Media Aware Overlay Routing in Ambient Networks

José Rey, Bertrand Mathieu, David Lozano, Stephen Herborn, Kamal Ahmed, Stefan Schmid, Stephan Goebbels, Frank Hartung and Markus Kampmann

***Abstract* — With the increasing heterogeneity of Internet enabled devices and Internet access technologies, media content may often need to be adapted to best meet the user or application needs. However, server-side adaptation has shown its limitations, leading to the development of network-side adaptation techniques. Overlay Networks have emerged as a possible solution to enable data processing between users and the content server, as it allows to transparently include media processing nodes into the end-to-end media path. Nevertheless, to deal with multimedia streams in an efficient way, current proposals lack application awareness when it comes to routing decisions. Typically overlay networks only consider IP header information when routing packets. In this paper we introduce a new concept that shall enable media aware routing via the use of overlay networks. The proposed routing approach considers service and application information to improve routing decisions based on the user/application requirements. In our proposal, service-specific overlay networks (SSONs) are created for every service, which allows customisation of the network resources. SSONs include overlay nodes with adaptation capability. The data routing in the SSON is performed using the media aware overlay routing logic. This paper presents the concept, along with a proposed architecture designed to provide a media aware routing service. The advantage of the proposed functionality is illustrated based on a mobile user scenario.**

***Index Terms*—Overlay networks, Content Adaptation, Network Processing, Adaptive Network Technologies, Media Awareness, Media Routing.**

I. Introduction

RECENT years have shown an increasing divergence in the capabilities of devices capable of internetworking, such as PDAs, mobile phones, and notebook computers. Such devices differ in terms of the content that they can handle; they are limited by characteristics such as display resolution, connection method and codec availability. Devices also differ in terms of their usage, for example notebooks are more likely to be static for a long period of time, whereas mobile phones can be expected to constantly change location. This divergence, in conjunction with the ongoing trend towards access network heterogeneity (e.g. Wi-Fi, UMTS, xDSL, Bluetooth), has created a situation where each individual media content consumer (i.e. a user) has a different set of content delivery requirements. As a result of this, *content adaptation* has become a hot topic in network and computer science research since quite some time.

Initial contributions to the content adaptation area have been based on a server-side approach: for instance the XML/XSL solution [1] enables web content to be adapted to user constraints (device capabilities and preferences). Another example is the stream switching capabilities of the 3GPP Packet-switched Streaming Services [12]. Advances in transcoding techniques have enabled video services to be delivered to different devices. However, server-side transcoding is an expensive option in terms of both processing cost and delivery delay, and requires an increased investment in server capacity (e.g. more processors, more memory and storage, clusters of servers), also increasing the administration cost. Furthermore, content providers like broadcasters (radio or TV channels) prefer to focus on the content itself and not on the format in which it is delivered, preferring to let a third-party entities supply any required adaptation. As a result of this, solutions have emerged that adapt content at some point in the network, on the end-to-end path between the user and the server.

Multimedia applications often use SIP (Session Initiation Protocol) [2] as a signalling protocol for the establishment of the communication sessions. However, the SIP architecture by itself does not provide content adaptation in the network. Furthermore, SIP allows only point-to-point communications, preventing broadcast communications. On top of that, mobility is not well managed by SIP and SIP components are not aware of the network properties such as congestion, latency, etc. In order to address these limitations, IMS (IP-based Multimedia Subsystem) [3] contains improvements to the SIP architecture, enabling broadcast communications and better support for mobility. Furthermore, components, named 'Application Servers', have been introduced in the architecture to carry out value-added services. However, these components are network operator specific and the IMS does not enable dedicated adaptation components to be easily deployed into a

Manuscript received March 18th, 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 (email: rey@panasonic.de).

B. Mathieu is with France Telecom, R&D Division, Lannion, France (email: bertrand2.mathieu@francetelecom.com)

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).

S. Goebbels is with Aachen University, Aachen, Germany (email:sgs@comnets.rwth-aachen.de).

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

legacy architecture.

Furthermore, IMS can not route individual media flows of the same session over different paths, which is an essential requirement if we consider that the individual media flows of a single session may have different QoS characteristics. For example, a media stream that consists of a video and an audio component may only require adaptation of the video stream. In such a case, the audio component may be routed directly to the end device in order to avoid placing unnecessary strain on adaptation infrastructure (see example in V). Flow-based routing is also useful given that many devices have more than one network interface. A PDA may have Wi-Fi, LAN, GPRS and Bluetooth connections available, thus media component flows may be routed to any of those interfaces depending on flow priority, link cost, and link status.

