The Design of an Extended AAAC Architecture

Hasan[1], Davinder Singh[2], Sebastian Zander[2], Moritz Kulbach[3], Jürgen Jähnert[4], Burkhard Stiller[1]

[1] Computer Engineering and Networks Laboratory TIK, ETH Zürich, Switzerland
[2] Fraunhofer Gesellschaft, Institute FOKUS, Berlin, Germany
[3] T-Systems Nova, Berlin, Germany
[4] University of Stuttgart, Super Computing Center, Stuttgart, Germany
E-mail: [hasan|stiller]@tik.ee.ethz.ch, [singh|zander]@fokus.gmd.de,
Moritz.Kulbach@t-systems.de, jaehnert@rus.uni-stuttgart.de

ABSTRACT

The design of a generic and flexible Authentication, Authorization, Accounting, and Charging Architecture (AAAC Arch) as well as the basic considerations of the AAAC System, which implements the architecture for the mobile Internet has been performed. This paper is based on the work carried out within the Moby Dick project, whose main objective is the facilitation of the deployment of an ubiquitous Mobile IPv6 QoS-aware infrastructure through a best-suited and pragmatic use of an evolutionary AAAC Architecture. While a number of orthogonal use cases of AAAC motivate the necessity of various distinct scenarios, covering Quality-of-Service (QoS) and mobility aspects, these use cases form the basis for the understanding and investigations on AAAC tasks, functions, components, and interactions to enable a problem description and requirements specification on a very fine-grained level. The AAAC Architecture design considerations take up those problems and define a clear solution space, covering in detail logical and physical components and their interactions, identifiers, a session model, profiles, and a security model. This paper closes with the projection of an implementation architecture for the proposed AAAC Arch, which will enable the instantiation of these concepts and their protocols in a given Internet.

I. BACKGROUND

Authentication, Authorization, Accounting, and Charging (AAAC) encompass a set of essential functions and tasks, which are located on the critical path for any commercial application being supported in the future and mobile Internet [9]. Adding a second dimension of the future, IPv6 will become the dominating and finally only network layer protocol in the Internet. With the increasing popularity of multimedia applications the demand for Quality-of-Service (QoS) also escalates. Finally, as a fourth issue, mobile users and mobility of applications, devices as well as services will show a larger distribution across our world. Therefore, the main objective of this Moby Dick work-package WP4 encompasses the facilitation of the deployment of an ubiquitous Mobile IPv6 QoS-aware infrastructure through a best-suited and pragmatic usage of an evolutionary AAAC Architecture based on the IRTF AAA proposal.

In providing services to mobile nodes, messages are exchanged between mobile nodes and network entities as well as among network entities, which may even be located across administrative domains. The exchanged messages will be associated with the different phases that a mobile node and the involving network entities pass through from the time the mobile node trying to acquire connectivity until she disconnects from the network. Following phases are identified: (1) acquiring connectivity and authentication, (2) session setup and authorization, (3) session running, monitoring and metering, (4) session termination, and (5) disconnection. While phase (2), (3), and (4) are related to service usage, phase (1) and (5) mark the beginning and the end of the mobile node's presence, respectively. Possible sequences that connect these five phases within the mobile nodes as well as the involved network entities are not simple as mobile nodes may move and, therefore, require handover which theoretically can happen during all these phases.

The following subsections describes two use cases, which investigate on applying the AAAC Architecture to Mobile IPv6 and a QoS supporting network. Mobile IPv6 and QoS can be treated as two separate services, whose usage needs to be authorized and charged. While Mobile IPv6 enables ubiquitous reachability, QoS offers different classes of communication quality. Prior to service usage, users need to be authenticated and service requests need to be authorized. During the session, resource and service consumption will be monitored and accounted for. This can be done via the use of accounting policies [3]. In case of pull authorization the policies must be passed in the AAA response from the server to the service equipment while for agent authorization policies are passed with the service equipment configuration request. It depends on the charging model to where the accounting and session information is sent and where the charging is done. The accounting information could be sent back to the home AAAC System (AAAC.H).

