Overview of the DCAN project

Timothy Roscoe and Sean Rooney

12th July 1995

1 Abstract

Multiservice networks are networks capable of transmitting simultaneously from producers to consumers, data from different origins and with different characteristics (in terms of maximum supported delay, basic error rate etc.) in such a way that the data can be synchronised and exploited by the consumers.

Networks supporting the switching of multiservice information have additional needs to those simply carrying homogeneous information such as voice. These additional needs have lead to new techniques in information transmission. Currently, the preferred solution for the switching of multimedia information is by fast packet switching or ATM as it is more commonly called.

The techniques needed for the management/control of a a Multi-Service Network are different from those used in a data or telephone network. The differences come from the fact that all communication, even information related to management and control, must take into account Quality of Service (QoS). Hence at all levels from the management applications down to the underlying network equipment it must be possible to make assertions about the QoS. In addition, any meaningful management of a Multi-Service network must be distributed. This is true to some extent in certain single service networks such as telephony, which for geographical and/or commercial reasons are distributed. However, it is implicit in Multi-Service networks due to the need to make assertions about QoS on every node of the network even if the nodes are physically close together and all under the control of the same operator

Orthogonal to the above is the trend to extract control and management information from the communication network and place it at some higher level. This is now a generally acknowledged principle in the management of telecommunication networks

This paper presents the DCAN project whose aim is to design and develop the necessary infrastructure for the scalable control and management of Multi-Service Networks.

2 Introduction

The aim of the DCAN (Distributed Control of ATM Networks) project is to design and develop the necessary infrastructure for scalable control and management of multiservice Networks.

DCAN starts from the premise that control/management functions of the various devices - typically ATM switches - that make up such a network should be extracted from the devices themselves and delegated as much as to possible to external workstations dedicated to that purpose.

By devolving the management and control of switches and end devices into a coherent external system, one escapes the need for switch and device manufacturers to capture these functions within their equipment. This enables customers to configure their systems in a manner suited to their needs. In addition, it enables new management and control policies to be introduced in the network without modifying every switch, and thus will allow legacy and new signalling and management systems to co-exist.

DCAN also presumes that the control/management of multiservice networks is necessarily distributed due to, amongst other reasons, the need for end-to-end resource allocation in order to give guarantees about Quality of Service (QoS).

Finally DCAN takes into the account the fact that extracting control functions from the networks device and placing them into workstations requires us to be able to place real-time constraints on the execution of the software performing those control functions.

Thus to be successfully, DCAN must address both real-time OS and Distributed Processing Environment issues.

It is intended that the designed and implemented infrastructure be scalable. Scalable here does not just refer to size; rather it means being able to cope with a range of system capabilities, including simple devices with limited functionality. Thus, while retaining compatibility with developments in telecommunication system management, the emphasis is on ATM networks used to interconnect workstations and simple devices.

Lastly, we should state our view that management is to networking, what debugging is to programming: a necessary evil that we would like to avoid as much as possible. Rose in [Rose 91] call this the fundamental axiom: *The Impact of adding networking management to managed nodes must be minimal, reflecting a lowest common denominator.* This axiom underpins management within Internet, but the desire for the network devices to spend as much of their time as possible transmitting information and as little time as possible doing management related tasks, is clearly true of all networks.

3 Project Overview

3.1 Motivation

In what follows we distinguish from the two related areas of management and control. Control:

1. is performed on a short time scale, management operation may takes minutes or even hours to perform;
2. does not require the intervention of a human operator, management in general does;
3. is management policy independent, i.e. control functions are carried out regardless of what management strategy is being used.

E.g. call set-up is a typical control function and alarm correlation a management function. Although, control and management to some extent overlap and are often referred together simply as *network management* it is still useful to make a distinction between them.

Current developments in ATM Networking are leading to the notion of devolved control in which a number of switches are controlled by a largely separate distributed system as shown in figure 1.

This approach is in line with current thinking in Intelligent Networking (IN), whereby typically the actual routeing of a call is achieved outside of the network by some higher level entity which then uses knowledge about the type of call being made in order to determine what script to run in order to perform the service e.g. freephones charge the callee rather than the caller.

