Intelligence on Top of the Networks: SIP based Service Control Layer Signaling

Wolfgang Kellerer
Institute of Communication Networks
LKN, Munich University of Technology
Arcisstr. 21, 80333 München, Germany
Phone: +49 89 289 23505, Fax: +49 89 289 63505
wolfgang.kellerer@ei.tum.de

IEEE Intelligent Network Workshop IN2001, May 6-9 2001, Boston, MA, USA
Formal paper version

***Abstract*-In order to support the integration of heterogeneous networks and diverse access devices we have developed a server architecture that provides integrated telecommunications and data services spanning heterogeneous networks. The server architecture interfaces to the networks via adaptation units to provide a network independent service control on top of the networks. We have structured the architecture according to the principles of the TINA service architecture in a user control part, a service control part and a communication control part. We have adopted the IETF Session Initiation Protocol to serve as a service control layer signaling protocol. Our Session Control Protocol (SesCP) supports the setup and control of complex multimedia communication services and provides flexibility in order to support a wide range of new services. In particular SesCP provides signaling mechanisms that support different levels of the service description, starting with the service 'called-by-name' and finally coming to a precise connection description that is realized in the networks. SesCP also supports the negotiation of service description parameters.**

I. INTRODUCTION

Innovative information and communication services will strongly influence our future life. The service users demand customized services that can be accessed from any terminal within any network. The service providers have to address these rising customer requirements. They will compete to provide new services in shorter and shorter life cycles and to offer their services to a large number of users.

But instead of one homogeneous network, today's communication infrastructure is faced with an emerging heterogeneity of fixed and mobile communication networks and a variety of access devices. Also in future, the network infrastructure is not considered to be one homogeneous network as we had the vision of B-ISDN. Rather, the longer-term industry vision of a next generation network (NGN) is supposed to consist of multiple access networks interconnected by a worldwide packet-based backbone [1].

However a NGN will be like, we need innovative solutions to cope with the increasing heterogeneity of systems. An integration of the heterogeneous multitude of networks would allow application services to be easily accessible independently of the infrastructure. In addition new services can be realized spanning multiple, previously separated networks. The key component for that would be a common platform for the development and provisioning of communication services. We focus on an approach to enable the convergence of heterogeneous networks on a service layer independent to all network specific elements. This means our proposed platform is located on top of the networks and interfaces via adaptation components to the specific networks.Therefore we have developed a service architecture for network independent service control [2] to fulfill the following requirements.

- Network independence: Allow the user to access the server architecture from any terminal and from any access network, and allow the integration of different networks for service control.
- User centric: Management of user access and user profiles separately from service execution.
- Multimedia call control: Support complex multimedia information and communication services.
- Flexibility: Allow a smooth integration of the server architecture in the existing infrastructure; reduce complexity by functional decomposition and reuse of existing interfaces and signaling systems.

Overall, our architecture provides a solution for cross network service control especially supporting personalized services and supporting the implementation of new services spanning multiple networks. In this way, it allows the introduction of a new business role: Using our platform a network independent service provider is able to run services without owning any network infrastructure. In this paper we present our architecture with a focus on the new signaling approach and discuss its general benefits for service signaling in next generation networks.

The remainder of the paper is structured as follows. First we give an overview over the architecture in Section II. In Section III, we present the Session Control Protocol and explain its features for service control layer signaling. We discuss some related work in Section IV before the conclusion.

II. ARCHITECTURE OVERVIEW

In order to fulfill the requirements, and to support service providers as new market entrants we have designed our architecture to reside on dedicated server systems interfacing to the networks. A decomposition concept derived from TINA strictly separates the user control, the service control, and the communication control components of the architecture. Service creation relies on an XML based service logic description. Service control uses an advanced multimedia session model. All control components interact via message based interfaces. For the exchange of the control messages we have defined the Session Control Protocol (SesCP) based on the IETF Session Initiation Protocol (SIP). Network adaptation components interface to heterogeneous networks and resources using existing or emerging APIs (e.g. PARLAY, hybrid DVB/GSM gateways).

