# An Overlay Networking Solution for Media Delivery in Future Mobile Networks

Frank Hartung, Bertrand Mathieu, Meng Song, Markus Kampmann, Stephen Herborn, Stefan Schmid, Stephan Goebbels, Pablo Gil, José Rey

***Abstract*—The principle of end-to-end network-layer communication is reaching its limits in today's networks in terms of multimedia service provision, especially for multi-user services. This paper describes a concept for media delivery in future mobile networks that uses service-specific overlay networks and media processing nodes in order to optimize media delivery to heterogeneous devices, and to enable discovery and composition of processing nodes. A prototype implementation is presented in order to demonstrate the validity of the proposed concept.**

***Index Terms*— media delivery, service delivery, media overlays, overlay networks, mobile networks, next-generation systems, network support, Ambient Networks**

## I. Introduction

THE delivery of multimedia services and multimedia data has become a major application of the data transport services provided by the Internet. Multimedia data, in contrast to generic computer data, has special properties and poses special requirements on the delivery system, especially when it needs to be delivered in real-time, and when the end user audience is heterogeneous with respect to receiver and network access properties. For example, multi-user services like Mobile TV services shall be delivered, with certain quality of service (QoS) guarantees, to different users using devices with different screen sizes, equipped with different codecs, and connected via different access links with different bit-rates. Thus, media services often require customization and adaptation for different users. This can be done on an end-to-end basis, i.e. by the servers. However, this model does not scale well and has its limits especially for multi-user services. A concept that moves adaptation and processing complexity away from the endpoints, especially the servers, and into the network, is thus desirable, This not only makes services more scalable, it also makes the network a more valuable part of the service delivery chain. Such a concept based on overlay networking principles is outlined in this paper. Overlay networking methodology has been demonstrated as a means to obtain some level of QoS optimisation by bypassing sub-optimal network layer routes, and by introducing routing on a virtual overlay level. The same methodology has also been used to generate and manage distributed network-based media processing services. However, fully integrated and/or standardised solutions do not exist yet.

In the solution presented here, which was developed in the context of the Ambient Networks networking research project [1][2], overlay networks are established on-the-fly and as needed to suit the requirements of a specific multimedia service. These overlay networks can include network-based processing nodes, and can be adapted in response to changes in network conditions or client requirements and capabilities. The solution is interesting for (but not restricted to) personal area networking usage scenarios such that show in Figure 1, in which a user may have several different networked media display devices available to them at various points in time, each of which has different media display and network connection characteristics. Similarly, distributed peer-to-peer and opportunistic communication scenarios may well benefit from the overlay networking concept presented here, although it also covers the classical server-client service delivery concept.

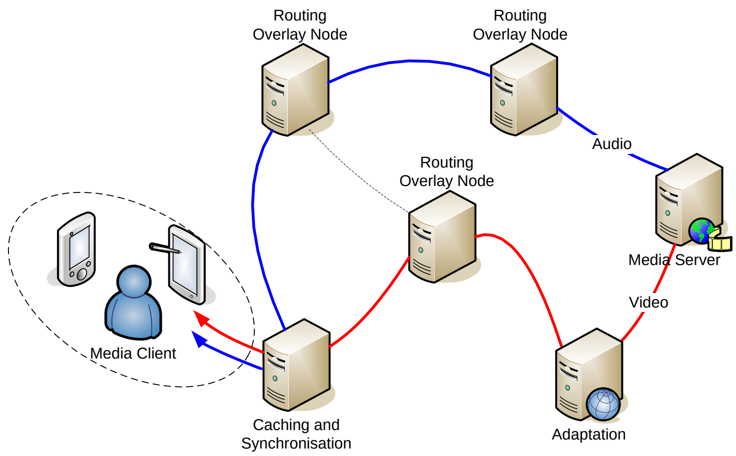


Figure 1: Overlay network media delivery scenario

Manuscript received February 10, 2006. This work was supported in part by the EU Research Project "Ambient Networks" (www.ambient-netwoks.org).
Frank Hartung and Markus Kampmann are with Ericsson Research - Multimedia Technologies, Aachen, Germany.
Bertrand Mathieu and Meng Song are with France Telecom, R&D Division, Lannion, France.
Stephen Herborn is with National ICT Australia (NICTA), Sydney, Australia.
Stefan Schmid is with NEC Europe Ltd., Network Laboratories Heidelberg, Germany.
Stephan Goebbels is with RWTH Aachen University, Aachen, Germany.
Pablo Gil is with Telefonica R&D, Valladolid, Spain.
José Rey is with Panasonic R&D Center Germany GmbH, Langen, Germany.

