MUST conform to the format described in Table 12 below.

Table 12 - PacketCable Diameter Surveillance AVPs

AVP Name	AVP Code	Clause Defined	Value Type	AVP Flag rules				
				Must	May	Should Not	Must Not	May Encr.
Event-Message-Type	214	A.3.2.6	Enumerated	V,M	P			N
Element-Type	213	A.3.2.5	Enumerated	V,M	P			N
Element-ID	212	A.3.2.11	UTF8String	V,M	P			N
Tap-Id	231	A.3.2.10	UTF8String	V,M	P			N
SIP-Message	229	A.3.2.9	OctetString	V,M	P			N
Direct-Message	211	A.3.2.4	Enumerated	V,M	P			N
Direction	210	A.3.2.3	Enumerated	V,M	P			N
Dialog-Id	203	A.3.2.2	UTF8String	V,M	P			N
New-Dialog-Id	219	A.3.2.8	UTF8String	V,M	P			N
Correlate-Reason	202	A.3.2.1	Enumerated	V,M	P			N

AVP Name	AVP Code	Clause Defined	Value Type	AVP Flag rules				
				Must	May	Should Not	Must Not	May Encr.
BCID	200	A.3.2.12	UTF8String	V,M	P			N
LI-Information	218	A.3.2.7	Grouped	V,M	P			N

A.3.2.1 Correlate-Reason AVP

The Correlate-Reason AVP (AVP code 202) MUST be of type Enumerated and indicate the reason that the Correlate message was sent. This AVP MUST have one of the following values:

0 – Unknown

1 – B2BUA

2 – Initial SIP Message sent by target's S-CSCF

3 – Additional target encountered

4 – Hand-off Occurred

5 – Origination from an Application Server as a result of a termination on that Application Server.

6 – BCID received in the P-DCS-LAES header.

A.3.2.2 Dialog-Id AVP

The Dialog-Id AVP (AVP code 203) MUST be of type UTF8String and contain the SIP dialog identifier in the form: Call-ID=x;FTag=y;TTag=z, where x is the value of the SIP Call-ID header, y is the contents of the From header tag, and z is the contents of the To header tag. If the To header tag value is not present in the SIP message then TTag field MUST not be present in the AVP.

A.3.2.3 Direction AVP

The Direction AVP (AVP code 210) MUST be of type UTF8String and indicate whether the reported message was sent "to" or "from" the intercept target. This AVP MUST have one of the following values:

0 – Undefined

1 – To target

2 – From target

A.3.2.4 Direct-Message AVP

The Direct-Message AVP (AVP code 211) MUST of type Enumerated and indicate if the reported message is exchanged directly between the IAP and the intercept target. This AVP MUST have one of the following values:

0 – False

1 – True

A.3.2.5 Element-Type AVP

The Element-Type AVP (AVP code 213) MUST of type Enumerated and identify the type of node where the intercept message was generated. This AVP MUST have one of the following values:

0 – S-CSCF

1 – P-CSCF

2 – I-CSCF

3 – MRFC

4 – MGCF

5 – BGCF

6 – AS

7 – UE

A.3.2.6 Event-Message-Type AVP

The Event-Message-Type AVP (AVP code 214) MUST be of type Enumerated and identify the type of surveillance message. This AVP MUST have one of the following values:

0 – Report

1 – Correlate

2 – Carrier-Info

A.3.2.7 LI-Information AVP

The LI-Information AVP (AVP code 218) MUST be of type Grouped, and hold all the other surveillance AVPs listed in Table 12. It MUST have the following ABNF grammar:

```
<LI-Information> :: =          < AVP Header: 218>
    { Event-Message-Type }
    { Element-Type }
    { Element_ID }
    [ Tap-Id ]
    [ SIP-Message ]
    [ Direct-Message ]
    [ Direction ]
    [ Dialog-Id ]
    [ New-Dialog-Id ]
    [ Correlate-Reason ]
    [ BCID]
```

A.3.2.8 New-Dialog-Id AVP

The New-Dialog-Id AVP (AVP code 219) MUST be of type UTF8String and contain the SIP dialog identifier in the form: Call-ID=x;FTag=y;TTag=z, where x is the value of the SIP Call-ID header, y is the contents of the From header tag, and z is the contents of the To header tag. If the To header tag value is not present in the SIP message then TTag field MUST not be present in the AVP.

A.3.2.9 SIP-Message AVP

The SIP-Message AVP (AVP code 229) MUST be of type OctetString and hold the entire SIP message or messages received by the IAP.

A.3.2.10 Tap-Id AVP

The Tap-Id AVP (AVP code 231) MUST be of type UTF8String and hold the Tap Identifier as provisioned by the DF.

A.3.2.11 Element ID AVP

The Element-Id AVP (AVP code 212) MUST be of type UTF8String and identify the PacketCable IAP sending an intercept message to the DF.

A.3.2.12 BCID

The BCID AVP (AVP code 200) MUST be of type UTF8String and hold the PacketCable 1.5 Billing Correlation ID as generated for a SIP session. This value is copied from the bcid field in the P-DCS-LAES header.

A.3.3 3GPP Accounting AVPs

These AVPs that are used for PacketCable Surveillance are defined in the 3GPP specification: Charging management; Diameter charging application [TS 32.299]. They are included here for convenience. The 3GPP Diameter Accounting AVPs contained within the ACR message defined in Table 9 and the ACA message defined in Table 10 MUST conform to the format described in Table 13 below.

Table 13 - 3GPP Diameter Accounting AVPs

AVP Name	AVP Code	Clause Defined	Value Type	AVP Flag rules				
				Must	May	Should not	Must not	May Encr.
IMS-Charging-Identifier	841	A.3.4.1	UTF8String	V,M	P			N
Inter-Operator-Identifier	838	A.3.4.2	Grouped	V,M	P			N
Outgoing-Trunk-Group-Id	853	A.3.4.3	UTF8String	V,M	P			N
Terminating-IOI	840	A.3.4.4	UTF8String	V,M	P			N
Trunk-Group-Id	841	A.3.4.5	Grouped	V,M	P			N

A.3.3.1 IMS-Charging-Identifier (ICID) AVP

The IMS-Charging-Identifier AVP (AVP code 841) MUST be of type UTF8String, and hold the IMS Charging Identifier (ICID) as generated by an IMS node for a SIP session.

