# Advances in Network-Supported Media Delivery in Next-Generation Mobile Systems

*Frank Hartung, Norbert Niebert, Andreas Schieder, and René Rembarz, Ericsson Research*
*Stefan Schmid and Lars Eggert, NEC Europe Ltd.*

## ABSTRACT

This article presents new concepts for network-supported media delivery in mobile networks. Automatic composition and merging of networks are central parts of these concepts. Media delivery is no longer an end-to-end service that only uses the network as an IP transport. Instead, these concepts create a service-aware network and provide customized delivery support through per-service overlay networks. They also integrate specialized processing nodes as part of the delivery topology, which include transcoders but also more complex processors, such as localized program insertions or personalized spam control. This article describes the underlying concepts and how these new network capabilities for media delivery services are requested, invoked, and managed.

## INTRODUCTION

Mobile networks and services have grown beyond voice-only communication services and are currently evolving towards IP-based multiservice and multipurpose networks. This trend is also reflected in an increasing number of new capabilities that improve network performance, protocols, and services. Examples include multiuser broadcast services and transmission improvements that enable high data rates and provide the means for multimedia services over mobile networks. Today, mobile networks are widely used for services such as picture messaging (MMS), video streaming, blogging, music downloads, and mobile TV. A few years ago, all these services required processing power and network bandwidth that was only available on stationary PCs.

The development and deployment of these services thus far mostly followed an end-to-end paradigm where services were controlled and managed only by the endpoints (i.e., servers and clients). The network connecting the endpoints acted only as a simple IP transport and did not add further value or functionality to the service. This end-to-end principle, however, has limits when many devices attempt to access a service at the same time, especially when these devices are very heterogeneous (e.g., mobile terminals with different capabilities that connect via different access networks). In this case, devices have to be treated individually, and large numbers of users can only be supported with high-performance server farms and very high bandwidths that allow concurrent transmission of many individualized data formats.

One solution to this issue is to enable the network itself to contribute to media delivery, for example, through media processing capabilities along the communication path. Examples of such media processing include *transcoding*, that is, the adaptation of content to different device capabilities or network conditions, or *caching*, that is, the temporary storage of original or transcoded content at locations other than the original servers. Other more complex functionality becomes possible as well. Media processing nodes exist in mobile networks today; for example, media gateways or MMS relay servers. However, currently such support nodes are specialized for supporting specific services. Consequently, they are not able to support other services and cannot be controlled by the endpoints.

In order to allow arbitrary services to invoke, configure, and deploy support processing functionality in the network, general concepts are needed that allow easy, on-the-fly integration of such nodes into deployed networks and also allow dynamic invocation on a per-service or per-user basis. These concepts have been developed in the partly EU-supported *Ambient Networks* project [1–3], which creates an overall network control architecture especially targeted at feature-rich IP-based mobile networks.

The remainder of this article is organized as

 0163-6804/06/$20.00  

follows. The next section introduces the general concepts of Ambient Networks; we detail the use of service-specific overlay bearers to provide customized media transport by incorporating network-side processing functions. Next, we outline some example processing functions and present a deployment example for SIP-based voice communication. The article discusses related and future work, and the last section summarizes and concludes this article.

## NETWORK SUPPORT FOR MEDIA DELIVERY

Mobile network users experience different network capabilities and connectivity qualities as they move between different access networks. The ability to adapt a service, for example, a media delivery service, to current individual network and endpoint conditions is thus an important requirement. The process becomes more complicated through the involvement of transit or access networks that have to jointly provide media delivery service to roaming users wherever they are and whatever access they use.

Ambient Networks addresses these challenges with a novel control overlay. This new ambient control space (ACS) registers and organizes all control functionality in a network. Figure 1 illustrates the logical organization and main features of the ACS, including some example functionality and the three ACS interfaces.

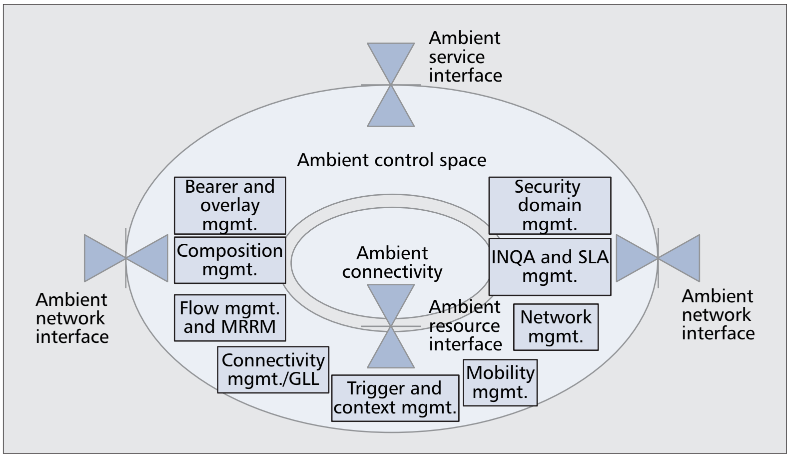
Key features of the ACS are the ambient network interface (ANI), the ambient service interface (ASI), and the ambient resource interface (ARI).

