WiMAX Forum® Network Architecture

(Stage 3: Detailed Protocols and Procedures)

[Annex: Prepaid Accounting]

Release 1.0 Version 4

WiMAX Forum APPROVED

WiMAX Forum Document Number

WMF - T33-002-R010v04

February 03, 2009

WiMAX Forum Proprietary

Table of Contents

Table of Figures

Revision History

March 2007	Initial draft
November 06, 2007	Implemented all Stage 3 accepted contributions from 00000_r055_NWG-Rel-1.0.0-CR-Tracking-Spreadsheet.xls.
January 11, 2008	Implemented all Stage 3 accepted contributions from 00000_r069_NWG-Rel-1[1].0.0-CR-Tracking-Spreadsheet.xls.
March 21, 2008	Implemented all Stage 3 accepted contributions from 00000_r084_NWG-Rel-1[1].0.0-CR-Tracking-Spreadsheet.xls.

Project: WiMAX Forum Network Working Group

Title: RADIUS Extension for Prepaid Number and file name Source(s)

Avi Lior Bridgewater Systems Corporation
Parviz Yegani Cisco Systems
Kuntal Chowdhury Starent Networks
Hannes Tschofenig Siemens
A. Pashalidis Siemens

Membership

Are all proposing organizations members of the WiMAX Forum? Yes [X] No [ ] Re: Rational

This document is a copy of an Internet Draft that is being proposed at the IETF. Since work at the IETF could take longer then the release cycles. The authors of the draft agree to provide the work as is for WiMAX. WiMAX may modify the document as it wishes. The authors will appreciate any feedback and would further appreciate if feedback be sent to the IETF as well.

Notice

Release

1. Introduction

This document specifies an extension to the RADIUS protocol that enables service providers to perform accounting and charging in an "online" fashion. In particular, they enable the service provider to (a) ensure that subscriber's remaining funds suffice before the service is delivered, and (b) interrupt service provision when the funds are exhausted. Note that these capabilities are typically used in scenarios where the subscriber maintains a prepaid account with the service provider; hence, this extension is called the "prepaid" extension for RADIUS. Also note that the above capabilities are not provided by the base RADIUS protocol.

It has been observed that subscribers prefer prepaid accounts to postpaid ones in many circumstances. Indeed, it is expected that offering a "prepaid mode of operation" will enable service providers to expand their existing customer bases. This is the main business driver behind the extensions defined in this document. The extensions were designed with the following goals in mind.

- Make use of existing infrastructure as much as possible (including enabling the interworking of RADIUS-based and Diameter-based infrastructures), and thereby limit the amount of necessary capital expenditures;
- provide the ability to rate service requests in an "online" fashion;
- provide the ability to charge the user's account prior to service provision;
- protect against revenue loss, i.e. to prevent an end user from obtaining service when the available funds do not suffice;
- protect against fraud, and;
- be deployable over dialup, wired and wireless networks.

The architecture between the entities that execute the RADIUS protocols, with the extensions defined in this document, assumes that the rating of chargeable events does not occur in the element that provides the service. Instead, the rating may be performed at a dedicated server, termed the "prepaid-enabled AAA server" or simply "prepaid server". Alternatively, the actual rating may occur in an entity behind this prepaid server. Furthermore, business logic may dictate a time-dependent tariff model, for example that the price for a service may switch at 8pm from a high to a low tariff. The extensions defined in this document support such scenarios.

Furthermore, this document assumes an architecture where a "quota server" is available which, through co-ordination with the rating entity and a centralized account balance manager, is able to provide a quota indication for a particular user when requested. This quota server may or may not coexist in the prepaid server.

1.1 Terminology

- Network Access Server (NAS): As defined in RADIUS;
- Prepaid client (PPC): The entity which triggers the RADIUS message exchange, including the prepaid extensions defined in this document. The PPC typically resides in the NAS;
- Prepaid Server (PPS): The entity that interacts with the PPC using the RADIUS prepaid extensions defined in this document;
- Home Network: The network which contains the user profile and the user's prepaid account;
- Authorize-Only Access Request: A RADIUS message of type "Access Request" (code field=1) that contains a "Service-Type" AVP (type=6) with value "Authorize-Only".

1.2 Overview

This section provides an overview of the prepaid charging models and architectures, which are supported by the extensions described in this document. A number of models of how to charge customers for data services in a prepaid manner are supported, as follows.

- Volume-based charging (e.g. 2 Cents/Kilobyte);

- Duration-based charging (e.g. 3 Cents/minute);
- Subscription-based charging (e.g. 10 Dollars/month);
- Event-based charging (e.g. 7 Cents/URL or email).

This draft assumes that the user maintains a prepaid account with his home network. This account may be used to fund multiple services, some of which may use the extensions defined in this document, and some may use other mechanisms. The interworking of these mechanisms is outside the scope of this document. Similarly, the means by which the subscriber obtains funds is also outside the scope of this document.

1.2.1 Architectural Model

The protocol extensions described in this draft assumes that the following entities are present in the network architecture.

Service Access Device (SAD): This entity provides a data service to the users, and typically coincides with the NAS. The SAD executes the RADIUS client which, for the purposes of this document, is termed the "PrePaid Client" (PPC). When the prepaid service is used the SAD collects service event information and reports it while services are provided to the user. This event information is sent to the PPS using the extensions defined in this document.

The PPS: The RADUIS server that supports the prepaid extensions defined in this document. If real-time credit control is required, the PPC (SAD) contacts the PPS with service event information included before the service is provided. The PPS performs a credit check and allocates a portion of the available credit to the service event.

The rating entity: This entity converts the credit that is allocated by the PPS into a time or volume amount, called the "quota". This quota is then returned to the requesting PPC (SAD) (via the PPS). The rating entity may also determine that during service provision a tariff switch will occur. In this case the rating entity will include details of when exactly tariff switch will occur.

The requesting PPC (SAD) meters the consumption of the service according to the instructions provided by the PPS. After service completion, or on reception of a subsequent request for service, the PPS deducts the corresponding amount of credit from the user account. When a user terminates an on-going service, the PPC informs the PPS with a suitable indication about the unused portion of the allocated quota. The PPS then refunds the user account accordingly. Note that multiple PPSs may be deployed for reasons of redundancy and load sharing. The system MAY also employ multiple rating servers.

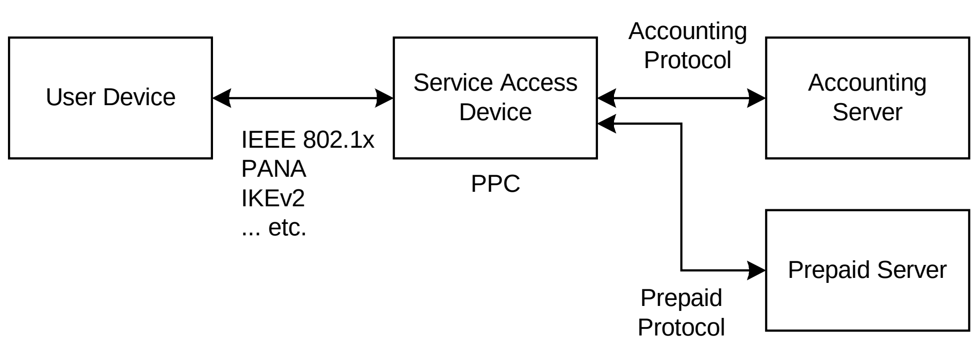


Figure 1 - Basic Prepaid Architecture

The PPS and the accounting server in this architecture MAY be combined. The SAD must have the ability to meter the consumption of a prepaid data session. This metering is typically based on time (i.e., seconds) or volume (i.e., octets). The device running the PPC may also have "Dynamic Session Capabilities" such as the ability to terminate a data session or change the filters associated with a specific data session by processing "Disconnect" messages and "Change of Authorization" messages as per RFC 3576.

This document assumes that the PPS is used as the AAA server. There are three types of AAA server, as follows.

The AAA server in the home network (HAAA) is responsible for authentication of the subscriber. In addition, the HAAA communicates with the PPS using the RADIUS protocol in order to authorize subscribers.

The AAA server in the visited network (VAAA), which exists only in roaming scenarios, is responsible for forwarding the RADIUS messages to the HAAA. The VAAA may also modify the messages. Note that, in certain roaming deployments, the visited network may be connected to the home network via one or more broker networks.

The AAA server in one of the aforementioned broker networks (BAAA), which is responsible for forwarding messages and does not play an active role in the prepaid data service delivery. A BAAA obviously exists only in those roaming deployments where the VAAA and the HAAA are connected via the BAAA of a broker network.

This document assumes that the PPS communicates with the HAAA for the purposes of authentication and authorization. The PPS, in turn, interfaces to entities which

- keep the subscriber's account balance (balance manager);
- rate access service requests in real-time (Rating Engine), and;
- manage quota for a particular prepaid service (Quota Server).

The above entities belong to the service provider's backend infrastructure and are outside the scope of this specification. In particular, as far as this specification is concerned, they are assumed to exist in the PPS. Three deployment scenarios are presented in the remainder of this section. The first scenario is depicted in Figure 2. In this scenario, the SAD, which runs the PPC, the HAAA, and the PPS are located in the same provider network.

The subscriber's device establishes a connection with one of possibly multiple SADs in the network. The selected SAD communicates with a HAAA server (directly or indirectly).

The interface between the HAAA and the PPS is implemented using the RADIUS protocol together with the extensions described in this document. However, in cases where the PPS does not implement the RADIUS protocol, the implementation would have to map the requirements defined in this document to a functionally equivalent protocol.

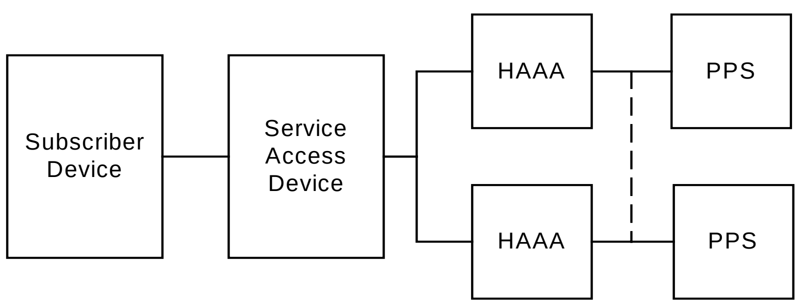


Figure 2 - Basic Prepaid Access Architecture

The second scenario, depicted in Figure 3, is based on a static roaming architecture that is typical of a wholesale scenario for Dial-Up users or a broker scenario used in Dial-Up or WLAN roaming scenarios.

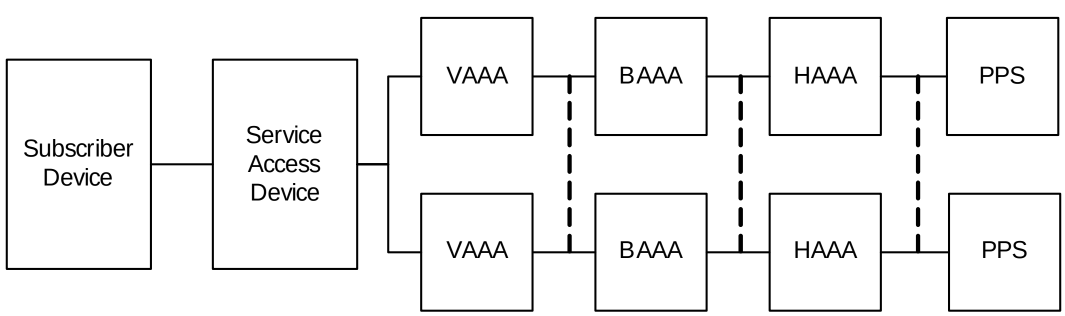


Figure 3 - Static Roaming Prepaid Architecture

Like in the basic prepaid architecture, the subscriber device establishes a connection with the SAD. The SAD communicates with the VAAA using the RADIUS protocol. The VAAA, in turn, communicates using the RADIUS protocol with BAAA servers in the broker network. There may be more than one Broker Network between the Visited Network and the Home Network. The Home Network is the same as in the architecture depicted in Figure 2.

Broker AAA (BAAA) servers SHALL support the Message-Authenticator(80) attribute as defined in RFC 2869. If they are used, they forward the RADIUS packets as usual to the appropriate RADIUS servers.

Accounting messages are not needed to deliver a prepaid service. However, accounting messages can be used to keep the PPS up to date as to what is happening with the prepaid data session. Therefore, a BAAA SHOULD deliver RADIUS Accounting messages using the pass through mode described in RFC 2866.

1.2.2 Motivation

Why not use existing RADIUS attributes to construct a protocol for prepaid scenarios? This could lead to a solution where no code has to be modified at existing devices.

It is indeed possible to construct a solution for prepaid scenarios using existing RADIUS attributes. The RADIUS server would send an Access-Accept message containing a Session-Timeout(27) and include a Termination-Action(29) in the RADIUS-request. Upon receiving the Access-Accept message, the NAS would meter the duration of the session and upon termination of the session the NAS would generate an Access-Request message again. The RADIUS server would then re-authenticate the session and reply with an Access-Accept message indicating the amount of additional time in a Session-Timeout(27). Alternatively, it could respond with an Access-Reject message if there were no more resources in the user account.

Moreover, if the user terminates the session prematurely, the NAS could indicate this in the accounting stream so that unused funds can be returned into the prepaid user account.

Unfortunately, the above "solution" has a number of drawbacks, including the following.

It only supports time-based charging. The solution presented in this document supports multiple charging metrics.

Using accounting messages to recoup unused time may be problematic because RADIUS accounting messages are not delivered in real-time. A RADIUS server may store-and-forward accounting messages in batches. Thus, relying on accounting messages for the purposes of prepaid may cause revenue leakage. The solution presented in this document does not rely on Accounting packets at all. It uses Access-Request messages, which are required to flow through any network in real-time.

Session-Timeout(27) is not a mandatory attribute. If a prepaid subscriber is served by a NAS that does not adhere to Session-Timeout then that subscriber may use the service for an undetermined period of time.

Termination-Action(29) presents its own issues. Firstly, the behavior of Termination-Action(29) is not mandatory. Secondly, according to RFC 2865, Termination-Action fires when the provision of the service has completed. However, service should not be terminated when negotiating additional quota, because this should happen in a manner transparent to the subscriber. Due to the fact that Termination-Action occurs when the service is completed, it is unclear whether or not user experience would be affected if this attribute would be used in a prepaid scenario. The RADIUS server might even allocate a new IP address to the subscriber device after a Termination-Action. Also, the RADIUS server has no way of telling why a given Access-Request message was generated. The RADIUS server might have to wait for the corresponding accounting packet to determine the reason. Finally, re-authenticating the subscriber may take too long. The solution presented in this document allows quota replenishing to occur without affecting user experience. No re-authentication is required and quotas can be negotiated before the available credit actually runs out.

