Calypso Service Architecture for Broadband Networks

Petteri Koponen, Juhana Räsänen and Olli Martikainen
Helsinki University of Technology
Laboratory of Telecommunications Software and Multimedia
Otakaari 1, 02150 Espoo, Finland
Telephone: 358 9 4514739, Fax: 358 9 4513293
{Petteri.Koponen,Juhana.Rasanen,Olli.Martikainen}@hut.fi

Abstract

The Calypso project aims at developing an extremely flexible control and service architecture for ATM-based broadband networks. This architecture provides various alternatives to distribute the network and service control functions among clients, servers and different network nodes. This means that a control or service function can reside not only in a network node, but in the customer's workstation or in the service provider's dedicated server. Instead of the traditional ATM or IN signalling, the Calypso architecture uses the TCP/IP protocol suite for the management and control of the network and services. The management, control and user data is transferred by means of IP switching. In addition to IP switching, the architecture will support end-to-end native ATM streams with guaranteed Quality of Service. In this paper we compare the Calypso architecture with the traditional B-ISDN and IN architectures. We focus on describing the Java-based Service Execution Environment that provides a flexible platform for the management and execution of both services and control functions.

Keywords

Intelligent Networks, ATM, Access Networks, Broadband Services, Multimedia Streams, Open Signalling

1 INTRODUCTION

It is expected that broadband media streams will be distributed over fixed and wireless networks to consumer segments within 5–10 years. The consumer media distribution has been delayed mainly due to the cost of transmission capacity. This cost is assumed to come down because of both the creation of the open, ATM (Asynchronous Transfer Mode) -based telecommunications infrastructure and the growing role of service and content providers.

New telecommunications technologies such as IP (Internet Protocol) switching, wireless access techniques and Digital Subscriber Line (xDSL) -based access technologies will accelerate the creation of a cost-efficient and flexible telecommunications infrastructure. IP switching increases the performance of IP-based networks, e.g. intranets and the Internet, by combining the traditional IP routing with ATM switching, while wireless and xDSL technologies, especially Asymmetric Digital Subscriber Line (ADSL), will provide affordable broadband access for consumers.

The Internet and the liberalization of the telecommunications market in Europe in 1998 are the major reasons for the growing role of service and content providers. The question of how to build services that utilize distributed data and applications belonging to other service and content providers leads to further questions: how to manage the interaction between the services; how to build service creation, management and accounting subsystems; and how to introduce security and payment methods in the subsystems. To achieve this, the service functions have to be separated from the network control functions. Also, the service and network architectures have to become considerably simpler and more cost-effective.

In this paper we present the *Calypso* architecture as a cost-effective and flexible alternative for distributing broadband services. The Calypso architecture provides various alternatives to divide the network and service control functions among clients, servers and network nodes. This means that a control or service function can reside not only in a network node, but in the customer's workstation or in the service provider's dedicated server. Instead of the traditional ATM or Intelligent Network (IN) signalling, the Calypso architecture uses the TCP/IP (Transport Control Protocol / Internet Protocol) protocol suite for the management and control of the network and services. The management, control and user data is transferred by means of IP switching. In addition to IP switching, the architecture will provide end-to-end native ATM streams with guaranteed Quality of Service (QoS).

2 MOTIVATION

We believe that the current ATM and IN standards do not meet all the heterogeneous networking requirements of the future. This opinion is not unique – we share it with several academic and non-academic research groups (Lazar, Lim and Marconcini 1996, Leslie, Crosby and Rooney n.d., Newman, Minshall, Lyon and Huston 1997). Below we have listed what we believe are the most vital issues for future broadband networking architecture and some of the problems that ATM faces when addressing these issues.

- The networking requirements will remain *heterogeneous*, i.e. networks of different types will provide different services and the services will require different control functions from the underlying network. This implies that the broadband networks should support a wide – possibly configurable –

variety of control functions for different needs, and that services that use these control functions should be easy to implement, deploy and manage. At present, ATM's control functions, e.g. signalling, are relatively fixed, and the services, e.g. IN service, are assumed to support different networking requirements. For example, support for mobility could be added to ATM networks by implementing the mobile specific features as IN services. However, it has proved to be very difficult to add efficient support for e.g. mobility (Mitts 1996) to ATM networks. This is mainly due to the somewhat heavyweight and rigid UNI/NNI and IN signalling standards.

