Deployment and experimentation of an active network at a large scale: AMARRAGE

Rim Hammi[1], Prométhée Sathis[2], Dany Zebiane[3], Ken Chen[1], Ahmed Serhrouchni[3], Kim L. Thai[2]
[1]Université Paris 13- Laboratoire L2TI; [2]Université Pierre et Marie Curie ; [3]Ecole Nationale Supérieure des Télécommunications

Abstract: This article describes an active network testbed, AMARRAGEbone, deployed across France as part of the RNRT[1] AMARRAGE[2] project. Established to demonstrate the feasibility of the active network architecture jointly conceived and developed by the consortium, this platform enabled the large-scale deployment of active software components. These components were evaluated and validated on the AMARRAGEbone, in the context of innovative and representative services of future technological needs and applications. This article details three network services specified and implemented in accordance with the AMARRAGE architecture: a reliable multicast protocol, a framework for responsive control of video communication, and a dynamic deployment mechanism for active services.

Key words: Active network, large scale platform, reliable multicast, video, service deployment and validation.

1 INTRODUCTION

Introduced by Tennenhouse and Wetherall [15, 16], the concept of the Active Networks becomes reality with the announcement of active routers from several companies and with the output of many architecture proposals, in particular the AMARRAGE RNRT project. If this concept promises to increase the operators and the services providers' reactivity, it is primordial to meet customers' needs for its introduction into the market. The AMARRAGE [2] project conceives a

[1] Réseau National de Recherche en télécommunications
[2] Architecture Multimédia & Administration Réparties sur un Réseau Actif Grande Echelle

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35620-4_43

D. Gaïti et al. (eds.), *Network Control and Engineering for QoS, Security and Mobility*

new active network architecture that answers to the future technologic needs of applications.

Under the aegis of the Economy, Finances and Industry Ministry, the AMARRAGE project started in April 2000. It gathers research laboratories and industrial companies: ENST (Paris), L2TI (U. Paris13, Villetaneuse), LAAS (CNRS, Toulouse), LIP6 (UPMC, Paris6), LORIA (INRIA Loraine), Thalès (Gennevilliers) and France Telecom R&D (Issy-Les-Moulineaux).

The contributions of the AMAGRRAGE project in the field of Active Networks are of two types: first we have defined, conceived and developed new active network architecture around significant applications. Second and for an industrial acceptance of the proposed architecture, we set up a demonstration platform at the geographical scale of France called AMARRAGEbone. Indeed, this platform allows us to evaluate and to validate the studied solutions within the framework of this project. The active software components, resulting from the implementation of the retained solutions, are currently deployed on a large scale. Some innovating and representative network services specified to validate the feasibility of the active network architecture, have been conceived and developed by the AMARRAGE consortium. This article presents three network services that have been tested on the AMARRAGEbone: a reliable multicast service called MAF (Reliable Active multicast) [13], an active reporting mechanism for responsive control of video communication [8] and a mechanism for the dynamic deployment of new services [18]. Other services such as the signature, the encryption of the active capsules and the composition of services are currently developed within this project.

The remainder of this paper is organized as follows. In the second section, we present the architecture of an AMARRAGE active network node. Section 3 gives a detailed description of the national demonstration platform. Section 4 details the specification and the implementation of the previously mentioned three services. In section 5, we discuss the experimental results obtained from the deployment of these three services on the platform. Based on our acquired experiment, section 6 outlines our conclusion and give a brief discussion on our future work.

2 ARCHITECTURE OF AMARRAGE NODE

We present, in this section, an overview of the architecture of an AMARRAGE node by following the terminology of the architectural framework being developed by the Active Network Working Group [3]. As depicted in figure 1, we describe the implementation of the three major components and interfaces identified by the architectural framework. The functionality of an AMARRAGE active network node is divided among the Node Operating System (NodeOS), the Execution Environment (EE) and the Active Applications (AA):

