# Content Adaptation on LANE Active Network Platform

Marko Lyijynen, Titta Koskinen, Sami Lehtonen, Juuso Pesola
VTT – Technical Research Centre of Finland
Information Technology
P.O. Box 1203, 02044 VTT
{Marko.Lyijynen, Titta.Koskinen, Sami.Lehtonen, Juuso.Pesola}@vtt.fi

***Abstract*—As the number of different content types in the Internet is increasing as well as the number of terminal types, the heterogeneity of both content and terminals has become a problem. A common solution is to convert the content to a suitable format for terminal application either in the server or in the client. More sophisticated solution is to use static proxy-servers in order to adapt the content, in real time and inconspicuously. However, if the terminal is a mobile device, the static proxy-servers are unable to accommodate to changing characteristics of network (i.e. topology, bandwidth, delays etc). Active networking enables the proxy-style adaptation but also meets the special requirements of mobile devices. At VTT, an Active Network (AN) system called LANE was developed. The LANE system introduces content converting active application. With LANE based implementations, it is possible to reach more efficiently located and more transparent adaptation compared to the conventional static proxy servers. Currently LANE introduces a demonstration, in which the Microsoft Word document is converted into other formats.**

***Keywords: Active Networks, Content Adaptation, LANE, Proxy***

## I. Introduction

Currently, there exists a very strong trend towards wireless networking. This means that more of the traffic in the Internet is caused by various wireless terminals, such as PDAs, mobile phones, or laptops with wireless network access. In practice, the key issues are support for mobility, security, and the content adaptation.

This paper studies the content adaptation problem, current solutions to it, and introduces a new active network based solution. The main emphasis is not in the existing content adaptation applications but in the ways that can be used to locate the adaptation optimally. Other publications in VTT's active network research are currently under construction.

In this paper the approach to solve the content adaptation problem is based on the active network technology. In this solution the network performs the adaptation, and thus it does not require any adjustments to terminals nor content servers. Active networking enables to make adaptation more transparently and flexibly than current techniques. In order to proof the theory, a demonstration system that performs conversions for Microsoft Word documents was developed. The demonstration system converts the documents either into Adobe's Portable Document Format (PDF) or in Hypertext format (HTML), according to the customer's personal profile.

## II. Active networks

An active network (**AN**) is a network that is aware of the content flowing through it, thus making it capable to make customized modifications to the data flow. An AN is built from hardware components, like any ordinary network, but a portion of the nodes of the AN contain special software that makes them capable to monitor and modify the data flow. [1]

There are many reasons to use an AN instead of traditional passive packet network. The main benefits are rapid development and especially deployment of new services; new service types are enabled by AN technology and AN enables far more enhanced network management than current networks. The services of AN can be customized to meet subscriber's needs. Furthermore, they can be placed ideally in the network and relocated according to the needs of a mobile user.

Tennenhouse et al. have introduced two active network architectures in their paper [2]. These two ways to reach programmability in networking are **Programmable Switches** and **Capsules** approaches. In the Programmable Switches approach the programming of the network is separated from the common data transfer while in the Capsules model every message is a program.

## III. Content adaptation

Network content (e.g. text documents, videos, and audio) can exist in many different formats. For example, there are several different formats for movie clips, such as RealVideo, Quicktime, MPEG, MVQ etc. In addition, the characteristics of different terminals and their software are diverse, limited, and alternating. For example, one terminal is able to play mp3 music files while some other has less processing power. Thus, it can not decode mp3s but it could be able to play music in a non-compressed wav-format. The compatibility is completely different issue; because of the commercial and technological reasons the consensus of this field is out of reach.

Providing and maintaining network services requires considerable resources. Thus it is not likely that content providers would support separate content versions for some special devices or for narrow band network use [3]. Because maintaining multiple contents concurrently is not a cost-effective way to offer content to all customers, users of the terminals, wireless or wired, are forced to have multiple players, plug-ins etc. installed. The maintenance of multiple applications that are intended to serve the same purpose is always frustrating. Furthermore, in the case of mobile terminals, this could be impossible due to limited storage resources. In addition, for mobile users, the cost of data transfer is a crucial issue that can be reduced by content adaptation (e.g. resizing images etc.).

For the content adaptation in the network, there exists many solutions such as APRO [4], Digestor [5], etc. A connective factor for existing solutions is that they are usually proxy-based and their location in the network is fixed. By using the active network based solutions, the location of the adaptation can be dynamically altered according the circumstances.

