Project Number :	**IST-1999-10561-FAIN**
Project Title :	**Future Active IP Networks**

Definition of evaluation criteria and plan for the trial

CEC Deliverable No :	**WP5-ETH-027-D6-Apr**
Deliverable Type :	**PU**
Dissemination:	**Pub**
Deliverable Nature :	
Contractual date :	**November 30, 2001**
Actual date :	**December 21, 2001**

Editor :	**Bernhard Plattner**
File Name	**WP5-ETH-027-D6-Apr.doc**
Contributors :	**See list of authors**
Version :	**1.1**
Version Date :	**December 21, 2001**
Internal Distribution :	**WP5**
Deliverable Status:	**Final release**

The FAIN Consortium consists of:

Partner	Status	Country
UCL	Partner	United Kingdom
JSIS	Associate Partner to UCL	Slovenia
NTUA	Associate Partner to UCL	Greece
UPC	Associate Partner to UCL	Spain
DT	Partner	Germany
FT	Partner	France
HEL	Partner	United Kingdom
HIT	Partner	Japan
SAG	Partner	Germany
ETH	Partner	Switzerland
FOKUS	Partner	Germany
IKV	Associate Partner to FOKUS	Germany
INT	Associate Partner to FOKUS	Spain
UPEN	Partner	USA

The FAIN Consortium

University College London	(UCL)
Josef Stefan Institute	(JSIS)
National Technical University of Athens	(NTUA)
Universitat Politecnica De Catalunya	(UPC)
T-Nova Deutsche Telekom Innovationsgesellschaft mbH	(DT)
France Télécom / R&D	(FT)
Hitachi Europe Ltd.	(HEL)
Hitachi Ltd.	(HIT)
Siemens AG	(SAG)
Eidgenössische Technische Hochschule Zürich	(ETH)
FOKUS Fraunhofer Institute for Open Communication Systems	(FOKUS)
IKV++ GmbH Informations- und Kommunikationstechnologie	(IKV)
Integracion Y Sistemas De Medida, SA	(INT)
University of Pennsylvania	(UPEN)

Project Management

Alex Galis
University College London
Department of Electronic and Electrical Engineering,
Torrington Place
London WC1E 7JE
United Kingdom
Tel ++44-(0) 207- 679 5738
Fax +44 (0) 207 388 9325
E-mail: a.galis@ee.ucl.ac.uk

Authors

Yannick Carlinet (FT)
Lawrence Cheng (UCL)
Spyros Denazis (HEL)
Dusan Gabrijelcic (JSIS)
Alex Galis (UCL)
Richard Gold (FhG)
Juan Luis Mañas González (INT)
Chiho Kitahara (HIT)
Tuan-Quoc Nguyen (DT)
Bernhard Plattner (ETH) - Editor
Arso Savanovic (JSIS)
Alvin Tan (UCL)
Julio Vivero (UPC)

Executive Summary

This document outlines the plan for the evaluation of the results achieved in the FAIN project. The methodology for evaluation as set forth in the description of work is (1) to create a test-bed, in which the system components developed in FAIN are deployed as prototypes and (2) to perform a qualitative and quantitative evaluation based on experiments carried out on the test-bed. This document therefore contains a description of the test-bed, a scenario which will be used to demonstrate the active service deployment capabilities of the FAIN architecture, a set of criteria to be applied for qualitative and quantitative evaluation, and a plan for practically establishing the FAIN test-bed.

It is to be noted that this deliverable is meant to provide a long-range evaluation plan, i.e. it basically covers the time frame up to the end of the project. Therefore, the scenario descriptions reflect the whole complexity of setting up an active virtual private network for a customer, deploying a service for a specific end-user and the management of the corresponding service level agreements. However, this time frame also contains two important milestones for the FAIN project, M5 (April 2002) and M6 (April 2003). This document therefore also serves as the blueprint for the demonstrations to be carried out at M5 and M6. Since the FAIN technology is not fully developed yet, it will however not be possible to completely specify the functionality that will be demonstrated for these milestones. Nevertheless, the functionality shown for M5 will be specified as precisely as possible.

Still, this document should be considered a living document. After its submission, it will be updated continually according to the project's progress, as an internal report; this report will eventually reflect what functionality is required for and will be shown at M5 and M6, respectively.

As a general rule, the demonstration shown at M5 will focus on functionality. It will be possible to demonstrate that an active service can be deployed, containing active functions in the data and the control plane, using a platform developed in the project. The demonstration shown will be the first demonstration worldwide of a heterogeneous active network, comprising active network nodes of different architecture and with multiple execution environments. While at M5, the heterogeneity is limited, the full scale of active functions in the control and management plane, as well as operation on different platforms (network processor and hybrid active router) will be demonstrated for M6. Quantitative criteria, such as performance and scalability, will also be a topic for M6.

Description of the Deliverable

This deliverable starts out with a description of the FAIN test-bed (chapter 1), consisting of active network nodes and management stations distributed throughout Europe at various partner sites. The method of interconnection used is the Internet; a corresponding protocol for the exchange of active packets has been agreed on in the project.

Chapter 2 outlines the scenario that will be used for demonstrating the FAIN architecture. The scenario consists of a series of generic network management steps that are to be executed at various stages in the deployment of an active service, and a specific example of an active service which serves as a tangible means to demonstrate the functionality offered by the FAIN architecture and its implementation.

Chapter 3 discusses the criteria to be applied in the evaluation of the FAIN results. We differentiate between qualitative criteria, which are needed for assessing functionality, and quantitative criteria, first of all relating to performance figures.

Chapter 4 finally completes the main part of this document, drawing some notable conclusions.

Appendix 1 outlines the SIP protocol and is provided for the convenience of our readers, since some basic knowledge of SIP is essential for understanding chapter 2.2.

An important addition, however, is Appendix 2. . It is the first version of a handbook for setting up the FAIN test-bed. This appendix provides guidelines for this task at the time of submission, and will be developed further extensively between December 2001 and the time of M5, April 2002, and also in preparation for M6.

Appendix 3 makes an attempt to take a somewhat more abstract view, trying to lay out the plan for re-assessing the initial requirements that were used to direct the work in FAIN from a business perspective. This section is by no means complete or fully thought-out; it is provided here merely as an indication to our readers that the project plans to consider how FAIN and its results may be positioned in today's commercial network provisioning environment. Eventually, a refined version of this section will be instrumental in the assessment of the overall achievements of the FAIN project.

The terms used throughout this deliverable have been defined in previous FAIN deliverables and reports, most notably D2 [1] and D3 [3].

Change History

1. Initial Draft October 11, 2001, version 0.1.
2. Draft with new structure and official FAIN template, version 0.1.1, by Bernhard Plattner. This version reflects input from the FAIN project meeting, Bled, October 24-26, 2001.
3. Draft 0.2, November 11, 2001. Integrates contributions as of November 8 and results of WP5 teleconferences of October 31 and November 8.
4. Draft 0.3, November 20, 2001. Integrates the following contributions
   - Reduced version of service deployment and management scenario (UPC)
   - Plan for evaluation of security properties (JSIS)
   - New version of Active Service scenario (FT)
   - Results of teleconference of November 15, 2001
5. Draft 0.31, 2001-11-28
   - Integrates new contribution of DT, DT-01-11-22_Draft_of_D6
   - Changes in chapter 1
   - Editorial changes in the whole document
6. Draft 0.4 of 2001-11-28
   - Integrates new contributions of JSIS, HEL, INT and FT
   - Exec summary and description of deliverable completed.
   - Introduction 1.1 completed
7. Draft 0.41
   - Integrates results of teleconference of November 29, 2001
   - New organization of chapter 3, HEL contribution WP5-HEL-035-D6-Int.doc
   - Details of setting up test-bed moved to Appendix 5
8. Draft 0.5
   - Integrated changes to chap 2 by Alvin Tan
   - Integrated changes to chap 3 by Lawrence based on Alex' comments
   - Integrated new version of Appendix 5
9. Draft 0.60
   - Assigned FAIN document id WP5-ETH-025-D6-Int, file name is WP5-ETH-025-D6-Int.vnnn.doc, where nnn is the version number (currently 0.60)
   - Incorporated contributions and comments delivered up to 2001-12-12
   - Incorporated discussions of teleconference on 2001-12-07
10. Draft 0.61
    - Includes Spyros' last minute revision of chapter 3
11. Draft release 1.0
    - Includes last revision of chapter 3 (HEL, FHG, JSIS)
    - Final editing by Bernhard Plattner, ETH
    - Additional editing by Alex Galis
12. Final release 1.1
    - Moved chapter 4 to appendix 3
    - Removed figure in appendix 3
    - Description of deliverable rewritten
    - Added Richard Gold as an author

Acronyms

AN	active network
ANN	active network node
ANSP	active network service provider
ASP	active service provisioning
C	customer (= client)
EE	execution environment
EMS	element management station
ISP	internet service provider
MS	management system
NMS	network management station
OS	operating system
PBNM	policy-based network management
PDP	policy decision point
PEP	policy enforcement point
PNO	public network operator
RCF	resource control framework
SIP	session initiation protocol
SLA	service level agreement
SP	service provider
VPN	virtual private network

Table of Contents

Table of Figures

Table of Tables

1 THE FAIN TEST-BED

1.1 Introduction

This section provides an overview of the topology of the test-bed, explains the type of nodes running at various sites and the hardware/software used to realize the nodes.

A detailed description of the test-bed and a handbook for setting it up is to be found in Appendix 5 of this document.

The FAIN test-bed will include different types of FAIN nodes:

- FAIN Active Network Nodes
- FAIN Element Management Station (EMS)
- FAIN Network Management Station (NMS)
- A FAIN Active Network Node runs active services and contains programmable management, data and control planes. We will develop three versions of this node until M6, and one version until M5. While active functions in the data and control plane will be demonstrated at M5, active management functionality remains for demonstration at M6.

All node types versions will exhibit the similar functionality vis-à-vis services and management components, i.e. they will all support the active service provisioning facilities (ASP) as developed in WP4. They will be different, however, in their respective Node OS architectures and performance characteristics.

We will develop a single version of each of the two types of management nodes listed above. All FAIN management stations can interact directly with FAIN Active Network Nodes.

In the first phase of the FAIN test-bed leading to M5 (April 02), we will install, configure and evaluate only Active Network Nodes of type A, and the corresponding Element Management Station and Network Management Station. An initial version of an active node of type C will be developed and demonstrated by HEL. Intel IXP-based Active Network Nodes (type B) and full-featured Hybrid Active Router Nodes (type C) will still be under development at that time and will be ready for demonstration and evaluation as part of the work for M6 (April 03).

Appendix 1 gives the functional components of the various FAIN nodes and management stations that we plan to realize for M5.

The FAIN test-bed, showing the partner sides, the deployment of the FAIN Active Network Nodes and FAIN Management nodes is detailed in the remaining part of this section.

1.2 Types of Active Network Nodes

1.2.1 Type A

Nodes of Type A are completely PC-based and provide active network functionality with the PromethOS node OS. PromethOS is the base active network implementation developed in the FAIN project, primarily (but not exclusively) for data plane active functions. PromethOS operates on a Linux platform; it comprises of a framework for managing active components installed in the Linux kernel, and the classification and demultiplexing facilities required to intercept active packets that need to be processed in the active network node, either in the data plane, control plane or management plane. PromethOS will also use a resource control framework (RCF)[1] to assure proper handling of resources local to the AN node.

[1] The RCF will only be ready for M6.

1.2.2 Type B

Nodes of Type B are functionally equivalent to type A nodes, however their hardware platform is the Intel IXP 1200 network processor. The IXP implementation will open the way to high-performance data plane active services, which will be implemented on the micro-engines offered by the IXP 1200 architecture. PromethOS will be used on the ARM control processor, which is part of the IXP. This of course implies that Linux needs to be ported to the IXP ARM environment[2].

1.2.3 Type C

Nodes of Type C (Hybrid Active Router) combine a commercial router with an active network EE provided by a physically separate PC (attached PC). In type C nodes, the commercial router does packet classification and demultiplexing, while active packet processing is done on the attached PC. For this purpose, a Linux/Java based NodeOS or PromethOS will be operated on the attached PC.

1.3 FAIN Network and Element Management Stations

FAIN is developing two types of management stations, the Element Management Station (EMS) and the Network Management Station (NMS).

Both management stations will have the same properties. The stations will be based in PCs with Linux OS. As programming platforms both stations need OpenORB and OpenCCM[3] CORBA platforms over which management components will be build. At the end of the project both management stations will be implemented over FAIN Active Node middleware (i.e. the RCF) in order to be able to limit the amount of management station resources different management instances are consuming.

FAIN currently allows only for one NMS per network. Therefore only one NMS will be operational during the demonstrations; however, partners may set-up their own NMS for testing purposes in their own realm.

One EMS may manage multiple active network nodes, which may be assigned dynamically. However, it is anticipated that each partner will run an EMS to be able to locally manage his active network node while testing.