1.1 Use Case 1: AAAC for Mobile IPv6

Mobile IPv6 defines the framework necessary for allowing mobile nodes to communicate to each other and to other nodes while changing their attachment point to the Internet. Nodes that implement this protocol can preserve their existing (TCP) connections to other

computers on the Internet even though they are moving from one physical network to another; this solution works in heterogeneous environments and within a handover scenario from one type of network to another.
From the user perspective, Mobile IPv6 offers plug-and-play features and the possibility of staying connected to the Internet, while moving from one place to another. From the network connectivity provider's point of view, this is different. No one will allow nodes to connect to the Internet using their infrastructure without means of authenticating. Charging for offered will be performed. That is why besides Mobile IPv6 entities the network has to be enriched with another kind of capability allowing for authentication and charging mobile nodes. This kind of capabilities will be implemented with the help of the AAAC infrastructure. A detailed description of Mobile IP and its requirements is given in [4].
In Mobile IPv6 the access router can play the role of an attendant, which requests user authentication and access authorization from the AAAC System. The attendant should collect information on resource usage and provides the AAAC System with accounting and session records to allow for a usage-based charging.

1.2 Use Case 2: AAAC for QoS Infrastructure

In Moby Dick the Differentiated Services (Diffserv) architecture [1] is used to provide QoS. The AAAC Architecture deals with the service provisioning to control service access and to be able to later account and charge for the provided QoS. The AAAC System needs to interface the Diffserv architecture via a specific Application Specific Module (ASM). Two alternative models are described in this section:

1. Deterministic end-to-end QoS (Bandwidth Brokers)
2. Probabilistic QoS with network dimensioning

In the first model it is assumed that an inter-domain QoS setup is facilitated by a Bandwidth Broker (BB) architecture as described in [5] and [8]. In the BB model a path is reserved for a user's session according to the QoS requirements given in the request, e.g., bandwidth or maximum delay. Either a path with the desired properties is reserved, if available or the users request is rejected. The first BB which receives the Resource Allocation Request (RAR) from the user contacts the local AAAC System via an ASM. The authorization request encapsulates data from the RAR needed for authentication and authorization. The AAAC System forwards the RAR via the AAA protocol to the AAAC System of the next downstream BB which performs authorization. This is repeated until the final domain is reached. If all authorization succeeded a Resource Allocation Answer (RAA) is passed back over the AAA protocol to the first BB, which forwards the RAA to the user. At every AAAC System the RAA is passed to the BB via the ASM so that the BB can setup network elements in the domain. In addition either the BB or a separate system needs to be configured to collect accounting information for each domain.
For a probabilistic QoS model - instead of an end-to-end reservation - the domain, where the user currently is located, marks requested flows with a certain QoS based on the users' requests. A resource management entity distributes resources on a wireless cell according to each user's contract which is reflected in the user profile of each user. Every provider on the end-to-end path has a certain maximum available quantity of each service class. QoS provided to a user depends on dimensioning of network resources i.e. how large the offered quantity is vs. how large the current demand of all users in each domain is. As long as demand is smaller than the amount offered, users will get their desired QoS. The amount of each service class in a domain is adjusted according to current contracts. Nevertheless, these adjustments will be less frequent than network condition changes resulting in a semi-static setup.
In the simple case the QoS provision could be statically for the user and tied to his network access. The SLA defines what QoS the user gets and this is setup after the network access has been authorized. In that case no further QoS-related AAAC communication takes place for the first two A. However the authorization answer for network access has to contain parameters for setting up the QoS and the AAAC System must setup accounting. The QoS sessions are released immediately after the user is disconnected from the network.

2. SESSION MODEL AND USER PROFILE

Two basic models and specification modules determine the basis for a generic and efficient AAAC Architecture.