- The success of the World Wide Web (WWW) and the Internet has made the TCP/IP protocol suite dominant, which implies that also the ATM networks should support the TCP/IP protocols. Doing this in a practical way has proved to be very hard. This is mainly due to some fundamental differences between the ATM and IP protocols; for example, ATM is connection oriented while IP is connectionless (Comer 1995, Cole, Shur and Villamizar 1996).
- It is expected that the distribution of both broadband and narrowband media streams such as TV and radio broadcasts, Video on Demand (VoD), video conferencing and telephony, is becoming an increasingly important service. Because of ATM's short, fixed-length cells and light-weight flow control and error handling functionality, it is possible to transfer streams of all these types over a single ATM link and to benefit from statistical multiplexing. ATM networks are also expected to guarantee streams end-to-end *Quality of Service* (QoS). However, interoperability problems between switches of different vendors have reduced the benefits gained from the ATM technology. The problems are usually related to the complexity of the standardized QoS mechanisms and the corresponding control functions.

To summarize, the broadband network architecture should support heterogeneous networking requirements; provide an efficient implementation of IP transmission; and implement end-to-end point-to-point and point-to-multipoint (and broadcast) streams with QoS guarantees. We – and e.g. (Newman et al. 1997) – believe that the ATM hardware is flexible and cost-efficient enough to be used to implement these features. However, the ATM control functions and the IN service model make this very hard and cause most of the problems in the current ATM networks. This suggests that a more flexible alternative should be studied, which is the aim of the Calypso project.

3 RELATED WORK

The Internet Engineering Task Force's (IETF's) IP over ATM working group discusses several proposals for routing and forwarding IP packets over ATM subnetworks in (Cole et al. 1996). The proposals require two logically separate networks – the IP and the ATM network. For example, all the proposals, excluding the Integrated Models that are planned to use extended Private Network-Network Interface (PNNI) (ATM Forum 1996) routing, implement both IP and ATM routing protocols. In addition, each proposal uses both IP and ATM addresses. We believe that all the architectures discussed in (Cole et al. 1996) are very hard to manage in practise due to their complexity. Also, the proposals are relatively hard to implement because they include several complex protocols. This may cause further interoperability problems.

There are a number of proprietary solutions that use ATM as a backplane- or bus-replacement within a router itself. These include e.g. Ipsilon Networks' *IP switching* (Newman et al. 1997) and Cisco Systems' Tag Switching (Rekhter, Davie, Katz, Rosen and Swallow 1997). Of these, we find IP switching a very innovative and practical solution for running IP over ATM (or, according to Ipsilon Networks, "ATM under IP").

An IP switch consists of an efficient workstation and a simple ATM switch. The workstation – also called the *switch controller* – is connected to the switch via an ATM link, and it manages the switch with *General Switch Management Protocol* (GSMP) (Newman, Edwards, Hinden, Hoffman, Liaw, Lyon and Minshall 1996). GSMP enables the external switch controller to e.g. establish and release ATM Virtual Channels (VCs), query status information and receive asynchronous notifications from the switch. GSMP is not specific to IP switching but it can be used also with other techniques such as the traditional ATM signalling. GSMP is an interesting concept as such, and we have experimented with it in the TOVE project (Puro, Koponen, Räsänen, Nummisalo and Martikainen 1996).

IP switching integrates ATM hardware directly with IP, not requiring logically separate ATM and IP networks. Instead, the switch controller acts as a normal IP router with one exception: in addition to GSMP, it implements a few relatively simple IP switching specific functions that enable it to switch IP packets belonging to long-lasting flows instead of routing them. Ipsilon estimates that this increases the routing performance about 4.5 times (Newman et al. 1997). Although IP switching can provide IP-based media streams with QoS guarantees (this requires careful configuration of the IP switched subnetwork, so that all the IP switches decide to switch the streams instead of routing them and that they establish VCs with the same QoS parameters) and it does not require ATM signalling or routing protocols, we find it too focused on IP as such. For example, connecting an IP switched network to a traditional ATM network requires complex gateway arrangements. Furthermore, IP switching does not guarantee end-to-end QoS for media streams because the decisions that IP switches make are *local in scope*.

