ISSUES IN REALISING THE TINA NETWORK RESOURCE ARCHITECTURE

George Pavlou
University of Surrey
UK

Telma Mota
Portugal Telecom/CET
Portugal

Frank Steegmans
Alcatel Telecom
Belgium

Juan Pavón
Universidad Complutense, Madrid
Spain

ABSTRACT

TINA aims to provide an architecture based on distributed computing technologies that will enable telecommunications networks to support the flexible introduction and control of new advanced services. The provided services and the supporting network and computing infrastructure will be managed in an integrated fashion. Network management aspects in TINA are addressed by its Network Resource Architecture (NRA), which has so far concentrated on Connection and Configuration management, with Fault, Accounting and Performance management aspects to follow. VITAL is a European collaborative research project aiming to validate a long-term telecommunication architecture based on TINA, through the design, realisation and trials of a large-scale system. This paper presents the VITAL work related to the realisation of the network resource architecture in the areas of resource configuration and connection management. The problems and issues encountered are highlighted and the architectural solutions adopted are presented, together with examples of operation that illustrate better the design choices for the realisation of the TINA-based NRA system.

KEYWORDS:

MANAGEMENT, CONTROL, TINA, TMN, ATM, IP, NETWORK ARCHITECTURE

1. INTRODUCTION

The Telecommunications Information Networking Architecture (TINA) has defined a Network Resource Architecture (NRA) [1] that covers the management areas of Connection, Configuration, Fault, and Accounting Management. Some of these areas have been defined in a great level of detail (e.g. Connection Management) while some others have only started being designed (e.g. Fault and Accounting Management) and some are not yet defined at all (e.g. Performance Management).

VITAL (Validation of Integrated Telecommunication Architecture for the Long-term) is a European collaborative research project of the ACTS (Advanced Communication Technologies and Services) programme that aims to validate a long term Open Distributed Telecommunication Architecture (ODTA) based on TINA. The concepts and specifications developed by TINA have been followed as much as possible in areas where the relevant models and specifications were relatively mature. In other areas, new models and specifications were developed, feeding back modifications and extensions to TINA.

This paper intends to present a summary of the work and the issues related in realising the TINA Network Resource Architecture. The problems encountered and the solutions adopted by VITAL are presented. In addition, examples are given in order to illustrate better the design of the NRA telecommunication software system. Some TINA concepts are introduced in the paper but the reader should be familiar with the basic aspects of TINA, the Network Resource Architecture and the relevant terminology.

In summary, this paper has the following structure. Section 2 discusses the status of TINA network resource architecture. Section 3 presents the design and realisation of VITAL network resource architecture. Both sections comprise separate subsections for resource configuration, and connection management. Section 4 presents scenarios concerning the deployment of the NRA system for networks of Asynchronous Transfer Mode (ATM) technology. Section 5 presents a scenario applied to Internet IP while Section 6 presents a scenario applied to mixed ATM and IP networks. Finally, the paper closes with a summary and conclusions.

2. CURRENT STATUS IN TINA

The overall TINA architecture is layered in the following fashion:

- at the lowest layer, computer and network resources represent the manageable entities;
- these entities are manipulated by Network Control and Management Logic organised in the Network Resource Architecture - this is the focus of this paper;
- end-user services are offered through Service Control and Management logic, organised in the Service Architecture (SA).

Since the overall architecture is object-oriented, manageable resources, control and management logic are realised through distributed objects that communicate across the ubiquitous Distributed Processing Environment (DPE). The basis for the TINA DPE is the OMG Common Object Request Broker Architecture (CORBA).

The NRA covers element and network management functions according to the hierarchical decomposition proposed by the Telecommunications Management Network (TMN). In addition, connection control aspects are addressed by the Connection Management NRA subsystem. In TINA, the TMN configuration management functional area is split into two sub-areas:

- Resource Configuration Management (RCM), which deals with the configuration of static resources; and
- Connection Management (CM), which deals with the configuration of dynamic network resources i.e. connections. In fact, CM replaces the traditional signalling procedures for on-demand switched connections.

The TINA approach in general represents a paradigm shift from the traditional telecommunications culture towards distributed object-oriented software principles. The following two sections provide a view of the TINA status in the areas of resource configuration and connection management, highlighting also some of the additions and modifications in VITAL.

2.1 Resource Configuration Management

While the functionality of Resource Configuration Management in TINA has been identified for quite some time, the current state of the relevant specifications are still rather immature. The latest specification [1] represents a complete change compared to the previous one [8]. The VITAL specification has taken into account input from all TINA specifications prior to [1] and has introduced additional concepts and models. The approach is documented in [9] and has been fed back to TINA for the next version of the Network Resource Architecture [1].

In the general case, RCM applies not only to network but also to service and computing resources. The main requirements of RCM for managing those resources are the following:

- RCM should maintain an inventory of resources under its influence and manage their relationships in the form of a resource map;
- it should ensure the map is updated with newly installed or deleted resources;
- it should allow the activation, deactivation, reservation and release of resources;
- and it should support automatic resource installation through the DPE services wherever possible.

A key aspect of RCM is the maintenance of resource related information. In the TINA NRA, network resource information is specified in the Network Resource Information Model (NRIM) [6]. The latter follows object-oriented modelling principles and uses a variation of the TMN Guidelines for the Definition of Managed Objects (GDMO) language.