In addition, it resonates with the approach to telecommunication management as specified by the ITU-TS Reference Model for Open Distributed Processing (X.901-4), which defines a Telecommunication Management Network (TMN) which manages the underlying telecommunication network.

However, both IN and TMN standards have been defined with a Public Switching Telecommunication Network in mind. Within such a wide-area environment one may assume that network elements are autonomous and powerful; neither processing power nor implementation complexity are primary limitations.

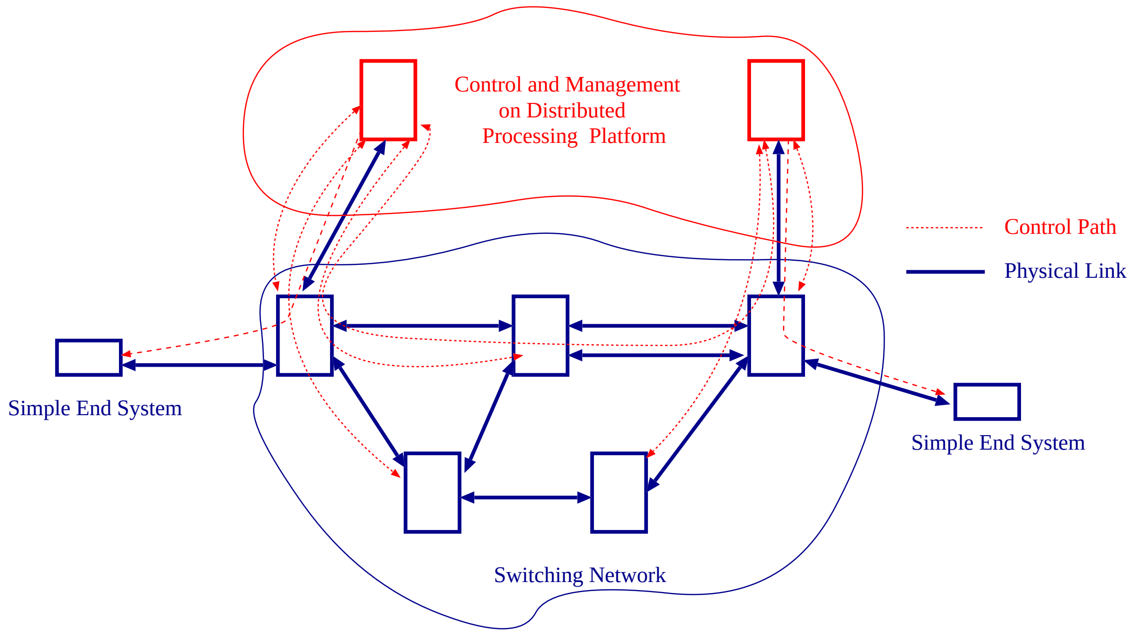

Figure 1: Devolved Management and Control

In local ATM networks, this assumption may not hold. Switches may only perform the relaying of cells (unicast or multicast) with even simple connection and network management functions performed by an external (distributed) system. There may also be end system elements, such as an ATM camera, which are so simple that they cannot perform network signalling and thus may not be able to initiate communication. They can be thought of as slaves.

Furthermore, the IN and TMN standards have defined complex protocols between the managing and managed layers. The rational seems to be that all embracing protocols will enhance their adaptability to new services. For managed elements to be able to communicate with a manager using such protocols require to have some knowledge of the management strategy.

We take the view that the protocols between the manager and the network should be minimalist. Additional management functionality e.g. synchronisation between streams, can be added in the management domain. See [Crosby 95] for an investigation of the issues involved.

We propose to develop a system which allows the distributed management platform to encompass a range of intelligences and to make as few assumption as possible about the management strategy to be used.

3.2 Project Architecture

The work divides naturally into four related, but distinct domains:

1. development of a suitable platform in which to run management and control applications and some test control/ management applications in order to demonstrate the concept;
2. development of a suitable means of communication between these applications;
3. development of a suitable simple protocol for the communication of management and control information between the network devices and these applications;

4. construction of a network consisting of devices - ATM peripherals and switches - capable of being managed by the above.

These areas of work are now detailed:

3.2.1 Platform

Control applications are in general real-time applications. Thus they must run in an environment in which precise guarantees can be made about the allocation of resources to the applications.

