3GPP TR 29.817 V0.2.0 (2013-08)

Technical Report

3rd Generation Partnership Project;
Technical Specification Group Core Network and Terminals;
Study on XML based access of AF to the PCRF
(Release 12)

MCC selects keywords from stock list.

Keywords

<keyword[, keyword, …]>

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

This clause is optional. If it exists, it is always the second unnumbered clause.

1 Scope

The Technical Report describes solutions for an XML based protocol (e.g. SOAP, Restful HTTP, etc.) between the AF and the PCRF for the case of non IMS applications. The scope of this work will be to provide an XML based equivalent to the Diameter based signalling that is presently specified in 3GPP TS 29.214 [5].

This Study will cover the following alternatives:

- XML based protocol between AF and PCRF
- Adding a new protocol converter (as a stand-alone entity) between PCRF and AF

Investigate a suitable transport protocol for XML, charging issues in case of 3rd party SPs, security issues.

Roaming both home routed and local break out and signalling routing aspects (e.g. DRA handling and others aspects) will also be included.

It will be investigated whether changes to 3GPP specifications are needed and if so which ones.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.203: "Policy and Charging Control architecture".

[3] 3GPP TS 29.212: "Policy and Charging Control (PCC); Reference points".

[4] 3GPP TS 29.213: "Policy and Charging Control signalling flows and QoS parameter mapping".

[5] 3GPP TS 29.214: "Policy and Charging Control over Rx reference point ".

[6] IETF RFC 6202: "Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP".

[7] IETF RFC 4627: "The application/json Media Type for JavaScript Object Notation (JSON)".

[8] IETF RFC 3588: "Diameter Base Protocol".

[9] Architectural Styles and the Design of Network-based Software Architectures, UNIVERSITY OF CALIFORNIA, IRVINE, Roy Thomas Fielding, 2000.

[10] IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

Application Function (AF): element offering application(s) that use IP bearer resources

Protocol Converter (PC): element converting 3rd party application layer protocol to Diameter in order to get an access to the PCC architecture

3.2 Symbols

For the purposes of the present document, the following symbols apply:

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

AF	Application Function
BBERF	Bearer Binding and Event Reporting Function
DRA	Diameter Routing Agent
IP-CAN	IP Connectivity Access Network
OFCS	Offline Charging System
OCS	Online Charging System
PC	Protocol Converter
PCC	Policy and Charging Control
PCEF	Policy and Charging Enforcement Function
PCRF	Policy and Charging Rules Function
SOAP	Simple Object Access Protocol
SPR	Subscription Profile Repository
UDR	User Data Repository
XML	Extensible Markup Language

4 Study area

The study area of the XML based Rx interface is to investigate XML based protocols between the AF of the 3rd party web application developer and the PCRF.

4.1 Architecture and deployment

The following architectures will be considered:

1. Proposal 1: XML based protocol support between AF and PCRF.

2. Proposal 2: Adding a new interworking entity between PCRF and AF. This interworking entity (protocol convertor) may integrate additional functionality. The set of possible functionality (e.g. security, addressing etc.) should be investigated.

3. Proposal 3: Support both Proposal 1 and 2.

The following perspective of architecture and deployment will be considered:

- In both solutions, how to provide XML interface API to 3rd party applications (e.g. SDK) should be also considered, considering the update impact of XML based protocol to 3rd party applications.
- The comparison and the applicable scenarios of the solution 1 and 2 (e.g. pros and cons).
- The deployment consideration of the 3rd party AFs (e.g. no concept of PLMN and Diameter domain).
- The deployment aspects and the required functionalities if solution 1 and 2 are used within a network simultaneously.
- PCRF addressing.
- Session management (e.g. traffic plane events and the stateless of XML based protocols.)

4.2 Functions

When the AF of the 3rd party is access to the PCC architecture, the entity function should be considered.

- For proposal 1, the functional entity over Rx interface (e.g. PCRF, AF, DRA, etc) should be considered according to the issues about information provision, charging, security, routing, etc.
- For proposal 2, the interworking entity functionality between PCRF and AF should be considered according to the issues about information provision, charging, security and routing, as well as the existing entity function of PCRF, AF, DRA and etc.

When the PCRF connects with AF of the 3rd party it should be investigated whether it is reasonable to provide a subset of the functionality of the Diameter based Rx interface. The following information and capabilities over Rx interface should be considered.

From the PCRF perspective, the AF could provide the information including:

- UE information (IP address, identity).
- The required service quality for the data transport (e.g.QoS and the corresponding service data flows).
- The correlation information between the application traffic and the corresponding SDFs.

From the AF perspective, the PCRF could provide the information as a web services including:

- The acceptable authorized resources (e.g. QoS bandwith) in the unsuccessful procedure.
- AF session management including the traffic plane events report.

4.3 Protocols and languages

The following information should be considered.

- The information and procedures needs to be provided by the 3rd application provider based on Rx interface should be considered. Whether all the information over the Rx interface shall be provided by 3rd party applications should be considered
- The proposed protocols and languages should be considered.
- The comparison of the proposed protocols and languages.

The following protocols and languages may be studied:

- SOAP
- REST
- XML
- JSON

The possible candidate languages and protocols will be presented, and the relationships of these concepts will be clarified. XML and JSON are considered as languages, SOAP and REST are considered as protocols. The protocols (SOAP or REST) are specified for exchanging structured information (XML or JSON).

Editor's Note: The impact from http transmitting characteristic is FFS.

4.4 Charging

The charging issue of this WID is based on the current charging feature of Diameter based Rx interface.

Editor's Note: The merging of new charging issue about on-going project (e.g. MOSAP) is FFS.

4.5 Roaming and routing

- For proposal 1, the AF addressing PCRF, and vice versa should be considered.
- For proposal 2, the routing between the interworking entity and PCRF and the routing between AF and interworking entity should be considered.
- Study how an AF located in a different IP address domain than the PCRF and/or PCEF can be supported.

Note: For instance, for local breakout and home routed, a 3rd party AF can be located either in the visited country or region of a roaming user or in his/her home country/region and can either contact the H-PCRF or V-PCRF.

4.6 Security

The security issues including service provider access control, media type and content authorization, requested service authorization based on contract between application provider and operators should be considered.

4.7 New consideration for XML based Rx interface

The XML based protocol (e.g. SOAP, Restful HTTP, etc.) between the AF and the PCRF brings new consideration:

- A two-way communication needs to be made available so that the PCRF can notify the AF of traffic plane events happening in the RAN/Core. Unlike Diameter, with XML based protocol, the PCRF cannot send requests to the AF.
- Unlike Diameter, XML based protocol is a short connection without heartbeats, how the PCRF can detect the status about AF and the impact to the Rx based interface procedures should be considered.
- Considering the network signalling impact, the kind of applications that are applicable to access to the PCRF and the procedure limitations of AF session should be considered.

5 The Rx interface

Editor's Note: Brief overview over the capabilities of PCC in relation to the Rx interface. E.g. meaningful elements, which may be used by an XML based Rx interface, could be mentioned.

6 Architecture of the 3rd party web service based on Rx

6.1 General

6.2 Protocol converter

6.2.1 Architecture description

The Rx reference point is defined between the PCRF and the AF. Protocol between the PC and AF is based on some Application level protocols, e.g. SOAP.

The PC converts the 3rd party application layer protocol to Diameter for PCRF. As defined in the stage 2 specifications (3GPP TS 23.203 [2]), information from AF is part of the input used by the PCRF for the Policy and Charging Control (PCC) decisions. The PCRF exchanges the PCC rules with the Policy and Charging Enforcement Function (PCEF) and QoS rules with the Bearer Binding and Event Reporting Function (BBERF) as specified in 3GPP TS 29.212 [3]. Signalling flows related to the both Rx and Gx interfaces are specified in 3GPP TS 29.213 [4].

The relationships between the different functional entities involved are depicted in figure 6.2.1.1.

