Streaming video contents in an active network over multicast IP using a layer model

Claude DUVALLET
LIH, Université du Havre
25 rue Philippe Lebon, BP 540
F-76058 Le Havre CEDEX, France
Claude.Duvallet@free.fr

Ronan KERYELL
ENST Bretagne/LIT
Technopôle de Brest Iroise, CS 83818
F-29238 PLOUZANÉ CEDEX
Ronan.Keryell@enst-bretagne.fr

Gérard BABONNEAU
France Télécom R&D
4 rue du Clos Courtel, BP 59
F-35512 Cesson Sevigné CEDEX
Gerard.Babonneau@rd.francetelecom.com

Abstract

The generalization of high bandwidth networks (cable, ADSL,...) permits us to consider a larger use of the Internet such as video streaming. We can imagine generalization of video on demand (VoD) and television channels on Internet. Nevertheless, the growth of multimedia contents transmitted on the network also increases the risk of congestion. Therefore, it is necessary to optimise the use of this one and to limit the traffic to the strict necessary while assuring fault tolerance. In the ReActiVE project[1] (Active Network for the transmission of Video at ENST Bretagne) we have designed a platform to transmit video contents on IP network. It is based on the distribution of contents and computing resources on the different nodes of the network, associated with multi-diffusion techniques in order to decrease stream circulation. In this article we describe the architecture we have implemented to build the platform.

1 Introduction

Although the goal of the Internet network at the beginning was rather to exchange textual information without specific real time constraints, many experts [8, 9] agree to say now that the next big challenge of the Internet deals with the streaming of video. Television is an inescapable media and it seems logical that telecommunication operators enter actively in this market in order to offer services that go beyond the plain broadcasting infrastructure supply.

[1]Project partially funded by France Telecom R&D and by the Foundation Louis Leprince-Ringuet

We understand that the underlying problems to the generalization of the multimedia stream transport on actual networks mainly deals with the transportation capacities of these networks. Paradoxically, the most important problem may be at the local level. In this setting, xDSL technologies are especially interesting since they provide (or will provide,in a short time) to the user enough bandwidth to follow one or several television channels simultaneously with a good quality because of video CODEC improvement at the same time. But the generalization of such technologies will lead to saturation issues on the intermediate network, between the local access and a backbone, or at the level of the backbones themselves. Indeed, the interest to open out such technologies—to be different of the classic broadcast network (and of the terrestrial digital video broadcast network)—deals with the ability to provide to the user interactivity, software and services with a high added value that have made the success of the Internet.

