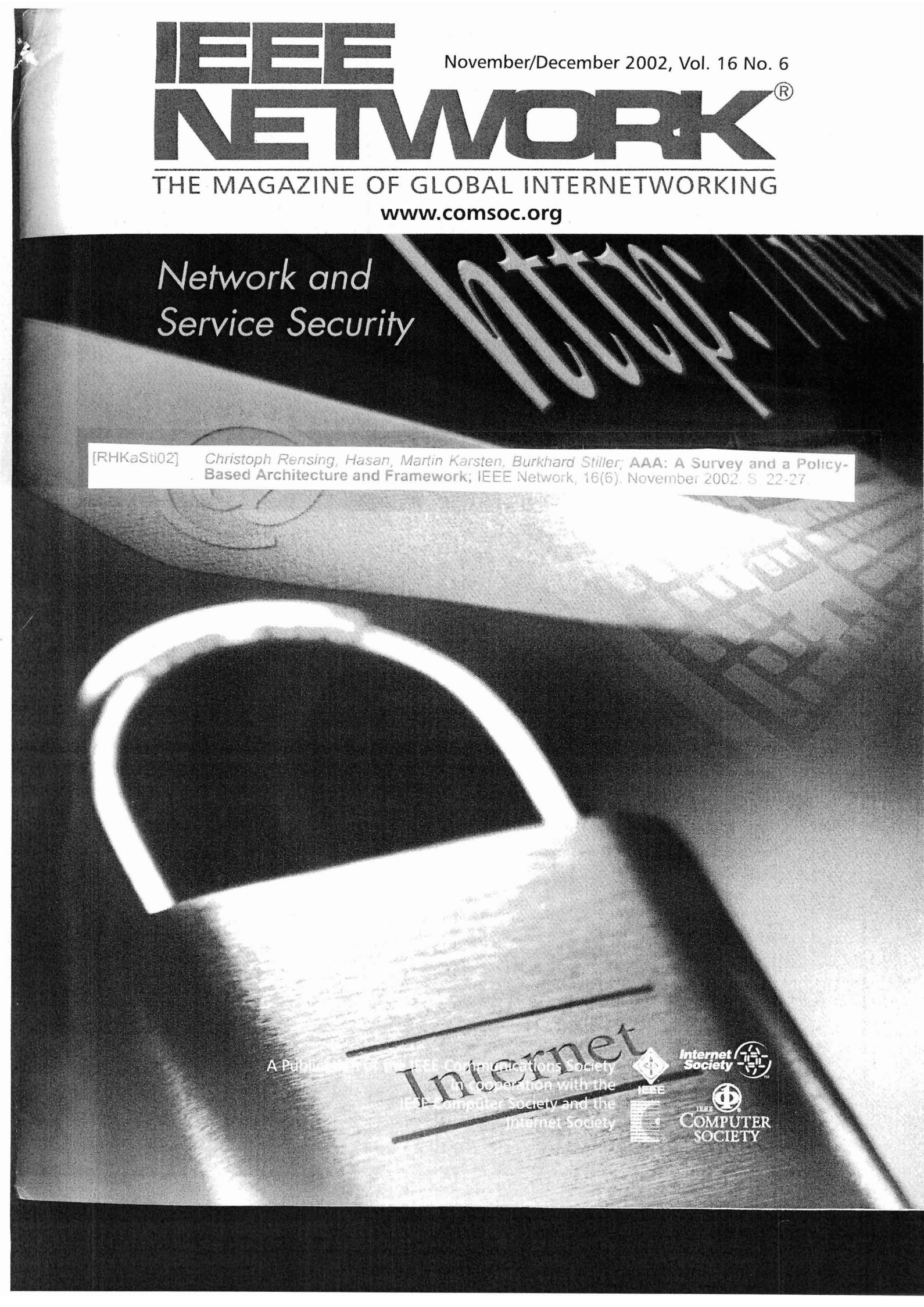
November/December 2002, Vol. 16 No. 6

IEEE NETWORK®

THE MAGAZINE OF GLOBAL INTERNETWORKING

www.comsoc.org

Network and Service Security



A Publication of the IEEE Communications Society
In Cooperation with the
IEEE Computer Society and the
Internet Society

AAA: A Survey and a Policy-Based Architecture and Framework

Christoph Rensing, Darmstadt University of Technology
Hasan, ETH Zürich
Martin Karsten, University of Waterloo
Burkhard Stiller, University of Federal Armed Forces Munich and ETH Zürich

Abstract

The commercialization of the Internet has led to a large variety of business models based on Internet technology. Therefore, the demand for standardized and efficient solutions in support of reliable, secure, open, and flexible remote and mobile service accesses has increased. Existing authentication, authorization, and accounting systems still consider dedicated cases, but lack a generic approach. More general AAA services can be built by extending existing mechanisms and protocols for access scenarios other than dialup or PPP connections. While this work is performed mainly by the IETF AAA Working Group, another approach proposed by the IRTF AAAArch Research Group is termed AAA Architecture. This article surveys the state of the art in AAA and develops a new generic policy-based approach, A^x, for AAA services and beyond, considering flexible levels of various services in an Internet service model, ranging from connectivity to content services.

