SMART: A Media Service Delivery Architecture for Ambient Networks

José Rey, David Lozano, Stephen Herborn, Kamal Ahmed, Stefan Schmid, Frank Hartung, Markus Kampmann and Michael Kleis

Abstract — **In this paper we introduce a new architecture for the provision of optimized Multimedia Delivery Services based on feedback and context information from the network and the user respectively. Different communication use cases are supported, ranging from point-to-point and peer-to-peer through to mass distribution broadcast/multicast communication. The proposed Smart MultimediA Routing and Transport (SMART) architecture, which is based on a novel overlay network solution, enables the transparent integration of advanced media routing, adaptation and caching mechanisms along the end-to-end media delivery path. A key characteristic of the SMART architecture is the clear separation of user and control plane functionalities. Ambient Networks (ANs) enable the inter-networking of heterogeneous networks for the provision of global connectivity and user-centric services. The concepts presented in this paper represent early results of the Ambient Networks Integrated Project in the area of multimedia routing and adaptation in ANs. As the project continues, the interfaces to AAA, QoS et cetera will be defined in more detail, implemented and finally put forward for standardization.**

Index Terms—**Overlay networks, Internetworking, Media Routing, Media Caching, Media Adaptation, Media Service Delivery, and Media Flows.**

I. INTRODUCTION

NOWADAYS packet-switched networks are everywhere. In our daily lives we use networked technology for work, education, innovation and also for fun. Networks are meant to *serve* users, that is, to provide *services*. Typically, the more connections a network provides, i.e. the better it integrates with other networks and delivers services, the more useful it is to users. A good example of this is the Internet: originally thought as a means for communication between institutions for government, research and educational purposes [1], it has become a continuously growing global network infrastructure and an indispensable daily tool for a significant proportion of the world's population.

Until now, the networking technologies used in the Internet, like link layer and packet transmission technologies in both wired and wireless networks have attained a high degree of maturity (although newer technologies have not ceased to appear). There are well-known protocol suites for the transport of any kind of media and a substantial collection of available optimizations for any kind of network topology, configuration, or properties, even including non-transparent nodes like NATs and firewalls. However, as past work has shown, there is much more to service provision than simply having to cope with link packet loss rate, delay or duplicate acknowledgments [2][3] or NATs and firewalls.

Clark [4] and others [5] point out that the Internet is no longer only a playground for researchers but is now also an important business platform for communications and all kinds of transactions. This requires, besides data transport, also novel mechanisms that provide policing, integrity, QoS, security, resource usage optimization, mobility, and scalability.

The challenge to deliver pervasive services is even more complicated when different networks with different technologies are interconnected. Thus, there is a need for an integrated solution for service provisioning that enables *mobility*, *cooperation* between heterogeneous networks and supports *flexible business models*, as the three most outstanding important requirements.

In order to cope with these distinct problem spaces, it seems natural to draw a line between those activities located in the *control plane*, intended for the supervision, coordination, and establishment of the communication and the *user plane*, which comprises those tasks related to the packet handling and packet transport from service provider to the user. The *control plane* thus represents the intelligence of the network and it is physically located *within* the network at selected nodes. This facilitates a user-centric architecture that takes care of user's requests and preferences, insulating the users (and network administrators as well) from tedious protocol details.

Manuscript received February 5th, 2005. This is a byproduct of the Ambient Networks project, http://www.ambient-networks.org, partially funded by the European Commission.

J. Rey is with the Panasonic R&D Center Germany GmbH, Langen, Germany (phone: +49-6103-766-134; fax: +49-6103-1188; e-mail: rey@panasonic.de).

D. Lozano is with Telefonica R&D, Boecillo (Valladolid), Spain (email: dll231@tid.es)

S. Herborn is with National ICT Australia and the University of New South Wales, Sydney, Australia (email: stephen.herborn@nicta.com.au).