Due to the fact that the standard RADIUS attributes are not mandatory, the correct prepaid operation is really an act of faith on the part of the RADIUS server. If Session-Timeout(27) and/or Termination-Action(29) are not supported, the prepaid subscriber might be able to obtain the service for free. The solution described in this document requires that a prepaid-aware SAD informs the RADIUS server, regardless of whether or not the latter supports the prepaid extensions. The RADIUS server can then determine whether or not service should be granted. For example, if a prepaid subscriber is connected to a NAS that does not support prepaid, the RADIUS server can either instruct the NAS to tunnel the traffic to another entity in the home network (e.g., an Home Agent) that supports prepaid, or cause it to provide only a restricted service.

The solution presented in this document requires the support of two mandatory and one optional attribute. Furthermore, it does not require a great amount of additional code at a NAS that already supports time or volume metering. The solution requires that RADIUS entities advertise their prepaid capabilities in an Access-Request and that they generate an Access-Request packet with Service-Type="Authorize-Only" in order to obtain more quota when or before the current quota is used up. It also requires the NAS to send an Access-Request with Service-Type="Authorize-Only" when the session terminates in order to refund the subscriber account.

1.3 A Simple Use Case

This section describes the sequence of events in a simple RADIUS prepaid transaction.

When an end host attaches to a network (for example, using PPP or PANA), as usual, the NAS (SAD) that is servicing the subscriber uses the AAA infrastructure in order to authenticate and authorize the subscriber with respect to the requested service. In order to do this, it sends a RADIUS Access-Request to the AAA server. This Access-Request contains the subscriber's credentials and may contain the prepaid capabilities of the SAD. Prepaid capabilities SHALL be included if the SAD supports them.

The authentication procedure proceeds. This may involve several message exchanges such as in EAP [RFC2284]. Once the subscriber has been successfully authenticated, the home AAA server determines that the subscriber is a prepaid subscriber and requests authorization from the PPS. This request SHALL include the prepaid capabilities of the serving SAD.

The PPS, possibly with the help of the backend infrastructure, validates that the subscriber has a prepaid account and that the account is active. It further validates that the SAD has the appropriate prepaid capabilities. If all is in order, the PPS authorizes the subscriber to use the network. Otherwise it rejects the request. The decision is sent to the AAA system in the form of a response message. In the case of success, this message contains attributes that indicate the allocation of a portion of the subscriber credit. This portion is called the "initial quota" and is expressed in units of time or volume. The response may also include a threshold value. Note that only a portion of the user's funds is allocated because the user may be engaged in other services that may draw on the same account. For example, the user may be engaged in a data session and a voice session. Although these two services would draw from the same account, they form separate parts of the overall system. If the entire quota was allocated to the data session then the user would have no more funds for a voice session.

The AAA system incorporates the attributes received from the PPS into an Access-Accept message that it sends to the SAD. Note that the AAA system is responsible for authorizing the service whereas the prepaid system is responsible for prepaid authorization.

Upon receiving the Access-Response, the SAD starts the prepaid data session and meters the session based on time or volume, as indicated in the message.

Once the consumption approaches the allocated limit (as expressed by the threshold), the SAD will request additional quota. Re-authorization for additional quota flows through the AAA system to the PPS. The PPS revalidates the subscriber account and subtracts the previously allocated quota from the current balance. If there is remaining balance, it reauthorizes the request with an additional quota allotment. Otherwise, the PPS rejects the request. Note that the replenishment of the quota is a re-authorization procedure and does not require the subscriber to authenticate himself again.

Upon receiving a re-allotment of the quota, the SAD continues to provide the data service until the new threshold is reached. If the request for additional quota cannot be fulfilled then the SAD lets the subscriber use the remaining quota and terminates the session. Alternatively, instead of terminating the session, the SAD may restrict the data session such that the subscriber can only reach a particular web server. This web server maybe used to allow the subscriber to replenish his account. This restriction can also be used to allow new subscribers to set up prepaid accounts in the first place.

Should the subscriber terminate the session before the quota is exhausted, the remaining balance allotted to the session SHALL be refunded into his account.

Note that the subscriber may have disconnected while the Access Device is waiting for the initial quota. The entire allocated quota will have to be credited back to the subscribers account in this case. Also note that the PPS maintains session state for the subscriber. This state includes how much account balance was allocated during the last quota enquiry and how much is left in the account. Therefore, it is required that all messages about the session reach the same (and correct) PPS.

For a simple message flow, along the lines of this use case, please see Appendix A.

2. Supported Features

This section describes the features that are supported by the extensions specified in this document.

2.1 Multiple Concurrent Services

Examples of services that the user may be using are browsing the web, participating in a VoIP conversation, watching streaming video and downloading a file. Some operators may want to distinguish between these services. Some services are charged at different rates and services may be metered differently. Therefore, the prepaid solution needs to be able to distinguish services, and allocate quota to the services using different unit types (time, volume) and allow for those quotas to be consumed at different rates.

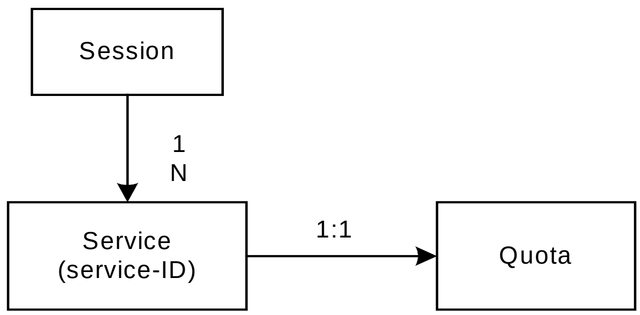


Figure 4 - Multiple Services within a Single Session

As shown in Figure 4, a session may be associated with multiple (N) services. Each service is identified by a service identifier (Service-ID). The format of the Service-ID is not in the scope of this document but it could be expressed as an IP flow using the 5-tuple {Source-IP and Port, Destination-IP and Port, protocol type}. Each service is associated with a quota metric. An example message flow that involves multiple such services within a single session is given in the appendix.

2.2 Resource Pools

When working with multiple services a new problem arises because one service may consume its quota faster than another service. When the user balance is close to exhaustion, a situation could arise where one service is unable to obtain quota while another service has plenty of quota remaining. Unless the quotas can be rebalanced, the SAD would then have to terminate the former service; moreover, if each service generates a certain amount of RADIUS prepaid traffic. In an environment with many users and chargeable services, this amount of traffic is considerable and could cause undesirable network congestion.

One method to circumvent the above situation is to use a so-called "resource pool". Resource pools enable the allocation of resources to multiple services of a session by allocating resources to a pool and have services draw their quota from the pool at a rate appropriate to that service. When the quota that has been allocated to the pool is close to exhaustion, the entire pool (rather than individual services) is replenished.

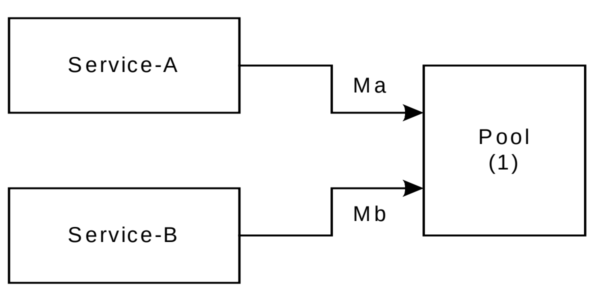


Figure 5 - Resource Pool Example

As shown in Figure 5, Service-A and Service-B are bound to Pool(1). Ma and Mb are the pool multipliers (that are associated with Service-A and Service-B respectively) that determine the rate at which Service-A and Service-B draw from the pool.

The pool is initialized by taking the quota allocated to service n and multiplying it by Mn. Therefore, the amount of resources allocated to a pool is given by Poolr = Ma*Qa + Mb*Qb + . . ., where Qn denotes the amount of quota that is allocated to service n. Further, the pool is considered to be empty if

Poolr <= Ca*Ma + Cb*Mb + . . .,

where Ca and Cb are resources consumed by Service-A and Service-B respectively.

Note that the resources assigned to the pool are not associated with a metric. That is, Service-A can be rated at $1 per MB and Service-B can rated at $0.10 per minute. In this case if $5 worth of resources are allocated for service-A to the pool and if Ma = 10, then 50 units would be placed into the pool. If a further $5 are allocated for service-B to the pool, then M=1 and 50 units are deposited into the pool. The pool would then have a sum of 100 units to be shared between the two services. The PPC would then meter the services such that each Mbyte used by Service-A will draw 10 units from the pool and each minute used by Service-B will draw 1 unit from the pool.

2.3 Complex Rating Functions

The rating of a service can be quite complex. While some operators follow linear pricing models, others may wish to apply more complex functions. For example, a service provider may wish to rate a service such that the first N MBytes are free, then the next M Mbytes are rated at $1 per MB and volume above (N+M) MB be rated at $0.50 per MB. Such a function could be implemented by repeated message exchanges in the prepaid system.

To avert the need to exchange many messages while still supporting such complex rating functions, the notion of a "Rating Group" is introduced. A Rating Group are typically configured at the SAD. As shown in Figure 7, a Rating Group is associated with one or more services and defines the rate that the services associated with the Rating Group consume an allocated amount of quota.

Figure 6 - Example of a Rating Group

During the usage of a service that is associated with a Rating Group, the PPC sends the ID of the Rating Group to the PPS. The PPS authorizes the Rating Group by allocating a quota to it and optionally assigning it to a Resource Pool. When an additional service that belongs to an already authorized Rating Group is instantiated, the PPC does not need to authorize this service. This effectively means that the PPC meters the service such that it draws from the already allocated quota. Therefore, no RADIUS messages need to be exchanged in this case. This limits the amount of traffic between the PPC and the PPS. An example of a flow that uses Rating Groups is given in Appendix A.3.

2.4 One-time Charging

One-time charging is a mode of operation of where the RADIUS prepaid extensions are used for charging of a service that is provided instantaneously, i.e., without an ongoing session. An example of such an event is the purchase of a ring-tone. Subscription based services can also be modeled as a one-time event. In this case the one-time service event is the purchase of a subscription.

For a given user, one-time charging may occur in parallel with other charging models. For example, the subscriber may access a website which is metered (based on time or volume) while he also purchases the right to use a ring tone (a one-time-based event). Note: it is up to the service providers to decide whether or not the user will be charged for the download of the tone and also be charged for the time and volume required to download the ring-tone. The facilities provided by this document give the service provider the capability to achieve their service charging business goals. For example, should the service provider choose not to charge for the download volume or time, they can treat the download IP flow as a separate service that is not subject to charging.

The SAD signals one-time charging to the PPS with an indication that identifies the service and the units that should be debited from the user account.

A SAD may decide to perform one-time charging for an event that was triggered by an unauthenticated user. In this case the SAD will have to authenticate the user before sending the relevant message to the user's home AAA server.

Note that one-time charging can also be used to credit the prepaid account. For example, the SAD can return resources to the subscriber by issuing a one-time charge request that includes the amount of resources to be credited into the account.

2.5 Tariff Switching

The PPC and the PPS may support tariff switching mechanism described in this section. This mechanism is useful if, for example, as shown in Figure 8, traffic before 18:00 is rated at rate r1 and traffic after 18:00 is rated at rate r2. The mechanism requires the PPC to report usage before and after the switch occurred.

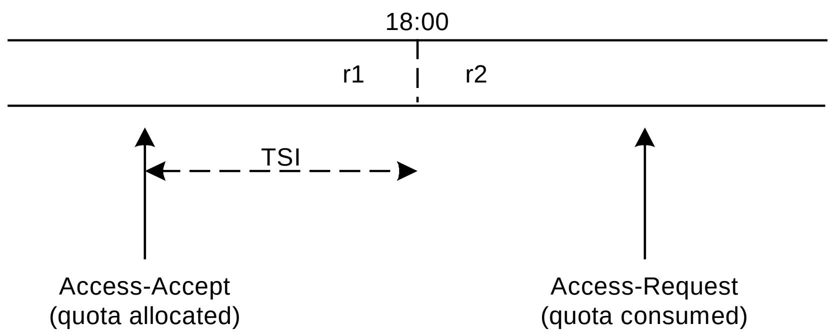


Figure 7 - Example of Tariff Switching

The PPC it indicates support for tariff switching by setting the appropriate bit in the PPAC. If the PPS needs to signal a tariff switch time, it will send a PTS attribute which indicates the point in time when the switch will occur. This indication represents the number of seconds from current time (TariffSwitchInterval TSI).

At some point after the tariff switch the PPC sends another Access-Request, as a result of either the user having logged off or the volume threshold being reached. The PPC reports how much volume was used in total (in a PPAQ attribute) and how much volume was used after the tariff switch (in a PTS VUATS subtype attribute).

In situations with multiple tariff switches, the PPS must specify the length of the tariff switch period using the TimeIntervalAfterTariffSwitchUpdate (TITSU) in the PTS attribute as shown below.

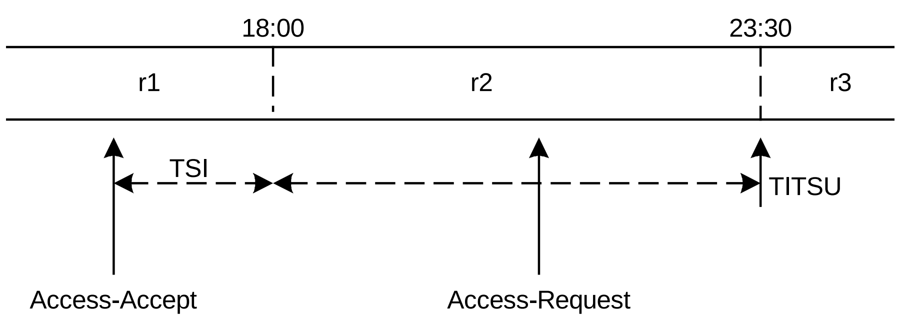


Figure 8 - Multiple Tariff Switches

When a TITSU is specified in the PTS, the PPC SHALL generate an Access-Request within the time after TSI and before TITSU expires. Note that, typically, the PPC will be triggered by the Volume Threshold. However, it is possible that, during period r2, resources are not entirely consumed and, thus, the threshold is not reached. The TITSU attribute ensures that, even in this case, the PPC will generate the new Access-Request in good time.