The set of mobile and fixed communications are on their way to being integrated on today's Internet Protocol (IP)-based networks. Since the Internet offers a public and private communication platform for a variety of services, covering business, academic, and private users, service providers need to differentiate themselves across a wide range of content and personalized services. In addition, they need to ensure returns on their investments in technology for communications, servers, and content. This statement defines the instantiation of the most abstract but crucial business policy of the provider to be followed in the market. Commercialized services do need authentication, authorization, and charging, based on accounting processes, and they drive important technology developments. Furthermore, all security-related issues gain more and more importance with the increasing popularity of user and device mobility.

Besides these economic and market-driven aspects, which motivate the necessity of authentication, authorization, and accounting (AAA) systems, communication technology, as both the environmental and technical basis for AAA systems, requires close investigation to enable the development of future AAA services. The heterogeneity in network components, their functionality, signaling protocols for end-to-end quality assurance, and service provisioning determine major characteristics of current Internet technology. The network of the near future will be the multiservice Internet, consisting of multiple cooperating domains offering access services, transport services, application services, and content. To illustrate this variety of services, protocols, and access regulations, a realistic communication scenario is shown in Fig. 1.

While more detail may be found in [1], this scenario considers different access technologies such as asymmetric digital subscriber line (ADSL), IP intranet, wireless LAN, and mobile access to a wireless WAN, possibly using different technologies like Global System for Mobile Communications (GSM) or Enhanced Data Rate for GSM Evolution (EDGE). Different transport services, ranging from best-effort IP to quality of service (QoS)-enabled transport services, are also provided. Different content and application services are offered as well.

The key problem is defined by a largely extended access control, now consisting of AAA, which reveals more complexity. It is necessary to authorize access to IP networks, transport services with QoS guarantees, and content. Decisions on authorization may be influenced by confidentiality, technical (e.g., remaining bandwidth), and financial (e.g., creditworthiness) aspects. Authentication can be based on different types of identity, such as personal user IDs, IDs of hardware devices, or even anonymous IDs. New access control has to be performed for roaming mobile users by service entities, which have no contract with the users requesting a service.

Most services are charged: for example, the telecommunication connection and the IP access depending on connection time or volume, the transport depending on QoS parameters, and content services depending on the type of content. Therefore, accounting is a must, and it includes more than metering the time a user is connected to the IP network. Additions to the traditional AAA approach are necessary, which concern further components for auditing, pricing, charging, and billing tasks. These extended functions and services are termed A^x services and are introduced in a later section.

Available authentication, authorization, and accounting solutions, commonly referred to as AAA systems [2, 3], exist in the form of protocols and implementations [4-6] that integrate AAA tasks, especially dedicated for dialup or Point-to-Point Protocol (PPP) connections. To meet all requirements for the scenario described above and for future demand, extensions to AAA systems are necessary, and more generic AAA services are required [1, 7].

0890-8044/02/$17.00 © 2002 IEEE

Terminology

The following terms define AAA and related functionality, listed in alphabetical order:

A^x Services: A^x services are services related to AAA as well as auditing, charging, and billing. They contrast so-called user services (services users invoke to meet their needs, e.g., a mail service) in the sense that they are valuable for the provider of user services, to achieve his business goals, and do not relate to the user in a direct way.

Accounting: Accounting is the collection and aggregation of information (accounting records) in relation to a customer's service utilization. It may be expressed in metered resource consumption or negotiated resource values.

Auditing: Auditing is the verification of the correctness of a process with respect to service delivery. Auditing is done by independent (real-time) monitoring or examination of logged system data in order to test the correctness of operational procedures and detect breaches in security. Auditing of accounting data is the basis for after-usage proof of consumed resources and customer charges.

Authentication: Authentication is the verification of the identity of a subject performing an action. The subject of authentication can be a service user or a service provider.

Authorization: Authorization is the verification of whether a subject is allowed to perform an action on (e.g., access to or use of) an object.

Network Policy: Network policies are derived from management goals and define the desired behavior of (and relationship between) different entities in the network by actions to be performed by entities. These entities refer to users, applications, network elements, and service providers.

Service: A service defines a set of capabilities offered by a service provider to a customer. Service equipment, controlled by the provider, generates the service for the user. Services range from connectivity services, which offer access to the Internet, and transport services, which provide pure transport of IP packets, to application and content services.

