

A Sustainable Framework for Multimedia Data Streaming

Alessandro Bassi², Jean-Patrick Gelas¹, and Laurent Lefèvre¹

¹ LIP INRIA RESO, Ecole Normale Supérieure de Lyon
46, allée d'Italie, 69364 Lyon Cedex 07, France

`laurent.lefevre@inria.fr`, `Jean-Patrick.Gelas@ens-lyon.fr`

² LoCI Laboratory - University of Tennessee
203 Claxton Building - 37996-3450 Knoxville, TN, USA
`abassi@cs.utk.edu`

Abstract. Streaming is the ability to use multimedia data while they are being sent over networks. We propose a realistic and sustainable approach called Active Logistical Cache as a framework to support an efficient caching of multimedia streaming inside the Internet. The approach is based on active networking support (dynamicity, new services and protocols) merged with logistical storage. The tools proposed are the Tamanoir¹ execution environment combined with the Internet Backplane Protocol middleware. By describing a scenario, we illustrate the realistic potential of this approach.

1 Introduction

It is easily predictable that in a near future multimedia streaming will play a major role in consuming a huge share of Internet resources, as it already does it today; and as it does not seem likely a future limitation of this kind of traffic on the commodity Internet, but instead an exponential growth, solutions beyond a blind faith in the Moore's law might preserve ourselves and younger generations to catastrophic traffic congestions. Unfortunately, we can notice that the remedies we see today proposed by the scientific community are not implemented on the mainstream equipment market, and this picture seems to not likely change in the near future.

Currently, the limiting factors of a scalable growth of multimedia streaming over the Internet are:

- Fixed protocols : streaming applications designers must choose network transport protocols (UDP, TCP, RTP...). This choice often does not fit with local and global configurations, such as firewalls, dropping UDP streams, or being TCP not adapted to high performance networks.

¹ This work is supported by the French RNTL Etoile project (<http://www.urec.cnrs.fr/etoile>).

- Proxy and mirrors : these equipments, commonly used in Web traffic, are not natively adapted to multimedia stream caching.
- Network services : multimedia streams rely on services seldom deployed like multicast. To bypass these limitations, stream applications must implement personalized solutions.

Another factor preventing the wide diffusion of clients for multimedia streaming can be identified in the dominion of proprietary solutions for streaming, from companies such as RealNetworks, Microsoft, and Apple; as their standards are not compatible between each other, consumers are forced to download and run more than one dedicated clients to ensure the capability of receiving any stream.

With this article we would like to introduce a framework, based on a combination of an high-performance active networking environment (Tamanoir [8]), providing the necessary dynamism for the deployment of new services, and the logistical networking storage middleware (Internet Backplane Protocol [2]), allowing data to be temporarily stored in the network and transferred efficiently between endpoints.

The Tamanoir Execution Environment is an active software dedicated to deploy and maintain personalized network services for streams transport in an efficient way. Its main achievement is the ability to deal with some of the historical problems that this kind of approach has to face, such as security and high performance, implementing an efficient multi-streams active transport and dynamic services deployment in the network. Tamanoir has an original and unexplored approach with regards to on-the-fly storage in the network, as it takes advantage of the Internet Backplane Protocol (IBP) suite to manage the ability to store data in the network. IBP is designed to allow applications to use shared storage resources in an exposed way. IBP is the corner stone of Logistical Networking, where the term logistical comes from the similarity with the industrial and military world, in which depots are used to store goods (instead of data) in transit.

This paper is organized as follows: after a brief description of the reasons of the sustainability of this approach in section 2, we provide a definition and the architecture of the Active Logistical Cache and their composing parts in the section 3; section 4 focuses on the basic Multimedia services the Tamanoir-IBP marriage is able to offer to a potential application, section 5 describes a quite common scenario, and how can various problems be addressed with our solution. After briefly talking about related projects in section 6, we present our conclusions and future works.

2 Sustaining the Visions

In the recent years many cache systems, with almost any level of intelligence, activeness, responsiveness, have been studied and proposed to the scientific community. We feel that our role is not to argue about how good or bad any of these approaches are. We actually believe that many, if not most of them, are based

on solid theory basis, and their widespread adoption could be very beneficial to the entire community.

Our focus is slightly different. Looking at all this theoretical and practical effort, a reader could ask himself why any of those systems are not adopted widely. This main question is not if those systems are scalable as they should, but instead (and much more pragmatically) if those systems are *sustainable*.