A.3.3.2 Inter-Operator-Identifier AVP

The Inter-Operator-Identifier AVP (AVP code 838) MUST be of type Grouped and hold the identification of the network neighbors (originating and terminating) as exchanged via SIP signaling and described in [RFC 3455]. It MUST be of the following ABNF grammar:

```
<Inter-Operator-Identifier>::=< AVP Header: 838 >
    [Originating-IOI]   -- not used
    [Terminating-IOI]
```

The Inter-Operator-Identifier AVP contains the CIC code present in the Carrier-info message.

A.3.3.3 Outgoing-Trunk-Group-ID AVP

The Outgoing-Trunk-Group-ID AVP (AVP code 853) MUST be of type UTF8String and identify the outgoing PSTN leg.

A.3.3.4 Terminating-IOI AVP

The Terminating-IOI AVP (AVP code 840) MUST be of type UTF8String (alphanumeric string) and hold the Inter Operator Identifier for the originating network as generated by the S-CSCF in the home network of the terminating end user as described in [RFC 3455].

A.3.3.5 Trunk-Group-ID AVP

The Trunk-Group-ID AVP (AVP code 851) MUST be of type Grouped and identify the incoming and outgoing PSTN legs. It MUST have the following ABNF grammar:

```
<Trunk-Group-ID>::=<AVP Header: 851>
```

[Incoming-Trunk-Group-ID] -- not used

[Outgoing-Trunk-Group-ID]

Annex B TAP-MIB (Normative)

The TAP-MIB MUST be decomposed into two components:

- The Intercept MIB specified in Section B.1: a generic stream table that contains fields that are common to all intercept types.
- The IP TAP MIB specified in Section B.2: the specifics for tapping content at layer 3 (IP).

Note that the term Mediation Device in the following MIB is used synonymously with the term Delivery Function.