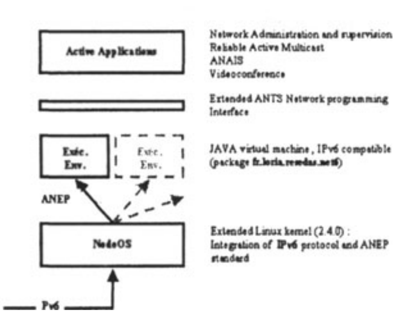

Figure 1: AMARRAGE Node Architecture

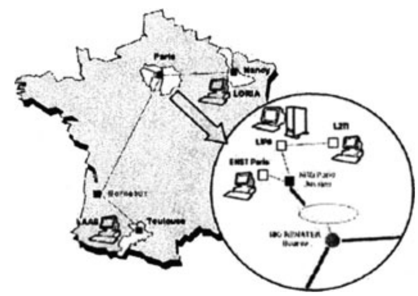

Figure 2: AMARRAGE Network

- The NodeOs is an extended Linux kernel. It performs: the demultiplexing of the active packets to multiple EEs located on the same node, by using the Active Network Encapsulation Protocol (ANEP) [1, 7]. Unlike the Abone [5], the AMARRAGE NodeOs implements communication channels using IPv6. Suppressing UDP removes an indirection level in the active packets processing.
- Currently, each AMARRAGEBone node supports a JVM-based EE, which is permanently available to AA developers. The virtual machine is fully compliant with IPv6 protocol. It provides a programming interface used to access to the IPv6 network services.
- The programming interface used to build all service prototypes is supplied by the execution environment (EE). The exported network API is derived from ANTS [3, 17]. The initial architecture was extended to support the specifics additional needs of significant active applications like those that MAF and the Active Video require. Those additional functions are secure deployment, configuration, and control of AN software.

All these Active Network components are currently supported at the edge active routers provided by each participating research site.

3 AMARRAGEBONE: THE NATIONAL TESTBED

We have implemented these prototypes in accordance with the active network architecture jointly conceived and developed by the AMARRAGE project consortium. The AMARRAGEbone is an RNRT-funded wide-area testbed that supports the AMARRAGE active network research program. The AMARRAGEbone forms a virtual network infrastructure on which active network (AN) components are currently tested and experimentally deployed. This testbed provides an experimental infrastructure to support the deployment, testing, and integration of the active software components being produced by the AMARRAGE research project. Figure 2 illustrates the AMARRAGEbone physical and logical topology. The experimental environment is distributed across Paris, Toulouse and Nancy. Because of the technological choices, the testbed is built upon the RENATER infrastructure and it benefits from the provided IPv6 native connectivity. AMARRAGEbone has connectivity to the rest of the IPv6 world through native links and tunnels. Connectivity varies between partners' sites. Edge nodes located at ENST and L2TI get access to the other sites by being connected to the nearest RENATER NRD, through the use of IPv4 tunneling. LIP6 and LORIA are connected to the national Backbone that REANTER provides and have thus native IPv6 connectivity.

4 DEPLOYED SERVICES

The active network architecture developed in AMARRAGE [2,3] project meets the needs of significant network services. The services presented in this article are: Deployment of new services, Reliable active Multicast and Active Real-Time Video. Other services have also been developed but they are not detailed in this paper, like supervision, administration, service composition, etc.

4.1 Deployment of new services in the AMARRAGE architecture

4.1.1 Architecture

In the traditional platforms of active networks, we find only one plan gathering the active processing and the data transport. In the AMARRAGE project [2], our contribution consists in structuring the active network, such as in the CAN architecture [18]. The CAN's[3] idea is inspired from the traditional networks such as the telephone networks, this architecture is composed of three plans: The *control plan*, the *active plan*, And the *transport plan*.

The control plan gathers specific nodes - control and management - which deal with the following mechanisms:

- The loading of the new services within a code server, called control node in [18], so it releases the normal active nodes and preserves their resources to another type of processing.
- The configuration and administration of the active nodes and the services which they support.
- The maintenance and the loading of user profiles in the normal active nodes. Example of user profiles are the authorization given to a user to install certain services or applications within a specific node of the network, or the resource policy given to a user, ...