1.4 Network Topology and Interconnection

The following partners will host active network nodes:

Table 1 - Partners hosting active network nodes in the FAIN test-bed

Partner	Location	Type of Node for M5	Host for management stations[4]	Type of nodes planned for M6	Access BW (Mbit/s)	Remarks
University College London (UCL)	London	A	EMS	A, B	open	
Josef Stefan Institute (JSIS)	Ljubljana	A		A	45	
National Technical University of	Athens	A				

[2] The port of Linux to the ARM processor need not be a FAIN achievement. Since such a port is not a research achievement, the project has decided to schedule an effort for porting Linux only if the software is not available from third parties (e.g., Intel).
[3] The use of OpenCCM platform is conditional of the availability of a new version of this platform during the project life offering the necessary functionality.
[4] Note that for a testbed-wide demonstration, only one NMS may be in operation at one time.

Athens (NTUA)						
Universitat Politecnica De Catalunya (UPC)	Barcelona	A	EMS, NMS	A	N/A	
France Télécom / R&D (FT)	Paris	A	EMS	A		This node may not be accessible from the outside, due to security restrictions imposed by the company
Hitachi Europe Ltd. (HEL)	Cambridge, UK	C (with minimal features)	EMS	C	155 or 64k	155 Mbit/s access possible if TEN-155 can be used
Siemens AG (SAG)	Munich	A		A	tbd	
Eidgenössische Technische Hochschule Zürich (ETH)	Zurich	A	EMS	B	100	
FOKUS Fraunhofer Institute for Open Communication Systems (FOKUS)	Berlin	A	EMS	C		
University of Pennsylvania (UPEN)	Philadelphia	A				

The FAIN AN sites will be interconnected using the Internet. The encapsulation protocol used in FAIN over the Internet is the Active Network Encapsulation Protocol (ANEP), as defined in [4]. In order to operate the FAIN test-bed in a effective way, the partners may have to mutually grant root access rights to each other, providing for ease of management and test-bed maintenance. Suitable tools, such as a VPN based on IPSec or SSH will be used to allow for a secure operation of this mode of access. The FAIN test-bed will therefore be configured as an overlay network on top of the Internet, i.e. any virtual connections between the various FAIN nodes will be possible. For the actual demonstrations and trials, a suitable virtual topology will be chosen.

We anticipate that the performance achievable over standard Internet connections will be sufficient for the purpose of M5; however, for M6, it will be investigated whether a high-performance solution such as a direct connection via GÉANT or a similar high-speed infrastructure can be negotiated.

Figure 1 shows the geographic locations of the FAIN nodes and a possible virtual topology.

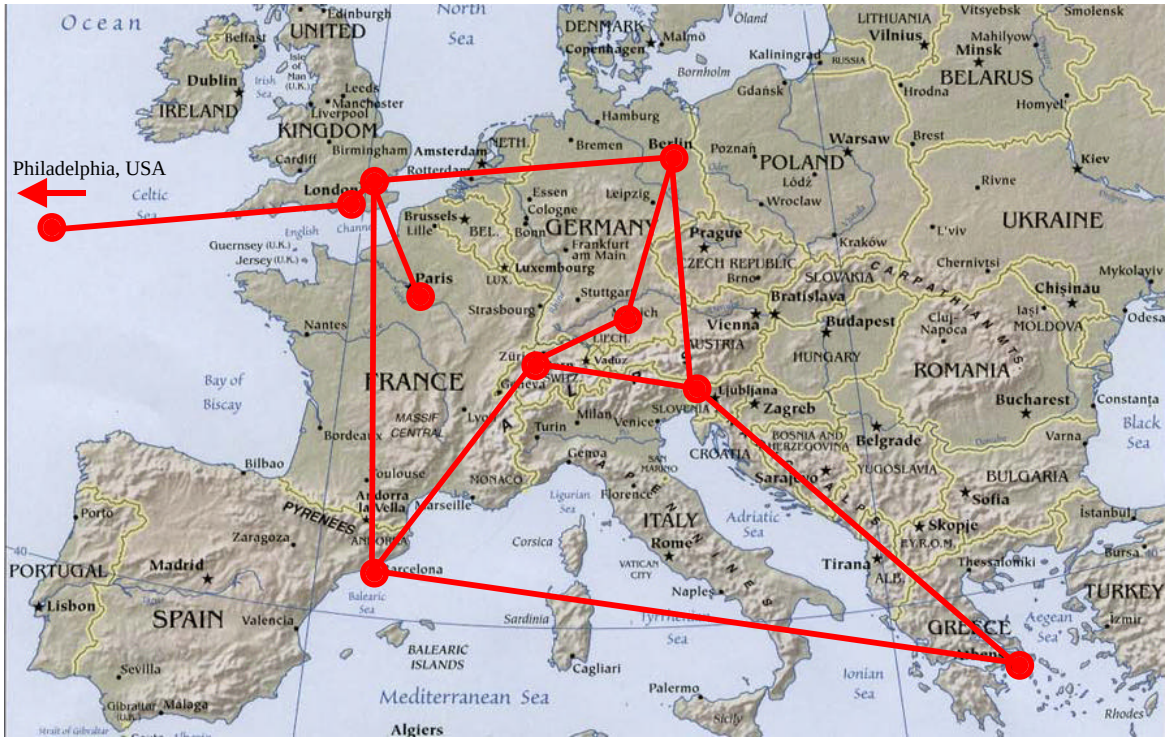


Figure 1 – Geographic topology of the FAIN test-bed

2 DEMONSTRATION SCENARIOS

The scenarios outlined in this section are based on the enterprise model developed by FAIN, which introduces various types of actors who take on specific roles. The model has been introduced in the deliverable D1 [1].
The realization of the full set of generic scenarios is planned for M6; for M5, a partial implementation is planned, which will be sufficient to demonstrate the specific active service scenario described in section 2.2.

2.1 Generic Scenario for Service Deployment and Management

2.1.1 Introduction

The scenario presented in this section shows some of the advantages of merging policy-based management and active networks technologies. The overall description of the scenario is related with a Service Provider (SP) trying to offer Active Virtual Private Network (AVPN) Services to his Consumers, which in their turn will use this AVPN to run an active service for their end-users. The SP obtains its Active Virtual Network (AVN) to build the AVPN service from an Active Network Service Provider (ANSP). That is, the SP negotiates a Service Level Agreement with an Active Network Service Provider obtaining the virtual active network and, also Consumers negotiate SLAs with the Service Provider in order to be able to obtain the AVPN service with certain QoS. Finally, in the scenario, a Consumer will use its AVPN to offer an Active Service to its end-users.

At this point, the FAIN demonstration scenario starts. It can be decoupled in five different sections

- ANSP-SP SLA Enforcement: This section shows how the ANSP configures its active network infrastructure in order to satisfy the SLA reached with the SP (e.g. creating the virtual active network).
- C-SP SLA Enforcement: In this part of the scenario, the SP configures its virtual active network infrastructure in order to satisfy the SLA reached with the Consumers. In order to do that, it makes use of the allocated resources and delegated functionality as result of the first section of the scenario.
- Consumer Reservation of Resources: The Consumer creates, configures and uses its AVPN service.
- Active Service Downloading: The Consumer installs in the active router the active service (i.e. the video transcoder) using its AVPN resources, which will be offered to its end-users.
- Consumer Bandwidth Reallocation: This section of the scenario shows how an automatic resource (i.e. bandwidth) adjusting mechanism of the AVPN service works in order to achieve a higher efficiency of resources and thus, saving costs.

The main concepts shown in this scenario are:

- Delegation of management functionality: We cover this functionality in the scenario in several ways. First with the SP's access to ANSP management functionality (within the node and the element manager). Also the SP may use its own code to manage allocated resources in order to offer a service. In order to allow the SP to do that the ANSP restricts the node interface offered to that code.
- Creation of an Active Virtual Network for an SP: In the scenario we show how an SP obtains several isolated computational and communication resources which conform a virtual environment in a group of interconnect active nodes across the active network, thus creating an active virtual network.

- Dynamic downloading of service-specific management components from the SP, in order to be able to specifically manage its offered service. The downloading of this component is requested to the ASP system by the management station at the network or element level. This component is then installed within the management stations and might interact with some of its components, e.g. the monitoring system in order to be able to make service decisions (e.g. the Consumer bandwidth reallocation scenario).
- Dynamic extensibility of the management stations functionality downloading new PDPs when they are needed[5]. As in the previous statement, the downloading of this code is requested to the ASP system and installed within the management stations (both network and element level).
- The dynamic installation of an Active Service within the active router, which is composed by two components: one running at the control plane and the other one in the data plane. The former controls the behaviour of the latter. The downloading of this component within the active router is mainly carried by the ASP components inside the active node.

2.1.2 Description of FAIN Actors

Consumer (C)

The Consumer is the user of the active services offered by an SP. In FAIN, a Consumer may be located at the edge of the information service infrastructure (i.e., be a classical end user) or it may be an Internet application, a connection management system, etc. In order to find the required service among those offered by the SP, the Consumer might contact a Broker to use its service discovery features.

Service Provider (SP)

An (SP) composes services that include active components delivered by a Service Component Provider, deploys these components in the network via the ANSP, and offers the resulting service to Consumers. The services provided by an SP may be value-added services or communication services. An SP may federate with other SPs in order to build more complex services. Descriptions of the offered services are published via the Broker service.

Active Network Service Provider (ANSP)

An ANSP provides facilities for the deployment and operation of the active components into the network. Such facilities come in the form of an active middleware, support of new technologies etc. Together with the NIP, they provide the communication infrastructure.

The ANSP owns an active network offering one or more environments where active code from SPs can run.

Network Infrastructure Provider (NIP)

An NIP provides managed network resources (bandwidth, memory and processing power) to ANSPs. It offers his network platform to the ANSPswho, in turn, can build their own environments. These environments are connected with basic IP connectivity which may be based upon traditional transmission technology as well as emerging ones (both wired and wireless).

In our scenario, for the sake of simplicity, the NIP assigns its entire network infrastructure to only one ANSP who will build upon it one or more Execution Environments.

[5] It is to be noted that PDP functions are service specific, i.e. these are active functions on the management plane. For M5, PDP code required for the active service will be pre-configured, while for M6 demonstrations, PDP functionality will be dynamic.

2.1.3 ANSP-SP SLA Enforcement

As stated previously, the ANSP offers part of his active network resources in the form of virtual environments (VE) and its Execution Environments (EE), which he has built over the programmable network. The latter was bought, as a whole, from the Network Infrastructure Provider.

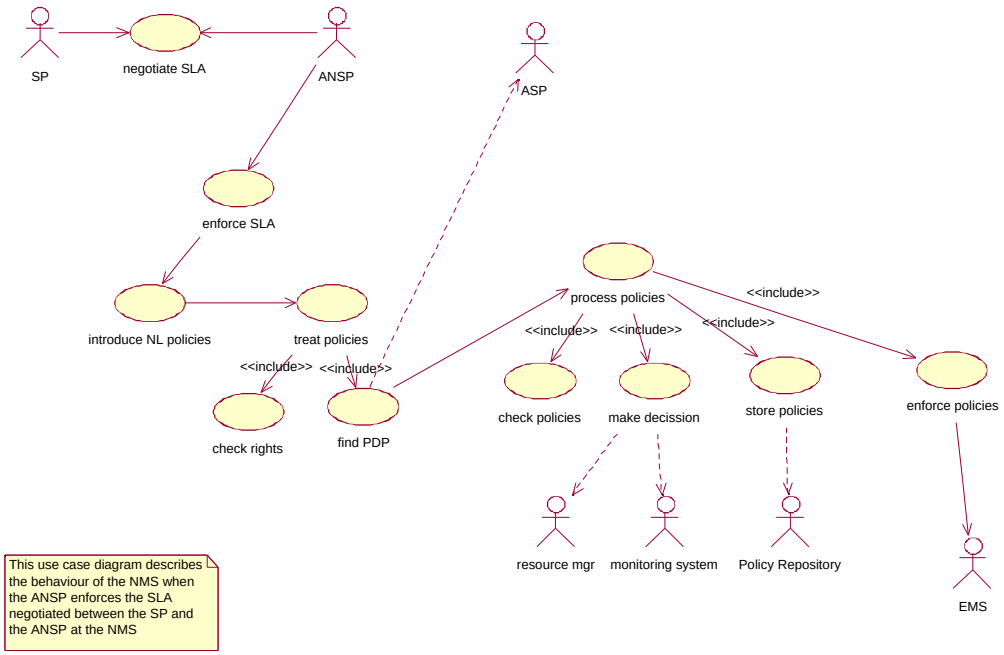
The SP obtains through the SLA negotiated with the ANSP: a) communication and computational resources, which conform to a virtual active network, b) access to Virtual environments and to Execution Environments, where active services from the SP could run using the allocated resources, c) delegated management functionality, in order to be able to manage its own resources, and d) the possibility of using part of the management system infrastructure of the NIP, now owned by the ANSP. As have been seen in detail in the previous section the management system owned by the ANSP is a two-tier policy-based management system, although in this paper we focus on the element level: the Policy-based Active Network Element Management (PBANEM) system.

Once agreed, the ANSP has to configure his infrastructure in order to enforce the SLA reached. Therefore, the ANSP uses his management system (the FAIN management system) so as to realise all the correspondent actions. This implies, basically, the introduction and enforcement of policies from two domains, QoS policies and delegation policies.