1.2 Session Model

A session concept is needed for auditing and accounting in an inter-domain generic AAAC environment. The notion of a session ID is introduced to bind together a set of related activities. These activities can be single messages, transactions, or sessions. For accounting purposes different accounting records must be tied together and linked to a user ID in order to produce a bill. Auditing requires a binding of authentication, authorization, accounting, and the actual service delivered in order to trace back all occurred actions. For both a link must be set between different sub-sessions a service is composed of. A session record consists of common attributes, which are present for all types of sessions (e.g., session ID) and attributes which are service specific (e.g., current IP address for Mobile IP).
A session is linked to one or more "users", who might be an individual, a hardware device, or a piece of software. A "user" may act as a "customer" or with the permission of the customer. The user might be identified with a public key, a user-ID, an IP address, or a phone number. A session also relates to one or more service "providers", which support the session with resources of some sort. Furthermore, a session may incorporate one or more "broker" which supports information on existing providers. A session concept is needed for the binding of three topics:

- Authentication, authorization, and accounting with the service provisioning process (Service Session)
- Accounting records (maybe generated by different hosts) which provide the accounting data for the services a user has used
- Different Service Sessions belonging together

Authentication, authorization and service usage needs to be linked to make a later auditing and accounting possible. This is even more important if these functions are performed by different entities (in different domains). In the case of fraud it must be checked whether the authentication authenticated a wrong person or the rights granted by the authorization process were wrong. The service usage must be linked to authentication and authorization to associate the accounted data in the service session to the person, who requested the service and has been identified during the authentication. Furthermore, in case of misuse it must be possible to backtrack the malicious service user which hopefully could be done having a link to the authentication and authorization.

A session should have a session ID, which allows each provider and user in the session to audit session activity and compare it with what other providers and users believe happened in the session. The session ID should allow a provider to merge accounting data for the different activities and generate a bill. Each user and provider may have a different view on the overall session. In particular, a user or provider may only know a part of the session that took place. In case more than one accounting record is generated per session (service), the session ID is used to link the different records together. [7] shows that in a dialup service at least two accounting records exist, which must be tied together to provide useful information for charging and billing. Other services may produce more than two records, for example, if interim reports are used or have different service equipment is involved.

A service may be composed of different services from different providers. For the user this may still look like a single service. To correlate accounting and auditing information the different sessions, which are part of a service need to be linked together. This can be achieved via session IDs. Several alternatives for linking are possible. One alternative would be the use of a common session ID. Another alternative would be to use individual session IDs for each sub-session, in combination with a binding function that would allow for obtaining information about the related services.

The session ID is assigned by the first entity which issues a AAAC request for a new service to be established. This entity must be a AAAC server or AAAC client. This first request is normally an authentication or authorization request. The session ID is used in all subsequent messages of this session until the session is terminated. The session ID must be globally unique. This can be achieved by concatenation of a globally unique ID with a locally unique ID. The global unique ID is the identification of the AAAC server, which must be globally unique, because otherwise AAA servers can not be addressed properly. It also could be a globally unique ID of an AAAC client. This ID could be the IP address or some other global unique identifier from an AAAC server namespace. The locally unique ID identifies the session within the AAAC server and be composed of two parts: service identification and session identification. The AAAC server needs to have a namespace of unique service ID. Otherwise it would not be possible to non-handle ambiguously a service request. The service itself has some means to identify uniquely a single session. The service ID could be an ID assigned by a standard signaling protocol (e.g. SIP or RTSP). Thus the full globally unique session ID can be composed by concatenation of AAA server ID, service ID, and session identification.