In this context, by *sustainable* we mean a solution that can be deployed and managed world-wide in an easy way.

Why do we feel that the work we are presenting has better chances than any other one, and why do we feel that it is sustainable ? Mainly because what we are proposing is a *framework* to build active and intelligent caches distributed in the network. Logistical (IBP) and Active (Tamanoir) Networking are two independent general-purpose projects, complementing each other perfectly, whose success or failure will not depend solely on multimedia stream treatment. IBP middleware is used already in many different areas, from basic services like check-pointing or data staging, to more complex ones such as overlay routing or massive data transfers. IBP depots have already been deployed in around 300 different sites around the whole world, offering a publicly available storage space of around twenty Terabytes. The IBP traffic monitored by Abilene shows that this protocol is used for everyday operations. Tamanoir is an active networking execution environment, allowing any data stream to receive differentiated treatment according to the service linked to it. While the core team is focused on the development of the execution environment, many on-going projects are developing different services to fit their particular research needs, from reliable multicast to programmable transport protocols for scientific grid projects.

Their deployment follows a different strategy also, as IBP depots tend to be positioned as close as possible to very well-connected nodes in the core network topology, offering storage services in places easy to reach, while Tamanoir Active Nodes lie more on the network periphery, between the core network and the clients, to optimize the transport performance. Both network storage and active services can be, if they are not already, top items in a service (or infrastructure) provider agenda, giving a chance to this particular environments to be widely adopted.

A typical ISP can be interested in deploying this two frameworks in order to offer to his customers additional services, like the capacity of storing huge quantities of data for a certain period of time, or provide opportunistic encryption between two sensitive network points.

In this paper, we are proposing a framework to multimedia streaming application developers, to let them implement new solutions, such as active caches schemes or protocols. ALCs will give them the possibility of using and managing a storage space of several Terabytes distributed worldwide, and to use an high performance execution environment to develop new active services. Or, in other words, a mean for sustaining their visions.

3 The Active Logistical Cache Framework

3.1 Basic Definition

An Active Logistical Cache (ALC) framework is a network element composed by a Tamanoir Active Node and an IBP depot. During streaming transport, any crossed ALC can dynamically store streams on the fly not only on the crossed node, but on any convenient depot, and for any stream size. ALC can cache various kind of streams :

- Stored streams : considered like files with possible missing parts, streams can be dynamically and entirely stored in ALCs;
- Live Internet Television or Radio : to allow new users to join a multicast group and to apply on the fly services on multimedia data, only a temporal part of the stream is stored in the ALC (variable window);
- VoD streams : to allow asynchronous users to join a video broadcast session by sending them earlier part of the data, stream movies can be stored since the beginning of the broadcast.

Multimedia stream services can be efficiently and dynamically transported and deployed thanks to the Tamanoir Active Network infrastructure. Therefore, services such as store and forward, dynamic compression of video streams, multicast, and video transcoding can be easily applied by any ALC.

3.2 The Logistical Networking Infrastructure

IBP has been developed to allow the sharing of distributed storage resources, scaling to any size in all the dimensions involved, which are the aggregate storage space, the number of separate depots, and the community size. IBP gives to any application the possibility of allocating a discrete amount of space for a discrete amount of time, on a specific server. The mechanism and his philosophy have been clearly identified, but the protocol itself has not yet been standardized, although there are plans to submit its specifications in the very near future to the Global Grid Forum and to the Internet Engineering Task Force.

The IBP software has been successfully tested for different Oses and hardware architectures (such as Linux on i686 and ia64, Solaris, Windows 2000, AIX, DEC alpha, OS X), and its current version is 1.3.1.

IBP depots publicly available through the Logistical Backbone are currently around 150, deployed in around 60 different sites in the whole world, with a total aggregate storage space of around 10 Terabytes. IBP has also been adopted by projects such as 6Net, an European IST project, and, together with Tamanoir, by e-Toile, a French-scale Grid project.

3.3 Tamanoir : High Performance Active Networking

The integration of new and standard technologies into the shared network infrastructure has become a challenging task, and the growing interest in the active

networking field [13] might be seen as a natural consequence. In “active” networking vision, routers or any network equipments (like gateway or proxy) within the network can perform computations on user data in transit, and end users can modify the behavior of the network by supplying programs, called *services*, that perform these computations.

The Tamanoir architecture design does not interfere with the core network, mainly to guarantee higher performance results, and it’s deployed only on the network periphery.

