Computer Networks 53 (2009) 2408–2426

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A generic accounting scheme for next generation networks

Alexandros Tsakountakis *, Georgios Kambourakis, Stefanos Gritzalis

Laboratory of Information and Communication Systems Security, Department of Information and Communication Systems Engineering, University of the Aegean, Samos GR-83200, Greece

ARTICLE INFO

Article history:
Received 1 February 2009
Accepted 25 April 2009
Available online 7 May 2009

Keywords:
AAA
Accounting
Heterogeneous networks
Performance evaluation
Security

ABSTRACT

Accounting is generally considered as one of the most challenging issues in modern and future mobile networks. As multi-domain complex heterogeneous environments are becoming a common terrain, accounting procedures performed by network and service providers have turned into a key aspect. However, in order for these networks to reliably deliver modern real-time services, they should, among other things, provide accurate accounting services, particularly billing. This work elaborates on the accounting process, proposing a novel and robust accounting system. The requirements of the proposed mechanism are defined and all the accounting scenarios that the system should cope with are examined. All the proposed accounting extensions are implemented by means of Diameter AVPs and commands. Our mechanism is generic and capitalizes on the existing AAA infrastructure, thus providing secure means to transfer and store sensitive billing data. More importantly, it can be easily incorporated into the providers' existing mechanisms regardless of the underlying network technology. At the same time, its generic nature allows for interoperability between different network operators and service providers. Through extensive experimentation, we can also infer that our scheme is lightweight, scalable, and easy to implement requiring only minor modifications to the core Diameter protocol.

1. Introduction

Accounting along with authentication and authorization comprise the well-known concept of authentication, authorization and accounting (AAA) [1]. In heterogeneous environments, where different administrative domains and diverse wired and wireless technologies are utilized, these principles are often hard and complex to correctly implement and evaluate. Specifically, accounting, which is our topic of interest, is in many cases a complicated procedure since many and often hidden aspects need to be taken into careful consideration. Such issues include Quality of Service (QoS) requirements, security, and in particular privacy of user profiles as well as Service Level Agreements (SLAs) between users and operators or between operators as the case may be. The latter issue is expected to become more complex in future networks as it follows a many-to-many relationship model. Other key parameters may be the frequent handoffs occurring inside an administrative domain (intra-network handoffs) or between different administrative domains (inter-network handoffs), and the diversity of network access technologies utilized by operators and supported by users' mobile terminals.

In this respect, a distributed, flexible, robust, secure and generic accounting system is required which is able to determine which user has acquired which services and for how long at each operator realm. Such an accounting mechanism must support the collection of usage data and provide the means to securely transfer accounting information between different network entities. Traffic and service usage data need to be constantly measured and accounting information has to be generated reliably. Moreover, accounting records must be reliably and securely transferred towards the administrative party in charge.

* Corresponding author. Tel.: +30 2273082010; fax: +30 2273082009.
E-mail addresses: atsak@aegean.gr (A. Tsakountakis), gkamb@aegean.gr (G. Kambourakis), sgritz@aegean.gr (S. Gritzalis).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.04.009

The topic of AAA services has been under constant study and attention by researchers especially in the last few years. However, as far as accounting is concerned, little work has been done as researchers mostly focus on the authentication and authorization factors as well as on security aspects of the AAA architecture rather than on accounting itself [2–9]. For instance, works [2,7] deal with authentication schemes while [8,9] are devoted to QoS and mobility issues. Accounting is considered straightforward and the Internet Engineering Task Force (IETF) draft [10] directions are followed by all current AAA implementations with only limited variations and room for modifications. Most studies in the literature so far propose accounting systems that build on standard AAA protocols such as Remote Authentication Dial In User Service (RADIUS) and are suited for specific environments and technologies. At the same time, such accounting schemes usually have been designed based on pre-defined number of users and pre-configured relationships between users and existing network providers [11–13]. In our opinion, though, accounting should be performed in a more generic way thus avoiding the limitations stemming from the underlying network access technology, the specific AAA protocol being utilized, the population of the users/providers, etc.

The remainder of the paper is structured as follows: The next section presents related work in the topic focusing on accounting mechanisms. Section 3 provides basic background information regarding the current AAA protocols and the main principles of accounting procedure. Section 4 identifies the different accounting scenarios our system should be able to handle. Section 5 presents the proposed architecture in detail and discusses a real-usage scenario. Since our scheme is built around the Diameter protocol, Section 6 elaborates on the proposed architecture and describes our accounting extensions in terms of Diameter Attribute-Value Pairs (AVPs) and commands. Our test-bed architecture and performance evaluation are given in Section 7. The last section concludes the paper and gives some directions for further research.

2. Related work

Though limited, the issue of accounting as part of the AAA framework is not completely ignored by researchers. In [14,15] accounting is recognized as an essential task for commercial service usage, while generic accounting systems are regarded essential to cope with present and future ubiquitous challenges. The authors identify accounting requirements and define the roles and relationships between entities participating in mobile networks. They also propose an accounting system based upon three discrete roles that an Authentication, Authorization, Accounting, Auditing and Charging (A4C) Server may have when responsible for the accounting process. The communication between the A4C servers is performed via the A4C protocol, which is specified as an extended Diameter protocol. The accounting system the authors propose is seconded by a Configuration Repository in order for the operator to store the required information for business processes (e.g., user profiles and SLAs). A Generic Data Storage also exists to store the accounting and charging records generated by the A4C servers. In [15] the authors define some accounting extensions to the standard Diameter protocol and study both intra-A4C and inter-A4C handoff scenarios. In case the user shifts to a foreign administrative domain the same principles apply and the accounting management treats the new handoff as if it occurred inside the home administrative domain.

The work presented in [16] studies the issue of accounting management for session mobility in ubiquitous environments. Specifically, this study considers an example scenario where a user transfers a running session between different devices. Session Initiation Protocol (SIP) is used for providing session transfer service and an extended Diameter protocol handles the accounting requirements. The authors describe the required interactions between the signaling and the accounting protocols to support session mobility and present a use case scenario.

The last work [17] presents a custom-tailored network architecture that consists of SQL database servers for offering access control and accounting in large WLAN systems. The proposed architecture relies upon the MySQL master/slave chain replication and is destined to provide high levels of reliability, scalability and availability. The authors define the communication framework as well as all message flows between the engaged network elements and the corresponding database servers. It is worth noting that none of the aforementioned works deals with performance issues, i.e., none of them provides performance results either theoretical or experimental.

Also, in this section we would like to summarize and clarify the contribution of this paper compared to our previous work. The theoretical background of this paper lies on our previous work already presented in [15]. However, as discussed further down in Section 5.2, here we substantially revise our initial architecture by incorporating several changes and improvements. Also, we elaborate on the enhanced architecture by extensively describing all message exchanges between network elements. Moreover, we emphasize on the proposed changes to the base Diameter protocol in terms of commands and AVPs. This analysis, provided in Section 6, is also introduced in this paper. The test-bed experimental results included in Section 7 are also novel and to the best of our knowledge the first of its kind in the AAA literature. As already pointed out in the previous section, several works in the AAA literature do provide performance evaluation facts but none of them is devoted to accounting.

3. Background

3.1. AAA protocols

The generic AAA scheme as described by IETF defines the necessary mechanisms for dispatching the functions of Authentication, Authorization and Accounting. Every network operator integrates AAA into his mechanisms thus providing interworking between different operators [19]. A number of AAA-enabled servers (called AAA servers) are scattered throughout the network in order to provide the required functionality. Depending on the network, the AAA server has only one role performing a single task; say validating authentication credentials as part of the authentication process for users requiring 802.11 network access, or be responsible for all AAA services. The required communication is performed by utilizing an appropriate protocol as the case may be.

At present, several AAA protocols [19,20] have been proposed. Incipient protocols, such as [21,22], were immature having several limitations. For example, most of them present several difficulties when integrating into networks that utilize new technologies. That is, technologies that were not present prior to the formation of the protocols. The RADIUS protocol [23,24] was designed by IETF for transferring authentication, authorization, and accounting data between a Network Access Server (NAS), which determines a RADIUS client, and the corresponding RADIUS server holding the information to authenticate and authorize a user. A RADIUS server can also act as a proxy, i.e., a client to other RADIUS servers. Originally, RADIUS was limited to support dialup connections, but today is able to support different situations and technologies. Nevertheless, several shortcomings and weaknesses have been discovered in RADIUS [25] and because of them it is no more considered widely acceptable as a modern AAA protocol. Furthermore, RADIUS is only applicable over TCP.

The Diameter protocol [26,27] was defined as a successor to RADIUS, in order to overhaul known deficiencies of its predecessor [25]. Diameter fully satisfies the requirements for accessing heterogeneous network technologies, including wireless packet data technology and distributed security models for multi-domain and roaming scenarios. Diameter consists of a base protocol that defines header formats, and security extensions as a number of mandatory commands and AVPs. The base protocol is session-oriented based on the Peer-to-Peer (P2P) model. Besides TCP, Diameter operates over Stream Control Transmission Protocol (SCTP) as a transport protocol. Information is exchanged by means of AVPs. Different extensions to the base protocol allow the utilization of different network access technologies by defining special command codes and AVPs. The NAS server requirements (NASREQ) extensions are able to support RADIUS authentication protocols, Extensible Authentication Protocol (EAP) methods [28], and authorization needed by NAS services. Also, Mobile IP extensions define AVPs to support Mobile IP across disparate administrative domains. This enables a Diameter server to authenticate, authorize, and collect accounting information for services requested by a mobile node. The same accounting extension defines a set of generic accounting AVPs that can be used for all services supporting real-time accounting as well.

Other related but less accredited protocols not discussed here include the Common Open Policy Service (COPS) [21] and the Terminal Access Controller Access Control System (TACACS) [22].

3.2. Accounting

Once a user successfully authenticates himself with the network and gains the appropriate authorization privileges he is granted access to network resources. From that time on, the user activities need to be constantly tracked and metered in order for the network operator to calculate and accordingly charge the user. As already mentioned, this procedure is called accounting and is extremely important for both the customer, in order to keep his faith towards the network operator, and the network operator or service provider as his revenue relies upon it. The main purpose of the accounting procedure is to bind user-related activities with accounting metrics. The latter may be the overall time the user spent connected to the network, the kilobytes of data downloaded, or even some pre-defined tariffs correlated with a specific service the user acquired.

In a typical accounting scenario several entities involved. First, the customer (or subscriber) who is actually a user utilizing the appropriate device to gain access to a network. The customer holds a subscription with a network operator who is responsible for offering and supporting network access to his customers. This operator is called the Home Operator (HO) and is the only party holding a user profile, consisting of detailed information regarding the user. A user SLA is also required to provide additional information for the services the user has subscribed to and to any other special parameters. Charging is also determined by the user's SLA. An external or foreign network operator, known as the Foreign Operator (FO), may be utilized in case of roaming. This allows the user to continue receiving network access outside the area covered by his HO. In most cases an FO holds a roaming agreement with the user's HO. The last party involved is called the Foreign Service Provider (FSP). This is a third party capable of providing add-value services to his subscribers. A user may have contractual agreement directly with the FSP or the services provided by the FSP may be part of his agreement with the HO. These services are in most cases charged separately, but this partial cost is finally added to the overall cost of network access by either the HO or the FO. Additionally, the role of FSP may be taken by the HO or FO, meaning that the entities granting network access may also offer a wide variety of services. In many scenarios though, especially when sophisticated services are the case, a foreign third-party service provider may be necessary. Fig. 1 depicts all participants in a typical accounting scenario along with the network connections between them. Note that in the figure the user is only connected directly with the home network as he is subscribed to that corresponding network operator only.