B.1 Intercept MIB

```
PKTC-ES-TAP-MIB DEFINITIONS ::= BEGIN

IMPORTS
        Counter32,
        Integer32,
        MODULE-IDENTITY,
        NOTIFICATION-TYPE,
        OBJECT-TYPE,
        Unsigned32
                FROM SNMPv2-SMI

        MODULE-COMPLIANCE,
        NOTIFICATION-GROUP,
        OBJECT-GROUP
                FROM SNMPv2-CONF

        InetAddress,
        InetAddressType,
        InetPortNumber
                FROM INET-ADDRESS-MIB

        DateAndTime,
        RowStatus,
        TruthValue,
        TEXTUAL-CONVENTION
                FROM SNMPv2-TC

        SnmpAdminString
                FROM SNMP-FRAMEWORK-MIB

        InterfaceIndexOrZero
                FROM IF-MIB

        pktcESSupportMibs
                FROM CLAB-DEF-MIB;


 pktcESTapMib MODULE-IDENTITY
        LAST-UPDATED  " 200604060000Z"
        ORGANIZATION  "PacketCable"
        CONTACT-INFO
              "Bernie McKibben
               Cable Television Laboratories, Inc.
               858 Coal Creek Circle,
               Louisville, CO 80027, USA
               Phone: +1 303-661-3823
               Email: mibs@cablelabs.com

               Primary Author: Srinivas Dhulipala, Cisco Systems
               "
        DESCRIPTION
                "This module manages intercept feature.
                This MIB defines a generic stream table that contains
                fields common to all intercept types. Specific
intercept
                filters are defined in extension MIBs, e.g., the
                IP-TAP-MIB for IP intercepts."
        REVISION        "200604060000Z"
        DESCRIPTION
                "Initial version of this MIB module."
        ::= { pktcESSupportMibs 1 }


pktcESTapMibNotifs         OBJECT IDENTIFIER ::= { pktcESTapMib 0 }
pktcESTapMibObjects        OBJECT IDENTIFIER ::= { pktcESTapMib 1 }
pktcESTapMibConform        OBJECT IDENTIFIER ::= { pktcESTapMib 2 }


pktcEScTapMediationGroup   OBJECT IDENTIFIER ::= { pktcESTapMibObjects
1 }
pktcEScTapStreamGroup      OBJECT IDENTIFIER ::= { pktcESTapMibObjects
2 }
pktcEScTapDebugGroup       OBJECT IDENTIFIER ::= { pktcESTapMibObjects
3 }


--
-- textual convention
--

PktcEScTapDscp ::= TEXTUAL-CONVENTION
     STATUS     current
     DESCRIPTION
        "An integer that is in the range of the DiffServ codepoint
        values."
     SYNTAX INTEGER (0..63)


-- pktcEScTapMediationNewIndex is defined to allow a network manager
-- to create a new Mediation Table entry and its corresponding
-- Stream Table entries without necessarily knowing what other
-- entries might exist.


pktcEScTapMediationNewIndex OBJECT-TYPE
     SYNTAX     Integer32 (1..2147483647)
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
        "This object contains a value which may be used as an index
        value for a new pktcEScTapMediationEntry. Whenever read, the
       agent will change the value to a new non-conflicting value.
       This is to reduce the probability of errors during creation of
       new pktcEScTapMediationTable entries."
     ::= { pktcEScTapMediationGroup 1 }


-- The Tap Mediation Table lists the applications, by address and
-- port number, to which traffic may be intercepted. These may be -- on
the same or different Mediation Devices.


pktcEScTapMediationTable OBJECT-TYPE
     SYNTAX     SEQUENCE OF PktcEScTapMediationEntry
     MAX-ACCESS not-accessible
     STATUS     current
     DESCRIPTION
        "This table lists the Mediation Devices with which the
        intercepting device communicates. These may be on the same or
different Mediation Devices.



        This table is written by the Mediation Device, and is always
        volatile. This is because intercepts may disappear during a
        restart of the intercepting equipment.

        Entries are added to this table via pktcEScTapMediationStatus
in
        accordance with the RowStatus convention."
     ::= { pktcEScTapMediationGroup 2 }


pktcEScTapMediationEntry OBJECT-TYPE
     SYNTAX     PktcEScTapMediationEntry
     MAX-ACCESS not-accessible
     STATUS     current
     DESCRIPTION
        "The entry describes a single session maintained with an
        application on a Mediation Device."
     INDEX      { pktcEScTapMediationContentId }
     ::= { pktcEScTapMediationTable 1 }


PktcEScTapMediationEntry ::= SEQUENCE {
        pktcEScTapMediationContentId          Integer32,
        pktcEScTapMediationDestAddressType    InetAddressType,
        pktcEScTapMediationDestAddress        InetAddress,
        pktcEScTapMediationDestPort           InetPortNumber,
        pktcEScTapMediationSrcInterface       InterfaceIndexOrZero,
        pktcEScTapMediationDscp               PktcEScTapDscp,
        pktcEScTapMediationTimeout            DateAndTime,
        pktcEScTapMediationTransport          INTEGER,
        pktcEScTapMediationNotificationEnable TruthValue,
        pktcEScTapMediationStatus             RowStatus
}
```

```
pktcEScTapMediationContentId OBJECT-TYPE
     SYNTAX     Integer32 (1..2147483647)
     MAX-ACCESS not-accessible
     STATUS     current
     DESCRIPTION
        "pktcEScTapMediationContentId is a session identifier, from the
        intercept application's perspective, and a content identifier
        from the Mediation Device's perspective. The Mediation Device
        is responsible for making sure these are unique, although the
        SNMP RowStatus row creation process will help by not allowing
        it to create conflicting entries. Before creating a new entry,
        a value for this variable may be obtained by reading
        pktcEScTapMediationNewIndex to reduce the probability of a
value
        collision."
     ::= { pktcEScTapMediationEntry 1 }


pktcEScTapMediationDestAddressType OBJECT-TYPE
     SYNTAX     InetAddressType
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The type of pktcEScTapMediationDestAddress."
     ::= { pktcEScTapMediationEntry 2 }


pktcEScTapMediationDestAddress OBJECT-TYPE
     SYNTAX     InetAddress
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The IP Address of the Mediation Device's network interface
        to which to direct intercepted traffic."
     ::= { pktcEScTapMediationEntry 3 }


pktcEScTapMediationDestPort OBJECT-TYPE
     SYNTAX     InetPortNumber
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The port number on the Mediation Device's network interface
        to which to direct intercepted traffic."
     ::= { pktcEScTapMediationEntry 4 }


pktcEScTapMediationSrcInterface OBJECT-TYPE
     SYNTAX     InterfaceIndexOrZero
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The interface on the intercepting device from which to
        transmit intercepted data. If zero, any interface may be used
        according to normal IP practice."
     ::= { pktcEScTapMediationEntry 5 }
```

```
pktcEScTapMediationDscp OBJECT-TYPE
     SYNTAX     PktcEScTapDscp
