A Configurable Service Architecture for ATM 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 4515014
{*Petteri.Koponen,Juhana.Rasanen,Olli.Martikainen*}*@hut.fi*

Abstract

This paper presents a configurable, light-weight Calypso service architecture for ATM networks. The architecture enables coexistence of multiple control, management and service architectures in a single network node. Furthermore, the architecture defines only the minimum functionality for a network node; the rest of the functionality can be configured by injecting or removing service components. Finally, the architecture enables the flexible distribution of service components between clients, servers and network nodes.

The Calypso architecture is based upon the observation that although ATM has proven to be a superior data transfer technology, its control, management and service architectures do not fullfil all the heterogenous requirements of the future broadband services. However, the recent advances in both the ATM and the software technology enable the introduction of more flexible, innovative architectures while retaining compatibility with the existing standards. This is going to be proved by the ongoing implementation of a pilot system that will be based upon the Calypso architecture.

Keywords

ATM, Open Switch Control, Programmable Networks, Service Architecture, Intelligent Networks

1 INTRODUCTION

The broadband transmission technologies such as Asynchronous Transfer Mode (ATM) and optical switching will increase the capacity of backbone networks. In addition, the emerging wireless and Digital Subscriber Line (DSL) access technologies will provide the residential user with low-cost broadband connectivity. It is envisioned that due to these advances in transmission technology, bandwidth becomes a commodity in the near future, which makes the rapid creation and deployment of innovative services the key success factor of the future broadband networks.

Although the transmission capacity has rapidly increased, the software-based control and management architectures of ATM still partly rely on technologies a couple of decades old. For example, the signalling protocols of ATM are based upon the equivalent Integrated Services Digital Network (ISDN) and Signalling System No. 7 (SS7) protocols. Also, the Broadband ISDN (B-ISDN)* service architecture is planned to follow the relatively heavy-weight and rigid Intelligent Network (IN) model [3]. We, and for example [5, 11], believe that the current control, management and service architectures of ATM do not fulfill all the heterogeneous requirements of the future broadband services.

In this paper we present a novel service architecture for ATM networks. We adopt here the broad definition of a service by Magedanz and Popescu-Zeletin [7]. They indentify (in the IN architecture) services offered by the underlying network platform (bearer services*) and the services offered by the service platform to the end users (supplementary services or value-added services*). We extend this definition to include also the management functions. This definition is proper for our architecture in which the distinction between the traditional control, management and service functions is rather vague.

We call our service architecture the Calypso architecture. It is based upon three key principles. Firstly, the architecture enables the coexistence of multiple control, management and service architectures* in a single network node; for example, a network node may simultaneously implement User-to-Network (UNI) signalling and some domain- or application-specific signalling protocol. Secondly, the Calypso architecture defines only the minimum functionality for a network node; the rest of the functionality can be configured by injecting or removing service components. Thirdly, the architecture enables the flexible distribution of service components between clients, servers and network nodes. To summarize, the Calypso architecture is service-centric; its primary target is to provide a flexible, efficient platform for servive creation, deployment and management.

This paper is organized as follows. Section 2 describes the motivation behind the Calypso architecture, that is, why we believe that an alternative for the IN-based ATM architecture needs to be and can be developed. Section 3 presents the three-layer architecture of the Calypso network node. Section 4 describes how we plan to implement a prototype network node. Section 5 compares

---

*ATM has been chosen as the implementation technology of the B-ISDN reference architecture.

*Bearer services include audio, video and data transmission and signalling services.

*Supplementary services in the IN model, for example, include call forwarding and televoting services.

*Here we apply the traditional separation between the control, management and service functions.

the Calypso architecture with related work. Finally, Section 6 presents some conclusions and future work.

2 MOTIVATION

Our hands-on experience of ATM signalling [14] and the delayed standardization of the Broadband IN (B-IN) service model as well as concerns of its applicability to heterogenous services encouraged us to seek alternatives. We list below the key trends in broadband networking that in our opinion call for a network and service architecture more flexible than the IN-based ATM architecture.