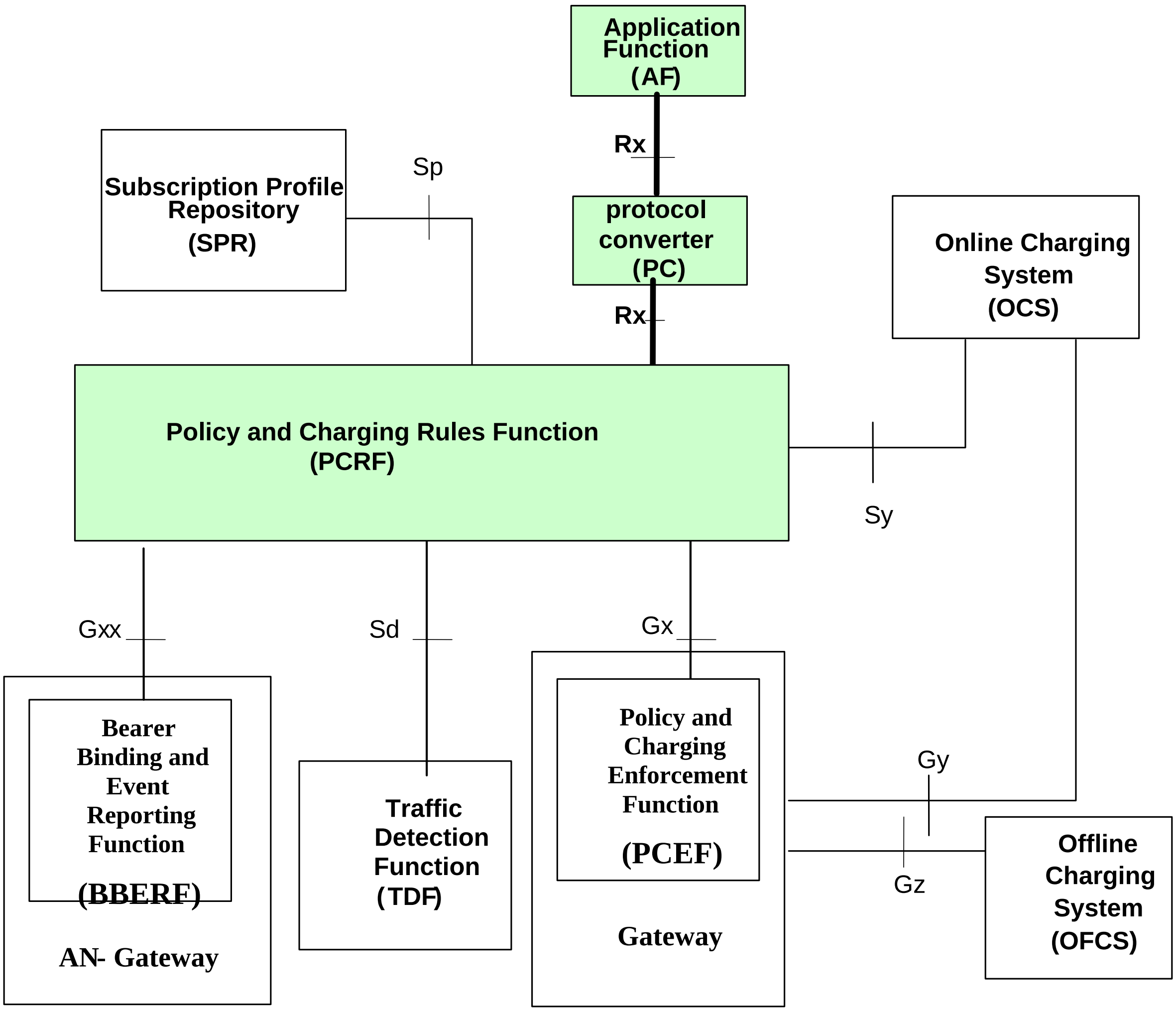

Figure 6.2.1.1: PC at the PCC architecture with SPR for non-romaing scenario

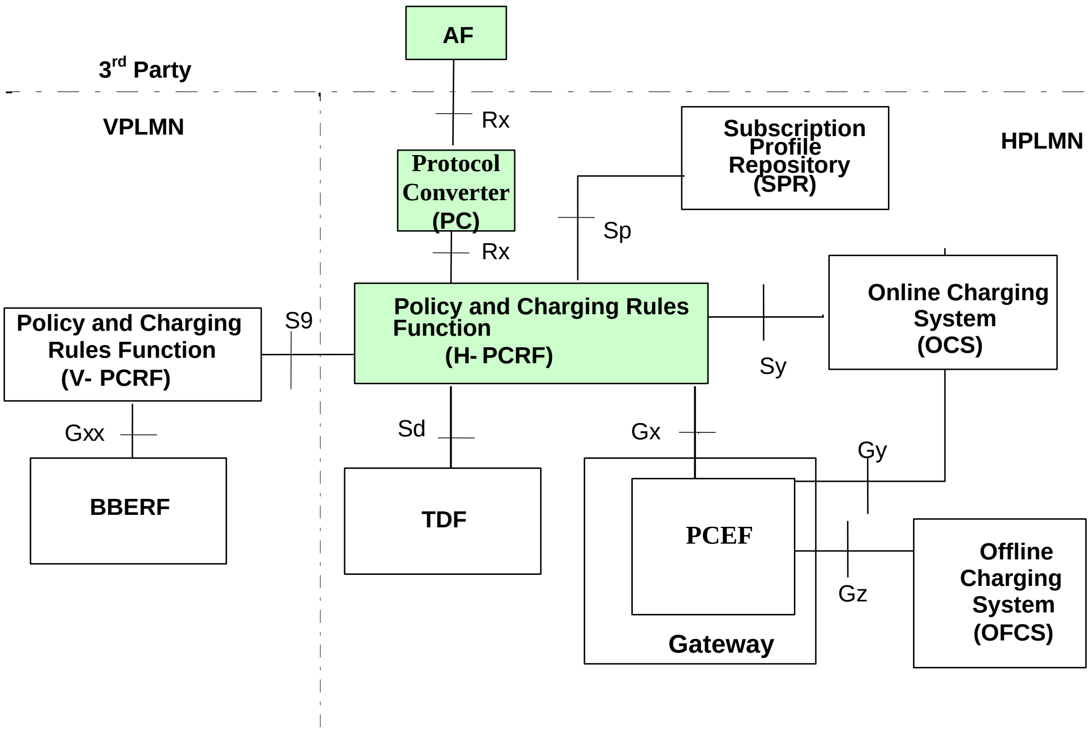

Figure 6.2.1.2: PC at the PCC architecture with SPR for home routed scenario

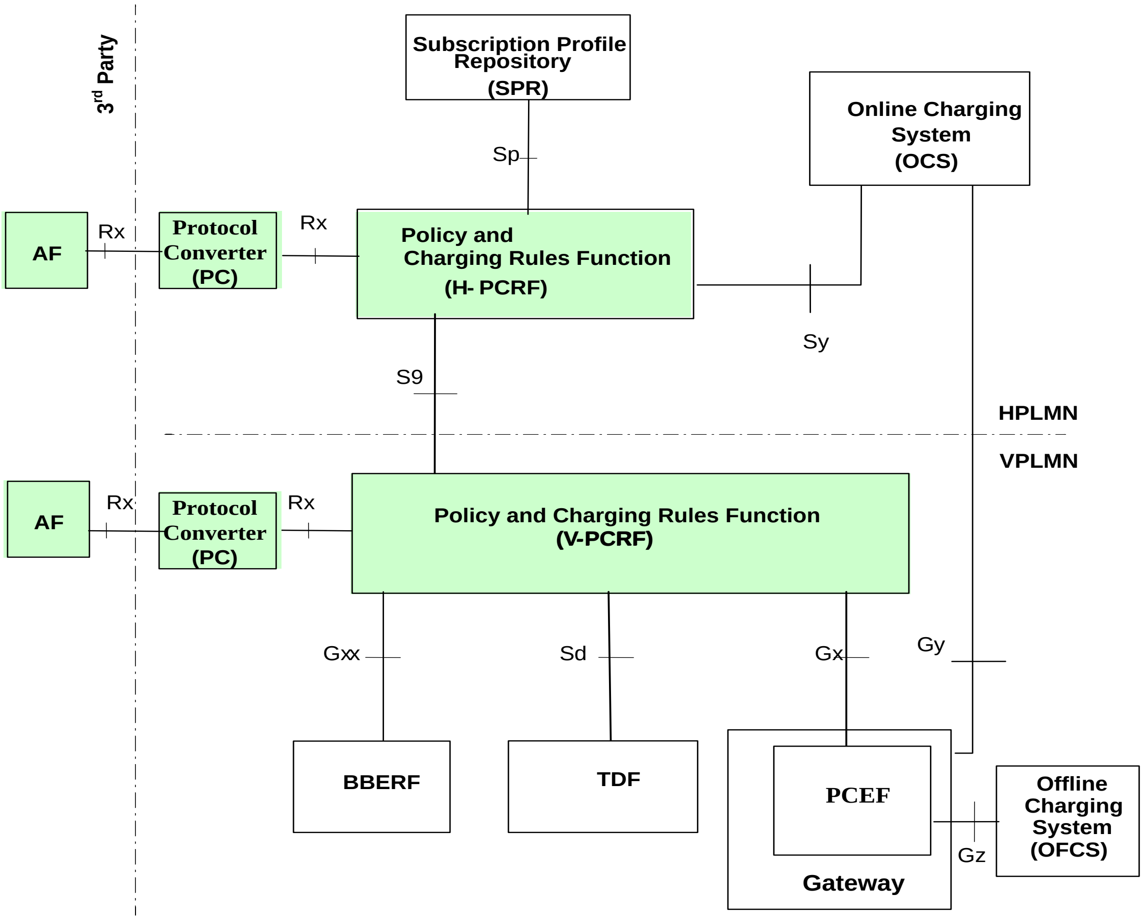

Figure 6.2.1.3: PC at the PCC architecture with SPR for local breakout scenario

NOTE 1: The PCEF may support Application Detection and Control feature.

NOTE 2: When UDC architecture is used, SPR and Sp, whenever mentioned in this document, are replaced by UDR and Ud.

Editor's Note: The protocol between the PC and PCRF is Diameter-based. The protocol between the PC and AF is FFS.

6.2.2 Procedures of protocol converter

6.2.2.1 AF Session Establishment

This subclause describes the signalling flow for the AF session establishment procedure through the PC.

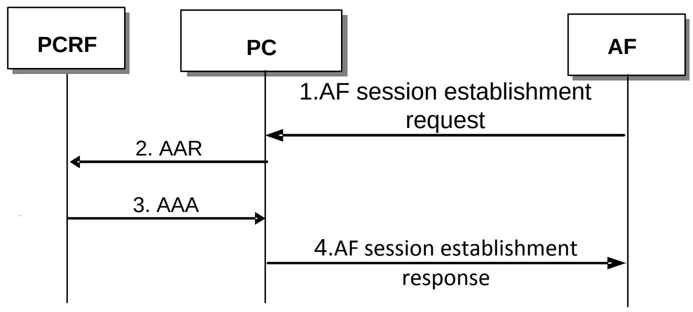

Figure 6.2.2.1.1: Initial AF session request

1. The AF sends the AF session establishment request to the PC including the UE IP address and the corresponding Service Information. UE identity, PDN information and domain identity may be provided if available.
2. The PC converts the request to AAR command. The PC provides the service information to the PCRF by sending a Diameter AAR for a new Rx Diameter session.
3. The PCRF stores the received Service Information, checked the subscription and other related information, sends a Diameter AAA to the protocol converter.
4. The PC converts the AAA and provides the acknowledge information to the AF.

Editor's Note: The protocol between the PC and AF is FFS.

Editor's Note: It is FFS whether there are multiple PCs in one Diameter domain. And whether the PCRF discovery needs to be enhanced is FFS.

6.2.2.2 AF Session Modification initiated by AF

This subclause describes the signalling flow for the AF session modification procedure through PC.

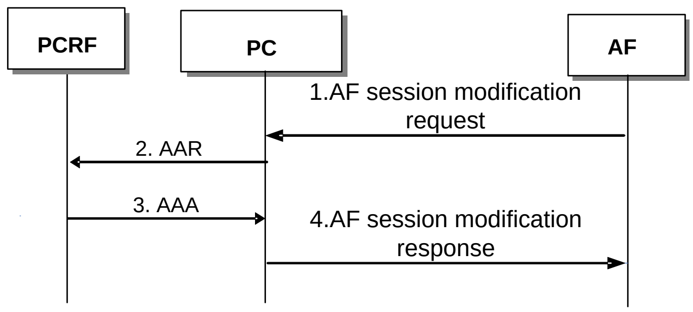

Figure 6.2.2.2.1: AF session modification initiated by AF

1. The AF sends the AF session modification request to the PC including the UE IP address and the corresponding Service Information. UE identity, PDN information and domain identity may be provided if available.