## IV. RELATED WORK

### A. *Active networks*

Currently there is number of ongoing research and development projects related to the active networking area. In addition to the VTT's LANE research at least University of Tel-Aviv is developing their ABLE/DINA AN platform, DARPA's RADIOACTIVE project and European Union's ANDROID and CONTEXT projects.

### B. *Content adaptation*

Content adaptation issues are under ongoing work and there are even some efforts of standardization emerging. One of the most viable of such ongoing standardization efforts is a Internet Content Adaptation Protocol (**ICAP**) [6]. ICAP draft is published by IETF.

One interesting research project close to LANE content adaptation is MARCH [7] which develops a content adaptation architecture for mobile terminals. In the MARCH architecture there are Mobile Aware Servers (**MAS**) that are coupled to content servers and Compute Servers (**CS**) that are located all over the network. The functionality of MARCH is divided so that content adaptation is performed in CSs and the management logic of the system is located in MAS. In the MARCH solution a client entity needs to be installed in the client device in order to route applications' traffic to proxies. Therefore, MARCH is less transparent than LANE.

## V. LANE PLATFORM

Lightning Active Node Engine (LANE) is an active network platform developed at VTT (Technical Research Centre of Finland). It is the result of two years work and two active networking projects; Caspian and MAO. Caspian [8] was a project in which an AN prototype and "Mobile Active Mail"- Active Application (**AA**) [9] was produced. The MAO-project was VTT's internal project which aim was to produce a new AN platform from the basis of the outcomes of the Caspian project and active applications for demonstration and testing purposes.

The architecture of the LANE is illustrated in the Figure 1. The figure presents the components and interfaces of LANE. The figure shows that there are two active components; an Active Server (AS) and an Active Router (AR), thus the execution environment and the activity of the network are separated. The benefit of the separation is that the tasks, which require considerable amount of processing power and other resources are handled in the AS, while resources of the AR are conversed, thus keeping the routing delays low. A rule of thumb is that IP-layer actions lay in the AR and actions above are located in the AS.

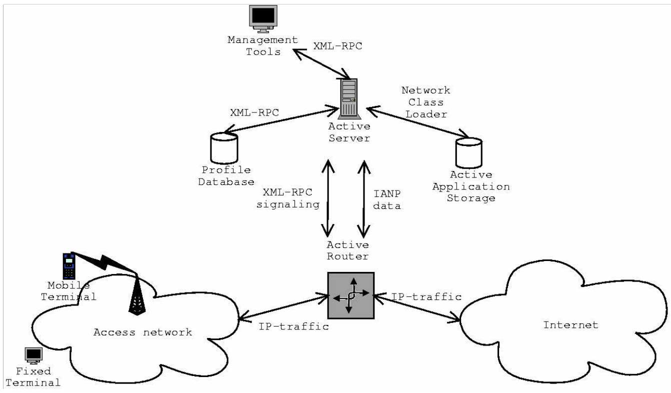


**Figure 1. LANE Architecture**

Node operating system of LANE is Linux in both AS and AR. Active router is a FSR router developed at VTT and AS is just a regular Linux server. In the future, if (and when) LANE is in production use, it is planned that AS can be located in the same physical unit as AR but on a special processing card. An execution environment of the AAs is based on Java. Active server is completely written in Java.

As Tennenhouse et al. has presented two basic AN technologies: Programmable Switches and Capsules. Neither of these is used in LANE but active applications are stored in repositories. Basically an AA could be stored by anyone (e.g. the subscriber, ISP, some third party AA provider, etc.), but the leading idea is that the operator is responsible for the AAs and takes care of AA uploading.

When a user arrives in the LANE AN, the network notices him and the active server fetches a profile for him from the profile database. Profile Database can be seen analogous to the GSM networks home location register. The profile contains user information such as a list of services for that user, parameters for these services, and the privilege level of that user. The AS starts active applications with proper parameters according to the profile information.

The adaptation should be made in the active networking way simply for optimization reasons. With active networks, the conversion can be located optimally within the network. Consequently, this makes it possible to adapt the network to the varying circumstances in each case.

