Programmable Active Networking supporting Next Generation Multimedia Services in Satellite Networks

Wolfgang Fritsche (lead author; email: fritsche@iabg.de; phone: +49 89 6088 2897; fax: +49 89 6088 2845)
Karl Mayer (email: mayer@iabg.de; phone: +49 89 6088 2066; fax: +49 89 6088 2845)
IABG, Einsteinstrasse 20, 85521 Ottobrunn, Germany

Prof. Peter Kirstein (email: p.kirstein@cs.ucl.ac.uk; phone: +44 20 7679 7286; fax: +44 20 7387 1397)
Saleem Bhatti (email: s.bhatti@cs.ucl.ac.uk; phone: +44 20 7679 3249; fax: +44 20 7387 1397)
Lionel Sacks (email: lsacks@ee.ucl.ac.uk; phone: +44 20 7679 3976; fax: +44 20 7388 9325)
Stefanos Zachariadis (email: s.zachariadis@cs.ucl.ac.uk; phone: +44 20 7679 0156; fax: +44 20 7387 1397)
UCL, Gower St, London, WC1E 6BT, United Kingdom

Abstract

Programmable Active Networking is one approach to providing open service engineering for telecommunications and networked services. The overall objective is to provide a platform which may be tailored to provide all the functionality required for many applications. Through the years a number of approaches have been developed, some of which have found substantial success, others are still in the research arena.

The vision of Programmable Active Networking is to increase the systems flexibility and to allow components and services to evolve and develop, without vendors or operators needing to completely re-build their systems. From this aspect especially the space segment of satellite networks could take benefit. The first commercial regenerative satellite systems are now available, providing mainly switching services on the link layer; in future also IP routing functionality will become available. However, from the design of these satellite systems throughout their long time of operation the continuous development of multimedia services progresses significantly, coupled with rapidly and dynamically changing requirements to the underlying networks.

As all these requirements cannot be foreseen during the design phase of satellite systems, Programmable Active Networking provides a mean for a dynamic adaptation of satellite networks to these newly arising requirements. During the installation of satellite networks Programmable Active Networking functionality is added to the various involved components, to satellite hubs, to satellite receiver equipment, or also to the space segment itself. This functionality will allow throughout the lifetime of operation of satellite components the remote uploading of new code and its execution in a specific environment. This new code could serve different purposes; for example it could add a new functionality, such as a newly available transcoder to be used for adapting multimedia streams to the characteristics of different satellite beams, or it could also transmit just a new policy for configuring an existing transcoding service.

To optimally exploit its benefits, a careful design of Programmable Active Networking functionality for satellite networks is required and needs to especially consider aspects of security, performance and network operation. For example the distribution of active components could be abused for performing network attacks; consequently appropriate security functionality has to be included, like the authentication and integrity protection of the distributed code. To avoid the execution of malfunctioning code on sensitive components like the space segment, one should perform plausibility checks and run the code in protected environments like a Java sandbox. As processing power and memory are expensive especially in the space segment, resource intensive functionality needs to be executed in the ground segment whenever possible. To make the introduction of Programmable Active Networking as economically as possible, its design should be applicable to a broad variety of multimedia services.

I. Introduction

The demand of users and operators on the features and services of the Internet is growing. However, networks are mainly built of closed systems (e.g. developed by router manufacturers) that do not allow a rapid and easy introduction of new services on demand but require a long standardisation process and involvement of manufactures. A solution to this issue is the installation and deployment of open programmable systems at the network edge but also within the heart of the network that can be extended with new features and services and programmed on demand.[1-3]

In order to realize open systems two basic approaches have been undertaken in the research community: In the first approach, called Active Networking (AN), open systems have been developed that allow the rapid introduction of new services through active code that is delivered and instantiated on demand, running in execution environments in active nodes. The second approach focused on specifying and standardizing open interfaces to the resources of network systems (switches, routers, etc.) that could be used by software developers to program and control network nodes in a common standardized way.[4] We will refer to both approaches by the term "Programmable Active Networking" (PAN) throughout this paper.

While programmable active networking have been investigated in detail for terrestrial networks, no research activity has taken into account satellite networks, however, from using Programmable Active Networking especially the space segment of satellite networks could take benefit. The first commercial regenerative satellite systems are now available, providing mainly switching services at link layer; in future also IP routing functionality will become available. However, from the design of these satellite systems throughout their long time of operation the continuous development of multimedia services progresses significantly, coupled with rapidly and dynamically changing requirements to the underlying networks. As all these requirements cannot be foreseen during the design phase of satellite systems, PAN provides a mean for a dynamic adaptation of satellite networks to these newly arising requirements. Hence, the focus of this paper is the investigation of the applicability and potential realization of Programmable Active Networking in satellite networks – the ground segment and the space segment - in order to prepare them for next generation multimedia services.

This paper will start in section II with a short introduction into the subject of Programmable Active Networking. Afterwards, in section III, we will investigate the applicability and benefit of using Programmable Active Networking systems in satellite networks from a commercial point of view and outline new business opportunities given by the introduction of PAN functionality concerning the provision of new as well as the enhancement of existing multimedia services. In section IV we will perform an analysis how a realization of for an operational service could look like and which aspects have to be taken into account concerning security, performance, and operation. Thereby, we will show how different multimedia services like transcoding of multimedia streams, intelligent dropping of application data, or reliable multicast can be smoothly integrated into the same PAN-enabled satellite network platform. For this analysis a promising component-based middleware called SATIN[5] has been selected for providing Active Networking functionality. The paper will conclude in section V by summarizing the lessons learnt from a demonstrator testbed, which makes use of a specific Programmable Active Networking implementation for the provision of intelligent and adaptable dropping and transcoding services.