The ANI is used for communication between the control spaces of different networks. It allows advertising and making control space functions available to other networks, and to connect the functions of an ACS with functions of another ACS. Thus, it enables the dynamic cooperation of different ACSs, referred to as composition, which is a major concept of Ambient Networks. In short, the ANI is the interface between different Ambient Networks.

The ASI exposes the connectivity and control functions in a uniform way for use by upper-layer applications and services. It allows end-user applications and services to issue requests to the ACS concerning the establishment, maintenance, and termination of end-to-end connectivity and the transport service between end-systems. The ASI also includes management capabilities and provides the means to make network context information available to the applications. In short, the ASI is the interface between service providers, service receivers, and Ambient Networks.

Finally, the ARI is located between the ACS and the connectivity layers. It offers control mechanisms to the ACS that are used to manage connectivity-plane resources. The resources are accessible through an abstraction layer, which allows the ACS functions to remain technology independent. In short, the ARI is the interface between the control functions and the underlying transport network, such as an IP network.

Note that this architecture represents an idealized vision of a future network architecture that represents a desired ideal architecture to manage individual control functions that improve today's networks. Similarly, the ACS interfaces are idealized reference points that can be implemented by different real interfaces and protocols in different ways. Two examples of architectural components included in the Ambient Networks concepts and driven by the project in standardization are the Host Identity Protocol (HIP) work in the IETF and the current 3GPP Study Item on network composition [4].

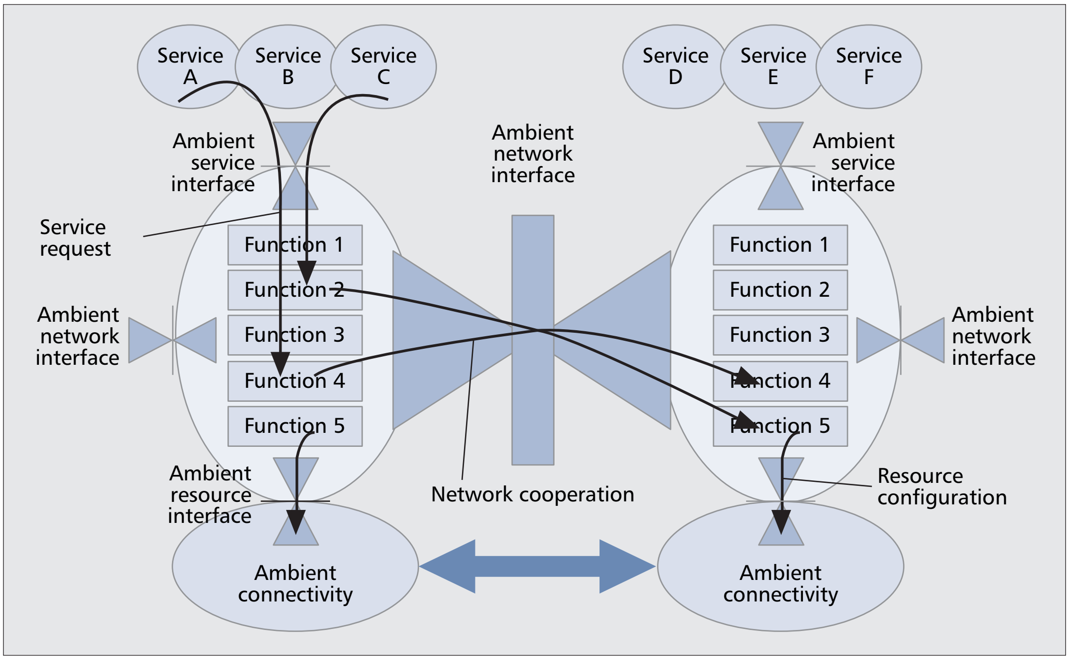



■ **Figure 1.** *Illustration of the modularized control space and its interfaces.*

The ACS interfaces play a major role in the implementation of services, which often involve cooperation between functions residing in different ACSs and networks. Figure 2 illustrates the general service provisioning concept in Ambient Networks. Depending on which functionality is requested via the ASI, different functions of the ACS are invoked to implement it. The services requesting certain functionality are unaware of the internal ACS structure. In some cases, the functions in the local ACS can already fulfill the service request. In other cases, functions in other ACSs need to be involved, because the required functionality is not available locally or the cooperation and coordination of different FEs is necessary. In these cases, the ANI is the interface used for the cooperation between networks. In the example in Fig. 2, the requested functionality requires the local ACS (on the left) to contact functions residing in a second ACS (on the right) via the ANI. These functions use the ARI in the respective local ACS to control and configure resources in the network to implement the requested functionality. Note that this scenario is an example that clarifies the role of the different interfaces. The illustrated sequence of actions is not fixed.

Because the ACS consists of cooperating control modules that serve different purposes, its interfaces are structured according to the division between control tasks. Thus, they are rather seen as reference points, each implementing a modular set of interfaces serving different control purposes. Each particular implementation of a reference point may include a different set of interfaces, depending on the functionality present in the ACS to which the reference point belongs.