The OPENSIG (OPEN SIGnalling) working group is an interesting opening towards open and programmable network architectures. The working group aims to "do research towards understanding open network control issues as they arise in signalling, middleware and service creation on ATM-, Internet- and Mobile-multimedia networking platforms." The Calypso architecture has

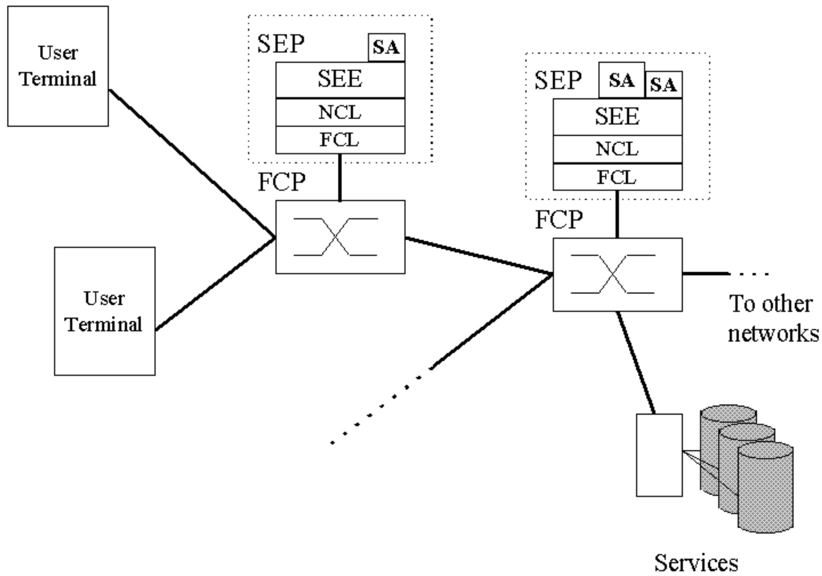

Figure 1 Calypso architecture.

somewhat similar objectives as the two OPENSIG projects: xbind (Lazar et al. 1996) at Columbia University and the Distributed Control of ATM Networks (DCAN) project at University of Cambridge (Leslie et al. n.d.). Both projects propose an open architecture where the control and management functions can reside on external workstations. The xbind project is focused on representing resources such as switches and multimedia devices as Common Object Request Broker Architecture (OMG 1995) (CORBA) objects that are manipulated in order to generate network services. In the DCAN project, the control and management functions are extracted from the network nodes (and end systems) to an external distributed processing platform.

4 CALYPSO ARCHITECTURE

One of the main principles in the Calypso architecture is the separation of the service functions, including the traditional control functions such as signalling, and the switching functions. Figure 1 illustrates this situation; the *Service Execution Point* (SEP) is both logically and physically separated from the *Fabric Control Point* (FCP). In Ipsilon Network's IP switching system and in the TOVE project (Puro et al. 1996) at Helsinki University of Technology, the *switch controller* is a close equivalent to the SEP.

Amongst other areas, the TOVE project has studied the benefits of separating the switch controller (SEP) from the switching fabric (FCP). In addition to the increased flexibility in updating the software and the hardware of the SEP, the approach enables a single SEP to manage a number of FCPs. This is a cost-efficient and practical solution in case the signalling load is relatively small compared to the amount of the data transmitted. A situation like this could occur in broadband access networks where a large portion of data consists of long-lasting audio and video streams. Also, if the SEP cannot handle the current signalling load, a new SEP can easily be introduced and the FCPs can be split into two groups.

The SEP contains the *Service Execution Environment* (SEE) and two layers: the *Fabric Control Layer* (FCL) and the *Network Control Layer* (NCL). The FCL provides means to manage the FCP. It can implement e.g. GSMP or some proprietary switch management protocol as long as it supports the basic functionality of an ATM switch. The NCL implements the TCP/IP protocols needed, including IP routing (or in this case, IP switching), and the basic connection mechanisms of the Calypso architecture. The SEE wraps these layers inside Java objects with well-defined interfaces. Both the SEE and the SAs use these interfaces.

4.1 Service Execution Environment

The SEE is a platform for the management and execution of *Service Agents* (SAs).

We adopt here the broad definition of a service by Magedanz and Popescu-Zeletin (Magedanz and Popescu-Zeletin 1996). They identify in the IN architecture services offered by the underlying network platform (i.e. *bearer services* such as audio, video, or data transmission and signalling services) and services offered by the IN platform to the end users (i.e. *supplementary services* or value-added services). Compared to the IN Service Control Function (SCF) (ITU-T 1995), the SEE contains more functionality because it supports not only the traditional service applications, but also control applications such as signalling. Hence, the above definition of a service is proper for the SEE.

SAs are *Java* programs that the SEE both manages and executes. The management includes e.g. providing means to safely distribute the SAs over the network, and allowing SAs access some of the resources the SEE manages. Also, the SEE executes the SAs and monitors their execution. The management and execution functions are strongly interconnected; for example, the SEE checks at run-time whether an SA is allowed to access a resource depending on the security level the management functions have given to the SA.