II. Programmable active networking (PAN)

Programmable active networking (PAN) has developed out of the overall enterprise of open service engineering for networked applications. Its objective is to provide an open platform that could be tailored and extended in order to change network capabilities and to provide all the functionality required for many different applications and services. The PAN system consists of programmable active nodes placed at the network edge but also within the network. Essential features are the ability to add new capabilities through instantiation of new software modules on demand and the possibility to deploy third party applications on top of the open platform.

In a PAN system active modules that contain executable code or instructions that are applied on specific packets or streams are delivered and instantiated on demand. Hence, new capabilities can be added to the network without having to re-engineer many parts of the system. Regarding active networking, to distinguish are systems that operate on packet level and systems that adapted to a specific application or session. In packet-level active networking (also denoted as capsule approach), in which each network packet carries code that is applied on itself in active nodes, has been developed in order to flexibly and rapidly introduce protocols and services on network level, e.g. a new IP protocol.[6] However, this approach has weaknesses concerning security and, especially, with respect to performance and scalability since each network node has to process the code encapsulated. Therefore, research activities have focused on transport and application level active networking (denoted also as discrete approach), in which software module that represent a new or updated service are distributed in the network and instantiated in active nodes, located at strategic network points, prior to service deployment.[7] Throughout this paper we will focus on transport and application level active networking.

PAN operation, illustrated in Figure 1, can be divided in three steps: First, one or several modules that contain executable code or instructions are exchanged between PAN-enabled nodes (e.g. downloaded from a code server) in order to instantiate or configure a certain service. Second, exchanged modules are processed in the receiving active nodes, which means that executable code is instantiated on the target system and the parameters transported trigger some action, respectively. Third, PAN nodes participate with other active nodes and/or conventional nodes in an application scenario, using the active processes or instructions in order to fulfill a specific application, e.g. manipulation of multimedia data.

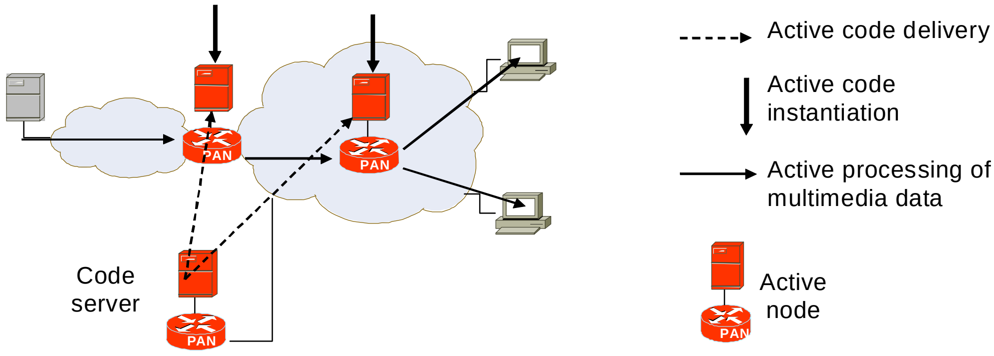

Figure 1. Active networking operation

One has to take into account that the focus of programmable active networking is not the realization of one specific application, i.e. the realization of a specific protocol functionality, but to provide a generic platform, consisting of several PAN-enabled nodes, that supports the introduction in and the operation of new protocols and services while runtime. The applications running on top of the PAN system can use the features of the PAN platform, i.e. the exchange of modules that could be used to upload and execute new processes or just to configure processes running on PAN nodes.

While most Active Networking related activities and projects have been performed in the past, a new research area is emerging, using the active networking principle, denoted as component-based middleware. A component-based middleware supports the dynamic loading and instantiating of components on demand in order to allow the middleware to adapt to environment conditions while operation. Examples of component-based middlewares are GRIDKIT[19] and SATIN[5], the latter will be discussed further in section IV.

III. PAN and its applicability for satellite based Multimedia Services

Prior to investigating how a PAN system could be integrated into a satellite network it is necessary to analyse whether it is applicable and beneficial to enhance a satellite network with programmable active networking features. We perform this analysis by considering several specific application scenarios, use cases, that look promising for being Next Generation Multimedia Services. The most promising are given in the following:

- *Reliable multicast scenario*: In the first application scenario envisaged applications use reliable multicast (RM) protocols in order deliver multicast data in a reliable way. While reliable multicast protocols operate end-to-end, RM processing in intermediary network nodes would improve the performance with respect to packet loss rate, bandwidth usage, or delivery time. For instance, satellite network nodes that cache multicast packets and perform retransmission in case of losses on behalf of the multicast source would reduce overall network traffic, increase the speed of multicast data delivery, and relieve the processing load in the multicast source. FEC coding between satellite network ingress and egress nodes could help to reduce packet loss rate in the satellite network and NACK suppression in satellite nodes would reduce backward traffic. Although promising, the question is which RM protocol to implement in hosts and satellite network nodes. A variety of reliable multicast protocols exists and some of them are just under standardization: some use functions like Automatic Repeat reQuest (ARQ) mechanisms with positive acknowledgements (ACK) or negative acknowledgements (NACK), others use packet level Forward Error Correction (FEC) methods in proactive or reactive manner.[8-11] The suitability of RM protocols depends on the respective application, the multicast data type transport, and the network conditions. Hence, it is desirable to have the flexibility to adapt the RM protocol to respective requirements, i.e. to upload a new RM protocol on demand while runtime to the relevant nodes. Furthermore, in case of FEC coding, it is desirable deploy an FEC codec that suits best on demand. A PAN system can provide the desired flexibility in case all relevant network nodes, hosts as well as satellite nodes, are PAN-enabled, allowing the on-demand loading and deploying of RM protocol instances. Of course, as with the other scenarios, security, performance, and management aspects have to be taken into account.

- *Intelligent dropping scenario:* Satellite networks often represent a bottleneck for data since satellite link capacity is scarce and expensive. Hence, arbitrarily dropping of packets is applied in satellite hub stations and, in the future, maybe also in the satellite in order to rate limit traffic. Some multimedia streams contain packets with different priorities, for example, an MPEG video stream consists of Group of Pictures (GOPs), each containing Intra coded frames (I), Predictive coded frames (P), and Bi-directional coded frames (B), with B and P frames depending on I frames.[12] Hence, I frames contain data of the highest priority and loosing an I frame means that dependent P and B frames are useless, so intelligent dropping of P and B frames and transmitting I frames is desirable in order to limit the stream bandwidth appropriately while keeping sufficient quality.[13] For this service, a satellite nodes faced to congestion requires a dropping module that understands the structure of the stream and the priorities given. Several different multimedia stream formats are already in use and new ones will be developed in the future. Hence, uploading new intelligent dropping modules to operational satellite nodes on demand is desirable and PAN provides a platform for on-demand deployment. A little modified scenario would be to use dropping modules that do not evaluate the priorities of packets within a single stream but to assign priorities to complete streams according to customer requests or operator demands. Here, the PAN system would provide a platform for dropping module instantiation and for delivering priority policies from network administrators towards satellite nodes that apply dropping. Figure 2 illustrates an example in which the satellite hub station is PAN-enabled and dropping modules are loaded and controlled from an administration node.

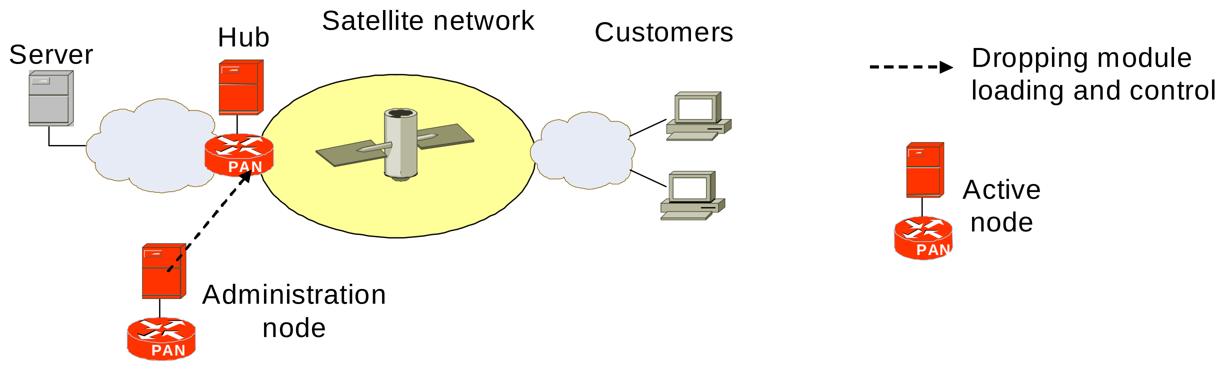

Figure 2. Intelligent dropping scenario

- *Transcoding scenario:* A satellite network usually represents a network part where network conditions change dramatically compared to terrestrial networks with respect to link quality and, most important, available bandwidth. When multimedia data passes a satellite network, it would be beneficial to adapt the multimedia codec and data rate to network conditions and user needs. Since the multimedia source may not be able to provide the best codec or a suitable data rate and may be totally unaware that data sent out has to pass a satellite network, transcoding in satellite network nodes is a promising solution, e.g. in the satellite hub station or in the satellite itself. When converting the codec, recoding is required, which can be performed user transparent in satellite egress nodes or with user involvement in the end user terminal. Since a huge variety of multimedia codecs exists and new ones will be developed in the future, dynamically loading and deployment of codecs to already operational satellite network nodes as well as the control of the transcoding modules with respect to the chosen data rate is preferable. Enabling the respective satellite network nodes (e.g. satellite ingress and egress terminal and the satellite itself) with PAN functionality would provide the desired flexibility.