K. Ahmed is with TNO Information and Communication Technology, Delft, The Netherlands (email: o.k.ahmed@telecom.tno.nl).

S. Schmid is with NEC Network Laboratories, Heidelberg, Germany (email: schmid@netlab.nec.de).

F. Hartung and M. Kampmann are with Ericsson Research, Aachen, Germany (email: {Frank.Hartung|Markus.Kampmann}@ericsson.com).

M. Kleis is with Fraunhofer Institut FOKUS, Berlin, Germany (email: kleis@fokus.fraunhofer.de).

Self-management [6] is an operative term here.

The European Ambient Networks Project (AN) envisions an architecture for Next Generation Networks that supports the ideas outlined above. Such architecture follows a set of high-level basic principles laid out for the AN project, i.e. an architecture based on 1) providing *open interfaces* and *open connectivity* 2) providing a framework for *network composition* and 3) enabling *mobility* of users [7].

This architecture incorporates the *Ambient Control Space (ACS)*, which abstracts control functions of existing networks to a common set of functionalities or Functional Areas (FAs). Each Functional Area is responsible for a certain value-adding functionality of an AN, like QoS, Mobility, Security, Policy or Network Management and, the object of this paper, Media Service Delivery. In accordance with this separation, our work focuses on the control plane intelligence that is needed to provide advanced Media Services to the user.

In the following sections, the Smart MediA Routing and Transport (SMART) architecture for the provision of optimized Multimedia Delivery Services in Ambient Networks is presented. Section II presents the *SMART Concepts*, Section III outlines the *SMART Architecture* and Section IV presents our conclusions and further work.

II. SMART CONCEPTS

A. *SMART Role Model*

SMART, *Smart MediA Routing and Transport*, is an overlay-based architecture for the delivery of Media[1] Services between service or content providers[2] and service consumers (users). This is depicted in the *role model* presented in Figure 1.

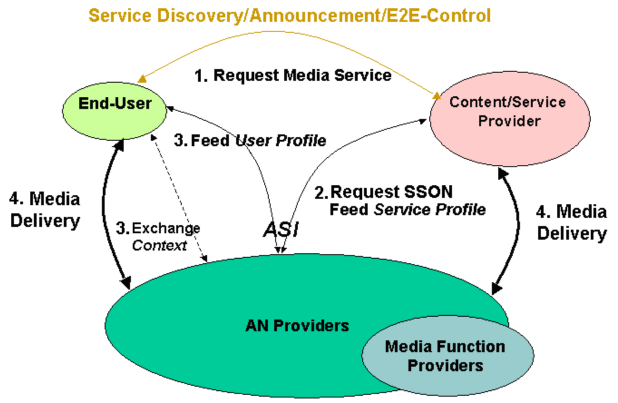

Figure 1. Role model in Ambient Networks.

However, not all the functionalities represented in Figure 1 are contained in SMART; only the *AN Providers* oval, the *Media Function Providers* oval and the arrows 2, 3 and 4 are part of the ASI (Ambient Networks Service Interface) interface specified in SMART.

In detail: (1) The user requests a service from a content or service provider. (2) Then, the content provider makes a request over the *Ambient Service Interface (ASI)* to the AN provider for a *Service Specific Overlay Network* (*SSON*). Thereby, a *Service Profile* is exchanged, which captures both, policies and the service contract (e.g. *Service Level Agreement*), between content provider and AN provider. (3) After a positive exchange, the AN overlay provider acquires information about the profile and context of the user. This is necessary in order to optimally satisfy user preferences and requirements. (4) Once this initial bootstrapping is done, the actual data is transported to and from the user using the established SSON. In this sense, the AN provider is delivering a *service* to the content provider and the user, thus acting as a mediator. The AN provider (where needed) selects a set of *intelligent nodes* that will perform the needed caching, adaptation and media routing (overlay-level routing) on the user data.