The paper is structured as follows. Section II presents an analysis of related work and differentiates the solution presented herein. Section III is partitioned into four sub-sections that introduce and explain aspects of the system design. Section IV discusses experimental results from a demonstration prototype, and the paper is concluded in section V with a summary and discussion of future work.

## II. Related Work

The traditional approach to solve the problem of media processing and adaptation is based on processing in the endpoints, that means in the servers and clients. This paradigm can be regarded as state-of-the-art and benchmark against which our proposal of network-side media processing has to be compared. However, it is well known that this traditional end point based approach, which is for example used in Internet streaming services, has its limitations especially for many concurrent users or sessions. Transcoding several hundreds of parallel video streams is a problem for video servers,

Much research work has been developed on the concept of “overlay networks” coupled with multi-path media routing or network-side media processing. X-Bone [15], and its enhancement GX-Bone, dynamically deploys and manages Internet overlays as a way to reduce configuration effort and increase network component sharing. Other research projects such as RON [8], QRON [10], and OverQoS [9] have introduced the concept of overlay networks aimed at improving the quality-of-service (QoS), mainly by routing around problem spots in the underlying networks. Together with the QoS provisioning, the inclusion of network-side media processing resources within the overlay network has been proposed by Nahrstedt et al. using Spy-Net [11], QUEST [12], and Qualay [13]. However such proposals concentrate on routing aspects, do not provide explicit support for mobility, and assume fixed and known support nodes. Additionally, Spy-Net uses flooding to maintain information about network topology, making the scalability of the system questionable.

This paper outlines a proposal that extends the concept of service overlays to allow the inclusion of supporting processing nodes, which we term MediaPorts, into the end-to-end transmission path. These “service specific overlay networks” are set up according to service requirements and include the needed MediaPorts for delivering media content adapted to user context. Since end-users may have different context, streams for users will not always follow the same path. The routing of media data within an overlay network is based on a per-session or even per-flow [6] basis. Both the client components, and to some extent service components, of the overlay networks are assumed to be potentially mobile. As a result of this, media streams may originate, be routed through, and terminate at different networked devices during the lifetime of a single multimedia delivery session.

## III. Media Delivery Using Overlay Networks and Network-Based Processing Nodes

### A. *The need for support from the network*

Today, the Internet serves mainly as a pure transport network. Services are mostly negotiated and managed end-to-end. For example, the bit-rate and codec of a streaming service is selected directly from the streaming server. Temporary storage or caching of data is also performed either by the server or by the client application. However, this end-to-end paradigm reaches limits for mass services, e.g. Mobile TV services, that require adaptations to individual receiver characteristics, for example for devices with different supported codecs, or different screen sizes. Furthermore, especially in multi-user scenarios, endpoint oriented communication has shortcomings. Although existing multicast protocols provide network side support to reach more than one user with a specific service, its capabilities are rather limited. It is required that each user consumes exactly the same content at the same time. Asynchronous delivery of data packets as well as adaptation to the end device is not provided. The inclusion of caching and transcoding nodes into the multicast tree would allow a personalized delivery of content adjusted to the needs of users. It would make sense to distribute adaptation and general multimedia processing functionality through the network, rather than having it concentrated at the endpoints. This network-based support would remove the burden of adaptation and processing from the endpoints, and also add value to the network by making it a component of the service delivery and processing chain that provides more than pure transport. In some rare cases such network-based processing nodes exist today for special services in mobile networks. Examples are MMS relay/servers for multimedia messaging, or media gateways for conversational services. However, such nodes are always very specialized, and cannot be explicitly invoked or controlled based on endpoint requirements.