For example, assume that there is a content server providing a mp3 audio stream and a subscriber with a low power mobile terminal. The terminal hasn't enough processing power to decode and play the mp3 stream. However, the terminal is able to play a wav stream that consumes less processing power than a mp3 stream. The problem in this example is that there is no wav stream to play. This can be solved by content adaptation. The subscriber has ordered a conversion service from his AN service provider; the AN knows to start conversion service for the subscriber. Because the bandwidth requirement is smaller with the mp3-stream than the wav-stream, the optimal place for conversion is as close as possible to the terminal. By making the conversion in the closest possible active server, the bandwidth consumption is minimized. An ordinary proxy conversion can not do this location optimization. Additionally, the AS can do stream caching in order to prepare for network interruptions and to prevent delays in the terminal end.

## VI. CONTENT ADAPTATION ON LANE PLATFORM

On the LANE platform, three different active applications have been demonstrated; traffic shaping, Mobile IP and content adaptation. In content adaptation, there are two different solutions: one that uses fixed proxies (Moxy) and another in which the conversion is implemented in an active application (Cproxy). Technically it is possible to merge both solutions, but as they are used only in demonstration purposes there is no reason to.

### *A. Moxy*

Content adaptation proxies in Moxy solution usually are active applications but they may also be traditional fixed proxy services. In this approach, the terminal must have a special software installed and running. The purpose of this software - called Moxy - is to act as a broker and negotiate with the active server. When client-side applications need to use active services, they are configured to use local ports as their proxies. Every protocol (e.g. HTTP. SMTP, etc.) have their own local ports that are listened by Moxy broker. The Moxy broker gets a report from the Moxy service in an active server. This report contains addresses to each subscribed content adaptation proxy. The Moxy broker redirects local ports to those addresses. The functionality of Moxy is presented in Figure 2.

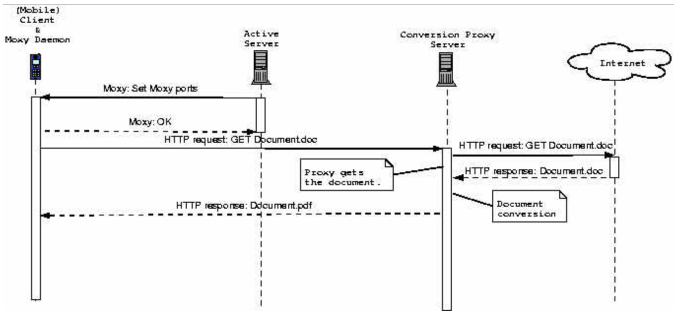


**Figure 2. Moxy sequence diagram**

In Moxy, the relocation of the conversion is done so that the conversion service in active server makes relocation decision and orders the client side broker to redirect ports to the new addresses. This could happen because of load balancing reasons, movement of the terminal, for congestion reasons etc.

Since the Moxy approach doesn't use totally active conversion but fixed proxies and the used proxy is chosen from a proxy pool, it could be called a “semi-active”. The main disadvantage of this solution is that additional software is needed in the terminal.

### *B. Cproxy*

The second approach, purely active and thus the most interesting one, is the active conversion solution. The content conversion is made by an active application in an active server. This active application is called Cproxy since it makes conversions and also performs common proxy duties.

Figure 3 presents the architecture of the Cproxy solution. The node operating system of the active server is Linux, and the execution environment is Java Virtual Machine (JVM). Inside the JVM is the active server, which executes the Cproxy active application. The Cproxy document conversion is based on the software called WvWare [10], which is a free tool for document conversions.

Service relocation in the Cproxy solution is more complex than in Moxy. Cproxy active service makes relocation decision and sends an *create_user*-request to some other AS, in which the Cproxy service is wanted to be transferred. After that, Cproxy sends request of removing old rule, which redirects HTTP traffic to the currently used Cproxy server. The new Cproxy service in a new AS sets a HTTP capture rule to the AR during its initialization.

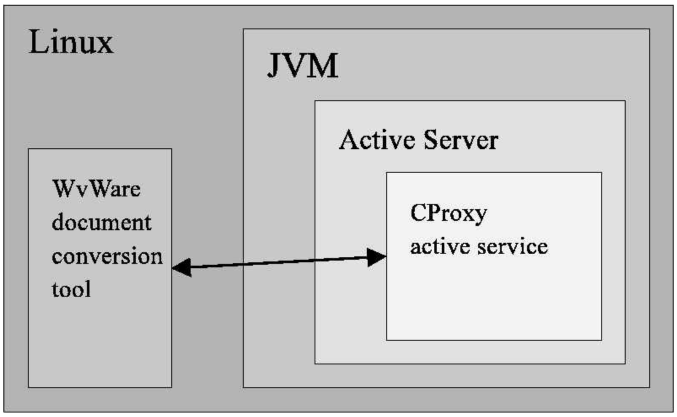


**Figure 3. Cproxy architecture**

The functionality of the Cproxy active adaptation is presented as a sequence diagram in Figure 4. The most notable thing is that the client terminal is untouched. There is no need for any additional software in the client terminal; all of the adaptation functionality is located in the network.