Note that it makes no sense to use the tariff switching mechanism described in this section for services that are metered based on time and the consumption of which is continuous (i.e., without interruption). Also note that separate services flows may have individual tariff periods.

2.6 Support for Roaming

In certain networks it is essential for prepaid data services to be available to roaming subscribers. Support for both static and dynamic roaming models is needed. In a static roaming scenario the subscriber connects to a foreign network which has a roaming agreement either directly with the home network, or through a broker network. When the subscriber logs into another foreign network, a new login procedure has to be executed.

In a dynamic roaming scenario the subscriber may move between networks while maintaining his connection. In such a scenario the data session is seamlessly handed off between the networks.

In both roaming scenarios, the subscriber always authenticates himself to the home network. Authorization for the prepaid session and quota replenishing occurs at the home network and more specifically at the PPS where state is being maintained.

Dynamic roaming is challenging because a subscriber who established a prepaid data session may move to another Access Device that does not support the prepaid extensions. Even in this case the system should be able to continue the prepaid session.

2.7 Dynamic Termination

When fraud or an error is detected, either only the affected session or all sessions, of the affected subscriber, should be immediately terminated. It may further happen that the prepaid system enters a state where it is unclear whether or not the data session is in progress. Under certain conditions, the system may wish to terminate the session in order to make sure that the user is not charged for this potential inactivity.

Certain handoff procedures used in dynamic roaming scenarios require that the system terminates the subscribers prepaid data session at a SAD. This is the case, for example, when time-based prepaid is used and the mobile subscriber performs a dormant handoff.

2.8 Querying and Rebalancing

It should be possible for the PPS to Query the current resource consumption at a SAD and adjust the user account balance. For example, a request to the PPS is made (e.g., a one-time charging event), the account is depleted and resources have been allocated to the SAD. The PPS should have the ability to query the SAD and if it has the spare resources to reassign the quotas to the SAD and to the pending request. Note that the PPS does not know resource usage until the SAD request for more resources. This can be a long time.

In the absence of this capability, the PPS can minimize the effect of this phenomenon by allocating small quotas, a practice which results in more message exchanges.

3. Operations

This section describes the operations that are implemented by a prepaid-enabled NAS (SAD).

3.1 Authentication and Authorization Operation

The SAD initiates the authentication and authorization procedure by sending a RADIUS Access-Request to the HAAA. Since the SAD has PPC capabilities, it SHALL include a PPAC attribute in the RADIUS Access-Request. The PPAC attribute indicates to the PPS which prepaid capabilities are possessed by the SAD. These are required in order to complete the prepaid authorization procedure. Moreover, if the SAD supports the Disconnect-Message or the Change-of-Authorization capabilities, then it SHOULD include the Dynamic-Capabilities attribute.

In certain deployments, there may be other ways to terminate a data session, or change authorization of an active session. For example, some SADs provide a session termination service via Telnet or SNMP. In these cases, the AAA server MAY add the Dynamic-Capabilities message to the Access-Request. Upon receiving the Change-of-Authorization message, the AAA server would then be responsible for terminating the session using the means that are supported by the device.

If the authentication procedure involves multiple message exchanges (as in EAP), the SAD SHALL include the PPAC attribute and the Dynamic-Capabilities attribute (if used) in at least the last Access-Request of the authentication procedure.

The Access-Request is sent, as usual, to the HAAA, possibly through one or more BAAA. Once the Access-Request arrives at the HAAA, the HAAA authenticates the subscriber. If this fails, the HAAA sends an Access-Reject message to the client. If authentication succeeds, the HAAA determines whether or not the subscriber is a prepaid subscriber. (How this is done is beyond the scope of this document.) If the subscriber is not a prepaid subscriber, then the HAAA responds as usual with an Access-Accept or an Access-Reject message. If the subscriber is a prepaid subscriber, the HAAA SHALL forward the Access-Request to the PPS for further authorization.

The Access-Request contains the PPAC attribute and the Dynamic-Capabilities attribute if one was included. The User-Name(1) attribute MAY be set to a value that identifies the subscriber. This attribute is used by the PPS to locate his account. For added security, the HAAA MAY also set the User-Password(2) attribute to the password used between the HAAA and the PPS.

The PPS locates the subscriber account and authorizes him. During this procedure, the PPS takes into consideration the SAD PPC Capabilities. Upon successful authorization, the PPS generates an Access-Accept containing the PPAQ attribute. The PPAQ attribute includes the following information.

- The QUOTA-ID, which is set by the PPS to a unique value that is used to correlate subsequent quota requests;
- Volume and/or Time quota, which is set to a value representing a portion of the subscriber's credit;
- It MAY contain a Time or Volume Threshold that indicates when the SAD should request additional quota;
- The IP address of the Serving PPS and one or more alternative PPSs. This is used by the HAAA to route subsequent quota replenishing messages to the appropriate PPS(s);
- A State attribute, as defined in RFC 2865. This is necessary in order to satisfy the requirements of section 5.44 of RFC 2865, which mandates that an Access-Request with Service-Type="Authorize-Only" must contain a State attribute. Since the SAD sends subsequent quota replenishment requests in the form of such "Authorize-Only" requests, a State attribute SHALL be present in all Access-Accept messages that also carry a PPAQ attribute.

Note: The Idle-Timeout(28) attribute can be used to trigger the premature termination of a prepaid service, for example as a result of inactivity.

Depending on site policies, after failed authorization, the PPS may generate an Access-Reject in order to terminate the session immediately. Alternatively, the PPS may generate an Access-Accept blocking some or all of the traffic and/or redirect some or all of the traffic to a location to a fixed server. (This feature could be used, for example, to prompt the user to replenish their account.) Blocking of traffic is achieved by either Filter-ID(11) or NAS-Filter-

Rule(see Redirect I-d). Redirection is achieved by sending Redirect-Id or Redirect-Rule, HTTP Redirection defined in the Redirect I-d. The time period before the session is blocked/redirected is specified by the Session-Timeout(27) attribute.

Upon receiving an Access-Accept from the PPS, the HAAA appends the usual service attributes and forward the packet to the SAD. The HAAA SHOULD NOT overwrite any attributes already set by the PPS. If the HAAA receives an Access-Reject message, it will simply forward the packet to its client. Depending on site policies, if the HAAA does not receive an Access-Accept or an Access-Reject message from the PPS it MAY do nothing or send an Access-Reject or an Access-Accept message back to the PPC.

3.2 Session Start Operation

The start of the session is indicated by the arrival of an Accounting-Request(Start) packet. The Accounting-Request (Start) MAY be routed to the PPS such that it can confirm the initial quota allocation.

Note that the role of the PPS is not to record accounting messages and therefore it SHOULD NOT respond with an Accounting Response packet. If the PPS does not receive the Accounting-Request(start) message it will only know that the session has started upon the first reception of a quota replenishment operation.

If the PPS does not receive indication directly (via Accounting-Request(start)) or indirectly, it SHOULD, after some configurable time, deduce that the Session has not started. If the SAD supports termination capabilities, the PPS SHOULD send a Disconnect Message to the SAD as a measure to ensure that the session is indeed dead.

3.3 Mid-Session Operation

During the lifetime of a prepaid data session the SAD may request the replenishment of the quotas using an Authorize-Only Access-Request message. Once either the allocated quota has been exhausted or the threshold has been reached, the SAD SHALL send an Access-Request with Service-Type(6) set to a value of "Authorize-Only" and the PPAQ attribute.

The SAD SHALL also include NAS identifiers, and Session identifier attributes in the Authorize-Only Access-Request. The Session Identifier should be the same as the one used during the initial Access-Request. For example, if the User-Name(1) attribute was used in the Access-Request it SHALL be included in the Authorize-Only Access-Request, especially if the User-Name(1) attribute is used to route the Access-Request to the Home AAA server.

The Authorize-Only Access-Request SHALL NOT include a User Password and SHALL NOT include a Chap Password. In order to enable the receiver to authenticate the message, the SAD SHALL include a Message-Authenticator(80). In order to satisfy the requirements of section 5.44 of RFC 2865, the SAD SHALL also include the State attribute. It is anticipated that the inclusion of the State attribute will enable the PPS to map the Authorize-Only Access Request to the authentication context that was established when the PPC authenticated itself at the beginning of the session. The SAD computes the value for the Message-Authenticator and the State attributes according to RFC 2869 and 2865 respectively.

When the HAAA receives an Authorize-Only Access-Request that contains a PPAQ,it SHALL validate the message using the Message-Authenticator(80), according to RFC 2869. If the HAAA receives an Authorize-Only Access-Request that contains a PPAQ and either no or an invalid Message-Authenticator(80) it SHALL silently discard the message. An Authorize Only Access-Request message that does not contain a PPAQ is either erroneous or belongs to another application (for example, a Change of Authorization message [RFC3576]). In this case the Authorize-Only Access-Request is either silently discarded or handled by another application.

Once the Authorize-Only Access-Request message is validated, the HAAA SHALL forward the Authorize-Only Access-Request to the appropriate PPS. The HAAA SHALL forward the Authorize-Only Access-Request to the PPS specified in the PPAQ. The HAAA SHALL add a Message-Authenticator(80) to the message, according to RFC 2869. As with the Access-Request message, the HAAA MAY modify the User-Name(1) attribute such that it identifies the user to the PPS. Note that the PPS may also use the Quota-ID sub-attribute contained within the PPAQ to locate the user account.

Upon receiving the Authorize-Only Access-Request containing a PPAQ attribute, the PPS SHALL validate the Message-Authenticator(80) as described in RFC 2869. If validation fails, the PPS SHALL silently discard the message. If it receives an Authorize-Only Access-Request message that does not contain a PPAQ, it SHALL silently discard the message.

The PPS locates the prepaid session state using the Quota Id contained within the PPAQ. The PPS takes the most recently allocated quota and subtracts it from the user balance. If sufficient balance remains, the PPS authorizes the PPS and allocates additional quota. The PPS may also calculate a new threshold value. Upon successful re-authorization, the PPS generates an Access-Accept containing the PPAQ attribute. The Access-Accept message MAY contain Servicetype(6) set to Authorize-Only and MAY contain the Message-Authenticator(80).

Depending on site policies, upon unsuccessful authorization, the PPS generates an Access-Reject or an Access-Accept with Filter-Id(11) or Ascend-Data-Filter (if supported) attribute and the Session- Timeout(27) attribute such that the subscriber can get access to a restricted set of locations for a short period of time. This feature could be used to enable users to replenish their accounts, create new accounts, or to browse free content.

Upon receiving the Access-Accept from the PPS, the HAAA SHALL return the packet to its client. If the HAAA receives an Access-Reject message, it forwards the packet. Depending on site policies, if the HAAA does not receive an Access-Accept or an Access-Reject message from the PPS it MAY do nothing or it MAY send an Access-Reject message back to its client.

Upon receiving an Access-Accept, the SAD SHALL update its quotas and threshold parameters with the values contained in the PPAQ attribute. Note that the PPS MAY update the PrePaidServer attribute(s) and these may have to be saved as well. If the Access-Accept message contains a Filter-Id(11), an Ascend-Data-Filter attribute, or Session Timeout(27), the SAD SHALL restrict the subscriber session accordingly.

3.4 Dynamic Operations

The PPS may take advantage of the dynamic capabilities that are supported by the SAD as advertised in the Dynamic-Capabilities attribute during the initial Access-Request. There are two types of action that the PPS may perform. Firstly, it may request the session to be terminated. Secondly, it may request the attributes associated with the session to be modified. More specifically, it may modify a previously sent PPAQ.

Both of these actions require that the session be uniquely identified at the SAD. As a minimum, the PPS SHALL provide

- either the NAS-IP-Address(4) or the NAS-Identifier(32), and;
- at least one session identifier such as User-Name(1), Framed-IP-Address(), the Accounting-Session-Id(44).

Other attributes could also be used to uniquely identify a prepaid data session.

3.4.1 Unsolicited Session Termination Operation

At anytime during a session the PPS may send a Disconnect Message in order to terminate a session. This capability is described in detail in [RFC3576]. The PPS sends a Disconnect Message that SHALL contain identifiers that uniquely identify the data session and the SAD servicing that session.

If the SAD receives a Disconnect-Message, it responds with either a Disconnect-ACK message (if it is able to terminate the session) or with a Disconnect-NAK packet (otherwise). Upon successful termination of a session the SAD SHALL return any unused quota to the PPS by issuing an Authorize-Only Access-Request containing the PPAQ which contains any unused Quota and the Update-Reason set to "Remote Forced Disconnection".

3.4.2 Unsolicited Change of Authorization Operation

At any time during the session the PPC may receive a Change of Authorization (CoA) message. A PPS may send a new Quota to either add or to remove quota that is allocated to the service. If the Change of Authorization contains a PPAQ then that PPAQ overrides a previously received PPAQ. The PPS SHALL NOT change the units used in the PPAQ.

If the newly received PPAQ reduces the amount of allocated quota beyond what is already used then the SAD accepts the new PPAQ and act as it normally would when the quota is used up. For example, if the threshold is reached then is request a quota update is requested.

3.5 Termination Operation

The termination phase is initiated when (i) the subscriber logs off, (ii) the subscriber balance is exhausted, or (iii) when the SAD receives a Disconnect Message.

In case the user logged off, or the SAD receives a Disconnect Message, the SAD sends an Authorize-Only Access-Request message with a PPAQ and Update-Reason attribute set to either "Client Service Termination" or "Remote Forced Disconnect". This message indicates the amount of consumed quota.

In case the currently allocated quota is exhausted, if the PPAQ contained the Termination-Action field, the SAD follows the specified action (which would be to immediately terminate the service, request more quota, or redirect/filter the service).

3.6 Mobile IP Operations

In roaming scenarios with Mobile-IP, the prepaid data session should be maintained transparently if the HA is acting as the SAD. As the subscriber device associates with a new SAD (AP or PDSN that supports PPC capability), the SAD sends a RADIUS Access-Request and the subscriber is re-authenticated and reauthorized. The SAD SHALL include the PPAC attribute in the RADIUS Access-Request. In this manner, the procedure follows the Authentication and Authorization procedure described earlier.

If the HA was acting as the SAD before handoff, the prepaid session does not undergo any change after the handoff because the Mobile IP session is anchored at the HA and the user's Home IP address does not change.