It is desirable to have a general, and ideally standardised, concept and architecture that allows the easy and scalable inclusion of network-based processing nodes into media delivery systems. This would allow us to move from the "dumb network" promoted by the end-to-end paradigm to a "smarter network" paradigm. A proposal for such an architecture, which builds on the concept of overlay networking, is described below.

### B. *Managing network support with Service-Specific Overlay Networks*

This work introduces the concept of *Service-Specific Overlay Networks (SSONs)* [4] to manage and control the delivery of media data in future networks. A SSON is defined by the set of overlay nodes and overlay links that constitute the ‘virtual network’ topology.

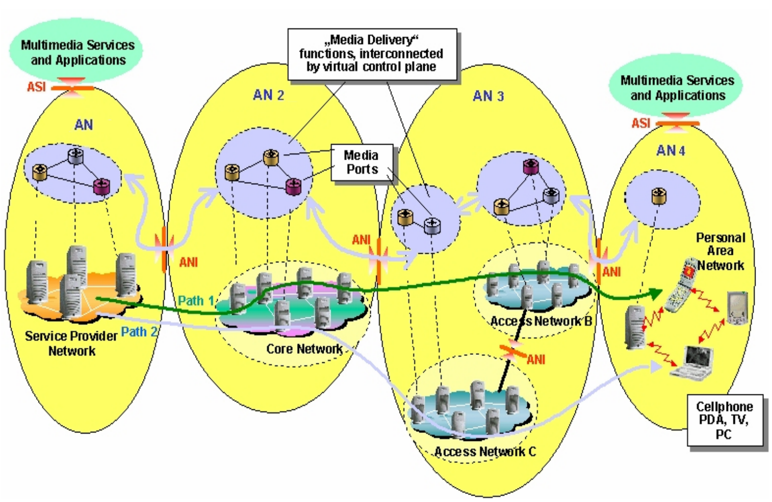


Figure 2: Visualization of the concept of Service-Specific Overlay Networks including processing nodes

A key motivation of the SSON concept is its ability to customize the virtual network topology – the addressing as well as the routing at the overlay level – according to the specific needs of a particular multimedia service. For example, the routing at the overlay network level can be fully tailored towards the needs of the service for which the SSON was established, which allows the system to indirectly influence the routing of the media in the underlying network. This enables content dependent routing.

A vital feature of SSONs is their ability to transparently include network-based ‘media processors’, called MediaPorts, into the end-to-end communication path from the server hosts, called MediaServer, to the client devices, called MediaClients. Details on the inclusion of MediaPorts can be found in the next section.

The SSONs (overlay networks) can be tailored to the specific service it has been established for and thus include, besides the MediaClients and MediaServers, the MediaPorts that are required to provide the requested service. The most suitable media processing function can be selected based on appropriate metrics considering the desired capabilities (e.g. transcoding between different media formats), costs of using the processing function, transport costs to route data to and from the MediaPort, and the current processing load of the hosts providing this functionality

SSONs are managed throughout their lifecycle, from the initial setup until the final removal. This includes also the reorganization and adaptation of the SSON, meaning reconfiguration of the overlay topology and re-computation of the routes in order to optimize the network when changes regarding network conditions or user mobility or user and service context become apparent. The SSON-controlled routing itself is done on the overlay level (and routing between overlay nodes on IP layer, which is however left untouched). Indeed 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). Even temporal disconnection of endpoints can be handled, for example by inclusion of caching nodes that temporarily store service data (like audiovisual streams) for an end user. Adaptation of SSONs can happen on different timescales. “Fast” adaptation is required to respond critical changes that would otherwise disrupt a service, such as changes to the underlying network (e.g., if a link is congested or has failed). “Slow” adaptation can optimize operation in response to non-critical changes, e.g., when MediaPorts join or leave or when users enter or leave a SSON. For example, a SSON may adapt after the network handoff of a mobile node in order to optimize the virtual network topology and the routing at the overlay level.

