Network Provider Domain Federation in TINA

Juan Pavón
Alcatel Corporate Research Center
Ramírez de Prado, 5, 28045 Madrid (Spain)
pavon@alcatel.es

Edgar Montes
Alcatel Corporate Research Center
Route de Nozay, 91460 Marcoussis (France)
montes@aar.alcatel-alsthom.fr

Peter Komisarczuk
Ericsson Ltd.
WC3, Burgess Hill, West Sussex, RH15 9UB (UK)
etlpkzk@etlxdmx.ericsson.se

Abstract

Federation in TINA CMA (Connection Management Architecture) provides the mechanisms for cooperation between different interworking network domains possibly owned by different administrators. In order to be able to offer services to their users, these administrators must cooperate. In this paper we present an implemented and validated architecture including the federation techniques necessary. We illustrate the problem based on experience from a User Trial, in which different operators, and suppliers with different equipment participate.

1. Introduction

The CMA (Connection Management Architecture) defined by TINA, specifies the functions necessary to manage network transport resources. The CMA gives a uniform view of these resources and provides a model for their control and management that is independent of the underlying network technology. These functions are implemented as distributed software components that are made available to applications that need to provide communication and network management services.

The Architecture is divided into three levels:

Communication Session level providing service independent interfaces so that service components can establish end-to-end communication in an abstract way.

Connectivity Session level providing technology independent interfaces for the above level so that it can interconnect network termination points. It also handles the interworking of different network technologies, our basic concern here.

Layer Network providing an abstract view of the specific network technology.

Federation in the TINA CMA provides the mechanisms for cooperation between different interworking network domains. Different administrators own and manage sets of resources or domains. In order to be able to offer services to their users, these administrators must cooperate so that connections can be set up, information exchanged, and management assured across these domains.

In TINA, the definition of inter-domain reference points guarantees that inter-operation between ADs (Administration Domains) will be possible. Specifically, in this paper we study and validate the LNfed interface (the Layer Network federation inter-domain reference point) that allows different Connectivity Providers to implement their own version of the Connectivity Platform and be able to interact to establish connections across their respective domains.

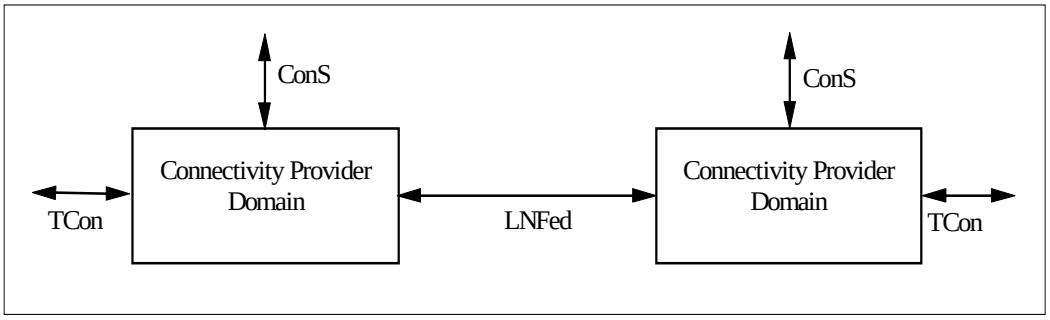

Figure 1. LNFed reference point between Connectivity Provider domains

Federation between Connectivity Provider domains provides more flexibility for routing, it de-couples different implementation strategies for each domain, and it provides a means to support interworking with legacy systems and protocols. As we will show later in this paper, the architecture implemented here not only allows interoperability between TINA compliant systems but is also a basis for the integration of legacy systems and protocols such as the existing broadband signaling systems.

Federation must not only establish connections between ADs, but it must also provide solutions to the control of management functions or systems across these boundaries. In this first implementation of the LNFed interface we only establish connections between multi-provider domains, but in the next phase we will investigate the problem of management services and present a solution which integrates these aspects.