Cambridge Computer Lab. has developed a multiservice operating system called Nemesis [Roscoe 95]. Nemesis addresses the problem of multi-plexing the resources of the OS by doing as much processing as possible in the application domain and having the minimum functionality possible in the kernel and shared servers.

Nemesis has certain aspects which make it suitable for DCAN, namely:

1. Nemesis was designed from the beginning to provide quality of service guarantees to applications, and thus does not suffer the limitations of Chorus as reported in [Blair 94];
2. a major goal in the Nemesis design was the support of high speed communication, not merely distribution;
3. the naming and distribution model in Nemesis is consistent with the ANSA model - see later -, indeed the interfaces within Nemesis are specified in a language derived from ANSA IDL;
4. Nemesis is still in development with major design features defined and implementation in progress. While this has some disadvantages, it means that the implementation of DCAN can still influence design decisions within Nemesis.

Nemesis shall serve as the OS for executing time critical DCAN control applications. However, Nemesis as of now is not a distributed OS. In addition, the means by which Nemesis applications view each other is unique to Nemesis.

DCAN views control and management of multiservice networks as necessarily a distributed problem. Hence we must have some way for DCAN applications running on different workstations to interact with each other.

ANSA is an architecture for Open Distributed Processing [APM]. ANSAware is an example implementation of that architecture which allows the development of distributed applications in a heterogeneous environment. ANSAware/RT is an extension of ANSAware which offers some support in the development of Real-time systems.

ANSA has been very influential on the work of the CORBA/OMG standard and the work that the ANSA Consortium are doing now can be expected to influence the future direction of that standard.

Although ANSAware/RT has been developed for an OSF operating system and depends on POSIX threads, the ANSA architecture itself is better suited for a Nemesis type operating system. In fact, in some sense Nemesis is the ideal OS on which to run ANSAware. Thus the porting of ANSA/RT to Nemesis appears as a natural initial solution to the problem of distribution.

The initial DCAN platform is thus a modified version of ANSA/RT running over Nemesis.

The features of the ANSA implementation over Nemesis will include:

- a socket-like API for ATM to support a connection per bound interface, having no copying overhead, and providing temporal and cross-channel synchronisation;
- distributed end-to-end QoS negotiation and binding using ANSA streams;

- application programmer's interface and tools based on the ANSA signals model and synchronous programming.

The final result will be a distributed control platform and supporting services which will be scalable in both resource requirements and size from telecommunication networks, through LAN switches controlled by intelligent end systems, down to simple slave devices.

3.2.2 Inter Application Communication

Obviously communication between applications running on different workstations will rely on the underlying Network. Hence at a certain level such a communication will involve the transmission of ATM cells. Hence we must define something analogous to Berkeley sockets as found in BSD Unix and its derivatives, built over ATM.

However, a multi-media application can be characterised by having to service many logical circuit endpoints, giving quality of service guarantees to each. This requires a fundamentally different approach to buffer management and connection management than the Berkeley Sockets model that dominates current applications level network interfaces. Thus, we shall also need a stream based management protocol.

This will allow the development of both Remote Procedure Call (RPC) and continuous stream communication between applications. It shall be possible to associate a QoS with both an RPC invocation and a continuous stream. The provision of the ATM communication infrastructure over Nemesis will produce what could be termed a distributed ATM operating system.

It is intend that communication be possible between applications on a variety of Operating Systems including: Nemesis, Unix and Window's NT. Clearly QoS constraints imposed on applications running non real time OSs such as Unix can only be facultative.

The interface between the application and the network we term the ATM API. The definition of an ATM API, although a clear requirement for a management platform should be approached as a more general problem. We see the ATM API as an important standardisation target and contribution to the successful development of portable multi-media ATM-based applications.

3.2.3 Management/Control Protocol

The distributed control of ATM switches will require the design of a distributed management system with control interfaces to currently available switches based on their often proprietary protocols, for example, using SPANS to control FORE switches. This work will allow a coherent management system to control a heterogeneous mix of ATM switches which may well have a mixture of capabilities and intelligence. In the case of a low level system this would allow workstations themselves to provide the control facility (including signalling) for a very simple switch. By using ANSAware we can allow both this low level strategy as well as a higher performance approach in which the control system runs over ANSAware running on an embedded system with time constrained performance guarantees.