- The residential user will be provided with low-cost broadband connectivity. This means that the field of broadband networking is expanding from the business sector and corporate networks to the mass market of residential access. This situation opens a completely new market for the distribution of a variety of services over a single network. These services range from Internet access and basic telephony service to broadcast TV and Video on Demand (VoD) as well as to some completely new areas such as interactive TV. Also mobile access is likely to be an important part of this environment.
- The TCP/IP (Transport Control Protocol / Internet Protocol) protocol suite will remain the de facto standard of global internetworking. The success of the World Wide Web (WWW) and the Internet imply that also the ATM networks must support the TCP/IP protocols. Doing this in a practical and efficient way has proven to be very hard. This is mainly due to the fundamental differences between the ATM and IP protocols*.
- The networking requirements will be heterogenous. A very flexible network architecture is essential in order to support a combination of a digital multimedia distribution network and the global, packet-switched Internet. ATM addresses this requirement at the data transfer level by providing several traffic classes and the Quality of Service (QoS) concept. However, the telecommunications-oriented call and connection control model of ATM is not flexible enough to efficiently support TCP/IP networking or even some of the more traditional fields of telecommunications; for example, adding mobility support to ATM networks has proven very difficult [9].

We conclude that although ATM is a cost-efficient and flexible data transfer technology, its service architecture should be reengineered to support creation, deployment and management of new services. Fortunately, such a trend can be seen within the academic community and the industry [5, 11, 17]. We believe that the following items are the primary factors that enable the reengineering of the ATM service architecture.

*For example, ATM is connection-oriented while IP is connectionless.

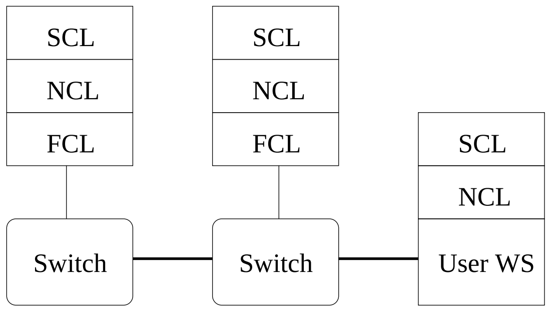

Figure 1 Three-layer control model

- The separation of switching functions from control and management functions in ATM switches. This is being studied by several groups [10, 15, 5, 13]. The separation enables the same hardware to be used with different control and management software and usually makes the system easier to configure and maintain. In a typical solution a switching fabric is connected to a general purpose workstation that runs the control and management software.
- The recent advances in software technology. The most promising technologies include object-oriented methods and component software models such as design patterns [16], frameworks [2], and Common Object Request Broker Architecture (CORBA) [12]. In addition, platform-independent mobile code technologies such as Java and Limbo/Inferno provide interesting opportunities for the implementation of services and telecommunications software in general. The separation of switching from control and management leveradges the benefits gained from the advanced software technology because standard computing environments and development tools can be used instead of proprietary embedded systems and tools.

3 THREE-LAYER CONTROL MODEL

In Section 2 we mentioned the separation of switching hardware from control and management software. To further structure the software needed to implement the control and management functions, we define three distinct control layers that are illustrated in Figure 1. We emphasize that these are abstract layers that specify a simple reference architecture; implementation choices of our pilot system are described in Section 5.

- The Fabric Control Layer (FCL) provides a generic, yet extendable interface for the management of an ATM switch. The FCL is similar to the Ariel interface of the Distributed Control of ATM Network (DCAN) project [6].

It can also be considered as a wrapper for the Ipsilon's General Switch Management Protocol [10]. Good descriptions about the features required from an open, generic switch control interface can be found in [17, 10], and we do not repeat them here.