The NRIM model has been used in connection management for modelling dynamic resources such as trails and subnetwork connections. This information is distributed in the various computational objects i.e. the Connection Performers (CPs), without a single unified view being available. Such a view though is absolutely necessary in resource configuration management. This means that a key RCM requirement is a unified view of static network resources, allowing RCM clients to build a global view of the overall network. In TMN systems, such a global view is provided by network management Operations Systems (OSs) which allow flexible navigation of the managed objects they administer. Since the TINA NRIM model follows TMN specification principles, we could provide TMN-like access mechanisms to TINA computational objects that administer NRIM-based information objects. This should allow for the navigation of their relationships in a flexible fashion. Such a generic access interface is provided by OSI System Management (OSI-SM) which is the access technology used in the TMN.

While TINA uses a CORBA-based DPE, it is possible to mirror OSI-SM capabilities in CORBA and provide a generic computational interface for the resource map. The adoption of this approach means we can exploit TINA NRIM specifications by mapping them directly to information objects handled by the resource map. In addition, the use of an OSI-SM-like computational interface implies that resource maps can be easily federated due to the hierarchical nature of the relevant information model. While this type of interface is powerful and generic, clients of the resource map may provide simpler, task-oriented interfaces, targeted to particular activities as described in [9]. It should be clear by now why the current TINA RCM models were not be used.

An additional key architectural addition to the TINA RCM is the consideration of the management resources themselves, i.e. the computational objects offering TINA management services, in addition to the network, service and computing resources. For example, the Connection Management Configurator (CMC) is a computational entity that sets up and manages the Connection Management computational objects, based on topological information. Such an object is a manager for management resources and should belong to a new domain defined in VITAL, the Management Configuration Management (MCM). Subsequently, the other RCM domain considered is the Network Topology Configuration Management (NTCM) domain which deals with the management of static network resources. In summary, the VITAL RCM modelling has started from existing concepts in TINA but modified and extended them to create a design suitable for a real prototype. The computational architecture of the VITAL RCM system is described in section 3.1.

2.2 Connection Management

Since the early days of TINA-C activities, Connection Management has attracted particular attention. Considered the core of TINA Network Resource Architecture, it was intentionally decoupled from the resource configuration management area. While this area deals with management of static resources, Connection Management addresses the management of dynamically allocated connectivity resources.

TINA connection management follows a hierarchical architecture, influenced from TMN-based hierarchical decomposition of management activities. At the top of the hierarchy, a Connection Coordinator (CC) object offers the Connectivity Service (ConS) reference point [2] to NRA service users in a technology independent fashion. At the next level of the hierarchy, the Layer Network Coordinator (LNC) is a technology specific object, managing an administrative domain of a common technology. It is also responsible for federating requests to other domains for end-to-end connectivity services. The LNC uses Network Management Level and eventually Element Management Level Connection Performers (NML-CPs and EML-CPs respectively), organised in a hierarchical fashion. A view of a simple connection management system for a small network comprising two network elements is shown later in Figure 4.

CM has been defined and specified in a great level of detail in [3, 4]. The *graph* information model was first proposed to represent connectivity but later on it was shown to be rather ‘heavyweight’. Therefore, TINA has proposed another information model, based on the first one but simpler and more appropriate from the terminology point of view. However, some important mechanisms have not yet been included in the latest official Network Resource Architecture [1].

Currently, Connection Management in TINA is satisfactorily defined regarding the general mechanisms required for setting up and releasing connections. Unfortunately though, it still contains a number of deficiencies. Some of them have already been resolved, for example the large number of operations required to provide simple functionality, but some others still remain. An overall approach on how to handle Quality of Service (QoS) in a technology independent manner, is an example of the remaining incompleteness of the relevant TINA specifications. QoS generic negotiation mechanisms need to be introduced at different levels of the Connection Management Architecture, as well as at the service management level [11]. Another open issue, is the mapping between stream flows, which are by nature unidirectional, and network flows which might be bi-directional. The model proposed by TINA does not explicitly preclude this mapping but does not explain either which is the most suitable approach. The mapping has to be flexible enough to allow different possibilities of supporting one or more stream flow connections over a single network flow connection (e.g. ATM VP). As an alternative, it maybe useful to send a stream flow over two different network flow connections, for security reasons for example. In addition, the use of Reference End Points (RFEPs) is ambiguous. This is also the case with the whole procedure of resolving the Stream Flow End Points by associating Network Flow End Points belonging to different layer networks. The use of special resources (e.g. Bridges, Interworking Units) might be needed in order to connect End Points located in different Layer Networks, enabling for example the use of different access network technologies and their interaction with one or more ‘core’ networks. Among the existing transport technologies, networks making use of the Internet Protocol (IP) are prime examples of connectionless networks. The TINA NRA has specified the mechanisms for the management of different connection-oriented transport networks (e.g. ATM, SDH) but it did not address the management of communication sessions over connectionless networks. Furthermore, Quality of Service based routing of connections, load balancing and performance issues have not yet been addressed. Finally, the most important limitation in current TINA specifications, which is pertinent not only to the Network Resource Architecture, is the lack of a consistent way of defining computational object behaviour.