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The Differentiated Services Code Point the intercepting
        device applies to the IP packets encapsulating the
        intercepted traffic."
     DEFVAL { 34 }        -- by default, AF41, code 100010
     ::= { pktcEScTapMediationEntry 7 }



pktcEScTapMediationTimeout OBJECT-TYPE
     SYNTAX     DateAndTime
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The time at which this row and all related Stream Table rows
        should be automatically removed, and the intercept function
        cease. Since the initiating network manager may be the only
        device able to manage a specific intercept or know of its
        existence, this acts as a fail-safe for the failure or removal
        of the network manager. The object is only effective when the
        value of pktcEScTapMediationStatus is 'active'."
     ::= { pktcEScTapMediationEntry 10 }


pktcEScTapMediationTransport OBJECT-TYPE
     SYNTAX     INTEGER {
                        udp(1)
                        }
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The protocol used in transferring intercepted data to the
        Mediation Device. The following protocols may be supported:
                  udp:     PacketCable udp format"
     ::= { pktcEScTapMediationEntry 11 }


pktcEScTapMediationNotificationEnable OBJECT-TYPE
     SYNTAX     TruthValue
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "This variable controls the generation of any notifications or
        informs by the MIB agent for this table entry."
     DEFVAL { true }
     ::= { pktcEScTapMediationEntry 12 }


pktcEScTapMediationStatus OBJECT-TYPE
     SYNTAX     RowStatus
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
       "The status of this conceptual row. This object is used to
        manage creation, modification and deletion of rows in this
        table.


        pktcEScTapMediationTimeout may be modified at any time (even
       while the row is active). But when the row is active, the other
        writable objects may not be modified without setting its value
        to 'notInService'.


        The entry may not be deleted or deactivated by setting its
        value to 'destroy' or 'notInService' if there is any associated
        entry in pktcEScTapStreamTable."
     ::= { pktcEScTapMediationEntry 13 }


--
-- pktcEScTapMediationCapabilities
--


pktcEScTapMediationCapabilities  OBJECT-TYPE
     SYNTAX     BITS {
                         ipV4SrcInterface(0),
                         ipV6SrcInterface(1),
                         udp(2)
                     }
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
         "This object displays the device capabilities with respect to
         certain fields in Mediation Device table. This may be
dependent
         on hardware capabilities, software capabilities.
         The following values may be supported:
             ipV4SrcInterface:  SNMP ifIndex Value may be used to
select
                                the interface (denoted by
                                pktcEScTapMediationSrcInterface) on the
                                intercepting device from which to
                                transmit intercepted data to an IPv4
                                address Mediation Device.


             ipV6SrcInterface:  SNMP ifIndex Value may be used to
select
                                the interface (denoted by
                                pktcEScTapMediationSrcInterface) on the
                                intercepting device from which to
                                transmit intercepted data to an IPv6
                                address Mediation Device.
             udp:              UDP may be used as transport protocol
                               (denoted by
pktcEScTapMediationTransport) in
                               transferring intercepted data to the
                               Mediation Device."
     ::= { pktcEScTapMediationGroup 3 }

--
-- The stream tables
--
-- This MIB defines a generic stream table containing fields that are
-- common to any kind of filter specification and a type of the
-- filter specification. Filter specifications can be for various type
-- of intercepts (eg. IPv4, IPv6, MAC, VoIP) and each of the filters
-- is defined in extension MIBs.
--

pktcEScTapStreamTable OBJECT-TYPE
     SYNTAX       SEQUENCE OF PktcEScTapStreamEntry
     MAX-ACCESS not-accessible
     STATUS       current
     DESCRIPTION
        "The Intercept Stream Table lists the traffic streams to be
        intercepted. The same data stream may be required by multiple
        taps, and one might assume that often the intercepted stream
        is a small subset of the traffic that could be intercepted.


        The Table consists of generic fields that are independent
        of the type of intercept. It contains type of the specific
        filter which is defined in an extension MIB and counters to
        account for packets intercepted or dropped by the attached
        filter specification.

        Note that the Mediation Device must make sure there is
        only one type of specific filter created with the same
        indices as that of a row in this table, otherwise the
        later creations will fail. For example, if there is a
        row in this table with index 1.2, there can be a
        corresponding row with the same index either in
        tapStreamTable, c8tapStreamTable or cuctTapStreamTable,
        but not all.


        The first index indicates which Mediation Device the
        intercepted traffic will be diverted to. The second index
        permits multiple classifiers to be used together.

        Entries are added to this table via pktcEScTapStreamStatus in
        accordance with the RowStatus convention."
     ::= { pktcEScTapStreamGroup 1 }


pktcEScTapStreamEntry OBJECT-TYPE
     SYNTAX     PktcEScTapStreamEntry
     MAX-ACCESS not-accessible
     STATUS     current
     DESCRIPTION
        "A stream entry indicates a single data stream to be
        intercepted to a Mediation Device. Many selected data
        streams may go to the same application interface, and many
        application interfaces are supported."
     INDEX { pktcEScTapMediationContentId, pktcEScTapStreamIndex }
     ::= { pktcEScTapStreamTable 1 }


PktcEScTapStreamEntry ::= SEQUENCE {
        pktcEScTapStreamIndex                 Integer32,
        pktcEScTapStreamType                  INTEGER,
        pktcEScTapStreamInterceptEnable       TruthValue,
        pktcEScTapStreamInterceptedPackets    Counter32,
        pktcEScTapStreamInterceptDrops        Counter32,
        pktcEScTapStreamStatus                RowStatus
}


pktcEScTapStreamIndex OBJECT-TYPE
     SYNTAX     Integer32 (1..2147483647)
     MAX-ACCESS not-accessible