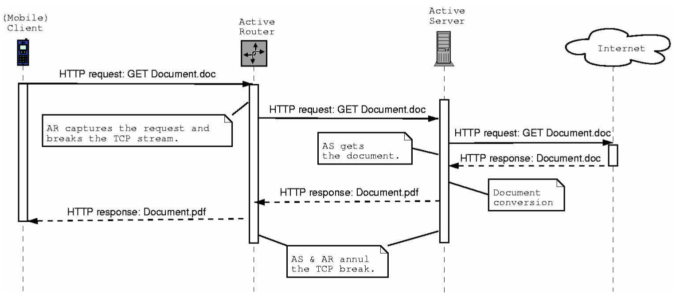


**Figure 4. Cproxy sequence diagram**

## VII. CONCLUSIONS AND FUTURE WORK

A working document conversion demonstration proves the concept of the AN based content adaptation as well as the whole LANE platform to be valid. There is still a lot of work to do in LANE but service demonstrations like active traffic shaping and document conversion, gives a strong faith to the future of AN technology and LANE.

Active networks offer a way to dynamically place content adaptation - as well as other services - into the network. The ability of free relocation of services is a notable benefit in the content adaptation since it offers diverse optimization capabilities. In content adaptation tests, it was found out that the Moxy approach was notably faster. Because in the Moxy solution, packet flow isn't rerouted in the AR, but goes directly to the proxy server. In Cproxy, every packet is rerouted from the active router to the active server, which takes time.

In future, it would be interesting to implement and test an AN based content adaptation for streaming media such as mp3 to ogg vorbis stream conversion. Another interesting issue would be a bandwidth optimization according to varying network load. This means dynamic altering of the voice stream quality. If these tasks do not offer enough challenge, a streaming media, a moving mobile terminal, and an alternating access network types combined should give brain teaser enough.

## ACKNOWLEDGMENT

I want to thank all my colleagues who have helped me to write this paper: co-authors and VTT's active network research team. These persons are, in alphabetical order: Kimmo Ahola, Mikko Alutoin, Markus Aroharju, Arto Juhola, Titta Koskinen, Sami Lehtonen, Juuso Pesola. Sami Pönkänen, Pertti Raatikainen, and Aki Rova.

## REFERENCES

[1] Marko Lyijynen, "Protocol Specific Embedded Operating System", Master's Thesis, Lappeenranta University of Technology, 2002, page 4

[2] David L. Tennenhouse and David J. Wetherall, "Towards an Active Network Architecture", DARPA Active Networks Conference and Exposition, 2002

[3] Markku Kylänpää and Immo Heino, "Personalized Transcoding Proxy – an Approach to Mobile Web Access", 9th International World Wide Web Conference, May 2000

[4] APRO project, VTT Information Technology, Finland, http://www.vtt.fi/tte/projects/apro/ [accessed 20. December 2002.]

[5] Digestor web page, http://www.fxpal.com/papers/bic97/ [accessed 20th of December 2002.]

[6] Jeremy Elson, Alberto Cerba, "The Internet Content Adaptation Protocol", IETF, 2002

[7] S. Ardon, P. Gunningberg, Y, Ismailov, B. Landfeldt, M. Portmann, A. Seneviratne, B. Thai, "Mobile Aware Server Architecture: A distributed proxy architecture for content adaptation", INET 2001, June 2001

[8] Website of the Caspian project, http://www.eurescom.de/public/projects/P900-series/P926/ [accessed 20th of January 2003 ]

[9] D. Glynos, C Boukouvalas, P. Bosdogianni, K. Ahola, A. Juhola, P.A. Aranda-Gutierrez, "Mobile Active Mail", Third Annual Working Conference On Active Networks (IWAN'01), 2001

[10] WvWare homepage, http://www.wvware.com/ [accessed 20th January 2003]