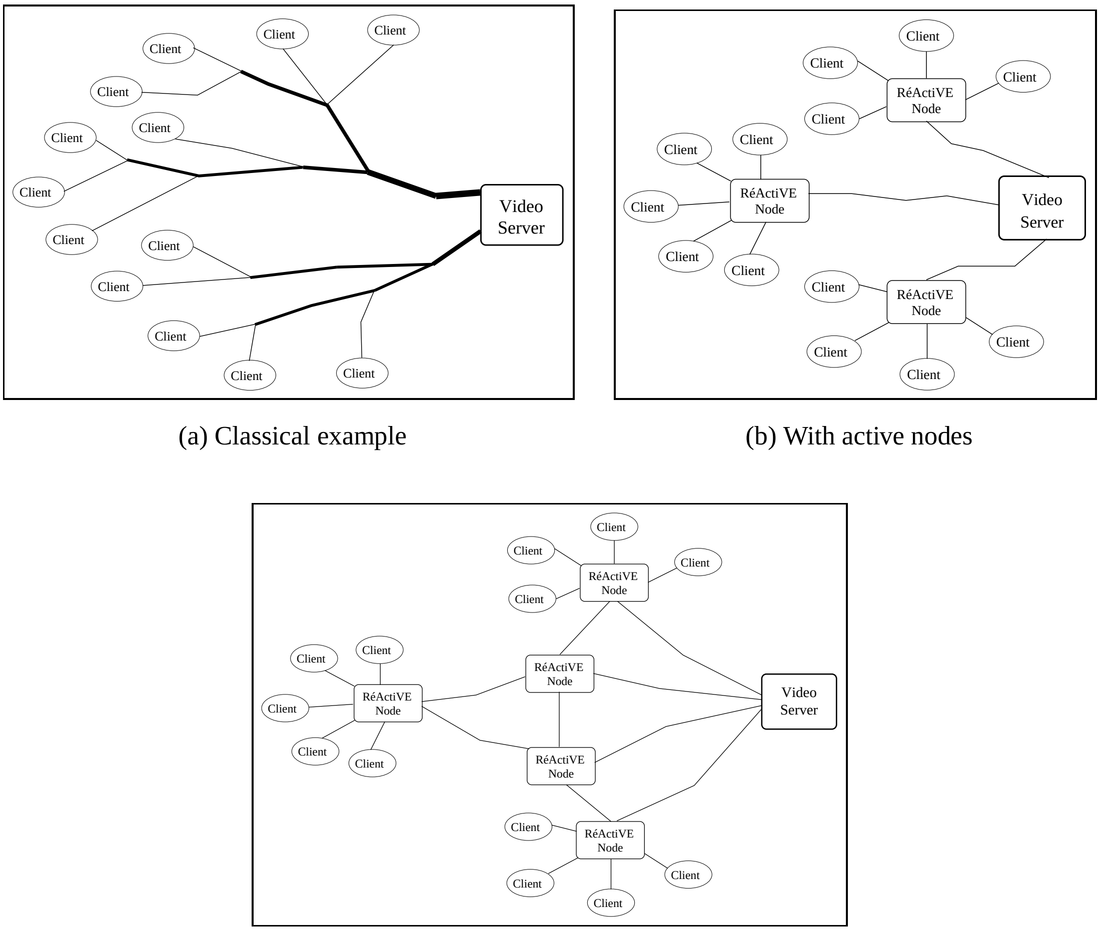
Let us imagine a simple service of video on demand [13, 12, 15]. The service does not have interest (competitively to the classical television, if we think about the digital television and the multiple channels offered) if it does not provide to the user the possibility to pause or to browse in the stream. It implies to associate with a stream one and only one customer. We understand that the generalization of video streaming on Internet would require to have a very important bandwidth even on the backbones and overshoots the present technologies. It is therefore necessary to reduce the bandwidth usage [7].

The objective of this article is to present the hardware and software architecture hold to design an active network [4] for multimedia streaming. At first, we present the global architecture of the designed network (section 2) before presenting the software architecture on which is based the design of the platform and more especially the active nodes used to support the architecture of the project (section 3). We conclude this article on the work we have already done and the possibilities to continue it.

2 Hardware architecture

Starting from the fact that it is necessary to reduce streams circulation on the intermediate networks, we propose in this article an architecture based on intermediate nodes (active routers) with a local storage for the more streamed video contents [11] as an “inside the network” peer to peer architecture.

The circulation of a video content between a video-on-demand server and multiple customers that are visioning the same movie simultaneously requires many



(a) Classical example

(b) With active nodes

(c) With meshed active nodes

Figure 1: Streaming of video contents.

transmission channels: a channel per customer. At the end, we risk to create a network congestion [10, 1]. In figure 1(a), we show how the creation of many streaming channels between customers and the video-on-demand server can clutter the bandwidth (representing the ways used by video streams. Our main hypothesis relies on the fact that often many customers ask for the same video content but at different times of its visualization and want some interactivity on the viewed content, thus a channel by customer is needed.

Our solution to face theses congestion troubles due to the multiple demands of a same movie consists in reducing the path between the customer and the video server. For this, we place intermediate nodes in the used network to store a part or the totality of the asked content. These nodes are placed closed to the customers, for them they play an access node role, so that the path between the storage's place of the movie and the customer is reduced to the strict minimum. Figures 1(b) and 1(c) present the setting up of these ReActiVE nodes within the network and demonstrate the contribution to the solution. This approach consisting in placing some active programs within the network come from the active network domain [19, 18].

Nevertheless, the solution remains incomplete because several channels continue to exist between ReActiVE nodes and the video server (figure 1(b)). The next step consists in using multicast technology [6, 14] to factorize the multimedia streams to be transmitted on the backbone [3]. Indeed, when one of the nodes asks for a movie, it is sent on a multicast channel on which the set of nodes interested in this content can register to get the stream and store it locally. The first customer that asks for a movie at a node initiates indirectly the retrieval from the video server so that the following customers take advantage from the content proximity on their access node.