In all cases the HO is accountable for posing the final charges as well as the preparation of the corresponding invoice. Actually, the HO is the only party the user has direct subscription to. Also, normally, the HO is also the only entity the user trusts. The invoice will include all charges that all participating parties have gathered on behalf of the corresponding user. Revenue shares between the charging parties are defined separately according to their bilateral agreements and contracts.

Charging can be a relatively straightforward and simple process or becomes highly complicated as more and more network operators and service providers are participating into. The factors affecting the accounting procedure are bipartite. On the one hand lay the different administrative domains the user visits during handoffs. Whilst, on the other, emerge the technological variations, as more and more different technologies are available to the user, but not all network operators offer complete support for all of them.

4. Accounting scenarios

The simplest accounting scenario involves the employment of a device to acquire a service offered by the network provider the user holds a contractual relationship with. Complexity arises in many scenarios including those the subscriber requires the acquisition of services offered by a different access technology domain. Also, this may happen due to several other reasons, for example, when the user is forced to change network access provider because the HO experiences connectivity problems or lack of connectivity in a given area (e.g., the user is roaming abroad).

More specifically, think of a scenario where a user utilizes his 3G mobile device in order to receive high-speed Internet connectivity from a specific network operator that allows for a video conference. As the user roams from place to place the required handoffs take place. Eventually, he may find himself in an area where 3G (i.e., UMTS) coverage is not available by his network operator. This scenario is frequent in places outside big residential areas. Upon reaching this area the user's connection will shift towards the old 2.5G (i.e., GSM/GPRS) system which is also provided by his network operator. The video conference will end abruptly while the user will experience a decreased Internet connection speed. At a later stage the user may move towards another UMTS cell again or even find an 802.11 hot-spot deployed by his HO to offer high-speed local connectivity.

Shifting from one technology to another must be fully transparent to the user but apart from other issues such as continuity of connection, fast handoffs, security considerations or any other technological bottlenecks and challenges, the procedure of accounting is also affected. Since the utilization of different access technologies implies possible variations in charging according to SLA's and QoS agreements, preparing the final invoice may not be an easy task.

In a similar way, apart from selecting different technological means to access the network, the user may also need to shift from one network operator to another and thus move to a different administrative domain. This applies to scenarios where either home provider connectivity issues temporarily arise or connectivity in a specific area is not present. Such cases also raise several difficulties and complications in the process of accounting. This is mainly due to the fact that the user usually holds a contractual relationship with one network operator only, but he may require services offered by many others his HO is keeping roaming agreements with. In this case, the charging process needs to take into consideration the details regarding the revenue derived from the roaming agreement between the two network operators as well as possible extra charges the user may be requested to pay.

In this respect, technological handoffs and administrative domain handoffs often become intertwined as the user enjoys the benefits of the new 4G heterogeneous environments and their services, which in turn are derived from the use of innovative network technologies. Thus, modern accounting systems need to meet and satisfy several challenges and demands in order to provide robust, secure and foolproof services to network operators. The possible

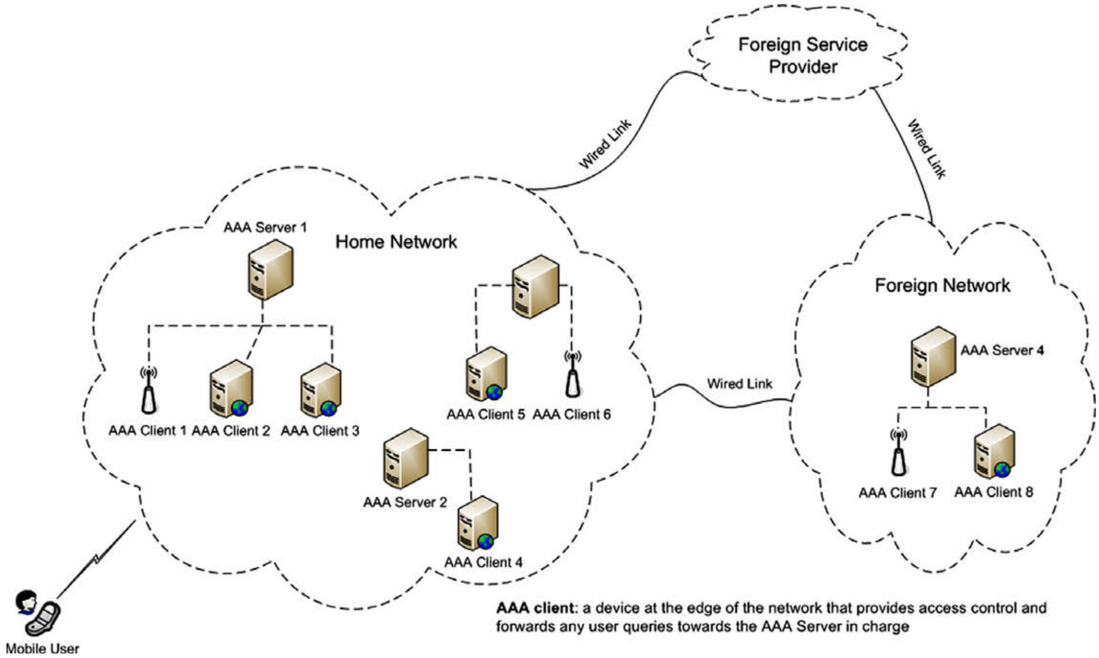


Fig. 1. Generic AAA Network architecture.

accounting scenarios can be classified into the following four distinct categories:

1. *Same administrative domain, same technological domain:* The user has a contractual relationship with his HO, accesses the network through the HO using, e.g., his UMTS enabled device. This is the simplest accounting scenario where the user is charged solely by the HO.
2. *Same administrative domain, different technological domain:* The same as before, but when the user needs to shift from a 3G network to a WLAN (also provided by the HO) he can do so since his device also supports 802.11 technology, for example. In this case, the user is charged by his HO for network access and services consumed, with possible change in pricing when shifting from 3G to WLAN.
3. *Different administrative domain, same technological domain:* The user has a contractual relationship with his HO. He accesses the network through HO but needs to shift to a different FO that supports the same technology. In this case the user is charged by his HO for the time he remains under the HO coverage and from the FO for the time he spends under FO coverage. The accounting data the FO collects are transmitted to the HO for the preparation of the final invoice. Increased tariffs are usually imposed when the user shifts to the FO. As far as the operators are concerned they most probably have a long-time bilateral agreement covering revenue issues and financial share between them.
4. *Different administrative domain, different technological domain:* The user has a contractual relationship with his HO. He accesses the network through the HO but shifts to different FOs and to different network access technologies. The same principles as in the aforementioned scenario apply with only difference the possibility of dissimilar charges involved due to the transition from one technology to another.

In case the user requests services from an FSP then he is charged by his HO for network access as well as from the FSP for service usage. The FSP collects accounting data regarding service utilization by the user, calculates the price according to his particular pricing scheme and sends the information back to the HO, since the user expects a single bill from his contractual operator. The agreement between the HO and the FSP determines the financial share between them, if any. Note that as the user may shift from the HO to FO while accessing a service provided by an FSP more complicated scenarios may arise as well.

5. Proposed accounting architecture

5.1. General issues and requirements

Every new accounting system should take into consideration all the parameters related to: (a) the heterogeneous environment, (b) the multi-network operator relationship model, (c) the existence of many innovative technologies possibly incompatible with each other, and (d) the large number of mobile user population as each one of them is a potential customer or service-requiring entity for all existing network operators and service providers. A solution based on well-structured, pre-defined contractual-like relationships between users and several likely network operators is far from desirable. This is because in the real world – where millions of mobile users constantly roam between disparate administrative domains, utilize different technologies and enjoy several services at the same time – pre-defined relationships is practically an infeasible solution. Thus, a more flexible and scalable approach should be proposed that relies on temporary or on-the-fly creation of the required relationships and transfer of data.

Apart from the practical difficulties this problem presents, security considerations should also be treated as of major importance for any new accounting system. User personal information (user profiles or contexts), SLAs and invoices issued to customers contain private data and need to be protected. Under these circumstances privacy should be guaranteed and mechanisms to reliably and securely store information or send accounting data from one network operator to another should be utilized. Again, it is stressed that only the network operator the user has contractual relationship with (the HO) is allowed to store and process personal information about that particular user. In a nutshell, the desirable requirements a new accounting system must meet are the following:

1. *Generic*: The new accounting system should be applicable regardless of the underlying network access technology used. In this way forthcoming technologies should be easily incorporated.
2. *Distributed*: The magnitude and complexity of current accounting demands can only be tackled with distributed architectures. A distributed architecture also helps mitigate future problems and technical failures.
3. *Secure*: Security is critical during the accounting procedure. Data privacy, confidentiality and integrity should be ensured. Also, of utmost importance is the protection of user personal information. Private personal data should be safely stored and never be transmitted to any party other than the one the user has a contractual relationship with. At the same time, accounting data regarding a user should be securely and reliably communicated between the administrative parties involved. Therefore, the confidentiality and integrity of accounting data in transit are of major importance here.
4. *Transparent to users*: Users must receive a single bill regardless of the number of operators or other charging parties are involved in the process of accounting.

5.2. Analysis

In our previous work [18] we proposed a novel architecture to successfully fulfill the security and other requirements pointed out in Section 5.1 of this paper. Nevertheless, during the implementation and evaluation of a prototype system destined to measure and prove the effectiveness of the proposed architecture several issues arose. In this work we present our revised architecture incorporating all the required amendments and additions. Throughout this section we point out all changes providing a detailed explanation and justification of their necessity. To further explain the proposed architecture we also demonstrate the mechanism responsible for storing the user's accounting records into the corresponding database.

5.2.1. Network entities and their roles

As already mentioned, the two key factors governing the overall accounting process are the vertical and horizontal handoffs occurring as the user moves from domain to domain. A vertical handoff involves changing the data link layer technology used to access the network, while a horizontal handoff takes place between different wireless Access Points (APs) that use the same technology. A handoff occurrence may only involve the same administrative domain or happen between different administrative domains. Whenever one of these events occurs or every time the user requires a new service, charging is most likely affected and thus the accounting system should provide a mechanism to store and process the associated event. To cope with this, the proposed accounting system relies upon the creation of hierarchical discrete identification numbers (IDs). Each ID corresponds to a single occurring event. In this way all user activities can be tracked and charged, while at the same time, it constitutes the basis behind the successful fulfillment of the requirements described previously in Section 5.1.

In a typical AAA architecture several network entities collaborate to perform accounting-related activities. Apart from the user device, also referred to as *supplicant*, it is important to introduce the concepts of AAA client and AAA server. As opposed to our previous work [18] where only the role of the AAA server was taken into consideration, we now choose to clearly define the properties of an AAA client. An AAA client is a device at the edge of the network that provides access control and forwards any user queries towards the AAA server in charge [1]. It utilizes an appropriate AAA protocol (e.g., Diameter) and generates AAA messages to request authentication, authorization and accounting services on behalf of the user. In most cases a NAS undertakes the role of the AAA client, but in fact any network device destined to provide network access, regardless of the utilized technology, can act as an AAA client. This is true provided that an AAA protocol is properly installed and configured. It is important to note that in the proposed architecture AAA clients are of major significance as they are the only parties that gather accounting metrics (also referred as accounting data).

AAA servers on the other hand are deployed by network operators inside their administrative domain. Such servers receive AAA messages from AAA clients and perform the authentication, authorization and accounting services, respectively. Regarding our accounting system, AAA servers are responsible for handling the accounting process, calibrate accounting settings on AAA clients and transform accounting metrics into accounting records. Similarly, in a typical AAA architecture, an AAA server can perform any or all of the AAA services and often collaborate with neighboring AAA servers (e.g., by proxying AAA messages). Hence, several AAA clients may be serviced by one or more of the AAA servers deployed by the network operator.