The enforcement of a set of QoS policies on the appropriate nodes will make the reservation of resources assigned to the Service Provider.

The enforcement of delegation policies will grant that the Service Provider will be able to manage his allocated resources and only these, thus isolating access to management functionality from different Service Providers.

Delegation policies will allow the Service Provider to use part of the management system infrastructure owned by the ANSP, which means that the Service Provider will be able to install and use its own service-specific management components within the Policy-based Active Network Element Management (PBANEM) system. Also, delegation policies will delimit the management functionality offered by the Active Network Node to the Service-specific PDP/PEP from the Service Provider, in order to guarantee isolation of the management functionality offered to that Service Provider. Therefore, the SP will not be able to manage other SP's functionality, and that other SP's will not be able to manage the functionality owned by this particular SP.



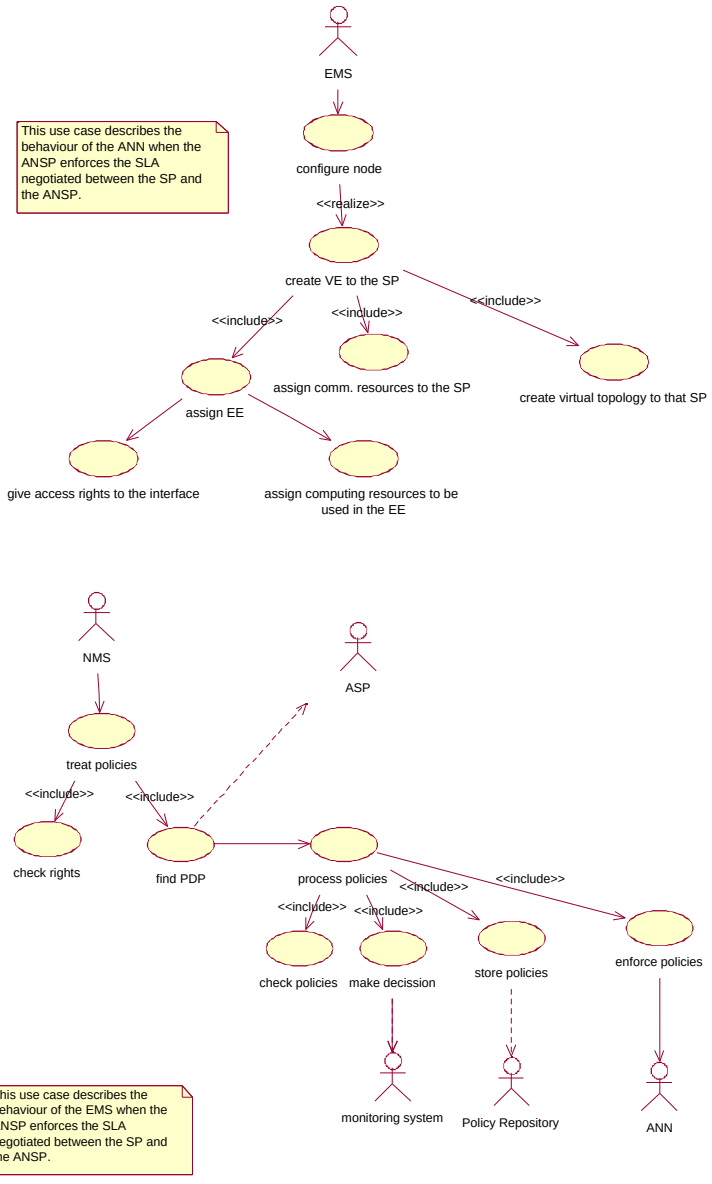


Figure 2 – ANSP-SLA Enforcement Use Cases

2.1.4 SP-C SLA Enforcement

Now, the Service Provider is able to offer Active Virtual Private Network services to his Consumers. When a Consumer is interested in this service, he needs first to negotiate, again, a Service Level Agreement with the Service Provider. The Consumer will buy the AVPN service from the Service Provider, which means that he will be able to reserve certain communication and computational resources, securely, between a number of sites, and that it will be able to use them to install an active service. As a result of the SLA achieved between the SP and the Consumer, the SP will create a number of policies that will allow that Consumer to request and use those resources. These policies will be enforced in the service-specific components when they are first installed in the ANSP's management system. Also, the SP will, if necessary, send to the ANSP's management system the appropriate policies in order to modify its virtual topology adding a new entry for the Consumers flows.

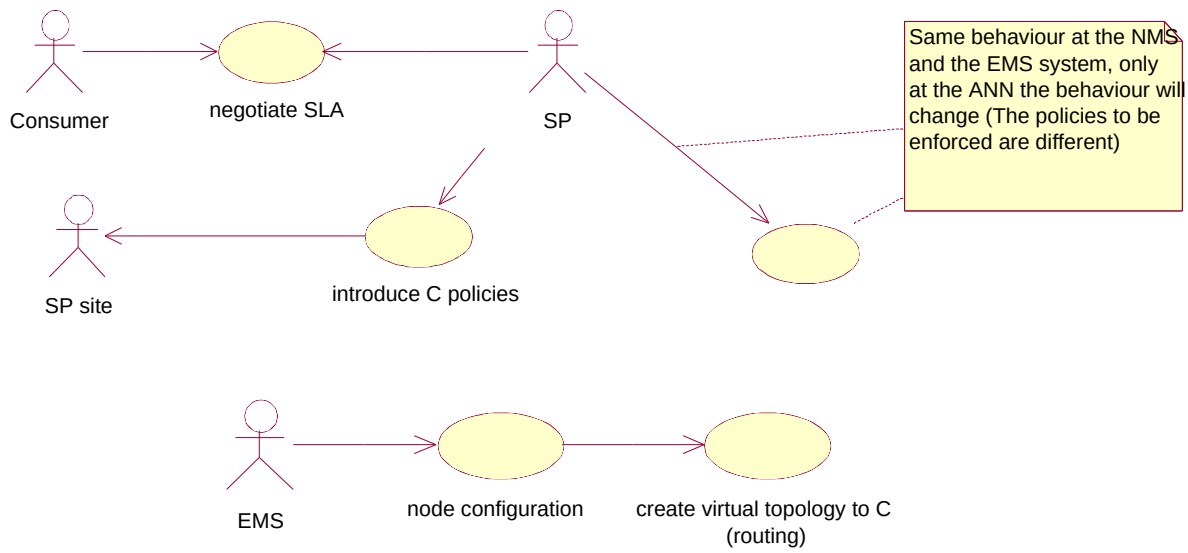


Figure 3 – SP-C SLA Enforcement Use Case

2.1.5 Consumer Reservation of Resources

The Consumer is now ready to request and use his AVPN service between the sites specified in the SLA and with the resources specified in it. The Consumer, in order to create a tunnel with certain QoS between two of these sites, sends an active packet that, basically, points to the code that should process it (in this case the service-specific components running in the ANSP management system) and carries a set of policies with which the Consumer requests the resources and conditions to use these resources (i.e. if the bandwidth used is less than the 50% of the allocated one, the reservation is reduced to the 75% of the initial amount, and a similar condition to increase the bandwidth if the resources used are over 90%).

The Active Network Node will check the pointer of the active packet and forward it to the ANSP management system (the PBANEM station). The PBANEM system will get the policies from the packet and look for the component responsible of managing those policies, install it if necessary, and forward to it the Consumer request.

Once installed, the service-specific component will decide upon the Consumers request based on status of SPs assigned resources and the Consumer's rights. Finally, if the service-specific components allow the Consumer to reserve the requested resources, it will enforce the Consumer policies in the node API offered to the SP, thus allocating part of the total SP's node resources to the Consumer. This process is repeated at each Active Network Node along the path between the two sites.

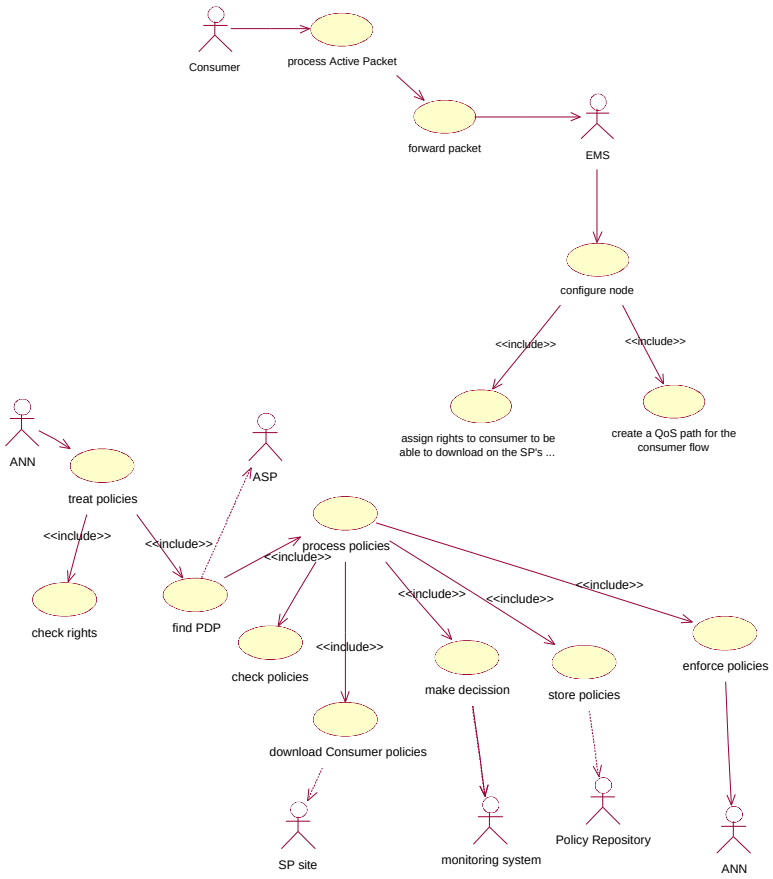


Figure 4 – Consumer Reservation of Resources Use Case

2.1.6 Active Service Downloading

Now, the Consumer is able to use its AVPN service. However, up to now, the Consumer will use communication resources only and he might have requested as well, some computational resources to be used by an active service that processes in some way packets flowing along his tunnel. The Consumer will, then, send an active packet that points to the requested service and that might include as well, the computational resources desired for that active service.

The ANN will detect that the packet has to be processed by an active service (e.g. the webTV transcoder) that is not currently installed on the node, so it forwards a download request to the Active Service Provisioning component inside the ANN. This will, after checking if the Consumer is allowed to use that service with the requested resources, download and install the Active Service.

Once installed, the Consumer's packets are forwarded to the Active Service. The functionality of the Active Service is not relevant to the scope of this scenario. It might be a videoconference service, a special encryption service, or whatever other. However, in order to demonstrate activity and composability both at the data plane and control plane, the active service will be composed by two modules, a data plane module, which processes data packets, and a control plane module, which monitors and controls the performance of the data plane module.

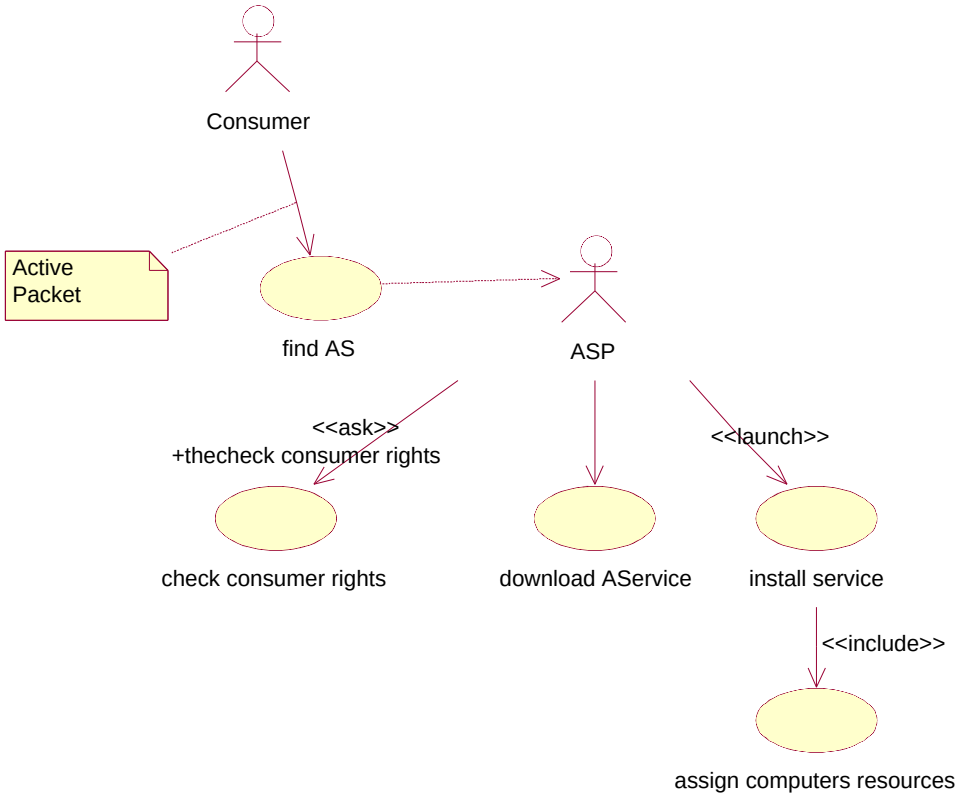


Figure 5 – Active Service Downloading Use Case

2.1.7 Consumer Bandwidth Reallocation

Finally, while the Consumer is using his resources, the monitoring system periodically reports the bandwidth used by this Consumer. Based on these reports the service specific components in the management station might decide, as an automatic resource adjusting mechanism, modify the bandwidth allocated to the Consumer. This automatic resource adjusting mechanism is realised according to the Service Level Agreement negotiated between the Service Provider and the Consumer, which has been enforced in the form of policies. The mechanism allows the Service Provider a more efficient use of its resources, and the Consumer to save costs.