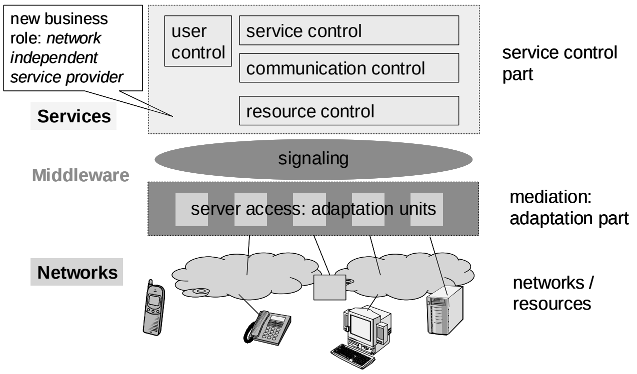

Fig.1. Overview over the architecture

For the realization of the server architecture neither terminal equipment or network infrastructure is affected. The server is designed to use the functionality provided by the existing infrastructure for the control of network spanning services. Because of its position to the different networks we call our approach intelligence on top of the networks.

A. Functional Decomposition

According to the TINA [3] session principles, we separate user session (TINA: access), service session, and communication session. This is expressed by the three main control areas.

In the User Control (UC) area, a User Proxy (UP) maintains the access session for each active user. The UP manages the user profile and preferences, the access to the service control, the user - service interaction, and the address resolution. It is tracking the user configuration and is the point for user side supplementary services.

The Service Control (SC) is independent of user and network specific data. For service execution the service control instantiates a Service Session Manager (SSM) that has a centralized control over the service session.The session itself is described by an object-oriented multimedia call model.

Similarly to the service session, within the Communication Control (CC) area, each instance of a communication session is controlled by a central Communication Session Manager (CSM). The CSM is responsible for the establishment and control of communication relationships in the networks. It selects suitable networks according to user preferences/actual user configuration and to the availability of external interfaces.

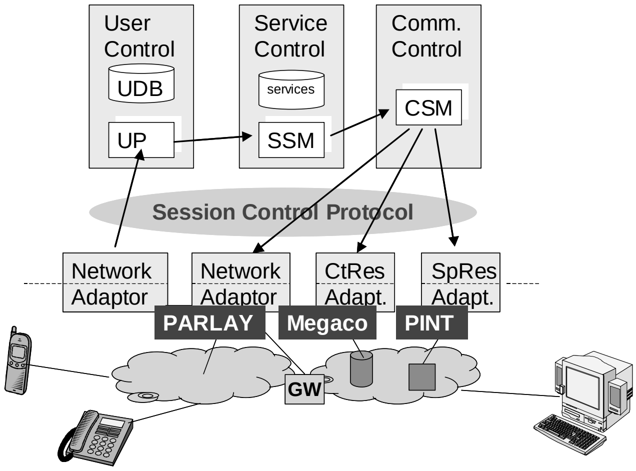

Fig. 2. Functional Decomposition

The interfaces of the server architecture to the different networks are realized by adaptation components (Adaptors, NA), that act as gateways between the internal signaling protocol and the network specific signaling protocols [4]. Our architecture therefore bases on open interfaces (APIs) like PARLAY [5], Megaco or PINT.

B. Service Modeling

Within the architecture the instance of an executed service is described in different levels of detail. The different service models correspond to the three main control areas. Overall, the service description becomes more and more detailed when it is sequentially processed by user, service and communication control. We distinguish the service descriptions that are maintained in each control area (session descriptions) and the service descriptions that are exchanged between the control areas (graphs).

Level	**Description**	**Parameters**	**Examples**	**Area**
access level	user session description	user profiles, service name	user name, service preferen.	UC
user level	user service graph	service name, service param.	MyConference with Alice & Bob	UC-SC
service level	service session description	#party, role, relation, QoS-type	oo-service model (see page 12)	SC
media level	media connection graph	media class, #, relationship	real-time audio, PSTN quality	SC-CC
end-to-end l.	communication session descript.	network types, endpoint addr.	a-b: H.323, G.711; b-c: ISDN, G.711	CC
Network level	connectivity connection graph	network, QoS-value	a-b: H.323, G.711, 64kbit/s	CC-NA

Table 1: Levels of the service description

The User Control deals with all user specific data like the user's registration information, the user's actual network address, and the user's charging schema. After the user has selected a service for execution, only the user's internal address (uid) and the name of the selected service together with some service parameters is passed to the instantiated Service Session Manager.

The Service Control's view on the service is independent of user specific or network specific details. An object-oriented service description reflects the service participants (parties) and their communication relationships. Parties are mainly described by their identifier (uid), and their roles (owner, moderator) and rights during a service. The information exchange between the parties is modeled by different media descriptions. Each media is characterized by attributes that determine the QoS-type, media flow and time. The media description together with the corresponding party identifiers are passed to the Communication Control. Information that is not relevant, is kept in the SSM, e.g. party roles.