AAA servers, according to their specific role regarding accounting, can be referred as either Accounting servers or Billing servers. As described in the generic AAA model [1] the main role of an Accounting server is to process the accounting data received from the AAA client. This processing may include summarization of interim accounting information, elimination of duplicate data, or generation of session records. Accordingly, the Billing server typically handles rating and invoice generation, but may also carry out auditing, cost allocation, trend analysis or capacity planning functions. Real-life network operators usually choose to distinguish between the role of Accounting and that of Billing server. This is because a considerably large number of Accounting servers are required as opposed to Billing servers, where all relative activities can be performed by a single machine. Concerning our proposed model, both Accounting and Billing servers maintain the same attributes and properties as in the generic AAA model, but their role is not distinguished as we choose to use the notion of AAA server for all accounting-related activities. It is important to keep in mind that AAA clients gather accounting metrics while AAA servers transform these accounting metrics into accounting records.

In this context, the proposed accounting system relies on typical AAA servers already used by network operators. According to our model, during the accounting process, an AAA server can take either the role of the Root server or that of the Administrative server. The Root server is an AAA server inside the home domain responsible for the AAA client (usually a NAS). That is, the server that has already successfully completed the authentication and authorization process and granted access to the user. In several scenarios the AAA server that the user initially attaches to might not be suitable to provide the required services and thus AAA requests may be proxied to a new AAA server that will be granted the role of the Root server. Therefore, as opposed to our previous work, the Root server should no longer be defined as the server the user first attaches to, but as the one that has successfully completed authentication and authorization requests via the AAA client on behalf of the specific user. From now on, in terms of Accounting, the Root server will be responsible for that specific user. Therefore, the same AAA server will be used for collecting accounting records from the respective Administrative servers throughout the entire user session. In a nutshell, the Root server initializes and terminates the accounting process for a given user.

5.2.2. User ID mapping scheme

Upon granting network access to the user the Root server creates a unique identification number (ID) and at the same time stores in the corresponding database a record mapping the newly created ID with the actual user ID as shown in Fig. 2, case A. The actual user ID may be the user's International Mobile Station Identifier (IMSI) (this is a permanent ID), a temporary ID such as Network Access Identifier (NAI) [29], or even a pseudonym. The first ID that the Root server creates is called *Master ID*. This ID can be altered, updated or deleted only by the Root server. The Root

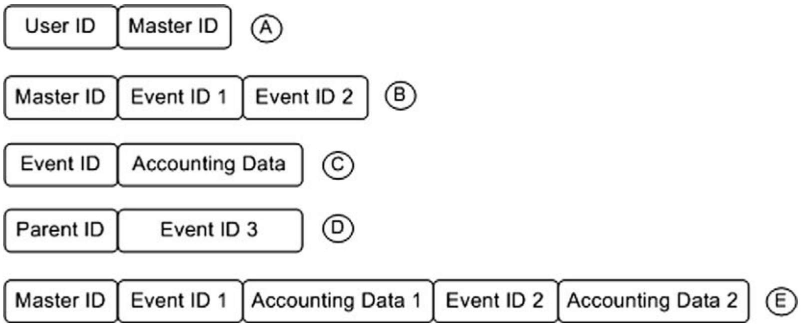


Fig. 2. (a) Mapping user information with the Master ID, (b) mapping the Master ID with Event IDs 1 and 2, (c) mapping the event ID with accounting data , (d) mapping the Parent ID with Event ID 3 and (e) record stored in the database by the Root server.

server is also responsible for accepting frequent requests for accounting information by the Administrative servers as well as for the preparation of the final invoice to be sent to the subscriber. Any Administrative server on the other hand, will respond to an accounting query sent by a Root server.

The Administrative server is initially the same as the Root server. As the user moves from one domain to another, handoffs occur and the user may need to attach to a different NAS or even require the services of a new AAA server. For example, when moving inside the area of a WLAN the user's device may shift from one AP to another. Consequently, the Administrative server is the local AAA server, which is at the given moment responsible for the user. It is important to note that the Administrative server can be an AAA server that is located in the administrative domain of a foreign network operator. This server is responsible for collecting accounting records and keeping track of the user activities while the user remains under its supervision. Practically, the Administrative server configures accounting parameters on the AAA client and orders the initiation or termination of the accounting data collecting procedure from the same entity. Moreover, it receives all accounting metrics that are later on converted to accounting records to be sent towards the Root server. While the user moves from one AAA client to another the current Administrative server terminates accounting on the old AAA client and asks the new one to take charge and initialize the proper accounting procedures. Each Administrative server holds only limited information about the actual user. Specifically, it keeps only the required SLA parameters needed for charging as well as a reference to an ID sent to it by the previous Administrative server.

Each time the user initializes an event that needs to be tracked and metered the Administrative server will create a new unique ID, called *Event ID*, mapped to that particular event. Each event ID must be globally unique, so for instance, it could take the form of $\{Administattive\_Server\_Name\_or\_IP \| Event\_ID \| Current\_time\_in\_milliseconds\}$. The server will securely store in the corresponding database the correlation between the newly created event ID and the received *Reference ID* as shown in Fig. 2, cases B and D. For multiple events created by the same user the corresponding IDs will be utilized to track all user activities. A database is accessed to securely store records binding the user event IDs with accounting data as shown in Fig. 2, case C. When a user leaves the current Administrative server, or when required for other purposes, all gathered accounting data will be sent towards the Root server. The Root server will eventually combine all events and store an accounting record in the form of that shown in Fig. 2, case E.

The notion of the Reference ID contains two discrete entities. While the user remains inside the administrative domain of the home network the Reference ID is the Master ID created by the local Root server. On the other hand, while the user remains under the coverage of a foreign administrative domain a *Parent ID* takes the role of the Reference ID, as described further down. For a given user the same Master ID is used during a session while several Parent IDs may be utilized. This happens because it is vital that every time a foreign AAA server is involved a new ID should be used as a reference. The Root server in the case of the Master ID and the previous Administrative server in the case of the Parent ID can be extracted from the Master and Parent ID values correspondingly, so that the current Administrative server knows where to send the accounting records.

In case the user moves to the domain of a foreign network operator the same principles apply but security requirements suggest the use of a new identifier other than the Master ID to be utilized as a reference for any new event IDs. This is fulfilled by a new identification number we call *Parent ID*. The Parent ID is an ID created by the Administrative server in the home domain to be sent to the new Administrative server inside the foreign domain. This Parent ID will thereafter be used as a reference to any newly created event IDs. The Parent ID notion serves a dual purpose. First, it constitutes a completely new reference neither created nor relevant to the initial Master ID or the actual user ID. Thus, even when the Master ID is used as a reference to event IDs it remains inside the home domain and has never become available to a server inside the FO. This assists to further protect the user real identity and other related confidential information. Secondly, the Parent ID helps to clearly distinguish the role of the Parent ID and the Master ID. Note that in our previous work [18] both entities were referred to as Master ID although their role was distinctly defined.

At this point another amendment to our initial architecture [18] was considered crucial and is introduced in this work. Our previous work suggested that while remaining inside the foreign domain all vertical handoffs should be treated as if the user was inside the home domain. However, we now require that in case of a vertical handoff it is preferable that the new Administrative server does not contact directly the previous Administrative server inside the foreign domain. Instead, request all necessary information to be sent during authentication by the latest Administrative server inside the home domain. This server will send the required SLA and any other charging instructions as well as the user's Parent ID. The latter is a new Parent ID different than any other previously used for the same user. As the user terminates all actions, or when asked for other purposes, each engaged Administrative server inside the foreign domain that tracked user activities for some time will send the relevant accounting records to the corresponding Administrative server inside the home domain. The Administrative server will later on forward them along with its own collected accounting records to the Root server inside the home domain.

It is also noted that it is not necessary for the Administrative server inside the home domain to wait for the accounting data from the server(s) placed inside the foreign domain(s). On the contrary, interim data could be sent to the Root server, having them updated several times. As soon as the complete accounting records from foreign Administrative servers are available they may be forwarded to the Root server. The above amendment is expected to add both a bandwidth and a time penalty compared to our previous mechanism but should be regarded rather minor as it is reported in the performance evaluation section further down. On the other hand, this new scheme offers a greater level of security and complies with the respective common practices and current standardized mechanisms [26,30]. Finally, if an FSP interferes, the current Administrative server will generate a new Parent ID to be used as a reference to the newly created event IDs.

5.2.3. Security and privacy considerations

Considering the security analysis of the proposed architecture it is stressed that the confidentiality and integrity of the accounting data in transit are guaranteed since the wired links between providers are normally secured by IP secure (IPsec) or sometimes by Transport Layer Security (TLS) protocol. For instance, modern AAA protocols such as Diameter suggest the use of IPsec for data exchanges inside the same administrative domain and TLS in case of data crossing administrative domains [26]. Of course, additional security mechanisms either symmetric or asymmetric can be employed but this is out of the scope of this paper. On the other hand, privacy is also assured considering the fact that no user's direct or indirect personal information (permanent identity, Master ID, etc.) leaves the possession of Root AAA server in charge inside the home domain (see Section 5.2.2).

Another important concern is the protection of user's location privacy. It is known that while a handoff allows for seamless service delivery to mobile users, it seems that it comes with a cost in their location privacy [31]. For instance, with the use of the Context Transfer Protocol (CTP) [32] to support seamless handoffs, every administrative domain is aware of the previous and the next administrative domain of the user, without excluding itself. This means that every domain can track a part of the user's movement. Even worse, the user's movement can be completely tracked, given that some administrative domains collude. Note that this does not imply that all administrative domains in the path of the user movement are required to collude for such an attack, but every second domain in that path.

The same issue is relevant to accounting event IDs required by our mechanism. An eavesdropper may manage to associate some event IDs with a given user so as to track his movement. However, this attack is very difficult to implement because (a) as already mentioned all communications between AAA servers are IPsec/TLS protected; so having access to an event ID means getting access to the AAA itself, (b) the generated event IDs can be sufficiently random making it very difficult for passive or active eavesdroppers to correlate them with the actual identity of a user.

5.3. Real-usage scenario

To better understand the proposed accounting system this section presents a real-usage typical scenario and demonstrates how the proposed system will respond. For the need of the scenario all matters regarding user authentication and authorization as well as other security and network considerations are left out. Only accounting issues will be discussed. The user is supposed to carry a mobile device which supports GSM and UMTS networks as well as 802.11. He also holds a contractual relationship with a network provider.

Fig. 3 depicts the network architecture of the described scenario. The Home domain represents the administrative domain owned by the HO. It offers three AAA servers and the respective AAA clients. AAA server 1 handles queries transferred from AAA clients 1, 2 and 3. Likewise, AAA server 2 is responsible for AAA client 4, while AAA server 3 is accepting AAA messages from AAA clients 5 and 6. Respectively, the foreign domain represents an FO the user needs to attach to for roaming purposes. Also, keep in mind that between the Home and the foreign domain a contractual agreement or a pre-defined relationship is not required [26].

The actions (steps) performed by the user upon successful connection to the network are the following:

1. The user connects to the WLAN utilizing his dual UMTS/802.11 device. He is connected through the corresponding hot-spot that beacons in the area. The hot-spot has been deployed and is being managed by the HO for high-speed network access.
2. The user is accessing the Internet by visiting several web-pages.
3. The user ends the current Internet session and moves outside the area. As a result, his device is now forced to connect to the UMTS network provided by the same network operator.
4. At this point the user initiates a new Internet session and visits the network operator web-page where he buys and downloads one mobile game.