The injection of new functionalities, called services, is independent from the data stream: services are deployed on demand when streams reach an active node which does not hold the required service. New services can be downloaded, when required, either from a service repository or, from the last Tamanoir Active Node crossed by the data stream.

For the implementation process of the Tamanoir execution environment we choose the JAVA language, because it provides great flexibility, shipped with standard library and support dynamic class loading. Each new service is inherited from a generic class, called *Service* which is a Java *Thread*. Performances achieved with light services, which do not require a lot of processing power, are adapted to current networks requirements. As an example a Tamanoir Active Node is able to support, over local and wide area Gbits Link, from 450 Mbps to 1.6Gbits (depending on number of processing unit, network technology and number of active data streams).

3.4 Architecture

Both Tamanoir Active Node (TAN) and IBP depot are two independent processes communicating through a socket over a reliable transport protocol (TCP). (see Figure 1). As an ALC generally access a local depot, tools on a higher level of the Logistical Networking stack such as the exNode tools or the full Logistical Run-Time System suite are not necessary, and the active node can communicate directly with the IBP depot using the IBP API.

Each service willing to use logistical storage has to instantiate its own IBP client classes in order to communicate with an IBP depot. These classes provide constructors and methods to create *Capabilities* (cryptographically secure URLs to IBP allocations) on any IBP depot, capabilities with whom the Tamanoir service can write, read and manage data remotely on the IBP depot.

We also measure the overhead introduced by the caching action in an IBP depot close to the Tamanoir Active Node. For these results we ran an IBP depot on the same node than the TAN.

Our testbed, disconnected from the production network, is set up with one active Tamanoir node which is a PC (Dual-Pentium III, 1 Ghz, 256MB RAM) shipped with several *Ethernet 100Mbps* network interface cards and a standard IDE hard-drive. We link to this PC different other PC who have the role to either feed the network or receive data. These PCs run under GNU/Linux with a 2.4.16 kernel (distribution *Debian 2.2r3*).

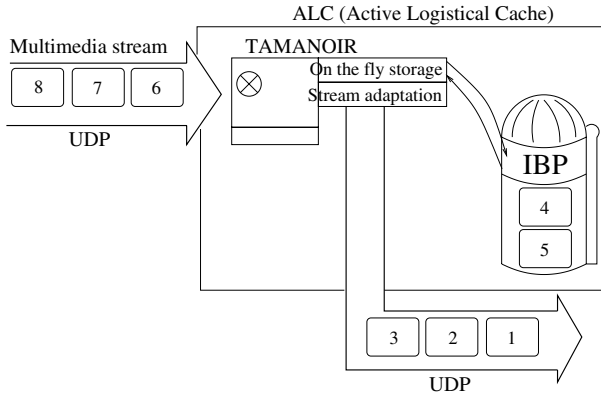


Fig. 1. Inside an Active Logistical Cache : Tamanoir and IBP frameworks for a multimedia stream adaptation service

When an ANEP packet reaches the TAN, its payload is extracted and sent towards the required service. At the beginning, the service copies the payload in the local IBP depot. For the measures we have set up only one client which consumes data as fast as possible. Next, the payload is read in the cache and forwarded to the client immediately. For each packet, we measure the time to cross a TAN (latency) with and without caching action. A chronometer is located as low as possible in the kernel. We use *Netfilter* [12] which is a package for filtering in the Linux kernel. NetFilter allows us to introduce timer code just behind the network interface card. When an ANEP packet reaches the TAN, our NetFilter module analyzes the packet, and the timer is started. When the same ANEP packet leaves the TAN, the timer is stopped.

In figure 2 we present the latency of a TAN by an ANEP packet. In the first case (lower curve) packets are immediately forwarded and not cached in the IBP depot, in the second case packets are cached and next forwarded.

As shown, caching doesn't introduce a very important overhead. For any packet size, overhead remains constant. For small packets size, performances are weak due too the policy of transmission of Linux kernel TCP implementation which tries to aggregate small packets before being transmitted. So, small packets are released just after a timeout.

4 The ALC Service Tool-Box for Data Streaming and Multimedia

Tamanoir offers various network and high level services to multimedia and data streaming applications. These services, dynamically deployed, allow users and operators to manage multimedia streams with active logistical caches efficiently.