At the end, the user has effectively only interacted with the service provider and the complexity of the described process is hidden from him, namely behind the AN overlay. Note that from Figure 1 the actual description, announcement, discovery and the language used to request a service (uppermost arrow and title above it) are out of the scope of Ambient Networks. There are other proposals for dealing with these tasks [8][9].

B. *SSONs: Definition and Motivation*

A SSON is an *overlay network* and can be defined by the set of overlay nodes and overlay links that constitute a *virtual network*. SSONs are formed by the inclusion of the appropriate network entities (*intelligent nodes*) that interact with end-nodes for the service. These network entities, or *Overlay Nodes (ONodes),* represent one of the basic components of the SMART architecture. ONodes enable the provisioning of media processing capabilities, such as caching, adaptation, flow splitting, synchronization and smart routing, inside the network. A single ONode can provide several media functions at the same time and, also may belong to many SSONs at the same time.

We have chosen to employ overlay networks because they fit well to the key requirements that motivate the work on SMART:

1. The development of a transport architecture for *Media Delivery* that supports pervasive and massively scalable services spanning heterogeneous networks.
2. Compatibility with varied communication use cases including unicast, multicast, peer-to-peer and content delivery networks. Although the smallest overlay can be established as a point-to-point service, current media architectures [10][11] do not support adaptive network-based processing of media. Past work on media delivery has shown that the provisioning of adaptive, network-based multimedia services on top of a

---

[1] Hereby, *media* is a general term covering all kinds of information consumed by users and not just limited to traditional real-time applications. E.g. SMS messages or cached data is also *media*.

[2] Note that a single user could also be a *service provider*, e.g., in a video-telephone call.

heterogeneous network infrastructure is best accommodated using the overlay approach [12][13][14][15].

3. The provision of *User-centric Media Services*: services tailored to the user's preferences and to the capabilities of both the users and the network. For this purpose SMART builds overlays using ONodes that implement the *Media Routing*.

C. Media Routing and Transport: what they are and where they are located

Overlays build a *virtual topology* upon a specific network technology (hereby the components of a network technology are typically layer 3 boxes, such as IP routers). The term *virtual topology* denotes that the nodes at the overlay level do not necessarily map to a node in the level immediately below. Figure 2 illustrates this fact: the links between the nodes and links at the SSON level do not correspond to the nodes and links at the network level (see bottom cloud in Figure 2).

This separation allows for a cleaner distribution of the problem spaces: on the one side, the underlying network provides the necessary connectivity and QoS for the flows between the routers using available QoS technologies such as DiffServ [16] or IntServ [17], multi-path routing [18] and diverse traffic engineering practices [19]. Note that the routing at the network level employs *traditional* media-unaware routing, such as OSPF [20] or RIP [21]. Main reasons for this trade-off are simplicity and processing costs, which have proved to be good design choices. On the other side, at the overlay level, the ONodes contain the necessary *state* to associate individual flows to a specific SSON (or one of its Media Flows). Thus, the routing at the overlay level adds the media-awareness by instructing the Media Flows to be re-routed, undergo necessary processing and/or pass through certain ONodes in order to meet the service requirements, as needed. This *layered routing* is in line with recent architecture proposals that speak up for introduction of a further level of name resolution, namely through the use of Service Identity (SID) [22][23] or stacked identifiers [24].

ONodes are thus virtually connected by instantiating *Media Routing state* at them. This Media Routing state consists mainly of three components:

1. *Media Routing* tables: the next Overlay Node or the set of Overlay Nodes (*Media path*) to be followed.
2. *Media Caching or Adaptation* required: the set of operations to be performed on the media, i.e., whether the media needs to be cached or synchronized, or the user has changed its context and a new media format is needed.
3. Finally, these pieces of *state* also cover the actual requirements on the underlying network, e.g., the QoS, and priority of Media Flows to be provided. These requirements are expressed in a technology-independent way and translated at the ONodes by the *OSL* as explained further below.