A session associates a service consumer (user or customer) with the service usage. The service consumer part contains information about the consumer. The service usage part lists all relevant information of the service usage, which is needed for auditing and accounting purposes. Requirements on which information have to be captured are derived from the business model (the respective charging and billing scheme). The information, which has to be recorded for auditing, depends on the negotiated SLA (QoS auditing), non-repudiation issues, or rules given by law. A session itself must be identified through a globally unique ID, which can be used as reference by other sessions. The referenced session is called sub session of the referencing session, while the referencing session is called super session. This may build complex hierarchies, which can be depicted as session graphs. A session also has some common attributes, which are always defined regardless of the service used (see Figure 1).

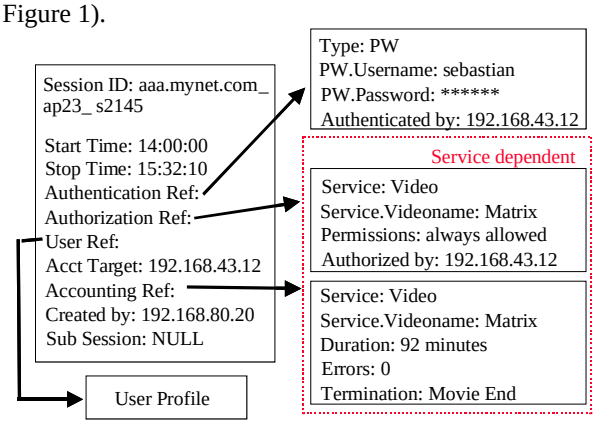

Figure 1: Session Structure

A session starts with the service request and ends with service termination (either by a client or the server). However, the session definition (especially session start, session end) depends on the service and the business model. For example for an application service provision each application use could be seen as single session. On the contrary a single session could be the use of any (or group of) applications of a certain provider.

2.1 User Profile

In order to allow a user the usage of applications and services in the home network and foreign networks, information on the user is necessary for authentication, authorization, and accounting purpose. An AAAC service provides such information to authenticate a user and to establish authorized network services for a user. A main idea of a distributed AAAC service is that all data associated with users are stored centralized in the user database located at the AAAC server of the home

domain. Each user is assigned, from a contractual point of view, to the network provider or any other service provider of its home domain. All data in this database associated with the same user is called user profile of this user.
From a mobile user's point of view there is only one contractual relationship between a service provider and a customer (the mobile user/subscriber). However, a customer will use services offered by more than one service provider. This can be another network service provider providing connectivity (roaming partner), an application service provider, and content service provider. It is expected that the use of services from a third party provider are transparent to the mobile user. A user profile distributed to another service provider enables this other service provider to assign resources to the mobile user according to a description of the profile. This means a visiting service provider gets paid only for a service according to the profile description. This user profile additionally can reflect contractual relationships between different service provider.
A user profile is a data record of user-specific data; it contains all data associated with the user, e.g. user-name, authentication data, service level agreements, charging info, or policies (see Figure 2).

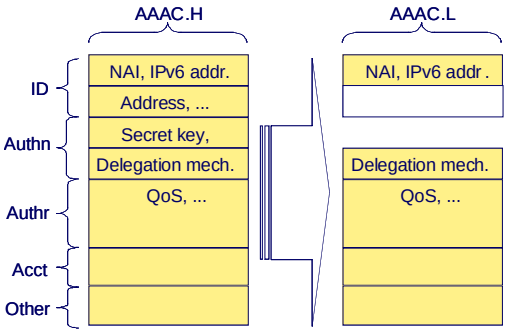

Figure 2: User Profile

The user profile is unique, every user, who is able to access the network and use some application or network services, must have one user profile. The information contained in the user profile is divided into the following different groups:

- User-specific information for authentication: e.g., username, password, secret key
- Service-specific information for authorization: e.g., service level agreements, max bandwidth
- Charging/tariff-specific information needed for accounting: e.g., tariff group
- Mobility-related information
- Paging information