     STATUS     current
     DESCRIPTION
        "The index of the stream itself."
     ::= { pktcEScTapStreamEntry 1 }

pktcEScTapStreamType OBJECT-TYPE
     SYNTAX     INTEGER {
                ip(1),
                mac(2),
                userConnection(3),
                msPdsn(4)
     }
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "Identifies the type of intercept filter associated to this
        generic stream. The following type of streams is supported:
               ip:             The specific filter is an IP filter
                               with same indices as that of this
                               table. The exact filter is a row in
                               tapStreamTable of IP-TAP-MIB."
     ::= {pktcEScTapStreamEntry 2 }

pktcEScTapStreamInterceptEnable OBJECT-TYPE
     SYNTAX     TruthValue
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
         "If 'true', the tap should intercept matching traffic. The
         value for this object should be set to 'true' only after an
         additional filter specification has been attached to this
         stream."
     DEFVAL { false }
     ::= { pktcEScTapStreamEntry 3 }
```

```
pktcEScTapStreamInterceptedPackets OBJECT-TYPE
     SYNTAX     Counter32
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
        "The number of packets matching this data stream specification
        that have been intercepted."
     ::= { pktcEScTapStreamEntry 4 }

pktcEScTapStreamInterceptDrops OBJECT-TYPE
     SYNTAX     Counter32
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
        "The number of packets matching this data stream specification
        that, having been intercepted, were dropped in the lawful
        intercept process."
     ::= { pktcEScTapStreamEntry 5 }


pktcEScTapStreamStatus OBJECT-TYPE
     SYNTAX     RowStatus
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The status of this conceptual row. This object manages
        creation, modification, and deletion of rows in this table.
        pktcEScTapStreamInterceptEnable may be modified even the
        value of this entry rowStatus object is 'active'.  When other
        rows must be changed, pktcEScTapStreamStatus must be first set
        to 'notInService'."
     ::= { pktcEScTapStreamEntry 6 }

--
-- The debug information
--

pktcEScTapDebugAge OBJECT-TYPE
     SYNTAX     Integer32 (1..2147483647)
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
        "This object contains the duration in minutes for which an
        entry in pktcEScTapDebugTable is maintained by the implementing
        device after which the entry is deleted. The management
        station also has the option of deleting the entry itself
        by setting pktcEScTapDebugStatus."
     ::= { pktcEScTapDebugGroup 1 }

pktcEScTapDebugMaxEntries OBJECT-TYPE
     SYNTAX     Integer32 (1..2147483647)
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
        "This object contains the maximum number of debug messages
        maintained by the implementing device at a time. If this
        limit is crossed, most recent message will replace the
        least recent message."
     ::= { pktcEScTapDebugGroup 2 }

pktcEScTapDebugTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PktcEScTapDebugEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains Lawful Intercept debug messages
        generated by the implementing device. This table is used
        by pktcESTapMediationDebug and pktcESTapStreamDebug
        notifications.

        An entry in this table contains a debug message which is
        regarding either a Mediation Device or a intercept stream
        created by a Mediation Device. The Mediation device is
        identified by pktcEScTapDebugMediationId whose value is
        that of pktcEScTapMediationContentId of
pktcEScTapMediationEntry.
        The stream is identified by pktcEScTapDebugMediationId and
        pktcEScTapDebugStreamId whose values are that of
        pktcEScTapMediationContentId and pktcEScTapStreamIndex of
        the corresponding pktcEScTapStreamEntry.

        Note that pktcEScTapDebugStreamId may be zero for an entry,
        in which case the debug message is regarding a Mediation
        Device.

        Entries are added to this table via pktcEScTapDebugStatus in
        accordance with the RowStatus convention."
    ::= { pktcEScTapDebugGroup 3 }


pktcEScTapDebugEntry OBJECT-TYPE
    SYNTAX      PktcEScTapDebugEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of the debug messages."
    INDEX { pktcEScTapDebugIndex }
    ::= { pktcEScTapDebugTable 1 }


PktcEScTapDebugEntry ::= SEQUENCE {
        pktcEScTapDebugIndex       Integer32,
        pktcEScTapDebugMediationId Unsigned32,
        pktcEScTapDebugStreamId    Unsigned32,
        pktcEScTapDebugMessage     SnmpAdminString,
        pktcEScTapDebugStatus      RowStatus
}


pktcEScTapDebugIndex OBJECT-TYPE
     SYNTAX         Integer32 (1..2147483647)
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
```

```
        "Index to the debug table."
     ::= { pktcEScTapDebugEntry 1 }


pktcEScTapDebugMediationId OBJECT-TYPE
     SYNTAX        Unsigned32
     MAX-ACCESS    read-only
     STATUS        current
     DESCRIPTION
        "The value of this object is pktcEScTapMediationContentId
        identifying an entry in pktcEScTapMediationTable. Note this
        object may contain a value for which an entry in
        pktcEScTapMediationTable
        does not exist. This happens when creation of an entry in
        pktcEScTapMediationTable fails and this debug message conveys
        more detailed information regarding the failure."
     ::= { pktcEScTapDebugEntry 2 }

pktcEScTapDebugStreamId OBJECT-TYPE
     SYNTAX        Unsigned32
     MAX-ACCESS    read-only
     STATUS        current
     DESCRIPTION
        "The value of this object is that of pktcEScTapStreamIndex of
an
        entry in pktcEScTapStreamTable. This object along with
        pktcEScTapDebugMediationId identifies an entry in
        pktcEScTapStreamTable.
        The value of this object may be zero, in which this debug
        message is regarding a Mediation Device, but not a particular
        stream.  Note this object may contain a value for which an
        entry in pktcEScTapMediationTable does not exist. This happens
        when creation of an entry in pktcEScTapStreamTable fails."
     ::= { pktcEScTapDebugEntry 3 }

pktcEScTapDebugMessage OBJECT-TYPE
     SYNTAX       SnmpAdminString
     MAX-ACCESS   read-only
     STATUS       current
     DESCRIPTION
        "A text string contains the debug message."
     ::= { pktcEScTapDebugEntry 4 }

pktcEScTapDebugStatus OBJECT-TYPE
     SYNTAX     RowStatus
     MAX-ACCESS read-write
     STATUS     current
     DESCRIPTION
        "The status of this conceptual row. A row in this table is
