Supporting Network Transparency in 4G Networks

Gianmarco Panza, Enrico Balatti, Gianfabio Vavassori, Catherine Lamy-Bergot, Filippo Sidoti

***Abstract*— 4th generation of wireless systems (4G) is foreseen as a globally integrated communication network interconnecting, in a transparent way, a multitude of heterogeneous networks and systems. Optimal allocation of user and system resources may be effectively achieved with the co-operative optimisation of communication system components. Innovative schemes enabling joint optimisation over wireless links include the development of flexible channel coding and modulation schemes, the adaptation of existing source coding schemes with respect to their ability for Joint Source Channel Coding/Decoding (JSCC/D) and the specific development of new optimised ones.**

In this paper we tackle the issue of transmitting joint optimisation control and signalling information through wired and wireless networks using cross-layer design and network transparent communication methods as addressed by the PHOENIX IST project.

A theoretical analysis is provided while simulation results show a performance evaluation of the mechanisms that could implement the concept of the Network Transparency and therefore highlight, for each specific control or signalling information to be transferred, the mechanism that could be most suitable.

I. INTRODUCTION

Following the path opened by GSM systems, the under-deployment UMTS system is leading to more and more configurable, dependable, adaptable, intelligent, secure but also complex wireless solutions. Aiming at handling digital data of different nature (text, voice, image, video...) that will be used in various contexts (home, office, on the move,...) these systems rely on inner software that make them more and more efficient and easy to use. Systems are targeting flexible re-configurable architectures and the 4th generation of wireless systems (4G) is foreseen as a globally integrated communication network interconnecting, in a transparent way, a multitude of heterogeneous networks. Optimal allocation of users and system resources may be effectively achieved with the co-operative optimisation of communication system components. This approach, following the already known joint source channel coding and decoding (JSCC/D) [1] one, aims at developing strategies where the source coding, ciphering, channel coding, modulation, and, possibly, network parameters are jointly determined to yield the best end-to-end system performance. These strategies are currently under study by the PHOENIX project [2].

PHOENIX is an FP6 IST European project started at the beginning of 2004 and will be finished at the end of 2006. It is collaboration between industrial partners, industrial research laboratories, Small Medium Enterprise and specialised academic institutions of different countries.

G. Panza, E. Balatti, G. Vavassori are with CEFRIEL Network Systems Unit, Milan, Italy
C. Lamy-Bergot is with THALES Communications, Colombes, France
F. Sidoti is with WIND Telecomunicazioni S.p.A., Rome, Italy

The aim of the PHOENIX project is to develop a scheme offering the possibility to let the application world (source coding, ciphering) and the transmission world (channel coding, modulation) to talk to each other over an IPv6 protocol stack (network world), so that they can jointly develop an end-to-end optimised wireless communication link. To reach this goal, the following main axes will be pursued:

- development of innovative schemes to enable end-to-end joint optimisation over wireless links: flexible channel coding and modulation schemes, adaptation and development of source coding schemes with respect to their ability for JSCC/D, Quality of Service (QoS) and bandwidth optimisation.
- establishment of efficient and adaptive optimisation strategies jointly controlling the coding blocks
- building of a global network architecture based on joint optimisation for future wireless systems. This objective includes the development of the *transparent network communication* approach

PHOENIX approach relies on the overall architecture presented in the figure below (Fig. 1):

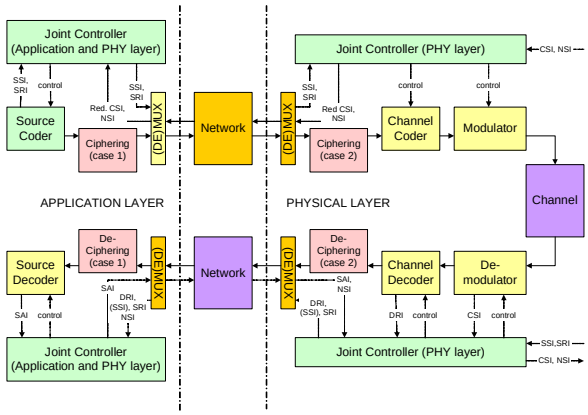

Fig. 1. End-to-end communication system over an IP based network

Constituted by the key elements of the architecture to be deployed to develop suitable optimisation strategies for achieving adaptive/dynamic optimisation for multimedia information transmission over an IP-based wireless link, this new architecture presents in particular co-ordinating tools, named *joint controllers*, which implement the necessary controlling strategies. Their task is to drive the whole communication chain by providing selective protection of the transmitted data, by co-ordinating the scalable source encoder, the adaptive channel encoder and the dynamic modulator.