- The Network Control Layer (NCL) provides secure channels for the transfer of service data. This channels are used to manage the services* and to provide standardized means for services to communicate with each other. The communication at the NCL occurs between adjacent nodes, and the its implementation technique must provide at least a connectionless network layer protocol for transferring data between nodes, a routing protocol for exchanging protocol information and a protocol for establishing connections through the network.

  In addition to supporting secure data transfer channels the NCL includes methods for controlling end-to-end ATM connections. Although this function could be implemented solely by the services*, we believe that including it in the service platform makes the implementation of simple services much easier. The NCL supports both point-to-point and point-to-multipoint Virtual Channel (VC) and Virtual Path (VP) connections. The NCL also supports leaf-initiated join, which is an essential feature in implementing mobility-related services such as handovers.
- The Service Control Layer (SCL) is the highest layer of the Calypso architecture. The SCL manages service components. It provides the services with different interfaces according to their access rights. In the traditional IN model, services control the network via control functions by exchanging Intelligent Network Application Protocol (INAP) messages. The greatest difference between this model and the Calypso architecture is that in the latter services that have enough access rights can directly* interact with the FCL or the NCL. According to this model, thus, the control and management functions are just services with certain (high) access rights and they can be managed just like the traditional service components. Also, because all the resources of the network node are accessed via the SCL, multiple service architectures can coexist in the node if the resources—for example the VPI/VCI space—are appropriately allocated and reserved.

  In Figure 1 also the client workstation implements the NCL and the SCL. However, these implementations do not have to be nearly as rich as the network side ones. For example, the NCL does not have to include routing or end-to-end connection management functionality, and the SCL may be for example a WWW browser that allows downloading and execution of Java applets. We believe that the minimum requirements for the client-side layers become clearer as the pilot service implementations advance.

---

*The service management consists of injecting and removing services, monitoring them, and collecting statistics.

*For example, an UNI signalling service can be used to establish ATM connections.

*In this context directly means the same as "via the SCL", that is, "not via the control functions".

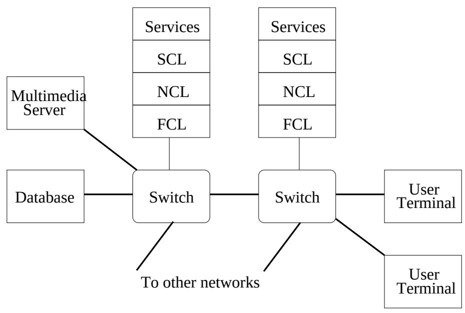

Figure 2 Calypso network architecture

4 IMPLEMENTATION ISSUES

We are currently working on a pilot system that will implement most of the key components of the Calypso architecture. The system consists of a network node, a couple of different types of service applications and the client and server components that these services require. The alpha version of the pilot system should be ready before the Christmas 1997. Because of the increased importance of the residential user, the pilot system focuses on providing broadband services for both the residential user and the small- and medium-sized company.

The design has started from service scenarios and is progressing towards an architecture that will realize these scenarios. An important goal is to reuse existing, preferably publicly available software components and designs as much as possible to enable rapid prototyping of the ideas proposed in this paper. Due to this we have directed the research towards experimenting with early prototypes instead of focusing on performance and quality of the subsystems. Our initial goal is to assemble a working system in which some elements such as routing or connection admission control may not be optimal.

An example of the architecture of the planned pilot system can be seen in Figure 2. A typical Calypso access network consists of a group of access switches connected to a high-performance backbone network. The users are connected to the network with for example DSL* modems over the telephone wire. A typical terminal device of the user is a set-top-box or a workstation with similar functionality. One access switch can handle 100–500 users, so there may be, for example, a switch located in each block of flats and the switch controllers (workstations) are located either in the premises of the network provider or besides the switches.