Information in the user profile is represented by attributes. An attribute consists of the name of the attribute, a value, and the Attribute-Value Pair (AVP) [2]. The user profile contains a set of mandatory attributes, which must be present, and a set of optional attributes, which may be present. Mandatory attributes are set during the creation process of the user profile with a user value or the default value. RADIUS and DIAMETER [6], [2] specify numerous attributes.
The user profile contains only static attributes. Static attributes are not being changed during a session. Dynamic data and accounting data are stored in the session and accounting record(s). Maybe some (limited) sections of the user profile could be dynamic. One example of this is a remaining budget in the case of a pre-paid user. In any case, the user has no access to its profile. It is completely controlled by the service provider involved in the service provisioning process.
The interface to a user's profiles is the AAAC.H to which a user has registered. After the AAAC.H receives an authentication and authorization request or a usage of an application or a network service from a AAAC client or server, the AAAC.H retrieves all needed information from the user profile, in order to make a decision. If the decision is successful, the home AAAC server includes in the authentication and authorization response all attributes of the user profile, which are needed to provide the service to the user. Only the result of this decision and the configuration data are send to the AAAC entity, which has issued the authentication and authorization request. The entire user profile is never send to other AAAC entities for privacy and legal issues. Only the AAAC.H has full knowledge of the user profile. The conveyance of the user profile data over the AAA protocol must be secure. Therefore, at least data integrity for the end-to-end communication must be guaranteed and the data must be encrypted

3. AAAC ARCHITECTURE

The work performed in MobyDick enhances the AAA Architecture (Authentication, Authorization, and Accounting) proposed by the Internet Engineering and Research Task Forces (IETF, IRTF) with charging and auditing functionality and targets this generalization at the Internet Protocol version 6 (IPv6). The fact that the AAAC service will be used in a QoS-enabled Mobile IPv6 (MIPv6) environment will be considered in this architecture and protocol design to provide features which will enable new functionality as well as performance optimization of the overall system.
Figure 3 shows the aforementioned enhancement to the generic AAA Architecture. The AAAC System defines the formerly known AAA Server with Moby Dick extensions, covering as an aside charging and auditing functions. Auditing enables further functions with respect to the evaluations of audit trails generated by the AAAC System and other entities. Note that the Policy Repository should be considered as being part of the policy-based AAAC System. AAAC Systems may communicate with each other via an AAA protocol, which will show extensions for additional functionality. This protocol is termed AAAC protocol, but may consist of an advanced AAA protocol, such as the DIAMETER base protocol, and additional extensions.
Via ASMs a variety of service equipment can be addressed, in which case the AAAC System to ASM communication will be operated with AAAC protocol and the ASM to service equipment communication may happen by service equipment-specific protocols. Note that the service equipment provides the services to the user. While in pull sequence the requests for authentication and authorization come from service provisioning entities, in agent and push sequence these

requests come directly from service user. Therefore, in order to support agent and/or push sequence a communication link between the AAAC System and the service user needs to be provided.

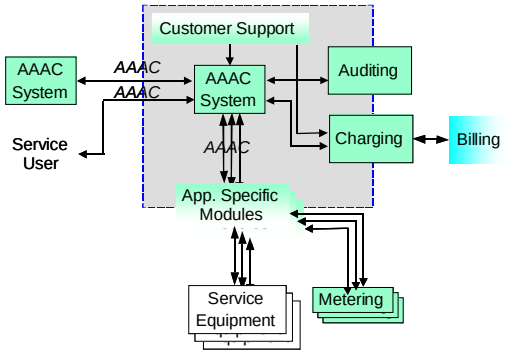

Figure 3: Enhanced Generic AAAC Architecture

Furthermore, a clear distinction is made between particular services offered to the user, such as QoS-enabled or Mobile IPv6-enabled services, and services required for an operational AAAC System, such as charging. Therefore, the former ones are in general accessed and provided via an ASM and the extended AAAC protocol, while the latter ones may communicate directly with the AAAC System, utilizing dedicated communication means if required. To enable an operational system‘s design, the customer support component is required for maintaining customer-specific data, such as customer identifiers and the respective shared secrets, contracts, and tariffs.
As shown in Figure 4, the enhanced generic AAAC Architecture is applied to QoS-enabled Mobile IPv6 environment resulting in the instantiation of the ASM and the respective service equipments.