2. The PC converts the request to AAR command. The PC provides the service information to the PCRF by sending a Diameter AAR for a new Rx Diameter session.

3. The PCRF stores the received Service Information, checked the subscription and other related information, sends a Diameter AAA to the PC.

4. The PC converts the AAA and provides the acknowledge information to the AF.

Editor's Note: The protocol between the PC and AF is FFS.

Editor's Note: It is FFS whether there are multiple PCs in one Diameter domain. And whether the PCRF discovery needs to be enhanced is FFS.

6.2.2.3 Gate Related Procedures

This subclause describes the signalling flow when the AF initializes the gate related procedures with the PCRF through the protocol converter.

Depending on the application, in the Service Information provisioning, the AF may instruct the PCRF when the IP flow(s) are to be enabled or disabled to pass through the IP-CAN.

The procedures is the same as Figure 6.2.2.2.1, the only difference is protocol converter should converts the gate status into Flow-Status including in Media-Component- Description AVP of AAR to PCRF.

6.2.2.4 AF Session Termination

This subclause describes the signalling flow for the AF session termination procedure through the PC.

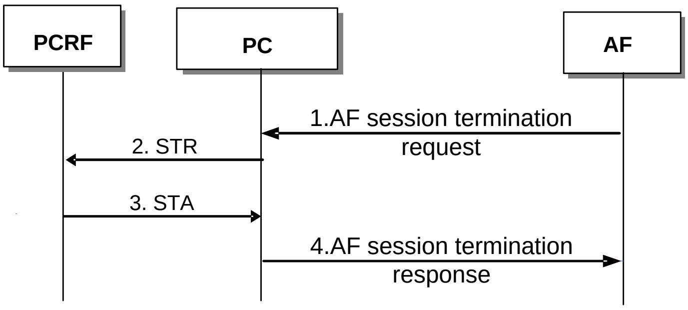

Figure 6.2.2.4.1: AF Session Termination

1. The AF sends the remove resource request to the PC including the UE IP address and the corresponding Service Information to terminate the session and remove the related information. UE identity, PDN information and domain identity may be provided if available.

2. The PC converts the request to STR command. The PC provides the terminated inform for the AF session to the PCRF by sending a Diameter STR for this Rx Diameter session.

3. When the PCRF receives a ST-Request from the PC, indicating an AF session termination, it shall acknowledge that request by sending a ST-Answer to the protocol converter. Afterwards, the corresponding procedures specified at 3GPP TS 29.214 [5].

4. The PC converts the STA and provides the acknowledge information to the AF.

Editor's Note: The protocol between the PC and AF is FFS.

Editor's Note: It is FFS whether there are multiple PCs in one Diameter domain. And whether the PCRF discovery needs to be enhanced is FFS.

6.2.2.5 Subscription to Notification of Signalling Path Status

An AF may subscribe to notifications of the status of the AF Signalling transmission path. And it will be reported to the AF through the PC when the subscribe event happened.

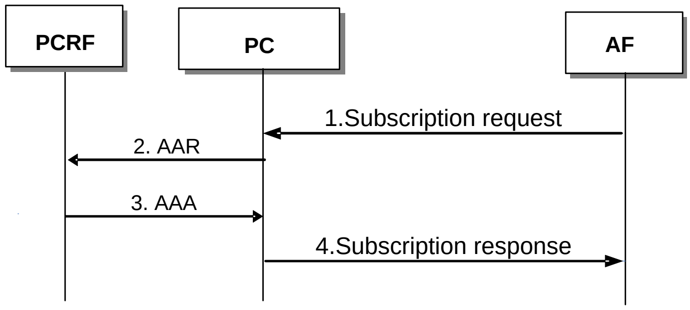

Figure 6.2.2.5.1: Subscription to Notification of Signalling Path Status

1. The AF sends the subscription request to the PC including the UE IP address and the event Information for the subscription. UE identity, PDN information and domain identity may be provided if available.

2. The PC converts the request to AAR command. The PC provides the subscribe event information to the PCRF by sending a Diameter AAR for a new Rx Diameter session.

3. The PCRF stores the received Service Information, performs session binding as described in 3GPP TS 29.213 [4] and acknowledge that request by sending an AAA to the PC. Afterwards, the corresponding procedures specified at 3GPP TS 29.214 [5].

4. .The PC converts the AAA and provides the acknowledge information to the AF.

NOTE: The AF may subscribe other traffic plane events, such as IP-CAN session termination, IP-CAN type changer and access network charing information notification, etc. The procedures are similar with the subscription to Notification of Signalling Path Status.

Editor's Note: The protocol between the PC and AF is FFS.

Editor's Note: It is FFS whether there are multiple PCs in one Diameter domain. And whether the PCRF discovery needs to be enhanced is FFS.

Editor's Note: The protocol converter converts the 3rd party application layer protocol to Diameter in order to get an access to the PCC architecture.

6.2.2.6 Notification of traffic plane events

This sub-clause describes the procedure for the notification of traffic plane events.

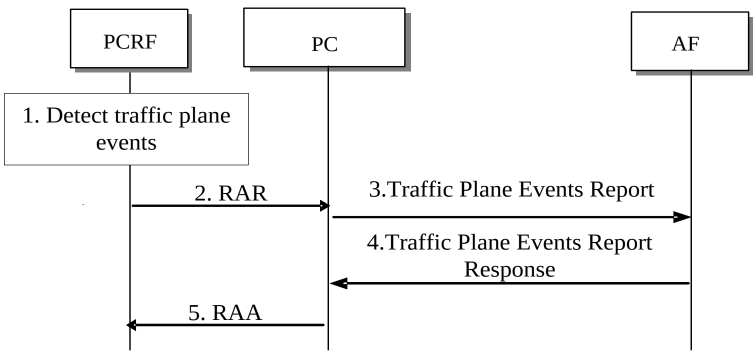

Figure 6.2.2.6.1: Notification of user plane events

1. The PCRF detects the occurrence of the traffic plane events.
2. When the traffic plane events occur, the PCRF notifies the corresponding events to the PC via a Diameter RAR command.
3. The PC converts the Diameter RAR command and sends the Traffic Plane Events Report message to the AF.
4. The PCRF responds the Traffic Plane Events Report Response message to the PC.
5. The PC converts the Traffic Plane Events Report Response message and sends the Diameter RAA command to the PCRF.

Editor’s Note: The protocol between the PC and AF is FFS.

6.2.3 Roaming and routing of protocol converter

6.2.3.1 PC located in the PLMN but outside of the PCRF realm

6.2.3.1.1 General

This subclause specifies the romaing and routing when PC is deployed, including the routing for application level session information between the PCRF and PC, between the PC and the AF.

Editor's Note: The routing between the AF and PC, i.e. how the AF finds the PC, is FFS.

Subclause 6.2.3.1.2, routing between the PCRF and the PC, describes the scenario that multiple and separately addressable PCRFs are present in one Diameter Realm. The PC may use pre-configured information to find the PCRF for single addressable PCRF in one Realm scenario.

The principles for PCRF selection and discovery are as the description in 3GPP TS 23.203[2] clause 7.Within such a deployment, DRA as specified in 3GPP TS 29.213[4] clause 7 is needed.

The messages between PC and PCRF are based on Diameter protocol as specified in IETF RFC 3588 [8]. All the procedures between PC and PCRF involve the DRA functional element.The DRA mode in which it operates (i.e. proxy or redirect) shall be based on operator’s requirements.

6.2.3.1.2 Routing between the Protocol converter and PCRF

As the description in subclause 6.2.2, the PC as one Diameter client of the PCRF converts the request from AF to AAR command, and then sends it to DRA for the PCRF selection to establish the AF session.

In roaming scenario, home routed or local breakout, vPCRF is selected by the DRA located in the visited PLMN, and the hPCRF is selected by the DRA located in the home PLMN.

If the PC has the realm identification (i.e. FQDN from a UE NAI) and is located in the H-PLMN, the PC sends the user identity in the Subscription-Id AVP and PDN information (i.e. APN) if available in the Called-Station-Id AVP in a Diameter request to the DRA which acts as a Diameter agent. If the PC does not have proper knowledge about the user identity and the PC is located in the HPLMN, the PC may use pre-configured information to find the DRA.

The PC shall provide the DRA of the PCRF realm with identity parameters upon the first interaction between the PC and the PCRF realm.

If redirect agent is used for the DRA, the DRA shall use the redirecting requests procedure as specified in IETF RFC 3588 [8], and include the PCRF address (Diameter Identity) in the Redirect-Host AVP in the Diameter reply sent to the PC.

If proxy agent is used for the DRA, the DRA should use the proxy procedure as specified in IETF RFC 3588 [8].

The parameters from the PC may comprise the UE IP address in either the Framed-IP-Address AVP or the Framed-IPv6-Prefix AVP, PDN information in the Called-Station-Id AVP, user identity in the Subscription-Id AVP and domain Identity in the IP-Domain-Id AVP (3GPP TS 23.203 [2]). The AAR command may also comprise the application service information in the AF-Application-Identifier AVP (3GPP TS 29.214 [5]).

6.2.4 Study on location of XML-DIAMETER Protocol Converter

6.2.4.1 General

Part of the Study on XML based access of AF to the PCRF in CP-130617 approved at CT#59 (Vienna, March 2013) is to investigate the adding of a "*protocol converter (as a stand-alone entity) between PCRF and AF*".

This paper proposes some possibilities where this Protocol Converter might reside and discusses the advantages and disadvantages of the proposed locations.

6.2.4.2 Locating the Protocol Converter

6.2.4.2.1 Option A: Protocol Converter within 3GPP PLMN

In this option the protocol converter is within the 3GPP PLMN.

In this option it is possible that:-

- there is one logical Protocol Converter per PLMN, see figure 6.2.4.2.1.1;
- there is one logical Protocol Converter per Diameter realm, see figure 6.2.4.2.1.2;

Editor's note: The logical Protocol Converter may consist of multiple entities.

Editor's note: It is FFS whether CT3 intends to study other possibile locations of the protocol converter within a 3GPP PLMN.