- *Multimedia multiplexing scenario:* Multimedia application like Multimedia Home Platform (MHP)[14] or MPEG-4[15,16] applications use different media objects to fulfill a certain service. For example, in the MPEG-4 scene composition feature audio, video, and graphics may be composed to a single visual scene at the user side, e.g. representing an e-learning service. Thereby, in case of using satellite networks for media delivery, it is likely that customers of these services are located in different regions with different cultural background. Hence, it is desirable that multimedia components can be adjusted to the customer, e.g. that multimedia objects provided by different media servers are multiplexed and routed in satellite network nodes in an intelligent way. Since media object types and services imaginable are various, a flexible, extensible PAN-enabled satellite node platform that allows the on-demand deployment of new or updated multiplexing software would be a solution. Here, especially the satellite platform looks promising for performing multiplexing and routing of media content since representing a strategic network point.

In conclusion, several application scenarios providing Next Generation Multimedia Services are imaginable that require or at least benefit from a PAN-enabled satellite network. The effort for installing and maintaining a PAN system, however, should not be justified by considering a single application but by taking into account the variety of services that can be based on a PAN system and the flexibility it provides to introduce new services dynamically on demand into an operational network.

IV. Integration of PAN in satellite networks

In section III we have investigated PAN in satellite networks from a commercial point of view and evaluated whether it is applicable and beneficial to deploy PAN functionality to realize and support applications of the next generation multimedia services. In this section we will have a look an the integration of PAN into satellite networks from a technical viewpoint, identify issues regarding architecture, security, and performance, and derive a suitable PAN system realization many multimedia applications could take advantage from. For this investigation, in order to become concrete, we will select a suitable PAN approach

For the integration analysis, of special interest is the space segment. While on-ground PAN functionality could be easily realized in separate maybe high performance network nodes that are inserted in the data path, in the space segment the constraints are much harder since PAN functionality has to be integrated in the satellite payload itself. Hence, prior to investigating integration aspects of PAN in satellite networks in general, we will have a specific look into the space segment.

A. Selection of a suitable PAN approach: SATIN

A variety of programmable active network approaches have been developed so far; hence, to investigate integration aspects of a PAN system in satellite networks for all PAN systems imaginable is not possible. Capsule approaches have weaknesses concerning performance and security so we have chosen to consider a discrete active networking approach for our investigating. As mentioned in section II, a new technology area is emerging that uses the active networking principle: component-based middleware. One of these component-based middlewares is SATIN, developed at the University College London. SATIN supports the delivery of objects – so called Logical Mobility Units (LMUs) - between SATIN instances. Within these objects anything could be transported from a parameter set that configure a certain process to a dynamic SATIN component that is loaded and instantiated on demand. Furthermore, SATIN supports the deployment of an automatic advertisement and discovery mechanism that allows component repositories to advertise the existence of a component with specific features and SATIN instances to discover them.

SATIN has been implemented using Java 2 Micro Edition (J2ME), with the Connected Device Configuration (CDC), Personal Profile, designed for embedded systems with limited resources. In combination with a Java Virtual Machine (JVM) that provides an execution environment for dynamic loaded components, SATIN provides a full featured programmable active networking system. SATIN in relation to the huge programmable active networking research field is illustrated in Figure 3. Thereby, two dimensions have been considered:

- *Granularity:* In packet-level active networking (capsule approach) each network packet transports code or instructions that are just applied on the packet itself, hence, a new service (e.g. a new network protocol) can treat each packet independent, providing packet granularity of active networking. Transport and application level active networking (discrete approach) support services that operate on session level (stream granularity) or network level (network granularity), i.e. the service is applied for all traffic belonging to a certain network or user. Clearly, packet granularity supports the broadest range of active services (highest flexibility) that can be developed and supported.

- *Content:* Active modules that are exchanged between active nodes can contain binary code, interpreted code, scripts or just parameters to control processes. The flexibility the PAN approach offers clearly depends on this content. While the distribution of code allows new processes to be instantiated on demand, delivering scripts requires at least the a priori installation of an appropriate script interpreter, and delivering parameters requires that a process that is controlled be the parameters is already running; hence the degree of flexibility of a new service that is instantiated on demand is highest when having a PAN approach that allows the distribution and instantiation of code.

Considering SATIN, various content can be transported in LMUs, from components (code) to parameters. New SATIN services are usually instantiated to facilitate a certain user service that may change on a stream basis; hence, SATIN supports a huge part of programmable active networking, except packet granularity.

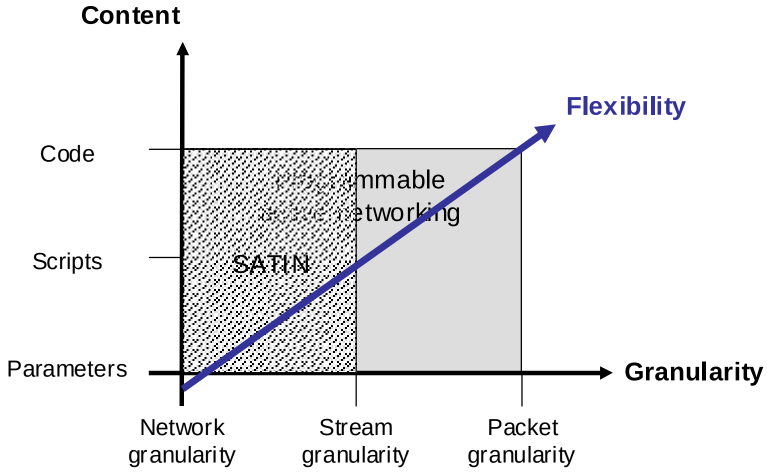

Figure 3. Programmable active networking and SATIN

When we look at the area where LMUs just transport parameters that configure or control a certain process, we are close to a usual signaling protocol. No clear borderline between SATIN operation and a signaling protocol can be drawn here; however, this is also not necessary. It is the functionality of SATIN to provide a flexible platform a variety of service can be based upon, from parameter exchange to the instantiation of code.