In this paper we present an implemented and validated architecture *vis-à-vis* federation (another paper presents the overall project [2]). To achieve this we illustrate the problem based on experience from a User Trial, in which different operators, and suppliers with different equipment participate. Figure 2 shows the system developed by two manufacturers, Ericsson (on the left), and Alcatel (on the right), interacting through the LNfed interface.

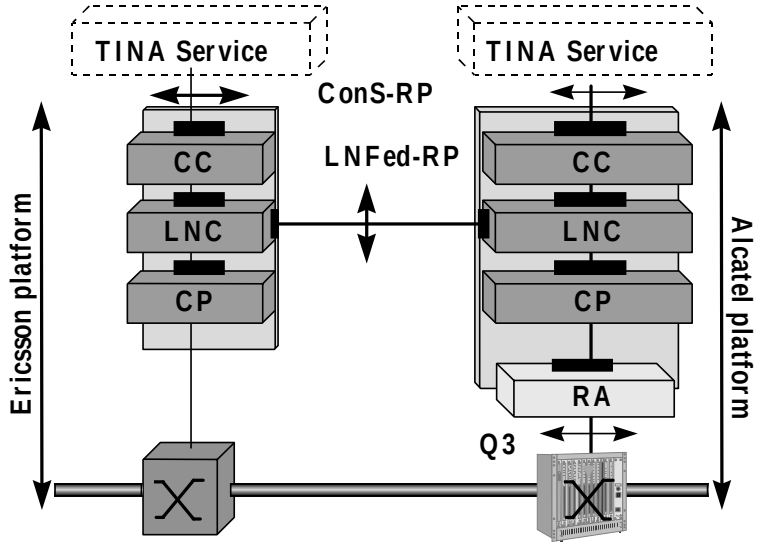

Figure 2. TINA Connection Management Architecture in SPOT

In section 2 we present the implemented components and illustrate how they work with an example scenario for the setup of a trail over two federated ADs. We then briefly describe, in section 3, the procedure used to test federation. In section 4 we

present some original solutions to interworking with legacy systems. Finally, in the conclusion we review the results obtained through this work, the possible extensions and the issues yet to be resolved.

2. Description of components

The TINA architecture has adopted an object-oriented approach that permits the division of the system into different computational objects (COs) that interact through well defined interfaces. Their interactions are supported by a distributed processing environment (DPE), which is based on CORBA. Thus, the Connection Management Architecture defined by TINA [1] consists of several COs of two different types, as shown previously in Figure 2:

- those that offer an interface to the TINA services, rendering them independent of the underlying network structure: basically these COs are the Communication Session Manager (CSM), and the Connection Coordinator (CC).
- those that allow connection management within a layer network; basically these COs are the Layer Network Coordinator (LNC), and the Connection Performer (CP).

Through the use of this architecture the services can request connections specified in the form of a connection graph (CG). The CC is basically responsible of giving an abstract view of the network connection, whereas the LNC serves trail and tandem connection requests on a layer network. The tandem connection requests allow LNCs of different domains in the same layer network to federate and establish a trail. The LNC acts as a single point of access to a layer network within an AD, through which is provided a simple means of access to the entire federated layer network, spanning many ADs.