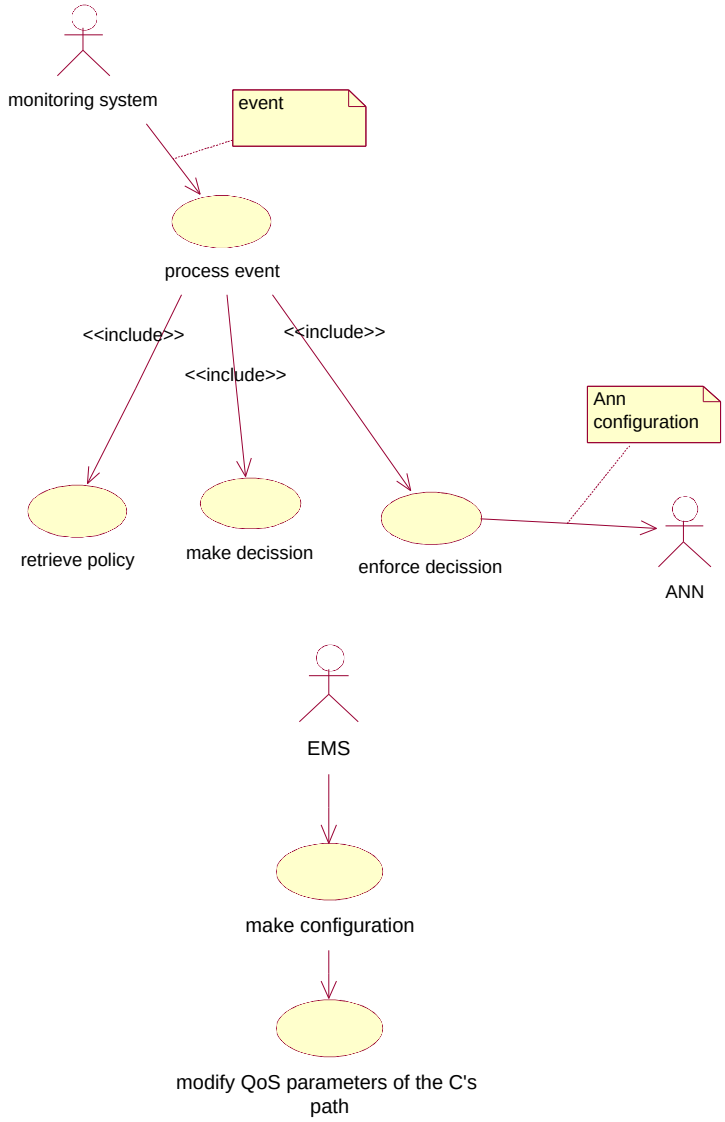


Figure 6 – Consumer Bandwidth Reallocation Use Case

2.2 Active Service Scenario

2.2.1 Mapping of the Active Service Scenario to the Generic Service

In this paragraph the different possibilities for mapping the active service scenario with the generic one are described.

First of all, the WebTV SP does not necessarily need an AVPN in order to offer the web television service. It just needs an Active Virtual Network (AVN) maybe as simple as some computational resources in one active router to install the transcoder, and the flows go through it using best effort. In this case, the WebTV SP is in a similar level of the FAIN actors' hierarchy, i.e., just as how the AVPN SP is provisioned with the underlying node resources, the WebTV SP gets the same.

Obviously if the WebTV SP would like to offer some guaranteed QoS in its connections, it can buy the AVPN service from the AVPN SP to enhance the web television service. In this case, the WebTV SP becomes the AVPN SP's Consumer, hence moves a level down the FAIN actors' hierarchy.

To take this mapping a step further, we can contrast the above two options with today's mode of delivering video over best effort IP (e.g., movie trailer clips seen over the Internet). In this case, the WebTV SP does not use active networking technologies, hence may only broadcast it's video on PCs only. We call this the simple, restricted web TV service.

To summarise, In the first and second examples (which brings forth the advantages of active networking by enabling transcoding) the web television service can even be delivered to mobile telephones, thanks to the transcoder.

Alternatively, in order to simplify the mapping of the WebTV Active Service scenario to the Generic Scenario for Milestone M5, although we intend to provide the WebTV service with guaranteed QoS, we will consider that the ANSP may also play the role of an AVPN SP. Thus, the WebTV SP obtains its resources directly from the ANSP. Since the AVPN SP is the ANSP, there is no Active Virtual Network (AVN) instantiation because the AVPN SP (the ANSP), obviously, already has all the active network resources. In this way, we simplify the scenario by removing one step.

2.2.2 Description of the WebTV Active Service

This section provides a complete scenario that exemplifies the way(s) the FAIN network may be used and how services may benefit from an AN infrastructure that claims to be flexible, secure and with high degrees of performance. It also binds together all the use cases of the previous section that support the FAIN business model as well as it describes the functionality of a customisable management plane based on policies by means of a realistic service. To this end, the service itself is not the essence of the demonstration but the tangible and observable means to support the FAIN claims. This service will be the subject of M5 demonstration.

In this context we will be able to demonstrate the creation of the SP's virtual infrastructure (virtual network), the instantiation and use of the SP's management architecture for the purposes of service management, and finally the customisation of a specific service according to the requirements of the SP's Consumer (end-user).

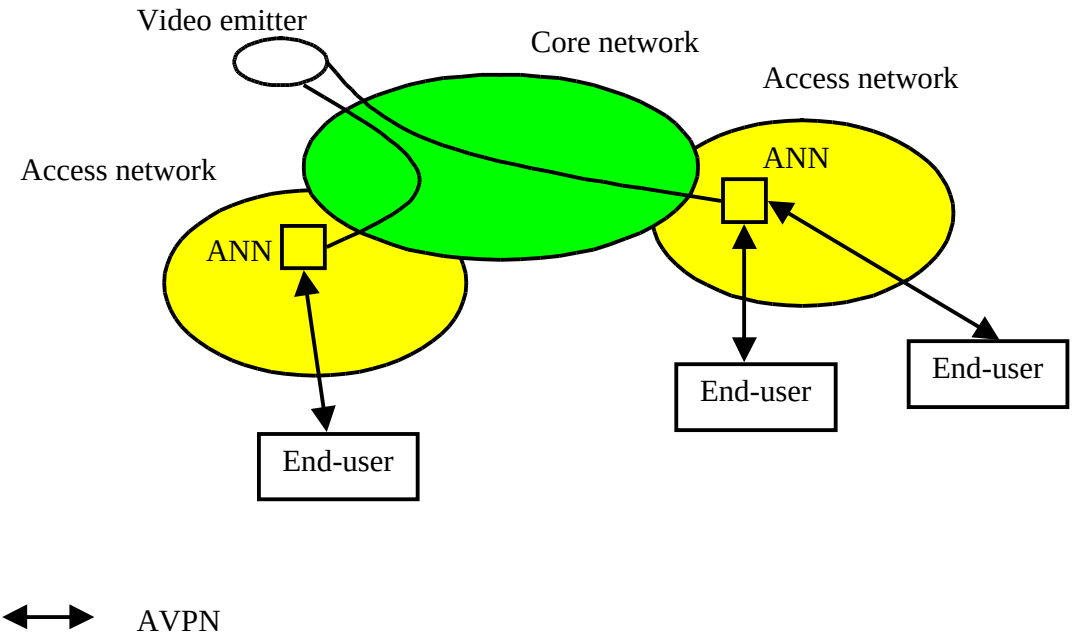


Figure 7 – Environment of the Web-TV active service

The service that has been selected is a WebTV service. Although different variations of the same service regarding the roles assumed by different actors may apply, in our demo we assume that the WebTV service is offered by the Service Provider who is the owner of the WebTV source (video

emitter). This service is used by a number of Consumers (end-users) that have completed an SLA with the SP. Furthermore, the SP's infrastructure is in fact a virtual one, which is the result of another negotiation and enforcement of an SLA between himself and the ANSP. Figure 7 (above) depicts the environment within which the WebTV service will be used and run.

More specifically, an SP decides to deploy a WebTV service to offer to his Consumers. Such a service may have specific resource and service management requirements. To this end, the SP contacts an ANSP from whom purchases an Active Virtual Private Network (AVPN) with certain amount of resources (computational and communication), access rights and the ability to manage the service to be offered through a management architecture of which he is the only owner. This is achieved and formalized in the form of an SLA between the SP and the ANSP (see section 2.1.3 ANSP-SP SLA enforcement). For the purposes of our example part of the agreed SLA is the permission to download service components to the Active Nodes.

After the creation of the AVPN the SP is ready to offer the WebTV service to end-users (his Consumers). To this end, it advertises the WebTV service to a portal (see figure 8 below) by means of the SIP protocol. The portal also knows the video format used by the WebTV emitter (step 1). The portal in return gives a multicast address for the emitter to emit to. (step 2).

At this point the WebTV may start transmission using the multicast address (step 3). No one has registered to this service so far, thus none is receiving such a service. Finally, an end-user accesses the portal and registers with the WebTV service offered by the SP (step 4) and an SIP is opened between the Consumer and the portal (step 5). The registration is captured by an SLA between the Consumer and the SP (see section 2.1.4).

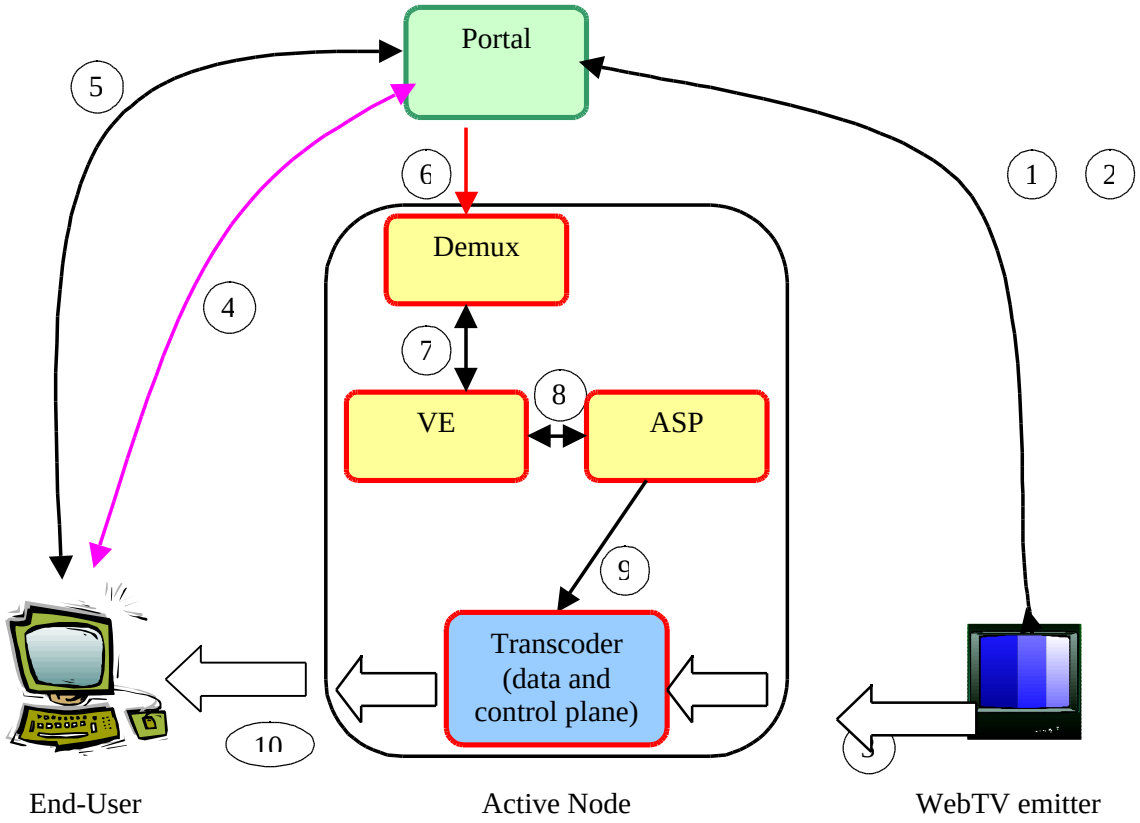
Part of this SLA is information about the Consumer's QoS requirements and capabilities. The Consumer's QoS requirements regarding this service are resolved by means of the resource reservation use case (see section 2.1.5).