III. SMART ARCHITECTURE

A key concept of the SMART architecture is the concept of Service-Specific Overlay Networks, or SSONs, as mentioned above. The novel aspect of this concept is that a different *virtual network* is deployed for every Media Delivery Service (or group of services, i.e. aggregation). This allows the configuration of appropriate overlay-level routing paths that meet the exact service requirements, e.g. for example, QoS, media formats, responsiveness, cost, resilience or security. Moreover, this enables the transparent integration of network-side media processing capabilities (such as caching, adaptation and synchronisation) into the selected end-to-end delivery paths. Additional fine-grained adaptation and flexibility is provided since these actions can be performed separately for each flow of the media service within the SSON (i.e., Media Flows*)*.

Some other IST projects, such as MAMBO, MOBY DICK, WINE GLASS, etc, are also working on service architectures for advanced multimedia delivery services, mainly considering QoS-related constraints and the transparent adaptation of the communication. However, these approaches tackle the problem by directly controlling and monitoring the traffic at IP level and, therefore, achieve technology-dependent solutions. I.e., networks not using IP as network substrate cannot be served by such architectures, since the application needs mapped to the IP layer are lost when using another technology. Hence, the use of overlays, dynamically tailored for each services, introduces another level of abstraction that enables, together with the rest of the AN system, the provision of the following main benefits: 1) a generic solution which is independent from the details of the underlying network and, hence, more capable to be deployed in a heterogeneous environment and 2) flexibility for the definition of routing constraints (or rules) in the virtual network, resulting in a generic and extensible framework for context-aware routing of media content.

In order to proceed with the description of the SMART architecture in a structured way, it is worth considering the functionalities that need to be deployed within the system. The first fundamental requirement is intelligent selection of ONodes to be included in the SSON, configuration of state at these entities, including the possible adaptations of the media at these nodes and of the overlay network topology. These actions are based on the specific needs of the service and the user (user context and preferences). All of these functionalities are controlling actions and, therefore, can be conceptually grouped into the *control plane*. The second fundamental requirement in SMART is a set of functionalities for transport and handling of user-data that must be present in the SSON to actually carry on the caching, adaptation, synchronization, etcetera, as instructed by the Media Routing. This last set of functionalities can be referred to as belonging to the *user plane*.

Both parts of the SMART architecture are reflected in Figure 2 and Figure 3 and are further detailed in the following subsections.

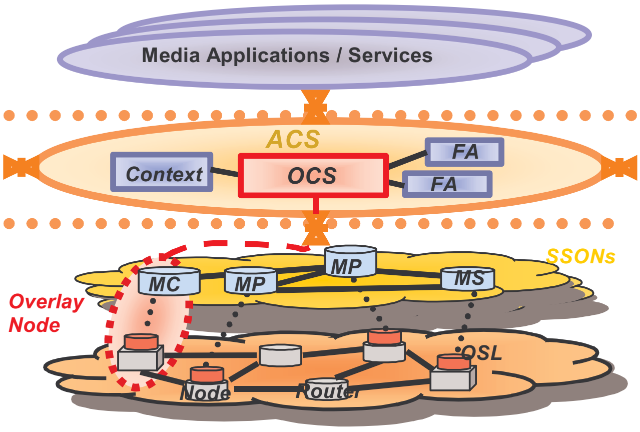

Figure 2. SMART Architecture: a possible representation.

A. *Control plane*

The main control entity in the SMART architecture is the *OCS (Overlay Control Space)*, which in turn constitutes the Media Delivery Functional Area of the AN *Ambient Control Space (ACS)* [23]. It can be seen as an abstract entity that is responsible for all the decisions regarding SSONs. Upon requests from users of the AN (e.g. service providers, customers, etc.), the OCS manages the creation of SSONs through the inclusion of suitable ONodes (Media Ports, MediaClients and MediaServers) and the reservation of the appropriate media processing capabilities for the media service. Additionally, it controls the re-organisation/adaptation of existing SSONs based on changes in the underlying ANs (connectivity, composition, QoS, mobility, media processing functions, etc), changes in the context of a user (e.g., location, user profiles, device capabilities) or changes in the service policy/context. This re-configuration or adaptation may imply on-the-fly re-structuring of the SSON in order to optimise the network according to the current network, user and service context.