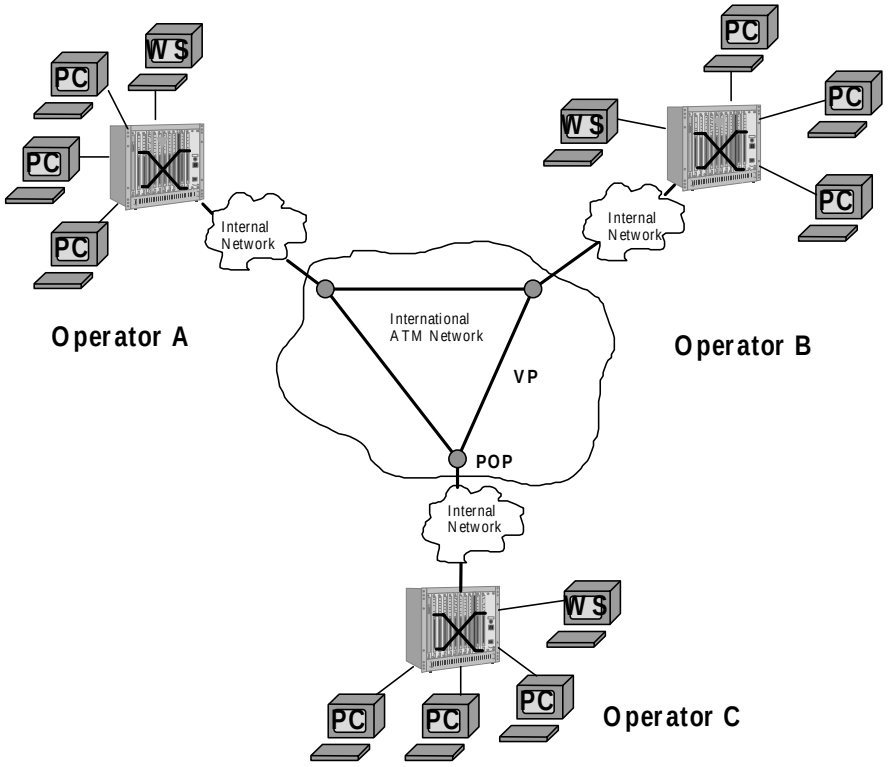

Figure 3. Scenario for LNC federation in SPOT

2.1 LNC and federation interfaces

When establishing a trail, portions of it may belong to different administrative domains; these, in turn, may belong to different connectivity providers. Network federation involves cooperation between different administrative domains so that the trail can be set up. To be able to implement the federation of layer

network domains, i.e., between LNC computational objects, we must first define the operations that are needed at the federation interface, which has been named LNFed. To formally define this interface we use the OMG Interface Description Language (IDL) [4] but could also use the extended version of this language defined by TINA-C (ODL) [5].

Figure 3 shows the network infrastructure that has been used to explore the LNfed issues. There are three ATM VC layer network domains (called Operator A, B, and C). These are interconnected through an international ATM network. This is the configuration considered in the SPOT project with Unisource, where the three operators correspond to Telia, Swiss Telecom, and KPN, all interconnected through the JAMES European ATM Network. In the first two domains, Alcatel 1100 HSS series 1000 ATM switches are used, and a FORE ASX 200WG switch is used in the third domain. Note that this configuration involves a heterogeneity of physical equipment, and the interconnection of a diversity of networks.

2.2 Configuration of LNC

For each domain there is one LNC computational object which serves requests for trails from clients (e.g., the CC), and serves requests for tandem connections from neighbor LNCs. Each LNC has to be configured with the information required to determine with which domains it is federated and to which domains it must delegate in order to set up tandem connections. Each LNC is a client of a NML-CP which will setup and establish subnetwork connections among points visible at the top level of the partitioning in the respective domain. In the figure some of these points are shown.

Initially, each LNC is configured in order to know which is the top NML-CP in its layer network domain, and which end points are visible at its borders (these include the access and transit end points). Information is also provided on the neighboring LNCs. More explicitly, at each of these LNCs the following information is provided:

- The federation interface reference of the neighbor LNC .
- The topological links [3] between the domains. Topological links between domains can be defined by pairs of link termination points (e.g., a121-c211, or a133-b132) including the bandwidth associated to the topological link.
- Routing information (e.g., based on prefix or explicit enumeration of ranges of end point identifiers, which is the solution that has been implemented).

For the neighbor LNCs that are to be served the LNC creates a federation interface on which to receive tandem connection requests. Note that neighbor LNC objects should be configured in a consistent way. Another possibility was to define an LNC-Federation server object, which is asked for by LNC objects when this kind of information is required. The problem with this second approach is how to determine the owner of such an object, and the risks of having to centralize this information.