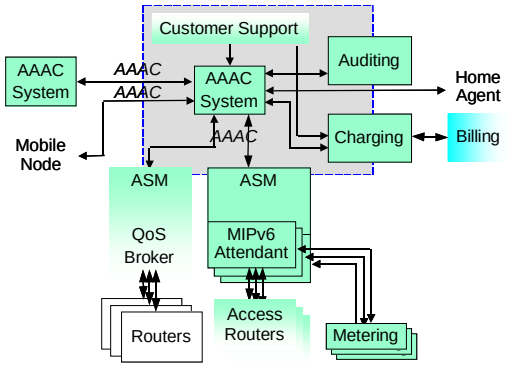

Figure 4: AAAC Architecture for QoS-enabled Mobile IPv6 environment

It is envisioned that Moby Dick will require an ASM for a Mobile IPv6-centered Attendant as well as one for a QoS Broker. In this particular application, the service user is the Mobile Node (MN) on behalf of the human user. It is assumed that any request with respect to the authentification and authorization may originate from alternate AAAC Systems, from the MN (therefore, the user), or from the service being supported by the AAAC System, e.g., the QoS Broker for a QoS-enabled service or from the attendant (which can be the Access Router) for a Mobile IPv6-enabled service.

There are two reasons for the need of a communication link between a AAAC System and a Home Agent (HA). The first reason relates to the optimization of the registration process, where binding update messages are piggybacked on AAAC messages. Secondly, the need to support dynamic establishment of security association between MN and HA with the help of Home AAAC System. This interaction will require a modification in the HA code of traditional Mobile IPv6.

4. CONCLUSIONS

This work addressed a number of highly relevant questions with respect to the extension of the traditional AAA Architecture with respect to charging, mobility, and security support.
This design forms the basis for the implementation of the AAAC System for Moby Dick and its application scenarios. The implementation will require refinements on the interface level between the AAAC System and the QoS entity as well as the mobility entity. However, the AAAC design has outlined, due to its module and device concept of logical and physical entities, that a variety of different implementation solutions will exist, according to and driven by the specific scenario requirement as well as the underlying network technology structure and functionality. The evaluation of the presented work above will follow based on the prototype being built during the next project phases.

5. REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss: *An Architecture for Differentiated Services;* RFC 2475, December 1998.

[2] P. Calhoun et al.: *Diameter Base Protocol;* draft-ietf-aaa-diameter-07.txt, July 2001.

[3] G. Carle, S. Zander, T. Zseby*: Policy-based Accounting;* draft-irtf-aaaarch-pol-acct-03.txt, Internet Draft, Informational, August 2001.

[4] S. Glass, T. Hiller, S. Jacobs, C. Perkins*: Mobile IP Authentication, Authorization, and Accounting Requirements;* RFC 2977, October 2000.

[5] R. Neilson, J. Wheeler, F. Reichmeyer, S. Hares: *A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment;* V 0.7, August 1999

[6] C. Rigney et al.: *Remote Authentication Dial In User Service (RADIUS);* RFC 2865, June 2000.

[7] C. Rigney: *RADIUS Accounting;* RFC 2139, April 1997.

[8] J. Strassner, A. Westerinen, B. Moore: *Information Model for Describing Network Device QoS Mechanisms;* draft-ietf-policy-qos-device-info-model-05.txt>, July 2001.

[9] Hasan, J. Jähnert, S. Zander, B. Stiller: *Authentication, Authorization, Accounting and Charging for the Mobile Internet;* Mobile Summit, Barcelona, Spain, September 2001.