This philosophy can be extended to the the control of slave devices. A simple protocol for the control of the AVA-200 has already been developed by Nemesys. Although this protocol works well, it lacks industry acceptance and support. This workpackage will use ANSA principles to examine and where necessary improve the protocol. The goals here are to retain the simplicity to ensure that simple devices can implement the protocol (and thus avoid implementing a complete ATM signalling system) while making it applicable to a range of devices. The work will include the development of a front end to translate from an ANSA domain to the simple protocol.

While a simple protocol currently exists for the control of the FORE Systems AVA200, this was designed largely in isolation, rather than fitting into an overall management structure. Although we have argued for an approach of different intelligences with translation front ends in between, there is still the opportunity to define the control protocol knowing that this translation will be occurring. Moreover, while the AVA200 is the only existing simple ATM device to which we have access, we must ensure that the protocol designed will be general enough to be applicable to a range of devices, such as set top boxes used when delivering video over ATM to the home.

The ANSA architectural principles will be applied to the design of the protocol. Our current view is that we will make the protocol consistent with meta-signalling proposals arising from the ITU-T over the next year, since this is likely to be the only other simple protocol which ATM end systems will use.

4 Background

In this section we look at some of the standards bodies and consortiums that are working on issues related to the management of advanced networks.

4.1 ITU-T Standards

The ITU-T standards are concerned with the Public Switching Telecommunication Network.

ITU-T divides management into five functional domains:

1. fault management i.e. the logging and treatment of alarms;
2. configuration management i.e. the exploration and definition of the topology of the network;
3. accounting management i.e. calculating the charge that customers are required to pay;
4. security management i.e. ways of protecting against hostile connections to the network;
5. performance management i.e. ways of ensuring that the network is performing in the desired way.

The ITU-T standards define a network whose responsibilities are the management of the underlying telecommunication network. This network is called a Telecommunication Management Network (TMN). Certain of the nodes of the network are abstractions of physical or logical network elements (NEs), others contain implementations of management functions (OSFs).

The nodes of a TMN communicate with each other using the CMIP protocol over an OSI protocol stack, such an interface is generally called a Q3-interface. During an exchange between nodes we consider that one node is an agent and the other a manager. These are only roles, so a manager during one operation, may become an agent in another. It should be noted that a given node within a TMN maybe (and probably will be) itself distributed.

The vision that a manger has of the management data in its agents is defined in terms of managed objects using the GDMO interface description language. The manger manipulates this data using CMIP. The CMIP protocol offers 5 different forms of data exchange between manager and agent. The manager can perform Create, Read, Update and Delete operations on the MOs of the agent and the agent can send notifications of Events to the manager. A protocol offering these basic operations is often termed CRUDE. It should be noted that CMIP allows the initiation by the manager of a predefined action on the agent.

The TMN is organised hierarchically into five layers:

1. the element layer;
2. the element management layer;

3. the network management layer;
4. the service management layer;
5. the business management layer.

An entity which manages a set of agents in a lower level, may itself be considered to be an agent by a lower level.

Figure 2 gives a schema of the interaction between the entities of the TMN and the Public Switching Telecommunication Network. It shows two layers of the TMN: the element layer and the element management layer.

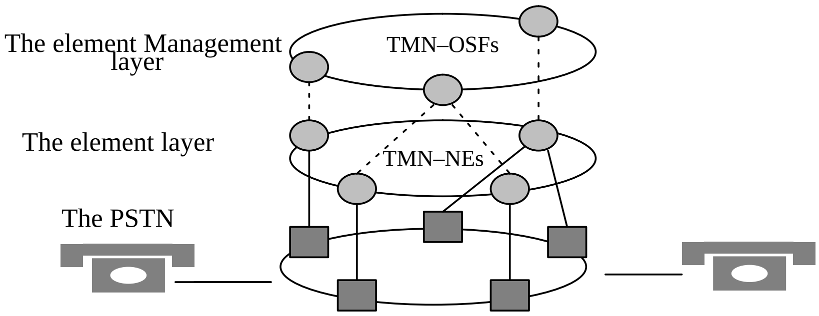

Figure 2: Schema of a Telecommunication Management Network

4.2 ATM forum standards

The ATM forum has been concerned with the management of customer installations and is thus complimentary to that of the ITU-T which is concerned with public networks.