Several open issues have been identified at this time, these include: how federation at the Resource Configuration Management level takes place, and what form of dynamic routing information may be exchanged with regard to network reachability. In the present trial scenario the planning of the configuration of federation interfaces, topological links, and routing information is considered off-line as part of the federation negotiation mechanisms.

2.3 Scenario for establishment of a trail between end points in domains A and C

The following scenario is based on the network configuration shown in Figure 3, and explains how a trail between different ADs is established. Operations that illustrate the scenario have been simplified to consider only the parameters which are relevant to the purpose of discussing federation.

A request to establish a trail can be accepted by any of the LNCs in the three domains. A way of improving communication resources usage is to make that an LNC analyzes the root, and delegates the construction of the trail to the LNC of the root's domain. In this example, let's consider that the LNC in domain A (LNC-A) receives a request from a CC to create a trail between two network access points a1 and c2. The CC requests the LNC to:

create_trail (a1, c2, type, conn_desc)

where *a1* is the root (origin) of the trail, which is a network access point associated to a terminal in the domain A, and *c2* the leaf (destination), which is a network access point associated to a terminal in domain C. The *create_trail* primitive indicates the type of trail required (point-to-point uni- and bi- directional, or point-to-multipoint unidirectional) and the connection

description structure which contains bandwidth description parameters.

LNC-A will determine first whether *a1* and *c2* are in its domain. In this example, *a1* is in its domain, but not *c2*. There should be some mechanism (a simple static routing table was used in the trial) which determines through which other domain the connection is to continue. In the example, LNC-A can route directly to domain C, or it can choose a route transiting domain B (for instance, when all topological links with domain C are busy).

Once LNC-A has determined to use domain C, it should decide which topological link to use. This can be determined by LNC-A and dictated to LNC-C, or it can be negotiated with LNC-C. A tandem connection primitive containing a negotiation parameter is used to create the tandem connection. If LNC-A, for example, chooses topological link *a121-c211* (*a121* being a link termination point in domain A and *c211* in domain C) but is prepared to negotiate, LNC-A can request LNC-C for a tandem connection using the following primitive:

create_tc (c211, negotiable, c2, type, conn_desc)

LNC-C will provide LNC-A with a confirmation of the use of *c211* (note that the link termination point is given as parameter) or propose the use of another topological link, e.g., *c212* (assuming there is a topological link *c212-a113*, for instance). In the case where LNC-A wants to use *a121-c211* and no other topological link, it would set the flag to non-negotiable status. As a result of the execution of this operation, LNC-C will return the following information:

- whether it has succeeded or not in creating (setting-up and establishing) the tandem connection
- which topological link has been used
- which link connection has been allocated in the topological link
- which network trail termination point (NWTTP) has been allocated to *c2*.

If the chosen topological link is different, LNC-A will have to confirm to LNC-C its use or release the tandem connection.

With this information, LNC-A can request its top NML-CP to establish the corresponding subnetwork connection in domain A.

Note that several strategies can be envisaged in order to improve the connection setup time. For instance, if LNC-C returns the link connection for the topological link between the domains without waiting to complete the tandem connection establishment, the subnetwork connection in domain A can be setup and established while the subnetwork connection is being setup in the other domains. However, the release would take more time in the case of a failure to create the tandem connection.

LNC-A will be able to return the network trail termination point (NWTTP) associated to *a1* and *c2* to its client. The NWTTP for *a1* is managed by its own domain whilst the NWTTP for *c2* was obtained as a return parameter of the operation invoked on LNC-C. In case that there is interactions with TLA objects (see section 4 below), LNC-A provides this information to TLA in domain A, and LNC-C to TLA in domain C. Note that this requires that references to TLA objects should be passed also in the interface together with the network addresses.

It is important to consider the similarity of these procedures with current signaling protocols.