        created by the implementing device. A management station cannot
        modify any of the objects in this row, except deleting the row
        by setting this object to 'destroy'."
     ::= { pktcEScTapDebugEntry 5 }


-- notifications
```

```
pktcESTapMibActive   NOTIFICATION-TYPE
     STATUS     current
     DESCRIPTION
        "This Notification is sent when an intercepting router or
        switch is first capable of intercepting a packet corresponding
        to a configured data stream. The value of the corresponding
        pktcEScTapStreamType which identifies the actual intercept
        stream type is included in this notification.


        This notification may be generated in conjunction with the
        intercept application, which is designed to expect the
        notification to be sent as reliably as possible, e.g., through
        the use of a finite number of retransmissions until
        acknowledged, as and when such mechanisms are available; for
        example, with SNMPv3, this would be an InformRequest.  Filter
        installation can take a long period of time, during which call
        progress may be delayed."
     ::= { pktcESTapMibNotifs 1 }


pktcESTapMediationTimedOut NOTIFICATION-TYPE
     OBJECTS    { pktcEScTapMediationStatus }
     STATUS     current
     DESCRIPTION
        "When an intercept is autonomously removed by an intercepting
        device, such as due to the time specified in
        pktcEScTapMediationTimeout arriving, the device notifies the

        of the action."
     ::= { pktcESTapMibNotifs 2 }


pktcESTapMediationDebug NOTIFICATION-TYPE
     OBJECTS    { pktcEScTapDebugMediationId, pktcEScTapDebugMessage }
     STATUS     current
     DESCRIPTION
        "When there is intervention needed due to some events related
        to entries configured in pktcEScTapMediationTable, the device
        notifies the manager of the event.


        This notification may be generated in conjunction with the
        intercept application, which is designed to expect the
        notification to be sent as reliably as possible, e.g., through
        the use of a finite number of retransmissions until
        acknowledged, as and when such mechanisms are available; for
        example, with SNMPv3, this would be an InformRequest."
     ::= { pktcESTapMibNotifs 3 }


pktcESTapStreamDebug NOTIFICATION-TYPE
     OBJECTS    { pktcEScTapDebugMediationId, pktcEScTapDebugStreamId,
                  pktcEScTapDebugMessage }
     STATUS     current
     DESCRIPTION
```

```
        "When there is intervention needed due to some events related
        to entries configured in pktcEScTapStreamTable, the device
        notifies the manager of the event.


        This notification may be generated in conjunction with the
        intercept application, which is designed to expect the
        notification to be sent as reliably as possible, e.g., through
        the use of a finite number of retransmissions until
        acknowledged, as and when such mechanisms are available; for
        example, with SNMPv3, this would be an InformRequest."
     ::= { pktcESTapMibNotifs 4 }


pktcESTapSwitchover NOTIFICATION-TYPE
     STATUS     current
     DESCRIPTION
        "This notification is sent when there is a redundant (standby)
        route processor available on the intercepting device and the
        current active processor is going down causing standby to
        takeover. Note that this notification may be sent by the
        intercepting device only when it had a chance to know before it
        goes down.

        Mediation device when received this notification should assume
        that configured intercepts on the intercepting device no longer
        exist, when the standby processor takes control. This means
that
        the Mediation device should again configure the intercepts."
     ::= { pktcESTapMibNotifs 5 }


-- conformance information


pktcESTapMibCompliances OBJECT IDENTIFIER ::= { pktcESTapMibConform 1 }
pktcESTapMibGroups      OBJECT IDENTIFIER ::= { pktcESTapMibConform 2 }


-- compliance statement


pktcESTapMibCompliance MODULE-COMPLIANCE
     STATUS  current
     DESCRIPTION
        "The compliance statement for entities which implement the
         Intercept MIB"
     MODULE        -- this module
        MANDATORY-GROUPS {
                pktcESTapMediationComplianceGroup,
                pktcESTapStreamComplianceGroup,
                pktcESTapMediationCpbComplianceGroup,
                pktcESTapNotificationGroup
        }
     ::= { pktcESTapMibCompliances 1 }
```

```
-- units of conformance

pktcESTapMediationComplianceGroup OBJECT-GROUP
     OBJECTS {
        pktcEScTapMediationNewIndex,
        pktcEScTapMediationDestAddressType,
        pktcEScTapMediationDestAddress,
        pktcEScTapMediationDestPort,
        pktcEScTapMediationSrcInterface,
        pktcEScTapMediationDscp,
        pktcEScTapMediationTimeout,
        pktcEScTapMediationTransport,
        pktcEScTapMediationNotificationEnable,
        pktcEScTapMediationStatus
     }
     STATUS     current
     DESCRIPTION
        "These objects are necessary for description of the data
        streams directed to a Mediation Device."
     ::= { pktcESTapMibGroups 1 }


pktcESTapStreamComplianceGroup OBJECT-GROUP
     OBJECTS {
        pktcEScTapStreamType,
        pktcEScTapStreamInterceptEnable,
        pktcEScTapStreamInterceptedPackets,
        pktcEScTapStreamInterceptDrops,
        pktcEScTapStreamStatus
     }
     STATUS     current
     DESCRIPTION
        "These objects are necessary for a description of the packets
        to select for interception."
     ::= { pktcESTapMibGroups 2 }


pktcESTapNotificationGroup NOTIFICATION-GROUP
     NOTIFICATIONS {
         pktcESTapMibActive,
         pktcESTapMediationTimedOut,
         pktcESTapMediationDebug,
         pktcESTapStreamDebug,
         pktcESTapSwitchover
     }
     STATUS     current
     DESCRIPTION
        "These notifications are used to present status from the
        intercepting device to the Mediation Device."
     ::= { pktcESTapMibGroups 3 }


pktcESTapMediationCpbComplianceGroup OBJECT-GROUP
     OBJECTS {
        pktcEScTapMediationCapabilities
     }
     STATUS     current
     DESCRIPTION
        "These objects are necessary for a description of the
        mediation device to select for Lawful Intercept."
     ::= { pktcESTapMibGroups 4 }


pktcESTapDebugComplianceGroup OBJECT-GROUP
     OBJECTS {
        pktcEScTapDebugAge,
        pktcEScTapDebugMaxEntries,
        pktcEScTapDebugMediationId,
        pktcEScTapDebugStreamId,
        pktcEScTapDebugMessage,
        pktcEScTapDebugStatus
     }
     STATUS     current
     DESCRIPTION
        "These objects are necessary for debug information."
     ::= { pktcESTapMibGroups 5 }


END
```