Commands issued through the ARI configure connectivity resources or perform additional tasks, such as configuring transcoders. Once the end-to-end connectivity has been established, the corresponding point of application attachment (e.g., an extended "socket") is returned to the application.




■ **Figure 2.** *The role of the three control space interfaces in service provisioning.*

One major issue with today's networks is increasing heterogeneity. To address this problem, an abstraction of these heterogeneous network technologies is employed by Ambient Networks. Applications do not need to implement different mechanisms to set up and maintain end-to-end connections in different connectivity technologies. They simply create an abstract connection entity. ACS functions are responsible for setting up and maintaining this end-to-end connection across any given connectivity technology. This model ensures that the implementations of control space functions can remain as generic as possible.

The connectivity abstractions define different views on the underlying physical connectivity that provide different levels of detail [5]. They present a generic, technology-independent view of the underlying network connectivity to ACS control functions. The control space functions interact with network resources through the ARI, without knowing the implementation details (e.g., to set up or configure network resources).

The actual transfer of data over these resources is referred to as *flow*. In addition, the Ambient Networks abstraction model provides a higher-level view of the connectivity to the application or service interacting with the ACS via the ASI: a *bearer* is the communication primitive provided on this level. It hides the implementation of the connectivity and provides an end-to-end transport service to the user applications and services.

For certain applications (e.g., file transfers) the transport service provided by a bearer can be quite simple, requiring very little in addition to what the underlying flow provides. (This is similar, for example, to how the UDP transport protocol provides little more than pure IP service to the applications.) For other applications, such as media delivery, bearers can become more complex. They may involve functionality such as transcoding or specialized media routing. In such cases, bearers use the advanced capabilities provided by the control space functions, such as mobility, quality-of-service (QoS) reservation, or media adaptation.

From the service or user point of view, bearer establishment takes place as follows: once a bearer has been requested via the ASI, the control space functions use abstract topology and capability information maintained by the ACS to determine which flows can be reused or need to be set up in order to construct the bearer. In addition, the control space functions can use the ANI to negotiate with other networks to make additional resources available to the new bearer. Commands issued through the ARI configure connectivity resources or perform additional tasks, such as configuring transcoders. Once the end-to-end connectivity has been established, the corresponding point of application attachment (e.g., an extended "socket") is returned to the application.

Once these steps have completed, the service can use the application point of attachment to transparently transmit data to its peers. Note that application-level signaling traffic, such as RTSP, is also transmitted through such bearers; the ANI is purely used for inter-ACS communication.

## Transport Overlay Bearers for Media Delivery

In a traditional end-to-end media delivery concept, as illustrated in Fig. 3, no interface exists through which service providers or end users can request specific functions from the underlying (transport) network — the network only provides a simple best-effort transport service.

This is different in the service model supported by Ambient Networks, as illustrated in Fig. 4.

The ASI offers advanced media delivery capabilities that may be present in the network to enable services that go beyond best-effort transport. It is the point of contact for any application or service that wants to take advantage of an Ambient Network's capabilities. The network nodes that provide media processing functionality are called *MediaPorts*. The ASI allows applications to request information about advanced capabilities of the network (e.g., MediaPorts), to choose specific support resources for a transport bearer, and finally to transmit media across the bearer using these support functions. It also allows applications to modify the characteristics and requirements of endpoints and bearers, to add and remove endpoints, and to terminate bearers. Thus, the ASI allows management of bearer-based media delivery functionality over the entire lifetime of a service.

The ASI can also be used to reserve resources for a service that has not yet been invoked: a service provider can anchor services in the network, and thus be sure the network provides enough resources and the required support functions for the service. The ASI thus exposes the service interface of the Ambient Networks media delivery and transport support.

The ASI allows user applications or services to negotiate and control Ambient Networks transport services. More precisely, it allows a service to:

• Request information about the available resources and support an Ambient Network can provide to a service. For that purpose, the service provider specifies the requirements of the service, including the required MediaPort capabilities and transport QoS. The Ambient Network replies with one or more possible service-level agreements (SLAs), which offer the requested transport service.

• Select one of the offered SLAs and thus establish the "contract" for the service. Note that this contract negotiation is carried out between the service application and the network in an automated fashion, which does not require human interaction. From that moment on, the Ambient Network is ready to provide the desired transport service and network support or processing capabilities to the service. It establishes a so-called Service-Specific Overlay Network (SSON) for that purpose (see below for details on SSONs).

• Reselect a new SLA among the set of SLAs previously offered by the Ambient Network. If the properties or requirements of the service change significantly and are not covered by the existing SLAs, the SLA negotiation process starts again.

• Add or remove endpoints (i.e., the clients or receivers of a service) to/from the SSON, (e.g., in a multiterminal or multi-user scenario). The capabilities and properties of the endpoints (e.g., supported codecs or bit-rates) are signaled as part of this request.

• Modify the capabilities or properties of a endpoint.

• Tear down the SSON and release all its resources.