Considering the similarities between SSONs and VPNs, the management performed by the OCS is mainly related to the control of the overlay addressing and overlay routing mechanisms used. For this reason, all the decisions taken by the OCS are reflected in the configuration of appropriate overlay routing tables in the user plane that establish which ONodes, or Media Ports (MPs), are to be traversed by each media flow of the session and which media processing capabilities are to be applied for each flow. Thus, the OCS needs to know the set of available MediaPorts (MPs) and its capabilities prior to the SSON setup or re-configuration. To do so, the OCS makes use of a *MediaPort Directory Service (MPDS),* which maintains information regarding the location, processor load, cost, etc. of the available MediaPorts. Other mechanisms such as dynamic resource discovery may be used as an alternative to the MPDS.

Since the SMART environment may be subject to some level of change, a distributed implementation of the OCS is preferable. Thus each ONode implements an atomic component of the OCS, the *ONode Control* (see Figure 3). This ONode Control is responsible for the signalling exchange with the rest of the OCS (through the use of internal interfaces) and the management of the user plane part of the ONode. Although the OCS is composed by the aggregation of all the ONode Controls in the AN, more capable ONode Controls (masters) may assume more responsibilities than others (slaves) and, thus, the decision taking can be freely implemented, i.e. in a distributed, semi-distributed or centralized fashion.

Finally, for the communication with other entities and the underlying network, the OCS uses several interfaces (see Figure 3) that can be grouped into three general types:

- Internal to SMART: The *OCI (Overlay Control Interface)* is used for the communication between control and user plane (configuration, charging, etc) and for the triggering of the reservation of actual physical resources in the underlying network (see OSL below). This interface is a subset of the AN *ARI (Ambient Resource Interface)* [23]
- Internal to the ACS: The OCS is directly connected to other FAs of the ACS by signalling interfaces. This *connection* is used to collect context, network information and possible triggers of interest for the routing logic, in relation to adaptation, monitoring and management of the overlay. This information includes, for example, underlying network characteristics, mobility triggers, user context and security information. In the case of SSONs that span more than one AN, this type of communication may be performed through the *Ambient Networks Interface (ANI)* [23].
- Towards Applications external to the AN: The OCS exchanges signalling with the external Applications through the *AN ASI (Ambient Service Interface)*, as discussed previously in the role model.

B. *User plane*

Within the user plane, the following entities are defined:

1) *Overlay Support Layer (OSL)*: The OSL embodies the basic overlay network functionality required in every ONode for the transport and handling of user data. The OSL resides on top of the underlying IP network and establishes overlay or virtual links to other OSL entities using Overlay Node addresses. The OSL also communicates and exchanges data internally with the application modules in charge of performing media processing operations within an ONode. The OSL is responsible only for packet transport and handling on the overlay, e.g. using tunnelling techniques by introducing another level of encapsulation with overlay headers. However this is not the only solution, as suggested by Balakrishnan et al [22] all these identifiers need not be present in actual headers as long as a relevant mapping exists at the *control plane*.

The forwarding behaviour is controlled by SSON-specific Media Routing tables, which are tailored and configured by the OCS.

These Media Routing tables are used by the OSL to map Media Flows to flows in the underlying network. E.g., in case of an ONode included in several SSONs at the same time, the OSL is responsible for the demultiplexing of incoming packets to the correct SSON, based on a specific SSON Identity tag (SSON-ID) included explicitly in the overlay header or, as mentioned above, implicitly present in a mapping of a set of packet headers to a certain SSON.