B. SATIN in the satellite platforms

Looking at the space segment, especially at the satellite opportunities for new services that could take advantage from PAN and constraints given are contradictory. On the one hand side, satellite payloads should be lightweight and small to reduce the costs for launching, and power consumption should be as low as possible to reduce the size of the battery. These constraints limit the available memory and computational power in the satellite. Furthermore, stability and reliability of the payload is crucial, demanding for low complexity. On the other hand side, large coverage, different conditions on different spot beams, and support for high data rate multimedia applications demand for sophisticate functionality in the space segment. When considering recent developments in the space segment area, it becomes obvious that the satellite payload becomes more and more sophisticated. Systems like the AmerHis[20] satellite or the WEB 2000 and the WEST architectures[21] perform switching and multiplexing at link layer between uplink and downlink spot beams, hence, supporting meshed connectivity between arbitrary on-ground stations. Furthermore, the UK-DMC satellite even implements an IP stack, however do not route IP packets between uplink and downlink beams but uses TCP/IP for communicating with the network control centre. These examples show that it is not unrealistic that a future satellite will implement an IP layer that performs routing, and on top of this a programmable active networking instance like SATIN could be integrated. For SATIN integration in the satellite, the following functions are required:

- a Java Virtual Machine SATIN components as well as dynamically loaded processes can be executed in,

- an appropriate mechanisms for filtering packets that require active processing from the normal routing routine and forwarding them to the respective active processes, and

- appropriate interfaces to satellite platform parameters that could be deployed by active processes in a read-only manner to adjust an active services, e.g. adjust the multimedia codec to the experienced bit error rate on a specific satellite link.

C. PAN-enabled satellite architecture

In order o fully exploit PAN functionality and to justify PAN system integration and maintenance effort, the first step in designing a PAN-system is to choose a suitable architecture that supports the realization of a variety of customer services. Having discussed various scenarios in section III, we will choose here a network architecture the Intelligent dropping scenario, the Transcoding scenario, and the Multimedia multiplexing scenario can be realized with (see Figure 4). The network architecture consists of media servers in the Internet that host and serve multimedia data. These servers are connected to users via terrestrial Internet links but also via a satellite network, the latter being of special interest here. The satellite network consists of a Hub station, two branch station terminals (ST1 and ST2), and a multiple beam regenerative satellite (SAT). Satellite links are meant to be bi-directional and the branch stations are in different spot beams.

For supporting all the scenarios mentioned above, the PAN system architecture needs to consist of a PAN instance in the Hub station, a PAN instance in the satellite, and a PAN instance in one administrator terminal at each user site. Depending on the scenario, not each of these nodes may require PAN processing. Additionally, the PAN system requires component servers somewhere in the Internet from where active components are loaded on demand by active nodes.