B.2 IP TAP MIB

```
PKTC-ES-IPTAP-MIB DEFINITIONS ::= BEGIN

IMPORTS
        Integer32,
        MODULE-IDENTITY,
        OBJECT-TYPE
                FROM SNMPv2-SMI

        MODULE-COMPLIANCE,
        OBJECT-GROUP
                FROM SNMPv2-CONF

        InetAddress,
        InetAddressPrefixLength,
        InetAddressType,
        InetPortNumber
                FROM INET-ADDRESS-MIB

        SnmpAdminString
                FROM SNMP-FRAMEWORK-MIB

        RowStatus
                FROM SNMPv2-TC

        pktcEScTapMediationContentId,
        pktcEScTapStreamIndex
                FROM PKTC-ES-TAP-MIB
        pktcESSupportMibs
```

```
                FROM CLAB-DEF-MIB;


 pktcESIpTapMIB MODULE-IDENTITY
      LAST-UPDATED  " 200604060000Z"
      ORGANIZATION  "PacketCable"
      CONTACT-INFO
              "Bernie McKibben
               Cable Television Laboratories, Inc.
               858 Coal Creek Circle,
               Louisville, CO 80027, USA
               Phone: +1 303-661-3823
               Email: mibs@cablelabs.com

               Primary Author: Srinivas Dhulipala, Cisco Systems"
      DESCRIPTION
              "This module manages intercept feature for IP.

              This MIB is used along with TAP-MIB to
              intercept IP traffic. TAP-MIB along with
              specific filter MIBs like this MIB replace
              TAP-MIB.

              To create an IP intercept, an entry pktcESTapStreamEntry
              is created which contains the filter details. An entry
              pktcEScpktcESTapStreamEntry of TAP-MIB is created, which
              is the common stream information for all kinds of
              intercepts and type of the specific stream is set to
              ip in this entry."

      REVISION        " 200604060000Z"
      DESCRIPTION
              "Initial version of this MIB module."
      ::= { pktcESSupportMibs 2 }

pktcESIpTapMIBNotifs           OBJECT IDENTIFIER ::= { pktcESIpTapMIB 0
}
pktcESIpTapMIBObjects          OBJECT IDENTIFIER ::= { pktcESIpTapMIB 1
}
pktcESIpTapMIBConform          OBJECT IDENTIFIER ::= { pktcESIpTapMIB 2
}

pktcESTapStreamEncodePacket OBJECT IDENTIFIER ::= {
pktcESIpTapMIBObjects 1 }

--
-- The filter specifics for intercepting IPv4 and IPv6 traffic
--

pktcESTapStreamCapabilities  OBJECT-TYPE
     SYNTAX     BITS {
                         tapEnable(0),
                         interface(1),
                         ipV4(2),
                         ipV6(3),
                         l4Port(4),
                         dscp(5),
                         voip(6)
                     }
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
         "This object displays what types of intercept streams can be
         configured on this type of device. This may be dependent on
         hardware capabilities, software capabilities. The following
         fields may be supported:
             tapEnable:   set if table entries with
                          pktcEScTapStreamInterceptEnable set to
'false'
                          are used to pre-screen packets for intercept;
                          otherwise these entries are ignored.
             interface:   SNMP ifIndex Value may be used to select
                          interception of all data crossing an
                          interface or set of interfaces.
             ipV4:        IPv4 Address or prefix may be used to select
                          traffic to be intercepted.
             ipV6:        IPv6 Address or prefix may be used to select
                          traffic to be intercepted.
             l4Port:      TCP/UDP Ports may be used to select traffic
                          to be intercepted.
             dscp:        DSCP (Differentiated Services Code Point) may
                          be used to select traffic to be intercepted.
             voip:        packets belonging to a voice session may
                          be intercepted using source IPv4 address and
                          source UDP port."
     ::= { pktcESTapStreamEncodePacket 1 }

--
-- The 'access list' for intercepting data at the IP network layer
--


pktcESTapStreamTable OBJECT-TYPE
     SYNTAX       SEQUENCE OF PktcESTapStreamEntry
     MAX-ACCESS   not-accessible
     STATUS       current
     DESCRIPTION
        "The Intercept Stream IP Table lists the IPv4 and IPv6 streams
        to be intercepted.  The same data stream may be required by
        multiple taps, and one might assume that often the intercepted
        stream is a small subset of the traffic that could be
        intercepted.


        This essentially provides options for packet selection, only
        some of which might be used. For example, if all traffic to or
        from a given interface is to be intercepted, one would
        configure an entry which lists the interface, and wild-card
        everything else.  If all traffic to or from a given IP Address
        is to be intercepted, one would configure two such entries
        listing the IP Address as source and destination respectively,
        and wild-card everything else.  If a particular voice on a
        teleconference is to be intercepted, on the other hand, one
        would extract the multicast (destination) IP address, the
        source IP Address, the protocol (UDP), and the source and
        destination ports from the call control exchange and list all
        necessary information.


        The first index indicates which Mediation Device the
        intercepted traffic will be diverted to. The second index
        permits multiple classifiers to be used together, such as
        having an IP address as source or destination. The value of the
        second index is that of the stream's counter entry in the
        pktcEScTapStreamTable.

        Entries are added to this table via pktcESTapStreamStatus in
        accordance with the RowStatus convention."
     ::= { pktcESTapStreamEncodePacket 2 }


pktcESTapStreamEntry OBJECT-TYPE
     SYNTAX     PktcESTapStreamEntry
     MAX-ACCESS not-accessible
     STATUS     current
     DESCRIPTION
        "A stream entry indicates a single data stream to be
        intercepted to a Mediation Device. Many selected data
        streams may go to the same application interface, and many
        application interfaces are supported."
     INDEX      { pktcEScTapMediationContentId, pktcEScTapStreamIndex }
     ::= { pktcESTapStreamTable 1 }


PktcESTapStreamEntry::= SEQUENCE {
        pktcESTapStreamInterface             Integer32,
        pktcESTapStreamAddrType              InetAddressType,
        pktcESTapStreamDestinationAddress    InetAddress,
        pktcESTapStreamDestinationLength     InetAddressPrefixLength,
        pktcESTapStreamSourceAddress         InetAddress,
        pktcESTapStreamSourceLength          InetAddressPrefixLength,
        pktcESTapStreamTosByte               Integer32,
        pktcESTapStreamTosByteMask           Integer32,
        pktcESTapStreamFlowId                Integer32,
        pktcESTapStreamProtocol              Integer32,
        pktcESTapStreamDestL4PortMin         InetPortNumber,
        pktcESTapStreamDestL4PortMax         InetPortNumber,
        pktcESTapStreamSourceL4PortMin       InetPortNumber,
        pktcESTapStreamSourceL4PortMax       InetPortNumber,
        pktcESTapStreamVRF                   SnmpAdminString,
        pktcESTapStreamStatus                RowStatus
}


pktcESTapStreamInterface OBJECT-TYPE
     SYNTAX     Integer32 (-2..2147483647)
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The ifIndex value of the interface over which traffic to be
        intercepted is received or transmitted. The interface may be
```

```
            physical or virtual. If this is the only parameter specified,
            and it is other than -2, -1 or 0, all traffic on the selected
            interface will be chosen.


            If the value is zero, matching traffic may be received or
            transmitted on any interface.  Additional selection parameters
            must be selected to limit the scope of traffic intercepted.
            This is most useful on non-routing platforms or on intercepts
            placed elsewhere than a subscriber interface.


            If the value is -1, one or both of
            pktcESTapStreamDestinationAddress and
            pktcESTapStreamSourceAddress must be specified
            with prefix length greater than zero.
            Matching traffic on the interface pointed to by ipRouteIfIndex
            or ipCidrRouteIfIndex values associated with those values is
            intercepted, whichever is specified to be more focused than a
            default route.  If routing changes, either by operator action
            or by routing protocol events, the interface will change with
            it. This is primarily intended for use on subscriber interfaces
            and other places where routing is guaranteed to be
            symmetrical.


            In both of these cases, it is possible to have the same packet
            selected for intersection on both its ingress and egress
            interface.  Nonetheless, only one instance of the packet is
            sent to the Mediation Device.


            If the value is -2, packets belonging to a Voice over IP (VoIP)
            session identified by pktcESTapStreamSourceAddress,
            pktcESTapStreamSourceLen & pktcESTapStreamSourceL4PortMin may be
            intercepted, as a specific voice session can be identified
            with source IP address and udp port number. Other selection
            parameters may be not considered, even if they are set by
            the Mediation Device.