### *C. Network-based processing nodes*

In the previous section, it has been explained that SSON are set up to enable media processing functions, which could include adaptation, caching, stream splitting, or/and synchronization. In a SSON, those media processing functions are performed by MediaPorts (MP), hosted by ONodes (overlay nodes). Once the SSON is established with the required MPs, the media data is routed along the overlay network through the MPs towards the MediaClient(s).

As noted previously, several MPs may run on a single ONode. In the same way, a single ONode may be part of several SSONs. Therefore it is essential to be able to instantiate the right MP within the most optimal ONode for the intended SSON. For this to be possible, a search and/or directory function is needed in order to be able to locate available MediaPorts based on a given service specification. The remainder of this sub-section focuses on the MediaPort directory and search services that form an integral part of the architecture presented in this paper.

The lookup of media processing functions located in the network is performed by a directory function called MediaPort Information (MPI). Two possible means of implementing this function are with a database-like directory service as in [5], or with an ad-hoc search as presented in [3] . This paper focuses on the MediaPort Directory Service (MPDS) based on a database approach. Due to the fact that a centralised architecture has obvious scalability limitations, the choice has been made to design the MPDS as a distributed database where each ONode hosts a part of the database service. Two levels of MediaPort description have been identified.

The higher level refers to the general properties of the MediaPort, such as which kind of service it can provide, e.g. caching or adaptation. In the latter case additional information about supported codecs has to be given.

It is also relevant for routing decisions to have information about the current available capabilities of the MP as present in the lower level of description. The current processor load of a transcoding devices or the remaining memory capacity of a cache is an example for that.

When up and running, each ONode registers the availability of the MPs that it hosts in the MPDS. After this registration, the information is updated if the status of any indexed MPs changes.

Therefore 6 different procedures are specified. To further extend the MPDS, new nodes can be added employing the **join** function. This may generate to redistribute the directory entries. The complementary procedure is **leave** which requires a backup of service entries in case no information should be lost. The **register** message is needed to give each MP the opportunity to advertise its capabilities in the MPDS. Its counterpart, the **deregister** function is used to release entries in the MPDS. To keep information in the database always up-to-date the MP can inform the node(s) where it is registered by an **update**. In order to inquire the MPDS for appropriate MPs the **search** function can be used. The input is a MP description. The return value is a set of transport addresses belonging to ONodes hosting a matching MP. Optionally the MPDS is able to perform pre-filtering to downsize the output of a search, based on constraints provided with a search request.

### *D. Using network support: The Ambient Service Interface*

The previous sections have outlined the general concept and how setup and management of SSONs, and inclusion of network-based processing nodes, are enabled. However, the detailed mechanisms and processes are usually hidden from the endpoints, i.e. from the service providers and end users. They should rather have the possibility to specify their needs, requirements and properties, and in response get the required functionality and characteristics provided from the network.

This is enabled through the exposition of a service provider interface, the Ambient Service Interface (ASI). The ASI allows service providers to:

- Specify requirements of a service, e.g. in terms of connectivity and QoS and processing/support functionality
- Request and negotiate capabilities from the Ambient Network, in correspondence to the requirements
- Specify and modify properties of the endpoints, both on server and client side
- Add and remove endpoints
- Start and terminate services for which a SSON has been negotiated and established before
- Add processing nodes to the network (this enables the business role of a "MediaPort Provider", i.e. a service provider that can offer and deploy processing nodes. For example, there could be a transcoding server provider, who advertises and registers his services into networks)

Thus, the ASI represents a high-level interface that allows service providers to use the functionality provided by AN in a straightforward and simple way.

## IV. Proof of Concept: Ambient Media Delivery Demonstrator

Media delivery using overlay networks and network-based media processing nodes have partly been implemented into a demonstrator. For the demonstrator, a Video-on-Demand (VoD) service was chosen. Figure 3 shows the used scenario as well as the chosen experimental setup. For the practical demonstrator setup, a DHCP server is running on each node representing the ACS of the Ambient Network, and assigning IP addresses to the registering nodes. Furthermore, a GUI is implemented and executed at the ACS nodes showing the internal status of the corresponding Control Space. A VideoLAN Client (vlc) is running at each MediaClient. Furthermore, a streaming server (also vlc) runs at the MediaServer node.