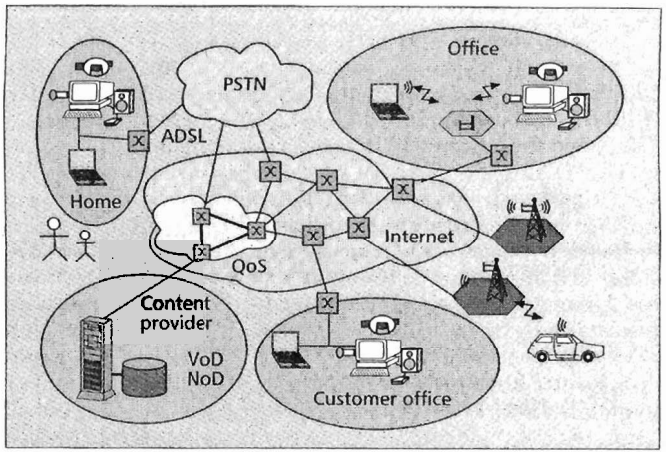
This article is organized as follows. We describe the state of the art in terms of existing AAA work, including mechanisms, protocols, and the Internet Research Task Force (IRTF) AAA Architecture. We discuss weaknesses of existing approaches and major objectives of a new generic approach, the basis of a new design and implementation of AAA services. Finally, we develop the new framework and generic policy-based architecture for AAA services and beyond, A^x.

AAA Mechanisms, Protocols, and Architectures

While AAA mechanisms determine methods to perform authentication, authorization, and accounting, AAA protocols specify appropriate interaction schemes for a distributed system. Finally, AAA architectures address the interworking between various components.

AAA Mechanisms

Authentication denotes the verification of the identity of a subject. The identity can be personal; logical; bound to an organization like Network Access Identifier (NAI) and International Mobile Subscriber Identity (IMSI) stored in the Subscriber Identification Module (SIM) card; bound to an infrastructure like an IP address; or bound to a device, like a medium access control (MAC) address and the International Mobile Equipment Identity (IMEI). Therefore, different classes of authentication mechanisms exist. They can be classified as follows: knowledge-based, cryptography-based, biometrics-based, and secure-tokens-based.



■ Figure 1. *An application scenario.*

Authorization mechanisms are categorized as:

- Authentication-based mechanisms requiring authentication of the subject as a precondition for authorization
- Credential-based mechanisms, which use trustworthy information (credentials) being held by subjects of an authorization

An accounting system covers two major tasks:

- Collect data from metering systems.
- Aggregate and store these data in accounting records.

Accounting records can be generated periodically or triggered by signaling events. An accounting policy specifies which data has to be metered by a metering system, how often it is metered, and how it is aggregated. Call detail records (CDRs), originating from telephony-based telecommunication systems, and IP detail records (IPDRs), used inside packet-based networks, define two examples for accounting records as agreed on data structures. Accounting data may be used for charging and billing, auditing, capacity planning, and security analysis. Beyond those, a multitude of mechanisms for auditing and charging exist [1].

AAA Protocols

From different protocols in support of AAA, those being discussed within the Internet Engineering Task Force (IETF) AAA working group and related ones are outlined. The focus of the AAA working group is on AAA applied to network access. Authentication protocols for network access operate between a service user and an authentication server in general. A network access device such as a dial-in server acts as a relay device to the authentication server. Common protocols used are PPP Password Authentication Protocol (PAP), PPP Challenge Handshake Authentication Protocol (CHAP), or PPP Extensible Authentication Protocol (EAP). The following protocols are of major importance in the context of AAA: RADIUS, Diameter, COPS, SNMP.

The Remote Authentication Dial In User Service (RADIUS) protocol [6] was designed for transferring authentication, authorization, and configuration data between a network access server (NAS), which determines a RADIUS client, and a particular RADIUS server holding the information to authenticate and authorize a user. The RADIUS server itself can act as a client to other RADIUS servers. Originally, RADIUS was defined to support dialup connections, but today it is being used in various situations. RADIUS applies different authentication protocols. Extensions for delivering basic accounting information (e.g., start, stop, and activity data) to a RADIUS accounting server exist. There are major shortcomings in RADIUS because of which it is not considered acceptable as a generic AAA protocol [4, 6]. These shortcomings are protocol-specific, resulting from the original usage scenario, or application-specific, limiting usage of RADIUS in new scenar-

ios: the limited size of attribute data, limited session control as used for accounting, lower fault tolerance due to the use of UDP, and the lack of end-to-end security support.