In the case of a wireless access point or PDSN acting as the SAD, it is likely that the user's (care-of) IP address will change. The prepaid session will be affected by this. In this scenario the SAD shall send an Access-Request message which is routed to the home network and SHALL reach the PPS that is serving this session. The PPS correlates the new authorization request with the existing active session and assigns a quota to the new request. Any outstanding quota at the old SAD SHALL be returned to the PPS if the Mobile-IP nodes (HA and FA) support registration revocation (Mobile IPv4 only). Specifically, the quota SHOULD be returned when the SAD sends the Authorize-Only Access-Request with PPAQ Update-Reason set to either "Remote Forced Disconnect" or "Client Service Termination". In order to trigger the sending of this last Authorize-Only Access- Request, the PPS may issue a Disconnect Message [3576] to the SAD.

Even if the subscriber moves to a SAD that does not have prepaid capabilities can the prepaid data service continue. This can be done by requesting the Home Agent (assuming it has such capabilities) to take over the responsibilities of the SAD (i.e., metering). This scenario will be discussed in detail in a later version of this document.

3.7 Operation Considerations for Multiple Services

This section describes the support for multiple prepaid services on a single SAD. Message flows illustrating the various interactions are presented in Appendix A.

A SAD that supports prepaid operations for multi-services SHOULD set the "Multi-Services Supported" bit in the PPAC. When working with multi-services, we need to differentiate between the services. A Service-Id attribute is used in the PPAQ in order to uniquely differentiate between the services. The exact definition of the Service-Id attribute is outside the scope of this document.

A PPAQ that contains a Service-Id is associated with that Service. A PPAQ that contains a Rating-Group-Id is associated with that Rating-Group. A PPAQ SHALL not contain both a Rating-Group-Id and a Service-Id. A PPAQ that contains neither a Rating-Group-Id nor a Service-Id applies to the Access Service.

3.7.1 Initial Quota Request

When operations with multi-services are desired, the SAD requests the initial quota for the Service by sending a PPAQ containing the Service-Id for that Service in an Authorize-Only Access-Request packet. Similarly, if the SAD supports Rating-Groups then it may request a quota for the Rating-Group by sending a PPAQ containing the Rating-Group-Id. In both cases the Update-Reason is set to "Initial-Request".

The Authorize-Only Access-Request message may contain more than one PPAQ. The Authorize-Only Access-Request SHALL include one or more attributes that serve to identify the session so that it can be linked to the original authentication. Which Session Identifiers are included is up to specific deployments. The Authorize-Only message must contain the Message-Authenticator(80) attribute for integrity protection of the Authorize-Only Access-Request message.

Upon receiving an Authorize-Only Access-Accept message containing one or more PPAQs, the PPS allocates resources to each PPAQ. Each PPAQ is assigned a unique QID that SHALL appear in subsequent PPAQ updates for that service or rating-group. Additionally, the PPAQ SHALL contain the Service-ID or Group-ID, unless the PPAQ is the generic "Access Service".

3.7.2 Quota Update

Once the services start to utilize their allotted quota they will eventually need to replenish their quotas (either the threshold is reached or no more quota remains). In order to replenish the quota, the PPC sends an Authorize-Only Access-Request message containing one or more PPAQs. Each PPAQ SHALL contain the appropriate QID, Service-ID or Group-ID (or neither the Service-ID or Group-Id if the quota replenishment is for the "Access Service"). The Update-Reason filed indicates either "Threshold reached"(3), or "Quota reached"(4). The Authorize-Only message must contain session identifiers.

Upon receiving an Authorize-Only Access-Request packet with one or more PPAQs the PPS responds with a new PPAQ for that service. The PPAQ contains a new QID, the Service-Id or Rating-Group-Id, a new Quota. If the PPS does not grant additional quota to the service it SHALL include the Termination-Action subfield in the PPAQ that will instruct the SAD what to do with the service.

3.7.3 Termination

When the allotted quota for a service is exhausted, the SAD shall act in accordance with the Termination-Action field set in the Quota. If the Termination-Action field is absent then the service SHALL be terminated. If the service is to be terminated, then the SAD shall send a PPAQ with the appropriate QID, the Service-Id, the used quota, and the Update-Reason set to "Client Service Termination".

If the Access Service has terminated, all other services must be terminated as well. In this case the SAD SHALL report on all issued quotas for the various services. The Update-Reason field should be set to "Access Service Terminated".

3.7.4 Dynamic Operations

Dynamic operations for multi-services are similar to dynamic operations described for single service operations. The prepaid system may send a COA message containing a PPAQ for an existing service instance. The SAD matches the PPAQ with the service using the Service-ID attribute. The new quota could differ from the previously allocated value. The SAD must react to the new value accordingly.

A disconnect message terminates the "Access Service". As such the SAD SHALL report all unused quotas by sending an Authorize-Only Access Request message containing a PPAQ for each active service. The Update-Reason shall indicate that the reason for the update.

3.7.5 Support for Resource Pools

If the PPC supports pools as indicated by setting the "Pools supported" bit in the PPAC then the PPS may associate a Quota with a Pool by including the Pool-Id and the Pool-Multiplier in the PPAQ. When Resource Pools are used, the PPAQ must not use the threshold field.

3.7.6 One-time Charging

To initiate a One-Time charge the PPC includes the PPAQ attribute in an Access-Request packet. The Access Request packet SHALL include a Message-Authenticator(80) and an Event-Timestamp(55) attribute. The Service Id field of the PPAQ identifies the prepaid service. The amount to be charged is specified using the Resource Quota and Resource Quota overflow subtypes. If the value specified is negative then the resources are credited to the user account.

The QID field SHALL be set to a unique value and is used by the PPS to detect duplicates. The Update Reason field SHALL be set to One-Time Charging. Upon receiving a One-Time charge PPAQ, the RADIUS server authenticates the user and, if successful, passes the PPAQ to the PPS. The PPS locates the account and debits or credits it accordingly. The PPS SHALL respond to the PPS with an Access-Accept message if successful, or an Access-Reject message otherwise.

The RADIUS server shall respond to the SAD with an Access Accept message. Since this is a one-time charge the SAD must not allow the session to continue. Therefore, the RADIUS server should include in the Access-Accept a Session-Timeout set to 0. Upon receiving an Access-Accept response the SAD shall generate an Accounting Stop message.

A PPAQ used for One-Time charging may appear in an Authorize-Only Access Request. This is the case when the session already exists. The PPS responds with an Access-Accept to indicate that the user account has been debited or an Access-Reject otherwise.

3.7.7 Error Handling

If the PPS receives a PPAQ with an invalid QID it SHALL ignore that PPAQ.

If the PPS receives a PPAQ containing a Service-Id, or a Rating-Group-Id that it does not recognize, then it SHALL ignore that PPAQ.

If the PPC receives a PPAQ containing a Service-Id, or a Rating- Group-Id that it does not recognize, then it must ignore that PPAQ.

If the PPC receives a PPAQ that contains a Pool-Id without a Pool-Multiplier or a Pool-Multiplier without a Pool-Id it must ignore that PPAQ.

3.7.8 Accounting Considerations

Although typically generated, accounting messages are not required to deliver a prepaid data service. When generated, accounting messages are used for auditing purposes and for billing. Accounting messages associated with prepaid data sessions should include the PPAQ attribute.

3.7.9 Interoperability with Diameter Credit Control Application

The RADIUS prepaid extensions need to interoperate with the Diameter protocol. Two interoperability scenarios exist, as follows. Either the AAA infrastructure is Diameter based and the SAD are RADIUS based, or the SAD is Diameter based and the AAA infrastructure is RADIUS based.

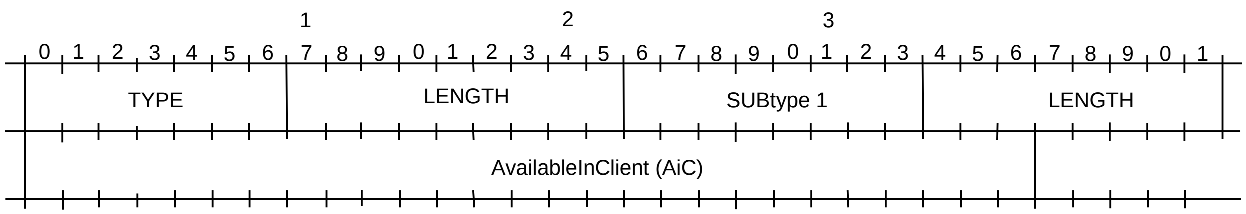
The Diameter Credit Control Application [DIAMETERCC] describes how to implement a prepaid accounting system using a Diameter based infrastructure.

4. Attributes

This section specifies the attributes that implement the RADIUS extensions for prepaid. Their general format follow that of the base RADIUS [RFC2865] and take also account current design guidelines that are proposed in the RADEXT working group. The type field of these attributes contains a value that is drawn from the type value space specified in [RFC2865]. The exact value for the type field of each attribute is to be allocated by IANA. Note that, unless otherwise specified, the format of the value field of each of the AVPs defined in this section adheres to one of the formats specified in section 5 of [RFC2865]. In particular, the labels "string", "integer", and "address" are used to indicate the format in the remainder of this document.

4.1 PPAC Attribute

The PrepaidAccountingCapability (PPAC) attribute is sent in the Access-Request message by a prepaid capable NAS and is used to describe the prepaid capabilities of the NAS.



TYPE : value of PPAC

LENGTH: 8

VALUE : String

The value SHALL be encoded as follows:

Subtype (=1) : Subtype for AvailableInClient attribute

Length : Length of AvailableInClient attribute (= 6 octets)

AvailableInClient (AiC):

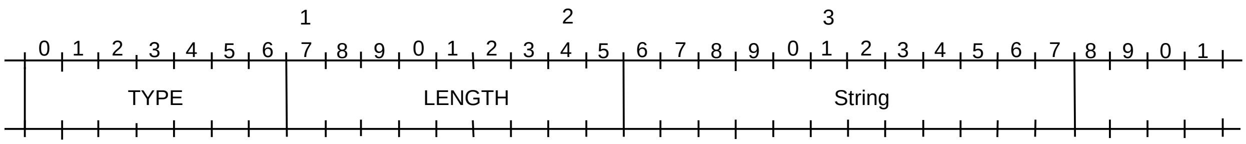
The optional AvailableInClient Subtype, generated by the PPC, indicates the metering capabilities of the NAS and shall be bitmap encoded. The possible values are as follows.

0x00000001 Volume metering supported.
0x00000002 Duration metering supported.
0x00000004 Resource metering supported.
0x00000008 Pools supported.
0x00000010 Rating groups supported
0x00000020 Multi-Services supported.
0x00000040 Tariff Switch supported.
Others Reserved.

Figure 9 - PPAC Attribute

4.2 Session Termination Attribute

The value is bitmap encoded. This attribute is included in a RADIUS Access-Request message to the RADIUS server and indicates whether or not the NAS supports Dynamic Authorization.



TYPE : Value of Session Termination Capability

LENGTH = 4

String encoded as follows:

0x00000001 Dynamic Authorization Extensions (rfc3576) is supported.

Figure 10 - Session Termination Attribute

4.3 PPAQ Attribute

One or more PPAQ attributes are sent in an Access Request, Authorize-Only Access-Request and Access-Accept message. In an Access Request message, the PPAQ attribute is used to facilitate One-Time charging transactions. In Authorize-Only Access-Request messages it is used for One-Time charging, report usage and the request for further quota. It is also used in order to request prepaid quota for a new service instance. In an Access-Accept message it is used in order to allocate the (initial and subsequent) quotas.

When multiple services are supported, a PPAQ is associated with a specific service as indicated by the presence of a Service-Id, a Rating-Group-Id, or the "Access Service" (as indicated by the absence of a Service-Id and a Rating-Group-Id).

The attribute has a variable length (greater than 8, encoded into one octet), and consists of a variable number of subtypes. Unused subtypes are omitted from the message. In the following subsections the various subtypes of the PPAQ attribute are specified.

4.3.1 Quota Identifier AVP

The value of the type field of the QuotaIDentifier AVP is defined in 5.2. The length of this AVP is 6 octets. Its value is encoded as a string. It is generated by the PPS together with the allocation of new quota. The online quota update RADIUS Access-Request message that is sent from the SAD to the PPS includes a previously received QuotaIdentifier AVP.

4.3.2 VolumeQuota AVP

The value of the type field of the VolumeQuota AVP is defined in 5.2. The length of this AVP is 12 or 18 octets. The AVP is only present if volume-based charging is used. In a RADIUS Access-Accept message (PPS to SAD direction), it indicates the volume (in octets) allocated for the session by the PPS. In an RADIUS Authorize-Only Access-Request message (SAD to PPS direction), it indicates the total used volume (in octets) for both inbound and outbound traffic. The attribute consists of a Value-Digits AVP and optionally an Exponent AVP (as indicated in the length field). The Exponent AVP, if present, SHALL NOT encode a negative number or zero.

4.3.3 VolumeThreshold AVP

The value of the type field of the VolumeThreshold AVP is defined in 5.2 and its length is 12 or 18 octets. This AVP is optionally present if VolumeQuota is present in a RADIUS Access-Accept message (PPS to SAD direction). It is generated by the PPS and indicates the volume (in octets) that shall be consumed before a new quota should be requested. This threshold should not be larger than the VolumeQuota. The attribute consists of a Value-Digits AVP and optionally an Exponent AVP (as indicated by the length field). The Exponent AVP, if present, SHALL NOT encode a negative number or zero.

4.3.4 DurationQuota AVP

The value of the type field of the DurationQuota AVP is defined in 5.2 and its length is 6 octets. This optional AVP is only present if duration-based charging is used. In RADIUS Access-Accept message (PPS to SAD direction), it indicates the duration (in seconds) allocated for the session by the PPS. It is encoded as an integer. In an on-line RADIUS Access-Accept message (PPC to PPS direction), it indicates the total duration (in seconds) since the start of the accounting session related to the QuotaID of the PPAQ AVP in which it occurs.

4.3.5 DurationThreshold AVP

The value of the type field of the DurationThreshold AVP is defined in 5.2 and its length is 6 octets. This AVP shall optionally be present if DurationQuota is present in a RADIUS Access-Accept message (PPS to PPC direction). It represents the duration (in seconds) after which new quota should be requested. This threshold should not be larger than the DurationQuota. It is encoded as an integer.

4.3.6 ResourceQuota AVP

The value of the type field of the ResourceQuota AVP is defined in 5.2. The length of this AVP is 12 or 18 octets. This optional AVP is only present if resource-based or one-time charging is used. In the RADIUS Access-Accept message (PPS to SAD direction) it indicates the resources allocated for the session by the PPS. In RADIUS Authorize-Only Access-Request message (SAD to PPS direction), it indicates the resources used in total, including both incoming and outgoing chargeable traffic. In one-time charging scenarios, the subtype represents the number of units to charge or credit the user. The attribute consists of a Value-Digits AVP and optionally an Exponent AVP (as indicated by the length field).