Moreover, research projects such as RON [4], QRON [5], and OverQoS [6] 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. In general, overlay networks provide an abstraction of the underlying networks towards the applications. Overlay networks are therefore more generic than IMS and allow dedicated application components, hosted on an overlay node, to perform some service-specific processing. Moreover, these proposals route data based on IP addresses, independently of the application specific components, and do not take the characteristics of the service (e.g., application running between peers), user or network constraints into account when deciding the best path.

In this paper, we propose a novel architecture based on overlay networking concept, which allows dedicated components to process media data on the path, and also provide media and service specific input to routing components when deciding the optimal network route: we call this approach "Media aware Overlay Routing".

The overall architecture of our approach is presented in section II before describing in more details the Media aware Overlay Routing in section III. In section IV, we shortly describe the relation of our work to QoS provision and we introduce its use within the Ambient Networks projects in section V. Finally, section VI concludes this paper and introduces the future work.

II. Description of the Architecture

This section provides a general overview of the Smart Multimedia Routing and Transport (SMART) architecture. 'Smart' multimedia routing and transport in this context means the optimisation of media delivery services by taking advantage of network-side media processing capabilities present along the end-to-end delivery path.

A key innovation of the SMART architecture is the concept of service-specific overlay networks (SSONs). The novel idea is that a different *virtual network*, or *overlay network*, is deployed for each (media delivery) service (or group of services), see Figure 1. This allows the configuration of appropriate high-level (media aware) routing paths that meet the exact requirements of a media service, e.g., QoS, media formats, responsiveness, cost, resilience, or security. Additionally, it enables the transparent integration of network-side media processing capabilities, such as caching, adaptation and synchronisation, into the selected end-to-end delivery paths.

Network-side processing actions can be performed separately for each media component, or *Media Flow*, of a media service, e.g., audio and video streams of one videoconference session, within the SSON. The proposed SMART architecture (Figure 1) incorporates Overlay Nodes (ONodes) and the Overlay Support Layer (OSL), which are all controlled by the Overlay Control Space (OCS). The OCS is distributed and a part of OCS is inside the ONode.

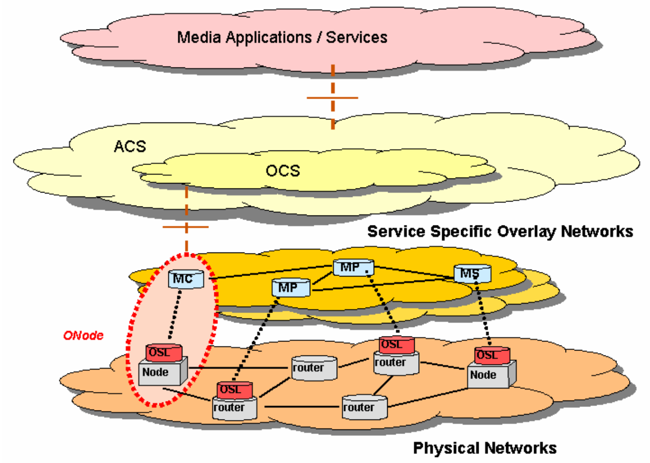

Figure 1. The Smart MultimediA Routing and Transport architecture (SMART).

In short, the architecture has the following characteristics:

- **ONodes** are specialized Ambient Network nodes that implement the necessary functionality to become part of the SSONs, like the provisioning of network-side media processing capabilities, such as caching, adaptation, synchronization and Media aware Routing (specified in III), inside the network. ONodes can be described in terms of the user and the control plane. For each SSON, of which the ONode is part of, Media Ports (MPs) are instantiated. MPs are responsible for Media Routing in the control plane and, in the user plane, host the so-called *application modules*, each responsible for a particular network-side media processing functionality, as mentioned above. Furthermore, and depending on the required media processing functionality, overlay nodes can take on the roles of Media Clients (MC), Media Servers (MS), and Media Ports (MP) – or any combination of those. Note that a single ONode can be part of many SSONs at the same time. The UML model of an ONode is represented in Figure 2. *Application modules* depicted there are the cache, the adapter and the synchroniser.