According to our definition of services, almost all the SEP's functionality is implemented by SAs that are either *fixed* or *mobile*. The fixed SAs cannot be transferred over the network and they can include native, i.e. non-Java, code because of e.g. efficiency reasons or existing implementations that are to be reused. However, the fixed SAs have to implement the necessary interfaces for management and monitoring purproses, i.e the native code must be wrapped in the corresponding Java classes. The mobile SAs are more flexible in the sense that they can be dynamically injected into an SEE , although this requires quite advanced security mechanisms. How these mechanisms will be implemented is still open.

4.2 Relation to IN and ATM Architectures

Implementing services as SAs and separating the SEP from FCP enables the use of the same hardware and SEE in a very wide range of networking domains. We believe that the most important differences between the Calypso architecture and traditional IN and ATM architectures are due to the additional flexibility that this approach introduces. Among the differences are the following:

- Both services and control functions are implemented as SAs. This differs fundamentally from the IN and ATM architecture where control functions, i.e. signalling, are more or less fixed and services communicate with these by sending IN Application Part (INAP) messages. INAP is not needed in the Calypso architecture because the SAs can communicate with each other using domain or service specific means, e.g. Java Remote Method Invocation (RMI), CORBA or some customized ATM- or IP-based protocol. In the Calypso architecture, thus, fixed INAP messages are replaced with service-specific messages or e.g. RMI or CORBA method calls. Of course, an INAP agent can be implemented as a gateway to IN services if necessary.
- In Calypso architecture, the authorized services can directly control the network by ordering the FCL and NCL to establish end-to-end or local ATM streams. This means that there is no clear separation between services and control functions. Also, the management functions can be implemented as SAs, which opens interesting opportunities to experiment with different management architectures.

We believe that the Calypso architecture is more programmable than the traditional IN and ATM architectures, which reduces the time to create the services and also allows more advanced services to be implemented. However, we must emphasize that we have not planned the Calypso architecture to be a global replacement for the IN and ATM architectures. Instead, it is a cost-effective and flexible alternative to implement subnetworks providing innovative IP and ATM stream-based services.

4.3 Implementation Issues

The implementation of the prototype Calypso SEP platform is based on the results of the TOVE project. The TOVE platform is a standard ATM switch controller which is connected to switching fabric via a 155-Mbps ATM link and is able to control the fabric with GSMP or similar protocol. The operating system used is Linux with ATM extensions and the hardware is common PC/Intel technology. The platform and signalling protocols developed have been tested with the Frame Synchronized Ring (FSR) switch (Raatikainen 1996) developed by the Technical Research Centre of Finland (VTT). The TOVE platform readily implements the FCL functionality of the Calypso SEP, on top of which the NCL and SEE can be implemented.

The NCL will use the TCP/IP stack and ATM extensions of it (such as Classical IP over ATM) for transporting control data between network nodes. The NCL also contains an interface to open end-to-end ATM streams and management of the connections.

The SEE will be implemented as a Java environment that provides a class framework, object repository, authentication and security functions etc. for the Service Agents to use. Java has been selected because it provides a wide selection of tools, TCP/IP support and easy code mobility accross network.

5 FUTURE WORK

The Calypso platform will be applied in the service development in the MediaPoli environment. The MediaPoli consists of a broadband access network in Otaniemi (Helsinki University of Technology) campus area with 1000–3000 customers and digital multimedia services on top of the network. Typical services include ATM telephony, electronic commerce in the WWW and digital media stream distribution. Some service scenarios are given below. The services are developed in research projects and also commercial pilots are allowed. The MediaPoli environment will be open for both domestic and foreign partners.

A scenario of a basic stream service (e.g. a TV channel broadcast) is presented in Figure 2. A TV channel is broadcasted from the server as an ATM point-to-multipoint stream. When User 1 wants to watch the channel, she clicks a Service Icon (SI) (e.g. a Java applet) that represents the service in her browser. The icon connects (1) to the corresponding SA in the SEE that controls this particular service. The icon and agent exchange authentication data and verify that the user has access to the service, e.g. that she has purchased subscription to the channel. In the next phase the SEE uses NCL (2) to establish a new branch (4) to the existing multipoint stream (3), and when the user is informed of the established stream, the icon can launch a viewer application in the user's workstation.