The ATM forum recognise that it is too early to give a complete solution for the management of ATM equipment so they have defined what they call the Interim Local Management Interface Specification (ILMIS) for the control and management of private ATM equipment and their connection to the public network.

ILMIS uses the SNMP (Simple Network Management Protocol) as its basic means of exchanging management information. SNMP is similar to CMIP, in as much as it is CRUDE based, however SNMP and its successor SNMP-2 are both much simpler than CMIP. Some care has to be taken when comparing SNMP and CMIP because they use the same terminology to identify similar but different things, for example in both the management information is defined in terms of Managed Objects (MO), but while in CMIP the MO is a true object in the sense that it has both state and behaviour, in SNMP the MO has only state.

The solution proposed is the placing of a Management entity at each User-Network Interface, this entity is called a UME and is capable of replying to SNMP about the state of the equipment that makes up that interface and also potentially of generating SNMP requests to ask other UME about their state. A piece of equipment has an agent which is capable of accessing all the UME of that equipment and of replying to management requests from a distant Network Management System (NMS) The connection between the NMS and the agents may be performed using SNMP or some other method.

Figure 3 shows a simplification of the definition and context of the Interim Local Management Interface Specification (ILMI)

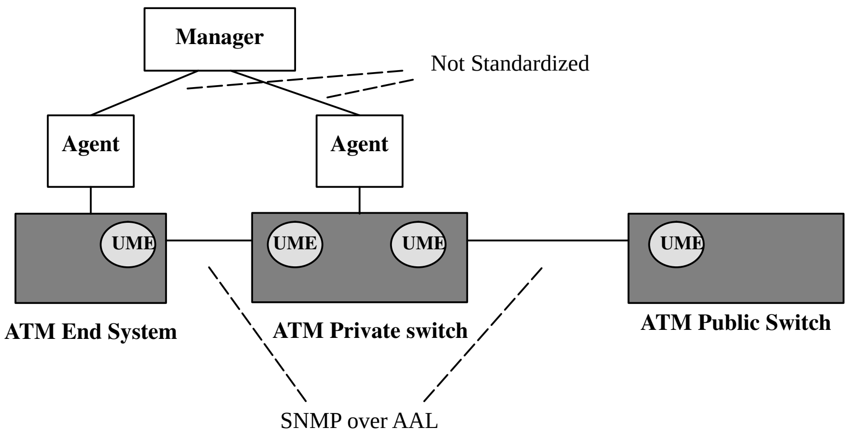

Figure 3: Schema of ILMIS

4.3 TINA-C

TINA-C is a consortium of telecommunications operators and equipment vendors collaborating on an architecture for Telecommunications Information Networking, particularly in the context of broadband systems.

TINA-C funds a core team, based in the USA, using seconded staff. In its first year TINA-C developed a baseline architecture drawing heavily on ISO RM-ODP, ITU TMN and Intelligent Network standards.

The TMN standard we considered previously defines how a management system should interact with the underlying network in order to perform management operations, but has nothing to say about how this is to be implemented. The objective of TINA (Telecommunication Information Networking Architecture) is to offer a flexible software architecture for access and management services, allowing the rapid introduction of new services and the ability to manage these services. It draws on both Telecommunication standards, such as TMN and Intelligent Networks (IN) and distributed computer standards such as CORBA/OMG .

APM as custodian of ANSA and RM-ODP, has cooperated closely with the TINA-C core team to ensure TINA-C is aligned with ANSA principles and concepts. Many TINA-C members are using

ANSAware in auxiliary projects alongside the core team effort. A majority of the ANSA sponsors are TINA-C members.

In their work plan for 1994-5, the TINA-C core team are focussing on consolidation of their architecture together with animations and laboratory sized experiments to validate the architecture. They have explicitly identified the need for ANSA input on explicit binding, streams, signals, quality of service negotiation, dependability, performance, naming and federation to help them in their task. TINA-C has defined concepts relating to service and connection management which are compatible with ANSA and will be followed in this project, with an emphasis on lightweight efficient scalable implementations.