• Register a MediaPort with an Ambient Network. The capabilities of the MediaPort are communicated to the Ambient Network, which stores the information for future use when setting up or modifying SSONs that might use this MediaPort.

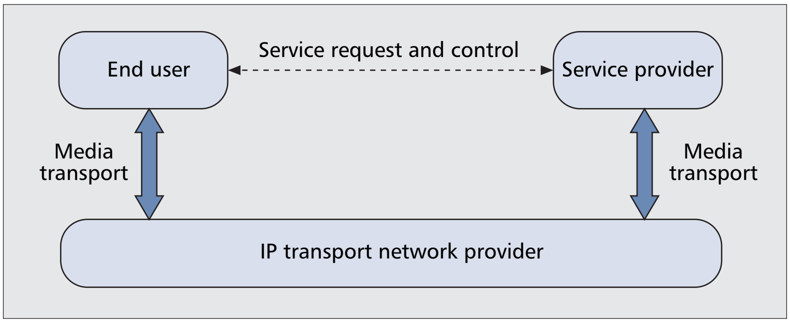
• Change the properties of a MediaPort (i.e., modify the information about its capabilities and status).

• Remove a MediaPort by deregistering it.

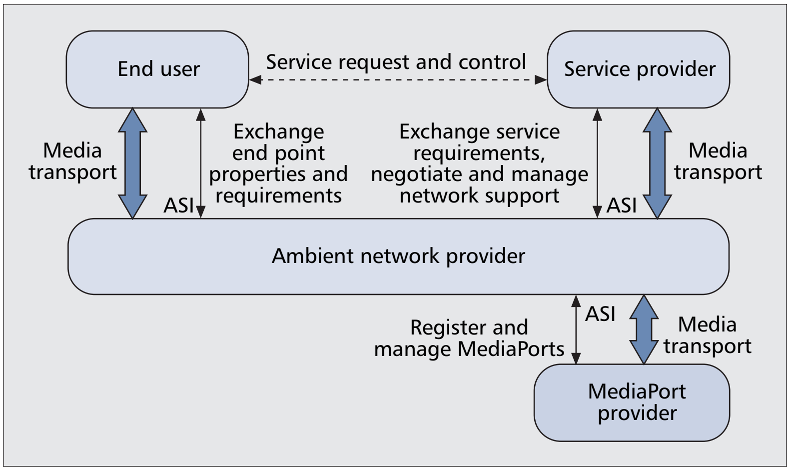
Although the ASI exposes a rich service interface to end-user applications and services that allows them to tailor the network resources and media delivery functions according to their specific needs, the complexity and actual mechanisms of achieving this "network customization" remains invisible to them. To provide this customization on a per-service basis, the control space configures separate virtual or "overlay" networks for each service.

The use of such SSONs also enables the inclusion of media processing functions into the end-to-end communication paths [6]. Because routing at the overlay level can be fully controlled to satisfy the service for which the SSON was established, any support function in the network can be flexibly included into the end-to-end path where appropriate, independently of its physical location. Examples of such network-side functions (MediaPorts) are provided below.

The establishment of an individual overlay per service may seem excessive. However, an important benefit is that the routing of flows (as defined above) for a service, including routing through media processing functions, becomes

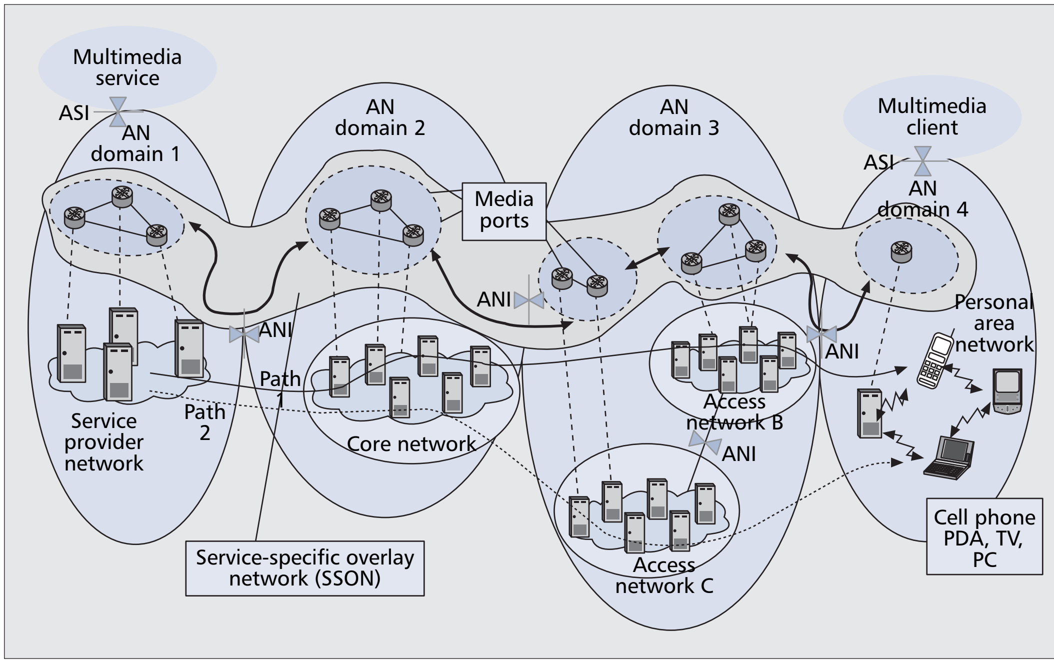



■ **Figure 3.** *End-to-end media delivery without processing support by the network.*