In the AMARRAGEbone, we install one control node per site, the administration of these sites is maintained by one management node, installed for our entire platform. The active nodes which constitute the active plan export a standard API, allowing the access to the resources of the node as well as the data transmission. These nodes handle the headers and the data capsules, they make some information reachable like their routing tables, they store the services code in their caches and finally they communicate

[3] Controlled Architecture for active Network

with the other active nodes to have a global view of the network. So these nodes offer some value added services to the users of the network.

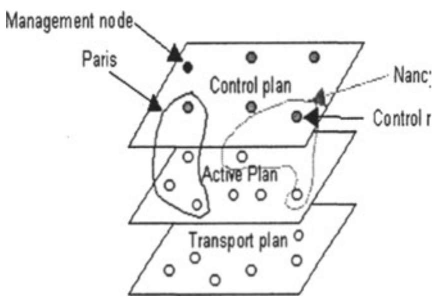

Figure 3: CAN architecture

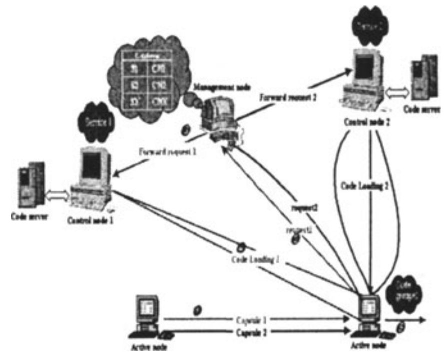

Figure 4: Loading protocol

The transport plan gathers the nodes transporting the active capsules, without carrying out a particular processing, they act as traditional routers. Figure 3 shows these three plans and the corresponding nodes.

The management node is an active node supporting the administration application. It is responsible for the configuration and the administration of the code servers and the registration of new services. This application offers a GUI allowing the observation of all the important registration events taking place in the network in a window status. Another window shows all the registered code servers at this management node, as well as their corresponding services. This node will be responsible for receiving the registration of the control node and the recording of new services for managing the network in a controlled manner. The management node switches the requests between the active nodes and the corresponding control nodes.

The control node is a normal active node enhanced with a Server Code; it is responsible for controlling its domain in the network and the code loading from the corresponding Server Code. The control node supports the control application; it is introduced into this suggested architecture to deal with the code distribution within the network. This application offers a GUI allowing the Control node to connect to the Management node, as well as to register its IP address and its services in the "administration.service" file on the management node after an authentication process. These services are also registered on a local file on the Control node: "Code.service". The interface also allows the removal of unsupported services.

4.1.2 Loading protocol for new service in the AMARRAGEbone

The AMARRAGEbone requires the conception and the implementation of a new protocol to load the code of new services. To identify control nodes, this protocol implements an explicit identification mechanism between the control nodes and the management node; in the current implementation the connection is based on password verification. If the identification succeeds, the control node owner can then add or remove dynamically services within the code server. The loading of new services in normal active nodes occurs as illustrated in Figure 4.

1 When a capsule reaches an AMARRAGE node, called *loading node*, where the code associated with its type misses in the node cache, a loading request is sent to the management node.
2 When the loading request arrives at the management node, it seeks in a local file called "code.service". This file contains the IP addresses of all the control nodes (Server Code) of the network and their supported services. A service is recorded with the capsules' TYPEID for each service code groups.
3 If the searches succeed, the TYPEID of the capsule matches a value in the code.service file, the management node signs the received loading request in order to be to authenticated, then forward it to the corresponding control node.
4 When the control node receives the signed loading request, it checks the signature of the management node.
5 If the checking succeeds, the code is retrieved from the Code Serve, and then a series of loading response capsules is produced. The response capsules convey the code group to *loading node.*
6 The *loading node* reassembles the response capsules, to constitute the required code group. Once the loading finish, the code reassembly is achieved. The result is then controlled. The new code group can then be used to process and then forward the capsule which triggers the loading order.

