

On the Manipulation of JPEG2000, In-Flight, Using Active Components on Next Generation Satellites

L. Sacks¹, H.K. Sellappan¹, S. Zachariadis², S. Bhatti², P. Kirstein²,
W. Fritsche³, G. Gessler³, and K. Mayer³

¹ Department of Electronic & Electrical Engineering,
University College London, London, UK

² Department of Computer Science University College London, London, UK
³ IABG mbH, Ottobrunn, Germany

1 Introduction

This paper describes two approaches to manipulating JPEG2000 frames with programmable and active networks. The first approach is the use of transcoding and the second is intelligent dropping. These two approaches were considered, in particular, for possible deployment with space based platforms; specifically, communication satellites which are not only IP enabled but may host active components. Each approach offers different possibilities and may be suitable for solving overlapping but different problems.

The work presented here brings together a number of background technical developments from the communications satellite world, video-coding and intelligent programming models. A detailed look at the developments of satellite based communication platforms shows that there is the possibility of a fully IP enabled system, supporting multicast and quality of service in space. This not only opens a range of possibilities, but presents new challenges. Further, emerging coding schemes open up new possibilities for manipulation of content within the networks. JPEG2000 was used as an example of the next generation of scalable codecs and it has been found that it lends itself easily to helping the kind of problems considered here, although much of these developments can be applied to other coding schemes.

The two scenarios considered in detail in this paper – intelligent dropping and transcoding – show two approaches to coping with varying available bandwidth, as will be available with DVB-S2. They also illustrate two approaches to programmable networks. Intelligent dropping is best performed by dedicated systems where queuing is performed (for example routers in the input to link modulators) and so is best managed through Policies. Transcoding, in contrast, is codec specific and needs to be performed with a procedural language; it is thus an excellent example of an application level active networking technology. The possibility of network level active networking is not demonstrated by these, but is considered as a general issue in this context.

The work presented here was initiated and funded by the European Space Agency (ESA), European Space Research and Technology Centre (ESTEC) ARTES 1 program. The complete project report will be made available in due time. The brief was to undertake a “study addressing the use of Active and Programmable Networking in Space and Ground Segment, to improve the delivery of Multimedia Services over Satellite”. The project covered a wide range of issues including architectural issues,

candidate technologies, standards, performance and security. It also included two demonstrations. The work presented here reflects this project with an emphasis on the demonstrations developed (it should be noted that the intelligent dropping scenario discussed in this paper was developed with Mr. Sellappan, as part of his MSc, and is not identical with that used in the study). Section 2 discusses the architectural issues both from the high level and business model perspective and considering some details of the space segment. Section 3 reviews some technologies used in this study. In particular it looks at the evolution of satellite platforms for broadcast media, JPEG20000 and SATIN, the mobile code technology used here. Section 4 reviews a number of applications for active networking in this context and outlines in more detail the two target scenarios of intelligent dropping and transcending.

2 Architectural Issues

To understand the application of active networking to satellite telecommunications it is important to develop a high level architecture and business model, describing both ways in which the technologies may be applied and the roles of organisations and players involved. It is also important to consider the detailed architecture of the space platform, as this is one of the least versatile components and has the most stringent performance requirements.

2.1 The High Level Architecture

The high level architecture and context of the technologies discussed here is not well defined; but is open to a number of scenarios, depending on business and application needs. To understand the overall concept, we define the basic service scenario in which some media (still images or a video stream) is sent from a single source to one or several users; and transits a satellite link. There may or may not be wide area or local area networks on the user side as the users may, for example, be within an organisation owning a satellite receive, they may be using a VSAT device or the satellite link may be providing a distribution system between two internet service providers.