TINA emphasizes the notion of service. A service is achieved by the interaction of a set of service components. A service component may belong to one of three categories:

1. service level components;
2. resource level components;
3. element level components.

The element level components are abstractions of the underlying physical resources, the resources level components are responsible for the management and control of the element level components and the service level components define the logic of the services themselves. In addition there are components which allow the infrastructure for the exchange of information between other components e.g. a notification service.

Figure 4 shows a schema for the TINA architecture with some typical components indicated.

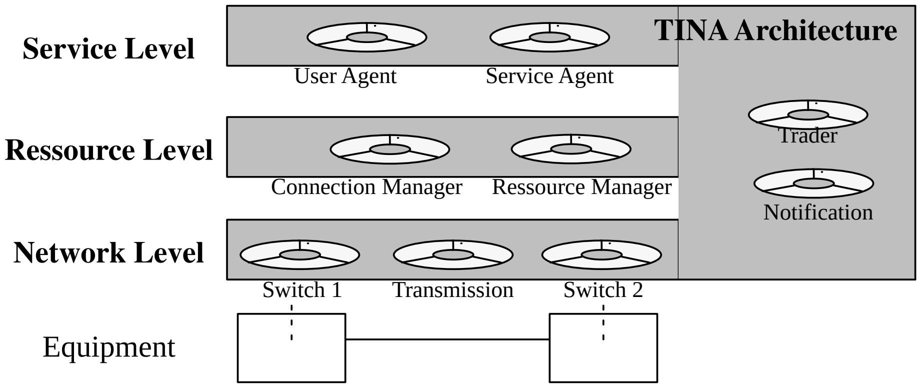

Figure 4: Schema of TINA

Management in TINA can be divided between:

1. computing management;

2. telecommunications management.

Computer management involves the management of the computers, DPE, operating system etc. Telecommunications management is organised very much in the way of TMN. One of the objectives of TINA is to be able to provide a high level management interface in order that managing applications can ignore the differences between these two very different types of management.

TINA has considered issues related to the management and exploitation of multimedia networks allowing for things such as Quality of Service negotiation. However, TINA is some what restricted in that it is mainly concerned with the Public Switching Telecommunication Network and with ensuring the inter-operability with existing standards such as TMN.

5 Project Management

The Consortium

The consortium is composed of APM Limited, Nemesys Research Limited and the University of Cambridge Computer Laboratory.

APM Limited, whose origins can be traced to the ANSA Alvey project begun in 1984,is a pioneer in the field of distributed computing. It has developed the ANSA architecture for distributed systems and ANSAware, an implementation of the architecture which runs on a number of vendor operating systems. It was founded as a limited company in 1989.

Nemesys Research, founded in 1993, is an ATM end system manufacturer. Its current product is the AVA-200, a device which takes multiple video and audio feeds and allows them to be sent over an ATM network. This device is manufactured in the UK and exported to the US and Far East through Fore Systems and in Europe through K-NET Limited. An integral part of the product is the software required to both control and present the video/audio data sourced by the AVA. Support is currently provided for UNIX, with Windows 3.1 and Windows NT under development.

The University of Cambridge Computer Laboratory has a great deal of experience in distributed computing and ATM networks. It developed the original IPR which led to the AVA-200 which has been exploited by Nemesys Research.

The Schedule

DCAN started in mid 1995 and is set to finish in mid 1998.

References

[APM] Architecture Projects Management Limited
ANSAware/RT 1.0 Manual, APM Technical Report APM.1441.00.04.

[Blair 94] G Blair, A Campbell, G Coulson, D Hutchison, M Papathomas and P Robin
On the Implementation of a Quality of Service Controlled ATM Based Communications System in Chorus
4th International Workshop on Protocols for High Speed Networks, Vancouver, Aug 1994

[Crosby 95] S. Crosby
Performance Management in ATM Networks
PhD dissertation May 1995

[Leslie 91] I. M. Leslie and D. R. McAuley
Fairisle: An ATM Network for the Local Area.
Computer Communications Review, 21(4):327–336, September 1991.

[Rose 91] M. Rose
The Simple Book, an Introduction to Management of TCP/IP based Internets
Prentice Hall 1991

[Roscoe 95] T. Roscoe
The Structure of a multiservice Operating System
PhD dissertation April 1995