3. Procedures used for testing LNFED

Several steps were followed to assure, as much as possible, that the federation would work when the final field trial was set up. First an agreement was reached concerning the LNFed IDL. Next local tests were performed by configuring locally defined domains. These test also permitted the elaboration of scenarios to be used for the final test between the different systems (Ericsson's and Alcatel's). But before establishing a direct connection through the international network, executable LNC-Fed servers were exchanged including dummy CPs (see Figure 2) so that the test scenarios could be performed and the results in form of traces could be verified. By dummy we mean a component that has de same behavior than the final component but only prints traces of the generated requests to the underlying element. The following figure (figure 4) illustrates this test configuration. Here we desire to create a connection from termination point A to B where A is in one domain and B is in the domain of the other network provider. The resulting behavior tested was as follows:

CC requests a trail from the LNC. The LNC determines that the end point is in another network provider domain or that it can be reached through this other domain. The LNC creates the trail up to the network interworking point B'. The LNC requests a tandem interface

from the other LNC. The LNC requests a trail from the interworking point terminating at the remote termination point B.

In this way we can ensure that most of the problems that may arise have been eliminated before establishing a physical connection and also verify that the system is properly configured.

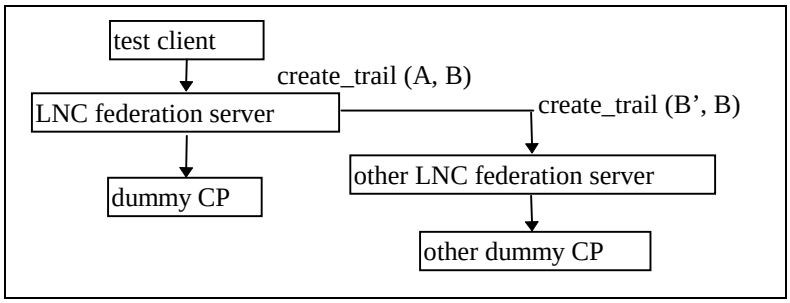

Figure 4. Configuration of initial test

4. Interworking with legacy systems

Interworking of LNC is considered with the customer premises equipment when connected to UNI based networks, and, secondly, at the interworking points between legacy and TINA networks. Two interworking scenarios between legacy and TINA networks are considered, these are interworking between broadband networks and the TINA CMA, and between TMN systems and the TINA CMA.

4.1 LNC at the Customer Premises Network (CPN-LNC)

An interesting consequence of this study is the possibility to consider that the customer premises equipment (e.g., the terminal) may have the capability to offer an LNC federation interface located on the TLA (Terminal Layer Adapter) [3]. Previously the TLA was called CPE-LNC (Customer Premises Equipment-Layer Network Coordinator) and one exists for each type of layer network in the consumer domain so that network flow endpoints can be created and managed. This is one potential way to provide interworking with current access signaling systems. Note that in the case UNI signaling is used, an adapter should be used in the same sense as discussed in the next section.

Current TINA-C specifications consider that this interface consists of three operations, from LNC to TLA, in order to propose the use of a trail termination point and some bandwidth and quality of service parameters. On reply from TLA, LNC should check whether TLA accepts or propose a new trail termination point and connectivity description parameters, and confirm the agreement of the TLA parameters.