C. Service Mapping

The Communication Session Manager (CSM) has to map this abstract description into a concrete and detailed description of all the communication paths that realize the service execution. The communication paths have to be determined according to the possibilities that are provided by the networks.

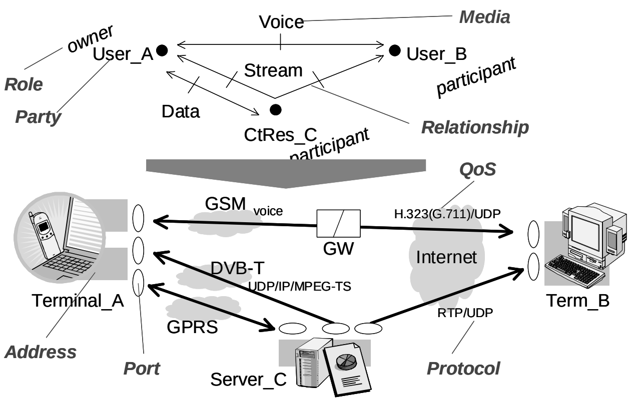

Fig.3. Mapping of the service level service description onto communication level

First the CSM has to collect all necessary data. It retrieves the current terminal and network parameters (network address, port) of each of the parties from the User Control. This information together with the media connection graph received from the Service Control is used for the selection of appropriate networks, i.e. the network adaptors, and the determination of the communication paths. A registry component supports the discovery of Network Adaptors and the negotiation of capabilities [4].

Depending on the availability of network adaptors, communication relationships (represented by media descriptions) have to be split into different communication paths terminated on one side by e.g. an interconnection gateway (GW).

III. SESSION CONTROL PROTOCOL

A. Requirements

To support the exchange of messages between the different components of the server architecture outlined above, we have defined the Session Control Protocol (SesCP) as a novel signaling protocol on the service control layer.

Regarding the overall functionality of the architecture, the following requirements have to be met by the message based signaling.

- Support the setup and handling of complex multimedia communication sessions.
- Flexibility in order to support new and combined services (extensibility).
- Support the service control with respect to the strict separation of functionality (user, service, and communication control components; including the different levels of service description and their negotiation).

Furthermore our considerations were influenced by the probable emerge of an IP-based next generation network. Therefore a novel signaling protocol has to be designed to smoothly interwork with existing Internet protocols and systems.

Regarding these requirements we have selected the IETF Session Initiation Protocol [7] as basis for our signaling protocol.

B. IETF Session Initiation Protocol

The Session Initiation Protocol (SIP) has been specified by the MMUSIC Working Group of the IETF as a proposed standard in 1999 (IETF RFC 2543, [7]). SIP supports party/resource location, invitation to service sessions and negotiation of session parameters for a large range of multimedia communication services. Its main target is VoIP services.

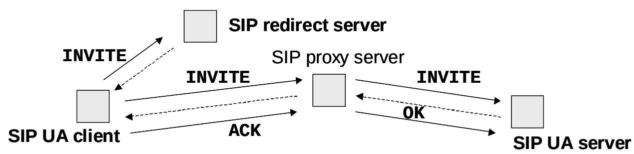

Fig.4. Typical Session Initiation Protocol topology

SIP provides a small number of text-based messages to be exchanged between the SIP peer entities (SIP user agent in a user terminal). Network entities like proxy servers or redirect servers that can be traversed by the messages, can be used for support, e.g. address resolution, service logic execution.

A SIP proxy server processes a SIP request and forwards it to the correct destination. A SIP redirect server only notifies the sender of a request of the destination. Furthermore, a proxy server is able to forward a request to different destinations simultaneously. This forking functionality is used to reach a user that may be present at alternating locations.

The session description in SIP is carried independently of the signaling message as a payload, specified by the Session Description Protocol (SDP). Since the message body is transparent to SIP, it is not restricted to telephony style session but can in general serve as a generic transaction protocol, as already done in the PINT approach [8].

SIP provides a solution fulfilling all the requirements, and in addition by using SIP we can adopt existing SIP components instead of creating new protocol handling entities. Furthermore, the use of SIP bridges to a future use in an Internet dominated NGN environment.