In our context we define the following roles (which, of course, may be taken by one or several players depending on the details of the business model). The Media Provider owns the content which may be in any format. Optionally, the media provider may participate 'actively' in the service by having a server which can encode the content in JPEG2000 format. As an alternative, the server may provide content in other formats which are encoded on the fly at the satellite link ingress. For the intelligent dropping scenario, the ingress point has to be involved in order to prioritise packets and add information to the data stream to support this. In either case, the satellite is considered there to be an active component in which specific flows can be directed to programmable components for treatment (e.g. transcoding or intelligent dropping). Finally, the stream may either be re-coded to its original format at an egress active component, or may be sent directly to a terminal device which can either directly decode the content through resident or downloaded components. The final role in this high level model is the Active Components Provider which hosts the active components to be deployed in each appropriate node as required. In practice this role may be played by a third party, the content provider or the satellite service provider.

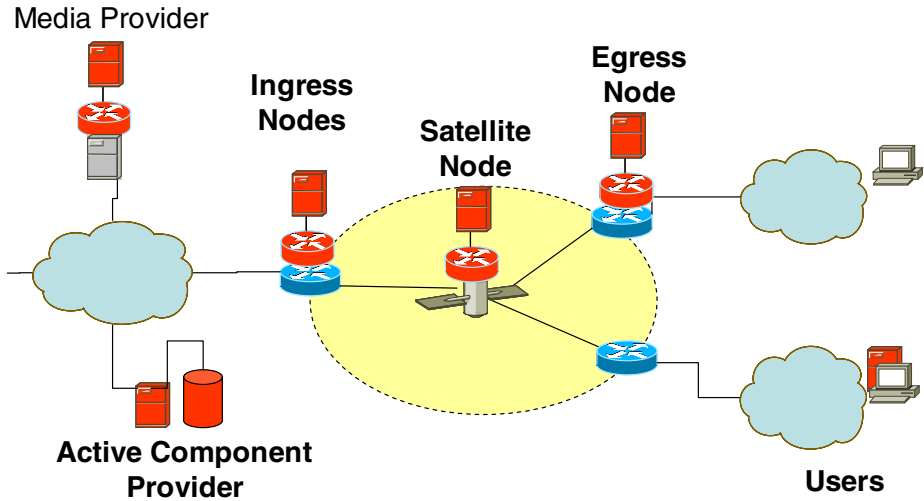


Fig. 1. High level Architecture

2.2 The Satellite Platform Architecture

For this study our primary focus was on the functional description of the satellite platform. For this we defined a detailed architecture as a starting point for developing a standard and the appropriate functional requirements. This architecture was developed so as to define the full context for active services on a satellite platform. However we did not define control and management concepts for this. The Active components are expected to be hosted in a virtual machine run time environment such as the JavaVM. Within this context we defined four types of component. The Dynamic Components are those which constitute the active service and are deployed on demand. To support these, we defined two sets of components which interact *via* the same semantics and interfaces as the dynamic components; the Interface and Resident components. The Interface components allow the service to interact with the infrastructure of the satellite and the Resident components allow the service to receive and send the stream data. The final class of components in the execution environment are those which manage the location and deployment of the active components themselves.

Around the execution environment is the transport system and the platform infrastructure. The transport system comprises the DVB and MPEG (multiplex) capabilities; and some sense of switching or routing capability. The transport system has to be capable of re-directing appropriate flows to the active components as required and so would integrate with an active routing technology (which was not explored in this study). Further, the transport system can be controlled through policies from the active components. The control through policies represents a second, widely accepted, view of active networking and thus our system facilitates both application level active networking and policy based programmability. Both of these are useful and have been explored in this study. The infrastructure of the satellite platform is accessed to retrieve and, possibly, manipulate key capabilities. For example the action of an active