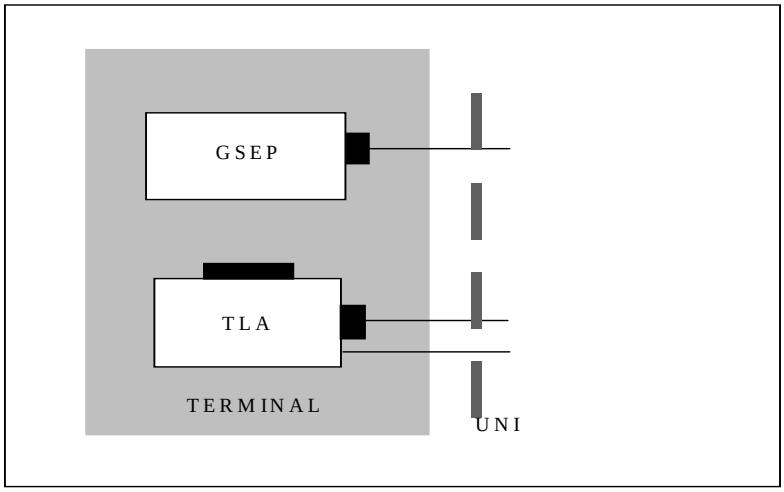

Figure 5. LNC in the terminal

4.2 Broadband Signaling and LNC interfaces

Current broadband signaling protocols are described in ITU-T Recommendations:

- Q.2764 (point-to-point)
- Q.2722.1 (point-to-multipoint)

The main difference with respect to the TINA approach is that both service and connection management information are described together (although there are some claims to separate call and connection models). Anyway, some primitives of the above protocols can be considered similar to those primitives we have identified for layer network federation.

The scenario for interworking TINA CMA (and also TINA service architecture) with current signaling can be based on some signaling adapter object as depicted in Figure 5. Note that current signaling protocols transport both call control and connection control information. Therefore, the Signaling Adapter computational object could offer interfaces to both LNC and Service Session Management objects, although the possibility of using TINA call control on an end-to-end basis can also be employed. In this proposal the LNC interface with the Signaling Adapter is the same as for federation with other LNCs.

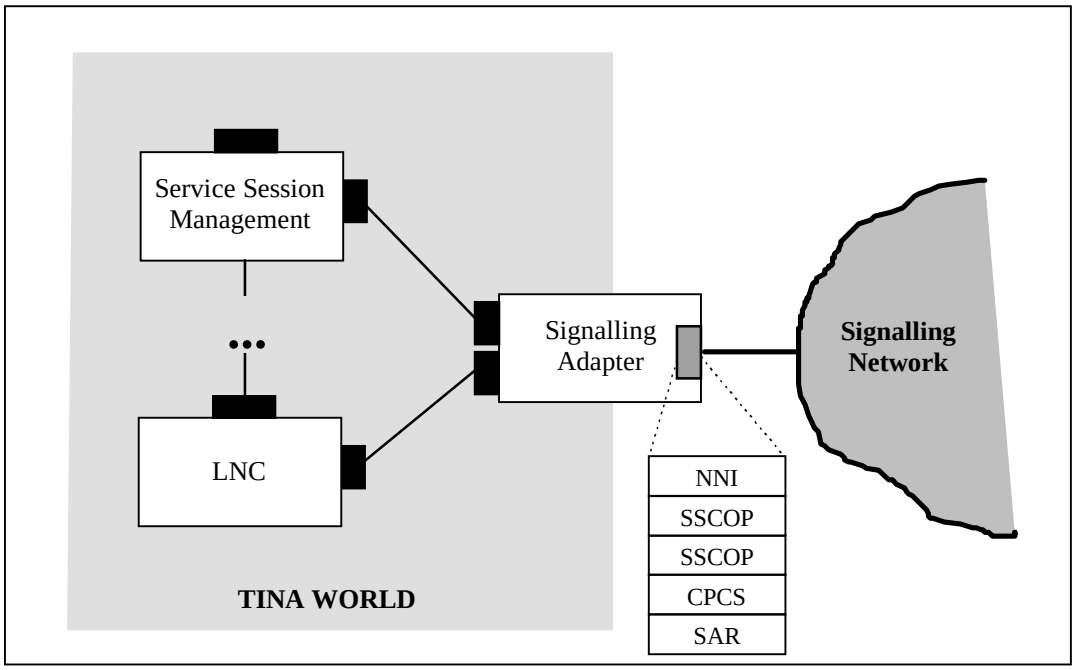

Figure 6. Interworking with current signaling protocols

4.3 TMN and LNC interfaces

An important issue to consider is interworking with TMN systems. The TINA Connection Management Architecture has been defined with a hierarchical nature similar to the one applied for TMN layering [1]. In the case of federation between TINA and TMN domains, we are interested in two types of interfaces for federation between domains:

- Federation for the establishment of a trail.
- Federation for configuration.