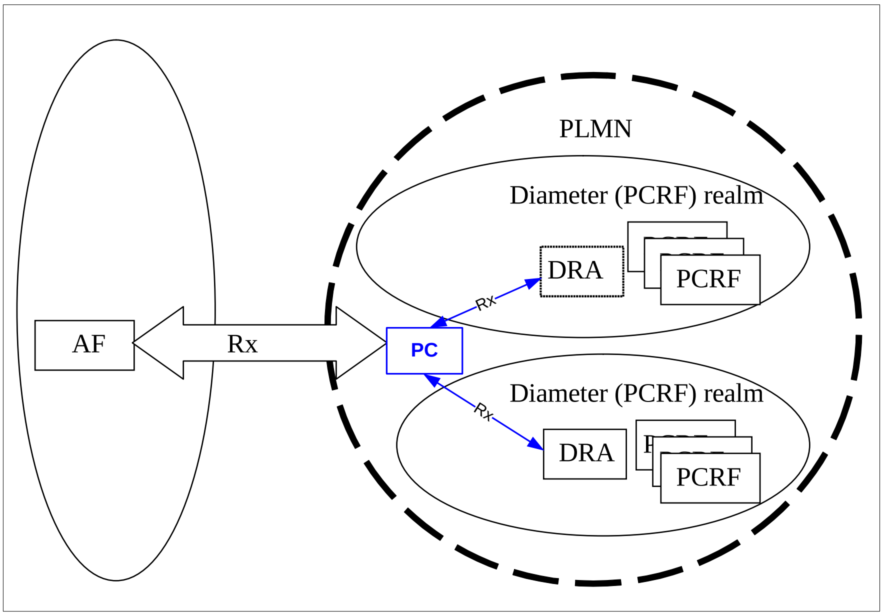

Figure 6.2.4.2.1.1: Protocol Converter placed within 3GPP PLMN, one logical Protocol Converter to a PLMN

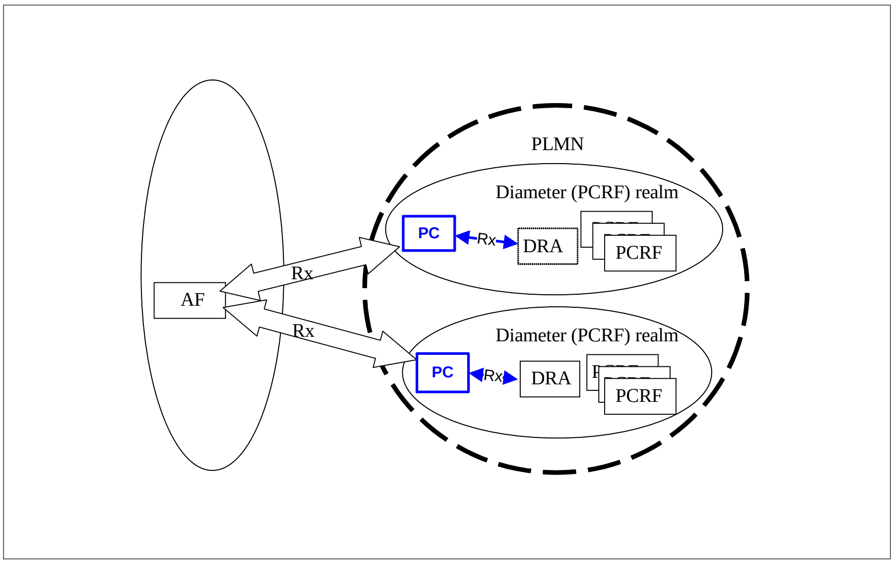

Figure 6.2.4.2.1.2: Protocol Converter placed within 3GPP PLMN, one logical Protocol Converter per Diameter realm

With Option A, there are the following *Pros* and *Cons*.

Pros	*Cons*
• No change to 3GPP architecture • Scoped and worked on by 3GPP to 3GPP domain • Quick and easy adoption by 3rd party applications providers • Standardised version of protocol conversion • Operator ease of managing 3rd party providers	• impact to CT3 specifications, total impacts yet to be analysed and determined

Editor's note: The *pros and cons* given in table above are incomplete and requires further studies.

6.2.4.2.2 Option B: Protocol Converter within AF (3rd party) server domain

In this option, the protocol converter is placed within the AF domain, see Figure 6.2.4.2.2.

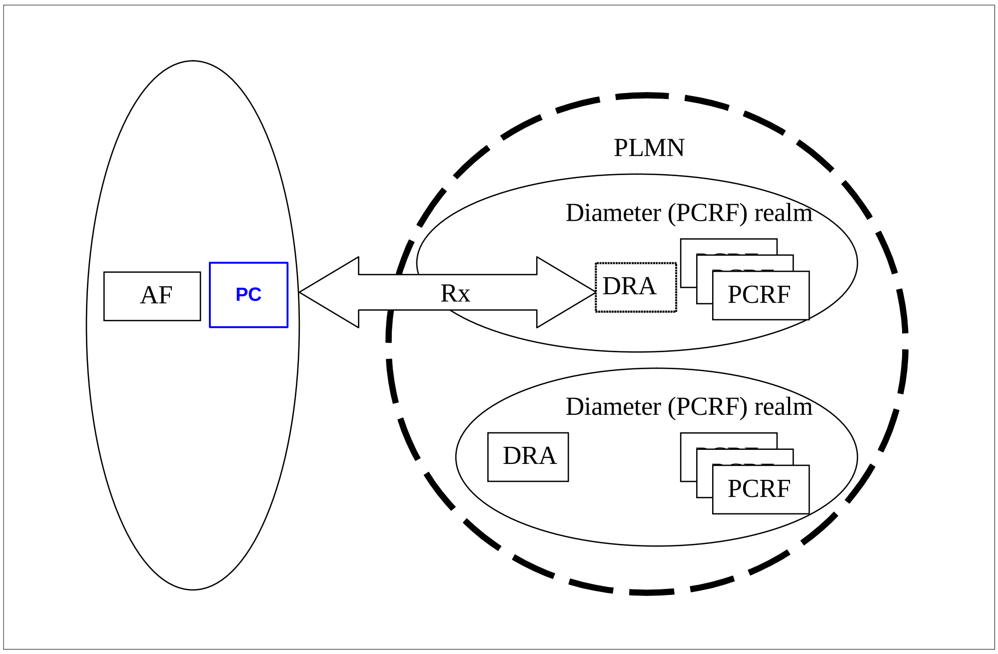

Figure 6.2.4.2.2: Protocol Converter within 3rd party server domain

With Option B, there are the following *Pros* and *Cons*.

Pros	*Cons*
• No likely impacts to CT3 interfaces and specifications	• Defeats the purpose of 'opening' the Rx to 3rd part application providers familiar with XML • Opens up many (non-standardised) variants of protocol conversion • Increases difficulty for operator's to manage 3rd party providers • Fragments market

Editor's note: The *pros and cons* given in table above is incomplete and requires further studies.

6.2.4.2.3 Option C: Protocol Converter between AF domain and 3GPP domain

For this option, the protocol converter is independently placed to bridge between the AF and 3GPP domain, see Figure 6.2.4.2.3.

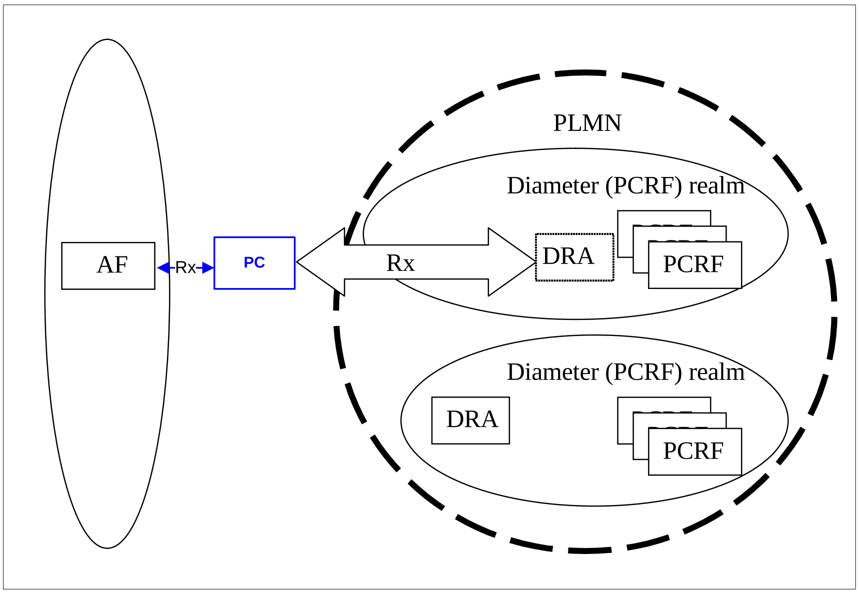

Figure 6.2.4.2.3: Protocol Converter bridging AF and 3GPP

With Option C, there are the following *Pros* and *Cons*.

Pros	*Cons*
• No likely impact to CT3 interfaces and specifications • Facilitate use of XML for 3rd part providers	• Opens up many (non-standardised) variants of protocol conversion • Increases difficulty for operator's to manage 3rd party providers • Fragments market

Editor's note: The *pros and cons* given in table above is incomplete and requires further studies.

6.3 PCRF web server

In this architecture the PCRF looks like an application server from AF point of view. One of the transport solutions, which are described in section 7.1 may be used to exchange complex information between the AF and the PCRF. The Rx related content may be presented by one of the application level languages described in section 7.2.

PCRF and AF are connected to each other without any protocol converter.

Editor’s Note: Description of the solution in which the PCRF is directly connected to the 3^{rd} party web application.

6.4 Evaluation

	Protocol Converter	PCRF Web-server
Complexity	No additional complexity of the PCRF: with this architecture, the PCRF still sees a Diameter Rx interface.	Potential more complexity for the PCRF to manage both Diameter and XML based interfaces in his network.
Extensibility	No need for the PCRF to support the same functionalities over different protocols.	More difficult to evolve since it requires that the PCRF supports two protocols over the same reference point. Limitations in the current deployments (e.g. roaming and routing limitations) would make the evolution more difficult.
Consistency	The "consistency" issue is handled at the PC level, no additional task for the PCRF (the PC is in charge of securing translation for messages generated by AF (e.g. correct parameter mapping from XML to Diameter)).	PCRF should check the consistency of the provided information.
Limitation for PCC deployment	Multiple PCRFs can be deployed.	Only one PCRF can be deployed (DRA cannot be used to find the PCRF).