The Diameter protocol was defined as a successor to RADIUS, removing known RADIUS deficiencies. The protocol satisfies requirements of network access using different access technologies, including wireless packet data technology and distributed security models for multidomain and roaming scenarios. Diameter consists of a base protocol that defines header formats and security extensions as well as a number of mandatory commands and attribute value pairs (AVPs). The base protocol is a session-oriented protocol based on a peer-to-peer model. Besides TCP, Diameter operates over Stream Control Transmission Protocol (SCTP) as a transport protocol, which is not widely used so far. Information is exchanged by means of AVPs. Different extensions to the base protocol allow the use of different access technologies by defining special command codes and AVPs. The NAS server requirements (NASREQ) extensions cover the support of RADIUS authentication protocols, PPP EAP, and authorization needed by NAS services. 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. This accounting extension defines a set of generic accounting AVPs that can be used for all services and supports real-time accounting. Due to its limitation on an IP network access it is not a generic AAA protocol [4]. Other important application areas for AAA services, particularly on the application level, are not considered so far.

The Common Open Policy Service (COPS) protocol [8] enables the exchange of policy information between a policy decision point (PDP) and policy enforcement points (PEPs). It is a query and response protocol in a client/server model. PEPs are clients, and a PDP acts as a server. COPS has been specified to allow authorization of Resource Reservation Protocol (RSVP) resource requests in networks supporting integrated services. However, the protocol was designed to be applicable in a much larger context. COPS is considered acceptable as an AAA protocol for requirements defined in the AAA working group [5].

The Simple Network Management Protocol Version 3 (SNMPv3) proposes a new management model from v2. It enables the design, development, and deployment of sophisticated management applications, including AAA applications. Especially the task of accounting is supported by transferring accounting records to and storing them in an SNMP management information base (MIB). But SNMP cannot be considered a general AAA Protocol [5], since it is restricted to a low-frequency access scheme for MIB information.

Finally, authentication and authorization is also performed for access control of application and content services. Application-independent protocols such as Secure Socket Layer (SSL) and application-specific ones like HTTP-Authentication or Secure Shell (SSH) exist.

The IRTF AAA Architecture

The IRTF research group AAAArch [2] aims to define an architecture and a model for interorganizational AAA. To achieve this, a policy-based approach is applied, where mechanisms as well as protocols such as discussed above are included. This group's work shall conform to the work of the IETF policy framework group [9]. Within the IETF a policy is defined as an aggregation of policy rules made up of conditions and policy actions.

AAA Components — In the IRTF research group the main focus is on authorization policies for service requests and accounting policies belonging to a requested service. Policies are stored in policy repositories (PRs). The rule-based engine (RBE) is a central component of the AAA system, which evaluates policy conditions to make a policy decision and executes respective policy actions. The enforcement of policy actions is done by different components, depending on the kind of action. Most actions belonging to a requested service have to be performed by the service equipment (SE), which may include different network elements. Other actions belong to support services, especially accounting, and are performed by AAA servers separated or even integrated into the SE [3].

AAA Services — The foundation of this AAA Architecture is the assumption of a multidomain Internet topology. In each administrative domain at least one AAA server resides. Distributed AAA servers offer user authentication, authorization, and accounting services. The authorization service is defined as the process of achieving an authorization decision to grant or deny a user's request for services in an authorized session by setting up the SE and logging the session's state. User authentication may be part of the authorization process, and the authentication information will be carried in the authorization request. Accounting services record relevant accounting information obeying the authorization's decision and the ongoing resource use of the authorized session.

To offer AAA services, secured and trusted relationships between different AAA servers are necessary. By contract, the user establishes a trust relationship with a dedicated service provider, his/her user home organization (UHO). This UHO operates an AAA server, as all service providers do, including the foreign organization (FO) from where users request a service. An FO can trust a user if the chain of trust relationships between the relaying proxy AAA servers and the user and the UHO can be resolved. Therefore, authentication between peer AAA servers is part of these services [3].

AAA Architecture and Protocols — All those components are part of the AAA Architecture (Fig. 2). In a pull sequence the AAA server receives service requests from the SE via an application-specific module (ASM), whereas in pull and agent sequence requests come from service users. The RBE resides inside an AAA server to evaluate current requests, which may also originate from other AAA servers acting as agents, according to predefined policies. A request received by the AAA server is inspected by AAA servers considering policies stored in the PR. To evaluate policy conditions, it may be necessary to consult other AAA servers or the status of the SE. This is done first by sending requests to other AAA servers and second via an ASM. ASMs are needed to enforce policy actions. Therefore, ASMs configure SEs to provide a service. Furthermore, policy actions are taken by the AAA server itself. It maintains session states, records accounting data, and logs actions [3].

Protocols and interfaces used in this architecture include the following, labeled according to Fig. 2:

- (1) Special AAA protocol, which is assumed to be standardized in the research group of the IRTF
- (2) Particular application programming interface (API) or the AAA protocol
- (3) Depending on the PR's implementation, the Lightweight Directory Access Protocol (LDAP) or an API
- (4) An application-specific protocol

Problem Areas, Weaknesses, and Goals

The work on AAA has reached a status where a selected number of mechanisms and algorithms are well understood, and proposals for supporting protocols as well as extensions have been made. However, often this work is performed in isolation for shortened tasks and limited scenarios, such as connectivity control through an NAS or content delivery control through a billing system.