. The demo scenario is as follows: A service provider, a network operator and the end user are the parties involved in this scenario. In the experimental setup, a MediaServer capable of streaming the media content is part of the service provider's Ambient Network. The media delivery function of the network operator's Ambient Networks will take care of adapting the transport service when changes in the service requirements or users' needs become apparent. Two MediaPorts are included in the operator's network, one MediaPort is capable of transcoding a video stream (i.e. reducing the data rate of a video stream), whereas the other provides advanced flow routing capabilities. The user's network consists of two different end devices, a fixed MediaClient with a large screen (e.g., a TV or PC) and a mobile MediaClient like a PDA for mobile media access.

For executing the VoD service, a SSON tailored to the particular requirements of the service is established (see Figure 3). Dependent of the current context of the user, the media flows are either routed through the media transcoder or not.

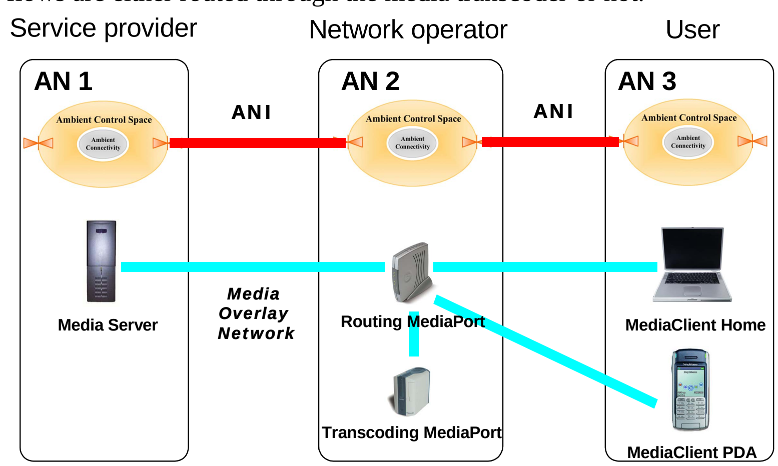


Figure 3 Ambient Media Delivery Demonstrator Setup.

Dependent on the physical distance between the fixed and mobile MediaClients, either one of the device is used for displaying the video stream. In the case that both devices are close together the device with the larger screen (e.g., the TV screen) is chosen for displaying the media stream. This implies that an ongoing streaming session on the PDA will be transferred to the TV screen if the PDA comes into the range of the device with the larger screen. Similarly, if the user moves away from the TV screen with the PDA, the media stream is transferred from the TV screen to the mobile device. For the proximity detection, our experimental system uses the signal strength of the wireless link between these two devices. The outcome of the proximity detection is the SSON adaptation.. At the beginning, the transcoding MediaPort is not included in the SSON and in the media routing path. When the proximity detection signals that the PDA is leaving the range of the TV screen, the SSON is adapted by inclusion of the transcoder into the overlay network, and the media routing path is changed for the respective flow. The media flow is first delivered to the transcoder, where the encoding is adapted to the specific display and processing capabilities as well as the access technology of the PDA.

## V. Summary & future Work

In this paper, we have developed the concept of Service Specific Overlay Networks. SSONs are overlay networks that are created for every media service and tailored to the specific requirements and preferences of the service, and its users respectively. The SSON routing logic is configured in a way that routing on the overlay-level is optimal for the particular service that is being delivered, allowing the dynamic reconfiguration of the overlay in response to changes in any aspect of the involved actors (user, network, service). The main benefit of the implementation of these "virtual networks" is the their network-side media processing capabilities, that is, the transparent inclusion of media processing functionalities, such as media adaptation or caching, into the end-to-end media delivery path. To that end, we introduce the Media Port Directory Service function as a distributed repository for the lookup of the required network-side processing functionalities to be included within a SSON. We have also introduced the Ambient Service Interface (ASI). This interface allows the end points (end users, and multimedia content services) to specify its requirements and preferences to the system without any knowledge of SSONs. Finally, we have presented a demonstrator showing media delivery using overlay networks, tailored for a Video-on-Demand service. Both fast and slow SSON adaptation scenarios are demonstrated and a network-based media processing node adapts the media content to the characteristics of the different end user devices.