7 Rx protocols for 3rd party web application developers

7.1 Transport solutions for Rx messages

Rx related information has to be transported between the AF and the PC or the PCRF web server. It can be transported by different protocols as content on top of them.

RESTful HTTP and SOAP are interpreted as a transport protocols. Bidirectional communication has to be possible between the PCRF and the AF.

7.1.1 Transport protocols

7.1.1.1 RESTful HTTP

Representational State Transfer (REST) style was developed by W3C Technical Architecture Group (TAG) in parallel with HTTP/1.1, based on the existing design of HTTP/1.0. The World Wide Web represents the largest implementation of a system conforming to the REST architectural style. REST exemplifies how the Web's architecture emerged by characterizing and constraining the macro-interactions of the four components of the Web, namely origin servers, gateways, proxies and clients, without imposing limitations on the individual participants. As such, REST essentially governs the proper behaviour of participants.

REST defines a set of architectural principles, which can be used to design Web services. REST is described in chapter 5 of Fielding's dissertation "Representational State Transfer (REST)" [9] in detail. Therein it is described as an architectural style consisting of the set of constraints applied to elements within the architecture that better reflects the desired properties of a modern Web architecture.

The following constraints can be listed

- Client-Server
- Stateless
- Cacheable
- Layered system
- Code on demand
- Uniform interface

The goals of REST are

- scalability of component interactions
- generality of interfaces
- independent deployment of components
- reduction of latency, enforce security and encapsulate legacy systems

REST is based on the strict usage of the following main HTTP methods:

- POST: May create a resource state
- PUT: May modifiy a resource state
- GET: May query a resource state
- DELETE: May delete a resource state

There is no restriction concerning to the information that the HTTP methods may transport in comparison to the definition of HTTP in RFC 2616 [10]. Therefore, XML or JSON based Rx information can be transported easily by RESTful methods.

REST facilitates the transaction between web servers by allowing loose coupling between different services. REST is less strongly typed than its counterpart, SOAP. The REST language uses nouns and verbs, and has an emphasis on readability. Unlike SOAP, REST does not require XML parsing and does not require a message header to and from a service provider. This ultimately uses less bandwidth. REST error-handling also differs from that used by SOAP.

RestFul HTTP may be used to design an easy to use interface from application developer point of view.

7.1.1.2 SOAP

SOAP is a W3C recommendation. SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services. It relies on for its message format, and uses other Application Layer protocols, which may be the Hypertext Transfer Protocol (HTTP) or the Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

SOAP can construct the foundation layer of the web service protocol stack for providing a basic messaging framework upon which web services can be built. This XML based protocol consists of three parts:

- an envelope, which defines what is in the message and how to process it
- a set of encoding rules for expressing instances of application-defined data types
- a convention for representing procedure calls and responses.

As an example of how SOAP procedures can be used in Rx interface, a SOAP message is sent to the web site (i.e. the PCRF in this scenario) in which web services are enabled, with the parameters needed for multiple kinds of requirements. The PCRF then returns an XML-formatted document with the result data. The data are returned in a standardized machine-parsable format.

The disadvantages of SOAP is its high complexity. It can be slow due to the complex XML format. Therefore the complexity may be too high for the Rx application.

7.1.2 HTTP and bidirectional communication

7.1.2.1 Long polling and streaming

Standard HTTP was not defined for bidirectional communication. It uses short polling. The client sends regular request and in case of no data the server returns an empty response.

The server cannot initiate a connection with a client and cannot send unrequested HTTP response to the client. Therefore, clients need to poll the server periodically. This consumes bandwidth and is not efficient due to reduction of responsiveness of the application.

Solutions for mechanism that work within the current scope of HTTP 1.0/1.1 are HTTP long polling or HTTP streaming.

With HTTP long polling the server responds to a request when a particular event, status or timeout has occurred. The server defers the response and holds open a long poll request. The server is able to asynchronously initiate communication.

HTTP streaming keeps a request indefinitely open and never terminates the request or close the connection after server pushes data to the client.

7.1.2.2 Two connections

In case that the full control of the Diameter Rx application shall be used the PC or the PCRF web server must be able to sent notifications to the AF. This could be reached by the definition of a notification URL to which the notification can be send. In this sense the endpoints can be interpreted as a server or a client in dependence of the action required. The notification URL can be created during session set-up. The following shows an example dialogue in which the notification URL is sent to the PC or the PCRF using the HTTP POST method:

HTTP POST https://pcrfserver/rxapplication/fullcontrol/settings

```
<settings>
 <notificationURL>http://notificationserver/rxnotify</notificationURL>
</settings>
```

The settings dialogue may contain further parameters in order to prepare the session. The session itself is set-up by a further HTTP POST dialogue, which may contain a complete Rx message mapped to XML, for example.

Editor's Note: All issues that have to be addressed in relation to the transport of messages may be described in this section. For example, HTTP issues concerning to server initiated communication have to be analysed.

7.2 Application level protocols and data structures

Editor's Note: Any application level protocol or data structure, which could be used to transport Rx data, may be analysed in this section.

7.2.1 Proposal a: XML

The Extensible Markup Language (XML) is a generally accepted code of practice developed by W3C. With XML it is possible to describe complex content in a hierarchical model and it is a format to exchange information between different applications and platforms. It does not have a defined tag list and it is expandable. Tags may be defined by Document Type Definitions (DTDs) or XML schemas.

A part of content of an AA-Request (AAA) command may look like as reflected in the following figure:

```
<?XML version="1.0" encoding="UTF-8"?>
<!-- This is an example of some AAR command content -->
<AAR command>
    <IP-Domain-Id IP-Domain-ID="..."></IP-Domain-Id>
    <AF-Application-Identifier AF-Application-Identifier="..."></AF-Application-Identifier>
    <Media-Component-Description>
        <Media-Component-Number Media-Component-Number="..."></Media-Component-Number>
        <Media-Sub-Component>
            <Flow-Number Flow-Number="..."></Flow-Number>
            <Flow-Description Flow-Description="..."></Flow-Description>
            <Flow-Status Flow-Status="..."></Flow-Status>
            <Flow-Usage Flow-Usage="..."></Flow-Usage>
            <Max-Requested-Bandwidth-UL Max-Requested-Bandwidth-UL="..."></Max-Requested-Bandwidth-UL>
            <Max-Requested-Bandwidth-DL Max-Requested-Bandwidth-DL="..."></Max-Requested-Bandwidth-DL>
            <AF-Signalling-Protocol AF-Signalling-Protocol="..."></AF-Signalling-Protocol>
        </Media-Sub-Component>
        <AF-Application-Identifier AF-Application-Identifier="..."></AF-Application-Identifier>
        <Media-Type Media-Type="..."></Media-Type>
        <Max-Requested-Bandwidth-UL Max-Requested-Bandwidth-UL="..."></Max-Requested-Bandwidth-UL>
        <Max-Requested-Bandwidth-DL Max-Requested-Bandwidth-DL="..."></Max-Requested-Bandwidth-DL>
        <Min-Requested-Bandwidth-UL Min-Requested-Bandwidth-UL="..."></Min-Requested-Bandwidth-UL>
        <Min-Requested-Bandwidth-DL Min-Requested-Bandwidth-DL="..."></Min-Requested-Bandwidth-DL>
        <Flow-Status Flow-Status="..."></Flow-Status>
        <Reservation-Priority Reservation-Priority="..."></Reservation-Priority>
        <RS-Bandwidth RS-Bandwidth="..."></RS-Bandwidth>
        <RR-Bandwidth RR-Bandwidth="..."></RR-Bandwidth>
        <Codec-Data Codec-Data="..."></Codec-Data>
    </Media-Component-Description>
    <Service-Info-Status Service-Info-Status="..."></Service-Info-Status>
    <AF-Charging-Identifier AF-Charging-Identifier="..."></AF-Charging-Identifier>
    <SIP-Forking-Indication SIP-Forking-Indication="..."></SIP-Forking-Indication>
    <Specific-Action Specific-Action="..."></Specific-Action>
    <!-- Further Tags of the AAR command -->
</AAR command>
```

Figure 7.2.1.1: AAR command (incomplete example) as XML message

This example only shows independent of whether all information is meaningful or something is missing that XML can be used to transport already defined data structures between the AF and the PCRF.

The XML schema (XSD) may be used to define the conditions of validity for the XML document. Thus it defines the data types, elements and attributes that are allowed to be used in the document.

7.2.2 Proposal b: JSON

JSON or JavaScript Object Notation, is a text-based open standard which is designed for human-readable data interchange format. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for many languages.

The JSON format is described in IETF RFC 4627[7]. The official Internet media type for JSON is application/json. The JSON filename extension is .json.

The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML.

An example of a JSON based AA Request (AAR) command is shown in the following:

```
{ “AAR”:
{
 “Auth-Application-Id”:  “Value1”,
“Session-Id”: “Value2”,
“Origin-Host”: “Value3”
“Origin- Realm”: “Value4”
“Destination- Host”: “Value5”
“Destination-Realm”: “Value6”
“AF-Application-Identifier”: “Value7”

“Media-Component-Description”:
{
“Optional”: “YES”
  “Media-Component-Number”: “Value8”,
  “AF-Application-Identifier”: Value9,
```

```
      "Media-Sub-Component":
      {
         "Flow-Number": "Value10",
         "Flow-Status": "Value11",
         "Flow-Usage": "Value12",
         "Max-Requested-Bandwidth-UL": Value13,
         "Max-Requested-Bandwidth-DL": Value14,
         "AF-Signalling-Protocol": "Value15",
         "Flow-Description: "Value16",
      }
      }
   }
```


7.2.3 Data type mapping

XML is a technology-independent language of data format as Diameter, the main data types are represented by the two languages as in figure 7.2.3.1.

	XML	Diameter
String	String	OctetString
Integer	Int/NegativeInteger	Integer/Unsigned
Float	Float	Float
Enumeration	Enumerations	Enumerated
Group	Compound types	Grouped

Figure 7.2.3.1: Data types in XML and Diameter.

The data types of Float, Enumeration and Group have the same definition in both languages. And the Integer of XML is a 32-bit signed while the Integer of Diameter is a 32-bit unsigned. The String of XML is just normal string, and for Diameter, OctetString specifies octets of binary or textual information.