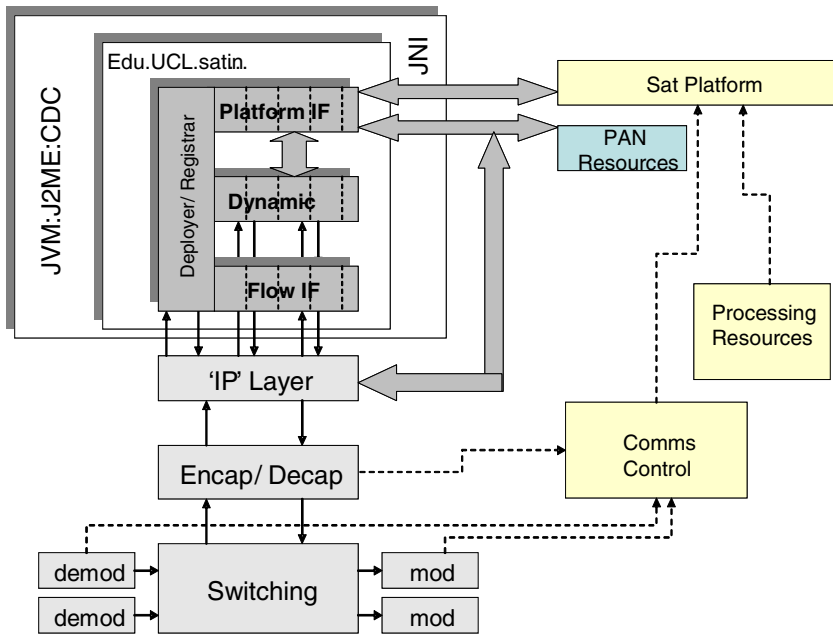


Fig. 2. Satellite Active Architecture

component may need to know the quality of the down link (discussed in detail below), the state of IP queues in the switch/router or the current loading on the execution environment resources (CPU and Memory).

The function of the resident components is to provide consistent interfaces to the dynamic components, which preserving the security and integrity of the satellites systems. Although components to be deployed may be checked 'off line' and will be certified, it is still important that the platform is capable of protecting its self. Thus the Interface components would not just provide APIs, but should restrict and schedule access to the platform; and should monitor for integrity issues such as deadlock and livelock situations.

3 Technologies

This section discusses the technologies used to demonstrate transcoding and intelligent dropping with JPEG2000. The overall project reviewed a range of possible technologies, both for media transport and programmable network implementation. The SATIN mobile code platform was selected as an interesting example of a framework for reasons discussed below. JPEG2000 was selected as an interesting scalable codec, compatible with a wide range of future applications.

3.1 The Programmable Component System: SATIN

We used the SATIN [1] platform to implement our active network system. SATIN is a component metamodel, which can be used to build adaptable systems. It is instantiated

as a middleware system, which can adapt itself or component based applications which run on top of it. Its use as an active networking platform was initially outlined as a case study in [2]. SATIN uses the principles of reflection and logical mobility to offer adaptation primitives, and offers them as first class citizens to applications. Reflection offers a meta-interface to the system, that allows SATIN applications discover which components are available locally, to be notified of changes in component availability, to load and discard components, to discover the interfaces that they offer and to read and write metadata attached to a component. Logical mobility allows SATIN systems to send and receive components, classes, objects and data. SATIN was designed for mobile devices and devices with limited resources, such as mobile phones and PDAs, and is implemented using Java 2 Micro Edition (Connected Device Configuration / Personal Profile). It occupies 150329 bytes of memory, and provides the following services which are relevant to an active networking system:

- **Reflection.** The use of reflection allows the system to reason about what it can currently do, dynamically invoke components, use an event service which notifies when there are changes in component availability, and to discover new APIs that are offered by components.
- **Logical Mobility.** The use of logical mobility allows components to be dynamically transferred from one SATIN node to another.
- **Dynamic Component Instantiation.** Components can be dynamically discovered, instantiated and invoked.
- **Advertising and Discovery.** SATIN provides an adaptable framework for component advertising and discovery.
- **Security.** SATIN nodes can employ digital signatures and trust mechanisms to verify incoming components. Moreover, SATIN uses the Java sandbox to instantiate components in.

In conclusion, SATIN is a small footprint middleware system that can be used to build adaptable systems. It provides a minimal runtime API that can be used to discover, request, deploy and invoke new components and an interface to reason about what individual components or the system itself can do.

3.2 The JPEG2000 Codec

The final component used in this study was the JPEG2000 scalable codec. The fast expansion of multimedia and Internet applications led to the development of a new still image compression standard, JPEG2000 [3]. This image coding system uses latest compression techniques based on wavelet technology. Designed to complement the older JPEG standard, JPEG2000 provides low bit-rate operation, error resilience and superior compression index. Some of the important features of JPEG2000 are lossless and lossy compression, progressive transmission by resolution or component and region-of-interest (ROI) coding.