4.3.7 ResourceThreshold AVP

The value of the type field of the ResourceThreshold AVP is defined in 5.2. The length of this AVP is 12 or 18 octets. The semantics of this AVP follow those of the VolumeThreshold and DurationThreshold AVPs. It consists of a Value-Digits AVP and optionally an Exponent AVP.

4.3.8 Value-Digits AVP

The value of the type field of the Value-Digits AVP is defined in 5.2 and its length is 10 octets. This AVP encodes the most significant digits of a number, encoded as an integer. If decimal values are needed to present the number, the scaling SHALL be indicated with a related Exponent AVP. For example, the decimal number 0.05 is encoded by a Value-Digits AVP set to 5, and a scaling that is indicated with the Exponent AVP set to -2.

4.3.9 Exponent AVP

The value of the type field of the Exponent AVP is defined in 5.2. The length of this AVP is 6 octets. This AVP contains the exponent value that is to be applied to the accompanying Value-Digit AVP. Its value is encoded as an integer.

4.3.10 Update-Reason AVP

The value of the type field of the Update-Reason AVP is defined in 5.2. The length of this AVP is 4 octets. This AVP shall be present in the on-line RADIUS Access-Request message (PPC to PPS direction). It indicates the reason for initiating the on-line quota update operation. Update reasons 6, 7, 8 and 9 indicate that the associated resources are released at the client side, and that therefore the PPS shall not allocate a new quota in the RADIUS Access Accept message.

1. Pre-initialization
2. Initial Request
3. Threshold Reached
4. Quota Reached
5. TITSU Approaching
6. Remote Forced Disconnect
7. Client Service Termination
8. "Access Service" Terminated

9. Service not established
10. One-Time Charging

4.3.11 PrepaidServer AVP

The value of the type field of the PrepaidServer AVP is defined in 5.2. The length of this AVP is 6 or 18 octets, for IPv4 and IPv6 addresses respectively. This optional AVP indicates the address of the serving PPS. If present, the Home RADIUS server uses this address to route the message to the serving PPS. The attribute may be sent by the Home RADIUS server. Multiple instances of this subtype MAY be present in a single PPAQ AVP. The value of this AVP is encoded as an address.

If present in the incoming RADIUS Access-Accept message, the SAD shall send this attribute back without modifying it in the subsequent RADIUS Access-Request message, except for the first one. If multiple values are present, the SAD shall not change their order.

4.3.12 Service-ID AVP

The value of the type and length fields of the Service-ID AVP are defined in 5.2. The value field of this AVP is encoded as a string. This value is handled as an opaque string that uniquely describes the service instance to which prepaid metering should be applied. A Service-Id could be an IP 5-tuple (source address, source port, destination address, destination port, protocol). If a Service-ID AVP is present in the PPAQ, the entire PPAQ refers to that service. If a PPAQ does not contain a Service-Id or Rating-Group-ID, then the PPAQ refers to the Access Service.

4.3.13 Rating-Group-ID AVP

The value of the type field of the Rating-Group-ID is defined in 5.2. The length of this AVP is 6 octets. This AVP indicates that this PPAQ is associated with resources allocated to a Rating Group with the corresponding ID. This AVP is encoded as a string. A PPAQ SHALL NOT contain more than one Rating-Group-ID.

4.3.14 Termination-Action AVP

The value of the type field of the Termination-Action AVP is defined in 5.2. The length of this AVP is 3 octets. This AVP contains an enumeration of the action to take when the PPS does not grant additional quota. Valid actions are as follows. (Note that the value 0 is reserved.)

1. Terminate
2. Request More Quota
3. Redirect/Filter

4.3.15 Pool-ID AVP

The value of the type field of the Pool-ID AVP is defined in 5.2. The length of this AVP is 6 octets. This AVP identifies the resource pool that the quota included in this PPAQ is associated with. It is encoded as a string.

4.3.16 Pool-Multiplier AVP

The value of the type field of the Pool-Multiplier AVP is. The length of this AVP is 12 or 18 octets. The pool-multiplier determines the weight that resources are inserted into the pool that is identified by the accompanying Pool-ID AVP, and the rate at which resources are taken out of the pool by the relevant Service or Rating-Group. The attribute consists of a Value-Digits AVP and optionally an Exponent AVP (as indicated by the length field).

4.3.17 Requested-Action AVP

The value of the type field of the Requested-Action AVP is defined in 5.2. The length of this AVP is 3 octets. This AVP can only be present in messages sent from the PPC to the PPS. It indicates that the user or the PPC desires the PPS to perform the indicated action and to return the result. The PPAQ in which a Requested-Action AVP occurs SHALL NOT contain a QID, and SHALL contain a Service-Identifier that, possibly in combination with other AVPS, can be used by the PPS to uniquely identify the service for which the indicated action is requested.

The following actions may be requested.

1. Balance Check
2. Price Enquiry

4.3.18 Check-Balance-Result AVP

The value of the type field of the Check-Balance-Result AVP is defined in 5.2. The length of this AVP is 3 octets. This AVP can only be present in messages sent from the PPS to the PPC. It indicates the balance check decision of the PPS about a previously received Balance Check Request (as indicated in a Requested-Action AVP). Possible values are 0 for "success" and any other value for "failure" and mean that sufficient funds are available (resp. are not available) in the user's prepaid account. The PPAQ in which a Check-Balance-Result occurs SHALL NOT include a QID, because no quota is reserved by the PPS.

4.3.19 Cost-Information AVP

The value of the type field of the Cost-Information AVP is defined in 5.2. The length of this AVP is variable. This AVP is used in order to return the cost information of a service, which the PPC can transfer transparently to the end user. This AVP is sent from the PPS to the PPC as a response to a "Price Enquiry", as indicated by the Requested-Action AVP. This AVP consists of four further AVPs, as follows.

1. Value-Digits ASP: this encodes the most significant digits of the monetary value that represents the cost in question.
2. Exponent AVP: this encodes the exponent that applies to the Value-Digits AVP.
3. Currency-Code AVP: the value of the type field of this AVP is defined in 5.2. The length of this AVP is 4 octets. It encodes the currency code, as defined in the ISO 4217 standard.
4. Cost-Unit AVP: the value of the type field of this AVP is defined in 5.2. The length of this AVP is variable. It carries a UTF8String encoded human readable string that can be displayed to the end user. It specifies the applicable unit to the Cost-Information when the service cost is a cost per unit (e.g., cost of the service is $1 per minute). The Cost-Unit can be minutes, hours, days, kilobytes, megabytes, etc.

Example: the cost of 7.75 Malawi kwacha per hour would be encoded as follows. Value-Digits = 775, Exponent = -2, Currency Code = 103, and Cost-Unit = "hour".

The PPAQ in which a Cost-Information occurs SHALL NOT include a QID, because no quota is actually reserved by the PPS.

NOTES: Either Volume-Quota, Time-Quota, or Resource-Quota SHALL appear in the PPAQ attribute. A PPAQ SHALL NOT contain more than one Service-Id, SHALL NOT contain more than one Rating-Group-Id, and SHALL NOT contain both a Service-Id and a Rating-Group-Id. A PPAQ that does not contain a Service-ID or a Rating-Group-Id refers to the "Access Service". A PPAQ SHALL NOT contain more than one Pool-Id. A PPAQ that contains a Pool-Id SHALL also contain a Pool-Multiplier AVP.

4.4 Prepaid Tariff Switching Attribute (PTS)

This specification defines the PTS attribute which allows for changeovers from one rate to another during service provision. Support for tariff switching is optional for both the PPC and the PPS. PPCs use the flag "Tariff Switching supported" of the AvailableInClient subtype of the PPAC attribute in order to indicate support for tariff switching. PPSs employ the PTS attribute in order to announce their support for tariff switching. Details of this will be specified after the format of the PTS attribute has been defined.

If a RADIUS message contains a PTS attribute, it SHALL also contain at least one PPAQ attribute. If a RADIUS Access-Request message contains a PTS attribute or a "Tariff Switching supported" flag, it SHALL also contain an Event-Timestamp RADIUS attribute (see [RFC2869]).

The value of the type field of the PTS AVP is defined in 5.2. The length of this AVP is variable. It contains one or more subtypes, as follows. Every PTS AVP SHALL include a QuotaIdentifier AVP as specified in Section 4.3.1. In an online RADIUS Access-Request message sent from the PPC to the PPS, the QuotaIdentifier AVP must contain a quota identifier that was previously received from the PPS and SHALL be the same as a quota identifier of one of the PPAQ attributes included the same RADIUS message.

A PPAQ attribute that is transported along with a PTS attribute and has the same quota identifier value as the PTS attribute in its own QID subfield is referred to as the "accompanying PPAQ attribute". If a PPS receives an Access-Request message from a PPC, it associates a unique quota identifier to this request. Thus, a quota identifier also identifies a particular service.

The PTS AVP contains a number of other subtype AVPs which are specified in the following subsections.

4.4.1 VolumeUsedAfterTariffSwitch AVP

The value of the type field of the VolumeUsedAfterTariffSwitch AVP is defined in 5.2. The length of this AVP is 12 or 18 octets. The VolumeUsedAfterTariffSwitch subtype SHALL be used in online RADIUS Access-Request messages (PPC to PPS direction). It indicates the volume (in octets) used during a session after the last tariff switch for the service specified via the QID subfield and the accompanying PPAQ attribute. The attribute consists of a Value-Digits AVP and optionally an Exponent AVP (as indicated in the length field).

4.4.2 TariffSwitchInterval AVP

The type of the TariffSwitchInterval is defined in 5.2 and its length 6 octets. This AVP SHALL be present in each PTS attribute that is part of a RADIUS Access-Accept message (PPS to PPC direction). It indicates the interval (in seconds) between the value of Event-Timestamp RADIUS attribute (see [RFC2869]) of the corresponding RADIUS Access-Request message and the next tariff switch condition.

4.4.3 TimeIntervalafterTariffSwitchUpdate AVP

The value of the type field of the TimeIntervalafterTariffSwitchUpdate (TITSU) AVP is defined in 5.2. The length of this AVP is 6 octets. The PPS SHALL include this AVP if there is another tariff switch period after the period that ends as indicated by the TSI attribute. The value of the TITSU AVP in encoded as an integer, and contains the number of seconds of the tariff period that begins immediately after the period that ends as indicated by the TSI attribute. If the TITSU attribute is not present, the PPC assumes that the tariff period which ends as indicated by the TSI attribute lasts until further notice. If TITSU is specified, the PPC SHALL send a quota update before the point in time specified by the TITSU attribute (see Figure 9).

If a RADIUS message contains a PTS attribute, it SHALL also contain at least one PPAQ attribute. The PTS is associated with the PPAQ by the QID. If multiple services are supported and if the PPAQ is associated with a service as indicated by the Service-ID AVP, then the PTS refers to the tariff switch for that service. If the PPAQ does not have a Service-ID, then the PTS refers to tariff switch for the Access-Service.

If a PPC supports tariff switching then it SHALL set the 0x00000040 (Tariff switching supported) flag of the AvailableInClient subtype of the PPAC attribute that is contained in the Access-Request packet starting the session.

5. Translation between RADIUS prepaid and Diameter Credit Control

In scenarios where the service metering device uses the "RADIUS prepaid" (RPP) protocol for accounting and prepaid charging while the AAA infrastructure uses the "Diameter Credit Control" (DCC) protocol, a translation agent that enables the interoperation of both systems, is desirable. This also applies vice versa, i.e., in scenarios where the AAA infrastructure uses RADIUS and the service metering device uses Diameter.

The idea of such a translation agent would be to convert incoming RPP (resp. DCC) messages into outgoing DCC (resp. RPP) messages. It would be, in principle, desirable for the translation agent to be stateless. That is, the agent should not be required to internally maintain information about each ongoing RADIUS or Diameter session. However, under the current specification of RPP and DCC, this appears to be impossible due to a number of reasons. These include the following.

1. The transport mechanism for DCC is TCP, which requires per-session state to be maintained at both endpoints of the communication. Note, however, that, in principle, each DCC message could be sent over a dedicated TCP connection which is torn down as soon as the message is sent. This, however, is likely to be unacceptable in terms of efficiency.
2. While RPP messages encode the cumulative amount of consumed/requested resources, DCC messages carry the difference from the previous message. This means that the translation agent has to maintain the current amount of consumed/requested resources in order to be able to calculate the correct amount to be put into an outgoing message.

The translator maps each incoming RPP (resp. DCC) message into an outgoing DCC (resp. RPP) message, and possibly establishes or updates local state that is associated with the session. The translated (i.e., outgoing) message is a function of the incoming message as well as existing state that is associated with the current session.

Translation occurs on an attribute-by-attribute basis. Certain attributes are translated without consideration of local per-session state. Other attributes, namely those that are bound to a particular session, require such consideration. The translation agent has to identify the session (and possibly subsession) an incoming message belongs to in order to consult the appropriate local per-session state.

Note that certain DCC attributes cannot be translated due to their semantics not being present in RPP, and vice versa. This results in the messages, in which these attributes occur, not being delivered to their intended destination. In such cases it is desirable to inform the originator about the failure and terminate the session.

In each scenario (i.e. RPP client / DCC AAA infrastructure and DCC client / RPP AAA infrastructure), the translator operates in two directions, namely RPP to DCC and vice versa. In the following sections, the notation c->s means that the attribute in question may occur only in the direction from the client to the server. The notation s->c denotes the converse and the notation c<->s denotes that the attribute may occur in messages that are directed in either direction.

5.1 Session Identification

The translation agent has to keep per-session state in order to perform its task. A session may be identified based on the RPP identifier or the DCC session identifier. That is, the translation agent should always maintain a pair of (RPP, DCC) session identifiers and maintain the per-session state in association with that pair. This per-session state must be addressable by either of these two identifiers. Moreover, an RPP session identifier must uniquely correspond to a DCC identifier. (If this holds, the converse also holds.) Each subsession identifier within an RPP session must also uniquely correspond to a subsession identifier within its corresponding DCC session. (If this holds the converse also holds.)

5.2 Translation between RADIUS prepaid client and Diameter Credit Control AAA infrastructure

This section describes the translator in the "RPP client / DCC AAA infrastructure" case. In other words, in this section it is assumed that the client "talks" RPP and the AAA infrastructure "talks" DCC. The translator is assumed to sit somewhere in the middle and to mediate between client and server.

For each RPP AVP (i.e., AVP that is specified in the present document), the transformation into a semantically equivalent DCC AVP (if such an AVP exists), along with what per-session state the translator has to create or consult, is described. For clarity of exposition, each RPP AVP is addressed in a separate subsection. Since in this scenario, the PPC is typically the initiator a session, the focus is on the RPP AVPs.