Among the Consumer's capabilities is the media format that he can support. Unfortunately, this is different from the media format used by the WebTV service therefore customisation of the service to adapt to the Consumer's needs is necessary. The incompatibility between the different media formats is detected in the portal and an active packet is sent to a conveniently located AN Node (step 6). This active packet contains a reference to a code that implements the right transcoder from the Consumer's format to the WebTV's format and vice versa. The active packet contains additional configuration data that consist of the multicast address and the IP address and port number of the end-user. The latter is to be used for control plane activity.

Such active packet is received by the ANN and is forwarded by the Demux component (step 7) to the proper virtual environment and EE for further processing. The entity within the EE that processes the aforementioned information of the active packet may be considered as part of the management plane. If the transcoder is not already there then the management entity contacts the ASP (step 8), which, in turn, locates, downloads and installs the proper transcoder in the ANN (step 9) (see also section 2.1.6) as part of the datapath of subsequent flows destined for the Consumer. Part of the installation includes the initial configuration of the component and the exporting of the control interface thereof to be used by control plane entities during reconfiguration, e.g. new address by the Consumer.

At this point the ANN is ready to listen to the video flow from the emitter on behalf of the Consumer and based on his QoS profile to process and forward traffic to this Consumer in the right video format (step 10).

During the period that a Consumer uses the WebTV service, the Consumer is able to renegotiate his quality of service (see section 2.1.7).



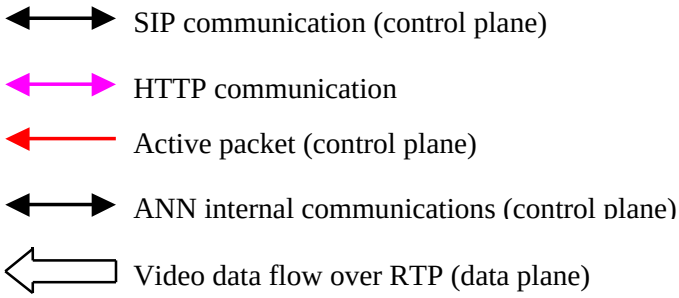


Figure 8 – Illustration of the step-by-step explanation of the active service scenario

2.2.3 Data and control plane functions of the transcoder

For clarification and emphasis, this sub-section differentiates between the functions relevant for the data and control plane, respectively.

The WebTV active service demonstrates the use of active functions in both the data and control plane.

Specifically, the data plane functions of the transcoder are:

- To listen to the multicast address used by the emitter
- To read the video flow in format A
- To transcode the video flow delivered in format A into format B
- To transmit the video flow in format B to the end-user, using the configured end-user address.

The control plane functions are:

- To configure the multicast address the transcoder listens to
- To configure the end-user address

It is to be noted that the type of video formats the transcoder is using is fixed, once the transcoder is deployed. It is the job of the ASP to select and install the right code of the transcoder, i.e. the one that handles the needed video formats both in input and output. This is a distinctive feature of active network technology, which allows just-in-time deployment of functional components (i.e. just the code needed in the specific use case).

3 THE FAIN EVALUATION FRAMEWORK

3.1 Foundations of the framework

The purpose of section 3 is to provide an evaluation framework that allows not only to evaluate the FAIN architecture –both at the network and node level- as a standalone system but also to provide the means to compare FAIN with other active or programmable network architectures developed by the research community. We note here that to the best of our knowledge this is the first attempt to come up with such a complete scheme that attempts to create a common evaluation platform not only for evaluation but also for meaningful comparisons. As this is a very ambitious objective to be achieved by this deliverable in one attempt, we have approached this deliverable, and in particular this section, from the angle of a working document that will be periodically updated. Accordingly, in this section we aspire to lay down the foundations of the FAIN evaluation framework and then refining and adding details to it as we accumulate more experience and insights while working with the actual FAIN test-bed.

In this context, we will create a taxonomy in the form of interrelated tables that will constitute the basis for the evaluation framework. This taxonomy is comprised of two major interrelated parts: a) the AN technical specification in the form technological ingredients and methodologies used to build an active network [5], [6] (section 3.2), and b) a number of major properties that an AN manifests as a direct result of its technical specification (section 3.3). Each one of these properties represents different aspects of the evaluation process of a network

These two parts are interrelated in the sense that as different technical specifications may result in same properties, which, in turn, may be measured, a quantitative evaluation of a property provides strong evidence about the capabilities and usefulness of a technology.. Accordingly, the result of the evaluation process may be summarised in the form of both qualitative and quantitative data that allows the observer to safely draw conclusions about the behaviour and the capabilities or shortcomings of the system. In addition, one of the major outcomes of the evaluation process would be to provide the means to unambiguously define and assess what we call as the “level of programmability” supported by an active/programmable network architecture [1].

In D2 we have defined the “level of programmability” as the trade-off between the following properties to be possessed by an active network:

- Flexibility
- Security
- Interoperability
- Performance

In our approach, the functionality of an AN is part of the technical specification. We evaluate the functionality with regard to a particular system. The assessment of functionality is in the properties. It should be noted that in our approach the degree of manageability and controllability of an AN are determined through an overall assessment of the above listed properties. For instance, the degree of manageability of our AN can be determined through evaluating the flexibility, security and interoperability (and the corresponding performance) properties of the technology(s) used for the installation / de-installation of services on our AN. Another example to determine the degree of manageability would be to evaluate the flexibility and interoperability properties through determining the level of support for control / management of nodes by several independent management stations of different types.

3.2 AN Technical Specification

In our approach, the functionality embedded in an AN and the technology used to build such functionality is considered as part of the technical specification. In this section our aim is to classify functionality and technology in a number of well established and understood conceptual categories

that facilitate comparison among different systems. The evaluation of the functionality and the technology will be part of the process described in the next session when we evaluate system properties and establish the link between technical specification and the entailed properties.

In so far, we have identified the following categories of our AN technical specification :

- Networking technology

  Different networking technologies have an impact on how programmability or activeness may be applied [6]. ATM favours more programmability for engineering a QoS network whereas IP networks (Internet) favours more packet processing within the data path. In contrast, mobile networks impose their own requirements on an active network as a result of bandwidth limitations.

  Possible values for the specification of the networking technology are: IP, ATM, Optical, Mobile

- Network Element technologies

  Similarly, emerging innovative network element technologies may have an impact on the achievable level of programmability. For instance, the advent of network processors has increased the processing capability of the network elements thus allowing for more intelligence to be built.

  Possible values for the network element technologies are: software routers, high performance routers, hybrid routers, switches, network processors, etc.

- Network Element OS

  Possible values for this specification: Linux, Extended Linux, research prototype etc…

- Middleware technology

  Possible values are: Java with RMI, Corba, and Mobile Agents

- Service6 composition frameworks

  Although service composition frameworks may be seen as part of the middleware technology, due to their importance in active and programmable network we treat them here separately.

  A number of composition frameworks have been suggested in order to create and deploy services. Obviously, the choice of the composition framework will greatly influence the flexibility offered by the network. Different composition frameworks may be used in different planes depending on the problem space to be addressed.

  For example we may opt for Enterprise Java Beans or Corba Component Model for composing control services whereas for transport plane composition we may use approaches like P1520, which are less complex than the previous ones.

  Possible values for the specification of the service composition framework: EJB, CCM, P1520, composition languages like netscript etc.

In addition to these categories more categories may be introduced in the future versions of this section. Such categories may either appear as specialised sub-categories of the existing ones or brand new ones that make the technical specification of an AN complete.

3.3 Properties

As discussed previously, one major goal of this evaluation is the identification and analysis of the level of programmability. Based on the FAIN evaluation framework we should be able to define and measure the level of programmability of an active architecture. Two types of evaluations are necessary

[6] Here the term “service” is used in a more general context that encapsulates offered functionality that resides in any of the three planes, namely, transport, control and management

to be carried out: firstly, we validate to what extend such property is indeed owned by the architecture; and secondly, what is the cost of possessing such property. In this section, we provide an initial suggestion of how this may be realised.

The verification about the ownership of property may be done by an exhaustive list of test cases that the system undergoes successfully. Similarly, evaluation of the cost of the property may be done by an exhaustive list of measurements. However, the creation of such an exhaustive list is beyond the scope of this project, as such an endeavour may constitute a project of its own. In this project we will try to establish a minimum set of test and measurement cases that will provide a minimum qualitative and quantitative guarantee.

Our approach is to connect manifestations of these properties in the form of features7 to specific test cases and performance metric(s). To this end, features may be verified by their associated test cases that will be built in testing the property of the system. In addition, they can be measured by connecting them with a specific cost incurred as a result of adding such feature to the functionality of the system. For instance, a feature of the security property may be the denial of service. The association of this feature with a performance metric could be the calculation of the overhead in terms of, for example, packet throughput, as a result of adding this feature to an AN architecture. This approach will also facilitate comparisons among different secure AN architectures both claiming the same feature. Finally, features of a property may further be distinguished into mandatory and optional ones. The mandatory features define the minimum level required by an AN to claim ownership of a particular property.

Another aspect of the evaluation framework results from the existence of distinct operational planes, namely, transport, control and management. Manifestations of these properties and features thereof may occur in either of these planes. As a result, focusing on an operational plane and looking at corresponding features you may draw conclusions that are specific to that plane for example manageability and so on and forth. Furthermore, in the subsequent sections we focus on specific aspects of some operational planes and features of the properties where we provide additional information about the evaluation framework. The choices are dictated by the current needs emerged in FAIN and will be enhanced with additional ones as we carry out FAIN evaluation during the next and final stage of FAIN.

Finally, the fact that a property that may be manifested by a node is not necessarily a property that the network possesses and vice versa. Accordingly, it is essential to define how a property claimed by a network is manifested. Is it the result of a property possessed by all nodes, some of the nodes or just one specialised node?

The following table 2 formalises the previous discussion and provides a template of tables to be used in the subsequent sections when the evaluation process is detailed.

Table 2 - Table Template for Property Types

Property Type							
Features	*Transport*	*Control*	*Management*	*Network*	*Node*	*Technology*	*Comments*
Mandatory							
Optional							

[7] Our aim here is to use a list of features such as they collectively define a property

3.3.1 Flexibility

Table 3 - Table for Flexibility Property Type

Property: Flexibility							
Features	*Transport*	*Control*	*Management*	*Network*	*Node*	*Technology*	*Comments*
Mandatory							
Composability	X[8]		X	X	X		
Extensibility							
Scalability							
Virtualisation	X	X	X	X	X		
Optional							
Composability		X			X		
Code Mobility							

Flexibility is a quite generic property. By this we mean the ability of a system to change dynamically its behaviour, adapt to new requirements, cope with increases in information volumes and functionality, and reuse or synthesise its existing services.

In particular, we consider as part of the flexibility property the following features:

- Composability

  Composability allows the system to reuse and recombine its functional components into forming new services and functionality.

- Extensibility

  Extensibility allows the system to evolve as new requirements and services are needed while these can be introduced and incorporated in the existing system in a seamless way.

- Virtualisation

  Virtualisation allows for the partitioning of network resources among different user communities. This results in supporting more liberal business models and customised usage of resources.

- Scalability

  Scalability refers to the network architecture design and the distribution of the network functionality in such a way that the network can account for increasing volumes of user requests.

[8] The entries at this and subsequent tables are provided here as an example and they do not represent the result of an evaluation. Tables of this form are expected to be the outcome of the evaluation process.

3.3.2 Security

Table 4 - Table for Security Property Type

Property: Security							
Features	*Transport*	*Control*	*Management*	*Network*	*Node*	*Technology*	*Comment*
Mandatory							
Authentication							
Authorisation							
Enforcement							
Integrity							
Logging							
Optional							
Verification							
Availability							

Evaluation of FAIN security architecture should cover issues like

- what security features are provided in FAIN ANN?
- how and where are they provided?
- What “level” of security do these features provide?
- how does FAIN security architecture compare against other existing approaches?
- how does FAIN security architecture perform (in an experimental test-bed environment)?

Evaluation shall be qualitative and quantitative. Comparing different security architectures directly, i.e. in a quantitative manner, is difficult to say the least. The same applies even for quantitative evaluation of a single security architecture, since it is hard to define sensible criteria for the "measurement" of security in a system. Thus, FAIN security architecture will be mostly evaluated in a qualitative way, although we will strive to give some measurement results mainly on performance overhead aspects.

Qualitative evaluation covers the first four questions posed. It is based on the analysis of FAIN security architecture with regards to a set of security features/requirements restricted to high-priority security requirements as defined in D2, [1,page 70]. This should give an overall sense of what "level" of security is provided within FAIN. In addition it includes evaluation of other AN security approaches and comparing them against FAIN, which should give a reasonable feeling of whether security level provided by FAIN is higher or lower than that of other approaches. The results of qualitative evaluation shall be depicted in the form of a table 3-3.:

This table first breaks down the somewhat abstract security property into its basic components or features, which can be more easily evaluated in a qualitative manner. We consider the following features o security:

- Authentication

  Authentication allows the system to securely verify the identity of a principal.

- Authorisation

  Authorisation decides whether requested action by a principal shall be allowed or denied.