■ **Figure 4.** *Ambient networks media delivery with processing support in the network.*

*SSONs are dynamic and can be reconfigured in response to changing network conditions (e.g., end-to-end delay or congestion), changes in network context (e.g., handovers), or changes to user policies (e.g., user profiles, device capabilities, media/content types).*




■ **Figure 5.** *Service-specific overlays connect endpoints of a media delivery service and include MediaPorts for network-based processing.*

straightforward once the SSON is established. Virtualization of the network using SSONs helps by dividing the routing and management challenge into multiple isolated problems that can be addressed more easily. Depending on the characteristics and requirements of the media service, the SSON routes media streams through different media processing functions and/or network domains (e.g., WLAN or UMTS). Further, different components of a service (e.g., different media streams) can be routed over different paths, as proposed in [7]. The result is the seamless provision of value-added services inside the network, beyond what is possible today. This concept is illustrated in Fig. 5.

SSONs are tailored to the specific service they have been established for and thus include only those nodes that are required to provide the requested service. Among all MediaPorts in an Ambient Network, the most suitable ones are selected using appropriate metrics, which consider capabilities, costs of using the processing function, transport costs to route data to and from the MediaPort, and the current processing load of the MediaPorts.

For this, the ACS contains dedicated MediaPort information function that enables efficient lookups of MediaPorts based on media processing capabilities and properties. The MediaPort information function maintains a directory of existing MediaPorts and has the capability to dynamically search for MediaPorts, for example, in a restricted neighborhood near MediaPorts that are already involved in a given SSON [8, 9].

SSONs are dynamic and can be reconfigured in response to changing network conditions (e.g., end-to-end delay or congestion), changes in network context (e.g., handovers), or changes to user policies (e.g., user profiles, device capabilities, media/content types). Adaptation of SSONs can happen on different timescales. "Fast" adaptation is required to respond to critical changes that would otherwise disrupt a service, such as changes to the underlying network (e.g., if a link is congested). Due to stringent performance constraints, mainly localized optimizations (e.g., a route change or the inclusion of additional media processing capabilities) that do not incur heavy signaling or long delays can be considered for this type of adaptation. On the other hand, "slow" adaptation can be applied to optimize operation in response to noncritical changes, for example, when a MediaPort joins an Ambient Network or when a user device changes its point of attachment in the network. In such cases, there is either no immediate need for the SSON to be changed (e.g., when a MediaPort joins an Ambient Network) or other underlying network mechanisms take care of the problem at hand first (e.g., in the case of a network handoff it is assumed that an underlying mobility solution takes care of the immediate handoff). Still, in response to such events, it might be beneficial to reconfigure or adapt the SSONs in a slow manner in order to optimize its overall performance in the long run. For example, SSONs may adapt their topology and/or MediaPorts after a network handoff of a mobile node, in order to reduce end-to-end transmission delays and/or provide newly required media processing capabilities, respectively.

The functionality for the establishment and control of SSONs is provided by the multimedia routing logic (MRL) of the ACS. It manages the entire lifecycle of the SSON, from the initial setup until the final removal, and controls the reorganization and adaptation of the SSON. This may include reconfiguration of the overlay topology and routing in order to optimize the network when changes regarding network conditions or user and service context become apparent. It adaptively selects suitable routes for the media flows of a service by influencing the routing decisions in the overlay nodes that constitute an SSON.

The physical nodes that constitute an SSON include the end-systems (server hosts and clients devices) as well as the MediaPorts that provide the desired media processing functions inside the network. A physical node can be part of many SSONs at the same time and implement one or more media processing function per SSON.

Each node participating in an SSON requires basic overlay network functionality for packet handling at the overlay level. This functionality within a node is called the overlay support layer (OSL). The OSL is responsible for sending, receiving, and forwarding packets at the SSON level. It provides a common communication abstraction (overlay-level network protocol and addressing) to all nodes of the SSON, so that they can communicate with each other independent of their differences regarding the underlying protocol stacks and technologies. The OSL is responsible for the demultiplexing of incoming data packets to the correct SSON and, if special media processing is required, to the appropriate media processing function. The OSL maintains SSON-specific routing tables, which are managed by the MRL of the ACS.

## EXAMPLE OF MEDIAPORT FUNCTIONALITY

The concept of service-specific overlays provides a generic mechanism for the provisioning of value-added media processing functionality inside a network. It enables the inclusion of arbitrary MediaPorts into the end-to-end communication paths of a service.