It is the federation requirements and interfaces for configuration which require a detailed analysis. This is needed also in the case that we plan to use federation using signaling, as each domain needs to be configured in order to determine through which neighbor domain it has to create a tandem connection.

A possible solution is to consider X-Coop. The next figure shows an interworking and federation scenario using an X-Coop adapter.

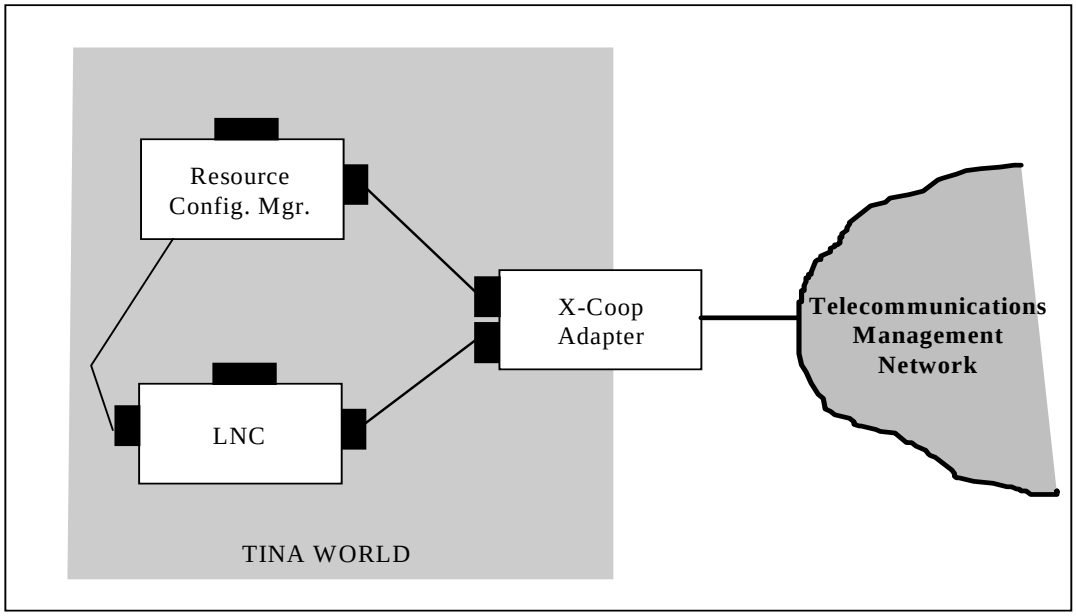

Figure 7: Interworking with management systems

5. Extensions to the LNFed interface

The basic LNFed interface described in the preceding sections has provided a mechanism for connection creation across Layer Network domains, and some potential legacy system solutions have been proposed. However connection creation is not the only function that the LNFed interface must deal with. These issues include routing information exchange, federation management, and topological link management (i.e. the application of FCAPS). From experience with work on the ConS reference point (TINA reference point between the components of the Service Architecture and the Connection Coordinator introduced in section 2) the creation of interfaces including FCAPS is non trivial and it has not been considered in this paper. Likewise federation management interfaces are not considered, however the exchange of routing information is briefly discussed in this section.

From a connection management perspective the routing infrastructure within the Layer Network will require information about network reachability, or better still it should be in a position to obtain, from neighboring domains, or from some third party, the information required to choose a route. The routing information obtained plus the local fault and performance data, and any local routing policies will affect the routing choices within the Layer Network. The fault and performance information will also have some bearing on routing information propagated by the Layer Network for use by other Layer Networks.