The extension of existing AAA systems to support new integrated requirements, based on protocols such as RADIUS or Diameter, particularly considers the implementation of mobility scenarios and roaming approaches. This extension faces problems due to being dependent on underlying technologies like IPv6 and Mobile IP that need to be solved in a technology-dependent approach. These areas of concern are also worked at in the Mobility and Differentiated Services in a Future IP Network (MobyDick) project [7].

The IRTF's AAA Architecture tries to resolve these restrictions by building generic servers and ASMs. However, as discussed earlier, this approach cannot solve all problems related to AAA services and beyond:

- Functions of policy decision and policy enforcement are not separated clearly. An AAA server makes policy decisions on authorization, but also enforces accounting policies by performing accounting tasks.
- Extensibility to functions beyond AAA, like charging and auditing, is complicated, since components are not defined in a generic way. Many enforcement functions are located in the AAA server itself or in the ASM.
- The deployment of AAA services and beyond for applications and content-level services remains difficult, since AAA services are provisioned mainly for transport and connectivity-level services. In particular, accounting, auditing, and charging are not defined for these upper levels.
- The functionality of the ASM has not been defined completely. It acts as an interface for those tasks that cannot be assigned in a generic fashion to the AAA server.
- The inclusion of QoS-related, handover, and paging support services has not been considered.

Therefore, an extended architecture embedded in a new framework is proposed in the next section. The objective of this approach is to define A^x services, not only AAA, in the most generic way and to build an A^x architecture enabling services to be used in support of different user services on different levels in different scenarios using different heterogeneous network components and service protocols.

A Generic Policy-Based A^x Architecture

The precondition for a generic architecture solution independent of existing technology and protocols includes:

- A distinct description of components involved in A^x services
- The identification of interaction schemes between them

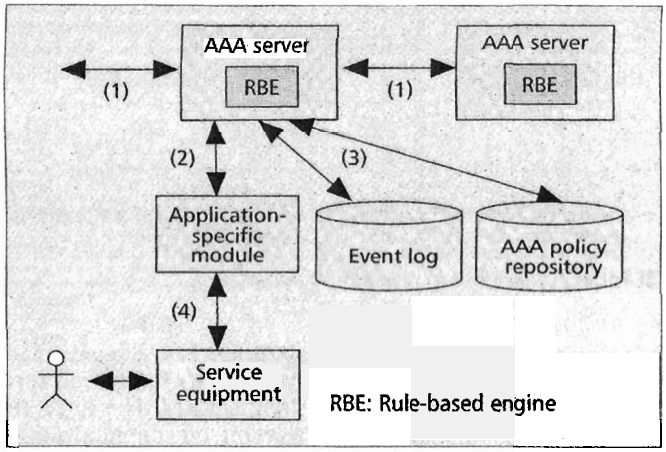
While fixed or wireless network technology or Internet protocols (such as IP, differentiated services, or Mobile IPv6) will be integrated, this work on A^x applies three basic concepts for the framework of an A^x architecture:

- Service separation (extended AAA point of view)
- Partitioning of service levels (new diversification)
- Policy paradigm (reuse of existing work)

Service Separation

Service providers offer services to customers and have to manage distributed systems in the Internet. This management task includes the configuration of networking devices (hardware) and the provision of protocol mechanisms (software). All existing A^x functions are part of these tasks and can be seen as a provider-internal service; they can be separated from those services offered to customers explicitly.

Therefore, it is recommended to separate user services and A^x services from the corresponding SE that provides those services. A^x services are offered to user SE in different phases, as shown in the simplified logical view of Fig. 3. During a service invocation or negotiation phase (preservice delivery phase), a user requests a service from user SE. This request is authorized by the SE delivering A^x services (defined earlier). Alternatively, a service request may be sent to the A^x SE to enable the direct submission of, say, user credentials. The reply configures user SE according to the A^x service's evaluation outcome.



■ Figure 2. *AAA architecture and interfaces [2].*

During the service delivery phase service usage is metered based on the applied policy and existing mechanisms. Finally, accounting and charge calculation as well as billing tasks are performed after or during the service execution. Note that only the A^x SE is responsible for the delivery of those A^x services.

This separation of services allows the definition of a common interface for generic A^x services, independent of network devices or protocols used to request and serve user services.

Partitioning of Service Levels in an Internet Service Model

The layered service model of Internet services is defined as a framework, consisting of a model with four horizontal levels and, where applicable, omitting a restricted view of network access and transport only. The lowest level, 1, is concerned with Internet connectivity, level 2 with transport, level 3 with application issues, and level 4 with content issues. Besides this partitioning into levels, vertical segmentation in the signaling and data path is done as depicted in Table 1. Note that, for example, Mobile IP provides functions that are part of the control path as well as those that are part of the data path due to its integrated definition. For example, ICMP is part of the transport and connectivity level, since it utilizes existing connectivity, but may be used by interdomain routing protocols to establish a particular connectivity.