MediaPorts can have different functionality and capabilities. Generally these can be classified as media processing capabilities, but also as customization capabilities, security-related capabilities, or any other capabilities that may monitor, process, or reroute data packets belonging to a service. Moreover, it is likely that there will be applications and deployments of the concept that the developers did not envision, and the concept is general enough to allow that. A few examples of possible MediaPort functionality are:

**Overlay routing:** Enabling basic transport routing functionality on the overlay level, that is, controlling the overlay path over which data is transported between MediaPorts and endpoints.

**Transcoding:** Conversion between different media formats, for example, between different video or audio formats, or between different bit rates using the same codec.

**Caching:** Temporary storage of arbitrary data belonging to a service, for example, to help to deliver seamless services to end users that suffer from intermittent connectivity.

**Flow synchronization:** Enabling synchronized playback of media flows that are routed over different paths, for example, the audio and video flows of a mobile TV program that are routed over different paths in order to be processed by different MediaPorts but need to be synchronized prior to the playback.

**Personalization:** Customizing a service according to end user preferences, for example, by adding or removing content or functionality.

**Localization:** Enhancing a service with location-dependent information or options.

**Parental control:** Protecting minors by including a MediaPort that allows customized filtering and removal of content or services that are considered inappropriate for them.

**Spam and virus protection:** Enhancing a service by including a MediaPort for spam filtering or virus scanning that detects unwanted or illegal content and malicious data and transparently removes it.

**Lawful intercept:** Allowing transparent interception of individual user flows for examination by the authorities.

The last three examples are conceptually different from the previous ones because they provide functionality that is usually not requested by the service provider. This may indicate that the concept of SSONs can also be used to enable functions that the end user or network provider wants or needs to provide (e.g., for legal reasons).

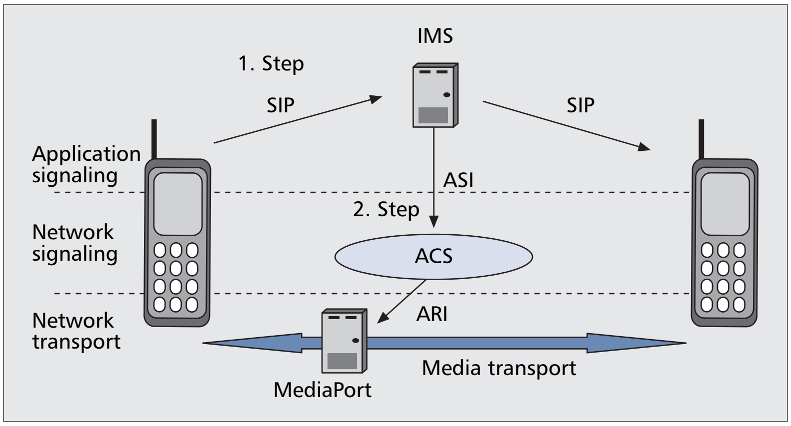
Since MediaPorts have access to data that is transported between endpoints, they need to be trusted. For example, if protected or encrypted audiovisual streams need to be transcoded, the transcoding MediaPort needs to receive the decryption key in order to be able to perform its task. Authentication and authorization of them is thus required, but is not further discussed here.

*Since MediaPorts have access to data that is transported between endpoints, they need to be trusted. For example, if protected or encrypted audiovisual streams need to be transcoded, the transcoding MediaPort needs to receive the decryption key in order to be able to perform its task.*

## DEPLOYMENT EXAMPLE

This section outlines a deployment example of the media delivery framework described above. It illustrates how peer-to-peer services can utilize this concept and also shows how legacy devices and protocols can interact with the new mechanisms described here.

Figure 6 shows an example scenario of a SIP-based voice service (e.g., IMS). It assumes that the mobile user, who establishes a voice session using his SIP client, wants to record the session, but does not have the required capability in his end device. It further assumes that the SIP server/proxy that is intercepting the session initiation messages is aware of this user requirement. At connection setup time, when one of the endpoints initiates the voice service (step 1 in Fig. 6), the involved SIP server/proxy, which implements the ASI, detects the need for a network based recording capability. Using the ASI, it requests and negotiates a transport service from the underlying network that includes a recording node (step 2 in Fig. 6). When receiving the request, the control space of the Ambient Network checks for the availability of a recording node by searching its MediaPort database or using MediaPort discovery mechanisms. If such a node (and all other required network resources) is available, the ACS offers the establishment of a suitable SSON for the service. After the final agreement on the establishment of the SSON is made between the control space and the SIP server/proxy, the SSON is actually established, which includes the recording MediaPort in the end-to-end path between the SIP peers (authentication and authorization of the recording MediaPort are not discussed here, but need to be considered). The recording node, which is




■ **Figure 6.** *SIP-based deployment example of ambient networks.*

now part of the SSON, is configured using the ARI interface. In addition, other overlay routing nodes in the overlay network (not shown in the figure) are configured, for example, by establishing overlay routing tables.

After the SSON has successfully been set up, the voice service starts using legacy streaming protocols (shown as "media transport" in Fig. 6). The recording MediaPort records the streams. At the end of the conversation, it may transmit the recorded data to a repository managed by the end user. In addition, the SSON is torn down after the end of the service, because it is no longer needed. The teardown can either be initiated by the SIP server/proxy, again using the ASI interface, or it can be triggered by a timer after inactivity of a certain time.