ALC active available services can be classified into various categories : transport services (multi-protocols, multicast, QoS...), network services (content

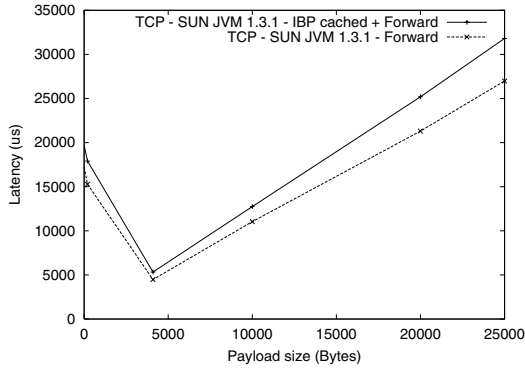


Fig. 2. Overhead introduced by caching data in a local IBP depot

based routing, dynamic network management, monitoring...) and stream services (transcoding, compression on the fly, multi-codes...).

Moreover, users and applications designers can deploy their own personalized services inside an ALC. We focus this section on few services dedicated to active cache and open a discussion about the ones under development.

4.1 Video Adaptation

Today's networks have to deal with three kind of heterogeneity. The first one is the *data* heterogeneity, which is depending by the applications requirements in terms of throughput, synchronism, jitter, error injected during transport, and so on. The second one can be identified in the *physical transport media* heterogeneity which, having different throughput characteristic, may limit the reliability and therefore introduce asymmetry. The last heterogeneity class comes from different *clients terminals* (PDA, cell phone, desktop, TV, ...), having different processing power and ability to restore informations. Active Networks give us the opportunity to create and/or adapt transport protocols and configure it dynamically for a targeted type of stream or an application. They also give the capacity to process the data stream to adapt it for the client's processing needs, its type of connection and its application requirements (see Fig. 1).

Services must be efficient and must guarantee a high performance to be able to sustain the bandwidth throughput without introducing too much latency. To achieve good results, we must use appropriate data structures for an efficient data processing (using hierarchical coding like MPEG-4 for video transfer) A service embedded in the TAN is able to reduce the quantity of informations just by dropping the surplus of data unused by light terminal at a very light cost (Fig. 1).

4.2 Multi-protocols: UDP – TCP

Using UDP or TCP protocols on a controlled “load free” local area network does not show spectacular differences in term of throughput and latency. But in a long distance context, even if some packets are lost with the UDP protocol, it is pretty difficult to use TCP for time constraint applications. TANs can be used to change the transport protocol during the packet’s journey, as they are deployed in the access layer. As an example, an application might choose to use any transport protocol in the first part of the transfer, in a campus or a company for example, until packets cross the TAN which will put the data in a user datagram (UDP) for traveling the long distance part through the network; and, finally, when packets reach the other side of the network, another TAN might change the protocol transport if requested, and send it towards the end nodes. In this context, TANs can be seen as gateway in charge to change the transport protocol in function of the application requirements. In this case TANs should manage the connections. One more time the Active Networking concept gives the opportunity to achieve better performance by injecting code and processing power into the network equipments.

5 A Possible Scenario

5.1 Multicast Today

Nowadays, when applications need to send data to many receivers they are have two possibilities to choose from: repeated transmissions or multicast (see Fig.3). Classic applications like the web or ftp have a client-server approach, which just needs a point-to-point protocol. But applications like audio and video broadcasting, information push or distributed virtual world obviously require point-to-multipoint communications, and this kind of communications does not seem to be well deployed on the current Internet [7] infrastructure.

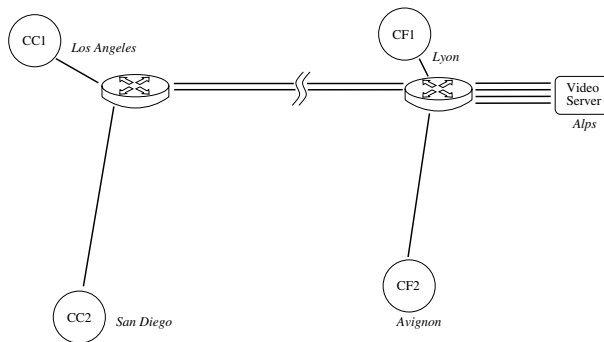


Fig. 3. A current scenario without multicast for multimedia streams

Although born as the same epoch (1990) than the web (which represents today more than 80% of all Internet traffic) and universally perceived as useful, multicast (class D of IP) is still not large-scale deployed and not proposed by a majority of ISP (Internet Service Provider).