For an implementation and validation project like VITAL it was very important to find a common understanding of the behaviour of TINA computational objects. A compromise between a generic but complex approach and a more specific, simple but comprehensive solution has been followed. In summary, VITAL did not intend to solve all the open issues but it has identified some important inconsistencies, proposed some solutions and provided input to TINA Network Component Specification (NCS). An important outcome though is that despite the deficiencies described above, this realisation demonstrated the feasibility of the TINA Connection Management Architecture (CMA).

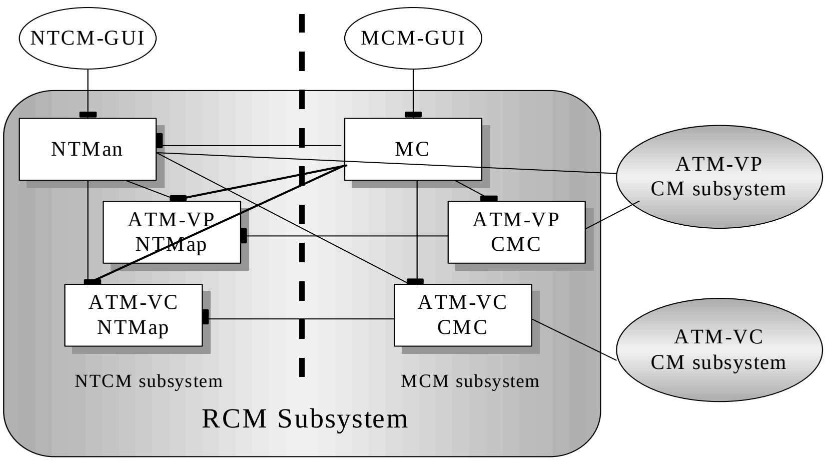
3. THE VITAL NETWORK RESOURCE ARCHITECTURE

3.1 Resource Configuration Management

The computational design of VITAL RCM follows TINA documents prior to NRA 3.0 [1] with some deviations and novel aspects documented in [7, 9]. A key deviation with respect to TINA is the division of VITAL RCM system into two subsystems, Management Configuration Management (MCM), managing management resources, and Network Topology Configuration Management (NTCM), managing network topology resources. In addition, the computational interface of the network topology map follows TMN principles, as it was explained before. The computational architecture of the RCM subsystem is depicted in Figure 1.

Both subsystems have a "master" manager computational object; the Management Configurator (MC) in the case of MCM and the Network Topology Manager (NTMan) in the case of NTCM. These objects are responsible for the operation of the respective subsystem. MCM comprises also the Connection Management Configurator (CMC) computational objects. There exists one CMC instance for every layer network within that domain, responsible for the configuration of the relevant computational objects in the Connection Management system i.e. network and element management layer connection performers. This configuration is tightly coupled to the topology of the layer network, as provided by the NTCM subsystem.

The NTCM comprises instances of the Network Topology Map (NTMap), one per layer network, in the same way to the CMCs in the MCM subsystem. Each NTMap manages the topology map of one layer network. There exist two layer networks in VITAL, ATM Virtual Path and ATM Virtual Channel, in a server-client relationship. As it was already mentioned, the NTMap is influenced by TMN methodologies and specifications in the sense that it provides a generic, Common Management Information Service (CMIS)-like computational interface. This allows clients to navigate the information objects that constitute the map in a flexible fashion. The NTMan provides a task-oriented, higher



Legend:

CM	Connection Management	MCM	Management Configuration Management
CMC	Connection Management Configurator	NTCM	Network Topology Configuration Management
GUI	Graphical User Interface		
MC	Management Configurator	NTMan	Network Topology Manager

Figure 1: The VITAL Resource Configuration Management Architecture

abstraction interface, with operations tailored to the needs of its client objects. The NTMan checks also the validity of clients' requests and thus maintains the integrity of the maps. This is particularly useful in the case of server-client networks, where a change in one layer / map potentially means changes to another layer and map respectively.

The VITAL RCM considers two cases of configuration: initial and dynamic reconfiguration. The initial configuration is a total configuration of the Connection Management system, carried out immediately after the bootstrapping of the system and reflecting the initial state of the network. The dynamic reconfiguration reflects any subsequent changes made to the map, with the consequence of reconfiguring partly the Connection Management system. A scenario for dynamically adding or removing a ATM VPC and the necessary reconfiguration are presented in section 4.1.

It should be finally mentioned that RCM supports federation. A limited view of "neighbouring" networks is modelled in the network map, including routing information regarding the address prefixes supported through those domains. This information is used to configure the connection management system, which provides federation across different administrative domains.

3.2 Connection Management

The set-up of a connection and negotiation process

The computational model implemented in VITAL follows the one proposed by TINA [1]. The TINA Business Model and Reference Points defined in [5] were also taken into account. The proper behaviour has been added and a minimal set of operations and parameters have been implemented in order to coherently set-up and release connections over both a typical connection-oriented transport technology (i.e. ATM) and a common connectionless network (i.e. Internet IP) . Because of the latter case, some modifications to the TINA model were necessary.