This horizontal partitioning defines service classes with similar characteristics and similar A^x requirements, too. On the connectivity level authentication based on a hardware device can be done; on the content level personal authorization is often necessary. The vertical partitioning helps to identify the point at which support services are necessary. Authentication and authorization must be done mainly during signaling tasks, whereas accounting has to be performed on data path information (e.g., if a volume-based scheme is applied). The overall partitioning defines protocols, application classes, policies, and mechanisms as abstract objects, which are considered separately on purpose in the enhanced context of AAA with respect to their service characteristics, value, or security requirements.

Policy Paradigm

Since the beginning of the 1990s the policy paradigm has been proposed to be applied in the area of network management. The first major application of policies was access control in distributed systems, often termed *role-based access control*. A

Level	Control path	Data path
Content	RTSP	News, streaming video
Application	http, H.245, SIP	Videoconferencing, IP telephony, Java applets
Transport	RSVP, RTCP, ICMP	TCP, UDP, RTP
Connectivity	DHCP, ICMP	SONET/SDH, DWDM

Table 1. *The generic structure of partitioning.*

broader application of policies in the Internet community was in QoS management, mainly addressing the integrated services and differentiated services architectures. The IETF IP security policy working group is working on communication security policies, mainly for IP Security (IPsec) architecture. The use of policies for network management has different advantages over, for example, manual (command line) configuration or management via SNMP. Special evolutions as described in the first section can be handled by applying the policy paradigm to network management. The separation of a policy from an implementation enables dynamic changes to the management of systems and modification of the behavior of the system. It also allows reusability of policies in different heterogeneous environments, especially inside different administrative domains. Due to these reasons and advantages, the policy paradigm is applied to build an A^x service architecture, which leaves the policy representation [10] out of scope here.

Starting from an abstract business policy of the provider to be followed in the market, policies for pricing, billing and payment, charging and accounting, as well as authorization and authentication can be derived [1]. As the foundation of a generic A^x architecture the common base scheme of a policy-based architecture is applied. Policies are edited via a policy management tool, which performs inconsistency and conflict checking. These are distributed to the PR or directly to a PDP by configuration. PDPs make decisions by evaluating policies along with other data and potentially other policies. If a policy maps, the decision is sent to a PEP, which translates those decisions into configuration data.

A^x Architecture

The generic A^x architecture proposed (Fig. 4) consists of modules and services that provide the extended AAA functionality as discussed.

Modules and Interaction — Necessary modules of the A^x architecture can be deduced from the base scheme. All different A^x policies [1] have to be stored in a PR. This can be a single integrated one or many distributed ones. To evaluate policies, a PDP is used as a module: a single PDP for each different kind of A^x policy or an integrated one. The instantiated design reflects those dependencies between different kind of policies. While the PDP is the major part of the A^x server, investigations of policy dependencies are for further work.

PEPs also define modules of the A^x architecture. They are located in the SE as defined in an earlier section, either the user SE to provide user-requested services or the A^x SE providing A^x services only. Different policies and functions previously presented must be considered to describe those PEPs.

Authorization policies are normally enforced by a decision with specified service parameters, whether or not a service is provided. This decision describes the behavior of the SE according to a user request. Authentication policies are enforced by a special module that owns necessary authentication information on identities to be authenticated, depending on the mechanism, and implements various authentication mechanisms. The PEP of a metering policy is located regularly in the user SE or an extension of it, since it meters service provisioning. Appropriate data can be transmitted to an accounting module ,or stored inside or outside the SE itself. For accounting and charging purposes PEPs are located in special modules that operate on metered data by aggregation and other mechanisms. Results of these operations are stored in accounting and charging databases. The location of PEPs for auditing policies depends on the dedicated mechanism. For real-time auditing the enforcement is performed in all modules providing the service, covering user services and A^x services (real-time auditing PEP). For after-usage auditing a special offline auditing PEP is necessary.

Therefore, the following components are inherently part of the A^x architecture, besides the A^x server, the A^x PR, a policy management tool, and the event log:

- A^x PDPs as a major part of an A^x server
- A^x PR
- Authentication PEP module
- Authorization PEP module inside the user SE
- Metering PEP module inside the user SE
- Accounting and charging PEPs with additional databases for accounting and charging records
- Auditing PEPs dependent on auditing policies located inside each other module or as an independent module

Within this generic architecture (Fig. 4) these modules are drawn as isolated, a single module instantiating a single A^x PEP, while only major relations are depicted. For example, the accounting PEP requires metered data, originating from the metering PEP, or the auditing PEP inspects an event log, accounting or charging records accordingly. After an in-depth examination of detailed dependencies between different A^x policies to be applied, this architecture can be implemented, mainly driven by dedicated performance and security issues. Finally, additional elements are required for implementing enforcement, such as event logs and session directories.