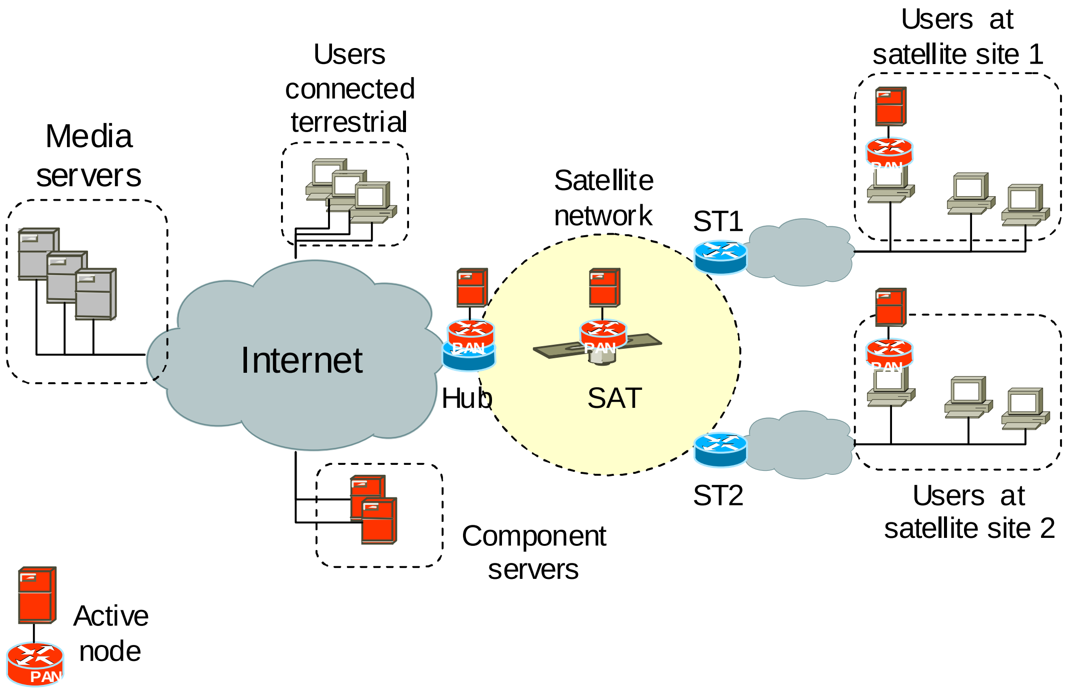

Figure 4. PAN-enabled architecture

For illustrating how a new service could be instantiated and operated, we will use here the Intelligent dropping scenario as an example. First of all a business case is required; here we envisage that users arbitrarily connect to media servers in the Internet for streaming audio and video files. Each satellite branch site has limited bandwidth available and the overall traffic requested by the users exceeds the available bandwidth. This leads to arbitrarily dropping of network packets in satellite network nodes that experience congestion. Most likely the satellite hub station experiences congestion, but in case the satellite itself performs multiplexing of several uplink channels to one downlink channel and overbooking is applied, in future also the satellite may be faced to congestion situations. Arbitrarily dropping of packets leads to bad performance of every stream, which is unacceptable for the users, and the users request the satellite network provider to active an Intelligent dropping service.

As mentioned in section III, this service can be twofold. In the first service, active modules parse multimedia streams and evaluate the different priorities of packets within a single stream, e.g. the active modules give priority to MPEG I frames and drop P and B frames. Upon user request, e.g. via email, the satellite network provider chooses the appropriate active modules and uploads them without further user involvement onto active nodes that experience congestion. In the second service, upon user request a dropping control module is loaded in the user terminal (this is only applicable to the SATIN-enabled administration terminal) and a dropping enforcement component is loaded in the Hub station (or also in the satellite). Afterwards, the administrator interacts with the control component and specifies his stream priorities (e.g. give priority to audio traffic among video traffic), and the control components sends priority policies via a SATIN LMU towards the enforcement components that afterwards applies the policies.

Similar to the Intelligent dropping service, active modules that perform transcoding or multimedia multiplexing could be loaded from component servers to the relevant SATIN-enabled nodes, realizing new advanced services. It has to be noted, however, that the Reliable multicast scenario outlined in section III cannot be realized in this architecture since this scenario requires multicast senders and all multicast receivers to be PAN-enabled and in the architecture envisaged here multimedia servers and user terminals are not PAN-enabled, except one administrator terminal per satellite branch site.

D. Security

When considering security aspects, two different view points have to be taken into account. First, security mechanisms that are deployed in the Internet have an impact on the applicability of PAN processing. Service processes that are running on top of a PAN system usually require access to application data and Intellectual Property Rights protection of multimedia content would prevent active processes from data manipulation. Furthermore, channel protection mechanisms like IPsec even prevent the manipulation of transport layer parameters. This aspect will not be discussed further in this paper.

Second, the PAN system has to be protected against attacks and bugs that threaten the integrity of the system. Vulnerable to manipulation are active modules exchanged. Successful manipulation of active software modules would empower an attacker to introduce malicious code into the PAN system, leading to disruption or even destruction of the whole system, requiring integrity protection of active modules while transport. Furthermore, active code must be authenticated; otherwise an attacker would be empowered to masquerade to be a component server and upload malicious code. Similar, a successful manipulation of SATIN advertisement and discovery messages could mislead a SATIN instance to contact a malicious component server, requiring integrity protection and authentication of advertisement and discovery messages. Integrity protection and authentication could be provided e.g. via digital signatures or message authentication codes. Note, that deployment of security mechanisms requires the existence of a security infrastructure like a public key infrastructure, which is a research topic on its own.

A system is never fully secure. Identification of malicious code is difficult. Plausibility checks prior to loading code are difficult to perform on code modules since not each state of the code module can be checked a priori. Plausibility checks are most reasonable in order to check parameter values. Hence, in order to limit the effect of a successful attack as well as the impact of bugs, active modules should be executed in a restricted environment. The Java sandboxing concept provides such a restricted execution environment where the software developer can choose the access rights of an active module running within the sandbox to the outside. This is especially important in the space segment since integrity of the satellite system has priority and the impact of bugs and malicious code must not affect normal operation.

One has to be aware that security mechanisms consume resources, especially CPU power. Iin the space segment CPU power is a scarce resource; hence, security algorithms that require a lot of CPU power should be best performed in on-ground proxies, connected to the satellite via secured (e.g. scrambled) channels. For instance, plausibility checks should be performed on-ground and active code that is intended to be uplinked should be tested in an emulation environment prior to loading it to the satellite in order to identify malicious code as well as bugs.

E. Performance

Several aspects concerning performance have to be taken into account. Most crucial is the available CPU power and memory that can be used by active processes. While on-ground active processing could be realized in server clusters, the limitations are severe in the space segment where CPU power and memory are scarce resources. Hence, the type of processes and the number of processes executed in the satellite have to be chosen carefully. In general, three key approaches are suitable to improve the performance of active components:

- compilation to native machine code,
- code optimisation
- and hardware assists.