Whenever a connectivity session is required, the Communication Session Manager (CSM) is contacted in the retailer domain (see Figure 4). As a result of the stream binding realised at the service level, i.e. the Service Session Manager (SSM) level, a set of connections is required in order to accomplish the service session demands. These connections are explicitly requested to the CSM. This computational object is responsible for negotiating the common information needed to run a session with the terminals involved. The type of codification of the stream flows (e.g. a_law for audio, MPEG for video), the session protocol and the possible transport protocols may be negotiated between the CSM and the Terminal CSM (TCSM) objects. At the consumer domain, the Terminal Device Controller (TDC) object manages the mapping between the codec information and the transport protocols. In VITAL, multiple choices (i.e. terminal capabilities [11]) can be offered to the CSM; it is up to this computational object to find a common agreement between the different participants in the communication session. The QoS negotiation process takes into account the available capabilities of each terminal and the preferences mentioned explicitly by the user. The high level QoS information is decided at the service level, the codec's are negotiated at the CSM level and once agreed, the Connection Coordinator (CC) will choose the underlying transport technologies, based on the network flow end points specified by the terminals and the availability of network resources. Some information is directly passed to a particular Layer Network Coordinator (LNC), which is the first technology dependent component in the CMA hierarchy.

The Connection Coordinator (CC) computational object is the contact point at the connectivity provider domain. In a similar fashion to the CSM, it is a generic technology-independent object and is instantiated at each connectivity provider domain.

The underlying object group composed of the LNC and its subordinate objects, i.e. the Connection Performers (CPs), is always technology specific; in TINA at least one instance of these object groups must exist for each technology domain. In the consumer domain, the object that takes care of the technology-specific information is the Terminal Layer Adapter (TLA).

The model described above was designed by TINA to handle connection-oriented networks using concepts, such as trails and subnetwork connections, based on similar concepts defined by the ITU-T in the G.805 and M.3100 recommendations. Since these concepts are not applicable to connectionless networks such as the Internet, the LNC object has been generalised in order to accommodate the Layer Network Binding (LNB) concept [12]. The LNB is a generalisation of the trail concept and represents an end-to-end connectivity across a generic layer network that could be either connection-oriented or connectionless.

The role of each computational object of the CMA becomes now clear; at the Retailer domain the CSM handles the communication session relevant information together with each TCSM at the consumer domain i.e. high level media QoS negotiation and mapping between the End Points at the stream flow level and the network level. The CC is the interface contact point at the Connectivity Provider Domain and handles the connectivity session related information; it chooses the underlying transport technology and respective protocol and can negotiate special resources (e.g. bridges and interworking units) in order to ensure the interworking between different transport networks. The LNC is responsible for managing one particular layer network, representing either a connection-oriented or a connectionless transport network and can federate with other LNCs located in other administrative domains.

The Federation Mechanism in VITAL

Federation provides another way to structure connection management systems. Instead of hierarchical relationships between computational objects, it defines peer-to-peer relationships. This implies that computational objects involved in a federation relationship provide the same interfaces to each other.

Federation is useful for two purposes:

1. Cooperation between different interworking domains. Different administrators can own and manage sets of resources or domains. In order to be able to offer end-to-end services, these administrators must cooperate so that connections can be set up and managed across domains.
2. Federation can be used to avoid a deep hierarchy of CP computational objects within a single administrative domain. This is a consequence of the fact that federation provides essentially a mechanism for the horizontal decomposition of network control.

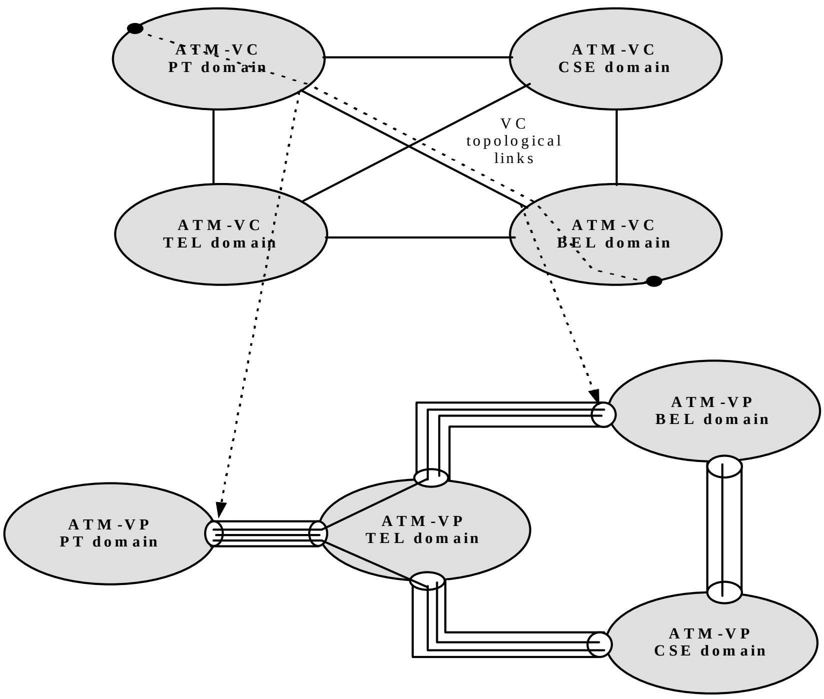


Figure 2: Federation at VP layer network supports federation at VC layer network

TINA defines the LNFed (the Layer Network federation) inter-domain reference point for interaction between different Connectivity Providers. Federation between domains provides more flexibility for routing, de-couples different implementation strategies for each domain and provides a means to support interworking with legacy systems and protocols (this last point is developed further in [10]).