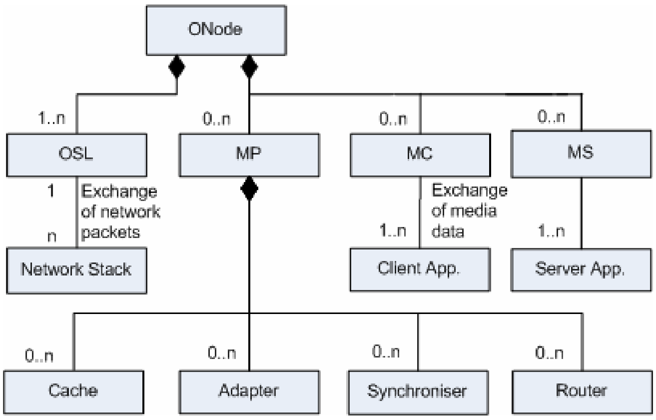

Figure 2. ONode Architecture Model.

- The **OSL** carries out the basic overlay network functionality required in every ONode. The OSL is responsible for the packet handling on the overlay: packet sending, receiving and forwarding on the overlay layer. To do this, it shall establish virtual links between the different ONodes, for example, by means of tunnels or other techniques. A connection between two ONodes is called a *bearer*. This forwarding behaviour is controlled by SSON-specific routing tables, in which the media-aware routing decisions are finally reflected. Each ONode in an SSON needs to be addressable, i.e., *reachable* somehow in the network. In our proposed architecture, the OSL within an ONode is the only entity that needs to be addressed with network addresses (IP addresses or equivalent locators). The reason for this is that the OSL is the interface between the network and the overlay layer, e.g., when de-multiplexing the packets to the appropriate SSON and so on.

  The functionality of the OSL can be summarized as follows:

  - Interconnection of the ONodes through *virtual links* (also called *bearers*) and forwarding of overlay packets over these virtual links
  - Use of SSON specific routing tables: the OSL must forward the overlay packets to the correct ONode (or next hop) in all situations. To do so, it uses routing tables configured by the OCS.
  - Data delivery to the upper application modules of MPs: the application state allows the ONodes to perform specific processing at each ONode and for each Media Flow. The exchange of data between the OSL and the appropriate application modules in the ONode when caching, doing adaptation or synchronization operations is governed by the *application state* (see III).

  It is important to emphasize that the OCS performs the routing at the overlay level. The OSL merely forwards or handles packets locally based on the information included in the Media Routing tables.

- Finally, the **OCS** embodies the routing functionality. As in every network, in overlay networks, routing is a core function. In our case, since ONodes perform media processing, the routing algorithm must take into account media service requirements. Media aware Overlay Routing is therefore not done at IP level but also at the overlay level, thus, enabling the introduction of any kind of routing constraints (user preferences, QoS needs, network context, etc) as specified in the overlay media routing logic we have designed. Details of the routing logic are specified in the following section. Note that the OCS is the part of the ACS that deals with establishment and control of SSONs.

III. MEDIA AWARE OVERLAY ROUTING

In this section we deal with the Media aware Overlay Routing logic, which is the core functionality of the OCS. As mentioned in the sections above, there is a necessity to deal with heterogeneous access networks and user capabilities and preferences that may change in time. We argue that the best way to cope with this problem is to introduce media processing capabilities in the network, namely as integral components of the SSONs.

The Media aware Overlay Routing Logic or Media Routing Logic (MRL), for short, is responsible for selecting and configuring the nodes that shall constitute the SSON for a particular service or set of services. The mechanism is triggered by the reception of service request, containing a set of service requirements that need to be fulfilled between two nodes (unicast) or between several nodes (multicast/ broadcast/ peer-to-peer). Based on the request, the MRL first decides whether a SSON is required at all and, if so, checks if an existing SSON may be used, thus aggregating SSONs for scalability purposes. In case a new SSON is required, the MRL selects the appropriate ONodes to be included in the overlay and configures the required routing table information at these ONodes. This means that packets belonging to a particular SSON are routed using service-specific routing tables, i.e., follow a *virtual topology*. Finally, after the SSON has been established, some monitoring is needed, as in any routing process.