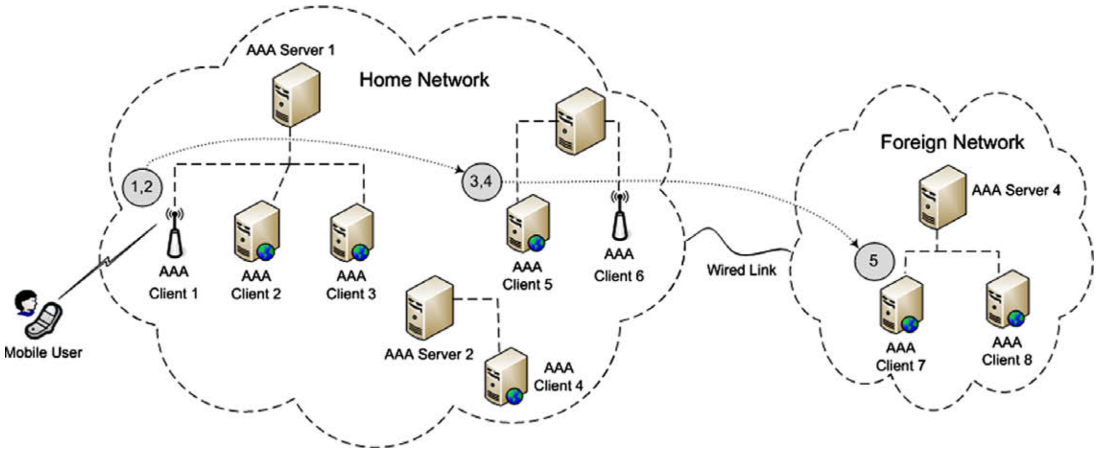


Fig. 3. Scenario architecture and steps.

5. The next day finds the user abroad without network coverage from his HO. His device connects to a foreign network operator through UMTS. The user makes two phone calls.

In order for the user to connect to the WLAN, one of the three AAA clients inside the Home network needs to react and serve the user's request. This is done by AAA client 1 (e.g., the local AP) that forwards authentication requests to the AAA server in charge. Upon successful authentication, and in terms of accounting, AAA server 1 takes over the role of the Root server for that specific user, while AAA client 1 will collect the corresponding accounting metrics. The server retrieves the user-related SLA and configures accounting parameters on AAA client 1 before initiating the accounting process.

First off, AAA server 1 creates a Master ID and stores in the corresponding database a record mapping the Master ID with the actual ID of the user as the case may be. Initially, AAA server 1 also plays the role of the Administrative server by beginning to track user activities. A new Event ID 1 is created with reference to the Master ID. The Event ID 1 is assigned to the accounting data corresponding to the user connection to the network. According to the user SLA he is charged with a monthly tariff for network access regardless the technology used or the time he stays connected. Considering the first step of user activities the accounting data collect results to no additional charge for the user. Thus, the first record corresponding to Event ID 1 contains the null value. The second step involves Internet access which, according to the user SLA, is charged separately and the actual cost is metered in terms of downloaded kilobytes. In this case the cost for Internet access through the WLAN for the kilobytes the user downloaded is charged with the amount of 10€. As the new event is discovered by AAA server 1 a new Event ID 2 is created with reference to Master ID and the cost of 10€ along with some comments is stored.

Step 3 requires that the access technology has changed and UMTS connection is activated. This requests that the user attaches to AAA client 5 that supports UMTS. Upon successful attachment to the 3G network, through AAA client 5, a new AAA server 3 takes the role of the Administrative server. The old Administrative server requires that AAA client 1 terminates accounting while AAA server 3, acting as the new Administrative server from now on, configures and initializes accounting on AAA client 5. At that time the AAA server 1 sends to AAA server 5 the SLA of the user as well as his Master ID. AAA server 5 creates a new Event ID 3 in order to store accounting data regarding the user access to the UMTS network which is null for this case. Event ID 4 is created in order to determine the cost regarding the Internet access. According to the user's SLA and the volume of data downloaded the cost is 2€. Also, a new Event ID 5 is necessary in order to keep the accounting data regarding the purchase of a mobile game (i.e., step 4). The game is priced 3€ by the network operator.

In step 5 the user leaves the Home Network and needs services offered by an FO. Thus, the user attaches to AAA client 7 that is being serviced by AAA server 4. Both machines reside inside the foreign domain. During the process of authentication and authorization, AAA server 3 sends the required SLA along with Parent ID 1 to be used as a reference for any new event ID(s) created by AAA server 4. AAA server 4 is now the Administrative server in charge and Event ID 6 stores accounting data regarding the user access to the 3G network which is null for this case. Event ID 7 is created in order to determine the cost regarding the two phone calls performed by the user.

Finally, according to the mechanism described in the previous section, all accounting records will be forwarded towards AAA server 1, that is, the Root server for the given user.

6. Analysis of communication messages and procedures

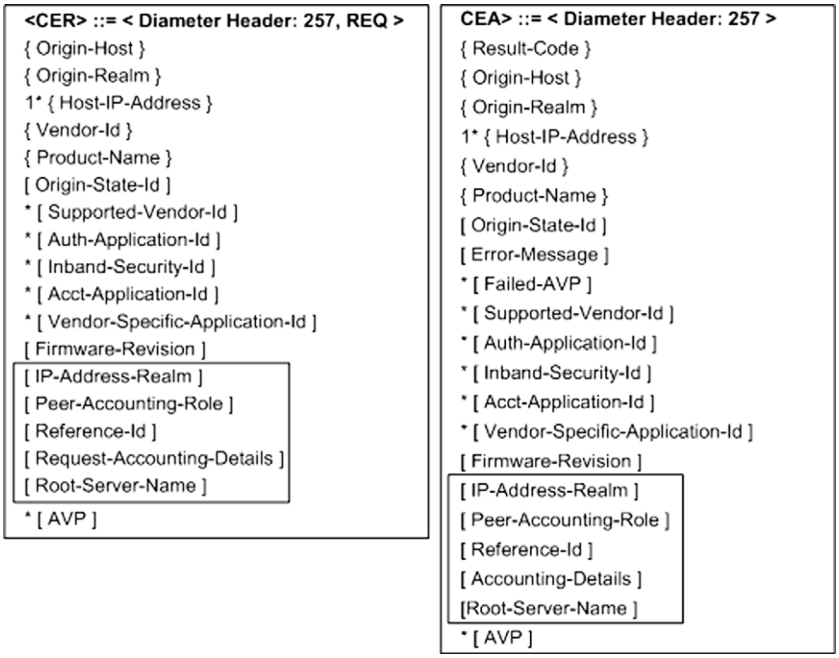
Communication between all network elements that participate in the accounting process as described in the previous section is performed via the Diameter protocol. This section elaborates on the proposed architecture in terms of message exchange between network elements emphasizing on the required Diameter commands (also

Table 1
Proposed extensions and additions to the base Diameter protocol.

New Diameter command/ AVP	Added to command	Purpose
IP-Address-Realm	CER and CEA	AVP to store the IP address to be assigned to a newly-arrived user
Peer-Accounting-Role	CER and CEA	AVP to define the role of the AAA peer regarding accounting
Reference Id	CER and CEA	AVP to transfer the reference ID between AAA servers
Request-Accounting-Details	CER	AVP used to request accounting instructions for a specific user
Accounting-Details	CEA	AVP to convey the appropriate accounting instructions for a given user
Root-Server-Name	CER and CEA	AVP to store the identity of the Root server
Technology-Type	AA-Request	AVP to inform the AAA server regarding the desirable access technology
Event Id	AA-Answer	AVP to convey a new event ID
Setup-Accounting	AA-Answer	AVP to inform the AAA client to initiate or terminate the accounting process. It carries all the accounting setup instructions
Accounting-Records-Request (ARR)	–	Diameter command used to transfer accounting records between AAA servers
Accounting-Records-Answer (ARA)	–	Diameter command used to acknowledge the delivery of the accounting records
Accounting-Records	ARR	AVP to convey the accounting records between AAA servers
Accounting-Response	ARA	AVP to acknowledge the correct delivery of accounting records or request a new transfer

referred to as Diameter messages) and AVPs employed. In the context of our work several Diameter commands and AVPs have been introduced or enhanced to incorporate new abilities to the base protocol. In accordance to the guidelines regarding the creation of new accounting applications [26] and for the sake of compatibility we have tried to keep the number of new custom commands and AVPs to the minimum possible and instead re-use pre-defined ones. All amendments and additions we made to base protocol are summarized in Table 1.

Before the establishment of any new session, Diameter capabilities negotiation must be carried out in order to determine what Diameter applications are supported by each peer. Diameter sessions must be routed only through authorized nodes that have advertised support for the Diameter application required by the session. By doing so, all entities inside a domain are aware of the capabilities of the neighboring peers as part of the network initialization/discovery procedure. It is important that throughout this process AAA peers know which AAA clients are serviced by which AAA servers and which users are serviced by which AAA clients. The latter is achieved as AAA clients keep a pre-defined number of IP addresses to be assigned to newly arrived users. Diameter protocol capabilities negotiation is actually performed every time two Diameter peers need to communicate before establishing the actual connection. The Diameter peers participating in capabilities negotiations can be either AAA clients or AAA servers.



```
<CER> ::= < Diameter Header: 257, REQ >
{ Origin-Host }
{ Origin-Realm }
1* { Host-IP-Address }
{ Vendor-Id }
{ Product-Name }
[ Origin-State-Id ]
* [ Supported-Vendor-Id ]
* [ Auth-Application-Id ]
* [ Inband-Security-Id ]
* [ Acct-Application-Id ]
* [ Vendor-Specific-Application-Id ]
[ Firmware-Revision ]
[ IP-Address-Realm ]
[ Peer-Accounting-Role ]
[ Reference-Id ]
[ Request-Accounting-Details ]
[ Root-Server-Name ]
* [ AVP ]
```

```
CEA> ::= < Diameter Header: 257 >
{ Result-Code }
{ Origin-Host }
{ Origin-Realm }
1* { Host-IP-Address }
{ Vendor-Id }
{ Product-Name }
[ Origin-State-Id ]
[ Error-Message ]
* [ Failed-AVP ]
* [ Supported-Vendor-Id ]
* [ Auth-Application-Id ]
* [ Inband-Security-Id ]
* [ Acct-Application-Id ]
* [ Vendor-Specific-Application-Id ]
[ Firmware-Revision ]
[ IP-Address-Realm ]
[ Peer-Accounting-Role ]
[ Reference-Id ]
[ Accounting-Details ]
[Root-Server-Name ]
* [ AVP ]
```

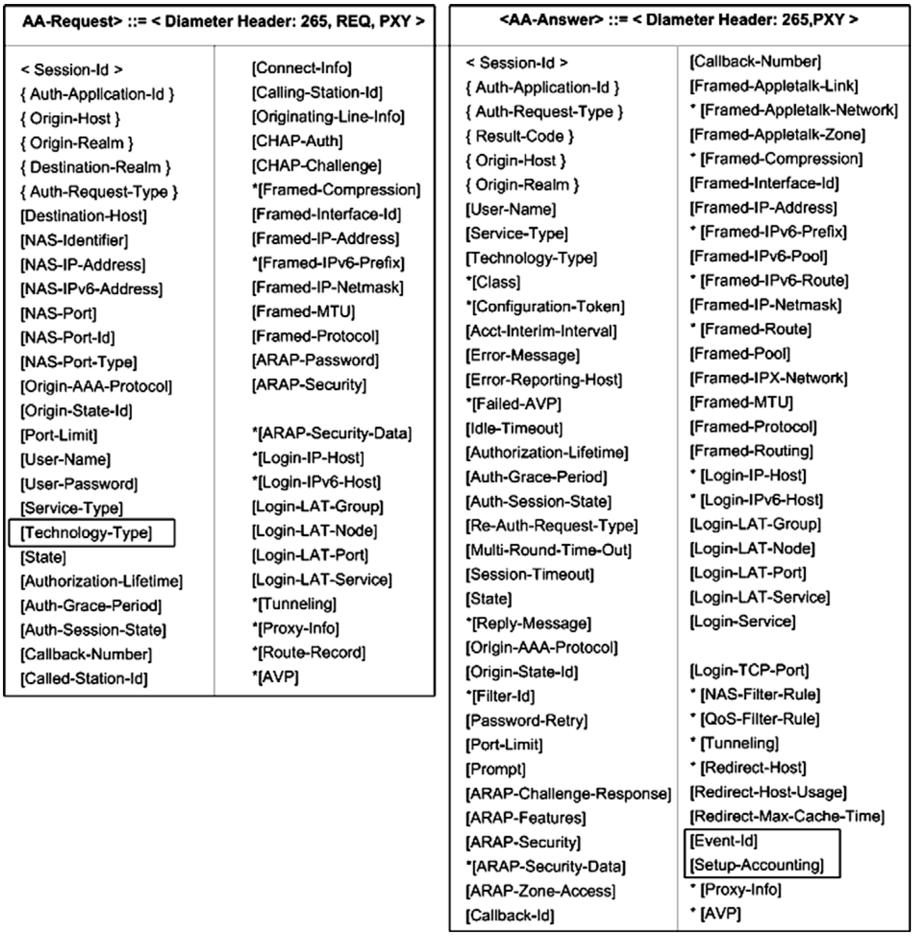
Fig. 4. CER and CEA message format.