In JPEG2000, images are typically divided into multiple tiles and encoded independently. This is done to avoid the need for complex and powerful processors when loading and encoding huge images in hardware. The Discrete Wavelet Transformation (DWT) is designed for this purpose. A tile-component is referred to a tile which consists of only one colour component. Each tile-component is then further divided down to different

resolutions and sub-bands with the use of DWT. Each resolution is then divided into multiple precincts which identify a geometric position of a tile-component of an image. Furthermore, each sub-band at each resolution is divided into multiple code-blocks, which will be coded into individual packets. The packets are arranged in the codestream following a particular progression order specified at the encoder.

Motion JPEG2000 (MJ2) has been defined in part III of the JPEG2000 standard [5] for compression and encoding of time sequences of images, like video streams. The standard was developed to generate highly scalable compressed video, which can be easily edited. Thus, MJ2 does not include motion compensation and every frame in the video stream is individually encoded [6]. The concept of intra-frame coding reduces the complexity of inter-frame interdependencies found in other video codec. Since each frame is treated like a still image, a single codec can be used for both JPEG2000 still pictures and MJ2 video compression and encoding.

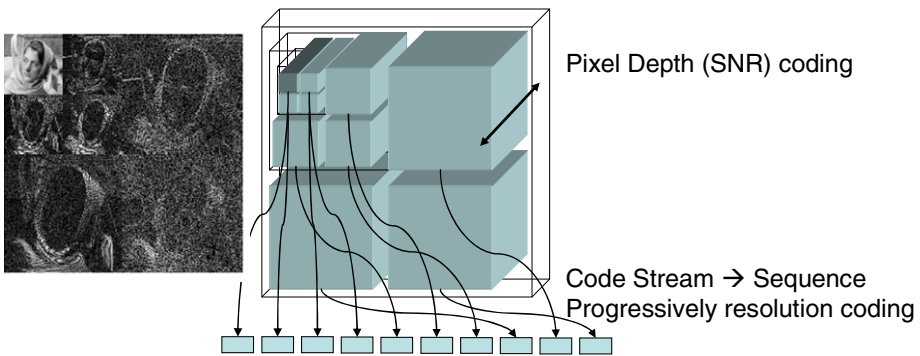


Fig. 3. JPEG2000 Coding by Resolution and SNR [4]

A key feature of JPEG2000 is that it supports two forms of scalability. The first SNR scaling can progressively reduce the amount of information per pixel. This may be performed in a number of ways, on the various coding layers. The affect of SNR scaling is shown in Figure 4. The second form of scaling is progressive resolution coding. This effectively changes then size of the transmitted frame. Illustrated in Figure 3 is the organisation of the JPEG2000 code stream. Four levels of resolution coding are shown. The dark square is a low resolution image, when combined with the neighbouring three code blocks; a higher resolution image is formed. This is then repeated with the next two sets of three code blocks. At the same time the number of bits per pixel (bpp) represents the SNR coding depth.

3.3 Satellite Transport

At present, the trend of satellite communication is heading towards deployment of next generation broadband satellites that provides multimedia applications with high demands for quality and availability of service. In line with the anticipated advancement and to meet the growing demand for high data rates, much research has been focused on regenerative and full on board processing (OBP) payloads, advance mesh

antennae, high speed communication system, optical satellite links, high speed transponders and miniaturisation of satellite components [7, 8]. The first regenerative satellite payloads are operational or under construction, performing on-board MPEG-2 cell multiplexing and switching (*e.g.* Skyplex or AmerHis) or ATM cell switching (*e.g.* WEST and WEB 2000 architecture), and the UK-DMC satellite even implements an IP stack. It is foreseeable that future communication satellites will implement an IP stack and on top of this a programmable network platform.