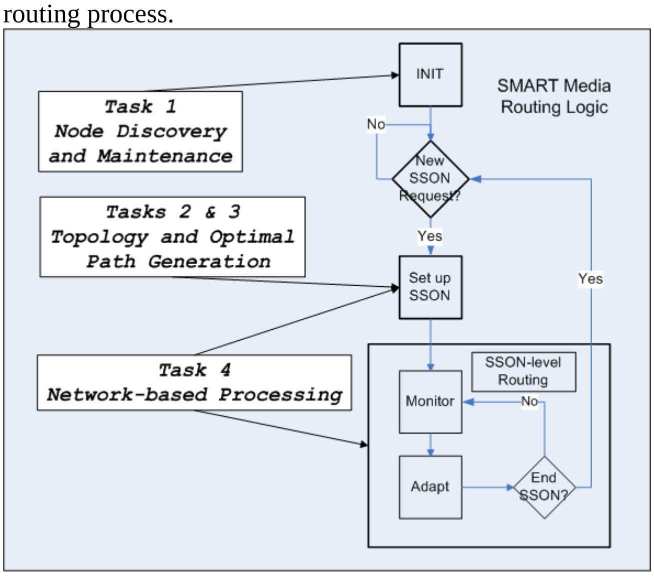

Figure 3. Steps of the Media Routing Logic.

The steps taken by the MRL are described with a flow diagram in Figure 3 above.

Until now, what we have described sounds very similar to a regular VPN with virtual routers [7]. There is, however, one major difference: in addition to the distribution of the service-specific routing tables, the MRL shall also take care of instructing the ONodes to perform processing on a per (media) flow basis, where needed. This per flow processing is performed by different application modules inside the MP that implement, e.g., transcoding, caching, flow splitting, thinning, shaping, etcetera.

Accordingly, some a*pplication state* shall be installed by the MRL at the participating ONodes so each ONode shall be told what to do with the packets that belong to a specific SSON.

Therefore, in order to enable MRL, we have identified the need for the following set of identifiers to be resolvable from an overlay packet:

- **SSON-ID**: this is the identity (ID) of the SSON and shall uniquely identify the service-specific routing table present at the OSL.
- **ONode-ID**: this is the identity of a particular MS, MC or MP in the SSON.
- A **Media Flow-ID**: in some cases, it is required to identify a Media Flow or flow of packets on a SSON that have different QoS sensitivities. I.e., audio and video flows belonging to a single service but having different bandwidth and delay jitter requirements. This ID enables the provision of different routing and processing actions on a per flow basis inside the same SSON.

A. *Where do all these identifiers go?*

There are several options to implement these identifiers in an actual packet:

- **Explicitly:** this can be accomplished using IP-in-IP tunnelling or application-level tunnelling. The former has some drawbacks since the addresses of the nodes have to be made unique within the SSON (much like the VPN-ID [10] problem). A similar approach has been adopted in X-Bone [11], which uses recursive IP-in-IP tunnelling for this purpose. However, this option does not accommodate QoS well, since nested headers incur significant overhead. It is also necessary to run the whole set of routing protocols in each of the SSONs and it requires a BGP/ARP hybrid protocol to resolve the identity of the IP ONode-to-ONode *links* used. On the other hand, as claimed in [9] the inclusion of separate overlay header containing the SSON-ID, ONode-ID and (if needed) Media Flow-ID in each packet is a straightforward solution to this problem.

  Figure 4 illustrates an example of an explicit solution and how these identifiers are used to forward packets by looking up next hop addresses in the specific SSON routing tables. A possible packet format is also depicted.

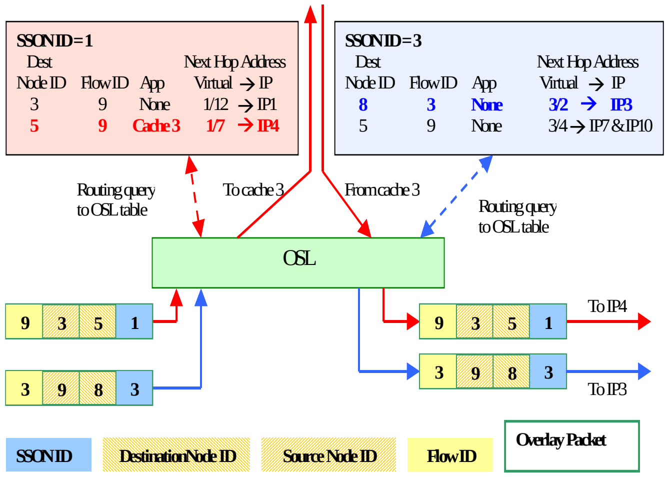

Figure 4. Basic Forwarding Behaviour for the Explicit Solution

- **Implicitly:** as Balakrishnan et al. [8] claim, these identifiers need not be present explicitly as new protocol headers in all packets. The disadvantage of this solution may be the added *Media Routing state* in the media routing tables, i.e., which protocol headers identify which SSON, in addition to the *application state* mentioned before.