4.2 Reliable active Multicast

MAF [13] (Reliable Active Multicast), is a fully reliable multicast protocol compliant with the AMARRAGE architecture. MAF takes advantage of the flexibility introduced by the AMARRAGE architecture by modifying router behavior in the multicast tree. MAF provides a scalable reliable multicast service through the introduction of error recovery mechanisms. The technologies specified in AMARRAGE allow us to dynamically deploy these mechanisms at active routers along the multicast tree. MAF organizes the receivers and active routers of the multicast tree into a hierarchical structure. Obtaining this structure amounts to dividing the logical tree that is associated with the multicast tree into subtrees. Each subtree has a depth of 1 and a degree of D. In a subtree, the root is the source or an active router while the leaves are receivers or active routers that are, in turn, the roots of lower level subtrees. We call the leaves of a given subtree a "subgroup," under the responsibility of the root of that subtree. We call the root the "chief" of the subgroup made up of its descendant leaves.

4.2.1 Error control

Inside each subtree, MAF executes a sender oriented approach, based on positive acknowledgments (ACKs) and retransmission timers. MAF delegates to subgroup chiefs the detection and the repair of losses located in the associated subtrees. A chief can be the source or an active router programmed to act as the source to its children. A chief stores and retransmits the data until all children have positively acknowledged correct reception. The decision to release data from memory is delegated to chiefs and depends on receipt of ACKs from the children of the subgroup. Children periodically send positive acknowledgments to their chief, without waiting their own children to send ACKs. MAF requires a chief to receive ACKs from all its children before it is allowed to release data from memory. A data packet not positively acknowledged is interpreted as being lost; the chief of the subgroup then retransmits the lost data packet. To detect lost acknowledgments, chiefs use a retransmission timer and know the composition of their subgroup. If there is a timeout before all ACKs have been received, the packet is assumed to be lost and is selectively retransmitted by the chief to its affected children. This mechanism prevents receivers from receiving unnecessary repairs.

4.2.2 Memory Management

A chief releases a packet from memory after it has been correctly received by all of its children. A chief can then receive new data for which it can guarantee reliable multicast to its children. A chief explicitly indicates to its own chief the identity of the data packets that it has released from memory by sending an "Aggregated ACK," or AACK. Chiefs effectively aggregate ACKs received from their children. A chief thus indicates to its own chief the amount of data that it is capable of receiving; this amount corresponding to the memory that it has liberated. This mechanism allows each chief to insure totally reliable data transmission to its children with finite memory. A chief systematically repairs losses affecting its own children, and does not call upon upstream chiefs to repair downstream losses.

4.3 Active Real-Time Video

4.3.1 Rationale

The handling of video communications is still a focus of investigation with today's networking infrastructure, which is dominated by Internet technology. This is because video traffics are real-time, heavy load and (selectively) sensitive to loss, and so need to be handled with a minimum of QoS control. One of the major concerns is the loss prevention.

In the Internet, the generic architecture for multimedia traffic has been standardized in H.323 [17]; the key element for traffic handling is the use of RTP/RTCP [6, 14] over UDP links. The RTP Protocol ensures the integrity and the facility of synchronization of video streams. The RTCP (Real Time Transport Control Protocol), the companion protocol of RTP, is designed to yield feedback, under the form of report packets, which contains QoS information such like low ratio and received bandwidth.

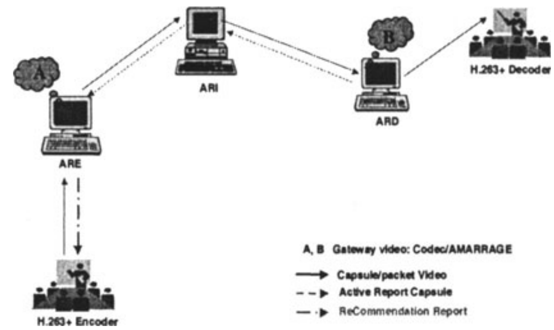

Figure 6: An overview of the active reporting mechanism

One usage of RTCP is to facilitate the rate adaptation. Indeed, RTCP reports allow the sender to get a hint about the link's state as well as reception quality. Nevertheless, several factors limit the efficiency of a RTCP based control mechanism. Indeed, a) the overall bandwidth of RTCP is limited to 5%, thus, the frequency of the feedback has to be lower than 1 report every 4 second (0,25 Hz); b) the feedback information is predefined and cannot be user-specific (for instance a more-accurate two states loss pattern).