Different from XML and Diameter, JSON is a data-interchange format. It is capable of representing numbers, Booleans, strings, `null`, and arrays (ordered sequences of values) and objects (string-value mappings) composed of these values (or of other arrays and objects). It doesn't natively represent more complex data types like functions, regular expressions, dates, and so on. If the developers need to preserve such values, they can transform values since they are serialized, or prior to de-serialization, to enable JSON to represent additional data types.

7.3 Session management

7.3.1 General

For the Diameter based Rx interface, both the PCRF and the AF can initiate an Rx procedure. However, XML based protocols such as SOAP or RESTful are a one-way protocols, which means only one network element can initiate a Rx procedure. In this case, the AF acts as the client and the PCRF acts as the server and the interaction request can only be initiated by the client AF.

In the 3GPP TS 29.214 [5], when the traffic plane events happen, PCRF needs to inform the AF by ASR or RAR messages, and then the AF takes corresponding actions. However, in context of XML based Rx, the behaviour that the server (i.e. PCRF) sends requests to the client (i.e. AF) doesn't fulfil the rules of SOAP or is treated as un-RESTful. Therefore, the traffic plane events reporting mechanism should be adapted to XML based protocols.

The keep-alive technology should be applied for messages pushing from the PCRF. Depending on different pushing mechanism, there are several candidate solutions to traffic plane events reporting.

7.3.2 HTTP streaming

For the stream based solution, the PCRF (or the PC in case the PC is involved) and the AF are connected by HTTP based protocol (such as SOAP), and the stream based keep-alive technology is applied for messages pushing from the PCRF/PC. The client sends a request firstly, and then the server replies multiple messages as responses. Those responses for the same request can be treated as a stream without terminating the HTTP connection. The responses will stop when the connection is released. The one request-multiple responses mode is in favour of the scenario that the

PCRF can detect multiple traffic plane events and report them to the AF under one HTTP connection without requesting it for multiple times. The procedure is shown in figure 7.3.1:

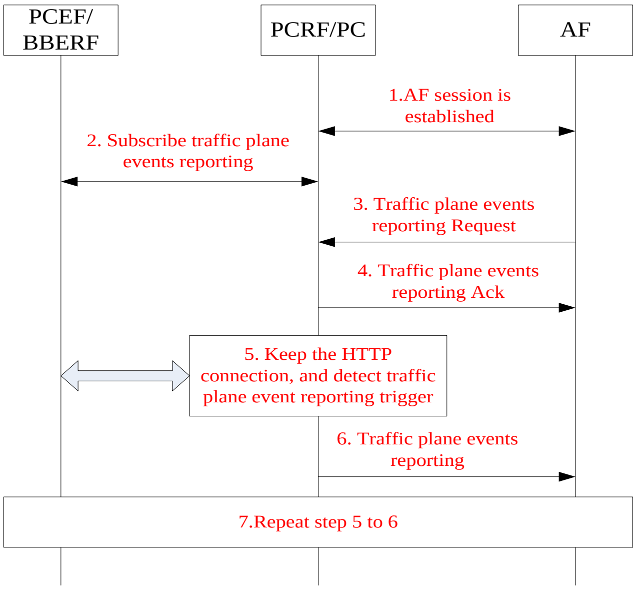

Figure 7.3.1. Procedure of stream based solution

In case that the keep-alive stream based AF session has been established, the AF initiates traffic plane events reporting procedure by sending a Traffic plane event reporting request to the PCRF/PC. The PCRF/PC acknowledges the AF after it receives the request. The PCRF/PC subscribes to the corresponding event and waits for the occurrence of corresponding traffic plane events. When the PCRF/PC detects one of the traffic plane events, it responds corresponding parameters to the AF. After that, the PCRF/PC will keep the connection and repeat the steps 5 to 6 of traffic plane event reporting. In the end, the connection is released when the AF session terminates.

7.3.3 HTTP long polling

For the solution of long-polling, the PCRF (or the PC in case the PC is involved) and the AF are connected by HTTP based protocol (such as SOAP), and the long-polling based keep-alive technology is applied for PCRF messages pushing. The client sends a request firstly, and the server doesn't reply the request immediately, but keeps the connection for a while and then responds until the preconfigured events happened. The response will stop when the connection is released. The procedure is shown in figure 7.3.2:

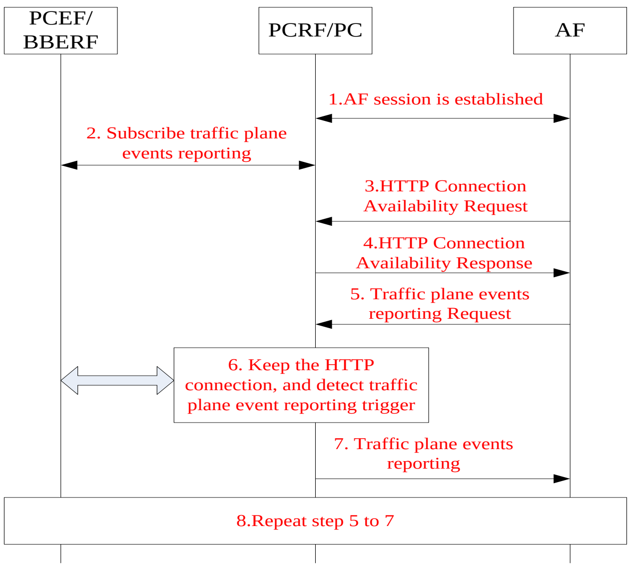

Figure 7.3.2. Procedure of long-polling based solution.

In case that the keep-alive based AF session has been established and the AF has subscribed to the required traffic plane events, the PCRF/PC will subscribe to the corresponding traffic plane events in the PCEF/PCRF. The AF optionally sends HTTP Connection Availability Request to the PCRF/PC, to check whether the long-poulling based HTTP connection between the AF and the PCRF/PC is available . The PCRF/PC responds the AF by a HTTP Connection Availability Response message. The AF initiates traffic plane events reporting procedure by sending a Traffic plane events reporting Request message to the PCRF/PC, and the PCRF/PC keeps the connection. When the PCRF/PC detects one of the traffic plane events, it reports corresponding event to the AF. After that, the steps 5 to 7 repeat until the connection is released when the AF session terminates.

NOTE: The mechanism of stream based solution is the same as "HTTP Streaming" mentioned in IETF RFC 6202 [6] section 3, and the mechanism of long-polling based solution is the same as "HTTP Long Polling" mentioned in IETF RFC 6202 [6] section 2.

7.3.4 Two TCP connections

The PCRF/PC and AF both support the HTTP client and HTTP server role. There are two TCP connections between the PCRF/PC and the AF. One is initiated by the AF, while the other is initiated by the PCRF/PC. If the AF establishes an AF session (i.e. initiates the initial provisioning of session information), the AF sends the HTTP request to the PCRF/PC. The HTTP request can re-use the existing TCP connection. The AF assigns an AF session id which is used to identify the AF session. The PCRF/PC stores the AF session id and service information, and then responds to the AF. After this transaction, the HTTP connection can be released but the TCP connection is kept. When the AF modifies the AF session (i.e. initiates modification of session information), the AF sends an HTTP request including the AF session id assigned in the AF session establishment procedure. The PCRF/PC updates stored session information according to the AF session id. When the PCRF/PC reports the traffic plane event corresponding to the AF session, the PCRF/PC sends an HTTP request to the AF. The HTTP request can re-use current existing TCP connections. The request includes the AF session id assigned by the AF and stored by the PCRF/PC during AF session establishment.

7.4 Traffic plane events

7.5 Evaluation

7.5.1 Comparison among XML based technologies

The relationship among several XML based technologies is described as follows.

XML	XML/JSON
SOAP	REST
TCP	TCP

Figure 7.5.1.1: The relationship among XML based technologies

XML or JSON is a technology-independent data format which represents the information exchanged between two entities in the Internet. SOAP protocol and REST architectural style are technologies which are used for exchanging the information formatted by XML. When constructing a XML based communication solution, SOAP is always tightly coupled with XML while both XML and JSON can be applied over REST architectural style.

The table below have a summary of comparison among SOAP, REST, XML and JSON. The following aspects are recommended when evaluating XML based technologies over Rx.

- Readability.
- Complexity of data processing in both the client and the server.
- Extensibility.
- Performance.
- Security.

	SOAP	REST	XML	JSON
Readability	Complex	REST is easy and understandability.		Simple and almost no modification of the client.
Complexity of data processing in both the client and the server	Based on any protocol such as HTTP, HTTPS, SMTP (Simple Mail Transfer Protocol), and even JMS (Java Messaging Service).	REST interfaces are much easier to design and implement	Complex	Simpler than XML.
Extensibility	Tightly coupled with XML	High expandability	-	-
Performance	Inefficient	High response rate depending on the cache.		Less consumption and more powerful.
Security	Safe		Safer than JSON.	Less safe than XML.

8 Charging

9 Security

10 Roaming and routing

11 Conclusion and recommendation

11.1 Impacts on the PCC architecture

11.2 Impacts on existing 3GPP specifications

11.3 Recommendation for an XML based access of AF to the PCRF

Annex A (informative): Example – XML schema (example 1)

A.1 Example - XML schema (example 1)

Editor's note: The following example XML schema and the analysis leading to it needs further analysis in CT3 and has not been agreed as technically correct. What is given herewith can be considered FFS.

A.1.1 Analysis of Diameter Data Types and XML Schema Data Types

For this anaylsis the Diameter AAR message is considered. From 29.214 (v11.8.0), subclause 5.6.1 the following is the AAR message format:-

```
<AA-Request> ::= < Diameter Header: 265, REQ, PXY >
                 < Session-Id >
                 { Auth-Application-Id }
                 { Origin-Host }
                 { Origin-Realm }
                 { Destination-Realm }
                 [ Destination-Host ]
                 [ IP-Domain-Id ]
                 [ AF-Application-Identifier ]
                *[ Media-Component-Description ]
                 [ Service-Info-Status ]
                 [ AF-Charging-Identifier ]
                 [ SIP-Forking-Indication ]
                *[ Specific-Action ]
                *[ Subscription-Id ]
                *[ Supported-Features ]
                 [ Reservation-Priority ]
                 [ Framed-IP-Address ]
                 [ Framed-IPv6-Prefix ]
                 [ Called-Station-Id ]
                 [ Service-URN ]
                 [ Sponsored-Connectivity-Data ]
                 [ MPS-Identifier ]
                 [ Rx-Request-Type ]
                *[ Required-Access-Info ]
                 [ Origin-State-Id ]
                *[ Proxy-Info ]
                *[ Route-Record ]
                *[ AVP ]
```

Following table A.1.1 consider the individual data types of the AVPs of this AAR command and considers the corresponding XML data type or the closest possible XML data type.