B. *Further steps*

In this section, we have presented the basic components of the Media aware Overlay Routing Logic, MRL, for short. Thereby, it seems of particular importance to identify the set of identifiers needed in each overlay packet as well as the means for identifying and performing the required set of operations on each overlay packet (as required).

Furthermore, we have encountered two main options for accomplishing the Media Routing Logic in today's networks by including the headers in the overlay packets in explicit or implicit form. Each of these has advantages and disadvantages. In our further work, we shall evaluate these and assess which solutions are best suited for each of the target scenarios.

IV. Relation to QoS

At this point, it should be noted that the SMART architecture is decoupled from QoS mechanisms. This means that any QoS mechanism may be used in the underlying network (DiffServ, MPLS etc.) Our work does not directly provide QoS assurance but, instead, we rely on other mechanisms within the Ambient Networks to provide it for us.

One of the main features of Ambient Networks is the existence of a common control entity for the whole network, namely the ACS (Ambient Control Space) [1]. In this *space*, there are some so-called Functional Areas (FAs) that control a particular set of functionalities within the network. Among them, there are a mobility FA, a Context FA as well as a QoS FA as well. The QoS FA is in charge of QoS provisioning, through the use of the mechanisms it considers most appropriate.

[1] This does not mean the ACS is one central physical entity. It has to be understood as an abstract space in which all the network and context control functionalities are placed

Therefore, based on the QoS and other constraints received in the service request, together with the requirements to perform media processing actions, the OCS decides which are the ONodes that will conform the SSON. At this point, it contacts the QoS FA and requests the establishment of the appropriate *flows* (different from Media Flows) in the underlying network that fulfil the requirements of each *virtual link or bearer* of the overlay[2]. In other words, the MRL determines what the ideal *bearer* type would be but it does not provide the *flows* itself. In case the requested *bearer* is not feasible, then the MRL iterates the (media) routing decision, taking into account the available QoS reported back from the QoS FA, and tries to provide a better alternative than just *best-effort.*

The introduction of QoS as an additional constraint to be taken into account in the MRL is achieved through the use of appropriate cost functions for each of the metrics (availability, bandwidth, cost, delay...) considered in the *virtual topology*. The separation between this decision taking and the actual QoS provisioning provides a cleaner vision on the Media Routing process.

Thus, generally speaking, for establishing QoS-aware overlay, an interface is required that 'talks' to the QoS broker as to which *flows* need to be provided. 3GPP networks typically provide such an interface. Others, like Internet rely more on over-provisioning, as a way to side-step the difficulties related with QoS provision and resource management.

V. Use Case

In the following an exemplary use case for the SMART architecture is given. Based on the situation of a business worker in his office, the capability of network-side media processing is described. In future office environments, the mobility of users will play a key role. Therefore ongoing sessions like conference calls or incoming VoIP connections have to be switched always to the most suitable end device. Thus, the media stream has to be adapted to the capabilities of the end device. Furthermore, splitting of the media session into separate audio and video flows might be necessary so that the user can for example see her dialog partner in a video conferencing session on her handheld, while the audio stream is directly routed to her Bluetooth headset. If the user is unavailable for a moment, temporary caching of the data targeted to this user might also be done in the network.

All these functionalities require an active network support, which is provided by the SMART architecture. Each communication takes place between MSs and MCs and several nodes, so called Media Ports, in between. These MPs accomplish the relevant adaptations or other processing of the media stream. If a session is transferred e.g. to a PDA and a headset, an adjustment to the small screen size and the lower connection bandwidth is required. Therefore a new ONode with transcoding capabilities is integrated into the overlay network so that the video stream can be sufficiently displayed on the PDA. This network-side support liberates the server as well as the client from this burden, and distributes these support functionalities from the endpoints into the network. Moreover, another ONode splits the incoming stream so that the audio part is separately transmitted to the headset, which is acting as a Media Client in this case (see Fig. 5).