Several ReActiVE nodes can be placed between the server or the video source and the final node at which the customers are connected. This final node supplies access to the ReActiVE network that is in charge to answer the demands of customers. ReActiVE nodes are going to have potentially three roles:

- offer the possibility to the customers to reach the ReActiVE network, therefore it acts as an interface for customers (access supplier);
- a central role of interconnection with the other ReActiVE nodes and management of available content on a node (caching policy, etc.);
- a role of interface with the video source (a video source may be a server of video content such as a television channel, a web cam or any other element able to provide video content after digitization in MPEG format).

The global architecture of the ReActiVE network is described by the figure 2. We have video sources that are connected to ReActiVE nodes and that permit to inject content in the ReActiVE network. Customers can join some nodes strategically placed closed to them. The set of ReActiVE nodes makes a mesh in which some have only a role of interconnection, used to make the link between several nodes and to store some content, always in the goal to reduce the distance between contents and customers. The placement strategy of the nodes relies on

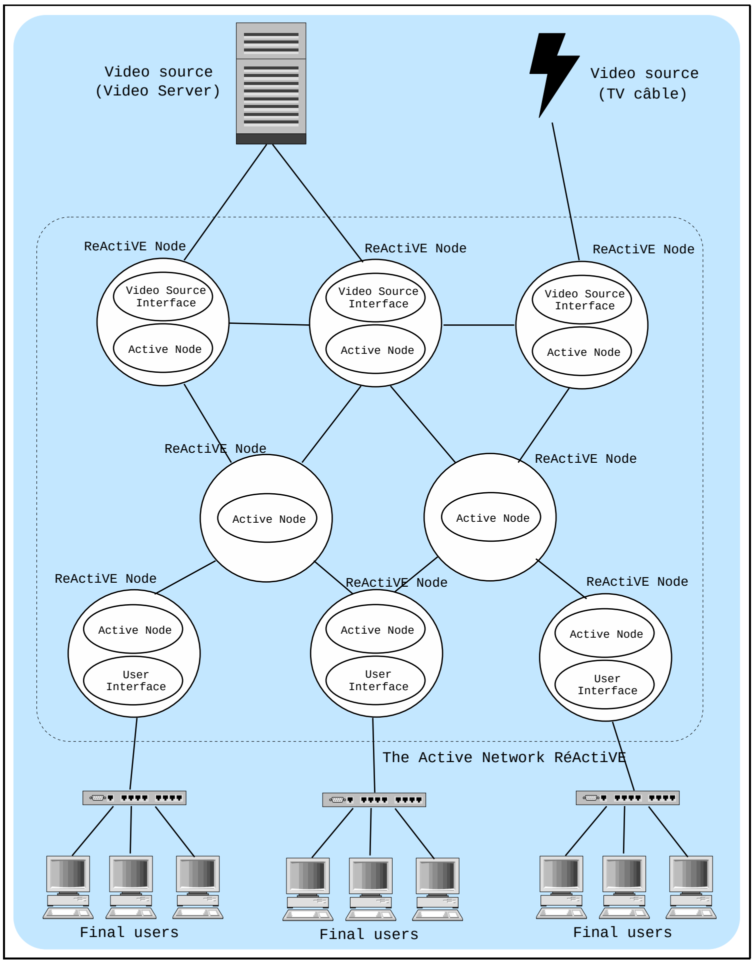


Figure 2: ReActiVE architecture with its differents roles

two aspects: the first one consists in distributing access nodes where the users are and the second one in placing some intermediate nodes to interconnect the access nodes in order to form a reasonable lattice. The interconnection nodes also contribute the more to the storage the closer they are to the users because the storage capacity of the access nodes is not unbound. Therefore, they are used to optimise storage nodes capacities in the most adequate way while storing the content that interest customers. For this last aspect, we must use algorithms based on a notion of interest in the content in relation to users for the hidden caching policies.