Future generations of communication satellites may support a range of new capabilities. The most immediate is the emergence of DVB-S2[9], an effective upgrade of the current Digital Video Broadcast over Satellite (DVB-S) transport technology, which is the dominant means for transporting most broadcast video to date. DVB-S2 includes adaptive Forward Error Correction (FEC) which means that the up and down link data capacities can adapt to changes in fading, for example from varying cloud cover. This also means, however, that both the up and down link transport capacities may independently vary; increasing with weaker FEC and contracting as stronger FEC is required. Each FEC frame has fixed size of either 64,800 bits for normal FEC frame or 16,200 bits for short FEC frame. Depending on the FEC code rates, the capacity of the frames may be anything between these limits. Further, work is already progressing on IP enabled capabilities for satellites. Although this may not be an obvious thing by its self, combined with developments in tuneable, multiple footprint platforms, we see a model of a satellite which not only may have to do routing, but multicast and some amount of Quality of Service management. Standards for DVB over IP are already in progress – and so we see the emergence of true IP based broadcast architecture.

4 Active Services in Satellites

A wide range of applications scenarios can be considered for application in the architecture described above. The following is a list of applications considered in some detail in the study.

- **Reliable Multicast (RM) scenario:** In the RM scenario, active networking may be used to load RM protocol instances and FEC codecs to relevant network nodes on demand while runtime.
- **Intelligent Dropping (ID) scenario:** In the ID scenario, intelligent dropping processes (or just policies) that evaluated the priority of packets or streams are loaded dynamically via PAN to satellite nodes that have to drop packets.
- **Transcoding (TC) scenario:** In the TC scenario, *via* active networking transcoders are controlled and loaded dynamically by satellite network nodes in order to adapt parameters (*e.g.* codec, data rate, etc.) of multimedia streams to network conditions or user requirements.
- **MHP scenario:** In the Multimedia Home Platform (MHP) scenario an intelligent process in the satellite routes certain MHP content only to specific spot beams and performs user feedback aggregation. *via* active networking, the intelligent process could be loaded by the satellite and adapted to new applications.
- **MPEG-4 scenario:** In the MPEG-4 multiplexing scenario content from different sources composing a single MPEG-4 scene are multiplexed in the satellite. *via*

active networking, the composition software can be loaded by the satellite and controlled from the base station.

- Advanced network management: In the advanced network management scenario, management or monitoring software is adapted on demand to operator needs *via* a active networking system.
- Advanced caching scenario: Caching software is loaded, updated, and controlled on demand *via* a active networking system in order to adapt the caching software to applications and services.

These scenarios represent a broad range of; applications, network and policy level programmable network problems. The scenarios described below focus on application level and policy level approaches. Both cases are to do with manipulating JPEG2000 images. Both where implemented on PC based test-beds and used the Java JJ2000 implementation [10] available from EPFL. At this time, there is no reference satellite platform defined for developing and testing the kind of applications discussed here; and one of the outputs from the Arties 1 project was the recommendation that such a platform be developed.

4.1 The Transcoding Scenario

The issue of transcoding has been widely explored (see for example [11]). This transcoding scenario considers a stream of JPEG2000 images, possibly forming a moving image and which requires that a constant frame rate is preserved (or considers cases where the download latency is just to slow for ‘full fat’ images). Issues impeding this can arise in a satellite situation when, for example, the down-link FEC reduces the available link bandwidth. It may also occur if there is routing in the satellite platform, which may have multiple foot prints. There are then two options; reduce the resolution or change the information depth (SNR) of the picture as illustrated in Figure 4. The former of these may be preferred if, for example, the terminal devices has a small screen.



Fig. 4. SNR Progressive coding (3.55bpp, .355bpp, .0355bpp and -0.009bpp)

To implement this application the following functions are performed, with regard to the reference architecture (shown in Figure 1):

- The Media Provider may re-code the images in JPEG2000 format. This may, optionally, be performed on the Ingress Node.
- The Satellite Node hosts the active components, loaded from the Active Component Provider.

- If the stream is encoded at the Ingress Node, it should be decoded back to the originating format at the Egress Node. Otherwise the User should have the appropriate decoder.