Amongst others, the future work that will be studied in the next months will extend the concept presented so far for enabling more services to benefit from such an architecture, and introduce overlay measurements, aimed to increase to Quality of Service and to introduce charging for the use of SSON.

## Acknowledgment

This work has been supported by Ambient Networks, IST-2002-507134, a research project supported by the European Commission under its Sixth Framework Program. The authors like to thank all the partners of Ambient Network Work Package 5, who actively contributed to this work.

## References

[1] Norbert Niebert, Andreas Schieder, Henrik Abramowicz, Göran Malmgren, Joachim Sachs, Uwe Horn, Christian Prehofer and Holger Karl, "Ambient Networks – An Architecture for Communication Networks Beyond 3G", IEEE Wireless Communications, Vol. 11, No. 2, April 2004, pp. 14-21.

[2] Bengt Ahlgren, Lars Eggert, Börje Ohlman and Andreas Schieder, "Ambient Networks: Bridging Heterogeneous Network Domains", Proc. 16th IEEE Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2005), Berlin, Germany, September 11-15, 2005.

[3] Eskindir Asmare, Stefan Schmid and Marcus Brunner, "Setup and Maintenance of Overlay Networks for Multimedia Services in Mobile Environments", Proc. 8th International Conference on Management of Multimedia Networks and Services (MMNS 2005), Barcelona, Spain, October 24-26, 2005, pp. 82-95.

[4] José Rey, Bertrand Mathieu, David Lozano, Stephen Herborn, Kamal Ahmed, Stefan Schmid, Stephan Goebbels, Frank Hartung and Markus Kampmann, "Media Aware Overlay Routing in Ambient Networks", Proc. 16th IEEE Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2005), Berlin, Germany, September 11-15, 2005.

[5] Bertrand Mathieu, Michael Kleis, Meng Song, " A P2P Approach for the Selection of Media Processing Modules for Service Specific Overlay Networks", International Conference on Internet and Web Applications and Services, Workshop on P2P Systems and Applications (ICIW P2PSA 2006), February 23-25, 2006.

[6] Jacob Chakareski, Bernd Girod, "Rate-distortion Optimized Packet Scheduling and Routing for Media Streaming with Path Diversity", Proceedings Data Compression Conference (DCC) 2003, pp. 203-212.

[7] A. Warabino, et al., "Video transcoding proxy for 3G wireless mobile Internet access." IEEE Communications Magazine, Vol 38, Issue 10, pp 66-71, Oct 2000.

[8] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, Robert Morris, "Resilient Overlay Networks", Proc. of 18th ACM SOSP, Banff, Canada, October 2001.

[9] Lakshminarayanan Subramanian, Ion Stoica, Hari Balakrishnan, and Randy Katz, "OverQoS: An Overlay Based Architecture for Enhancing Internet QoS", Proc. of 1st NSDI, San Francisco, CA, March 2004.

[10] Zhi Li and Prasant Mohapatra , "QRON: QoS-aware Routing in Overlay Networks", IEEE Journal on Selected Areas in Communications., special issue on Service Overlay Networks, 2003.

[11] D. Xu and K. Nahrstedt, "Finding Service Paths in a Media Service Proxy Network", Proc. of SPIE/ACM Multimedia Computing and Networking Conference, San Jose, CA, January 2002.

[12] Xiaohui Gu, Klara Nahrstedt, Rong N. Chang, Christopher Ward, " QoS-Assured Service Composition in Managed Service Overlay Networks", Proc. of 23rd IEEE International Conference on Distributed Computing Systems (ICDCS 2003), , May 19-22, 2003.

[13] Gu, Xiaohui, and Klara Nahrstedt, "A Quality-Aware, Composable Service Overlay Network", Internal report, Department of Computer Science Report No. 2342, University of Illinois at Urbana-Champaign, May 2003.

[14] IETF RFC 3303, "Middlebox communication architecture and framework", August 2002

[15] J. Touch et al., "A Global X-Bone for Network Experiments", IEEE Tridentcom 2005, pp 194-203