The above process is achieved via the Diameter protocol commands *Capability-Exchange-Request* (CER) and *Capability-Exchange-Answer* (CEA). The CER command is used to let the recipient know the exact capabilities of the sender. Respectively, the CEA is sent in response to a CER command. As shown in Fig. 4, a new AVP, namely, *IP-Address-Realm*, is added to these messages in order to store the IP addresses as described previously. Apart from standard AVPs we created a new one called *Peer-Accounting-Role*. This defines the role of the given peer as far as accounting is concerned. Peer-Accounting-Role carries a value when sent by one AAA server to another and the specified role can take three distinct values, i.e., *Root server*, *Home_Administrative server*, or *Foreign_Administrative server*.

It is also required that the Master ID or Parent ID is transferred from an AAA server to the next one through the new AVP *Reference Id* as shown in Fig. 4. The *Request-Accounting-Details* AVP added in a CER message is used in order to request specific accounting instructions for a given user (e.g., information derived from the user SLA). Also, the *Accounting-Details* AVP in the CEA command conveys the appropriate accounting instructions.

Finally, the new AVP *Root-Server-Name* is used to store the identity of the Root server. This is desirable so that every Administrative server is able to transfer the required accounting records towards the correct Root server. Alternatively, the Administrative servers are aware of the Root server by extracting the Root-Server-Name from the Reference Id field. However, the implementation of the Root-Server-Name AVP is used to speed up the process. This AVP carries a value only when sent from an AAA server to another.

When the user attaches to a serving AAA client, and before accounting is triggered, the authentication and authorization requirements must be successfully fulfilled. The AAA client contacts the appropriate AAA server and requires authentication and authorization on behalf of the user. This is achieved via the standard Diameter *AA-Request* (AAR) and *AA-Answer* commands. The AAR command is used to request authentication and/or authorization for a given AAA client. In response to this, the AA-Answer command is sent by the AAA server towards the AAA client providing all important information regarding the received query. In case the AAA server that controls the current AAA client is not capable or responsible for granting authentication to the specific user, these messages may be proxied to an appropriate AAA server. This point is of major importance in cases where the user is requesting



```
AA-Request> ::= < Diameter Header: 265, REQ, PXY >

< Session-Id >                [Connect-Info]
{ Auth-Application-Id }       [Calling-Station-Id]
{ Origin-Host }               [Originating-Line-Info]
{ Origin-Realm }              [CHAP-Auth]
{ Destination-Realm }         [CHAP-Challenge]
{ Auth-Request-Type }         *[Framed-Compression]
[Destination-Host]            [Framed-Interface-Id]
[NAS-Identifier]              [Framed-IP-Address]
[NAS-IP-Address]              *[Framed-IPv6-Prefix]
[NAS-IPv6-Address]            [Framed-IP-Netmask]
[NAS-Port]                    [Framed-MTU]
[NAS-Port-Id]                 [Framed-Protocol]
[NAS-Port-Type]               [ARAP-Password]
[Origin-AAA-Protocol]         [ARAP-Security]
[Origin-State-Id]
[Port-Limit]                  *[ARAP-Security-Data]
[User-Name]                   *[Login-IP-Host]
[User-Password]               *[Login-IPv6-Host]
[Service-Type]                [Login-LAT-Group]
[Technology-Type]             [Login-LAT-Node]
[State]                       [Login-LAT-Port]
[Authorization-Lifetime]      [Login-LAT-Service]
[Auth-Grace-Period]           *[Tunneling]
[Auth-Session-State]          *[Proxy-Info]
[Callback-Number]             *[Route-Record]
[Called-Station-Id]           *[AVP]
```

```
<AA-Answer> ::= < Diameter Header: 265,PXY >

< Session-Id >                [Callback-Number]
{ Auth-Application-Id }       [Framed-Appletalk-Link]
{ Auth-Request-Type }         * [Framed-Appletalk-Network]
{ Result-Code }               [Framed-Appletalk-Zone]
{ Origin-Host }               * [Framed-Compression]
{ Origin-Realm }              [Framed-Interface-Id]
[User-Name]                   [Framed-IP-Address]
[Service-Type]                * [Framed-IPv6-Prefix]
[Technology-Type]             [Framed-IPv6-Pool]
*[Class]                      * [Framed-IPv6-Route]
*[Configuration-Token]        [Framed-IP-Netmask]
[Acct-Interim-Interval]       * [Framed-Route]
[Error-Message]               [Framed-Pool]
[Error-Reporting-Host]        [Framed-IPX-Network]
*[Failed-AVP]                 [Framed-MTU]
[Idle-Timeout]                [Framed-Protocol]
[Authorization-Lifetime]      [Framed-Routing]
[Auth-Grace-Period]           * [Login-IP-Host]
[Auth-Session-State]          * [Login-IPv6-Host]
[Re-Auth-Request-Type]        [Login-LAT-Group]
[Multi-Round-Time-Out]        [Login-LAT-Node]
[Session-Timeout]             [Login-LAT-Port]
[State]                       [Login-LAT-Service]
*[Reply-Message]              [Login-Service]
[Origin-AAA-Protocol]
[Origin-State-Id]             [Login-TCP-Port]
*[Filter-Id]                  * [NAS-Filter-Rule]
[Password-Retry]              * [QoS-Filter-Rule]
[Port-Limit]                  * [Tunneling]
[Prompt]                      * [Redirect-Host]
[ARAP-Challenge-Response]     [Redirect-Host-Usage]
[ARAP-Features]               [Redirect-Max-Cache-Time]
[ARAP-Security]               [Event-Id]
*[ARAP-Security-Data]         [Setup-Accounting]
[ARAP-Zone-Access]            * [Proxy-Info]
[Callback-Id]                 * [AVP]
```

Fig. 5. AA-Request and AA-Answer message format.

services from a foreign domain or an FSP, meaning that an AAA server inside the home domain must be contacted during the authentication process.

Upon successful authentication the user is granted an IP address and is thereafter allowed to access network services and data, based on the authorization details contained in the AA-Answer message. At this point the accounting procedure must be triggered and thus all actions performed by the user need to be constantly monitored. Therefore, we introduced the following additions to the AA-Request and AA-Answer commands. First off, the AA-Request command has to inform the AAA server regarding the preferable access technology. Preferable access technology refers to the mechanism the user wishes to utilize to access the selected services. Possible values may contain GPRS, Bluetooth, UMTS, 802.11, 802.16 and other access technologies. The new AVP is called *Technology-Type*. In case this AVP contains the null value it implies that the user wishes to access the service using the current technology; this is the most common scenario in real-life network operation. It is worth noting that standard Diameter does provide a similar AVP, namely, *Service-Type*. However, the aforementioned AVP carries only general categories of services, "voice", "data", etc. and thus cannot serve our purposes.

Likewise, via the AA-Answer command the AAA server (i.e., the Administrative server in charge of accounting) sends an event ID, to which the AAA client will assign accounting metrics. This is achieved through the *Event Id* AVP we have incorporated into the AA-Answer command as shown in Fig. 5. In this way whenever the AAA client forwards accounting records to the AAA server the appropriate event ID will be utilized.

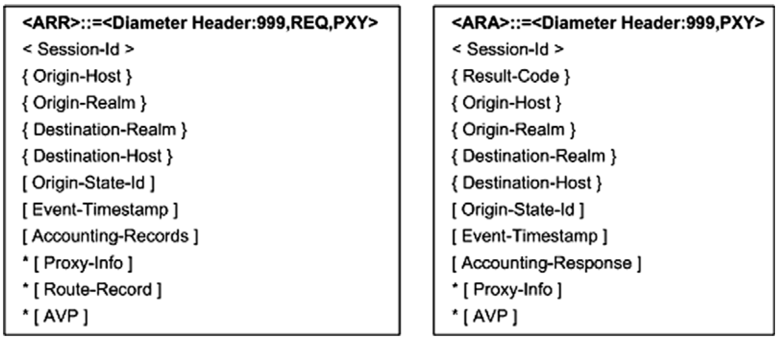
Additionally, the new *Setup-Accounting* AVP contained in the AA-Answer message informs the AAA client to either initiate or terminate accounting. It also conveys all the required parameters regarding accounting setup. This information is derived from the user's SLA and other user-related data.

It is stressed that we demand an AA-Request and AA-Answer command to be used each time a new event ID is created. This is mandatory as new event IDs are generated in case: (a) the access technology has changed, (b) the administrative domain has changed, (c) the user attaches to a new AAA client, and (d) the user requests a new service. In the first three cases authentication is mandatory and the user allocates a new IP address while in (d) authorization is executed prior to service delivery. For the latter case, the standard Diameter *RE-AUTH-Request* (RAR) and *RE-AUTH-Answer* (RAA) commands can also be utilized. These two commands are very useful for an AAA server that has initially authorized a session in case of prepaid services to confirm that the user is still receiving the service [30]. For the sake of brevity the format of the two aforementioned commands is not depicted here since they behave in a similar way as the AA-Request and AA-Answer ones.

When asked by the AAA server or when the user moves outside the coverage of the current AAA client, the AAA client sends all the gathered accounting metrics to the AAA server in charge. This is achieved via the *Accounting-Request* (ACR) command. In response to this, an *Accounting-Answer* (ACA) message is sent as an acknowledgement from the AAA server back to the AAA client.

Contrariwise to standard Diameter where accounting metrics are sent only from AAA clients to AAA servers, in our case, Administrative servers need a mechanism to reliably transfer all accounting records, produced after the manipulation of accounting data, to the Root server or to another Administrative server (i.e., in case of handoff to a foreign domain). To cope with this issue we introduce two new DIAMETER commands named *Accounting-Records-Request* (ARR) and *Accounting-Records-Answer* (ARA). The message structure of these two new commands is depicted in Fig. 6. The *Accounting-Records* AVP carries the actual accounting records while the *Accounting-Response* AVP included in the ARA message acknowledges the delivery of accounting records or requires a new transfer.