On the satellite node, the active component is loaded as a Dynamic Component. It should then have access to the link capacity from the modulator (Figure 2) to discover the available down link capacity and decide how to transcode the image. It should, of course, also know the rate at which the images are being sent for the target stream and the overall downlink load.

Implementation of the transcoding component may be approached in a number of ways. Within the scope of the ARTIS project, an expedient approach was taken and the encoder and decoder provided with [10] were used. However, the code streams of JPEG2000 are so designed that significant efficiency gains may be had. The resolution transcoding is the most straightforward as can be seen from Figure 3, this requires simply that the code stream is truncated at the right place. In this respect, it is similar to the intelligent dropping scenario below. SNR progressive scaling is more complex and detailed information about each tile is required. Nevertheless, these are all integer operations and so can be quite efficient from the performance perspective.

4.2 The Intelligent Dropping Scenario

As with the transcoding scenario, intelligent dropping may be appropriate when constant frame rates are required in the presence of congestion or reduced link capacity. However it may be that there are queuing mechanisms involved which by default drop packets randomly – where by randomly we mean, with out regard to the contents of the packets. The impact of this can be seen in Figure 5 in which the first image is the reference graphic and the second has 1% of packets lost at random. Of course, it is possible that dropping packets has little impact and the figure here shows a bad case, although not unusual.

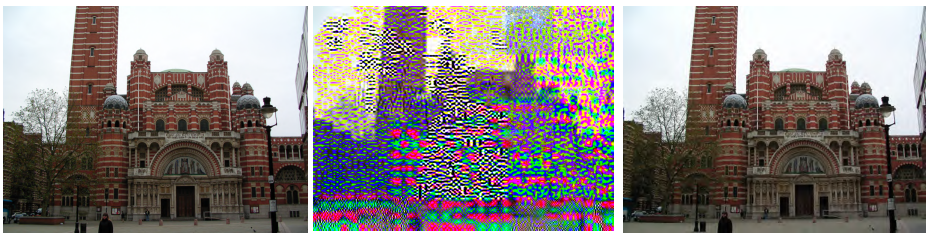


Fig. 5. Intelligent Dropping (0% loss, 1% Random loss, 80% intelligent dropping)

The effect seen, can be understood with reference to Figure 3. It can be seen that if a packet containing the lowest resolution ‘thumbprint’ of the coded image is lost, recovering the graphic will be almost impossible and the impact becomes progressively less severe as the outer code blocks are lost. Thus, prioritising loss progressively, following the flow of the codec would allow a more graceful information loss; and allowing for this was part of the initial intent of the JPEG2000 technology. The last image in Figure 5 has 80% of its data dropped, but using low priority packets.

Architecturally, the implementation of intelligent dropping is similar to the transcoding scenario. The two key changes are the introduction of an intelligent, adaptive encapsulator at the satellite ingress point and the control of the dropping mechanism in the space segment. These are shown in Figure 5.