2) *MCs, MSs and MPs*: on top of the OSL, and assisting in the process of Media Routing, are application modules that implement the behaviour of a Media Client, Media Server or Media Port. MPs may perform the caching, adapting or synchronizing functions that will be applied to the Media Flows in accordance with user and service provider requirements. MCs act as data sinks and forward the multimedia data to the end-point applications through the *ASI-u (Ambient Service Interface – user plane)*; whereas MSs act as data sources and receive the multimedia data from the end-point applications through the ASI-u interface.

Inside a SSON, the role associated to a certain ONode simply depends on the specific modules (MC, MS and/or MP) activated by each SSON of which it is a member and the Media Routing state present at the node. For that reason, the same ONode can easily take on different roles in a SSON through the activation of different modules. This ability provides the required flexibility to dynamically adapt the communication in a per-Media Flow basis, for example, a MC may become a MP for a specific Media Flow in order to adapt its content before its delivery to a new MC. Similarly, in a peer to peer context, ONodes may simultaneously act as MS and MC within a SSON.

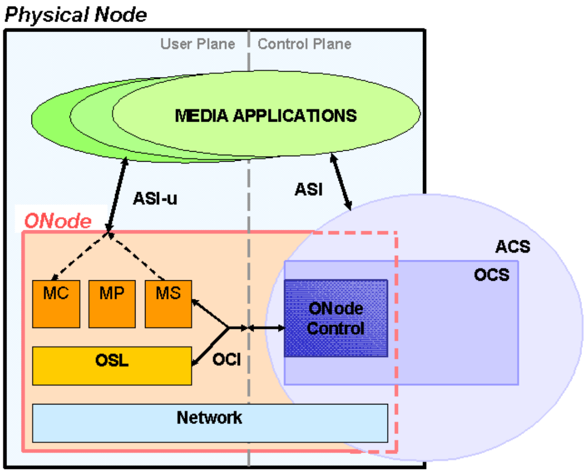

Figure 3. AN Node Architecture.

IV. CONCLUSION AND OUTLOOK

In this paper, we have presented a high-level design of SMART, a framework for the provision of advanced media delivery services within the context of Ambient Networks. In order to provide value added functionality and services to mobile users with different devices and spanning multiple heterogeneous networks, an overlay networking approach is adopted. To this end, we introduce the concept of Service Specific Overlay Networks. SSONs are overlay networks that are configured on a per-service basis and tailored to the specific requirements and preferences of the service and its users respectively. The SSON-level routing is configured in a way that routing on the overlay-level is optimal for the particular service that is being delivered. SSONs also allow the transparent inclusion of media processing functions such as media adaptation or caching into the end-to-end media delivery path. Further, every SSON can support a different service provision paradigm (e.g. multi-user or peer-to-peer). Moreover, SSONs provide also the basis for customized content provision and billing.

The realisation of the SSON concept requires careful consideration of the required components, functions and interfaces. Therefore, within our design we group the fundamental requirements and subsequent functionalities of the SSON approach into two distinct categories, namely the *control plane* and the *user plane*. This enables a modular design methodology. The control plane encompasses functionalities concerning the setup and administration of SSONs, and the user plane encompasses mechanisms for services requests, packet transport, and interaction with end-point applications. Control and user plane entities interact with each other as well as internal and external entities through a set of defined interfaces, namely the Overlay Control Interface, the Ambient Network Interface, and the Ambient Service Interface.

Finally, we outline the respective roles of the physical entities responsible for the management and functionality of the SMART architecture. These include both distributed entities such as the Overlay Control Space, and individual entities such as Media Servers, Media Clients, and Media Ports. These entities communicate via the Overlay Service Layer (OSL), a high-level communication stack layer that is implemented in all participating Overlay Nodes (ONodes).

In summary, we can say that the key novelty of SMART is that it provides a tool for content and service providers to make their content (or service) available to users in automated way using the available network resources in the most appropriate form. This automation allows content and service providers to delegate the details of the service delivery to the SMART transport. Content and service providers communicate with SMART through a fully defined interface: the ASI. In order to accomplish these goals, SMART makes a clear split between user and control plane functionality as reflected in the internal ONode structure. Moreover, this functionality split enables the easy definition of interfaces to other network components, e.g., network management or QoS, thus guaranteeing extensibility and interoperability.