While the example shown here is a peer-to-peer service, the concept is also applicable to client-server based services, such as content delivery to single users or large user groups. Within the Ambient Networks project, a prototype demonstrates the feasibility of the described mechanism for a content delivery scenario, where the signal of a media source is adapted to different end devices by inclusion of transcoding MediaPorts, and where the streams are split and rerouted to different end devices, depending on device mobility.

## Related Work

Several related research efforts propose solutions to support the delivery of multimedia data across various different network domains and to a heterogeneous set of end-user devices. Some of them also support media adaptation or transcoding through proxies at strategic locations inside the network in a static way [10].

The IP Multimedia System (IMS) standardized by the 3rd Generation Partnership Project (3GPP) is probably the most widely deployed system for the delivery of multimedia data to mobile user. IMS also has the ability to include certain media-processing functions (media gateways) into the communication paths. The difference with regard to the SSON-based media delivery framework of Ambient Networks is that IMS is restricted by its static configuration and lack of support for easy adaptation to specific service contexts. Furthermore, IMS also lacks the ability to route individual media flows of the same session over different paths, which is an essential requirement for multimedia transport across heterogeneous networks and end-user devices.

In general, overlay networks have the advantage that they can provide applications and high-level services with an abstraction of the underlying network that is tailored to their needs. Therefore, they are more flexible than IMS and allow dedicated application components, hosted on an overlay node, to perform some service-specific processing.

Other research projects, such as RON [11], OverQoS [12], and QRON [13], also make use of the concept of overlay networks for the purpose of media delivery. However, the focus of these proposals is limited to improving QoS, primarily by routing around problem spots in the underlying networks. Moreover, these proposals route data based on IP addresses, independent from the application-specific components, and do not take the characteristics of the service, user, or network constraints into account when deciding on the best path. The concept presented here extends the concept of service overlays and also allow the inclusion of support nodes, such as MediaPorts, into the end-to-end transmission path on a per session or even flow basis.

The inclusion of such processing nodes has been previously proposed [14]. These proposals include nodes into overlay service paths that supply functionality such as media transcoding and language translation. However, these proposals do not allow us to extend such overlays when networks merge and do also not address aspects such as discovery of support nodes.

Other proposals [15] discuss the concept of middleboxes, which are intermediate devices enforcing application-specific policy-based functions, such as packet filtering, Virtual Private Network (VPN) tunneling, application-level gateways (ALGs), intrusion detection, or security. Thus, they are more restricted than MediaPorts, and more targeted to fulfill control than media processing purposes.

## Conclusion and Future Work

This article has presented concepts for network-supported media delivery in next-generation mobile systems that have been developed in the Ambient Networks project. Based on the idea of service-specific overlay networks (SSONs) that provide customized media delivery to individual services, one main aspect of these mechanisms is the straightforward integration of network-based processing and adaptation support capabilities. This allows moving functionality that is traditionally realized in the server or client into the network, which can reduce system loads at servers and clients and increase the scalability and performance of mobile networks.

The basic functionality of the ambient service interface (ASI), which extends traditional transport interfaces by allowing end-user applications and services to tailor the network resources and media delivery functions according to their specific needs, have also been described.

In the first phase of the project, the general concept of SSONs and their management, as described in this article, has been developed. During the next phase of the project, the actual processing functionality in MediaPorts will be investigated and the concept refined in general.

## ACKNOWLEDGMENTS

This article is a product of Ambient Networks, a research project partly supported by the European Commission under its Sixth Framework Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks project or the European Commission.

## REFERENCES

[1] N. Niebert *et al.*, "Ambient Networks: An Architecture for Communication Networks Beyond 3G," *IEEE Wireless Commun.*, vol. 11, no. 2, Apr. 2004, pp. 14–21.

[2] B. Ahlgren *et al.*, "Ambient Networks: Bridging Heterogeneous Network Domains," *Proc. 16th IEEE Symp. Pers. Indoor and Mobile Radio Commun.*, Berlin, Germany, Sept. 2005.

[3] N. Niebert *et al.*, "Ambient Networks: a Framework for Future Wireless Internetworking," *Proc. IEEE VTC 2005 Spring*, Stockholm, Sweden, May–June 2005.

[4] 3GPP TR 22.980, Network composition feasibility study

[5] B. Ahlgren *et al.*, "Names, Addresses and Identities in Ambient Networks," *Proc. 1st ACM Wksp. Dynamic Interconnection of Networks*, Cologne, Germany, Sept. 2005, pp. 33–37.

[6] J. Rey *et al.*, "Media Aware Overlay Routing in Ambient Networks," *Proc. 16th IEEE Symp. Pers. Indoor and Mobile Radio Commun.*, Berlin, Germany, Sept. 2005.

[7] J. Chakareski and B. Girod, "Rate-Distortion Optimized Packet Scheduling and Routing for Media Streaming with Path Diversity," *Proc. Data Compression Conf. 2003*, pp. 203–12.