- Enforcement

Enforcement acts upon the authorisation decision, i.e. it either allows or denies the execution of principal's request.

- Integrity

  Integrity enables the system to detect any modifications of the information in transit over the network by unauthorised adversaries.

- Logging

  Logging allows the system to keep a trail record of security relevant (and other) events within the system.

- Verification

  Verification allows the system to dynamically assess the safety of the active code before executing it.

- Availability

  Availability refers to the ability of the system to retain operation despite malicious or unintentional practices leading to exhaustion of system resources, i.e. to denial of service.

The table then gives the reader a comprehensive view of the level of security provided by FAIN (or other A N) based on four sets of information for every security feature:

1. presence of the feature in the transport, control, and management planes
2. operation of the feature, i.e. whether it operates locally on the active node or it demonstrates network wide behaviour
3. which technology is the feature based on
4. nature of the feature, describing the level of security this feature provides by specifying what it protects against and how does it implement the protections

As mentioned, it is hard if not impossible to define security evaluation criteria, which can be measured in an experimental set-up. However, even though we can not measure the level of security, it may be interesting to measure the sheer performance aspects of security, i.e. he performance overhead incurred when security mechanisms are activated.

For the purpose of comparing security property of FAIN active node against other active nodes, a maximum of two other kinds of active nodes shall be evaluated. This comparative evaluation will be based on the same basic set of features as depicted in table 3-3. However, the comparative evaluation should capture other security features, which may be demonstrated by non-FAIN nodes.9 Quantitative measurements on non FAIN platforms should be avoided, only existing results, if any, should be used to compare FAIN platform to other platforms.

[9] Note that initial FAIN security architecture has a limited scope, and thus lacks some features, which are planned to be added later.

3.3.3 Interoperability

Table 5 - Table for Interoperability Property Type

Property: Interoperability							
Features	*Transport*	*Control*	*Management*	*Network*	*Node*	*Technology*	*Comments*
Mandatory							
Portability				X	X	Java Mobile agents, bytecodes	
Openness			X	X	X	Common Policies	
Openness		X			X	L-interface	
Optional							

The openness and portability features collectively describes the interoperability property of our AN. For instance, the openness feature can be determined via the evaluation on the number of types of foreign EEs that can be hosted by the FAIN network; or via the level of support for accessing, distributed control, co-ordination and management of network resources from external network service components. The portability feature can be determined via the types of implementation language supported for the EE.

3.3.4 Performance

Table 6 - Table for Performance Property Type

Property: Performance						
Metrics	*Transport*	*Control*	*Management*	*Network*	*Node*	*Technology*
Throughput						
Response Time						
Delay						
Jitter						
Packet Loss						
CPU cycles						
Number of VEs						
Number of EEs						

Unlike the other properties listed in the previous sections, the performance property has been introduced to enable quantitative-measurements. Each feature will be linked with a metric(s). Eventually each property table will be associated with a performance table that provides a quantitative assessment of the network thereby facilitating objective comparisons with other networks engineered in alternative ways.

For instance, the portability feature of the interoperability property can be determined via the number of the different types of implementation languages (for the execution environment) supported in our AN. A set of statistical data can then be gathered to evaluate the respective performance penalty, as a result of allowing such greater portability. Similarly, a NodeOS that supports virtualisation may be evaluated to find out how many VEs may be instantiated before a degradation in performance is observed.

Note that the list of metrics depicted by the table may, in no way, be considered as exhaustive and it will be extended as necessary.

3.4 User's Guide to the FAIN Evaluation Framework

This section attempts to provide a preview by means of some indicative examples of how the evaluation framework may be used to carry out the evaluation of a network (including that of FAIN) and how each entry in the tables may be justified? In other words how can we unequivocally provide a proof that a network possess a certain property/feature?

As mentioned before the justification procedure to be followed in filling the tables is to identify specific functionality that is associated with some feature(s) of the property tables. This will be used as a means of carrying out the qualitative assessment of the network. For instance, some of the entries in the tables may take the form of "high", "medium", "low", values that are inferred by the functionality embedded in the network. Sections 3.4.1, and 3.4.2serve as these indicative examples for using the FAIN evaluation framework to establish the flexibility and security properties.

Another important use of the evaluation framework is that it enables you to focus and draw conclusions about certain aspects of the system. For instance, you can "zoom-in" and evaluate the node as a whole or the network as well as different operational planes by just looking at specific columns and rows of the property tables. One such complete example is provided in section 3.4.1 due to the importance of this plane.

3.4.1 Flexibility Property

In order to assess to which extent systems possess the features laid out in our evaluation tables, we would need to ask ourselves various questions. These questions would allow us to ascertain how far systems go in fulfilling these features. For instance, questions could concern the support for rapid definition and provision of services, which can be activated, programmed and controlled by the end-users and the Service Providers. Answers to these questions would provide justification that a network or node possesses the features of composability and extensibility. A selection of questions follow which address various aspects of flexibility, it should be kept in mind that these questions do not represent an exhaustive listing, but rather a demonstration of the style of questions that should be asked in order to drive our evaluation and draw our conclusions.

- Does the system provide support for service scalability? To what size of network (i.e., number of nodes) is the system expected to support?
- Can dynamic service provision be supported by the system? How many components can be realistically composed together to provide a dynamic service?

- Is it possible for flexible and light-weight service management facilities to be defined, activated and controlled by the end-users and/or the Service Providers?
- Does the system support multiple levels customisation and programmability of services? Can the service architecture provide a set of components, which can be specialised and enhanced? If this capability is present, then it will enable service providers to differentiate themselves through their own service development.
- Are Open Service Resources APIs provided? If this is the case then a new service can be provided to end-users that manage the network and service resources as programmable entities.
- Does the system allow for the user personalisation of subscribed services so that the service is independent of the serving network and the terminal in use?
- Can new services/facilities be provided and deployed? How straightforward is this provision and deployment?
- Are subscriptions to new services allowed and can the subscription to a particular service be suspended or cancelled? Under which conditions is this functionality possible?
- Does the system support the creation of virtual networks? Virtual nodes could be created on some subset of AN nodes in the network to form a virtual network. Each virtual network (the virtual nodes that make up the network) could then be controlled by an independent Management Station. Can these virtual networks be dynamically configured?
- Are open APIs for the creation of plug-ins and service components supported?
- Can rapid service deployment be performed by the system? This is a concern for both active node design and the tools used for the design of Service Components. “Plug-and-play”-like functionality (to create services from service components, and to install these services on the active nodes) is needed to provide dynamic, on-the-fly service creation and deployment.
- Does the system in question support a multiplicity of services and applications each controlled via its own management station?
- Can services, applications and transport/media interworking at the network edge be provided by such a system?

By answering questions such as these we can gain a vital understanding of the extent that the system possesses the features laid out in the evaluation tables. When the questions are answered they must be qualified and it is this qualification, which provides the grading of success.

3.4.2 Security Property

Even though qualitative evaluation does not involve measuring any metrics, a simple three level scale can be used for “measuring” the level of security achieved by FAIN (other AN model) regarding specific features. In an example three level scale, security features can be evaluated as “weak”, “standard”, or “strong.” Based on the analysis of security features provided by a given AN model, one can then estimate the level of security by stating that e.g. “FAIN provides standard authentication based on digital signatures.” The security level is estimated based on the following feature specific criteria.

5. Authentication: does it protect against passive attacks or active ones (man in the middle)? does it use simple schemes (cleartext, encrypted passwords), or complex ones (digital signatures, secure authentication protocols)?
6. Authorisation: what is the granularity level? what objects/parts of the system are protected? Is there a single or multiple authorisation engines?
7. Enforcement: what objects/parts of the system are protected? is there a cache of authorisation decisions?

8. Integrity: does it protect against packet modification, bogus packets, replay attacks, cut and paste attacks? does it protect dynamic part of the packet?
9. Logging: what events are logged? which parts of the system are subject to logging? are logs protected?
10. Verification: is it based on digital signature, source code inspection, stack inspection, proof carrying code, restricted (safe) language?
11. Availability: how do authorisation and enforcement provide availability? are there any other mechanisms against DoS attacks? what types of DoS do they protect against?

Security criteria as proposed will evaluate level of security achieved in fulfilling high-priority security requirements in all plains: data, control and management plain. Common security architecture for all three plains was also a FAIN security requirement, but not set as high-priority in D2 [1,page 61].

3.4.3 Management Plane

3.4.3.1 Flexibility

3.4.3.1.1 Extensibility

This is a especially interesting feature in the active networks field, which would describe the capacity of the system to dynamically accept new policy enforcement or policy decision point types that complete or enlarge the functionality of the management system itself. The fact that the structure, interfaces or functional capabilities of these components might not be known in advance imposes several requirements in terms of flexibility and policy manageability that should be assess.

The associated mechanisms allowing to dynamically discover new components, and the ability of the architecture to let the user assign his own resources to the management tasks they are more interested in, form a part of the adaptability "machinery" that it is being developed, and therefore demand to be paid attention in the evaluation phase.

Two different points of view should be adopted in order to characterize the extensibility of the management system:

- Functionality extensibility: i.e., the capacity of the system to acquire new modules without affecting other processes, and make the obtained abilities available to the rest of running services. In this case it would be necessary to check whether the dynamically deployed objects are capable of exploiting the existing infrastructure[10].
- Impact on the manageability: the accepted modules should be controlled by the existing infrastructure. The use of a policy-based approach should benefit the reduction in that impact, meaning in practice an increased control transparency. From this perspective, our aim would be verifying that newly introduced policy enforcement points are manageable by the current set of policies.

As a proposal to evaluate this feature, we suggest the use of a s*uccessful insertion degree* metric. This percentage would measure the success when attempting the inclusion of external components and policies[11] into the system. It may be given as the relationship between the successful and unsuccessful attempts. This metric would help to assess also the adaptability of the management system.

3.4.3.1.2 Scalability

Since the high complexity of active networks may result in an increase in the number of entities (including services) to be managed, scalability becomes an essential requirement, which should be subject of evaluation. In order to ascertain the evolution of the *complexity vs network size* relationship, which may provide an indirect way to obtain information about the scalability degree, the behaviour of

10 This infrastructure would include –but are not limited to- the existent PDPs and PEPs, the monitoring system and the database. Notice that retaliation would also help scalability.

11 Resulting either from other management domains necessities or from service requirements.

the system should be observed as the number of network nodes gets increased -until eventually it reaches the maximum number of nodes expected in the test-bed.

A metric that may offer a measure of such complexity is the number of PDPs, PEPs and policies required to manage the network. The higher the number of components and policies within the management system the more the interaction complexity becomes when trying to keep it under a close control. However, even the use of different network topologies may influence such complexity and therefore have a certain impact on the scalability evaluation. These kind of factors must be taken into account and appropriately corrected when correlating the obtained results.

Other metrics that might be useful in evaluating the management system scalability are:

Management Traffic Statistics.-the amount of management traffic that traverse the network. These statistics should be collected for different network sizes.

Resource Consumption.- this metric would provide another way of measuring scalability, by analysing the *management resource use vs network size* curve.

3.4.3.2 Interoperability

3.4.3.2.1 Openness

The management architecture has been designed following a policy-based approach. This approach substantially increments the openness in the management plane at the network and element level, since it isolates the outer parts of the architecture from most of the peculiarities of the management interfaces. When defining network or element management policies, the *deviation from existing standards*[12] provides a criterion to evaluate the openness of the management system; the policy information models would be, in this sense, our "interfaces".

However, given the fact that the policy based management architecture is structured in several layers, there are also internal interfaces that may be of importance when evaluating interoperability. For instance, the interfaces between the decision logic components and the enforcement points should adapt as much as possible to available standard interfaces (such as COPS, P1520,…). The degree on which these standards are used provides a criterion to assess the openness of the designed components.

3.4.3.2.2 Manageability

A criterion that could be used to evaluate the manageability is the *efficiency on managing information*. Information management processes will be realised as part of the monitoring tasks. Two different aspects may be distinguished when trying to assess such monitoring functionalities. On one hand, the traditional monitoring capabilities in terms of data collection, manipulation and distribution; on the other hand the manageability of the monitoring processes, influenced by the policy-driven approach that has been selected.

In the first set of criteria, the *effectiveness of information distribution mechanisms* could provide useful feedback on the feasibility of the currently defined management architecture. The delay incurred between the moment a monitoring information is produced and the moment it reaches its destination, being finally consumed, offers a metric to indirectly measure the system capability to appropriately distribute such information. This indicator is critical when the network state quickly varies, since it could lead to incorrect management decisions based on obsolete information.

[12] Currently the more widely accepted standard is the IETF Policy Core Information Model.

4 CONCLUSIONS

This deliverable marks an important step in the FAIN project: The transition from the design and implementation phase to a phase in which the first results of the integration of the FAIN components will be visible. The results will be made visible and assessed by setting up a test-bed and by carrying out an evaluation of the functionality and the performance of the active network nodes and the corresponding management systems. To this end, a comprehensive evaluation framework has been established and a plan for setting up the test-bed has been defined.