We would like to emphasize that in this architecture the multimedia stream

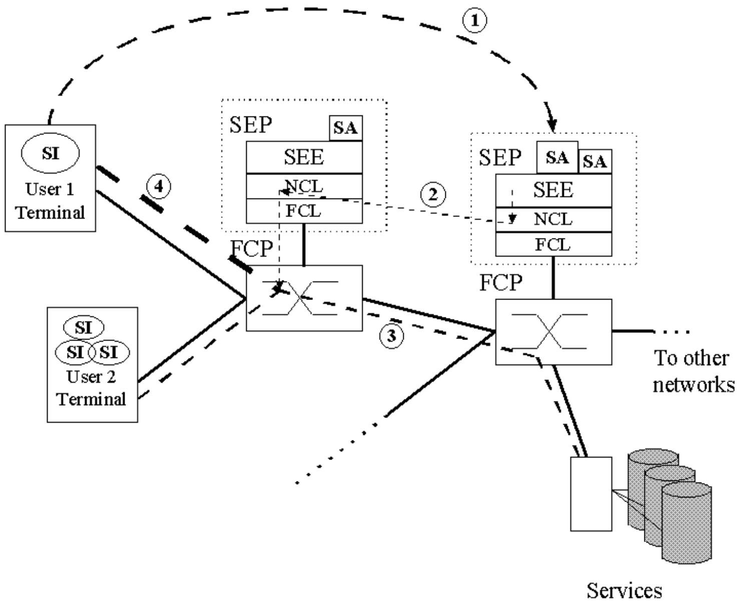

Figure 2 Service scenario.

distribution service can be much richer than the current broadcast TV. It is easy to incorporate auxiliary data into the broadcast, including Universal Resource Locators (URLs), transcript for people with impaired hearing or multilingual translations as subtitles etc. This would make a TV program a true multimedia application.

The ability to move service agents from a network node to another opens interesting possibilities to implement mobile network functions, e.g. location updates. A location update is implemented in the current GSM network as a MAP (Mobile Application Part) protocol dialogue that performs necessary database updates in Home and Visitor Location Registers. The Calypso architecture enables the Mobile Terminal (MT) to inject an agent into the network, and the agent would be responsible for updating the location of the MT.

REFERENCES

ATM Forum (1996). *Private Network-Network Interface Specification Version 1.0.*

Cole, R., Shur, D. and Villamizar, C. (1996). IP over ATM: A Framework Document, IETF RFC 1932.

Comer, D. E. (1995). *Internetworking with TCP/IP – Volume 1*, 3. edn, Prentige-Hall.

ITU-T (1995). *ITU Draft Recommendation Q.1224: Distributed functional plane for intelligent network CS-2*, Berlin.

Lazar, A. A., Lim, K.-S. and Marconcini, F. (1996). Realizing a Foundation for Programmability of ATM Networks with the Binding Architecture, *IEEE Journal of Selected Areas in Communications* **14**(7): 1214–1277.

Leslie, I., Crosby, S. and Rooney, S. (n.d.). The Distributed Control of ATM Network Project, `http://www.cl.cam.ac.uk/Research/SRG/dcan/`.

Magedanz, T. and Popescu-Zeletin, R. (1996). *Intelligent Networks – Basic Technology, Standards and Evolution*, International Thomson Publishing.

Mitts, H. (1996). *Architecture for wireless ATM*, Licentiate's thesis, Helsinki University of Technology.

Newman, P., Edwards, W., Hinden, R., Hoffman, E., Liaw, F. C., Lyon, T. and Minshall, G. (1996). Ipsilon's General Switch Management Protocol – Version 1.1, IETF RFC 1987.

Newman, P., Minshall, G., Lyon, T. and Huston, L. (1997). IP Switching and Gigabit Routers, *IEEE Communications Magazine* **35**(1): 64–69.

OMG (1995). *The Common Object Request Broker: Architecture and Specification Revision 2.0.*

Puro, V.-M., Koponen, P., Räsänen, J., Nummisalo, P. and Martikainen, O. (1996). TOVE in Universal Mobile Telecommunications System, *Proceedings of the 2nd Workshop on Personal Wireless Communications*, Frankfurt.

Raatikainen, P. (1996). *Analysis and Implementation of a High-Speed Packet Switching Architecture – the Frame Synchronized Ring*, PhD thesis, Helsinki University of technology, Espoo.

Rekhter, Y., Davie, B., Katz, D., Rosen, E. and Swallow, G. (1997). Cisco System's Tag Switching Architecture Overview, IETF RFC 2105.