Four network operators have been involved in the VITAL demonstrator, as shown in Figure 2 and Figure 3: Belgacom (BEL), CSELT (CSE), Telefónica (TEL) and Portugal Telecom (PT). Each operator provided a local ATM island while these islands were interconnected over the JAMES pan-European ATM infrastructure. In order to make TINA services available across all domains, there is a subsequent necessity to support the establishment of connections across those different network domains.

Relationships between different layer networks should be considered when defining federation. As an example, consider the ATM VC and VP layer networks shown in Figure 2. VP trails established in the VP layer network appear as VC topological links in the VC layer network. This implies that a “logical” VC layer network can be constructed through VP trails established over the underlying physical network.

ATM VC topological links are established using trails in the ATM-VP layer network. If a VP trail is set up from the Portugal Telecom (PT) to Belgacom (BEL) domain, this will appear as a topological link in the VC layer network between the corresponding domains as shown in the upper part of Figure 2. This situation implies that a request for a VC trail from a point in the PT domain to a point in the BEL domain passes transparently through the TEL domain. However, if such a VP trail is not set-up, the VC trail will be established involving also the TEL domain in the VC layer network as shown in Figure 3. These considerations are useful when defining federation relationships between network operators since they have direct implications on the enterprise viewpoint of the overall system.

Federation relationships are supported by the LNC computational object. Each LNC object offers the interfaces to create, control and destroy tandem connections through the corresponding domain. In principle, a LNC may be requested to establish a

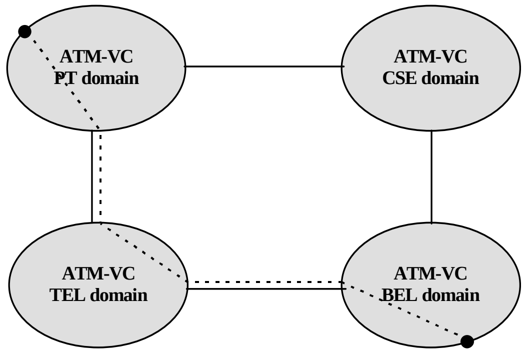


Figure 3: Different federation relationships

trail with an endpoint in other domains. The LNC analyses which part of the overall trail belongs to its own domain, and requests its corresponding NML-CP to establish and set-up that part. It also requests a neighbour LNC of a federated domain to establish the remaining part. This relationship may be recursive, continuing until the end-to-end trail has been set-up across all the necessary domains.

Routing decisions are made based on routing information that uses the prefix of the target address to decide which neighbour LNC to contact. Since federation relationships may change, LNC routing tables can be modified through the configuration interface, by specifying the name of the neighbour domain, the topological links between domains, and the list of network addresses which are reachable through that domain (these can be specified in a “group form” through their prefixes).

Operations and parameters defined at the interface offered to the CC are also considered in the federation interface, with the addition of parameters which define a negotiation policy between the domains. These parameters determine, for instance, how accounting is to be performed, or whether monitoring information on the relevant tandem connection is needed.

4. SCENARIOS APPLIED TO ATM NETWORKS

The following scenarios illustrate the interactions among the computational objects described above. Two steps are considered. First, the configuration of the resources needed to set-up and release connections. Second, the procedures for providing the connectivity required by the session level objects. In order to make clear the control of connections, a simple network configuration is considered in which only one connectivity provider domain is involved.

4.1 Configuration Scenario

In this scenario we consider the dynamic reconfiguration of connection management subsystems when adding or removing a new ATM Virtual Path Connection (VPC). As explained in section 3.1, the system is initially configured to reflect the initial topology of the network, both at the Virtual Channel (VC) and Virtual Path (VP) layers, which have a client-server relationship.

Initially, there are no VPCs at all i.e. the logical network over which VCCs are established is the same as the underlying physical network. The initial VPCs are configured according to network planning activities. From then on, the network operator may add new VPCs or remove existing ones according to changing requirements.

The process of adding or removing a VPC could be initiated by an automated program, such as a network planning / load balancing tool, or by a human network manager. In our example we assume the latter though the procedure would be the same in the former case. The steps involved in this process are the following, referring to the computational components of Figure 1.

1. The NTCM-GUI retrieves and displays the current network topology from the VP and VC NTMap objects through the NTMan. The network operator may then request the addition / removal of a new VC topological link or VP trail through the graphical interface.
2. The NTCM-GUI passes the request to the NTMan, which checks its validity by accessing information in the VP and VC NTMap.
3. In the case of a valid request, the NTMan asks the actual request implementation to the LNC of the VP Connection Management system. The latter implements it by passing it through the CPs and RAs to the actual ATM cross-connects.
4. Upon successful completion, the NTMan updates the VC NTMap with the topological link change. The NTMan subsequently notifies the VC CMC about changes on the VC layer network topology.
5. The VC CMC reconfigures the VC CM subsystem to take into account the topological change. From then on, VCC requests to it will be served according to the new topology of the VC layer network.

In this scenario, the addition and removal of VPCs is initiated through the RCM system for planing reasons as explained. In the general case, end-to-end VPCs could also be provided to consumers in the context of Virtual Private Network (VPN) services.