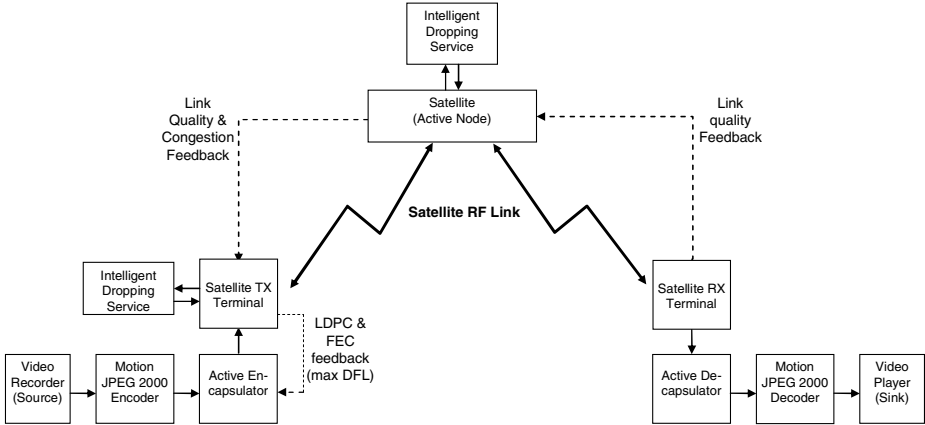


Fig. 6. The Intelligent Dropping Architecture

In this architecture [12], the active encapsulator has two jobs. The first is to decide, depending on things like link quality, how to place the JPEG2000 code stream in the IP packets. There is an important trade off here as, on the one hand, the finer the granularity this is done at the more precisely the available band width can be matched; but the overheads are increased. Too coarse a granularity and the quality of the image will be degraded too quickly. The second job is to add a header which describes the priority of the packets as well as which is the; first, last and sequence number of the packets of a given video frame. The ingress point can drop packets in response to the uplink quality as can the satellite platform its self. On the satellite platform, the condition of packet dropping can be controlled with policies transported via any policy management framework. Finally, the satellite egress point has to extract and reassemble the code stream.

5 Conclusions

This paper has reviewed the issues which arise when considering the role of programmable and active networking in the context of satellite based telecommunications. We have motivated the plausibility of such an architecture by considering the evolutionary path of telecommunications satellites. We have also tried to motivate the advantages of considering placing active components on the satellites themselves. Indeed, throughout our project trying to understand where the business advantage come has been more difficult then considering its technical viability. However, if we consider the combination of factors such as the adaptive FEC, multiple, tuneable footprints with onboard routing or switching, the importance of on-board processing

becomes more apparent. Finally, it is important to note that the timescales of satellite deployment are long compared to those in the fixed network world, so it is very difficult to know which application level protocols will be used once the satellites are commissioned. Thus, if it is important to have intelligence in the satellite at all, it is important that that intelligence is adaptable both to protocols and codecs which might emerge in the future; and can adapt to local conditions (such as link capacity variation and congestion).

To emphasise these issues of adaptation we have both used a codec which is very much in development and have explored how adapting to local conditions can be used to improve application performance in the face of degraded transport (down link) quality.

References

- [1] Zachariadis, S., Mascolo, C., Emmerich, W.: SATIN: A Component Model for Mobile Self- Organisation. In: International Symposium on Distributed Objects and Applications (DOA), Agia Napa, Cyprus. Springer, Heidelberg (2004)
- [2] Zachariadis, S., Mascolo, C., Emmerich, W.: Exploiting Logical Mobility in Mobile Computing Middleware. In: Proceedings of the IEEE International Workshop on Mobile Teamwork Support, Collocated with ICDCS 2002, pp. 385–386 (July 2002)
- [3] Boliek, M., Christopoulos, C., Majani, E. (eds.): JPEG 2000 Part I Final Draft International Standard (ISO/IEC FDIS15444-1), ISO/IEC JTC1/SC29/WG1 N1855 (August 2000)
- [4] Marcellin, M., Gormish, M., Bilgin, A., Boliek, M.: An Overview of JPEG 2000. In: Proceedings of IEEE Data Compression Conference, Snowbird, Utah (March 2000)
- [5] Information technology - JPEG 2000 image coding system - Part 3: Motion JPEG 2000, ISO/IEC 15444-3 (2002)
- [6] Dagher, J., Bilgin, A., Marcellin, M.: Resource-Constrained Rate Control for Motion JPEG 2000. *IEEE Transactions on Image Processing* 12(12), 1522–1529 (2003)
- [7] Iida, T., Suzuki, Y.: Communications satellite R&D for next 30 years. *Space Communications* 17, 271–277 (2001)
- [8] Verma, S., Wiswell, E.: Next Generation Broadband Satellite Communication Systems. In: 20th AIAA International Communication Satellite Systems Conference and Exhibit, Montreal, Quebec, May 12–15 (2002)
- [9] 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) (January 2005), <http://www.etsi.org>
- [10] <http://jj2000.epfl.ch/>
- [11] Gibson, J.D.: *Multimedia Communications: Directions and Innovations*. Academic Press, London (2000)
- [12] Sellappan, H.K.: *Active Networks in Satellite Communications: Intelligent Dropping Scenario for Motion JPEG 2000 Transmission* MSc. Dissertation, UCL 2005 (2005)