As we have chosen SATIN as PAN approach, written in Java, the importance of full cross-compilation to native machine code depends on the work done at the Java level. As the space platform processors are well defined it is entirely appropriate to specify a cross-compiler to be used for components deployed in the space segment. The next stage after cross-compilation is code optimisation; in particular, substantial gains in performance can be found by re-factoring the PAN code base prior to deployment. For issues that are reasonably well understood in advance, it would be appropriate to deploy ASICs with components tuned to the target application. There is a natural contraction between the use of traditional ASICs and the versatility of the PAN approach. One may consider an ASIC with a CPU to support the OS and VM. In this case, the PAN components would have Java Native Interface (JNI) APIs into the ASIC. A forward-looking approach is to consider the emerging generation of FPGA components. These can also have CPU cores, but their high performance functionality is provided by the re-programmable generic elements. These can perform arithmetic, logic, storage and DSP operations. However this functionality can be re-programmed to configure the units for a wide variety of applications.

Crucial for sufficient performance of the overall system is an appropriate monitoring and management of active modules and their respective resource consumption. Different services, e.g. belonging to different users, must not interfere with each other. Hence, the active modules that comprise a certain service should get assigned a limited amount of CPU power and memory while instantiation in active nodes. Deploying an appropriate resource management, the PAN operator is enabled to allocate resources on his priorities (e.g. customers may have different priorities) and to plan and predict service operation, i.e. the operator is aware of spare resource capacities or whether the system is already overloaded. This is especially crucial in the space segment. There, resources are very limited and uploading and executing of a new active process that blocks any other processes could disturb normal satellite processing severely, and resetting of out of control processes might be difficult in the space segment.

V. Conclusions

Many results have been obtained during the performance of the work described in this paper. This chapter outlines briefly the main conclusions.

There are mainly two variants of PAN, the capsule and the discrete approach. Within the packet level or "capsule" approach each packet sent from the sender to the receiver contains its own active code, that is, this active code determines which treatment the respective packet should experience from the intermediate active nodes. For example the active code could simply be some parameters, which will parameterize some packet processing functions at the intermediate nodes, or it could be the function code itself. Within the transport / application level or "discrete" approach the active code is not contained within the packets, but will be loaded to the executing intermediate active nodes from a code server prior to service operation. Again this active code could be parameters, or complete function code. Clearly the capsule approach has the finest granularity (packet level), however, from an implementation and operational point of view, the capsule approach has weaknesses concerning performance and scalability. Transport or application level active networking is processed only at strategic located active network nodes, allowing the system to scale well. Recently, a new research field has been emerging, developing component-based middleware approaches that uses the discrete active networking approach in order to adapt the middleware to environment conditions and user requirements.

It should also be outlined, that it is difficult to draw a clear borderline between the discrete approach and signaling protocols. In case a new parameter set for processing streams should be uploaded and configured on intermediate nodes, a discrete PAN approach can be used, but maybe also a signaling protocol. Which of these both alternatives will be more suitable, depends on the broader context of the application scenario to be supported. If there is, for example, only the necessity to transmit a certain parameter set, which is already know in advance, a signaling protocol adapted to this kind of scenario probably will be easier to implement. However, if there are different scenarios, using different parameter sets, which could require extensions in future, the implementation of a discrete PAN approach will provide the higher flexibility for addressing these requirements.

Several multimedia scenarios look promising for being used in combination with programmable active networking. Some examples have been outlined, e.g. a transcoding scenario, an intelligent dropping scenario, a reliable multicast scenario, and a multimedia multiplexing scneario. Within a transcoding scenario one would have the possibility to install new transcoding mechanisms actively from a code server on all nodes running PAN functionality. In case of bottlenecks these transcoders could decrease the bandwidth consumed by multimedia streams and thereby adapt the multimedia service in a flexible way to the network properties. As with the appearance of new media formats also new transcoders will be required, PAN would establish an open, flexible way for their deployment. Instead of transcoding one could also decide to drop traffic for accommodating bottlenecks. However, in order to minimize the effect on the user consuming the multimedia service, this dropping should be done in an intelligent way, that is, those parts of a multimedia streams should be dropped first, which have the smallest impact on the subjective user perception of the multimedia service. Again, this intelligent dropping needs to be further developed along with the progress of multimedia formats. As a third example for the usability of PAN one can consider the deployment of reliable multicast services, where PAN could be used e.g to upload new Reliable Multicast protocol instances or just new Forward Error Correction codices onto reliable multicast nodes. And finally, in the multimedia multiplexing scenario satellite network nodes could perform routing and multiplexing of multimedia content intelligently, e.g. route specific media to specific spot beams. In summary one can state that the deployment of a PAN approach should not be done for a single application service, as its benefit might not justify the related effort, but an operator should consider the variety of service that are supported and could take benefit from the flexibility and extensibility provided by a PAN system.

Integrating PAN functionality in satellite networks for supporting next generation multimedia services could require deployment in different places, such as within the nodes of a satellite access provider, within the satellite itself, or within user nodes. The integration in all these different locations has its specific advantage and challenge. Especially the integration in the space segment should be carefully considered. On the one hand side having PAN functionality in the space segment could efficiently improve many applications. For example in case of a multi spot-beam scenario with different QoS characteristics for the various beams, an upload of new transcoding and dropping mechanisms to the space segment could be done via PAN, allowing an efficient adaptation of next generation multimedia streams to the respective QoS characteristics. On the other hand, integrating PAN functionality into the space segment and allowing this way a flexible upload of new functionality is a challenge from the performance and security point of view.