4.2 Connection Set-up Scenario

One of the main issues for implementing TINA CMA was the mapping of information streams to network connections. A related issue was the mechanism through which computational objects in the Consumer Domain (TCSM and TLA) negotiate connection related parameters (e.g. QoS, VCI) with those located in the Connectivity Provider Domain (CSM and LNC). It is also important to highlight that the higher level components (i.e. SSM, CSM and CC) are not aware of technology specific information (e.g., VCI values); this is handled by the Consumer Domain computational objects and the low-level CMA components (i.e. LNC and CPs). It is the responsibility of the CC to choose the underlying transport technology through the appropriate LNC. The scenario presented in this section focuses on high-level mechanisms rather than technology specific ones. In addition, federation is not described due to its complexity but more details can be found in [10].

1. Initially, the SSM requests to the CSM the creation of a communication session and specifies how stream flow end points need to be interconnected. The stream flow endpoint information contains also the TCSM references and the required service-level QoS.
2. The CSM then contacts each TCSM object located in each terminal in order to know the characteristics of the corresponding stream flow end points e.g. media flow, codification, network-related protocols, layer network and network address. The CSM then performs the negotiation between the different terminals. It looks for commonalties among the parameters provided by them and makes the final decision

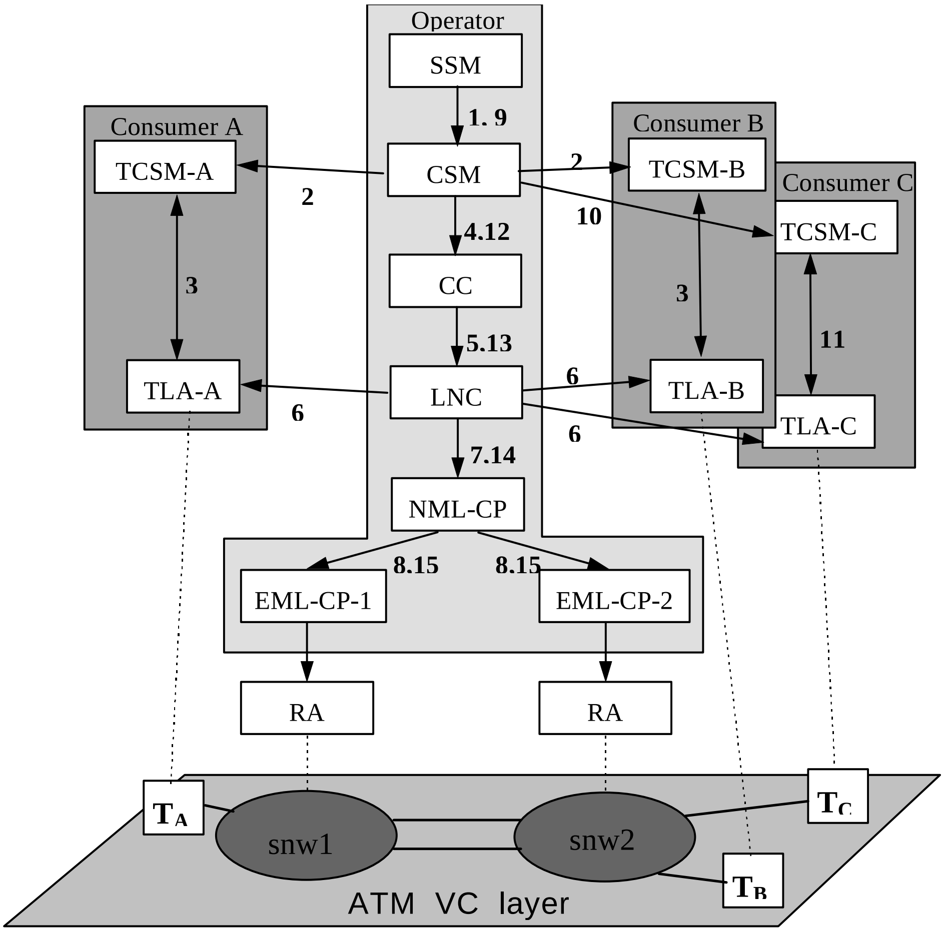


Figure 4: Connection set-up scenario

taking into account the fact that the source has the highest priority.

3. While each TCSM is a repository of the existing stream flow end points, a TLA keeps track of the technology dependent information, which in this example is ATM specific. The layer network type, the network address and the transport protocols are then requested by the TCSM to the TLA. Later, when the VCI is chosen by the LNC in co-operation with the CP objects, the TLA helps the TCSM to resolve the mapping between stream flow end points and network flow end points.

4. The CC chooses the layer network based on constraints or/and requirements arising from consumer domains, and filters the network flow end point information in the sense that only end points of the chosen technology, which is ATM in this scenario, will be sent to the respective LNC.

5. The LNC receives a request from the CC for the provisioning of a layer network binding. The CC passes the network flow end points that encapsulate network technology specific information (e.g. addresses of the terminals, references of the associated TLA objects and other transport network specific parameters i.e. QoS).

6. The parameters mentioned above allow the LNC to contact the TLA objects and negotiate the VPI/VCI values, together with the connection parameters (e.g. bandwidth, etc.), which are supported by the terminal for that connection. If a terminal is attached to a network end point in the scope of other connectivity provider domain, the LNC would delegate the negotiation with the

corresponding TLA to a federated LNC. The LNC is prepared to suggest some values of VPI/VCI and quality of service to the corresponding TLA.