Upon termination of an active session it is required that the AAA client informs the AAA server about the incident. The session termination could be either due to network malfunction or a normal procedure when the user chooses to disconnect. The *Session-Termination-Request* (STR) message is used by the AAA client to inform the AAA server regarding the termination of the session for the current user. In response to that message the AAA server will send the *Session-Termination-Answer* (STA) message, acknowledging the session termination. Similarly, the AAA server is able to force the AAA client to stop providing a service to a given user by sending an *Abort-Session-Request* (ASR)



```
<ARR>::=<Diameter Header:999,REQ,PXY>
< Session-Id >
{ Origin-Host }
{ Origin-Realm }
{ Destination-Realm }
{ Destination-Host }
[ Origin-State-Id ]
[ Event-Timestamp ]
[ Accounting-Records ]
* [ Proxy-Info ]
* [ Route-Record ]
* [ AVP ]
```

```
<ARA>::=<Diameter Header:999,PXY>
< Session-Id >
{ Result-Code }
{ Origin-Host }
{ Origin-Realm }
{ Destination-Realm }
{ Destination-Host }
[ Origin-State-Id ]
[ Event-Timestamp ]
[ Accounting-Response ]
* [ Proxy-Info ]
* [ AVP ]
```

Fig. 6. ARR and ARA message format.

message. The AAA client needs to respond to this message via the *Abort-Session-Answer* (ASA) command. All the aforementioned messages (STR, STA, ASR, ASA) are provided by the base Diameter protocol.

7. Performance evaluation

The performance of the proposed architecture is evaluated in a properly designed test-bed and the results are presented in this section. As already mentioned in Section 2 no other results either theoretical or experimental are available for a direct comparison. So, in the following we only compare our results with standard Diameter where applicable.

7.1. Test-bed setup

In mobile ubiquitous environments one of the key factors affecting user experience and market acceptance of a particular technology is the response time in terms of service time and utilization of resources. In this work we have enhanced standard Diameter-based AAA principles by adding new services and procedures, thus expecting network or/and service times to increase. So, the key question here is whether the perceived cost in terms of service time is affordable compared to common user (and provider) practices and requirements. This section aims not to measure time requirements for AAA procedures but actually to determine and evaluate the performance penalty imposed by our modifications compared to standard AAA Diameter.

Since AAA architectures comprise several time-consuming discrete procedures, service time measurements produce useful information only when they refer to the same scenario parameters. So, in order to provide a proper test-bed for our architecture, we build a basic and generic scenario without utilizing any sophisticated equipment. As no comparable metrics from relevant accounting architectures exist until now (see Section 2), we designed our test-bed in order not to simply measure the time penalty imposed by our custom-made accounting procedures but actually evaluate how these accounting-related additions or modifications affect the overall AAA operations. Specifically, the most important metric is the overall time required from the point the user attempts to attach to an AAA client until he finally acquires the requested service. This time window may include all authentication and authorization operations as well as accounting-related procedures. Also, it considers possible handoffs and network malfunctions, mainly in the form of packet losses in case of heavy traffic and network utilization.

It should be noted that all AAA operations other than those related to accounting are not affected by our architecture and thus are expected to perform according to the standard Diameter protocol. Nevertheless, as reported in Section 6, our architecture adds some new accounting-related commands and inserts data (i.e., AVPs) into several AAA messages used during the authentication and/or authorization process aiming at minimizing the overall bandwidth consumption. This is the case with AA-Request and AA-Answer messages described in Section 6. As a result, while the actual authentication and authorization procedures (i.e., cryptographic operations) should not withstand any time penalty, the creation, exchange and processing of the corresponding messages is expected to impose time and other resource consumption-related penalties. Of course, testing such a complex scenario may lead to ambiguous observations as a result of the large number of hard to evaluate events due to casual Internet and connection difficulties and bottlenecks. Instead of that, we argue that our accounting extensions as given in Section 6 affect only authentication and accounting-related procedures. This is especially true in case of frequent handoffs. Hence, we have constructed two different scenarios focusing specifically on these two services.

Apart from the service time measurement tests we also utilize other metrics to determine the CPU/memory utilization penalty imposed by our architecture. These tests are utilized to determine the soundness and robustness of the proposed scheme regarding resource consumption. This is equally important as carelessly designed accounting procedures may result to high resource requirements which of course is not suitable for mobile environments as discussed here.

In this context, our experimental test-bed comprises the following elements, also described in Table 2. Note that each element is required to take a corresponding role according to the deployed scenario:

- One PDA MIO P560 equipped with a 400Mhz Samsung 2443 CPU, 64 MB of RAM and 802.11g capabilities. The PDA runs Windows Mobile v 6.00 Classic.

Table 2
Test-bed components.

Machine/role	CPU	RAM (MB)	Operating system
Low-end user device/PDA	Samsung 2443 at 400MHz	64	Windows Mobile v. 6.00
High-end user device/PC	Intel Mobile Core 2 Duo T7500	2048	OpenSuse 11.0 (32 bit)Kernel v. 2.6.25.16-0.1
AAA client	AMD Mobile Athlon 4 downgraded to 350 MHz	256	-//-
Low-end AAA server 1	Intel Pentium 3 at 733 MHz	512	-//-
Low-end AAA server 2	Intel Pentium 3 at 800 MHz	348	-//-
Low-end AAA server 3	Intel Pentium 3 at 800 MHz	512	-//-
High-end AAA server 1	AMD Athlon 64 X2 3800+	2048	-//-
High-end AAA server 2	Intel Core 2 Duo 8200	2048	-//-
High-end AAA server 3	Intel Mobile Core 2 Duo T7500	2048	-//-

- One high-end laptop incorporating an Intel Mobile Core 2 Duo T7500 processor along with 2048 MB of 333 MHz RAM. This machine affords both Ethernet and 802.11 connectivity and is only used as a high-end user device.
- One low-end laptop machine incorporating an AMD Mobile Athlon 4 CPU along with 256 MB of 133 MHz RAM. To keep processing power to the minimum possible level we have downgraded the CPU to 350 MHz from the original 1200 MHz using Powersave daemon in version 0.14.0 [33]. This machine is connected to a Linksys WGK200G router via the Ethernet adapter and is only used as an AAA client.
- One Intel Pentium 3 733 MHz desktop PC that incorporates 512 MB of 133 MHz RAM. This machine is used as a low-end AAA server.
- One Intel Pentium 3 800 MHz desktop PC incorporates 348 MB of 133 MHz RAM. This machine is used as a low-end AAA server in scenarios which require an additional low-end one.
- One laptop machine equipped with an Intel Pentium 3 800 MHz processor and 512 MB of 133 MHz RAM that is used as a low-end AAA server. This machine is utilized in scenarios that require three low-end AAA servers.
- One AMD Athlon 64 X2 3800+ desktop PC with 2048 MB of 333 MHz RAM. This machine takes the role of a high-end AAA server.
- One Intel Core 2 Duo 8200 desktop PC having 2048 MB of 666 MHz RAM to acts as a high-end AAA server. This machine is utilized in scenarios that require an additional high-end AAA server.
- One high-end laptop incorporating an Intel Mobile Core 2 Duo T7500 processor along with 2048 MB of 333 MHz RAM. This machine is employed in scenarios requiring three high-end AAA servers.

All systems, except the PDA, utilize Linux OpenSuse 11.0 (32 bit) with Kernel v. 2.6.25.5-1.1. Our architecture does not require the employment of a remote database server for storage purposes. This functionality is instead provided by local databases implemented directly into the AAA server machines. For this purpose we use MySQL in version 5.0.45. The well-known MD5 and DES mechanisms are used for the cryptographic operations involved in scenarios that require authentication. The communication between all AAA nodes is based on OpenDiameter v.1.0.7-I [27]. In all scenarios we use the term "*standard Diameter*" to refer to the original version of the OpenDiameter. Conversely, the term "*Diameter with proposed extensions*" is used to denote the usage of our custom-made Diameter as it is described in Section 6. Also, the library *schedutils* which is part of the Linux operating system is used to force Diameter processes to run on a single CPU in multiprocessor systems. This is an important issue when measuring CPU utilization as we utilize both single and multi processor systems. In all scenarios we use both low-end and high-end client/server configurations.

In order to properly configure and being capable of measuring the performance of an AAA client a low-end laptop machine with high-speed direct Ethernet connection with the router is utilized. Usually, in real-life deployments, AAA clients (e.g., an AP) do not afford sufficient CPU resources so here we only utilize a low-end machine. In any case, an AAA client mainly proxies messages between a user device and an AAA server, hence measurements are not affected by this machine. Also, our accounting message extensions do not involve AAA clients by any means.

7.2. Scenario I: evaluation of the authentication phase

The first scenario determines the time penalty imposed by our architecture due to user authentication. That is, the time required for a successful user registration with the network. Recall that the main concern here is not to measure the overall authentication time but actually the penalties imposed by adding accounting-related information to the messages used during the authentication process. In this sense, we determine if the additional time cost due to the creation, exchange and processing of (our) more complex messages is acceptable in terms of service time. As described in Section 6 these messages are the: (a) CER and CEA commands used during the capability exchange procedure prior to session establishment, (b) AAR and AA-Answer commands utilized during the actual authentication and authorization process, and (c) RAR and RAA commands used within an already established session.

Instead of studying a simple authentication scenario that involves a user requesting authentication from his home domain we choose to examine a scenario where a user is attempting to register for the first time to a foreign network. Fig. 7 depicts this situation. The current scenario resembles real roaming incidents and requires

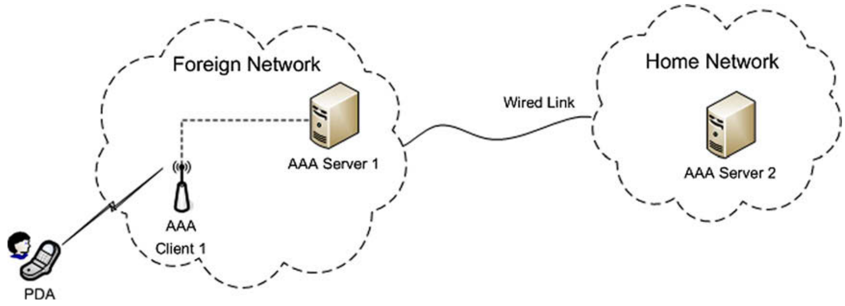


Fig. 7. Network architecture for Scenario I.

the involvement of the user's home domain AAA server apart from other AAA servers placed inside the foreign domain. This results in an increased number of authentication-related messages as well as an additional delay due to Internet roundtrip time. In fact, this is the worst case scenario regarding an authentication request and is thus ideal for our purposes. Achieving a mean authentication time close to that witnessed when standard Diameter is employed should ensure acceptable overall performance experienced by the end users.

Specifically, we measured the overall time required for the completion of the process that starts when the AAA Client 1 creates a CER message to be sent towards AAA Server 1, and ends upon the delivery of an AA-Answer message to the AAA Client 1. This time includes the following events: (a) exchange of CER/CEA messages between AAA Client 1 and AAA Server 1, (b) dispatch of an AAR message from AAA Client 1 to AAA Server 1, (c) exchange of CER/CEA messages between AAA Servers 1 and 2, (d) exchange of AAR and AA-Answer messages between AAA Servers 1 and 2, and (e) dispatch of an AA-Answer message from AAA Server 1 to AAA Client 1.

As depicted in Fig. 7 this scenario parameters include the use of the PDA as the user device, the AAA client along with Low-end AAA servers 1 and 2 for the low-end configuration and High-end AAA servers 1 and 2 for the high-end configuration. This leads to four discrete measurement groups, which correspond to the following variations:

(A) Standard Diameter with low-end configuration,
(B) Standard Diameter with high-end configuration,
(C) Diameter with proposed extensions with low-end configuration, and
(D) Diameter with proposed extensions with high-end configuration.

Measurements were taken from 500 runs for each of the four aforementioned configurations. Table 3 summarizes the results for each particular case. Apart from the mean delay metric we include in the corresponding table the standard deviation, and the 95% confidence interval. A graphical comparison between different scenarios mean authentication delay time is also provided in Fig. 8.

This scenario is also suitable to measure the CPU workload due to the creation and process of our new messages. Since authentication is the most CPU-intensive procedure, as it requires cryptographic operations, we can get a good estimation of the complexity added to the authentication-related messages due to accounting. To do so, we tracked and logged measurements of the AAA server 1 CPU workload for 500 runs. Table 4 illustrates the average percentage of workload as well as the standard deviation and confidence interval metrics.