            This value must be set when creating a stream entry, either to
            select an interface, to select all interfaces, or to select the
            interface that routing chooses. Some platforms may not
            implement the entire range of options."
         REFERENCE  "RFC 1213, RFC 2096"
         ::= { pktcESTapStreamEntry 1 }


pktcESTapStreamAddrType OBJECT-TYPE
     SYNTAX     InetAddressType
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The type of address, used in packet selection."
     DEFVAL     { ipv4 }
     ::= { pktcESTapStreamEntry 2 }
```

```
pktcESTapStreamDestinationAddress OBJECT-TYPE
     SYNTAX     InetAddress
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The Destination address or prefix used in packet selection.
        This address will be of the type specified in
        pktcESTapStreamAddrType."
     DEFVAL       { '00000000'H } -- 0.0.0.0
     ::= { pktcESTapStreamEntry 3 }


pktcESTapStreamDestinationLength OBJECT-TYPE
     SYNTAX     InetAddressPrefixLength
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The length of the Destination Prefix. A value of zero causes
        all addresses to match.  This prefix length will be consistent
        with the type specified in pktcESTapStreamAddrType."
     DEFVAL { 0 } -- by default, any destination address
     ::= { pktcESTapStreamEntry 4 }


pktcESTapStreamSourceAddress OBJECT-TYPE
     SYNTAX     InetAddress
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The Source Address used in packet selection. This address will
        be of the type specified in pktcESTapStreamAddrType."
     DEFVAL       { '00000000'H } -- 0.0.0.0
     ::= { pktcESTapStreamEntry 5 }


pktcESTapStreamSourceLength OBJECT-TYPE
     SYNTAX     InetAddressPrefixLength
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The length of the Source Prefix. A value of zero causes all
        addresses to match. This prefix length will be consistent with
        the type specified in pktcESTapStreamAddrType."
     DEFVAL { 0 } -- by default, any source address
     ::= { pktcESTapStreamEntry 6 }


pktcESTapStreamTosByte OBJECT-TYPE
     SYNTAX     Integer32 (0..255)
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The value of the TOS byte, when masked with
        pktcESTapStreamTosByteMask, of traffic to be intercepted.  If
        pktcESTapStreamTosByte&(~pktcESTapStreamTosByteMask)!=0,
        configuration is rejected."
     DEFVAL { 0 }
     ::= { pktcESTapStreamEntry 7 }


pktcESTapStreamTosByteMask OBJECT-TYPE
     SYNTAX     Integer32 (0..255)
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The value of the TOS byte in an IPv4 or IPv6 header is ANDed
        with pktcESTapStreamTosByteMask and compared with
        pktcESTapStreamTosByte.  If the values are equal, the
comparison
        is equal. If the mask is zero and the TosByte value is zero,
        the result is to always accept."
     DEFVAL { 0 } -- by default, any DSCP or other TOS byte value
     ::= { pktcESTapStreamEntry 8 }


pktcESTapStreamFlowId OBJECT-TYPE
     SYNTAX     Integer32 (-1 | 0..1048575)
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The flow identifier in an IPv6 header. -1 indicates that the
        Flow Id is unused."
     DEFVAL { -1 } -- by default, any flow identifier value
     ::= { pktcESTapStreamEntry 9 }


pktcESTapStreamProtocol OBJECT-TYPE
     SYNTAX     Integer32 (-1 | 0..255)
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The IP protocol to match against the IPv4 protocol number or
        the IPv6 Next- Header number in the packet. -1 means 'any IP
        protocol'."
     DEFVAL { -1 } -- by default, any IP protocol
     ::= { pktcESTapStreamEntry 10 }


pktcESTapStreamDestL4PortMin OBJECT-TYPE
     SYNTAX     InetPortNumber
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The minimum value that the layer-4 destination port number in
        the packet must have in order to match.  This value must be
        equal to or less than the value specified for this entry in
        pktcESTapStreamDestL4PortMax.


        If both pktcESTapStreamDestL4PortMin and
pktcESTapStreamDestL4PortMax
        are at their default values, the port number is effectively
```

```
           unused."
        DEFVAL { 0 } -- by default, any transport layer port number
        ::= { pktcESTapStreamEntry 11 }


pktcESTapStreamDestL4PortMax OBJECT-TYPE
        SYNTAX      InetPortNumber
        MAX-ACCESS read-create
        STATUS      current
        DESCRIPTION
           "The maximum value that the layer-4 destination port number in
           the packet must have in order to match this classifier entry.
           This value must be equal to or greater than the value specified
           for this entry in pktcESTapStreamDestL4PortMin.


           If both pktcESTapStreamDestL4PortMin and
           pktcESTapStreamDestL4PortMax
           are at their default values, the port number is effectively
           unused."
        DEFVAL { 65535 } -- by default, any transport layer port number
        ::= { pktcESTapStreamEntry 12 }


pktcESTapStreamSourceL4PortMin OBJECT-TYPE
        SYNTAX      InetPortNumber
        MAX-ACCESS read-create
        STATUS      current
        DESCRIPTION
           "The minimum value that the layer-4 destination port number in
           the packet must have in order to match.  This value must be
           equal to or less than the value specified for this entry in
           pktcESTapStreamSourceL4PortMax.


           If both pktcESTapStreamSourceL4PortMin and
           pktcESTapStreamSourceL4PortMax are at their default values, the
           port number is effectively unused."
        DEFVAL { 0 } -- by default, any transport layer port number
        ::= { pktcESTapStreamEntry 13 }


pktcESTapStreamSourceL4PortMax OBJECT-TYPE
        SYNTAX      InetPortNumber
        MAX-ACCESS read-create
        STATUS      current
        DESCRIPTION
           "The maximum value that the layer-4 destination port number in
           the packet must have in order to match this classifier entry.
           This value must be equal to or greater than the value specified
           for this entry in pktcESTapStreamSourceL4PortMin.


           If both pktcESTapStreamSourceL4PortMin and
           pktcESTapStreamSourceL4PortMax are at their default values, the
           port number is effectively unused."
        DEFVAL { 65535 } -- by default, any transport layer port number
```

```
     ::= { pktcESTapStreamEntry 14 }

pktcESTapStreamVRF OBJECT-TYPE
        SYNTAX SnmpAdminString
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
            "An ASCII string, which is the name of a Virtual Routing
            and Forwarding (VRF) table comprising the routing context
            of a Virtual Private Network. The interface or set of
            interfaces on which the packet might be found should be
            selected from the set of interfaces in the VRF table.
            A string length of zero implies that global routing table
            be used for selection of interfaces on which the packet
            might be found."
        DEFVAL { "" } -- by default, global routing table
        ::= { pktcESTapStreamEntry 15 }

pktcESTapStreamStatus OBJECT-TYPE
     SYNTAX     RowStatus
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "The status of this conceptual row. This object manages
        creation, modification, and deletion of rows in this table.
        When any rows must be changed, pktcESTapStreamStatus must be
        first set to 'notInService'."
     ::= { pktcESTapStreamEntry 16 }


-- conformance information


pktcESIpTapMIBCompliances OBJECT IDENTIFIER ::= { pktcESIpTapMIBConform
1 }
pktcESIpTapMIBGroups      OBJECT IDENTIFIER ::= { pktcESIpTapMIBConform
2 }


-- compliance statement


pktcESIpTapMIBCompliance MODULE-COMPLIANCE
     STATUS  current
     DESCRIPTION
        "The compliance statement for entities which implement the
         Intercept MIB for IP."
     MODULE        -- this module
        MANDATORY-GROUPS {
                pktcESIpTapStreamComplianceGroup
        }
     ::= {pktcESIpTapMIBCompliances 1 }

-- units of conformance

pktcESIpTapStreamComplianceGroup OBJECT-GROUP
     OBJECTS {
        pktcESTapStreamCapabilities,
        pktcESTapStreamInterface,
        pktcESTapStreamAddrType,
        pktcESTapStreamDestinationAddress,
        pktcESTapStreamDestinationLength,
        pktcESTapStreamSourceAddress,
        pktcESTapStreamSourceLength,
        pktcESTapStreamTosByte,
        pktcESTapStreamTosByteMask,
        pktcESTapStreamFlowId,
        pktcESTapStreamProtocol,
        pktcESTapStreamDestL4PortMin,
        pktcESTapStreamDestL4PortMax,
        pktcESTapStreamSourceL4PortMin,
        pktcESTapStreamSourceL4PortMax,
        pktcESTapStreamVRF,
        pktcESTapStreamStatus
     }
     STATUS     current
     DESCRIPTION
        "These objects are necessary for a description of IPv4 and IPv6
        packets to select for interception."
     ::= { pktcESIpTapMIBGroups 1 }

END
```

Appendix I Acknowledgements

We wish to thank the vendor participants who contributed directly to this document:

Michael Bilca, – Tridea Works

Bill Foster – Cisco Systems (Lead Author)

Robert Forsythe - Tridea Works

Louis Fourie – Cisco Systems

Tom Holzschuh – Siemens

Craig Muholland – Cisco Systems

Gil Yaakovi - Verint

Bernie McKibben, CableLabs

Appendix II Revision History

The following Engineering Change Notices are included in PKT-SP-ES-INF-I02-061013.

ECN	ECN Date	Summary
ES-INF-N-06.0317-1	6/26/2006	Modify text and add tables to Section 6.2.4.
ES-INF-N-06.0338-2	9/18/2006	Intercept AVP codes for DIAMETER ;messages are changed from TBD to formally assigned, project controlled values.