7. The LNC then requests the top NML-CP in its administrative domain to create a subnetwork connection among the identified network termination points at the boundary of its domain.
8. An NML-CP is responsible for an aggregated subnetwork and hence deals with routing functionality in this subnetwork. After routing the requested subnetwork connection in its own subnetwork, the NML-CP will request the set-up of subsequent subnetwork connections to the CPs responsible for the subnetworks through which the initial subnetwork connection is routed. Such a CP may be either another NML-CP or a EML-CP.

Eventually, each aggregated subnetwork is broken down to the level where each subnetwork represents a switch. The EML-CP controls the use and reservation of the resources associated to the switch. In the scenario presented here, we have one ATM NML-CP controlling a subnetwork that is composed of two other subnetworks representing an ATM switch that is controlled via two ATM EML-CPs.

Up to this moment, resources have been reserved but they have not yet been allocated in the real switches. This happens when there is a request to the CSM to activate the connection graph (or some specific part of it). This request goes from the CSM down through the CC, LNC, NML-CP, and, finally, EML-CP. A Resource Adapter (RA) class of objects is used for interacting with the real switches. RA objects provide to EML-CPs independence of the vendor specific interfaces supported by the switches. This allows the use of the same CP objects for switches of different vendors, as it is the case in VITAL, where each operator has its own, some of them with proprieraty interfaces.

Once the connections in the switches are established, multimedia flows can be transferred between the different terminals involved in the communication session.

Operations 9 to 15 in Figure 4 add a new branch to a point-to-multipoint connection. The CSM must re-negotiate the terminal capabilities with the parties involved. In a similar fashion to the other terminals, the technology dependent parameters (including QoS) are negotiated at the LNC-TLA level. The mapping of service level QoS parameters to network related ones is again performed within the terminal.

5. SCENARIO APPLIED TO IP NETWORKS

As already explained, the technology specific information related to the network flow endpoints is only used at the LNC-TLA level. Therefore, the setting-up, deletion, activation and deactivation process is generic and independent of the underlying network technology at the higher levels of the Connection Management Architecture, i.e. the CSM, TCSM and CC level.

At least one LNC in the Connectivity Provider Domain and one TLA in each Consumer Domain exist per technology specific layer network. In other words, if the connectivity provider supports IP, there will be an IP LNC in its domain. If the terminal in a particular consumer domain supports IP, there will be a IP TLA. The same is true for ATM or any other network technology.

Up to the LNC, the scenario for IP is very similar to the ATM scenario. However, a significant difference between the two technologies arises at the layer network level. While the Network Resource Architecture manages the whole infrastructure down to the ATM switch, this is not the case for IP. This is due to the connectionless, best-effort nature of IP, which implies that no connection resources need to be managed within the network.

Hence, conceptual Layer Network Bindings (LNBs) are implemented through a mechanism that either compares to 3rd party call set-up, i.e. ATM in VITAL, or 1st party call set-up, i.e. IP in VITAL. Apart from being used for connectionless networks, the latter can also be used to encapsulate legacy signalling networks such as N-ISDN. The interface between the LNC and TLA contains operations only for channel negotiation at the network access in case of 3rd party call set-up. For 1st party call set-up, this interface is extended with operations that allow the management of complete multi-point connections.

When the IP LNC is requested to set-up a LNB, it will contact the respective TLAs. First the local Network Flow End Point (NFEP) is associated with the already established protocol stack. The IP socket is created based on, possibly modified, addressing information encapsulated in the NFEP. The resolved NFEP is returned to the LNC. This NFEP encapsulates the complete addressing information, i.e. IP address, protocol, socket number. These resolved NFEPs are being passed to the relevant TLAs and related to the local NFEP. Hence, each TLA has a view on how the local NFEP relates to the rest of the LNB. For example, the source terminal TLA will know the complete configuration of uni-directional point-to-multi-point connection, while the sink terminal TLAs will only know the source. For a connectionless transport protocol such as UDP, this establishes a 'virtual' connection. For a connection-oriented transport protocol such as TCP, the LNC will explicitly instruct the sink NFEP TLAs to establish the connections. Finally the LNC will send an activate request to all TLAs, starting with the sinks.

Although VITAL has decided to implement the IP layer network through 1st party call set-up, this does not mean that the network can not be managed. Managing IP networks by applying TINA concepts is an interesting area for future research.

6. SCENARIO APPLIED TO MIXED ATM AND IP NETWORKS

An interesting example is how the TINA CMA can be applied to mixed transport technologies, like ATM and IP in order to support a single service level end-to-end connection. The relevant architecture is depicted in Figure 5. The CC detects that End Points A and B belong to different Layer Networks, which in this case are of different technology. As a result, the CC locates an Interworking Unit capable of handling the interworking between the two layer networks and requests an interworking connection. The interworking unit is modelled as a Connectivity Provider domain terminal with the special functionality of forwarding data from one layer network to the other. As soon as the IWU provides the connection, it returns the end points A’ and B’. The CC proceeds by requesting the set-up of two Layer Network Bindings using those end points to the IP and ATM LNCs respectively. The LNCs handle these requests as any other and see no difference between an interworking unit and a regular Consumer Domain terminal.

This example shows that the selection of transport networks and interworking units can be perfectly implemented using the TINA NRA. It could be applied, for example, to connect applications on terminals using Internet LANs as access networks to core networks or to interconnect applications on terminals connected via different technologies.