Unfortunately, the original IP multicast is based on an *open* service model which was not designed with a robust strategy or in function of clear commercial requirements. For example, there are no restrictions for users to create a multicast group, or to receive from and send data to a multicast group. There is also a lack of simple and scalable mechanisms for supporting: access control (group creation and membership), security (routing, data integrity), address allocation, network management (tools) and finally, multicast datagram are like IP datagram: best-effort and unreliable.

This is why, even though multicast is shipped with most commercial routers, most IP carriers have not yet enabled the service in the networks.

5.2 The Active Logistical Networking Solution

We would like to introduce a scenario to illustrate which services an active logistical caching architecture can provide in multimedia streaming delivery. We have a server producing video stream on the Alps nearby Lyon, France, and connected with a radio connection to a router in our lab, with two clients (CF1 and CF2) located in Lyon (with a ADSL connection to our Active Node) and in Avignon, France (with a 56K modem), and two clients (CC1 and CC2) located respectively in San Diego, Ca, and Los Angeles, Ca., one with an ISDN and the other with a T3 connection (see Fig. 4). For the purpose of this scenario, we will only add an Active Node in Los Angeles, while all the remaining infrastructure is already in place, and therefore we will use their real URL names in this example.

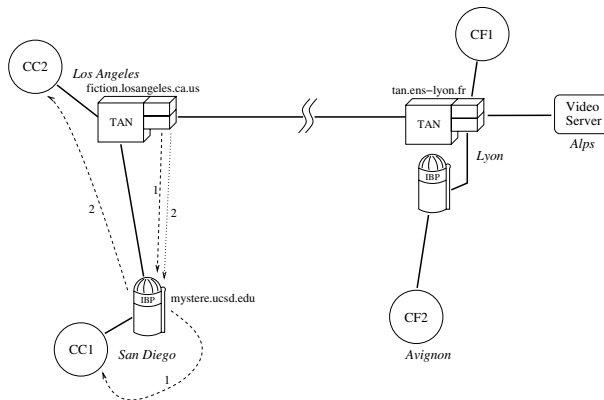


Fig. 4. Active networking and cache scenario

At a first glance, we can observe that, while for performance reasons the optimal solution would be to have the TAN and the IBP depot on the same machine (such as `tan.ens-lyon.fr`), our architecture works equally as well when the IBP depot is sitting on a remote machine (which is the case of California, where the TAN is installed in Los Angeles, and the IBP depot in San Diego). In other words, the deployment of TANs and of IBP depots can be done (as it is) independently and in parallel, and it is possible to reuse public depots not specifically dedicated to work with a TAN and already deployed in the Internet.

Moreover, if the clients are connecting at a different time, they can get the video stream from the beginning, as it's cached nearby the active node, and the only QoS connection they are depending upon is the one between the TAN and their machine. This is extremely important in case the first connection, as in our scenario, is particularly bad: if a packet is lost at the beginning of its journey, no client would be able to get it. In environments like the one described an application could use the Active Node to switch between TCP and UDP according to the importance of the link in the global architecture, the underlying network connections, real-time traffic considerations, QoS demands and so on.

Using distributed storage servers is also fundamental in relieving the network of having the same packets sent over and over the same wires to satisfy clients asynchronous needs; so, when a client (*CC2*, for instance) contacts the TAN in Los Angeles for the video stream, the TAN redirects its request to the San Diego depot holding the video; and, in case it's the first client with this particular request, the TAN forwards the request to the TAN running in Lyon, having the stream starting from Lyon, and not from the original source. As the bandwidth between Lyon and Los Angeles might be different from the "last mile one" between the Client and the TAN, as it is in reality, data can be buffered and cached in the nearest IBP depot, the San Diego one, using a classical store-and-forward service; and the IBP built-in fast data mover module and other facilities such as the *exNode* could be used for handling the data transfer between depots.

6 Related Works

IBP has often been compared with many network storage systems and technologies, such as *FreeNet* [6], *OceanStore* [10], *AFS* [9], even *NFS*, although with a different storage philosophy, making our research more similar to works carried for *Storage Area Network* in local networks. IBP can also show some similarity with projects coming from the scientific Grid community, such as *GASS* [3] and the *SDSC Storage Resource Broker* [1].