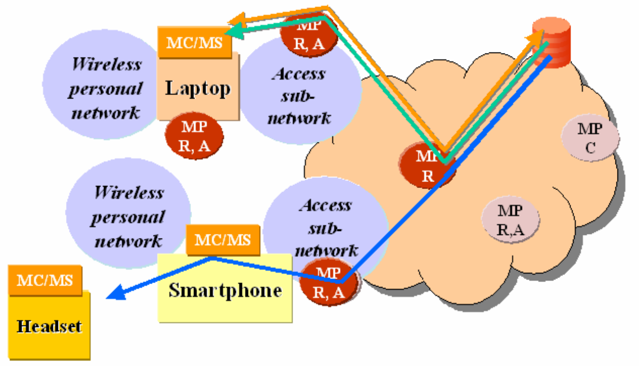

Figure 5. The audio flow is handed over to another access network and, eventually, to another device.

All synchronization is maintained by the MPs within the overlay network so that the above network layers and the application itself are completely isolated from this task. This means that applications do not have to be aware of the presence of an Ambient Network or, for that matter, of an SSON. This is part of the essence of network-side media processing.

VI. Conclusion & Future Work

In this paper, we have introduced the concept of Media aware Overlay Routing. We have also specified an architecture that illustrates the feasibility of this concept. Our architecture utilises the Media aware Overlay Routing Logic, thus introducing a coupling between media session characteristics, routing decisions and application (media processing) modules, which reside inside the Overlay Nodes (ONodes). In this manner, network, content, server and user characteristics may all be taken into account when generating the optimal path, discovering the next hop and performing network-based processing of the media (like adaptation or caching).

We have also presented how our work is related to ongoing work on QoS provisioning. Our solution offers a *generic* framework for service and content providers to be able to customize their services to the specific needs of a user or service, regardless of the QoS mechanism used.

In our further work, we will first assess the applicability of *explicit* and *implicit* solutions for the coding of required Media Routing information into the protocol headers, i.e., the use cases. This first step will shed some light on the advantages and disadvantages of these two options. Afterwards, we will develop the set of directives and logic required to configure the network-side processing at the ONodes. Next, we plan to evaluate our concepts via implementation and simulation work; resulting from this we expect to obtain detailed information regarding scalability, performance and overhead. We shall then evaluate the security issues, as well as the feasibility and deployment of our results into current and future standardization efforts like IETF or 3GPP. This also means to assess whether it is possible to implement our generic framework by means of

[2] Typically, each underlying *flow* will provide a specific set of QoS characteristics and, then, if different media flows of the session require different QoS, they would be transported over different underlying *flows*, which are mapped later to the same *bearer* at the overlay level.

extensions to existing protocols and frameworks, like SIP and IMS.

ACKNOWLEDGMENT

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

REFERENCES

[1] The Apache Cocoon Project, http://cocoon.apache.org.
[2] SIP : Session Initiation Protocol (RFC 3261) http://www.ietf.org/rfc/rfc3261.txt.
[3] IMS : IP Multimedia Subsystem : 3GPP TS 23.228 : http://www.3gpp.org/ftp/ specs/html% 2Dinfo/23228.htm
[4] Resilient Overlay Networks, David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, Robert Morris, Proc. 18th ACM SOSP, Banff, Canada, October 2001.
[5] QRON: QoS-aware Routing in Overlay Networks, Zhi Li, Prasant Mohapatra, IEEE JSAC, 2003.
[6] OverQoS: An Overlay Based Architecture for Enhancing Internet QoS, Lakshminarayanan Subramanian, Ion Stoica, Hari Balakrishnan, and Randy Katz , Proc. 1st NSDI, San Francisco, CA, March 2004.
[7] J. De Clerq, et al., " An Architecture for Provider Provisioned CE-based Virtual Private Networks using IPsec", IETF draft, draft-ietf-l3vpn-ce-based-02.txt, August, 2004.
[8] H. Balakrishnan, et al., "A Layered Naming Architecture for the Internet", Proc. ACM SIGCOMM Conference (SIGCOMM 2004), Aug. 30 - Sept. 3, 2004.
[9] Ambient Networks Work Package 5, "SMART: Draft Architecture and Multimedia Routing Decision Logic", http://www.ambient-networks.org/publications/D5-1_PU.pdf.
[10] B. Fox, B. Gleeson, “Virtual Private Networks Identifier”, RFC 2685, September 1999.
[11] J. Touch, S. Hotz, "The X-Bone", Globecom, 1998, Sydney, Australia, November 8-12; The X-Bone, http://www.isi.edu/xbone.
[12] 3GPP TS 26.234v6.2.0 “Transparent end-to-end Packet-switched Streaming Service (PSS);Protocols and codecs(Release 6)”, http://www.3gpp.org/ftp/Specs/html-info/26234.htm