An active networking capable network infrastructure provides more flexibility for individual flow handling and the capability of dynamically reconfiguration, by updating the code carried in a capsule. We believe, and our experimentation shows, that the active network can offer a more dynamic and use-specific rate control mechanism. The main specific contributions of the active technology are a) Loading of user-specific codes at user specific (observation and/or action) points with user specific code, b) Monitoring the flow and access to local information at a user specific rhythm.

4.3.2 Active reporting mechanism

Hereafter, we describe shortly the *active reporting mechanism*, a framework allowing responsive video communication regulation. This work has been conducted jointly with other Amarrage partners (a more detailed presentation of the framework can be found in [8]). A scheme of this framework is given in figure 6. The key element of our mechanism is the use of two user-agents (active probe, which can be dynamically installed) at respectively ARD (the Amarrage Router closest to the Decoder) and the ARE (the Amarrage Router closest to the Encoder) see figure 6 (ARI are possible intermediate active routers). The main role of the agent at ARD is to take information about the local link state, including particularly bandwidth, loss ratio and loss pattern, as well as the delay jitters. The user-agent installed at ARE decides the rate-control to be taken, according to the current link state. This is done through the sending of a control packet, called RCP (ReCommendation Packet), back to the encoder. An ARC (Active Report Capsule) is sent periodically from the ARD to the ARE. When leaving the ARD, it carried the channel state viewed by receiver, such as the maximum/minimum rate at which the user accepts to receive the communication. With this full and up-to-date vision of the channel, the ARE can provide an accurate feedback (RCP) to the encoder, in order to adapt the rate of the video communication.

5 EXPERIMENTATION RESULTS

5.1 Loading protocol for new service in the AMARRAGEbone

The CAN architecture [18] prove that there is a great interest in adding control and management components to an active network, in order to add more efficiency and reliability to the service deployment. The implemented code loading protocol for new services and the Server Code are two steps to secure the deployment of new services, like these only authorized users can remove and add new services. To improve the security of our platform a user defined policy mechanism will be implemented. [18] does not show any degradation in performance, in contrast some amelioration is detected in term of security for new protocol registration. To bring out our obtained results, the same experiments are done simultaneously on the native ANTS [4] prototype, as well as on our AMARRAGE platform.

Figure 6 shows the RTT (round-trip time) obtained by the deployment of an active ping service. The X-axis shows the sequence number of the capsules while the Y-axis represents the RTT. We observe that for the first capsule, the measured RTT in the AMARRAGE platform is raised by the loading time of the code implanting the active Ping service. However, once the code is downloaded, the two platforms have the same behavior. Figure 7 shows that the performances are not modified, following the introduced modifications. The throughput of the two platforms is quite equal and the ratio delay/security is very acceptable.