C. SesCP: SIP Based Service Signaling

SIP supports the control of any multimedia communication session since the session description is carried as a payload with the SIP message containers. In this way we are not bound to a fixed set of information flows that determine the capabilities of the service control like in the Intelligent Network (INAP).

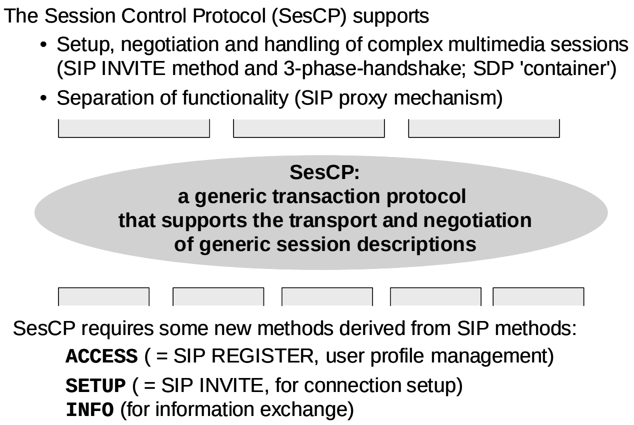

Fig.5. SesCP Principles

The components of our architecture act as proxy servers processing the session description. Each component (e.g. User Proxy, Service Session Manager) adds details to the session description until finally a concrete communication flow is requested from a network adaptor. With this proxy based approach we break the paradigm of the end-to-end character of traditional application layer signaling protocols like INAP. There, signaling is performed in end-to-end transactions between the SSP and the SCP.

To reflect the separate control areas of the server architecture we have enhanced the SIP message set (SIP methods) with some new methods. We derived the ACCESS method for user registration and the management of the user profile from the SIP REGISTER method. Whereas INVITE is used for service session initiation and control, communication control relies on the new method SETUP that initiates the communication session for the network adaptors. For information exchange, e.g. retrieval of the user profile, we have created the INFO method.But overall, all general characteristics of SIP are kept so that SesCP could be processed by any basic SIP intermediate server and the special control servers, when the components of the architecture are distributed over the Internet.

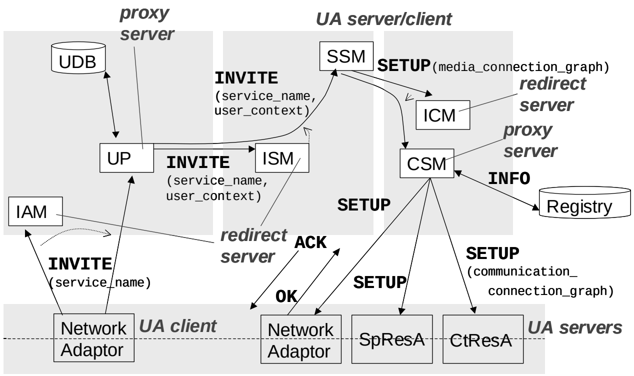

Fig.6. SesCP Signal Flow

The structure of the server architecture described in this paper can be mapped on a SIP topology in the following way. Each network adaptor that is a contact point for users accessing the server acts as a User Agent client issuing INVITE requests to request the execution of a selected service from the service control area. Since all main control components like User Proxy (UP) or Service Session Manager (SSM) are instantiated for a specific service execution, we provide persisting components for the creation and initialization of these control components. These 'initial' components act as redirect servers. For example the INVITE message is first sent to the Initial Access Manager (ISM), which creates the required UP and redirects the INVITE there. The UP proxies the INVITE request and modifies the session description according to user preferences. Finally the INVITE is received by the SSM.

We have split the end-to-end signaling at the SSM to reflect the separation of service and communication session. The service session controls the session initiation and mid call events. The communication session controls the establishment of communication flows in the networks. The INVITE method is used for the initiation of a service session and for mid call signaling. This means a session can be modified by subsequent INVITEs. The INVITE method and the INFO method (for the request of user information and for event notification) are both proxied through the UP.

For communication session signaling the SETUP method carries the connection graphs. The signal flow is similar to the service session. But in addition the Communication

Session Manager (CSM) acts as a forking proxy that distributes the corresponding connectivity connection graph to each selected network adaptor.

D. Service Session Description

The structure of the service description is derived from a generic service model [6] to allow a wide range of information and communication services. All levels of service description that are passed between the control components are a part of this object-oriented model. The model shown above is the basis for the service execution controlled by the service logic within the SSM. The parameters are described according to SDP principles.