3 Software architecture of the nodes

The three different roles played by ReActiVE nodes are directly translated in the software functionalities. In this section, we focus on the active aspect that constitutes the central role of these nodes The proposed model is based on 6-tiers architecture that we are going to describe in this section.

3.1 The proposed model

In the network area, the layer model (OSI - Open System Interconnection) is the reference [5, 17, 2] and is interesting because every protocol layer can be described, specified, implemented and used at its level by using only the described and specified services from its closest layer below. This classical model has proved itself useful for many years in the data processing and network area (opening, standardization, good development properties, etc.).

If this layer model can not be used in the system that we describe, an interesting parallel can be done with this model, and an analogous approach can be led with a slightly different perspective. Nevertheless, it is to notice that the layer model OSI is quite valid in the perspective of layers networks which takes the global system as a basis, but offer little interest to the look of the connection policies dawn to not generalized in the proposed architecture network.

The figure 3 presents the approach kept in the setting of the methodology of design of the ReActiVE software, and identify six layers.

3.2 Layer 6 : management of flowing strategies

This layer is the higher layer level since is deals with the notion of streaming strategy, that is the service given to the user. It takes into account the set of constraints on the network and transforms a user request in distributed streaming events. A strategy of diffusion is like a task scheduling process whose tasks are data streaming. A diffusion strategy may be for example to broadcast the first third of a content from the cache stored in the local node, while the second third will be meanwhile stored locally from another node and the third of the multimedia content will be streamed directly from a third node.

This layer permits to maintain at any moment the set of the current streaming strategies on all the network and therefore to offer the context allowing to build new optimized diffusion strategies to answer new user requests.

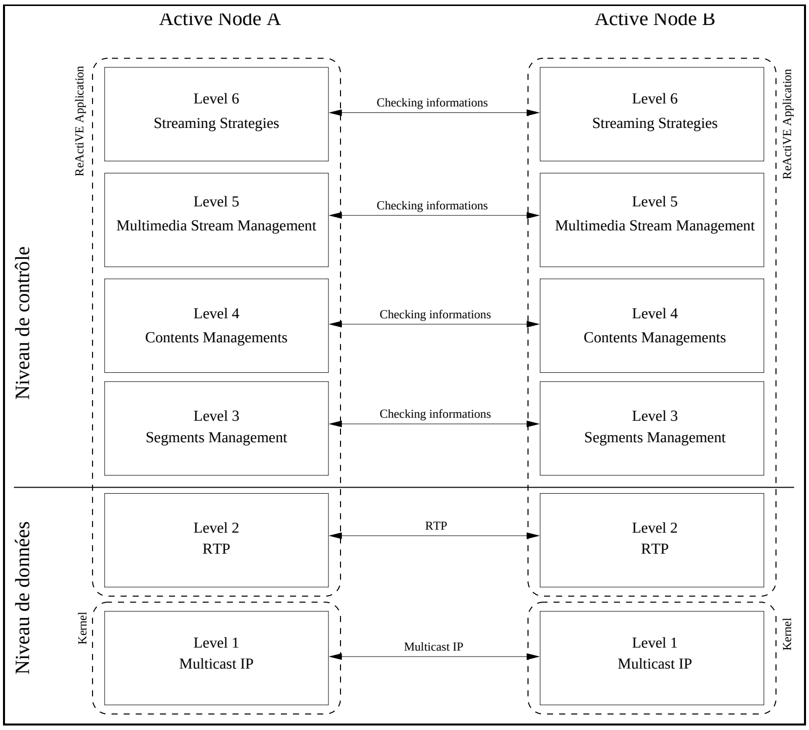


Figure 3: Layers model

Like all layers at the control level (from level 3 up to 6), the layer 6 is a communicating layer. Therefore it looks like a distributed layer on the set of active nodes, that means that every node cooperates with the set of the other nodes to offer the services of this layer.

3.3 Layer 5 : data flow management

This layer manages all the content streams in all the network. Since the multicast is heavily used, a stream is represented like a multicast tree with a source node and a set of recipients nodes. This notion of stream deals with a part of the multimedia content, that is the model of sending information, in a continuous mode (streaming) from a source node toward a set of nodes recipients and is at a higher level than the stream notion in the network area.