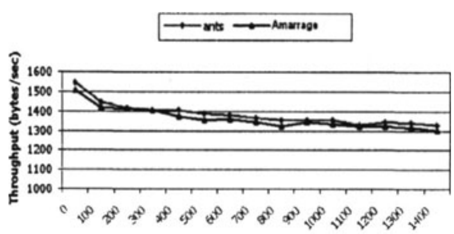

Figure 6: Round Trip Time

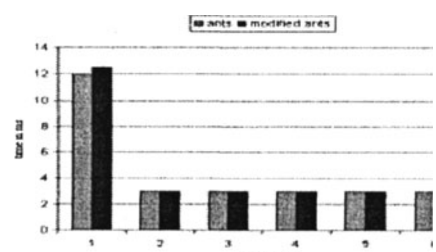

Figure 7: Throughput as a function of payload of capsule

5.2 Reliable Active Multicast

The AMARRAGEbone allowed us to deploy the MAF prototype on the geographical scale of France. We did conduct those national scale experiments to show the scalability and efficiency of MAF. As we expected because of MAF design decisions, the loss recovery delay, the time from when a loss occurs to when it receives the (first) repair for that loss is a function of its chief's retransmission timer value and the RTT between its chief and its chief's furthest child. Because chiefs systematically take charge of repairing the losses that affect their children, they spare (shield) the source and upstream chiefs from processing/receiving (the processing of) ACKs and AACKs sent by others than their immediate children. Chiefs also (as well) provide faster recovery and thus reduce/decrease end-to-end latency (handling local retransmission requests) even though MAF objective is to guarantee complete reliability at the expense of delay. Chiefs only process ACKs and AACKs from their immediate children. The subtrees that are rooted at a chief are thus hidden from the upstream chiefs or the sender. Thus, using active routers in the multicast tree not only reduces the number of ACKs the sender has to process but also improves the usage of network resources. Losses occurring in different parts of the multicast tree are concurrently recovered. During our experiments, no chief had to replace already buffered packets with new arriving packets before being successfully received by their children. Losses occurring in a given subtree thus caused no additional traffic in other parts of the multicast tree: the traffic needed to guarantee reliability in a given subtree is always confined to this subtree.

5.3 Active real-time Video

Our architecture has been implemented and evaluated on the AMARRAGE test-bed. The nodes we used for the video experiments are PC with Pentium II-450 MHz and 256 MB of RAM, run under Linux kernel 2.4 and are connected by 100 Mb/s Ethernet links. The codecs (H.263+) run on classical machine (actually users of active networks are not necessary based on active network. For our experiments, we use the Foreman sequence, encoded as CIF at full rate (30 frames/s). The stream is sent in a RTP unicast session. The RTP packets are converted into capsules (Video Capsule), and conversely, by a function integrated into the AMARRAGE platform. The RTCP packet generation is inhibited. .

The user-agent, described above, supervise the quality of transmission by Measuring loss ratio and loss pattern, particularly bandwidth and the delay jitters. We chose a frequency of measurement of one second. Figure 8 shows an experiment that illustrates the basic idea of the mechanism. We maintain a video rate at around 460 kbps with a step of 25 kbps (5,4 % of the video traffic). When loss occurs, a RCP is sent for rate diminution. As the flow is always under monitoring, we try to later augment the rate. We are still investigating an optimal way for rate changing.

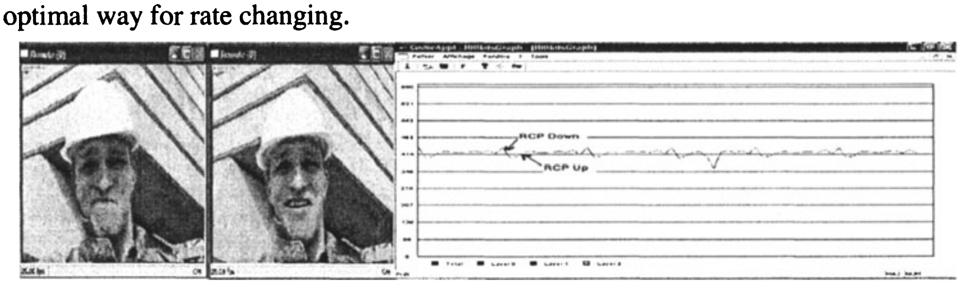

Figure 8: The test sequences Foreman (top row). The right frame shows the difference result between the left frame, after increasing smoothly the video traffic (RCP Up). The bottom row gives the throughput.

We have not observed during this experiment noticeable oscillation which could be visible by human eyes.

Our experiments show one major advantage of our active reporting mechanism, which is the fact that the active router can adapt the video traffic to the currently available bandwidth in a responsive manner, with a whole vision of the network. The active control mechanism offers a global and faithful sight of the state of the network borrowed by the stream and the state of the real-time video transmission. Indeed, the recommendations sent periodically by the AMARRAGE network to the encoder allow it to react more quickly to the variations of the available resources in the network as well as to the needs of the decoder. Moreover, the active reporting mechanism allows a feedback rate more adequate and specific to the application. We want to point out that our active network based mechanism offers a scheme to some other active network solutions, such as the proposition describing in [9] which is based on frame omitting and/or filtering.