To the best of our knowledge, an attempt to define an evaluation framework which would allow to compare different active network architectures among each other and with traditional, non-active network approaches has never been reported so far. In this deliverable, we defined a framework which allows a comparison between different architectures in a very regular way, using elements of a functional specification, a set of carefully chosen properties and – for each property – a list of features that can be assessed in the context of the transport, control or management plane and in a node-local or a network-wide scope. The actual evaluation of the FAIN architecture will show whether this novel evaluation scheme will fulfil its promise.

Even though the plan for setting-up the test-bed was placed in an appendix, it represents an important part of this deliverable. It will be even more important in the time after submitting this deliverable, as it will evolve into a handbook that will be needed and used by all partners who committed to install an active network node and/or a management station.

While it is obvious that this handbook needs to evolve over time, this is also valid for the evaluation plan. This deliverable just defines the framework, i.e. it sets forth the conceptual approach we take towards the evaluation of a complex system. In order to carry out the actual evaluation, this framework will have to be filled with the concrete parameters to be used in the evaluation. This will remain a task for the next future, and it is also a reason why this document is considered a living document, which needs to be kept up-to-date during the remaining time of the FAIN project.

5 REFERENCES

[1] Requirements Analysis & Overall AN Architecture, FAIN Deliverable D1

[2] Initial Active Network and Active Node Architecture, FAIN Deliverable D2

[3] Initial Specification of Case Study Systems, FAIN Deliverable D3

[4] D. Scott Alexander, Bob Braden, Carl A. Gunter, Alden W. Jackson, Angelos D. Keromytis, Gary J. Minden, David Wetherall, "Active Networks Encapsulation Protocol", Draft RFC, July 1997.

[5] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden, "A Survey of Active Network Research", IEEE Communications Magazine, Vol. 35, No. 1, pp80-86. January 1997.

[6] Campbell, A. T, H. De Meer, M. E. Kounavis, K. Miki, J. Vicente, and D. Villela, "A Survey of Programmable Networks", *ACM Computer Communications Review*, Vol. 29, No. 2, pp. 7-24, April 1999.

[7] Next Generation Networks Initiative, http://www.ngni.org/overview.htm

[8] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, The Network Working Group of the IETF, March 1999.

6 APPENDIX 1: CONCISE INTRODUCTION TO THE SESSION INITATION PROTOCOL (SIP)

SIP is a protocol specified at IETF in RFC 2543 [8], in the form of a proposed standard. SIP is a text-based protocol, similar to HTTP and SMTP, for initiating interactive communication sessions between users. Such sessions include voice, video, chat, interactive games, and virtual reality.

There are two types of elements in the SIP architecture:

- User agent: this is an and-user terminal that participates in a session. The UA (user agent) usually contains two components: an user agent client (UAC) and an user agent server (UAS). The UAC is used to initiate a call and the UAS to answer a call. When both components are present on a terminal, the latter can take part in a peer-to-peer connection.
- Network server: there are two types of network server, proxy and redirect. The role of the proxy server is to route SIP messages. It works very much the same way a mail server relays e-mail messages. Redirect servers tell user agents to which proxy they should send their messages.

The following figure shows a typical message exchange between two SIP phones when one is trying to call the other.

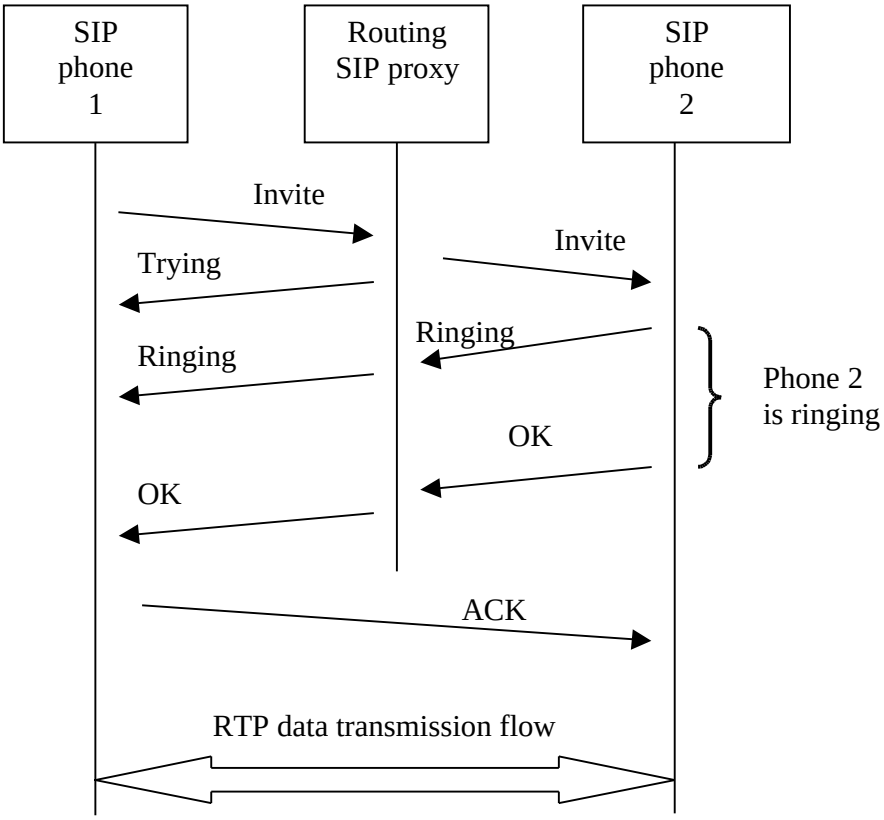


Figure 9 – SIP message exchange illustration

7 APPENDIX 2: PLAN AND GUIDELINES FOR SETTING UP THE TEST-BED

7.1 INTRODUCTION

This section presents a plan and a set of guidelines for the set-up of FAIN active network test-bed. This test-bed is going to serve as a permanent experimental network for active network technologies up to the end of the FAIN project and possibly beyond. However, the test-bed set-up is constrained by the FAIN milestone demonstrations in April 2002, since these demonstrations will use FAIN test-bed as an essential underlying infrastructure. Thus, the guidelines presented here represent our current view of the test-bed set-up process with the focus on successful preparation of project demonstrations.

Initially, FAIN test-bed will not include FAIN active nodes of Type B and Type C, except for an initial version of node Type C, which will be provided by one of the partners for the project demonstration in April 2002. We plan to add full-blown versions of these node types later on.[13] Thus, FAIN test-bed will initially comprise only management stations and FAIN active nodes of Type A, i.e. nodes based on standard PC with Linux as a general OS and PromethOS as an AN platform. Building the test-bed will involve several phases:

- Setting up a dedicated Linux-based PC at partners' sites
- Installation and configuration of PromethOS on the PC based ANN
- FAIN management stations set-up
- Network set-up
- Installation and configuration of service (application scenario) specific code.

The project milestone demonstrations M3, M4, and M5 are planned in April 2002. The timeline for test-bed set-up activities is defined so as to provide a timely delivery of operational and stable active network test-bed to demonstration developers.

Additionally, this document also presents some procedures for managing and operating the FAIN test-bed. These procedures are aimed at easing network set-up, maintenance, and troubleshooting and consequently maximise the test-bed network stability and availability.

7.2 SETUP OF PC-BASED NODE

FAIN active nodes Type A are based on the combination of standard PC and Linux operating system These nodes shall be dedicated systems, i.e. they shall be used solely as nodes in the FAIN test-bed. FAIN active technologies leverage the services provided by standard middleware technologies (e.g. CORBA, Java), which can require considerable amount of resources. Additionally, FAIN active technologies themselves can be demanding and active nodes may be required to handle broadband connections, so it may be necessary to set minimum hardware requirements (CPU, memory) for active nodes. An active node with 800Mhz Intel P-III and 256 MB RAM currently seems to be a sensible lower bound.

PC based active nodes will have to be configured such that they allow remote access to other partners for the purpose of installation an maintenance of active platform and active service specific software. This means that user accounts groups have to be set-up, a secure file transfer service has to be started, and a secure remote login service has to be started on the node.

The relevant information about the node has to be distributed to all other test-bed sites in a fashion, which provides at least some level of security (e.g. BSCW server can be used for this, since it provides basic security protections). The information that has to be disseminated includes:

[13] Note that these node types will still be under development within FAIN and they will be tested in the respective partner's local set-up, they will just not be experimented with in the active network test-bed until after M5.

- IP address of the PC-based node
- Open accounts for test-bed partners (login names) and appropriate passwords

Finally, participation in end-to-end experiments will require each site to configure additional PC, which will act as an end-host in application specific scenarios. Furthermore, end hosts will require additional equipment, such as a sound card, speakers, and a microphone for the purposes of testing video applications within the FAIN test-bed.

7.3 AN PLATFORM SET-UP

PromethOS is the AN platform that will be used in FAIN active node types A and B. PromethOS is an extension to linux kernel and leverages kernel modules mechanism to effectively turn a linux based PC into an active network node.

AN platform set-up includes the following steps:

- installation of PromethOS
- installation of any other software, that PromethOS may require (e.g. external kernel patches),
- configuration of PromethOS
- local testing of PromethOS

PromethOS installation may require patching the kernel (both with patches distributed with PromethOS and external patches), (re)compiling PromethOS and the kernel, and rebooting the node. The detailed installation and configuration procedure may change as the PromethOS framework evolves, so we do not provide these details here. However, these details will be available to partners via two channels: they will be distributed as part of the PromethOS framework and the latest revision of installation guidelines will be available at the central test-bed repository.

7.4 SETUP OF MANAGEMENT STATIONS

FAIN management architecture involves two kinds of management stations. There is a number of Element Management Stations (EMS), which is responsible for managing particular active network nodes. In addition, there is one Network Management Station (NMS) per domain, which is responsible for coordinating the work of EMSs and thus maintaining a consistent view of an active network throughout domain and keeping the active network in an operational state.

For experimental and demonstration purposes up to milestone M5 we envisage the set-up of five EMS stations and one NMS station,[14] which means that we will initially be experimenting with a single AN management domain. Not all sites will host an EMS, which means that some EMSs will manage more than one active node. Similarly to FAIN active node Type A, both types of management stations are based on the combination of commodity hardware PC, linux OS and the NMS/EMS software developed within FAIN. The management stations shall be distinct from active nodes, i.e. sites hosting a management station shall have two dedicated PC connected to FAIN test-bed: one acting as an active node and the other acting as an EMS/NMS.

The set-up of a NMS/EMS includes the following steps:

- installation of the NMS/EMS software
- installation of any other software that NMS/EMS system may require
- configuration of management stations
- local testing of managements stations

Detailed installation and configuration guidelines will be distributed as a part of the NMS/EMS software distribution package. These details and examples of working configuration scripts will also be available to partners via the central test-bed repository.

[14] Besides the “official” NMS site, other partners plan to do local experiments with the NMS station.

7.5 TEST-BED NETWORK SET-UP

FAIN test-bed will be set-up as an overlay (virtual) network on existing network infrastructure. If QoS is not an issue in tests then public Internet can be used as a suitable underlying network. However, if QoS aspects are important then GEANT can be used as an underlying network.

Overlay network is based on IP tunnelling and is created by appropriately configuring point-to-point tunnels between specific active nodes. The choice of tunnelling technology depends on the additional requirements. Simple IP tunnelling, which is supported by recent linux kernels, can be used when no additional requirements apply. However, if FAIN test-bed traffic is considered to be confidential, then one of the IPSEC implementations for linux (e.g. Free S/WAN) can be used to create secure tunnels between FAIN test-bed nodes. In this case tunnel set-up is more complicated, since it requires installation and configuration of Free S/WAN software package.

The layout of tunnels is determined by the choice of topology. Proposed topology for FAIN test-bed is a hierarchical tree with cross connections. The advantage of this topology in comparison with the full mesh is that the later provides only single hop paths between active nodes, while it may be more interesting to test applications over multi-hop paths. On the other hand, a tree with cross connections provides alternate paths between nodes, which is not the case with a simple tree topology. Finally, contrary to full or partial mesh, a carefully constructed tree topology accommodates for the fact that some partners will have a low bandwidth connection to the test-bed due to corporate policy. Configuring these sites as leaves of a tree we can make sure that these nodes can only be used as test-bed access nodes by respective partners testing their applications. They cannot be used as transit nodes for other traffic and thus become bottlenecks.

Management station are going to use the FAIN test-bed as a communications infrastructure for remote management of nodes. Nevertheless, management stations are organised in a structure of their own. This structure does not represent a topology per se, but instead depicts in a three level hierarchy the relationships between management stations and their respective managed nodes. For example, in order for a NMS to be able to manage a set of EMSs in its domain, it has to be configured with the addresses of these EMSs. Similarly, each EMS has to be configured with the address of all active node that it manages. Finally, when there is more than one management domain, a NMS also needs to be configured with the addresses of all other NMSs in order to provide for interdomain management of active networks. However, in the period till milestone M5 we restrict ourselves to a single management domain, i.e. a single NMS. Thus, management stations in the FAIN test-bed are going to be organised into a three-level tree. This structure is going to be set-up simply by appropriately configuring management stations and it does not require the set-up of any additional tunnels.