A^x Services — A^x services are provided for user SEs or to other A^x servers in case of a service access from foreign domain. An A^x server consults the PR to make a policy decision whenever an A^x service is requested. A^x services are performed by the A^x server through the enforcement of policies in different PEPs.

For instance, if a user requests a voice-over-IP (VoIP) application service, the VoIP server will send an authentication and authorization request to the A^x server. This request has to specify, among other things, the identity and credentials of the user, and the requested service, which may include a QoS specification. Depending on the authentication and

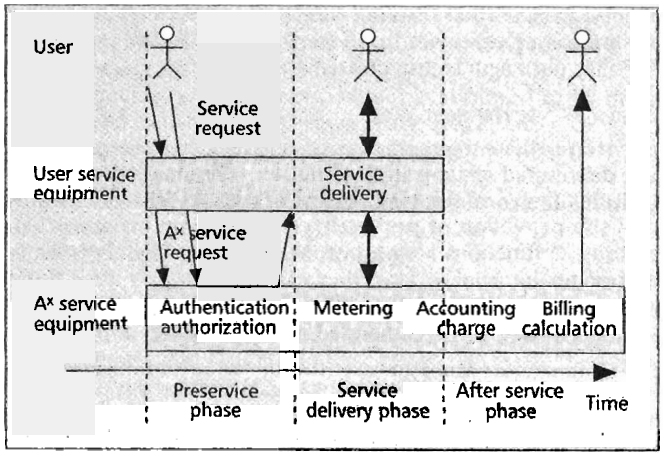
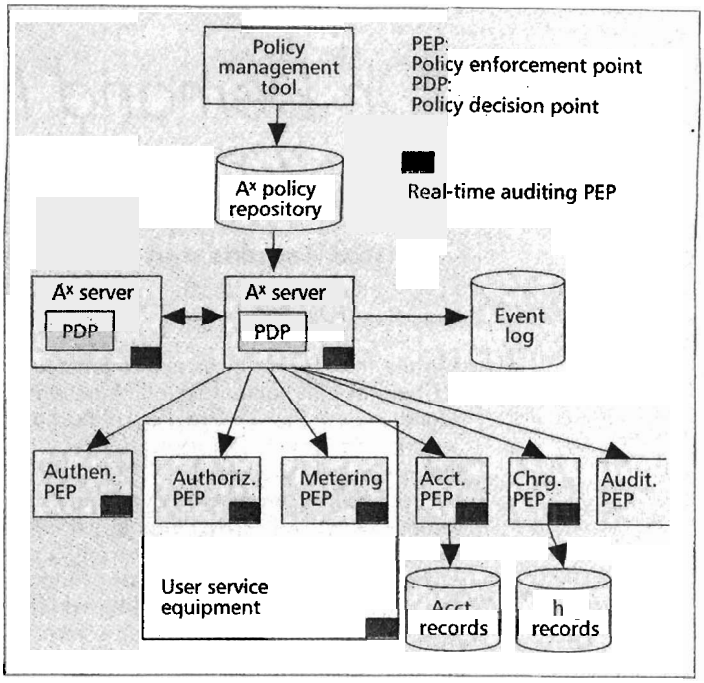


Figure 3. *Service interaction.*

authorization policy, further information may be needed. If the request is authorized, the A^x server configures the respective PEPs and sends a positive response to the VoIP server. All other A^x services will be enforced if the requested service has been authorized. The established charging policy and contractually stated tariffing scheme determine required metering and accounting configuration. For VoIP the effective conversation time may be one important metering parameter. Auditing will ensure that the VoIP service is delivered as specified, and an attack to the VoIP system as well as to the A^x infrastructure can be detected.

As depicted in Fig. 3, some services, like authentication and authorization, are delivered only once during the signaling request. Other services, like metering and real-time auditing, are delivered continuously during the service delivery phase. Finally, other services like accounting and charging can be performed afterward. Note that certain implementations may facilitate a user getting online charge advice during service usage. These dedicated service sequence details have to be ensured in the signaling phase by configuring PEPs accordingly. In addition, different protocols are needed to implement the proposed architecture. To request a service from a user, an A^x protocol is needed. It includes the definition of an interface, like LDAP, between the PR and the A^x server. Different interactions and interfaces between the A^x server and different PEPs are part of this protocol as well.



■ Figure 4. *A generic A^x architecture.*

Summary and Conclusions

There is an increasing need for AAA services and services beyond AAA to enable commercial deployment of services offered by a future all-IP and maybe even a market managed multiservice network. These A^x support services include auditing, pricing, charging, and billing. However, since current AAA architectures, protocols, and implementations do not cope with heterogeneous application scenarios and many requirements on different levels of services, ranging from connectivity to content, are not supported, this lack of a generic approach drives the proposed A^x development.