Table 3
Service time results for Scenario I.

Configuration	Mean time (ms)	Standard deviation (ms)	Confidence interval (95%)
A	53.529	8.638	(52.772, 54.287)
B	43.005	2.806	(42.759, 43.251)
C	56.030	6.978	(55.419, 56.642)
D	44.936	3.192	(44.656, 45.216)

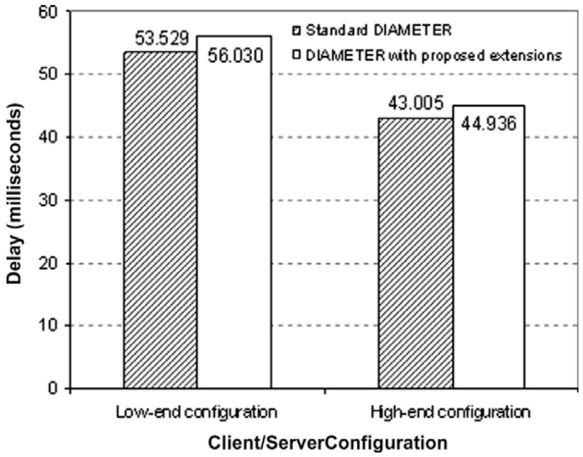


Fig. 8. Comparison of mean authentication delay: standard Diameter vs. Diameter with proposed extensions.

Table 4
CPU utilization for AAA server 1.

Configuration	CPU workload (%)		
	Mean	Standard deviation	Confidence interval (95%)
A	19.50	3.07	(19.19, 19.81)
B	13.12	2.57	(12.89, 13.35)
C	20.30	4.09	(19.80, 20.80)
D	13.23	2.31	(13.02, 13.44)

The results reveal that our architecture produces a negligible time penalty on the authentication procedure. Specifically, both configurations produce a time penalty between 2.501 (i.e., 56.030–53.529) and 1.931 (i.e., 44.936–43.005) ms. Also, standard deviation of all values remains low, showing that their majority is spread near the mean delay. This observation is further supported by the calculated confidence intervals. On top of that, this observation is perceived in a worst case scenario as already described. This is very important as authentication incidents are expected to happen frequently during handoffs, network malfunctions and events that require re-authentication on behalf of the user. Thus, it is desirable to keep the required time delay to the minimum possible. Other less demanding authentication scenarios that we have also tested include:

(a) first time authentication with the home network,
(b) re-authentication to another AAA server inside the same domain, and
(c) authentication after successive handoffs.

All the above cases also register an even smaller penalty. For instance, in case of Diameter with proposed extensions and high-end configuration the mean delay witnessed for the above three scenarios was 5.61, 3.91 and 19.07 ms correspondingly. Concluding, we can say that the overhead imposed by accounting in the overall authentication process in terms of service time is negligible. At the same time accounting is triggered with no additional bandwidth consumption as the authentication messages also carry accounting-related instructions and other required information (see Section 6). Regarding the CPU workload we can maintain arguably that even relatively weaker machines in the role of AAA servers would be able to cope with the demands that our extensions generate.

7.3. Scenario II: evaluation of core accounting procedures

The second scenario is destined to evaluate our new accounting procedures. Recall that until now there is no relevant work to compare our findings with. Moreover, a comparison with the default Diameter would not yield any reliable observations for this second scenario since our accounting extensions are not supported by the standard Diameter. Moreover, as mentioned in Section 4 and further explained in Section 5.1, standard Diameter is not able to cope with all possible accounting scenarios and their requirements. Therefore, the current scenario aims to determine: (a) the consistency and soundness of the proposed accounting system, and (b) how the engaged network elements respond to the new requirements in terms of resource consumption imposed by our accounting extensions.

The network architecture of this scenario is depicted in Fig. 9. Specifically, the user registers to his home network by connecting to an AAA client but sometime later a handoff occurs and he needs to attach to a foreign network through AAA client 1. AAA servers 1 and 2, which are placed inside the home network, take over the role of the Root server and Administrative server 1, respectively. Likewise, AAA server 3 inside the foreign network acts as the new Administrative server 2 for the time the user remains inside the foreign network and no further handoff occurs. AAA servers 1 and 2 reside in the same 100 Mbps LAN and connect with AAA server 3 through the Internet. Connection is realized through a 1 MB ADSL line, i.e., 1024 Kbps downlink and 256 Kbps uplink maximum speed. The average ping time between the two subnetworks is 23.4 ms but this value can only be considered as an indication.

Among all new accounting procedures the most interesting and sensitive one is the transfer of accounting records from all engaged Administrative servers back to the Root server. The current scenario, which follows the assumptions presented below, focuses on that specific incident. The Root server sends a message towards the Administrative servers involved to force them into sending all accounting data regarding the user. This is realized by the standard Diameter *Accounting-Poll-Ind* command [34]. In fact, the Root server is only aware of the Administrative server 1 (AAA server 2), but Administrative server 1 is expecting accounting records from the Administrative server 2 (AAA server 3) before sending all accounting records to the Root server. Such a situation is presented in Fig. 10.

During the execution of this second scenario we measured the mean time required for the completion of the process that starts when the Root server requests the user's accounting records and ends when all the corresponding accounting records are collected and stored in the database. This scenario was also repeated 500 times. Both the low-end and the high-end client/server configurations are utilized. We also logged the mean CPU workload trying to identify any overheads imposed by the proposed accounting extensions. In a typical scenario AAA servers are expected to simultaneously handle a large number of requests that demand a lot of processing. Hence, keeping CPU workload to the minimum possible is very important. As it is illustrated further down in Tables 5 and 6 we only present the results derived from the AAA servers as they are solely responsible for the new accounting extensions. AAA client workload does not yield any interesting or unexpected results and thus is not depicted here. Recall that our accounting message extensions do not involve AAA clients by any means. The following assumptions are regarded important:

- The user has already successfully authenticated himself to the home network.
- Later on, the handoff is completed with a successful authentication through AAA client 1 and no further reauthentication is required.
- Root server and Administrative server 1 have already gathered their accounting records for the specific user.
- Administrative server 2 is currently responsible for the user and has not yet gathered any accounting record as the user is still receiving the service.
- AAA client is responsible for collecting accounting metrics to be sent to the Administrative server 2.
- All accounting-related information including user SLAs, Reference IDs have already been transferred to the engaged AAA nodes.

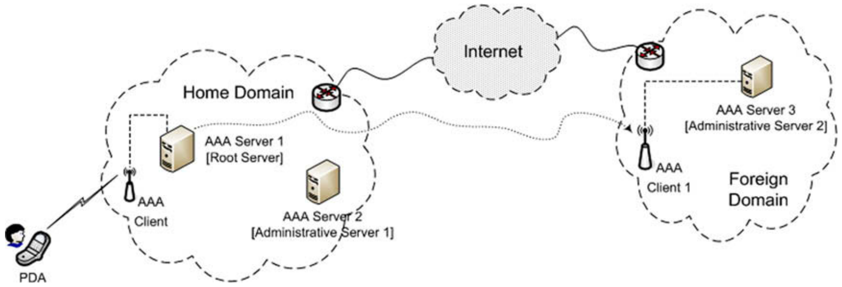


Fig. 9. Network architecture for Scenario II.

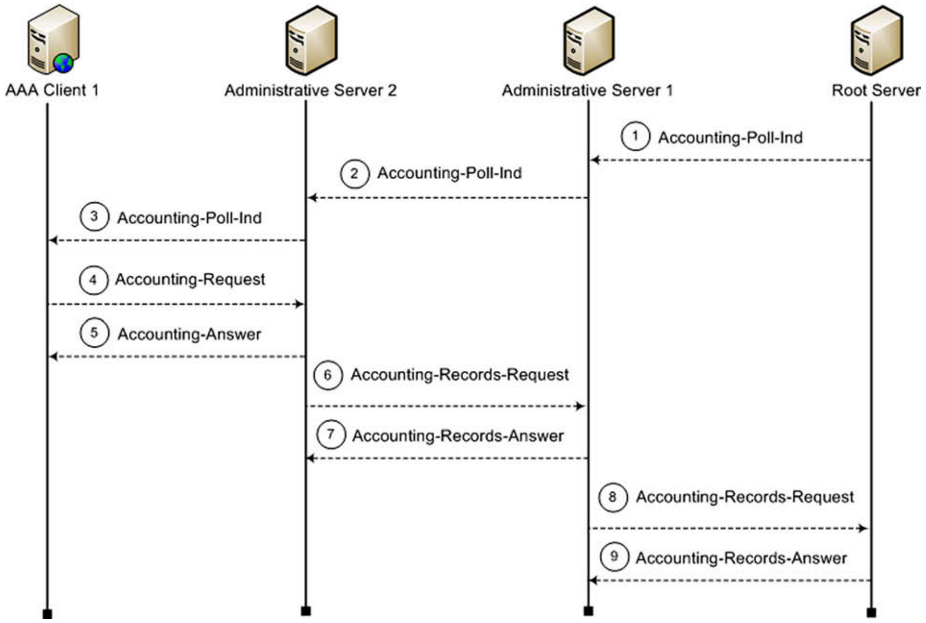


Fig. 10. Accounting scenario message workflow.

Table 5
CPU workload metrics for Scenario II.

Entity	Low-end configuration PU workload (%)			High-end configuration PU workload (%)		
	Mean	Standard deviation	Confidence interval (95%)	Mean	Standard deviation	Confidence interval (95%)
Root server	35.01	4.02	(34.66, 35.36)	14.89	2.54	(14.67, 15.11)
Administrative server 1	29.63	5.96	(29.11, 30.15)	9.72	0.84	(9.67, 9.79)
Administrative server 2	30.47	2.56	(30.25, 30.69)	10.13	2.01	(9.96, 10.30)

Table 6
Service time results for Scenario II.

Configuration	Mean (s)	Standard deviation (s)	Confidence interval (95%)
Low-end	10.312	1.510	(10.180, 10.444)
High-end	7.409	1.386	(7.287, 7.531)

This is a well-designed scenario to test the performance and most importantly the soundness of the proposed accounting extensions as it involves:

(a) creation, exchange and process of the new Diameter commands and extensions,
(b) gathering accounting metrics from an AAA client. Accounting data transformation into the corresponding records from the AAA servers is included as well,
(c) transfer of accounting records between two Administrative servers and between an Administrative server and a Root server and
(d) storage of the final accounting records having the proper form by the Root server.

Table 5 depicts the CPU workload induced to the AAA servers in terms of mean, standard deviation, and confidence interval metrics for both low and high-end configurations. From the results we can infer that high-end configuration keeps CPU utilization at a considerably low level, as is expected. When low-end machines take over the role of the AAA servers, CPU workload is increased but not in an unacceptable level. In realistic scenarios all machines are expected to be a lot more powerful than the ones we employed here for the low-end configuration. So, we believe that the proposed architecture is viable in real-life implementations.

To further investigate this remark we altered the parameters of the current scenario by adding one more client. This client corresponds to a new user. The client sends authentication request messages to the AAA servers repeatedly (i.e., following a negative exponential distribution), hence increasing the number of messages the AAA servers need to process. The AAA servers were configured to respond to all requests though denying user authentication as we did not wish to incorporate authentication procedures delays in this scenario. Also, the servers under stress were not allowed to forward requests to other servers. A small delay was witnessed in case of the low-end configuration, i.e., mean workload for Root server and Administrative servers 1 and 2 was increased by 3.2%, 2.7%, and 2.5% correspondingly. This penalty was even smaller when our high-end machines were utilized, i.e., 1.9%, 1.3%, and 1.4% correspondingly for the Root server and Administrative servers 1 and 2. This means that even under stress, the AAA servers respond well to the demands posed by our scheme.