FAIN also aims at interconnecting its test-bed with other existing active network test-beds, most notably the Abone, however this remains as an objective in the project year 3. Nevertheless, technical merits of this goal have to be investigated and incorporated into test-bed set-up plan from the beginning in order to minimise workload later required to implement the interconnection with Abone.

7.6 INSTALLATION OF SERVICE SCENARIO CODE

FAIN is developing several active applications, which will be used for the purposes of demonstrating the advantages of FAIN, approach to active networks and investigating how these advantages can be optimally exploited by active network aware applications.

The scenarios for these applications may involve one or more active nodes and one or more active and passive end-hosts (terminals and servers). Thus, software implementing application specific scenario may consist of several modules, each of which must be installed on a different node/host in the FAIN test-bed: some must be installed on client machines, some must be installed on various servers (Web server, streaming video server), and some must be placed in a code repository and then dynamically installed by active nodes during the test/demonstration of the application. This may require a coordinated effort of several partners, based on the application specific scenario. and respective guidelines provided by application developers.

Thus, installation of service scenario code involves the following steps:

- set-up of the required infrastructure within FAIN test-bed, e.g. web servers, streaming servers, client end-hosts
- coordinated installation and configuration of service code: if there is more than one partner, each of them is responsible for installing and configuring specific code module(s) to specific places within the test-bed
- testing the application

The service specific code and installation guidelines will be available via the central test-bed repository.

7.7 TEST-BED OPERATION PROCEDURES

In this section we describe some procedures, which are aimed at simplifying test-bed set-up procedures described in the previous sections.

Throughout the set-up process there will be a central test-bed repository, which shall be used by all partners for distributing and/or acquiring:

- Code: PromethOS, NMS/EMS code, application specific code, non-FAIN code (e.g. Free S/WAN, kernel patches, etc.)
- (examples of) working configuration scripts
- guidelines for test-bed set-up
- general information on test-bed, e.g. topology, node addresses, responsible technical contacts per partner, etc.

The advantage of this is that latest releases of code are easy to find and get, it is easy to obtain any relevant information and guidelines, and there is a common pool of knowledge which can be reused by partners

Monitoring tools, combined with graphical representation and alarming tools can greatly simplify the troubleshooting of test-bed during set-up and improve availability of nodes and links in an operational network. Therefore a selection of existing network monitoring tools for Linux will be used for this purpose.

Code developers shall follow the practice of packaging their software with at least the basic (up-to-date) documentation, which includes:

- readme file, which gives a brief description of the code package, including available configuration options and parameters, and simple testing procedures
- install file, which gives installation guidelines

This information is very important for simplifying code installation and configuration. It should be updated whenever code evolves and it should be distributed together with the code.

If we assume that only small portion of software used in FAIN test-bed will require superuser access to active nodes for installation and configuration and that this software will not be updated frequently, then node administrators shall be responsible for properly installing this software to their respective active nodes. This minimises the risk that each test-bed site would face, if it would have to enable superuser access to its node by other partners.

All active network nodes shall provide roughly the same environment regarding the development tools and runtime environments that FAIN active technologies will require (e.g. appropriate compilers, JAVA suite, libraries, etc.). The releases of this environment are expected to change with the evolution of development tools and runtime environments. It may be beneficial to use a common linux distribution at all test-bed sites in order to more easily set-up the same environment across sites.

8 APPENDIX 3: THE FAIN APPROACH IN A FUTURE GLOBAL NETWORKING AND TELECOMMUNICATIONS ENVIRONMENT

The concept of active networks presents a new paradigm for the design of network services. The future success of such an approach depends not only on the novel technical properties of the approach, but also on the business strategies of SP and NP which are driven by technology as well as by market demands. It is necessary to assess the AN approach in the context of SP/NP strategies regarding the development of next generation of networks/services.

Two questions need to be taken into consideration:

- How good is the FAIN approach in comparing with other AN concepts?
- Does the FAIN approach meet the expectations of SP and NP, as a potential concept for the next generation of networks?

To answer the first one, the evaluation criteria as outlined in section 3 of this document will be used. For the second one, established strategic assumptions made by the telecommunications industry, which are relevant to FAIN need to be identified. A detailed analysis of these statements will lead to the criteria for the evaluation of the FAIN approach. These criteria will represent a collection of metrics which will serve to make a comparison between the FAIN approach and other approaches of active, as well as and traditional (non-active) networking approaches pertaining to the service provisioning point of view.

Because the AN approach is still in the research stage, the evaluation will not show, how good the actual AN approaches meet the criteria, but it rather will provide an overall assessment of the potential of these approaches to fulfil these criteria.

The list of criteria developed here complements and refines the general criteria that were identified and described in the FAIN deliverable D1 [1], specifically chapters 2 (Operator's Expectations on Active Networks) and 5 (Requirements analysis).

This section puts active networks within the perspective adopted by the Next Generation Networks (NGN) Initiative of the IST Programme [7].

8.1 The vision of active networks

Current development of network technology is focusing on solving a large number of problems (QoS, openness, mobility, security, scalability, manageability etc.) and is being addressed by different organisations. Besides coordinating work between these developments there is still a lack for a vision leading towards the convergence of different technologies. AN could take this role as a general concept or vision for developing the next generation of networks, as shown in the following figure. It illustrates how conceptual ideas from TMN intelligent networks, mobile agent technologies, open technology-independent API development combine with IP technology to form a new technical approach that could solve the flexibility and scalability problems of today's telecommunications industry.

8.2 Strategic Assumptions

1) IP-technology will have a dominating role in the bearer network operated by telecommunications companies, but it still needs significant improvements

IP is the key technology for next generation networks. Telecommunication companies need to make profits and to compete with new entrants by differentiating their services, but the complexity of current networks, with multiple platforms, has made it difficult for the industry to adapt their services. IP has provided a focus for this change because it offers interoperability as well as a seamless service. In fact, IP networks begin to carry more and more commercial traffic. The data transport volumes on the Internet has exceeded the volume of voice traffic in leading markets. Furthermore, IP technology has the advantage of relatively low cost facilities and of the ubiquity of services, which have been spread in the Internet.

In spite of the Internet boom and the optimism that was associated with it, there are many barriers on the way of IP technology establishing itself to become the dominating network technology and to replace current circuit-switched infrastructure:

- Lack of QoS support
- Difficult to control and manage
- Weak traffic management
- Low level of security

Solving these problems is both an opportunity and a challenge for innovative IP-based network concepts like active networks. A successful active networks approach will have to demonstrate progress in these areas.

2) Universal interworking between networks requires a flexible approach

The term *flexibility* reflects the fact that we need networks with “soft” boundaries, so that they can be made to interwork with one another more easily. This is relevant both for networks operated by different carriers and to different networks operated by a single carrier, as well as for networks operated by companies.

IP is an open technology, but lacks effective functionality for the support of system operation and management. The PSTN is a closed system, but has a well-developed operational support system. However, neither network has the complete capability that is needed for a multiservice platform. Many more items can be added to the list of criteria, but simply scanning these shows how much progress remains to be made.

The network flexibility can be assessed by meeting following requirements:

- *Defined interfaces*: There must be defined APIs and protocols for interfacing to the network.
- *Negotiation capabilities*: The network must be capable of entering into negotiations associated with service requests and not simply return a `go' or `no go' response.
- *Location independence of addressing*: The network must decouple the static home address of the user from the address that is in use at any instant by the user at their current location.
- *Virtual topologies*: It must be possible to construct closed user groups and to manage them as single entities.
- *Service translation* (for example, for VoIP to circuit-switched networks): The network must advertise the level of transcoding and signalling translation it can provide.
- *Ability to export and import configuration information:*. Operators must (securely) export configuration information, such as routing tables, needed for the delivery of services across carrier boundaries.
- *Ability to export and import subscriber service information:*. The service profile offered to a user needs to be agreed between carriers.
- *Ability to exchange billing and accounting data:* Needed for e-commerce as well as inter-carrier settlements.

Again, a successful active networks approach will have to demonstrate progress in these areas.

3) Operating a full-service-platform remains an important requirement in the network strategies of incumbent telecommunication companies.

A Full-Service-Platform contains different physical structure and technologies in the network level but provides its services to users or customer as a consistent and transparent service platform by using a service middleware. Network resources/capabilities are provided as independent building blocks. A middleware supporting network transparency is required.

4) QoS: Over-provisioning versus traffic management

Over-provisioning of bandwidth alone does not present the long-term solution for QoS problems. There is a need to combine the bandwidth available with ability to control traffic, to be able to exploit a trade-off between the degree of over-provisioning and the degree of traffic management applied.

Different protocols in at different network areas

Different concepts/protocols for providing QoS service in different areas of a network are being developed. MPLS, Diffserv, RSVP are the current approaches for providing QoS, whereby the RSVP is likely to be used only at the edge of the network and within enterprise networks to allow the application to signal the QoS requirements. Diffserv will be used at the carrier network edge and allows a carrier to offer several classes of service. MPLS is used in the core, primarily for aggregation and traffic management. To provide QoS-aware end-to-end services across different areas of a network, a concept to combine these protocols is needed. Mapping of QoS definitions at the network boundaries is a possible solution.

Inter-carrier QoS

Most of the work on QoS is concerned with implementation within single carrier networks. There are concepts for extending QoS between carrier networks (e.g. using automated SLAs or bandwidth brokers), but these are still in the research stage. This lack of workable approaches is a barrier to interconnection between carriers and needs to be removed as soon as possible.

Requirements

- Protocols supporting QoS mechanisms in all areas of networks
- Concept to offer QoS-aware end-to-end services across the different areas of a network
- Support for inter-carrier QoS

5) Trends in services

Three major trends will occur in services: *Unification*, *customisation* and *multimedia content delivery*.

Unification is the process of breaking down the barriers between services, the networks that deliver them, the terminals used and the type of content. The goal is to offer service to users *anytime* and *anywhere* by *any terminal*. This requires on the one hand the *true mobility support* and on the other hand terminal and network adaptation.

Customisation refers to the ability of users to modify their service portfolios in real-time. They may choose to have different service profiles set up for business use, home use and for use when travelling. Different quality levels may be selected from the network, according to the usage context.

Multimedia content delivery With the advent of streaming and the rise of popular compression technologies such as MP3, delivery of multimedia content is becoming increasingly important.

These trends affect the network in several ways. In this context, telecommunication companies expect from the new network concepts:

- improved roaming protocols, such as mobile IP
- mobility supporting functional, such as mobile security (e.g. VPN supporting mobility), location management.
- Network and terminal adapting (e.g. VHE)
- online service configuration and provisioning, integrating into carrier's OSS, such as a subscription management system.
- provision of multimedia in realtime
- if the carriers are involved in the payment process, a requirement to process secure payments must exist.

6) **The need of more manageability and coordination**

Network layers must co-ordinate

Building a network that delivers high performance end-to-end requires co-ordination of the various layers. This is not present today for a variety of historical reasons. A goal for the next generation of network infrastructure must be to ensure effective interworking between, for example:

- the layers 1,2 and the transport layer
- policies, directories and QoS
- applications and the network.

Policy-based management must improve and link to commercial goals

Carriers need a variety of provisioning and customer management capabilities to turn the infrastructure into a commercial service platform. A present-day problem for carriers is that the operational control of the network is separate from the commercial policy. Operational control is very weak, since it relies on network management tools that have limited control over the routing infrastructure. Commercial policy, which defines the priority of customers, is an overriding concern for carriers that have service-level agreements (SLA). However, it is hard to translate abstract policy specifications into instructions that are understood by routers. If a mistake is made, then, in a worst-case scenario, low-priority traffic can bump off from the network traffic that has a higher commercial significance. Solving this problem requires better policy-based management systems.

Requirements

- Scalable concept for effective interworking between different layers
- Policy-based management system to support the implementation of commercial policy on the networks.

7) Security

Currently, the key security issues on the network level are:

- More protection of data being transported in networks for manipulating, destroying and intercepting
- More protection against denial of service attacks - whereby networks are flooded with artificial traffic to prevent users from obtaining a useful connection
- Both user traffic and management traffic must be protected

Since no security solution is perfect, waiting for the ultimate security technology is not a valid strategy.

8) Protection of investments

Telecommunication companies need to have the assurance that their investments in equipment and personnel are protected within a long-term strategy. Undoubtedly, using active networks technology implies that major parts of the network infrastructure needs to be replaced or at least updated. In addition, the operation of an active network infrastructure will be more complex (as seen by the operator) since it potentially will provide a more elaborate and flexible set of services. This implies that the personnel of the operators will need to be educated in the use of this new technology.

As anyway networking equipment has to be replaced in a certain pace, and operator personnel has to kept up-to-date with technology, the requirement of protection of investments is not a major hindrance to the introduction of active networks. However, it is crucial that active networks can be introduced in a piecewise fashion, i.e. there must be a transition strategy allowing for step-by-step introduction of this new technology. The transition strategy will have to demonstrate that by partial introduction of active networks technology, an improvement of the services and the business opportunities will result.

8.3 The role of this chapter in the overall evaluation of FAIN

The strategic assumptions discussed in the previous section will serve as a guideline and checklist for evaluating to what extent the FAIN architecture complies with the expectations of the telecommunications industry.