The generic A^x approach takes these aspects into account and clearly distinguishes between support services and user services. It fully deploys the advantages of the policy-based management architecture by separating decision points from enforcement points on a per-service basis. A^x services can be offered by a specialized A^x system. A^x services, apart from metering, can be offered from one provider to another because of their future separation based on A^x. Therefore, providers can build systems on their own business plans.

Acknowledgments

This work has been performed partially in the framework of the EU IST projects Mobility and Differentiated Services in a Future IP Network (MobyDick, IST-2000-25394) and Market Managed Multiservice Internet (M3I, IST-1999-11429), where ETH Zürich has been funded by the Swiss Bundesministerium für Bildung und Wissenschaft, Bern under grants nos. 00.0275 and 99.0536, respectively.

References

[1] C. Rensing *et al.*, A Survey on AAA Mechanisms, Protocols, and Architectures and a Policy-Based Approach Beyond: Ax; TIK Report no. 111, Comp. Eng. Net. Lab. TIK, ETH Zürich, Switzerland, May 2001.
[2] IRTF AAA Architecture Research Group; http://iridal.phys.uu.nl/~aaaarch/, Mar. 2002.
[3] C. de Laat *et al.*, "Generic AAA Architecture," RFC 2903, Aug. 2000.
[4] P. Calhoun *et al.*, "Diameter Framework Document," Internet draft, draft-ietf-aaa-diameter-framework-01.txt, Mar. 2001, work in progress.
[5] D. Mitton *et al.*, "Authentication, Authorization, and Accounting: Protocol Evaluation," RFC 3127, June 2001.
[6] C. Rigney *et al.*, "Remote Authentication Dial-In User Service (RADIUS)," RFC 2138, Apr. 1997.
[7] MobyDick: Mobility and Differentiated Services in a Future IP Network, http://www.ist-mobydick.org, Mar. 2002.
[8] J. Boyle *et al.*, "The COPS (Common Open Policy Service) Protocol," RFC 2748, Jan. 2000.
[9] IETF Policy Framework (policy) Working Group; http://www.ietf.org/html.charters/policy-charter.html, Mar. 2002.
[10] T. Ryutov and C. Neumann, "Representation and Evaluation of Security Policies for Distributed System Services," *DARPA Info. Survivability Conf. and Expo.*, Hilton Head, SC, 2000, pp 172–83.

Biographies

CHRISTOPH RENSING (Christoph.Rensing@KOM.tu-darmstadt.de) studied joint computer science and business economics at the University of Mannheim, Germany, and received his diploma degree in 1993. In March 1998 he joined the Multimedia Communications Lab — KOM at Darmstadt University of Technology, led by Prof. Dr. Ralf Steinmetz. Currently he is working as a research and teaching assistant and Ph.D. candidate in the research area of security infrastructures and mechanisms for Internet services.

HASAN (hasan@tik.ee.ethz.ch) received his diploma degree in computer science from ETH Zürich, Switzerland, in 1993. He joined the Computer Engineering and Networks Laboratory TIK as a research assistant from 1993 to 1994 before taking a position as a lecturer in the State Polytechnic Malang in Indonesia until 2000. Currently he is a Ph.D. student at ETH Zürich and researching in the area of AAA, mobility, policy-based networking, and auditing of Internet services.

MARTIN KARSTEN (mkarsten@bbcr.uwaterloo.ca, Martin.Karsten@KOM.tu-darmstadt.de) studied joint computer science and business economics at the University of Mannheim, Germany, and received his diploma degree in 1996. In December 1996 he joined the Multimedia Communications Lab at Darmstadt University of Technology, led by Prof. Ralf Steinmetz. He received his Ph.D. in computer science from Darmstadt University of Technology in July 2000 and has been a lecturer and group leader responsible for various research projects since then. He joined the University of Waterloo, Canada, in September 2002, where he is currently an assistant professor in the School of Computer Science.

BURKHARD STILLER (stiller@tik.ee.ethz.ch, stiller@informatik.unibw-muenchen.de) received his diploma degree in computer science and his doctoral degree from the University of Karlsruhe, Germany, in 1990 and 1994, respectively, where he has been a research assistant at the Institute of Telematics. He was on leave at the University of California, Irvine and the University of Cambridge Computer Laboratory, England. Currently he is an assistant professor of communication systems at ETH Zurich and a full professor of computer science at the University of Federal Armed Forces Munich, Germany. His areas of interest include Internet communications, QoS models, charging, and accounting, and he has co-chaired workshops such as IEEE/IFIP DSOM 1999 and LCN 2002, and is currently general chair of the QofIS/ICQT '02 Workshops.