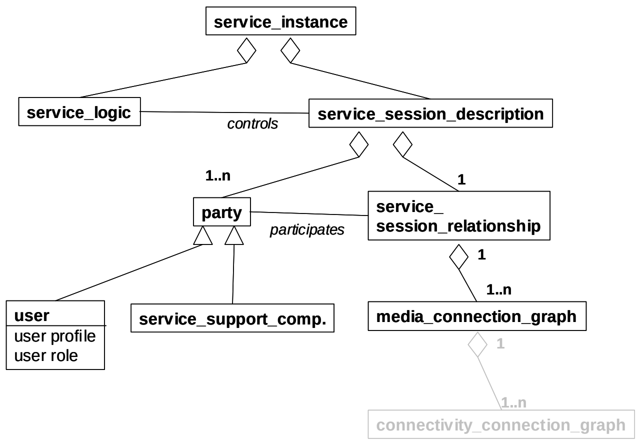

Fig.7. Service Session Description

An excerpt of the description of the service shown in Figure 3 is for example:

```
    si  = 12345     /* session identifier */
sna = 'MyConference'
sur = ssm17         /* ssm address */
sti = now none      /* start stop time */
szo = MEZ
smg = m2 m3         /* media group */

mid = m2            /* media id */       pty=user /* type */
mex = optional      /* existence */      pid=usera /* party id */
mty = stream        /* media type */     pro=owner /*  roles */
mbw = high          /* bandwidth type */        ...
mbw = stream        /* bw guarantee */          pty=user
mse = mediasource /* sender */           pid=userb
mre = usera userb /* receiver */         ...
...                 pty=servicesupport
mid = m3                      pid=mediasource
...        ...
```

E. Negotiation

Service and communication control in a heterogeneous and changing environment does not only need mechanisms for a reliable session description transport but also mechanisms for the negotiation of the session description to dynamically react on the capabilities of the network adaptors that are currently available. Again SIP provides a very well suited mechanism for negotiation with its 3-phase-handshake. The negotiation of capabilities in our architecture relies on this basic principle and in addition we have enhanced this mechanism to allow atomicity.

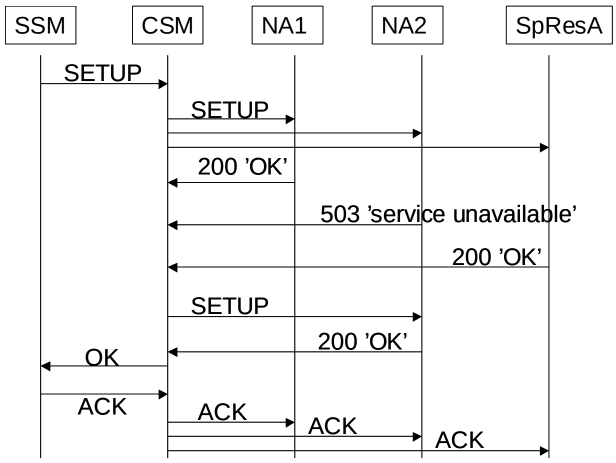

Fig.8. SesCP Negotiation

In the 3-phase-handshake a transferred set of parameters (e.g. media connection graph in INVITE/SETUP) is not valid, i.e. causes the establishment of a communication flow, until the ACK message is received by the server (e.g. NA). To negotiate the capabilities the server can issue respond messages to the INVITE/SETUP message indicating alternative parameters. The OK respond message (code: 200) finally accepts the parameters requested in the INVITE/SETUP.

Different to SIP, SesCP communication session signaling comprises several servers (network adaptors) as signaling endpoints at the same time. The CSM acts as a forking proxy distributing the connectivity connection graphs to the NAs. A complete connection graph is only established (ACK), when all mandatory parts could be set up by the NAs. Otherwise the current communication relationships remains unchanged and an alternative connection graph has to be issued (SETUP).

These negotiation and atomicity principles are well known from other complex signaling systems. In 1993 the ITU-T has already standardized a protocol for commitment, concurrency and recovery during capability exchange in its recommendations X.851/852 [9].

IV. Related Work

The use of SIP as a general transaction protocol for service signaling has also been recognized by other approaches, but with a different focus, as we will discuss in the following.