As part of future work on the SMART architecture, we focus in particular on the evaluation of the performance and scalability properties of our design. This future work will provide some early estimation of the cost and performance of the system and determine how the cost and complexity affect introduction of our media service architecture in future systems. This work will be based on our existing SMART prototype implementation, which we have used to demonstrate the basic functionality for advanced media delivery across composed Ambient Networks.

ACKNOWLEDGMENT

The authors like to specially thank all the partners of Ambient Network Work Package 5, who actively contributed to this work.

REFERENCES

[1] M. W. Murhammer, et al., "TCP/IP Tutorial and Technical Overview", Prentice Hall, 1999.

[2] P. Karn, et al., "Advice for Internet Subnetwork Designers", RFC 3819, July 2004.

[3] J. Nagle, "Congestion control in IP/TCP internetworks", RFC 896, Feb. 1984.

[4] D. D. Clark, et al., "Tussle in Cyberspace: Defining's Tomorrow's Internet", SIGCOMM'02, August 2002.

[5] M. S. Blumenthal, D. D. Clark, "Rethinking the design of the Internet: The end to end arguments vs. the brave new world", ACM Transactions on Internet Technology, Vol 1 No 1, August 2001, pp 70-109.

[6] K. Herrmann, G. Mühl, and K. Geihs, "Self-Management: The Solution to Complexity or Just Another Problem?" *IEEE Distributed Systems Online, vol. 6, no. 1, 2005*.

[7] Niebert, et al., "Ambient Networks: An Architecture for Communication Beyond 3G", IEEE Wireless Communications, pp 14-22, April 2004.

[8] E. Chistensen, et al., " Web Services Description Language (WSDL) 1.1", http://www.w3.org/TR/wsdl.

[9] "Universal Description, Discovery and Integration", http://www.uddi.org.

[10] J. Rosenberg, et al., "Session Initiation Protocol", RFC 3261, June 2002.

[11] 3GPP TS 22.228 v7.0.0 "Service requirements for the Internet Protocol (IP) multimedia core network subsystem: Stage 1 (Release 7)".

[12] S. Barnejee, et al., "Scalable Application Layer Multicast", Technical Report CS-TR 4278, University of Maryland, July 2001.

[13] Gnutella: http://www.gnutella.com.

[14] S. Brin, L. Page, " The Anatomy of a Large-Scale Hypertextual Web Search Engine", Computer Science Dept., Stanford, California, USA.

[15] Akamai White Paper, "Best practices in Media Delivery", http://www.akamai.com.

[16] S. Blake, et al., "An Architecture for Differentiated Services", RFC 2475, December 1998.

[17] R. Braden, et al., "Integrated Services in the Internet Architecture: An Overview", RFC 1633, June 1994.

[18] A. C. Begen, et al.,"Real-Time Multiple Description and Layered Encoded Video Streaming with Optimal Diverse Routing", Proc. IEEE International Symposium on Computers and Communications (ISCC), Antalya, Turkey, June 2003.

[19] D. Adwuche, et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[20] J. Moy . "OSPF Version 2" IETF STD 54, April 1998.

[21] C. L. Hedrick, "Routing Information Protocol", RFC 1058, June 1998.

[22] H. Balakrishnan, et al., "A Layered Naming Architecture for the Internet", Proc. ACM SIGCOMM Conference, Aug. 30 - Sept. 3, 2004.

[23] Ambient Networks Work Package 1, Deliverable 1.8 "Ambient Networking: Concepts and Architecture", due to publication under of http://www.ambient-networks.org.

[24] I. Stoica, "Internet Indirection Infrastructure" Proc. ACM SIGCOMM Conference, pp 73-88, August 2002.G. O.

[25] Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.