Multimedia servers, databases, and other servers are directly connected to the ATM network. The services provided by these servers may be distributed via native ATM connections that use advanced ATM-specific features such as

*Currently the most interesting DSL technology seems to be Asymmetric DSL (ADSL).

traffic classes and QoS. Because the NCL can establish connections, neither the servers nor the user terminals are forced to implement for example UNI or Network-to-Network (NNI) signalling.

4.1 Calypso FCL

The Virtual Exchange (VE) module developed in the TOVE project [14] will be extended and used in the Calypso FCL. The VE is a control and management framework for ATM switches. Currently it can be used with the experimental Frame Synchronized Ring (FSR) [13] switching fabric that is implemented by the Technical Research Center of Finland. However, we are working to add support for GSMP in the VE.

4.2 Calypso NCL

The Calypso NCL is based on the reuse of the TCP/IP protocol suite. The key idea is to implement the Calypso network solely based upon IP, that is, the Calypso network uses IP addresses, routing, and services such as Domain Name Service (DNS). An important issue is that also the emerging IP security model can be applied in the network. Due to this, secure (and non-secure) IP channels can be used to transfer service data as described in Section 3. IP offers several other benefits: Firstly, the technology is existing and it has been extensively tested. Secondly, IP tools and applications such as routing protocols, Simple Network Management Protocol (SNMP), and HyperText Transfer Protocol (HTTP) can be used in the Calypso architecture. Thirdly, there is no need to manage two overlaying architectures as most of the traditional IP over ATM solutions do [1].

Our initial plan is to connect the elements of the Calypso network with Classical IP over ATM [4] and Permanent VCs (PVCs). However, we emphasize that to keep the performance requirements of the IP routing as low as possible, the IP implementation of the NCL is used only to transfer service data. The IP routing service to the Internet or to intranets are implemented using a separate set of VCs that are switched directly to the a routing server. By establishing VCs of different bandwidth and traffic class, it is easy to provide the user with just the kind of Internet connection she has paid for. This can be considered as the Connection Admission Control (CAC) mechanism for the Internet.

4.3 Calypso SCL

The Calypso SCL of the pilot system is called the Service Execution Environment (SEE). The SEE is a platform for the management and execution of services. Due to our extended definition of a service, the SEE allows also the traditional control and management components to be injected into and removed from itself, which enables domain- and service-specific control and management functions. For example, the SEE can be configured to support mobility by introducing service components that implement the handover procedures.

In the pilot system the services can be either fixed or mobile. Most of the services will be implemented with platform-independent Java language and they are inherently mobile, that is they may be transferred over secure channels and injected into a SEE. The fixed services cannot be transferred over the network and they may include non-Java (native) called because of performance reasons or the need to reuse existing code. However, also the fixed services must implement the interfaces required for service management purposes. This means that the possible native code must be encapsulated inside Java classes.

The initial user interface to the services is going to be browser-based. The user accesses the services in her personalized service page that is stored in the network and updated when she buys new services. Services are either applets or Java applications that are able to communicate with the respective services located either in SEEs or in service providers' servers.

When the user buys a service, she actually obtains a credential to use some resources that can be for example services, connections, or storage capacity. Due to this the user part of a service must contain the credentials necessary if it needs to access resources protected by the SEE or by a certain service. However, the user part of the service may also include complex service logic, which enables the flexible distribution of service functionality between users, service providers' servers, and network nodes.

5 RELATED WORK

The OPENSIG (OPEN Signalling) working group is an interesting opening towards open and programmable network and service 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 is somewhat similar to xbind [5] architecture of the COMET group. The xbind architecture representing resources such as switches and multimedia devices as CORBA objects that are manipulated in order to generate network services. We believe that the relatively heavy-weight CORBA model is not suitable for all systems. Also, the xbind interfaces that are defined with CORBA Interface Definition Language (IDL) are fixed, which reduces the flexibility of the architecture*. However, we believe that the connection management algorithms and QoS models designed for the xbind architecture could be used in the Calypso pilot system.