Table A.1.1: Data types of AAR command

AVPs	AVP CODE	DIAMTER DATA TYPE	XML SCHEMA DATA TYPE
SessionId	263	UTF8String	String
Auth Application	258	Unsigned32	UnsignedInt
OriginHost	264	DiameterIdentity	String
OriginRealm	296	DiameterIdentity	String
DestinationRealm	283	DiameterIdentity	String
DestinationHost	293	DiameterIdentity	String
IP-Domain-ID	537	OctetString	String
AF Application Identifier	504	OctetString	String
Media Component description	517	Grouped	Group
Service Info Status	527	Enum	Int
AF charging identifier	505	OctetString	String
SIP-Forking-Indication	523	Enum	Int
Specific-Action	513	Enum	Int
Subscription-ID	443	Grouped	Group
****Supported Features***	628	Grouped	Group
Reservation-Priority	458	Enum	Int
Framed-IP-Address	8	OctetString	String
Framed-IPv6-Prefix	97	OctetString	String
Service-URN	525	OctetString	String
Sponsored connectivity data	530	Grouped	Group
MPS Identifier	528	OctetString	String
Rx-Request-Type	533	Enum	Int
***Required Access Info**	536	Enum	Int
Origin State ID	278	Unsigned32	unsignedInt
Proxy Info	284	Grouped	Group
Route-Record	282	DiameterIdentiy	String

In table A.1.1 there are certain AVPs which are of data type group, denoting embedded AVPs. There are also certain data types that do not match directly to corresponding XML data types or that such XML data types do not exist. These are considered and XML data types matches proposed in the following table A.1.2 to table A.1.8

Table A.1.2: Media component description AVP

MEDIA COMPONENT DESCRIPTION AVP	AVP CODE	DIAMETER DATA TYPE	XML SCHEMA DATA TYPE
Media component number	518	Unsigned32	unsignedInt
*Media sub component	519	Grouped	Group
AFAppIdentifier	504	OctetString	String
MediaType	520	Enum	Int
MaxRequestedBWUL	516	Unsigned32	UnsignedInt
MaxRequestedBWDL	515	Unsigned32	UnsignedInt
FlowStatus	511	Enum	Int
ReservationPriority	458	Enum	Int
RSBW	522	Unsigned32	UnsignedInt
RRBW	521	Unsigned32	UnsignedInt
*CodecData	524	OctetString	String

Table A.1.3: Subscription-ID AVP

SUBSCRIPTIONID AVP	AVP CODE	DIAMETER DATA TYPE	XML SCHEMA DATA TYPE
Sub-ID-Type	450	Enum	Int
Sub-ID-Data	444	UTF8String	String

Table A.1.4: Supported feature AVP

SUPPORTED FEATURE AVP	AVP CODE (628)	DIAMETER DATA TYPE	XML SCHEMA DATA TYPE
Feature List ID		Unsigned32	UnsignedInt
Feature List		Unsigned32	UnsignedInt

Table A.1.5: Sponsored connectivity data AVP

SPONSORED CONNECTIVITY DATA AVP	AVP CODE	DIAMETER DATA TYPE	XML SCHEMA DATA TYPE
Sponsor Identity	531	UTF8String	String
App Service Provider Identity	532	UTF8String	String
Granted Service Unit	431	Grouped	Group
Used Service Unit	446	Grouped	Group

Table A.1.6: Granted service unit AVP

GRANTED SERVICE UNIT AVP	AVP CODE	DIAMETER DATA TYPE	XML SCHEMA DATA TYPE
Tariff Time Change	451	Time	Time
CC-Time	420	Unsigned32	UnsignedInt
CC-Money	413	Grouped	Group
Total Octets	421	Unisgned64	UnsignedInt
Input Octets	412	Unisgned64	UnsignedInt
Output Octets	414	Unisgned64	UnsignedInt
Service Specific Units	417	Unisgned64	UnsignedInt

Table A.1.7: Used service unit AVP

USED SERVICE UNIT AVP	AVP CODE	DIAMETER DATA TYPE	XML SCHEMA DATA TYPE
Tariff Change Usage	452	Enum	Int
CC-Time	420	Unsigned32	UnsignedInt
CC-Money	413	Grouped	Group
Total Octets	421	Unisgned64	UnsignedInt
Input Octets	412	Unisgned64	UnsignedInt
Output Octets	414	Unisgned64	UnsignedInt
Service Specific Units	417	Unisgned64	UnsignedInt

Table A.1.8: Supported feature AVP

SUPPORTED FEATURE AVP	AVP CODE (628)	DIAMETER DATA TYPE	XML SCHEMA DATA TYPE
Unit Value	445	Decimal	Decimal
Currency Code	425	Unsigned32	UnsignedInt

A.1.2 "AVP Parameters" group

Diameter protocol is a message based protocol with two types of messages i.e., request message and answer message.

A diameter AVP (Attribute Value Pair) is a basic unit inside the diameter message that carries the data. There must be at least one AVP inside the diameter message. AVP code, AVP flags and AVPLength fields are mandatory fields for every AVP in an Rx message, see Figure 7.x.y.1.1

AVP Code	
AVP Flags	AVP Length
Vendor Id (Optional)	
Data	

Figure A.1.2.1 – Format of a diameter AVP basic unit

For these recurring three fields, one could group these as an "AVP Parameters" in XML schema equivalent messages and reference that in every AVP element in the XML Schema**.** This will optimize the XML schema representation for any AVP in the diameter protocol. The XML representation of this "AVP Parameters" type is as given below:

```
<xs:element name="AVPParameters">
        <xs:complexType>
                <xs:sequence>
                        <xs:element name="AVPCode" type="xs:int"/>
                        <xs:element name="AVPFlags" type="xs:byte"/>
                        <xs:element name="AVPLength" type="xs:int"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

This "AVP Parameters" group (given above) is referenced in every AVP in the diameter message using the reference attribute. An example of the usage of AVP parameter group is given below:

```
<xs:element name="SessionIDAVP">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element name="Data" type="xs:string"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

The above example is for the SessionID AVP included in the AARequest message (Rx message).The AVP parameter is referenced using the XML ref attribute which optimising the XML representation for the Rx messages instead of all the three AVP field being explicitly include for every AVP element represented in the XML schema.

A.1.3 XML schema for Diameter AAR applying "AVP Parameters" group

Applying the logic in subclauses 7.x.y.1 and 7.x.y.2, the following XML schema for the AAR message is derived.

NOTE: The following XML schema has been check and does compile without errors.

A.1.3.1 Diameter Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |                 Message Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| command flags |                  Command-Code                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Application-ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Hop-by-Hop Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      End-to-End Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AVPs ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

<XML equivalent>

```
<xs:element name="DiameterHeader">
    <xs:complexType>
        <xs:sequence>
            <xs:element fixed="1" name="Version" type="xs:byte"/>
            <xs:element name="MessageLength" type="xs:integer"/>
            <xs:element fixed="255" name="CommandFlags" type="xs:unsignedByte"/>
            <xs:element fixed="265" name="CommandCode" type="xs:integer"/>
            <xs:element name="ApplicationID" type="xs:integer"/>
            <xs:element name="HopToHopIdentifier" type="xs:int"/>
            <xs:element name="EndToEndIdentifier" type="xs:int"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

A.1.3.2 AVP Parameter Group

<XML equivalent>

```
<xs:element name="AVPParameters">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="AVPCode" type="xs:int"/>
            <xs:element name="AVPFlags" type="xs:byte"/>
            <xs:element name="AVPLength" type="xs:int"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

AVP code, AVP flags and AVPLength fields are mandatory fields for every AVP in an Rx message. Hence in the XML equivalent message these three fields are grouped as an AVP parameter group and referenced in every AVP element in the XML Schema.

A.1.3.3 Session ID AVP

<XML equivalent>

```
<xs:element name="SessionIDAVP">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="AVPParameters"/>
            <xs:element name="Data" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

A.1.3.4 Auth-Application ID AVP

<XML equivalent>

```
<xs:element name="AuthApplicationIDAVP">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="AVPParameters"/>
            <xs:element name="Data" type="xs:unsignedInt"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

A.1.3.5 Origin Host AVP

<XML equivalent>

```
<xs:element name="OriginHostAVP">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="AVPParameters"/>
            <xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

A.1.3.6 Origin Realm AVP

<XML equivalent>

```
<xs:element name="OriginRealmAVP">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="AVPParameters"/>
            <xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

A.1.3.7 Destination Realm AVP

<XML equivalent>

```xml
<xs:element name="DestinationRealmAVP">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

A.1.3.8 Destination Host AVP

<XML equivalent>

```xml
<xs:element name="DestinationHostAVP">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

A.1.3.9 IP Domain ID AVP

<XML equivalent>

```xml
<xs:element name="IP-Domain-ID">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

A.1.3.10 AFAppIdentifier AVP

<XML equivalent>

```xml
<xs:element name="AFApplicationIdentifierAVP">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

A.1.3.11 Media Component Description AVP

<XML equivalent>

```
<xs:element maxOccurs="unbounded" name="MediaComponentDescriptionAVP">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data">
				<xs:complexType>
					<xs:group ref="MediaGroup"/>
				</xs:complexType>
			</xs:element>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

This AVP is of group type. The mapping of this group type in XML schema involves defining a Media Group type defined as follows:

```
<xs:group name="MediaGroup">
	<xs:sequence>
		<xs:element name="MediaComponentNumberAVP">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:unsignedInt"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
		<xs:element name="MediSubComponentAVP">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:string"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
		<xs:element name="AFApplicationIdentifierAVP">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
```

```
<xs:element name="MediaTypeAVP">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data" type="xs:int"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
<xs:element name="MediaRequestedBWUL">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data" type="xs:unsignedInt"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
<xs:element name="MediaRequestedBWDL">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data" type="xs:unsignedInt"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>