5.2.1 PPAC (c<->s)

A DCC client is assumed to always support Volume metering, Duration metering, Resource metering, Pools, Rating groups, and Tariff Switching. Thus, if a PPAQ that indicates any of the above is sent client->server, the translator does the following: It lets message go through but remembers what exactly the client supports. If the server later requests (server -> client direction) an unsupported metering to be performed, send failure to server and cause the session to be terminated at the client.

If a PPAC indicates support for multiple services (0x00000020), the translator maps this onto a DCC Multiple-Services- Indicator AVP.

5.2.2 Service Termination Attribute (c->s)

The Diameter base protocol assumes that the client always supports dynamic session termination. If this AVP is present, the translator does not need to do anything, i.e., there exists no DCC AVP that this AVP can be mapped to. If this AVP is absent, the message in which it appears should either be discarded and originator should be informed of a failure, or the message can be passed on (without this AVP being mapped onto a DCC AVP). However, in the latter case, the translator has to remember that the client does not support dynamic termination. Thus, the translator has to initiate the normal session termination procedure with the client when (if) dynamic termination is later initiated by the server.

5.2.3 Quota Identifier Attribute (c<->s)

When quota is allocated for the first time by the DCC server, the translator has to create a QID AVP, as required by this specification. The translator later uses a QID AVP that is sent in the client-to-server direction in order to identify the corresponding DCC session. The QID has to be saved in the translator's per sessiontate.

5.2.4 Volume Quota Attribute (c<->s)

If this AVP occurs in a message that is sent in the server-to-client direction, it is translated into a Granted-Service-Unit AVP with an embedded CC-Total-Octets AVP.

If this AVP occurs in a message that is sent in the client-to-server direction, then it is translated into a Used-Service-Unit AVP with an embedded CC-Total-Octets AVP. Note that only the difference between current cumulative quota for the (sub)session and the quota in incoming messages is indicated in the translated DCC message. Local state is updated with cumulative consumed resources.

Conversely, if the server grants quota using the DCC Granted-Service-Unit AVP with an embedded CC-Total-Octets AVP, then the translation agent must translate this into a Volume Quota Attribute. Again, local state must be consulted so that the cumulative amount of octets is indicated in the Volume Quota attribute.

5.2.5 Duration Quota Attribute (c<->s)

If this AVP occurs in a message that is sent in the server-to-client direction, it is translated into a Granted-Service-Unit AVP with an embedded CC-Time AVP.

If this AVP occurs in a message that is sent in the client-to-server direction, then it is translated into a Used-Service-Unit AVP with an embedded CC-Time AVP. Note that only the difference between current cumulative quota for the (sub)session and the quota in incoming messages is indicated in the translated DCC message. Local state is updated with cumulative consumed resources (i.e., time).

Conversely, if the server grants quota using the DCC Granted-Service-Unit AVP with an embedded CC-Time AVP, then the translation agent must translate this into a Duration Quota attribute. Again, local state must be consulted so that the cumulative amount of seconds is indicated in the Duration Quota attribute.

5.2.6 Resource Quota Attribute (c<->s)

If this AVP occurs in a message that is sent in the server-to-client direction, it is translated into a Granted-Service-Unit AVP with an embedded CC-Service-Specific-Units AVP

If this AVP occurs in a message that is sent in the client-to-server direction, then it is translated into a Used-Service-Unit AVP with an embedded CC-Service-Specific-Units AVP. Note that only the difference between current cumulative quota for the (sub)session and the quota in incoming messages is indicated in the translated DCC message. Local state is updated with cumulative consumed resources (i.e., resources).

Conversely, if the server grants quota using the DCC Granted-Service-Unit AVP with an embedded CC-Service-Specific-Units AVP, then the translation agent must translate this into a Resource Quota attribute. Again, local state must be consulted so that the cumulative amount of resource units is indicated in the Resource Quota attribute.

Note that the "resource" type is application dependent. This means that a DCC application unit corresponds to n RPP application units, where n may be any real number. If n is not 1, then the RPP/DCC translator must be aware of that and translate resource units accordingly.

5.2.7 Value Digits Attribute (c<->s)

The encoding of this AVP is similar in RPP and DCC, and the value it holds may have to be evaluated in conjunction with an accompanying "Exponent" AVP. It should be kept in mind that, in RPP the cumulative amount of granted/consumed quota is typically encoded into an AVP of this type, while in DCC only the difference from a previous message.

5.2.8 Exponent Attribute (c<->s)

The encoding of this AVP is similar in RPP and DCC, and the value it holds may have to be evaluated in conjunction with an accompanying "Value Digits" AVP. It should be kept in mind that, in RPP the cumulative amount of granted/consumed quota is typically encoded into a related "Value Digits" and "Exponent" AVP pair, while in DCC only the difference from a previous message is encoded into such a pair.

5.2.9 Volume/Duration/Resource Threshold Attributes (s->c)

In DCC the concept of "threshold" does not exist. Instead, the DCC client is assumed to ask for the replenishment of quota in good time. In RPP, on the other hand, the server may optionally include a threshold AVP, as an indication to the PPC about when to ask for quota replenishment.

Thus, in this scenario, there is no need for the translator to ever include a threshold attribute into the messages that it sends to the PPC. If, however, there is a need for a threshold attribute to be present in order to avoid a possible service provision.

5.2.10 Update Reason Attribute (c->s)

The DCC AVP that is semantically closer to the Update Reason AVP than any other AVP is the CC-Request-Type AVP. This AVP indicates whether the message is which it appears is intended to indicate an "initial", an "intermediate" or a "final interrogation". Moreover, in case of the session being terminated at the client, it indicates the reason for this termination.

The following list lists the possible values of an "Update Reason" attribute, along with corresponding values for the CC-Request-Type AVP:

- Pre-initialization: No action/value defined;
- Initial Request: Typically an "initial interrogation" is triggered as a result of the reception of the message that contains this Update Reason AVP. Hence, CC-Request-Type AVP indicates "INITIAL_REQUEST";

- Threshold Reached: The reception of the message containing this Update Reason AVP typically triggers an "intermediate interrogation". Hence, CC-Request-Type AVP indicates "UPDATE_REQUEST";
- Quota Reached: The reception of the message containing this Update Reason AVP typically triggers an "intermediate interrogation". Hence, CC-Request-Type AVP indicates "UPDATE_REQUEST";
- TITSU Approaching: The reception of the message containing this Update Reason AVP typically triggers an "intermediate interrogation". Hence, CC-Request-Type AVP indicates "UPDATE_REQUEST";
- Remote Forced Disconnect: Reception of such an Update Reason indicates that the client has terminated the session. The corresponding value for the CC-Request-Type AVP is "TERMINATION_REQUEST";
- Client Service Termination: Reception of such an Update Reason indicates that the client has terminated the session. The corresponding value for the CC-Request-Type AVP is "TERMINATION_REQUEST";
- "Access Service" Terminated: Reception of such an Update Reason indicates that the client has terminated the session. The corresponding value for the CC-Request-Type AVP is "TERMINATION_REQUEST";
- Service not established: Reception of such an Update Reason indicates that the client has terminated the session. The corresponding value for the CC-Request-Type AVP is "TERMINATION_REQUEST";
- One-Time Charging: Such an Update Reason indicates that a one-time charging event is initiated by the client. The corresponding value for the CC-Request-Type AVP is "EVENT_REQUEST". Note that a "Requested-Action: AVP SHALL also be included in the outgoing DCC message. Typically, this would be of the type "DIRECT_DEBITING", or "REFUND_ACCOUNT", depending on other AVPs present in the message.

5.2.11 PrepaidServer Attribute (s<->c)

The PPC typically never sets the value of a PrepaidServer attribute. Instead, it repeats those values that it receives from the AAA infrastructure, in this scenario from the translator. This attribute is therefore not used in a translation scenario. Nevertheless, the translator must make sure that messages about the same RPP session are forwarded to the same DCC server, throughout the whole session. This may be easy to guarantee since the transport of Diameter is TCP.

5.2.12 Service-ID Attribute (s<->c)

The DCC equivalent of a RPP "Service-ID" AVP is the combination of Service-Context-Id and Service-Identifier AVPs. The translator must keep a static equivalence table of the RPP Service-ID and the corresponding DCC combination in order to correctly translate an RPP service identifier into DCC and back.

5.2.13 Rating-Group-ID Attribute (s<->c)

The DCC equivalent of a RPP "Rating-Group-ID" AVP is also called a "Rating-Group-ID". Depending on the configuration, this AVP may contain the same value on both the RPP and the DCC side of the communication. If, however, static rating groups are configured between the RCC client and the translator, and different rating groups between the DCC server and the translator, then the translator has to maintain a static translation table for the rating group identifier. In any case, the translation of a rating group AVP, is not a function of the translator's local per-session state.

5.2.14 Termination-Action Attribute (s->c)

The DCC equivalent of the "Termination-Action" AVP is called the "Final-Unit-Action" AVP. In this scenario (RPP client and DCC AAA infrastructure), a DCC "Final-Unit-Action" AVP is translated into a "Termination-Action" AVP. The following list contains the possible "Final-Unit-Action" values along with their "Termination-Action" equivalent.

- TERMINATE (DCC): This value has a direct equivalent in RPP, also called "Terminate".
- REDIRECT (DCC): If this value appears in a "Final-Unit-Action" AVP, then a "Redirect-Server-Address" AVP must also appear in the same DCC message. The translator translates these two AVPs into a "Termination-Action" with value "Redirect/Filter" and an equivalent NAS-Filter-Rule attribute (specified in http://www.ietf.org/internet-drafts/draft-ietf-radext-ieee802-00.txt).

- RESTRICT_ACCESS (DCC): If this value appears in a "Final-Unit-Action" AVP, then a "Restriction-Filter-Rule" AVP must also appear in the same DCC message. The translator translates these two AVPs into a "Termination-Action" with value "Redirect/Filter" and an equivalent Filter-ID attribute (specified in http://www.ietf.org/internet-drafts/draft-ietf-radext-ieee802-00.txt).
- In the absence of a "Final-Unit-Action" AVP, the DCC server assumes that the DCC client will ask for replenishment of quota at some suitable time. In RPP, this is explicitly conveyed via a "Termination-Action" AVP with the value "Request More Quota". Thus, in the absence of a "Final-Unit-Action" AVP, the translator in this scenario appends such an AVP into the outgoing RPP message.

5.2.15 Pool-ID Attribute (s<->c)

The DCC equivalent of a RPP "Pool-ID" AVP is also called a "Pool-ID". Typically, no translation needs to be done to the "Pool-ID" attribute.

5.2.16 Multiplier Attribute (s<->c)

The multiplier attribute, which is a pair of "Value-Digits" and "Exponent" AVPs, typically needs no translation, since the value it carries (inside a "Value-Digits" and an "Exponent" AVP) represents the rating of the service or rating group to which it refers, with respect to abstract units. As such, the same multiplier value would typically be conveyed from a DCC server to an PPC, and vice versa.

5.2.17 Requested-Action Attribute (c->s)

The "Requested Action" AVP can be directly translated into its DCC equivalent, which carries the same name.

1. Balance Check (PCC): CHECK_BALANCE (DCC).
2. Price Enquiry (PCC): PRICE_ENQUIRY (DCC).

5.2.18 Check-Balance-Result Attribute (s->c)

This attribute carries only a binary value. Hence, its translation is straightforward.

5.2.19 Cost-Information Attribute (s->c)

This attribute consists of a Value-Digits AVP, an Exponent AVP, a Currency Code AVP, and a Cost-Unit AVP. All these AVPs do likewise exist in DCC, and carry identical semantics in the context of the "Cost-Information" AVP. Thus, the translation of this attribute is straightforward.

5.2.20 VolumeUsedAfterTariffSwitch attribute (c->s)

This attribute carries the amount of octets that were consumed after a tariff change. It always appears in a message with an accompanying PPAQ attribute in which the total amount of octets (i.e., those that were consumed both before and after the tariff switch) is reported. Thus, the translation agent can compute the amount of octets that were consumed before the tariff change.

In DCC, the two amounts, i.e. the octets that were consumed before a tariff change and those that were consumed afterwards, are reported in separate Used-Service-Unit AVPs. The two Used-Service-Unit AVPs have an embedded CC-Total-Octets AVP that indicates the appropriate amount of octets. Furthermore, the Used-Service-Unit AVP that carries the amount that was consumed before the tariff switch also carries an embedded Tariff-Change-Usage AVP with the value UNIT_BEFORE_TARIFF_CHANGE (0). Similarly, the Used-Service-Unit AVP that carries the amount that was consumed after the tariff switch also carries an embedded Tariff-Change-Usage AVP with the value UNIT_AFTER_TARIFF_CHANGE (1).

6. Security Considerations

The extended RADIUS protocol described in this document is subject to a number of potential attacks, in a manner similar to the RADIUS protocol without these extensions. It is recommended that IPsec be employed to protect against certain of the attacks.

7. References

7.1 Normative References

[1] Bradner, S., "RFC 2026: The Internet Standards Process -- Revision 3", October 1996.

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

[3] Rigney, C., Rubens, A., Simpson, W., and S. Willens, "RFC 2865: Remote Authentication Dial In User Server (RADIUS)", June 2000.

[4] Rigney, C., "RFC 2866: RADIUS Accounting", June 2000.

[5] Rigney, C., Willats, W., and P. Calhoun, "RFC 2869: RADIUS Extensions", June 2000.

[6] Zorn, G., Leifer, D., Rubens, A., Shriver, J., Holdrege, M., and I. Goyret, "RFC 2868: RADIUS Attributes for Tunnel Protocol Support", June 2000.

[7] Chiba, M., Dommety, G., Eklund, M., Mitton, D., and B. Adoba, "RFC 3576: Dynamic Authorization Extensions to Remote Authentication Dial-In User Service (RADIUS)", February 2003.

[8] Adoba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "RFC 3748: Extensible Authentication Protocol", June 2004.

[9] WiMAX Forum Network Architecture Stage 3 – WiMAX Forum Network Architecture, Version 4

7.2 Informative References

[10] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J. Loughney, "RFC 4006: Diameter Credit Control Application", August 2005.

Appendix A - Example Flows

This section presents certain example flows that involve the RADIUS prepaid extensions. By no means is the intent of this section to specify or recommend business logic, rating strategies, and application-level behavior. The intent of this section is purely to illustrate some fictive scenarios and the RADIUS prepaid message flows that could be associated with these scenarios. The contents of this section should be regarded as a collection of informative examples that aim to provide guidance to implementers.