The switchlet concept[17] and the Hollowman architecture[15] are even closer to the Calypso architecture. The switchlet architecture enables multiple control architectures to coexist within the same network or even withing the same network node. The Hollowman architecture delegates as much control from ATM switches to applications in order to break the limits of the current signalling protocols. It loads Java applications—or Call Closures—into the call manager of the switch. The Call Closures can implement application-specific control behavior, which, according to Rooney [15] increases the flexibility of the call model and the allocation of network resources.

The Calypso architecture differs from the above architectures by allowing services to access not only the call-related resources, but also the interfaces provided by the SCL and the other services. Also, we believe that the credential-based mechanism to distribute services among different network entities, and the use of IP protocols and services at the Calypso NCL provide an insight into one possibility to implement a flexible service architecture for ATM networks.

6 CONCLUSIONS AND FUTURE WORK

The concept of distributing services between user terminals and network nodes has been presented already in [8], but due to the slow progress of B-IN standardization there has been no practical platforms on which the ideas could have been experimented. Although our pilot implementation has just started, we believe that the Calypso architecture will prove to an efficient and flexible platform for service creation, deployment and management.

Although we have adopted the IP security model, we still have to solve many issues regarding to the security of Java-based service applications. In the present model, the service applications need to be trusted, that is, they are supposed to be reasonable well-tested and at least they do not intentionally try to harm the platform or other services. However, our ultimate goal is to open the SCL for service provides, so that they could implement services of their own, buy credentials to use resources of a specific set of network nodes, and inject the service components into these nodes. Naturally, this kind of a service provision model requires more from the application security model than the current Java environment can deliver.

---

*The interfaces must be standardized to maximize their benefit. This process has already started (see http://www.comet.ctr.columbia.edu for details).

REFERENCES

[1] Cole, R., Shur, D., and Villamizar, C. IP over ATM: A Framework Document. IETF RFC 1932, Apr. 1996.

[2] Hueni, H., Johnson, R., and Engel, R. A Framework for Network Protocol Software. In *OOPSLA'95 Proceedings* (Austin, 1995).

[3] ITU-T. *ITU Draft Recommendation Q.1224: Distributed functional plane for intelligent network CS-2.* Berlin, Nov. 1995.

[4] Laubach, M. Classical IP and ARP over ATM. IETF RFC 1577, Jan. 1994.

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

[6] Leslie, I., Crosby, S., and Rooney, S. The Distributed Control of ATM Networks Project. Available at `http://www.cl.cam.ac.uk/Research/SRG/dcan/`.

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

[8] Martikainen, O., Naumov, V., and Samouylov, K. Call Processing Model for Multimedia Services. In *Proceedings of the IFIP Conference on Intelligent Networks and New Technologies* (Copenhagen, Aug. 1995), Chapman & Hall, pp. 241–251.

[9] Mitts, H. *Architecture for wireless ATM.* Licentiate's thesis, Helsinki University of Technology, Sept. 1996.

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

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

[12] OMG. *The Common Object Request Broker: Architecture and Specification Revision 2.0*, July 1995.

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

[14] Räsänen, J., Koponen, P., and Martikainen, O. Broadband network architectures. Accepted to Network Information Processing Systems, Sofia, Oct. 1997.

[15] Rooney, S. An Innovative ATM Control Architecture. In *Proceedings of the Fifth IFIP/IEEE International Symposium of Integrated Network Management* (May 1997).

[16] Schmidt, D. C. Using Design Patterns to Develop Reusable Object-

Oriented Communications Software. *CACM Special Issue on Object-Oriented Experiences 38*, 10 (Oct. 1995).

[17] VAN DER MERWE, J. E., AND LESLIE, I. M. Switchlets and Dynamic Virtual ATM Networks. In *Proceedings of the Fifth IFIP/IEEE International Symposium of Integrated Network Management* (May 1997).