Table 6 provides the mean overall run time for both scenarios. It is to be noted that the results depend highly on the Internet roundtrip time as proven by several similarities perceived in the time values gathered for both the low-end and high-end configurations. This was expected and actually desirable as this is the case in real-life scenarios with the weakest factor being the link quality between the different administrative domains.

Our findings show that the implementation of the proposed accounting extensions is sound and they do not contain any inherent design flaws. The mean times of 10.3 and 7.4 s for the low-end and high-end configuration, respectively, are highly appreciated for the current scenario. As in the first scenario, standard deviation of all values remains small, showing that their majority is spread near the mean delay. This observation is further supported by the calculated confidence intervals. We also incorporated the same amendment to the scenario parameters, by adding one more user client to stress the servers, receiving no noticeable variations to the mean CPU utilization times. Once more, no authentication-related procedures were initiated by the AAA servers and they were only allowed to process the incoming message. A pre-defined response was created in response and sent towards the corresponding AAA client.

8. Conclusions

In this paper we elaborate on the issue of accounting as part of the AAA concept. Our goal is to provide a practical and easy-to-implement accounting solution for next generation mobile heterogeneous environments. We briefly discuss the main principles of accounting in such realms and provide background information regarding current AAA protocols. We argue that an accounting system should be generic, distributed, flexible and above all secure. In this direction we analyze the desirable criteria and characteristics an accounting system should meet. Also, having in mind the demand for frequent handoffs occurring in mobile multi-administrative environments, we identify the most important accounting scenarios and elaborate on them by providing a real-usage scenario.

We particularly focus on Diameter because it is expected to dominate the AAA market in the near future. Hence, our implementation is based on this protocol and is realized by means of new Diameter AVPs and commands. Our design tries to minimize intervention with the core Diameter protocol so as to maximize compatibility. Indeed, the proposed scheme requires no modifications to hardware or software of any involved network entity (i.e., AAA server, router, client) and attains full compatibility with the base Diameter protocol. More importantly, our scheme can be easily conveyed into any present or future AAA protocol and support modern brokering environments [26].

A test-bed is designed to allow us to determine the performance penalty imposed by our Diameter extensions in comparison to the core protocol when applicable. We focus on time delays and servers CPU workload and investigate both low-end and high-end system configurations. Through extensive experimentation we can infer that the proposed accounting architecture is sound, robust, and above all, easy to realize. Additionally, the conducted results exhibit that our scheme is promising for real-life deployments. All tests show that the engaged parties respond well to the new accounting processes, and the imposed penalties in terms of service time and resource utilization are considered rather insignificant, if not negligible in some cases. Future work will focus on the scalability of our mechanism by incorporating real-time accounting procedures and exploring real-performance data.

Acknowledgements

This paper is part of the 03ED675 research project, implemented within the framework of the “Reinforcement Programme of Human Research Manpower” (PENED) and co-financed by National and Community Funds (20% from the Greek Ministry of Development – General Secretariat of Research and Technology and 80% from EU – European Social Fund).

References

[1] IETF Status Pages, Authentication, Authorization and Accounting services, <http://tools.ietf.org/wg/aaa/>.

[2] H. Kim, H. Afifi, Improving mobile authentication with new AAA protocols, IEEE International Conference on Communications (ICC), vol. 1, IEEE Press, 2003, pp. 497–501.

[3] C.E. Perkins, Mobile IP joins forces with AAA, IEEE Personal Communications 7 (4) (2000) 59–61.

[4] Fang Meng, Changqing An, Jiahai Yang, Implementing a secure AAA system in IPv6, in: International Conference on Network Communication Technology (ICCT), IEEE Press, 2006, pp. 1–4.

[5] M. Cappiello, A. Floris, L. Veltri, Mobility amongst heterogeneous networks with AAA support, IEEE International Conference on Communications (ICC), vol. 4(28), IEEE Press, 2002, pp. 2064–2069.

[6] D. Geneiatakis, G. Kambourakis, C. Lambrinoudakis, A mechanism for ensuring the validity and accuracy of the billing services in IP telephony, in: Proceedings of the 5th International Conference on Trust, Privacy and Security in Digital Business (TrustBus), LNCS, vol. 5185, Springer, 2008, pp. 59–68.

[7] Byung-Gil Lee, Hyun-gon Kim, Sung-Won Sohn, Kil-Houm Park, Concatenated wireless roaming security association and authentication protocol using ID-based cryptography, 57th IEEE Semiannual Vehicular Technology Conference (VTC-Spring), vol. 3, IEEE Press, 2003, pp. 1507–1511.

[8] V. Jesus, S. Sargento, M. Almeida, D. Corujo, R. Aguiar, J. Gozdecki, G. Carneiro, A. Banchs, P. Yanez-Mingot, Integration of mobility and QoS in 4G scenarios, in: 3rd ACM Workshop on QoS and Security for Wireless and Mobile Networks, Crete Island, Greece, ACM Press, 2007, pp. 47–54.

[9] H. Chaouchi, A new policy-aware terminal for QoS, AAA and mobility management, International Journal of Network Management 14 (2) (2004) 77–87.

[10] Internet Engineering Task Force, <http://www.ietf.org>.

[11] T. Janevski, M. Janevska, A. Tudzarov, P. Stojanovski, D. Temkov, G. Stojanov, D. Kantardziev, M. Pavlovski, T. Bogdanov, Interworking of cellular networks and hotspot wireless LANs via integrated accounting system, in: First International Conference on Wireless Internet (WiCON), IEEE CS Press, 2005, pp. 72–78.

[12] A. Munir, V. Wong, Interworking architectures for IP multimedia subsystems, Mobile Networks and Applications, vol. 12(5), Kluwer Academic Publishers, 2007(pp. 296-308, Dec.).

[13] H. Moustafa, G. Bourdon, Y. Gourhant, Authentication, authorization and accounting (AAA) in hybrid ad hoc hotspot's environments, in: International Conference on Mobile Computing and Networking (MobiCom), ACM Press, 2006, pp. 37–46.

[14] F. Eyermann, P. Racz, C. Schaefer, B. Stiller, T. Walter, Generic accounting configuration management for heterogeneous mobile networks, in: Third ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH), Cologne, Germany, ACM Press, 2005, pp. 46–55.

[15] F. Eyermann, P. Racz, B. Stiller, C. Schaefer, T. Walter, Diameter-based accounting management for wireless services, Wireless Communications and Networking Conference (WCNC), vol. 4, IEEE Press, 2006, pp. 2305–2311.

[16] S. Thakolsri, C. Schaefer, T. Walter, W. Kellerer, Accounting management for session mobility in an ubiquitous environment, in: International Conference On Wireless Communications And Mobile Computing (ICWCMC), Vancouver, British Columbia, Canada, ACM Press, 2006.

[17] J. Lorincz, G. Udovicic, D. Begusic, Architecture of SQL databases for WLAN access control and accounting, in: 15th International Conference on Telecommunications and Computer Networks (SoftCOM), IEEE Press, 2007, pp. 1–6.

[18] A. Tsakountakis, G. Kambourakis, S. Gritzalis, A new accounting mechanism for modern and future AAA services, in: Proceedings of the IFIP TC 11 23rd International Information Security Conference, LNCS, Springer, 2008, pp. 693–697.

[19] R. Marin Lopez, G. Martinez Perez, A.F. Gomez Skarmeta, Deployment of AAA infrastructures in IPv6 networks, in: Proceedings of the 2005 Symposium on Applications and the Internet Workshops, IEE CS Press, 2005, pp. 26–29.

[20] R.M. Lopez, G.M. Perez, A.F. Gomez Skarmeta, Implementing RADIUS and Diameter AAA systems in IPv6-based scenarios, 19th International Conference on Advanced Information Networking and Applications (AINA), vol. 2, IEEE Press, 2005, pp. 851–855.

[21] Common Open Policy Service protocol, <http://tools.ietf.org/html/rfc3084>.

[22] Terminal Access Controller Access Control System, <http://tools.ietf.org/html/rfc1492>.

[23] IETF RADIUS protocol, <http://tools.ietf.org/html/rfc2865>.

[24] RADIUS protocol open-source implementation, <www.freeradius.org>.

[25] Peng Zhao, Xuewu Cao, Ping Luo, Notice of violation of IEEE publication principles attack on RADIUS authentication protocol, International Conference on Communication Technology (ICCT), vol. 1, IEEE Press, 2003, pp. 208–212.

[26] IETF Diameter protocol, <http://www.rfc-editor.org/rfc/rfc3588.txt>.

[27] DIAMETER protocol open source implementation, <http://www.opendiameter.org/>.

[28] Extensible Authentication Protocol (EAP), <http://www.ietf.org/html.charters/eap-charter.html>.

[29] Network Access Identifier, <http://www.ietf.org/rfc/rfc4282.txt>.

[30] Diameter Network Access Server Application, <http://www.ietf.org/rfc/rfc4005.txt>.

[31] G. Karopoulos, G. Kambourakis, S. Gritzalis, Privacy protection in context transfer protocol, in: Proceedings of the PDP 2008 16th Euromicro International Conference on Parallel, Distributed and Network based Processing, Toulouse, France, IEE CS Press, 2008.

[32] J. Loughney, M. Nahkjiri, C. Perkins, R. Koodli, Context Transfer Protocol, RFC 4067, 2005.

[33] Powersave Daemon, <http://powersave.sourceforge.net/powersave/index.html>.

[34] Diameter Accounting Extensions, <http://tools.ietf.org/wg/aaa/draft-ietf-aaa-diameter-accounting/draft-ietf-aaa-diameter-accounting-01-from-00.diff.txt>.

Alexandros Tsakountakis was born in Heraklion, Greece in 1982. He holds a Diploma in Information and Communication Systems Engineering from the University of the Aegean and an M.Sc. in Information and Communication Systems Security from the University of the Aegean as well. He is currently a Ph.D. candidate, supervised by associate professor Stefanos Gritzalis, at the Department of Information and Communication Systems Engineering, University of the Aegean, Greece.

Georgios Kambourakis (www.icsd.aegean.gr/gkamb) received the Diploma in Applied Informatics from the Athens University of Economics and Business (AUEB), and the Ph.D. in information and communication systems engineering from the department of Information and Communications Systems Engineering of the University of Aegean (UoA). He also holds a M.Ed. from the Hellenic Open University. Currently Dr. Kambourakis is a Lecturer at the Department of Information and Communication Systems Engineering of the University of the Aegean, Greece. His research interests are in the fields of Mobile and Wireless networks security, VoIP security, security protocols, Public Key Infrastructure and mLearning and he has more than 55 publications in the above areas. He has been involved in several national and EU funded R&D projects in the areas of Information and Communication Systems Security. He is a reviewer of several IEEE and other international journals and has served as a technical program committee member in numerous conferences. Dr. Kambourakis is a member of the Greek Computer Society.

Stefanos Gritzalis (www.icsd.aegean.gr/sgritz) holds a B.Sc. in Physics, an M.Sc. in Electronic Automation, and a Ph.D. in Informatics all from the University of Athens, Greece. Currently he is the Head of the Department of Information and Communication Systems Engineering, University of the Aegean, Greece and the Director of the Laboratory of Information and Communication Systems Security (Info-Sec-Lab). He has been involved in several national and EU funded R&D projects in the areas of Information and Communication Systems Security. His published scientific work includes several books on Information and Communication Technologies topics, and more than 180 journal and national and international conference papers. The focus of these publications is on Information and Communications Security and Privacy. He has leaded more than 25 international conferences and workshops as General Chair or Program Committee Chair, and has served on more than 150 Program Committees of international conferences and workshops. He acts as Editor-in-Chief for 1 journal, an Editorial Advisory Board member for more than 10 journals and a Reviewer for more than 35 journals. He was an elected Member of the Board (Secretary General, Treasurer) of the Greek Computer Society. He is a Member of the ACM, and the IEEE.