<xs:element name="FlowStatus">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data" type="xs:int"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
<xs:element name="ReservationProirityAVP">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data" type="xs:int"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
<xs:element name="RSBW">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data" type="xs:unsignedInt"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

```
                <xs:element name="CodecData">
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="AVPParameters"/>
                                        <xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>
        </xs:sequence>
</xs:group>
```

A.1.3.12 Service Info Status AVP

<XML equivalent>

```
<xs:element name="ServiceInfoStatusAVP">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element name="Data" type="xs:int"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

A.1.3.13 AF Charging Identifier AVP

<XML equivalent>

```
<xs:element name="AFChargingIdentifierAVP">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

A.1.3.14 SIP Forking Indication AVP

<XML equivalent>

```
<xs:element name="SIPForkingIndicationAVP">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element name="Data" type="xs:int"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

A.1.3.15 Specific Action AVP

<XML equivalent>

```
<xs:element maxOccurs="unbounded" name="SpecificActionAVP" nillable="false">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data" type="xs:int"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

A.1.3.16 Subscription ID AVP

<XML equivalent>

```
<xs:element maxOccurs="unbounded" name="SubcriptionIDAVP">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data">
				<xs:complexType>
					<xs:group ref="SubscriptionGroup"/>
				</xs:complexType>
			</xs:element>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

This AVP is of group type. The mapping of this group type in XML schema involves defining a SubscriptionGroup type defined as follows:

```
<xs:group name="SubscriptionGroup">
	<xs:sequence>
		<xs:element name="SubsccriptionIDTypeAVP" nillable="false">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:int"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
		<xs:element name="SubscriptionIDDataAVP">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:string"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
	</xs:sequence>
</xs:group>
```

A.1.3.17 Supported Features AVP

<XML equivalent>

```
<xs:element maxOccurs="unbounded" name="SupportedFeatures">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data">
				<xs:complexType>
					<xs:group ref="SupportedFeaturesGroup"/>
				</xs:complexType>
			</xs:element>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

This AVP is of group type. The mapping of this group type in XML schema involves defining a SupportedFeaturesGroup type defined as follows:

```
<xs:group name="SupportedFeaturesGroup">
	<xs:sequence>
		<xs:element name="VendorID">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:int"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
		<xs:element name="FeatureListID">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:unsignedInt"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
		<xs:element name="FeatureList">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:unsignedInt"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
	</xs:sequence>
</xs:group>
```

A.1.3.18 Reservation Priority AVP

<XML equivalent>

```
<xs:element name="ReservationProirityAVP">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element name="Data" type="xs:int"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

A.1.3.19 Framed IP address AVP

<XML equivalent>

```
<xs:element name="FramedIPAddressAVP">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

A.1.3.20 Framed IPv6 prefix AVP

<XML equivalent>

```
<xs:element name="FramedIPv6PnameixAVP">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

A.1.3.21 Service URN AVP

<XML equivalent>

```
<xs:element name="ServiceURNAVP">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

A.1.3.22 Sponsored Connectivity Data AVP

<XML equivalent>

```
<xs:element name="SponsoredConnectivityData">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element name="Data">
                                <xs:complexType>
                                        <xs:group ref="SponsoredConnectivityGroup"/>
                                </xs:complexType>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

This AVP is of group type. The mapping of this group type in XML schema involves defining a SponsoredConnectivityGroup type defined as follows:

```
</xs:group>
<xs:group name="SponsoredConnectivityGroup">
        <xs:sequence>
                <xs:element name="SponsorIdentiy">
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="AVPParameters"/>
                                        <xs:element name="Data" type="xs:string"/>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>
                <xs:element name="ApplicationServiceProviderIdentifier">
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="AVPParameters"/>
                                        <xs:element name="Data" type="xs:string"/>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>
                <xs:element name="GrantedServiceUnit">
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="AVPParameters"/>
                                        <xs:element name="Data">
                                                <xs:complexType>
                                                        <xs:group ref="GrantedServiceGroup"/>
                                                </xs:complexType>
                                        </xs:element>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>

                <xs:element name="UserServiveUnit">
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="AVPParameters"/>
                                        <xs:element name="Data">
                                                <xs:complexType>
                                                        <xs:group ref="UserServiceGroup"/>
                                                </xs:complexType>
                                        </xs:element>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>
        </xs:sequence>
</xs:group>
```

The sponsoredConnectivityGroup has further level of nesting with the group data type which is grouped as GrantedServiceGroup and UserServiceGroup as defined below:

```
<xs:group name="GrantedServiceGroup">
        <xs:sequence>
                <xs:element name="TariffTimeChange">
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="AVPParameters"/>
                                        <xs:element name="Data" type="xs:time"/>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>
                <xs:element name="ccTime">
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="AVPParameters"/>
                                        <xs:element name="Data" type="xs:unsignedInt"/>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>
                <xs:element name="ccMoney">
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="AVPParameters"/>
                                        <xs:element name="Data">
                                                <xs:complexType>
                                                        <xs:group ref="MoneyGroup"/>
                                                </xs:complexType>
                                        </xs:element>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>

<xs:element name="TotalOctets">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element name="Data" type="xs:unsignedInt"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
<xs:element name="InputOctets">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element name="Data" type="xs:unsignedInt"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
<xs:element name="OutputOctets">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="AVPParameters"/>
                        <xs:element name="Data" type="xs:unsignedInt"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

```
                                        <xs:element name="ServiceSpecificUnits">
                                                <xs:complexType>
                                                        <xs:sequence>
                                                                <xs:element ref="AVPParameters"/>
                                                                <xs:element name="Data" type="xs:unsignedInt"/>
                                                        </xs:sequence>
                                                </xs:complexType>
                                        </xs:element>
                                </xs:sequence>
                        </xs:group>
                        <xs:group name="UserServiceGroup">
                                <xs:sequence>
                                        <xs:element name="TariffChangeUsage">
                                                <xs:complexType>
                                                        <xs:sequence>
                                                                <xs:element ref="AVPParameters"/>
                                                                <xs:element name="Data" type="xs:int"/>
                                                        </xs:sequence>
                                                </xs:complexType>
                                        </xs:element>
                                        <xs:element name="ccTime">
                                                <xs:complexType>
                                                        <xs:sequence>
                                                                <xs:element ref="AVPParameters"/>
                                                                <xs:element name="Data" type="xs:unsignedInt"/>
                                                        </xs:sequence>
                                                </xs:complexType>
                                        </xs:element>

                        <xs:element name="ccMoney">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element ref="AVPParameters"/>
                                                <xs:element name="Data">
                                                        <xs:complexType>
                                                                <xs:group ref="MoneyGroup"/>
                                                        </xs:complexType>
                                                </xs:element>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
                        <xs:element name="TotalOctets">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element ref="AVPParameters"/>
                                                <xs:element name="Data" type="xs:unsignedInt"/>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
                        <xs:element name="InputOctets">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element ref="AVPParameters"/>
                                                <xs:element name="Data" type="xs:unsignedInt"/>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
```

```
				<xs:element name="OutputOctets">
					<xs:complexType>
						<xs:sequence>
							<xs:element ref="AVPParameters"/>
							<xs:element name="Data" type="xs:unsignedInt"/>
						</xs:sequence>
					</xs:complexType>
				</xs:element>
				<xs:element name="ServiceSpecificUnits">
					<xs:complexType>
						<xs:sequence>
							<xs:element ref="AVPParameters"/>
							<xs:element name="Data" type="xs:unsignedInt"/>
						</xs:sequence>
					</xs:complexType>
				</xs:element>
			</xs:sequence>
		</xs:group>
```

The GrantedServiceGroup and UserServiceGroup have further level of nesting with the group data type which is grouped as MoneyGroup as defined below:

```
<xs:group name="MoneyGroup">
	<xs:sequence>
		<xs:element name="UnitValue">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:decimal"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
		<xs:element name="CurrencyCode">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:unsignedInt"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
	</xs:sequence>
</xs:group>
```

A.1.3.23 MPS Identifier AVP

<XML equivalent>

```
<xs:element name="MPSIdentifier">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

A.1.3.24 Rx-Request-Type AVP

<XML equivalent>

```
<xs:element name="RxRequestType">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="AVPParameters"/>
            <xs:element name="Data" type="xs:int"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

A.1.3.25 Required- Access- Info AVP

<XML equivalent>

```
<xs:element maxOccurs="unbounded" name="RequiredAccessInfo">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="AVPParameters"/>
            <xs:element name="Data" type="xs:int"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

A.1.3.26 Origin State ID AVP

<XML equivalent>

```
<xs:element name="OriginStateIDAVP">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="AVPParameters"/>
            <xs:element name="Data" type="xs:unsignedInt"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

A.1.3.27 Proxy Info AVP

<XML equivalent>

```
<xs:element maxOccurs="unbounded" name="ProxyInfoAVP">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="AVPParameters"/>
            <xs:element maxOccurs="unbounded" name="Data">
                <xs:complexType>
                    <xs:group ref="ProxyInfoGroup"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

This AVP is of group type. The mapping of this group type in XML schema involves defining a ProxyInfoGroup type defined as follows:

```
<xs:group name="ProxyInfoGroup">
	<xs:sequence>
		<xs:element name="ProxyHost">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:string"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
		<xs:element name="ProxyState">
			<xs:complexType>
				<xs:sequence>
					<xs:element ref="AVPParameters"/>
					<xs:element name="Data" type="xs:string"/>
				</xs:sequence>
			</xs:complexType>
		</xs:element>
	</xs:sequence>
</xs:group>
```

A.1.3.28 Route Record AVP

<XML equivalent>

```
<xs:element maxOccurs="unbounded" name="RouteRecordAVP" nillable="false">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="AVPParameters"/>
			<xs:element maxOccurs="unbounded" name="Data" type="xs:string"/>
		</xs:sequence>
	</xs:complexType>
</xs:element>
```

Annex <B> (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
06-2013					Includes the following TDOCs agreed at CT3#73: C3-130804, C3-130820, C3-130905, C3-130914, C3-130915, C3-130916	0.0.0	0.1.0
08-2013					Includes the following TDOCs agreed at CT3#74: C3-131093, C3-131231, C3-131232, C3-131233, C3-131234, C3-131235, C3-131260, C3-131294, C3-131297, C3-131283, C3-131300, C3-131313	0.1.0	0.2.0