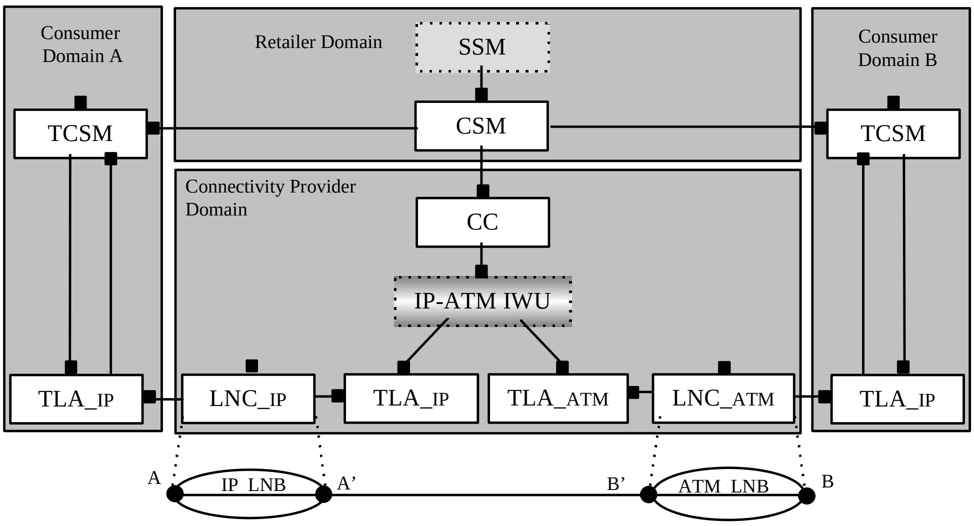


Figure 5: Connection set-up over mixed ATM / IP networks

7. SUMMARY AND CONCLUSIONS

The validation effort of the TINA architecture undertaken in the VITAL project has resulted in the identification of under-specified subsystems, components, behaviour and interfaces. In order to develop working services in a real multi-technology (i.e. ATM and IP) and multi-operator environment (for ATM), TINA models and specifications have been refined, modified and extended where necessary. This work is currently being considered by TINA-C core team and will be reflected in the final TINA-C specifications, in the concrete Network Resource Architecture [1], Connection Management [4] and Network Resource Configuration Management [8].

The VITAL realisation of the TINA NRA has focused on Resource Configuration Management and Connection Management. It is interesting to note that the models and specifications developed in VITAL for Resource Configuration Management are not only applicable to Network Resources, but also to the Service components of the TINA architecture, because of the generality of the relevant information and computational models. Another important VITAL result and contribution to TINA is the way application stream flows and QoS requirements are mapped onto network connections and specific transport network QoS parameters and the implications to the terminal architecture. Finally, VITAL experimented with transport technologies of different nature and QoS characteristics i.e. ATM and IP, enabling a rational selection of specific layer networks based on the requested QoS or the location of particular endpoints.

In summary, VITAL proved the feasibility, implementability and relevance of the TINA Network Resource Architecture as the underlying foundation for flexible, technology-independent multimedia multi-party telecommunication services. Aspects that need to be addressed by future research projects are scalability, adaptive routing / load balancing, co-existence with ATM signalling systems and use of IP integrated/differentiated services which will provide QoS guarantees.

ACKNOWLEDGEMENTS

This paper describes work undertaken in the context of the ACTS AC003 VITAL project, which is partially funded by the Commission of the European Union.

We would also like to acknowledge our colleagues in the VITAL project who did undertake design and implementation work related to the VITAL NRA.

REFERENCES

[1] TINA-C: TINA Network Resource Architecture, Version 3.0. TINA document no. TB_v3.0_97_02_10. February 1997.

[2] TINA-C: The ConS Reference Point, version 1.0. February 1997.

[3] TINA-C: Connection Management Architecture. TINA document no. TB_JJB.005_1.5.94. March 1995.

[4] TINA-C: Connection Management Specifications. TINA document no. TP_NAD.001_1.2_95. March 1995.

[5] TINA-C: TINA Business Model and Reference Points, version 4.0. Approved Public Version. May 1997.

[6] TINA-C: Network Resource Information Model Specification. TINA Baseline Draft. May 1996.

[7] VITAL: ODTA Validation Third Trial, VITAL D13 Deliverable. October 1998.

[8] TINA-C: Network Resource Configuration Management, version 2.0. October 1996.

[9] Pavlou, G.; Griffin, D.: Realising TMN-like Management Services in TINA. Journal of Network and Systems Management (JNSM), special issue on TINA. Vol. 5, No. 4, pp. 437-457. Plenum Publishing: 1997.

[10] Pavón, J.; Montes, E.: Network Provider Domain Federation. Proc. of TINA’97, pp. 334-342. IEEE Computer Society: 1997.

[11] Rajahalme, J; Mota, T.; Steegmans, F.; Hansen, P.F.; Fonseca, F.: Quality of Service Negotiation in TINA. Proc. of TINA’97, pp. 278-286. IEEE Computer Society: 1997.

[12] Steegmans, F.; Rajahalme, J.; Kim, H.C.; Narayanan, N.: Managing TINA Streams that use Internet Transport. Proc. of TINA’97, pp. 111-118. IEEE Computer Society: 1997.