Within this prototype, simple static routing has been used in order to test the LNFed connection management interface, however this is clearly not the most practical mechanism for real networks. Two routing architecture choices are briefly proposed in this section, firstly a hierarchical model and secondly a peer-to-peer model. Both models require the existence of a “Router” within the LNC that acts as the source and sink of the inter Layer Network routing information.

The Hierarchical Model

In the hierarchical model a routing information hierarchy is created with knowledge of all the layer networks and their interconnections. The routing hierarchy can be based on the same principles as the hierarchical TINA Connection Management Architecture where each Layer Network forms a managed element in the routing hierarchy. Routing information updates are propagated up through the hierarchy, originating at routing elements, and periodically processed routing tables are disseminated back to the routing elements. The routing information updates can be based on fault conditions and/or CAC

traffic levels exceeding predefined levels, as well as on the internal Layer Network congestion levels which could be passed as a "cost" parameter for types of through traffic. The routing hierarchy could also be provided with dynamic pricing information with the Layer Network routing information, so that shifting price considerations can be taken into account when routing connections. The scope of the routing information could be worldwide or localized to the immediate neighboring networks. The availability of worldwide information implies that the preferred route of a connection could be specified in the LNFed tandem connection request.

Obviously the routing hierarchy would need to be part of a neutral telecommunication infrastructure, similar to the Broker architecture to ensure fair play. Alternatively the architecture could be employed for a set of federated Network Providers. The scope of the information provided by the hierarchy could range widely, depending on the information that Network Providers are willing or able to contribute to the routing hierarchy.

The Peer-to-Peer Model

In the peer-to-peer model routing information between adjacent networks is propagated between the Routers. This model reflects the routing paradigm found in the Internet between Autonomous System (AS) which use Inter Domain Routing Protocols (IDRP) to exchange routing information about network reachability. The IDRP routing protocols used are classified as either distance vector or link state routing protocol. The BGP, border gateway protocol, defined in RFCs 1105, 1163, and 1267 is the most versatile IDRP protocol in general use on the Internet. It is a distance vector protocol with a path string to eliminate routing loops and policy based routing. It provides a full routing information exchange between AS's so that optimal routes can be chosen and load sharing enabled. It is feasible to port the BGP to an IDL interface for routing information exchange between Layer Networks. The Routing interface would require the addition of a federation parameter to limit the scope of the routing information propagation so that Federation boundaries could be honored.

6. Conclusions

The purpose of the work presented in this article was to define and implement a solution on how to manage federation between different domains and determine the extensions that are necessary. Its intention is to contribute to the TINA requirements to define what is needed concerning federation. The initial aim of this work, which was to achieve connectivity across Network Provider domains, has been achieved.

The first version of the system developed and used in the trials contained a number of simplifications. We concentrated mainly on allowing different network providers to federate in order to provide a complete connection through different domains. There was no topological link negotiation, no accounting, fault management and recovery. These issues will be treated in future work since they form part of our desire to show that the TINA architecture can be a framework for providing industrial quality services and connection platforms that adapt well to existing networks.

It was found that although the model can solve the trail creation problem at the software level, there are always hardware problems to consider in order to interface with legacy networks. Thus, it is also our intention to further study the implications, with respect to interworking with legacy systems, that might exist on the federation interface.

7. References

[1] J. Bloem et al., *TINA-C Connection Management Architecture*, in TINA'95, Melbourne, Australia, 13-16 Feb. 1995, conference proceedings vol.1, pp. 485-494

[2] J. Bengtsson, P. Hellemans, L. Lehman and N. Mercouroff, *Implementation of Services for Computer Supported Cooperative Work on TINA: The SPOT Project*, to be published in ISS'97, Toronto, Canada

[3] *Network Resource Architecture*, version 3.0, TINA-C Baseline NRA_v3.0_97_02_10, February 1997.

[4] *OMG, The Common Object Request Broker: Architecture and Specification*, rev. 2.0, July 95

[5] *TINA-C, ODL Manual*, ver. 2.3, June 96