6 CONCLUSIONS AND PERSPECTIVES

This article introduces AMARRAGEbone, a flexible and opened demonstrator platform, which allows the fast deployment of new services never been tested before at a large-scale network. The presence in AMRRAGE consortium of the majority of the French academic laboratories, dealing with active networks topics, and the industrial partners made possible the deployment of AMARRAGEbone on a geographical scale of France. The expertise acquired within AMARRAGEbone platform enables us to deploy at the national level three innovative services and representative of the future uses of the active network. AMARRAGEbone allows performance measurements, at a real scale, of value-added network services. These experiments validated the interest of the active network concept and the feasibility of the architecture adopted in a multi-sites environment (Nancy, Paris, Toulouse) by showing the facility of design, deployment, configuration of new services and protocols. The acquired experiment confirms that the suggested AMARRAGE architecture opens new prospects with a great interest for new services design. Indeed supplementary validations such as security still not deployed. The results obtained within the AMARRAGEbone platform show that the architecture of AMARRAGE meets the identified needs for the significant services

7 REFERENCES

[1] D. Alexander, B. Braden, C. Gunter, A. Jackson, A. Keromytis, G. Minden and D. Wetherall, "Active Network Encapsulation Protocol (ANEP)", RFC Draft, juillet1997.
[2] http://www.amarrage.net.
[3] AN WORKING GROUP. "Architectural Framework for Active Networks". Draft: http://www.dcs.uky.edu/ calvert/arch-1-0.ps, juillet1999.
[4] ANTS site web, http://www.cs.washington.edu/research /networking/ants/.
[5] S. Bersen, B, Braden., L. Ricciulli, "Introduction to the Abone". June 2000.
[6] C. Bormann, L. Cline, G. Deisher, T. Gardos, C. Maciocco. D. Newell, J. Ott, G. Sullivan, S. Wenger & C. Zhu, "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)", RFC2429, Oct 1998.
[7] S. D'Alu, G. Chelius, I. Chrisment, O. Festor and E. Fleury, "Intégration du support IPv6 dans l'environnement de supervision de réseaux actif ANAIS", in: CFIP, octobre 2000.
[8] R. Hammi, K. Chen, J.P Blin and F. Loras, "A framework for responsive control of video communication using Active Networks", WTC, 22-27 September 2002, Paris, France.
[9] R. Kilany, A. Serhrouchni, D. Zebiane. "Une architecture à base de composantes distribuées pour le déploiement de services actifs". Proceedings of CFIP2002, Montréal, Canada, 26-30 May 2002.
[10] ITU-T, Recommendation H.263 (version 2), "Video Coding for low Bitrate Communications", February 1998.
[11] H. Schultzrinne, S. Casner, R. Frederick & V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC1889, Jan. 1996.
[12] H. Schulzrinne, "RTP Profile for Audio and Video Conference with Minimal Control", RFC1890, Jan 1996.
[13] P. Spathis, K. Thai, "MAF : un protocole de multicast fiable". Proceedings of CFIP2002, Montréal, Canada, 26-30 May 2002.
[14] ISO Standard 13818-2, "Information Technology - Generic Coding of moving pictures and associated audio information: video (MPEG2)", July 1995 (also known as ITU-T H.262).
[15] D. Tennenhouse, D Wetherall, "Towards an Active Network Architecture". SPIE Proceedings of Conference on Multimedia Computing and Networking, San Jose, CA, janvier1996.
[16] D. Tennenhouse, and al, "A survey of active network research", IEEE Communications Magazine, 35(1), Jan 1997 pp 80-86.
[17] D. Wetherall, and al "ANTS: A toolkit for building and dynamically deploying network protocols", IEEE OPEN ARCH'98, San Francisco, CA, April 1998 pp 117-129.
[18] D. Zebiane, et al, "A Control Architecture for Active Network", IEEE SoftCOM 2001, Split, Croatia - Bari, Italy, 9-12 October 2001.