[8] E. Asmare, S. Schmid, and M. Brunner, "Setup and Maintenance of Overlay Networks for Multimedia Services in Mobile Environments," *Proc. 8th Int'l. Conf. Mgmt. of Multimedia Networks and Services*, Barcelona, Spain, Oct. 2005, pp. 82–95.

[9] B. Mathieu, M. Kleis, and M. Song, " A P2P Approach for the Selection of Media Processing Modules for Service Specific Overlay Networks," *Int'l. Conf. Internet and Web Apps. and Services, Wksp. on P2P Sys. and Apps.*, Feb. 2006.

[10] A. Warabino *et al.*, "Video Transcoding Proxy for 3G Wireless Mobile Internet Access." *IEEE Commun. Mag.*, vol. 38, no. 10, Oct. 2000, pp. 66–71.

[11] D. G. Andersen et al., "Resilient Overlay Networks," *Proc. 18th ACM SOSP*, Banff, Canada, Oct. 2001.

[12] L. Subramanian *et al.*, "OverQoS: An Overlay Based Architecture for Enhancing Internet QoS," *Proc. 1st NSDI*, San Francisco, CA, Mar. 2004.

[13] Z. Li and P. Mohapatra , "QRON: QoS-Aware Routing in Overlay Networks," *IEEE JSAC*, Special Issue on Service Overlay Networks, 2003.

[14] X. Gu *et al.*, "QoS-Assured Service Composition in Managed Service Overlay Networks," *Proc. 23rd IEEE Int'l. Conf. Distrib. Comp. Sys.*, May 2003.

[15] IETF RFC 3303, "Middlebox Communication Architecture and Framework," Aug. 2002.

## ADDITIONAL READING

[1] X. Gu and K. Nahrstedt, "A Quality-Aware, Composable Service Overlay Network," Dept. of Comp. Sci. report no. 2342, Univ. of IL at Urbana-Champaign, May 2003.

## BIOGRAPHIES

FRANK HARTUNG [M] (Frank.Hartung@ericsson.com) received an M.Sc. from RWTH Aachen, Germany, and a Ph.D. in telecommunications from University of Erlangen, Germany, in 1999. He has been working with Ericsson Research since 1999, now as a senior specialist in multimedia technologies. His main research interests include networking for content delivery, mobile multimedia, mobile TV, and digital rights management (DRM). He is a member of VDE, and was in 2003–2004 interim chairman of the German IEEE Signal Processing Chapter. He participates in Open Mobile Alliance (OMA) standardization.

NORBERT NIEBERT [M] (Norbert.Niebert@ericsson.com) studied at RWTH Aachen and received a Ph.D. degree in the area of network traffic evaluation in 1991. He joined Ericsson 1991 as research engineer in the UMTS research area, and is since 1997 manager of the mobile multimedia networks research department at Ericsson Research. He is a founding member of the UMTS Forum and Wireless World Research Forum (WWRF). Since 2004 he is technical manager of the Ambient Networks project. He has published numerous publications and presentations on conferences dealing with UMTS and beyond networks.

ANDREAS SCHIEDER [M] (Andreas.Schieder@ericsson.com) received his M.Sc. and Ph.D. in electrical engineering from the Technical University of Aachen in 1995 and 2003, respectively. He currently holds the position of a senior research engineer at Ericsson Research, Aachen, Germany. His research interests focus on future network architectures and the evolution of cellular systems.

RENÉ REMBARZ (Rene.Rembarz@ericsson.com) is a research engineer at Ericsson Research, Aachen, Germany. Since receiving his M.Sc. degree from Aachen University, he has worked on the evaluation and enhancement of IP-based signaling in telecom networks. Currently, his work is focused on beyond 3G network architectures, mainly within the IST Ambient Networks project, where he leads the prototyping and simulation work package.

STEFAN SCHMID [M] (Stefan.Schmid@netlab.nec.de) received his Ph.D. in computer science in 2002 from Lancaster University, United Kingdom. He works as a senior researcher at NEC Network Laboratories, Heidelberg, Germany. He has been active on research topics ranging from mobile and wireless networks, active and programmable networks, transport, service overlays, autonomic communications, to network architectures. He is a member of ACM and an active participant in next-generation network standardization fora. He is also a reviewer for numerous journals and conferences, and on the program committee of several international workshops and conferences in the area of networking and communications.

LARS EGGERT [M] (Lars.Eggert@netlab.nec.de) is a senior research staff member at NEC Network Laboratories, Heidelberg, Germany. He received his Ph.D. in computer science in fall 2003 from the University of Southern California. He has worked on research projects ranging from resource scheduling to Web caching, transport protocols, IP security, virtual networks, and network architecture. He is a member of the ACM, and an active participant in the IRTF and IETF, where he serves as Area Director of the Transport Area. He is a reviewer for numerous journals and conferences, and serves on the program committees of IEEE INFOCOM and the IEEE Global Internet Symposium, among others.

*In the first phase of the project, the general concept of SSONs and their management, as described in this article, has been developed. During the next phase of the project, the actual processing functionality in MediaPorts will be investigated and the concept refined in general.*