The fifth layer register the list of all the content streams on all the network every time. It also manages allocation problems about multicast channels such as unique multicast IP group address or port number across all the network.

3.4 Layer 4 : contents management

This layer offers content management services across the network. It maintains a valid state of the contents list and their availability in every network node at any time. XML messages are used to inform the other nodes about the available contents. In a same way, a demand of suppression or addition of content can be transmitted to all the connected nodes.

3.5 Layer 3 : segments management

This layer introduces the notion of content segmentation. Its main interest is to represent, in an identical way, video contents coming from movies or those coming from live digitalization such as a TV channel. Every segment is number sequentially. All the segments are split in RTP packet for transmission between the nodes and reassembled on the target nodes.

3.6 Layer 2 : RTP

This layer deals with the multimedia data stream using RTP (*Real Time Transport Protocol*), more suitable for non-reliable and voluminous real time data transport [16] by ordering packets and avoiding duplication.

This layer is the low level implementation and more classical streaming view of the layer 5 higher level stream representation.

The RTP packets have a payload identifier to describe the type of data (MPEG-2 video or audio,...), a sequence number to ensure the packet ordering and a time-stamp for just in time play-back.

Encoding and decoding, receiving and broadcasting of RTP streams are managed at this level using the multicast services of the layer below to offer a segment interface to the layer above.

Since RTP is a natively non-reliable UDP-based protocol and we may also use it to transfer information to the cache of some other ReActiVE nodes interested in the streamed content, we also use in this case a protocol to send again the lacking packets.

3.7 Layer 1 : multicast IP

The ReActiVE application heavily rely on IP multicasting that is implemented a this level.

Whereas the previous layers are implemented in user land by the ReActiVE application, this layer is indeed directly implemented within the GNU/Linux operating system with the help of the `mrouted` routing engine dealing with DVMRP.

Having the network layer independent of the ReActiVE node application improves the reliability of the system because, even if upper layers fail, the multicast infrastructure will go on for the other nodes.

3.8 Network management

This layer is a little bit special since it is shown on figure 3. It goes across all the different identified layers and that's why it is used by most other layers. It maintains in every node at all times the global state of the network, the list of the working active nodes, the network topology, the activity on each link, etc.

3.9 Summary

Whereas layers 1 and 2 deal with data, layers 3 to 6 are in charge of the control.

The layer model is analogous to the OSI model since every layer relies on some services offered by the layer immediately below. However, one of the major features of this model is that all layers from 3 and above are distributed, communicating and synchronized on the set of the active nodes connected at a given time.

Whereas the active network technology invalidates or at least fades the OSI model (it places all network functionalities at the applicative level), our model introduced a new view of a layer model very similar but that details the active network functionalities.

4 Conclusion

4.1 Present

We have presented a network architecture dedicated to the diffusion of multimedia contents on IP network. Our solution is based on the contents distribution in order to bring them closer to the users and we use multicast techniques to factorize similar flows. Our approach consists in placing programs and storage communicating in a peer-to-peer way within crossroads of the Internet roads in order to be able to control the multimedia data flows. ReActiVE is therefore a software architecture (active programs managed in an out-of-band way) using the present network infrastructures. Computers (active nodes) are deployed through the network to run these programs and to optimize the multimedia stream control.

A platform prototype has been developed to test the architecture presented in this article. This platform has been developed mainly in java (for the active node part) and in C/C++ (for the streaming video part) and use public domain software (VideoLan, mpgtx,...). Experimentations have been led on a computer network interconnected by 100-Mbit/s links and have shown the validity of our approach. Nevertheless more experimentations on a larger network with 340 computers using the student house at ENSTBr are in development.

This platform has also been deployed at France Telecom Research & Development in the frame of a research contract.

4.2 Perspectives

Several directions are possible to go on this work.

4.2.1 Infrastructure

A first track to explore consists in doing a quantitative assessment of performances of such an architecture deployment. It would be interesting to evaluate the gain in term of network resource economy, in relation with an classical client-server architecture, on a given topology. It could be made while comparing the performance of the ReActiVE architecture with the performance gotten by a classic architecture, on the same topology, and to value the bandwidth economy and therefore the global gain on the whole network.