In IETF PINT [8] SIP is used for the transfer of session description in IP networks between a PINT user (SIP client) and a PINT gateway (SIP server) to initiate PSTN services from the Internet, e.g. 'click-to-dial'. In order to enhance SIP based service systems with IN services control, [10] proposes a protocol for the interaction of SIP servers and IN SCPs carrying INAP operations over SIP. Both approaches are dedicated to a special class of services.

ICEBERG [11] similar to the architecture described in this paper focuses on a network independent service provisioning, but using the Internet as a core switch for signaling and content. SIP is only used for the initial setup of service sessions, and the session handling and communication establishment considerably differs from the one described in this paper. ICEBERG proposes a completely distributed session control supported by a separate signaling protocol for lightweight call sessions, i.e. there is no central service control, but all participants periodically announce their current session description.

Compared to traditional service signaling, i.e. INAP based signaling the use of SesCP brings more flexibility for service creation, since we are not limited to a set of information flows that are restricted by the call model. Our solution supports a full session control in contrast to supplementary features in the IN.

V. Conclusion

We have described an advanced service architecture that realizes service control in heterogeneous networks with different end user terminals. Services are described independently of network specific details. In this way our architecture supports the convergence of networks allowing services that span multiple networks. From a service providers point of view the separation of services and networks supports a new business role. Using our server system providers could offer their services independently of network providers.

We have proposed the Session Control Protocol (SesCP) as a transaction protocol for service control layer signaling. SesCP is derived from SIP and therefore supports the transport and negotiation of generic session descriptions.

Furthermore, the use of SIP bridges to a future use in an Internet dominated next generation network environment. Relying on an IP based signaling protocol, our architecture would be easily deployable in a NGN infrastructure that consists of one packet-based, i.e. Internet-based, backbone interconnecting heterogeneous access networks.

So far, for our concepts we have assumed a closed server architecture that is controlled by one single service provider for a limited group of registered users. In this way scalability issues and interworking mechanisms between different providers remain for further study.

References

[1] M. Finkelstein, J. Garrahan, D. Shrader, and G. Weber. The Future of the Intelligent Network. IEEE Communications Magazine Vol. 38 No. 6, June 2000, pp. 100-106.

[2] W. Kellerer. A Versatile Network Independent Server Architecture for Multimedia Information and Communication Services. In Proceedings SmartNet 2000, Sixth IFIP Conference on Intelligence in Networks, Vienna, Austria, September 2000.

[3] TINA Service Architecture, Telecommunications Information Networking Architecture Consortium, Http://www.tinac.com, 1997.

[4] W. Kellerer, P. Sties. Signalisierungsplattform zur netzübergreifenden Dienststeuerung in heterogener TIME Infrastruktur. In Proceedings KiVS2001, GI/VDE/ITG conference on communications in distributed systems, Hamburg, Germany, February 2001.

[5] PARLAY group. HTTP://www.parlay.org/.

[6] P. Sties, W. Kellerer. A Generic and Implementation Independent Service Description Model. In proceedings ICDCS-21W 2001, 21st International Conference on Distributed Computing Systems Workshops, Distributed Dynamic Multiservice Architecture Workshop (DDMA), Mesa, Arizona, USA, April 2001.

[7] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. SIP: Session Initiation Protocol. IETF Request for Comments, RFC 2543, March 1999.

[8] S. Petrack, L. Conroy. The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services. IETF Request for Comments, RFC 2848, June 2000.

[9] ITU-T Recommendations X.851/2. Information Technology Open Systems Interconnection - Protocol For The Commitment, Concurrency And Recovery Service Element. ITU-T, 1993.

[10] E. Evloguieva, and R. Glitho. Plugging IN SCPs in SIP Networks. In Proceedings IN2001, IEEE Intelligent Network Workshop, Cape Town, South Africa, May 2000.

[11] H. Wang, A. Joseph, R. Katz. A Signaling System Using Lightweight Call Sessions. In Proceedings IEEE InfoCom 2000, TelAviv, Israel, March 2000.

Biography

Wolfgang Kellerer received the Dipl.-Ing. Univ. degree in Electrical Engineering and Information Technology from the Technische Universität München (TUM), Germany, in 1995. Since 1996 he has been a member of the research and teaching staff at the Institute of Communication Networks (LKN), Prof. Jörg Eberspächer, at TUM. His research interests are service control architectures beyond IN and the use of FDT for service development. Currently he is working on his PhD/Dr.-Ing. degree. In his PhD thesis he presents an advanced architecture for an intelligent network server to enable network independent service control across heterogeneous networks.