Active networks projects using storage facilities are not intensively addressed. The most related projects are *Active Disks* [11] or *active-network-storage* [14] which explore the increasing of intelligence in *Network Attached Storage*. Our active logistical cache is more related with [4] and [5] by adding logistical and active networks functionalities. The main distinction between *ALC* presented in this paper and that being undertaken in other areas is that this proposed

architecture is low impact, efficient and easily adaptable for each stream or application.

7 Conclusion and Future Works

Although a particular implementation might be sub-optimal, we firmly believe that without explicit control of *computation* and *storage* resources in the network any caching system for multimedia streaming is doomed. Therefore, merging two environments like IBP and Tamanoir allow us to propose an architecture (named Active Logistical Cache Framework) as a central solution for building efficient streaming algorithms over the Internet.

We also believe that, to keep up with the always growing demand, only execution environments developed with a strong attention to performance, like Tamanoir, can show and prove the potential of Active Networking ideas to a wide public.

Active networks equipment usually do not propose storage services. Meanwhile, distributed storage technologies do not rely on dynamic services. ALC frameworks elements can be installed and deployed inside networks by operators, ISP or campus administrators. ALC services can be proposed and deployed by applications designer who want to efficiently broadcast streams.

Concerning future plans, both projects envisage to deploy more Active Nodes and IBP depots in wide area network, Tamanoir is looking upon the development of new services for Multimedia oriented applications, which make use of more layers of the logistical networking framework, such as the ex-Node, the Logistical Runtime System Tools (LoRS) and the Data Mover. Meanwhile, the IBP project aims to deliver new versions of the Data Mover and of Logistical Runtime System Tools.

References

1. C. Baru, R. Moore, A. Rajasekar, and M. Wan. The SDSC Storage Resource Broker. In *CASCON'98*, Toronto, Canada, 1998.
2. M. Beck, T. Moore, and J. Plank. An end-to-end approach to globally scalable network storage. In *Proceedings of the ACM SIGCOMM 2002 Conference*. SIGCOMM, august 2002.
3. J. Bester, I. Foster, C. Kesselman, J. Tedesco, and S. Tuecke. Gass: A data movement and access service for wide area computing systems. In *Sixth Workshop on I/O in Parallel and Distributed Systems*, may 1999.
4. E. Bommaiaha. Design and implementation of a caching system for streaming media over the internet. In *IEEE Real-Time Technology and Applications Symposium (RTAS)*, Washington D.C., USA, May 2000.
5. P. Cao, J. Zhang, and K. Beach. Active cache: Caching dynamic contents (objects) on the web. In *IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware '98)*, pages 373–388, The Lake District, England, sept 1998.

6. Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In Hannes Federath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, LNCS 2009. Springer: New York, 2001.
7. Christoph Diot, Brian Neil Levine, Bryan Lyles, Hassan Kassem, and Doug Balensiefen. Deployment issues for the ip multicast service and architecture. *IEEE Network*, 14(1):78–88, jan 2000.
8. Jean-Patrick Gelas and Laurent Lefèvre. Mixing high performance and portability for the design of active network framework with java. In *3rd International Workshop on Java for Parallel and Distributed Computing, International Parallel and Distributed Processing Symposium (IPDPS 2001)*, San Fransisco, USA, April 2001.
9. J.H. Morris, M. Satyanarayan, M.H. Conner, J.H. Howard, D.S.H. Rosenthal, and F.D. Smith. Andrew: A Distributed Personal Computing Environment. *Communication of the ACM*, 29(3):184–201, 1986.
10. Sean Rhea, Chris Wells, Patrick Eaton, Dennis Geelsand Ben Zhao, Hakim Weatherspoon, and John Kubiatowicz. Maintenance-free global data storage. *IEEE Internet Computing*, 5(5), september 2001.
11. Erik Riedel. Active disks - remote execution for network-attached storage. Technical Report CMU-CS-99-177, Electrical and Computer Engineering Carnegie Mellon University, Pittsburgh, PA 15213, Nov. 1999.
12. Rusty Russell. Linux Filter Hacking HOWTO. july 2000.
13. David Tennenhouse and David Wetherall. Towards an active network architecture. *Computer Communications Review*, 26(2):5–18, April 1996.
14. A. Tomita, Y. Takamoto, S. Inohara, H. Odawara, F. Maciel, M. Sugie, and N. Watanabe. A Scalable, Cost-Effective, and Flexible Disk System Using High-Performance Embedded-Processors. In *IEEE International Conference on Parallel Processing*, 2000.