Concerning performance, one needs a mechanism to verify in advance, that a new functionality to be uploaded and executed in the space segment will obtain the required computing and memory resources. Due to the long lifetime of the space segment, these resources are often not up to date with the newest hardware development progress, and therefore might not be sufficient for any new functionality. Also it needs to be guaranteed that resources assigned in the space segment to any new active process uploaded via PAN will not decrease the performance of other main tasks of the space segment, like switching user traffic.

Concerning security one needs to implement PAN in a way that under no circumstances the space segment could be misused or turned unusable. For this purpose the first to be made sure is that only correctly authorized code will be allowed to be executed on the space segment. Furthermore this new functionality needs to undergo detailed plausibility tests before uploading in order to minimize the probability of malfunctioning. After successfully passing these plausibility tests the upload of this new functionality from e.g. a terrestrial code server to the space segment needs to be done via secured communication links in order to protect the integrity of the transmitted data. Finally there still will exist a remaining probability of having overlooked some malfunctioning during the plausibility tests. To still prevent the space segment from major damage by the execution of malfunctioning code uploaded via PAN, this code should be executed in a secure environment like a Java sandbox. This way a malfunction will only affect therespective active process but no other tasks executed by the space segment.

Finally it should be noted that while plenty of work has been done concerning PAN in terrestrial networks, less emphasis has been put so far on the use of PAN for satellite networks. Consequently the work referred to in this paper should be seen as a first step in this field, requiring additional work to follow. For example the performance and security aspects of deploying PAN in satellite networks needs to be further investigated, resulting in suitable architectures addressing the specific security and performance requirements. Also the management aspects of the PAN functionality itself as well as of the functionality deployed via PAN needs to be analyzed in more detail. As an accompanying measure it would be beneficial to install PAN functionality in an operational satellite network, in order to verify study results and to do appropriate performance measurements.

Acknowledgments

The work presented in this paper has been sponsored by the European Space Agency (ESA) project "Programmable Active Networks for Next Generation Multimedia Services". The authors thank especially Frank Zeppenfeldt and Erling Kristiansen from ESA for supporting our work.

References

[1]Psounis, K., "Active networks: Applications, Security, Safety, and Architectures", Stanford University, IEEE Communication Surveys, First Quarter of 1999

[2]Kulkarni, A. B., and Minden, G. J., "Active networking Services for Wired/Wireless Networks", University of Kansas

[3]Nguyen, H.-B.; Duda, A.: “An Active Node Architecture for Proactive Services”, LSR-IMAG Laboratory

[4]Hariri, S., Kyou Ho Lee, et al., "Survey & Classification of Programmable Network Technologies", University of Arizona and Electronics and Telecommunications Research Institute of South Korea

[5]Zachariadis, S., Mascolo, C., and Emmerich, W., "SATIN: A Component Model for Mobile Self Organisation", University Colleage London, August 2004

[6]Wetherall, D. J., "Service Introduction in an Active Network", Massachusetts Institute of Technology, Feb 1999

[7]Fry, M., and Gosh, A., "Application Level Active Networking", University of Technology, Sydney, Sep. 1998

[8]Speakman, T.: “PGM Reliable Transport Protocol Specification”, IETF RFC 3208, Dec. 2001

[9]Adamson, B., C. Bormann, C., Handley, M., and Macker, J., “NACK-Oriented Reliable Multicast Protocol (NORM)”, IETF RFC 3940, Nov. 2004

[10]Handley, M. et al., “The Reliable Multicast Design Space for Bulk Data Transfer”, IETF RFC 2887, Aug. 2000

[11]NRL Protean Research Group, "Multicast Dissemination Protocol version 2 (MDPv2)", http://mdp.pf.itd.nrl.navy.mil/

[12]Kingsbury, N., “The MPEG standard”, http://cnx.rice.edu/content/m11144/latest/

[13]Gürses, E., Akar, G. B., and Akar, N., ”Selective frame discarding for video streaming in TCP/IP networks”, Middle East University and Bilkent University, Ankara, Turkey

[14]ETSI TS 102 812 V1.2.1, “Digital Video Broadcasting (DVB);Multimedia Home Platform (MHP) Specification 1.1.1”, ETSI Technical Specification, June 2003

[15]Law, C., and Furht, B., “The MPEG-4 Standard for Internet-based multimedia applications”, Jan. 2001

[16]MPEG-4 Industry Forum “MPEG-4 – The Media Standard”, Nov. 2002

[17]“ETSI: Final Draft EN 302 307 (v.1.1.1) Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2).” http://www.etsi.org, January 2005.

[18]M. Boliek, C. Christopoulos, E. Majani (editors). JPEG2000 Part I Final Draft International Standard. (ISO/IEC FDIS15444-1), ISO/IEC JTC1/SC29/WG1 N1855, August 2000.

[19]Grace, P., Coulson, G., Blair, B., Mathy, L., Yeung, W., Cai, W., Duce, D., and Cooper, C., "GRIDKIT: Pluggable Overlay Networks for GridComputing", Lancaster University and Oxford Brookes University, UK, June 2004

[20]Alcatel Espacio, "Alcatel 9343 DVB On-Board Processor", http://www.alcatel.es/espacio/dvb_e.jhtml

[21]Cornfield, P. S., "Broadband Digital Processor Architectures and Implementation for Advanced Regenerative Communications Satellites", Astrium Ltd., 2001