4.2.2 New services

A second track, based of the present software platform, is to develop new services with higher value for the user by using the node computation power such as real-time program grids, program mosaic to visualize the set of available contents by users, implicit video recorder, TV channel digitalizing, video security services,...

4.2.3 Optimizations

Many parts of the application are still to be optimized to get better performance. Most of the low level Java code has to be moved out of the application down to the VideoLAN server or the kernel. The algorithms used in the management of the strategies of diffusion and caching of contents according to criteria such as the behaviour of users connected in a particular region (statistics on the type of movies often asked,...) are to be refined too.

4.3 Acknowledgements

Authors would like to thank people who have contributed to the project: Yerom-David BROMBERG, José Antonio DE LA FUENTE, François FERRAND, Sylvain GUÉRIN, Zuzanna JANIAK, Sylvia JIMÉNEZ, Didier MENUT, Jean-François MERCEL et Benoit PECCATTE.

References

[1] A. Charny, D. Clark, and R. Jain. Congestion control with explicit rate indication. *In Proceedings of IEEE International Conference on Communications*, pages 1954–1963, 1995.

[2] Douglas Comer. *Internetworking with TCP/IP*. Prentice Hall, 1988.

[3] Asit Dan, Dinkar Sitaram, and Perwez Shahabuddin. Scheduling Policies for an On-Demand Video Server with Batching. In *ACM Multimedia*, pages 15–23, 1994.

[4] DARPA Defense Advanced Research Projects Agency. Active Network Mission. Technical report, DARPA, 1997-2001.

[5] J. D. Day and H. Zimmerman. The OSI Reference Model. In *Proceedings of the IEEE*, volume 71, pages 1334–1340, 1983.

[6] S. Deering. Host Extensions for IP Multicasting. RFC 1112, Août 1989.

[7] Derek L. Eager, Mary K. Vernon, and John Zahorjan. Minimizing Bandwidth Requirements for On-Demand Data Delivery. *Knowledge and Data Engineering*, 13(5):742–757, 2001.

[8] Roy Friedman, Lior Bram, and Shiri Abarbanel. Fault-Tolerant Multi-Server Video-On-demand Service. In *International Parallel and Distributed Processing Symposium*, April 2003.

[9] Leana Golubchik, Richard R. Muntz, Cheng-Fu Chou, and Steven Berson. Design of Fault-Tolerant Large-Scale VOD Seervers: With Emphasis on High-Performance and Low-Cost. *IEEE Transactions on Parallel and Distributed Systems*, 12(4):363–386, April 2001.

[10] Raj Jain, K. K. Ramakrishnan, and Dah-Ming Chiu. Congestion avoidance in computer networks with a connectionless network layer. *Proceedings of the IEEE Computer Networking Symposium. Washington, DC*, pages 134–143, 1988.

[11] Vana Kalogeraki, Alex Delis, and Dimitrios Gunopulos. Peer-to-Peer Architectures for Scalable, Efficient and Reliable Media Services. In *International Parallel and Distributed Processing Symposium*, 2003.

[12] Victor O.K. Li and Wanjiiun Liao. *Handbook of Internet and Multimedia*, chapter Video-On-Demand Systems, pages 511–529. CRC Press LLC, 1999.

[13] Xue Li, Mostapha H. Ammar, and Sanjoy Paul. Video Multicast over the Internet. *IEEE Network Magazine*, April 1999.

[14] K. Obraczka. Multicast transport protocols: a survey and taxonomy. *IEEE Communications Magazine*, 36(1):94–102, January 1998.

[15] Juan J. Sanchez, Javier Mosquera Victor M. Gulias, and Alberto Valderruten. State of the art and design of vod systems. In *Proceeding of SCI'2000*, Orlando, USA, July 2000.

[16] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889, March 2001.

[17] A.S. Tanenbaum. *Computer Networks*. Prentice Hall, 1996.

[18] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G.J. Minden. A Survey of Active Network Research. *IEEE Communications Magazine*, 35(1):80–86, 1997.

[19] David L. Tennenhouse and David J. Wetherall. Towards an Active Network Architecture. *Journal of Computer Communication Review*, 26(2), 1996.