A.1 A Simple Flow

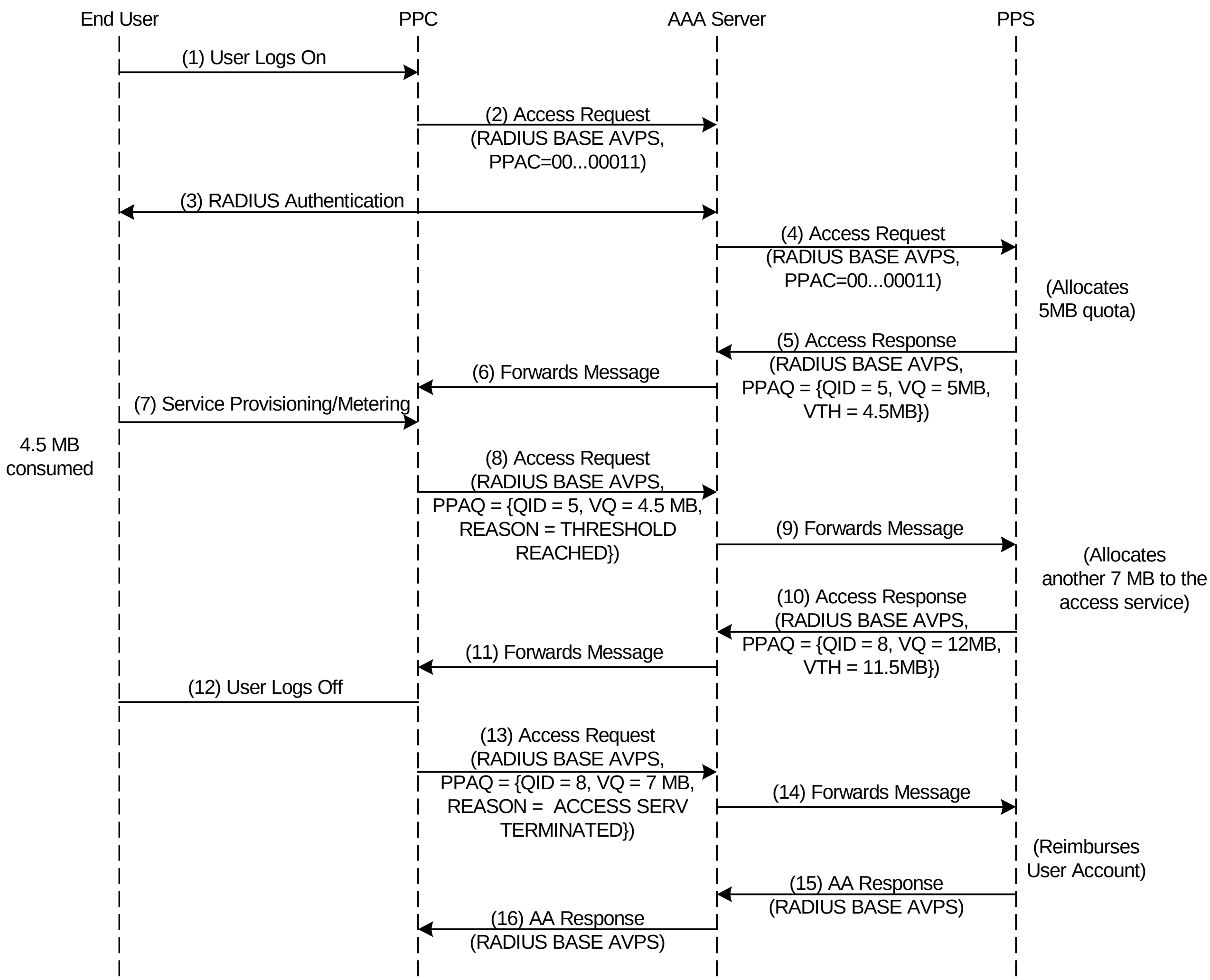


Figure 11 - A Simple Example Message Flow

The user logs on (1). The PPC sends a RADIUS Access Request message to the home AAA server (2), and includes the prepaid-specific PPAC AVP. This AVP indicates that both duration-based and volume-based metering is supported. However, it also indicated that multiple services, rating groups and resource pools are not supported. Note that, since this is not an "Authorize-Only" message, no PPAQ AVP with Update Reason="initial request" is included (see Section 3.7.1). The home AAA server then authenticates the user and authorizes the access service, as is usual in RADIUS (3). Note that the PPAC AVP is appended by the PPC in at least the last message that is sent to the home AAA server during this possibly multiple-round exchange.

If authentication and authorization is successful (in this example this is assumed), the home AAA server forwards the final Access Request to the PPS (4). The PPS identifies the user's prepaid account from the included base RADIUS AVPs, and determines the capabilities of the PPC from the PPAC attribute. Assuming that sufficient funds are available in the user's prepaid account, the PPS reserves some of these and rates the service. In this example, the PPS reserves, say, 2 Euros and determines that the access service is rated at 0.4 Euro per MB. This results in 5 MB of quota being granted. The PPS also determines that the PPC should ask for this quota to be replenished once 4.5 MB have been consumed. Thus, it creates an Access Accept message with a Volume-Threshold indication of 4.5MB. It further associates the QuotaIdentifier QID=5 to this quota reservation. This identifier can be used to later uniquely identify the prepaid session, user, account, etc. The resulting Access Accept message is sent to the home AAA server (5) and forwarded to the PPC (6).

Upon reception of message (6), the PPC provides the access service to the user and meters it accordingly.

At some point in time, the threshold is reached, i.e., 4.5MB of "access service" have been consumed by the user. At that point, the PPC generates an Authorize-Only Access Request that contains the usual RADIUS attributes and a PPAQ AVPs that reports the amount of consumed quota, and the request for replenishment, i.e., the Update-Reason= THRESHOLD REACHED (8). Note that the QID in this message is the same as the one previously received from the user's home AAA server. This message is forwarded to the PPS (9).

Upon reception of message (9), the PPS identifies the user and his account from the QID. In also determines that a prepaid session is ongoing, and that enough credit remains in the prepaid account in order for the access service to continue being provided. Since 4.5 MB have been consumed, the PPS subtracts 1.8 Euros from the user's prepaid account. The PPS decides to reserve another 2.8 euros from the user's account. (This results in 3 euros being reserved in total at this point in time.) As the access service is rated at 0.4 euros per MB, the PPS determines that another 7 MB of quota should be granted. This results in a total cumulative quota allocation of 12 MB for the access service. The PPS further calculates the new threshold value of 11.5 MB. Since this is a new quota reservation, the PPS also allocates a new QuotaIdentifier to it, in this example QID=8. The resulting RADIUS message is sent to the home AAA server (10) and forwarded to the PPC (11).

Upon reception of message (11), the PPC updates its records and continues provisioning access to the user. At some point the user logs off (12). The PPC must then report how many resources were consumed, so that the PPC can subtract the appropriate monetary amount from the user's prepaid account. To this end the PPC constructs an Authorize-Only Access Request message with a PPAQ AVPs for the access service. In this example, 7 MB were consumed by the access service in total. The PPC reports 7 MB its final message (13). This is forwarded to the PPS (14) which correlates the report, using the QID, to the previous session state. It determines, from the previous records, that the access service had consumed another 4.5 MB before (as indicated in message (9)). This means that, of the 7 MB, only 2.5 MB have not yet been subtracted from the user's account. Thus, the PPS subtracts another 1euro from the user's account and, since the session is to be terminated (REASON=ACCESS SERVICE TERMINATED), releases any reserved monetary amount.

The PPS responds with an Access Response as required by the RADIUS base specification (15). So does the home AAA server (16).

A.2 A Flow with Prepaid Tariff Switching

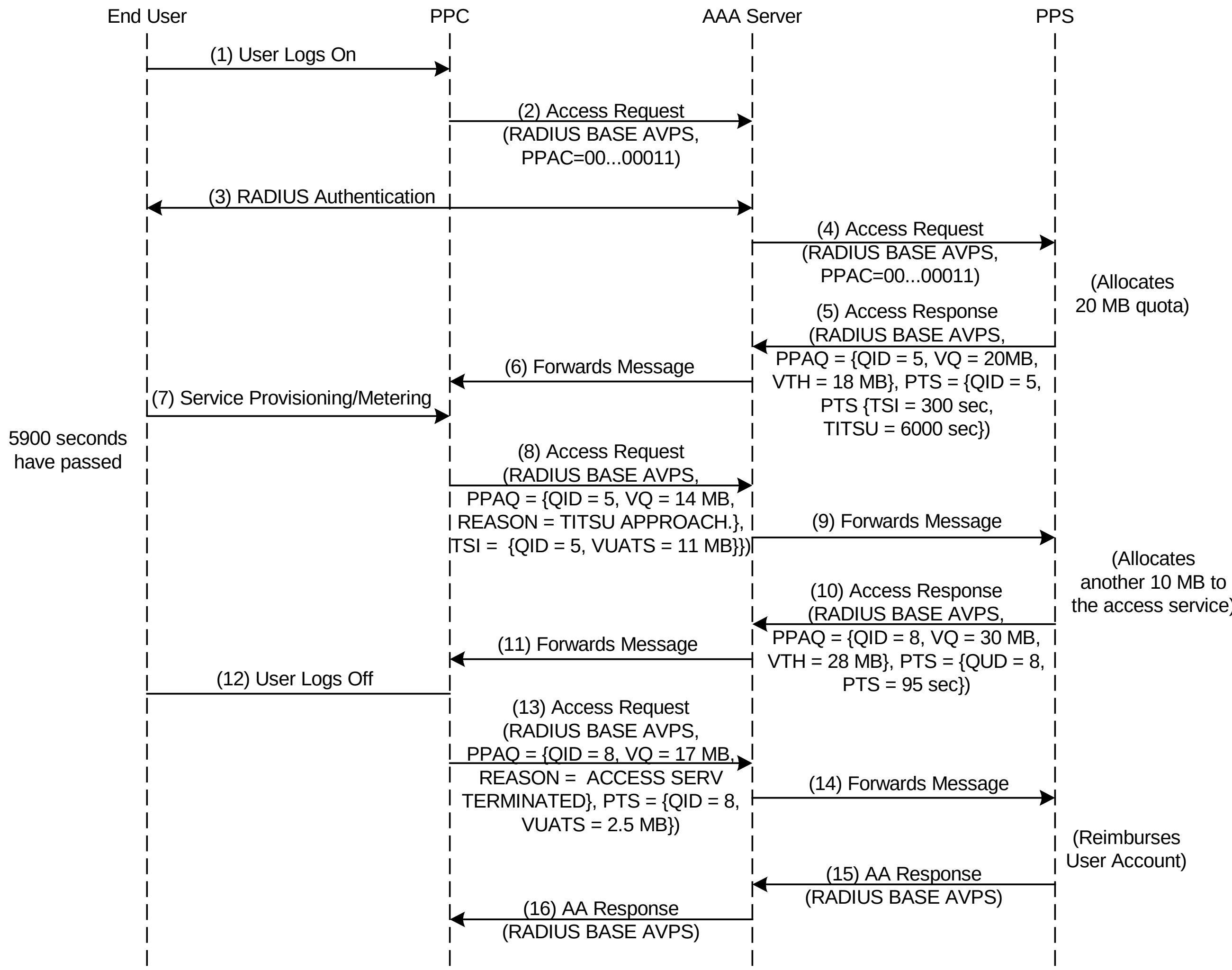


Figure 12 - Example Message Flow with Tariff Switch

The user logs on (1). The PPC sends a RADIUS Access Request message to the home AAA server (2), and includes the prepaid-specific PPAC AVP. This AVP indicates that both duration-based and volume-based metering is supported, as well as tariff switching. The home AAA server then authenticates and user and authorizes the access service, as is usual in RADIUS (3). Note that the PPAC AVP is appended by the PPC in at least the last message that is sent to the home AAA server during this possibly multiple-round exchange.

If authentication and authorization is successful (in this example this is assumed), the home AAA server forwards the final Access Request to the PPS (4). The PPS identifies the user's prepaid account from the included base RADIUS AVPs, and determines the capabilities of the PPC from the PPAC attribute. In this example, it is assumed that a tariff switch is about to occur in 300 seconds from the current time. Suppose that the access service is currently rated at 0.5 euros per MB and in the next tariff period it is rated at 0.6 euros per MB. Suppose further that a third tariff period is about to start in 6000 seconds from current time and that that access service is rated at 0.8 euros per MB in that period. The PPS then decides to reserve 12 euros from the user's account. Since it is conceivable that the user may consume all allocated quota in the (more expensive) "0.6-euro" period, the PPS reserves 20 MB of quota, and determines a threshold value of 18 MB. It constructs a RADIUS Access Accept message with a PPAQ attribute that reflects these choices, and carries a QuotaIdentifier QID=5. It further adds a PTS AVP in the message which is linked to the PPAQ via the common QID value. The PTS AVP contains a TSI attribute indicating that a tariff switch will occur in 300 seconds. It also includes a TITSU attribute with the value of 6000 seconds. This is included in order to make sure that the PPC will report the consumed quota before the "2-euro" tariff period will start. The message is sent to the AAA server (5) and forwarded to the PPC (6).

Upon reception of message (6), the PPC provides the access service to the user and meters it accordingly (7). It also keeps track of time. That is, it remembers how many octets are consumed before and how many after the tariff switch that will take place in 300 seconds.

In this example it is assumed that the user consumes the allocated quota rather slowly. In particular, nearly 6000 seconds (the value indicated by TITSU) pass without the threshold of 18 MB being reached. The PPC notices this and must therefore report usage and request the quota to be replenished despite the fact that the threshold has not been reached. In this example, it decides to do so 100 seconds before the 6000 seconds are reached. To this end, it constructs an Authorization Access Request message including a PPAQ that indicates that 14 MB have been consumed up to now. It also includes a PTS AVP in order to indicate, using the VUATS AVP, that 11 MB of these were consumed after the tariff switch. The message is sent to the AAA server (8) and forwarded to the PPS (9).

The PPS can link the message to previous session state via the QID. It now rates the consumed volume as follows. The 11 MB that were consumed after the tariff switch correspond to 11 * 0.6 = 6.6 euros and the remaining 14-11=3 MB to 3 * 0.5 = 1.5 euros. Thus, the PPS subtracts the amount of 6.6+1.5=8.1 euros from the user's account, which leads to a remainder of 12 - 8.1 = 3.9 euros being reserved.

The PPS now determines that message (9) was sent in order to replenish the quota for this prepaid session. This can be deduced from the UPDATE REASON field, which indicates that the PPC sent this message because the time indicated by the TITSU AVP is approaching. The PPS now determines that enough credit remains in the user's prepaid account in order for the access service to continue being provided and decides to reserve another 8.9 euros from the user's account. Since it is conceivable that the user will consume the 6 unused MB of quota from the previous allocation, as well as the entire quota that is to be allocated now, entirely in the "0.8-euro" period, the quota that should now be granted in addition to the previous 20 MB should be 10 MB. This is because 0.9 of the 8.9 euros are being reserved in order to "cover the worst case scenario". The fact that 0.9 euros are reserved for this purpose is due to the fact that the unused 6 MB from the previous allocation correspond to 4.8 euros (with 0.8 euros per MB). This is 4.8 - 3.9 = 0.9 euros more than the amount of funds that are still "reserved" from the previous allocation. (After this reservation, the total amount of reserved money is 8.9 + 3.9 = 12.8 euros, which corresponds to 16 (10+6) MB being consumed in the "0.8-euro" period.)

Since quotas are encoded in a cumulative way in RADIUS, the PPS includes a VolumeQuota of 30 MB into the Access Accept message (10). The PPS further calculates the new threshold value of 28 MB. Since this is a new quota reservation, the PPS also allocates a new QuotaIdentifier to it, in this example QID=8. The resulting RADIUS message is sent to the home AAA server (10) and forwarded to the PPC (11).

Upon reception of message (11), the PPC updates its records and continues providing access to the user. At some point the user logs off (12). The PPC must then report how many resources were consumed, so that the PPC can subtract the appropriate monetary amount from the user's prepaid account. To this end the PPC constructs an Authorize- Only Access Request message with a PPAQ AVPs for the access service. In this example, 17 MB were consumed by the access service in total. The PPC reports 17 MB its final message (13). This is forwarded to the PPS (14) which correlates the report, using the QID, to the previous session state. It determines, from the previous records, that the access service had consumed 14 MB before (as indicated in message (9)). This means that, of the 17 MB, only the monetary equivalent for 3 MB have not yet been subtracted from the user's account. The PPS calculates how much should be deducted from the user's account as follows. Since the VUATS AVP indicates that 2.5MB were consumed after the tariff switch, only 0.5 MB were consumed before that. Thus, the monetary equivalent is 0.5 * 0.6 + 2.5 * 0.8 = 3.6 euros. That is, the PPS subtracts 3.6 euros from the user's prepaid account. Since the session has by now be terminated by the PPC (REASON=ACCESS SERVICE TERMINATED), the PPS now releases any reserved monetary amount, in this example 12.8 - 3.6 = 9.2 euros.

The PPS responds with an Access Response as required by the RADIUS base specification (15). So does the home AAA server (16).

Remark: In this example, two tariff switches take place. In other scenarios, of course, only one tariff switch may occur. In such scenarios the TITSU AVP is not used.

A.3 Resource Pools and Rating Groups

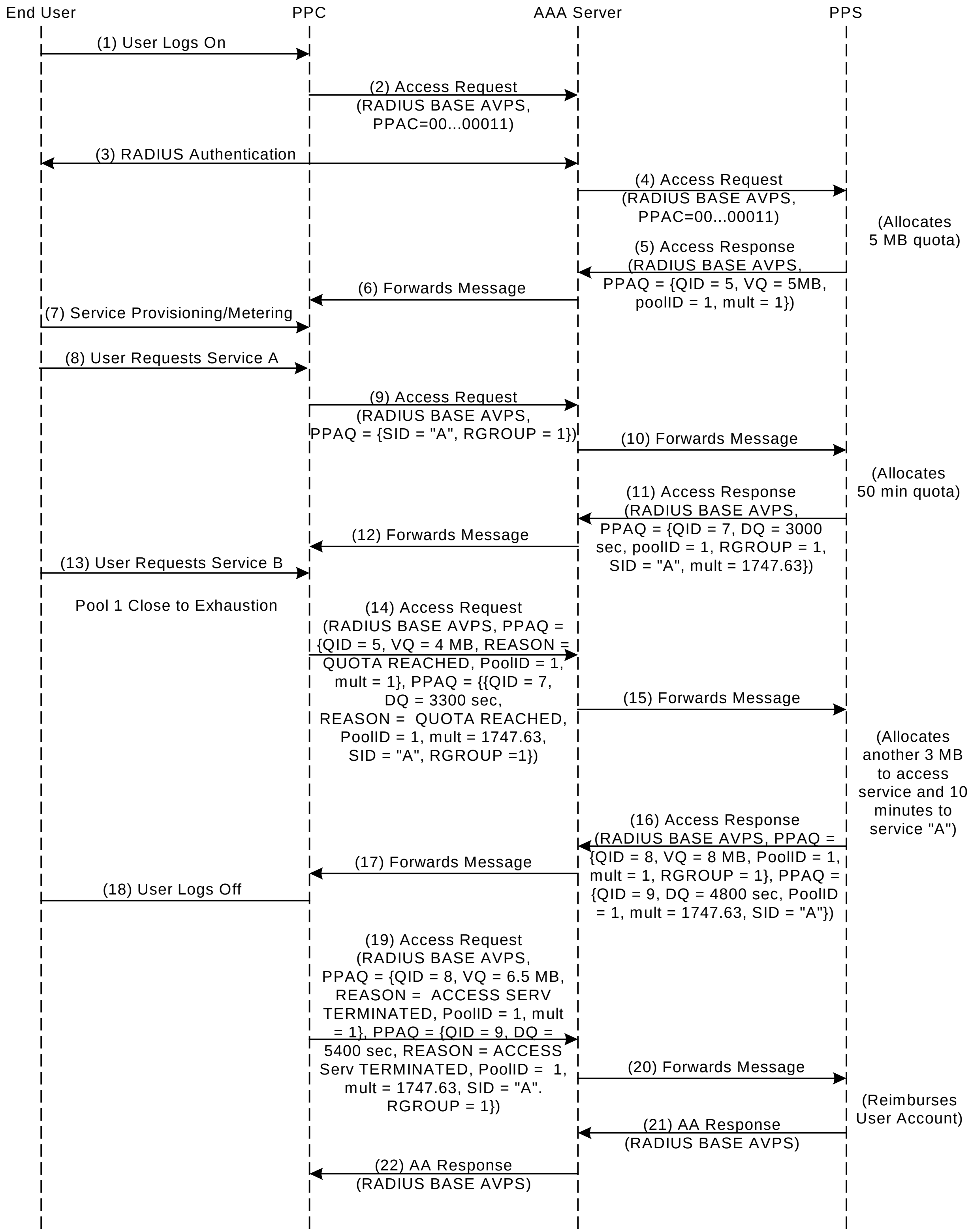


Figure 13 - Example Message Flow with Resource Pools and Rating Group

The user logs on (1). The PPC sends a RADIUS Access Request message to the home AAA server (2), and includes the prepaid-specific PPAC AVP, indicating that multiple services, rating groups and resource pools are supported. Note that, since this is not an "Authorize- Only" message, no PPAQ AVP with Update Reason="initial request" is included (see Section 3.7.1). The home AAA server then authenticates the user and authorizes the access service, as is usual in RADIUS (3). Note that the PPAC AVP is appended by the PPC in at least the last message that is sent to the home AAA server during this possibly multiple-round exchange.

If authentication and authorization is successful (in this example this is assumed), the home AAA server forwards the final Access Request to the PPS (4). The PPS identifies the user's prepaid account from the included base RADIUS AVPs, and determines the capabilities of the PPC from the PPAC attribute. Assuming that sufficient funds are available in the user's prepaid account, the PPS reserves some of these and rates the service. In this example, the PPS reserves 5 Euros and determines that the access service is rated at 1 Euro per MB. In anticipation that the user requests more chargeable services throughout this prepaid session, and since this is supported by the PPC, the PPS further associates a resource pool with this reservation, in this example PoolID=1. The PPC also specifies the multiplier = 1 for the access service. Note that, since 5MB = 5242880 octets, 1 unit in the resource pool corresponds to 5 / 5242880 euros, which is about 0.000095367431640625 Eurocents. (However, the PPC does not need to know that.) Moreover, the PPS associates the QuotaIdentifier QID=5 to this quota reservation. This identifier can be used to later uniquely identify the prepaid session, user, account, etc. The resulting Access Accept message is sent to the home AAA server (5) and forwarded to the PPC (6).

Upon reception of message (6), the PPC provides the access service to the user and meters it accordingly. That is, for every octet consumed, the PPC subtracts 1 unit (since the multiplier is 1) from the resource pool with PoolID=1.

At some point in time, the user requests another chargeable service, namely service A (8). The PPC generates an Authorize-Only Access Request that contains the usual RADIUS attributes and the Service-ID identifying service A (9). The PPC has determined that service A is rated in an identical way as at least one more service. Thus, service A has been configured to belong to a rating group, in this example the group with Rating-Group-ID=1. This identifier is included is message (9), which is then forwarded to the PPS (10).

Upon reception of message (10), the PPS identifies the user and his account from the base RADIUS attributes, the fact that a prepaid session is ongoing, and determines that enough credit remains in the prepaid account in order for service A to be provided. The PPS also determines that service A is rated at 0.10 euros per minute. The PPS decides to reserve another 5 euros from the users account; this corresponds to 50 minutes or, as encoded in the DurationQuota AVP, 3000 seconds. As service A draws from the same prepaid account as the access service, the PPS associates this reservation with the same resource pool as the previous reservation (QID=5), namely the pool with PoolID=1. Note that, in order for the abstract units in the pool to be consistent, the multiplier has to be 1747.63. This is because each second corresponds to about 0.10 / 60 = 0.00167 euros, which is about 1747.63 times the value of an abstract resource pool unit, as this was determined by the first allocation of quota to the pool (i.e. 0.000095367431640625 Eurocents). Since this is a new quota reservation, the PPS also allocates a new QuotaIdentifier to it, in this example QID=7. The resulting RADIUS message is sent to the home AAA server (11) and forwarded to the PPC (12).

Upon reception of message (12), the PPC adjusts the units in resource pool 1. That is, it first determines how much quota had been allocated to service A in the past, and subtracts this from the quota reservation found in the message. Since this is the first quota reservation for service A, there is nothing to subtract. Thus, it adds 3000 * 1747.63 = 5242890 units to the pool and remembers that 3000 seconds have been allocated to service A during this prepaid session. The PPC then provides service A to the user, and meters it against resource pool 1. That is, for every second it subtracts 1747.63 units from the pool.

At some point in time, the user requests service B (13). The PPC determines that service B is rated exactly in the same way as service A, i.e. that they belong to the same rating group, namely the one with Rating-Group-ID=1. Since this rating group has been effectively authorized by the allocation of quota with QID=7, the PPC provides service B to the user immediately. It is rated in the same way as service A, i.e. for every second provided, 1747.63 units are subtracted from credit pool 1.

At some point in time, resource pool 1 is close to exhaustion. (For example, the PPC may determine that the pool is "close to exhaustion" when has less than 10% its initial amount of units.) At that point, the PPC needs to ask for replenishment for the pool. Suppose that, at that point in time, 4MB of "access service", 45 minutes of "service A", and 10 minutes of "service B" were provided to the user. Note that this corresponds to (4*1048576) +

(55*60*1747.63) = 4194304 + 5767179 = 9961483 abstract service units from the pool. The PPC constructs an Authorize-Only Access Request message that reports the usage for the "access service" and "service A". This message contains two PPAQ AVPS, is sent to the home AAA server (14) and forwarded to the PPS (15). Note that is the message it appears that "service A" has consumed more than it was allocated (i.e., 55 minutes although only 50 minutes were initially allocated to it). This is not a problem since the PPS knows that "service A" was drawing from the same pool as the "access service" and that the "access service" did only consume 4 out of the 5 MB it was allocated.

Upon reception of message (15), the PPS subtracts 4 euros from the user's account for the "access service" and another 5.5 euros for "service A". (This includes the charge incurred by "service B" up to that point in time, although the PPS is not aware of "service B" being provisioned to the user.) The PPS then determines that sufficient funds remain in the prepaid account in order for both services to be continued. The PPS decides to reserve another 3MB for the access service and 30 minutes for "service A". This corresponds to 3+3=6 euros. Since in RADIUS prepaid the quotas are encoded in a cumulative manner, the PPAQ attribute that grants the quota for the "access service" contains a Volume-Quota AVP of 8MB (8388608 octets), which is the 5MB that were initially allocated, plus the 3MB allocated now. The resource pool identifier is, as previously, PoolID=1 and the multiplier is 1. Similarly, the PPAQ that grants quota for "service A" contains 4800 seconds (the initial 3000 plus 1800 that correspond to the 30 additional minutes). Again, the PoolID=1 and multiplier=1747.63. The resulting Access Response message is sent to the home AAA server (16) and forwarded to the PPC (17).

When the PPC received message (17) it checks how much quota has been allocated previously to the "access service". It finds that the answer is 5MB (5242880 octets); thus, out of the 8MB (8388608 octets) that are indicated by the PPAQ with QID=8, only 3MB (3145728 octets) have not yet been added to resource pool 1. The PPC thus adds 3145728 abstract units to resource pool 1 (since the multiplier is 1). The PPC then acts similarly on the other PPAQ attribute that exists in message (17). That is, the PPC determines that 3000 seconds of quota for "service A" had already been added to the pool. Thus only 1800 out of the 4800 should be additionally added to the pool. Since the applicable multiplier here is 1747.63, the PPC adds further 3145734 abstract units to the pool 1.

The PPC then continues to provide the access service, "service A" and "service B" to the user, and meters them against the pool, as previously.

At some point the user logs off (18). The PPC must then report how many resources were consumed, so that the PPC can subtract the appropriate monetary amount from the user's prepaid account. To this end the PPC constructs an Authorize-Only Access Request message with two PPAQ AVPs; one for the access service and one for "service A". Suppose that, in total, 6.5MB were consumed by the access service, 70 minutes were consumed by "service A" and 20 minutes by "service B". The PPC reports 6.5MB (6815744 octets) and 90 minutes (5400 seconds) in its final message (19). This is forwarded to the PPS which determines, from the previous records, that the access service consumed another 2.5MB (since 4MB out of the 6.5MB were already reported in message (15), and that "service A" consumed further 600 seconds. This corresponds to 2.5 + (600/60)*0.1 = 2.5+1=3.5 euros. Thus, the PPS only subtracts 2.5 out of the 6 previously reserved euros from the user's prepaid account and responds with an Access Response as required by the RADIUS base specification.

The home AAA server likewise responds with an Access Response.