The integrated framework provided by PHOENIX study aims at helping to offer the end user a versatile and adaptable secure infrastructure. This structure is meant to provide more bandwidth efficient transmissions, typically targeting about 3 dB gain in terms of useful signal-to-noise ratio, or conversely about 50% gain in terms of bandwidth. Another aspect is also the possibility to offer the end user differentiated services, based on different classes of services thanks to specific joint controlling of the transmitted data.

In the rest of the paper we give an overview of the Network Transparency concept (section II) and the different mechanism to implement it (section III). Section IV explains the PHOENIX signal and control information to be transferred between relevant entities while section V shows simulation results. Finally, we give conclusions and state the future work.

II. NETWORK TRANSPARENCY

Network Transparency is a fundamental aspect that allows an effective realisation of the PHOENIX JSCC/D system. It is somehow an abstract idea of making the underlying network infrastructure almost invisible (from which the transparency) to all the entities involved in the jointly optimisation of the source and channel (de)coder, as well as of the (de)modulator. Almost transparent is related to the fact that the telecommunication infrastructure by its own, inevitably affects in some extent the overall system, such as introducing delay, loss and various types of errors, but without actually interacting with the control-plane of the concerned deployed devices and providing sufficient delivery guarantees to the video streams in order to accomplish a defined quality of service (QoS) for the end-user.

The goal is twofold:

- to realise communication exchanges between entities differently located in the network (including the end-terminals)
- not to interact anyhow with non-JSCC/D aware devices.

The primary goal is referred to the capability of transferring signalling/control data between both different network nodes and link layers as needed, in a transparent manner, in spite of the strict rules of the ISO OSI model [3], which impose a modular and independent design of each link layer of a network node with well defined interfaces and the delivering through a telecommunication infrastructure that carries data only of a specific format (IP datagram). The second objective aims to ensure as much as possible backward compatibility, not only with the existing standards as also addressed by the first goal, but even with the nowadays telecommunication infrastructure that constitutes the basis for the next generation networks, allowing for a smooth migration to IPv6-enabled devices eventually supporting JSCC/D functionality.

The design of the solution for the Network Transparency must take into account the deployable security mechanisms, providing authentication and encryption features, even if working at different layers simultaneously, e.g. at both application and data-link layers, as well as compression techniques. Security and header compression facilities can introduce further constraints, not only in terms of additional complexity and hence delay, but also of theoretical and practical feasibility of the available mechanisms to support the Network Transparency. Some mechanisms that could implement the concept of the Network Transparency are: IPv6 data packets and extension headers, ICMPv6 messages, direct socket-to-socket communication, external databases and service profiles stored in shared memory spaces. Other possible methods relies either on the introduction of adaptation layers at the transmitter and receiver side to allow for the exchanges implicated by the joint source and channel (de)coding system, or the exploitation of already existing and deployed ad-hoc signalling protocols.

III. MECHANISMS TO IMPLEMENT THE NETWORK TRANSPARENCY

A. IP Packets

The IPv6 packets can transport a payload of a maximum allowable size, depending on the maximum transmission unit (MTU) of the relaying telecommunication infrastructure. This is typically a transport service data unit (SDU) containing application information. If the application data generated is bigger than the MTU, it is fragmented at the sender side by the network layer process and transmitted in more IP packets delivered completely independently. However, the SDU can be created by grouping signalling or control information.

B. IPv6 Extension Headers

IPv6 [4] has a mandatory header (IPv6 base header) and some optional extension headers. There are currently six optional headers available, among which two can be used for exchanging control/signalling data of the PHOENIX JSCC/D proposal: Hop-by-Hop Options and Destination Options.

The Hop-by-Hop Option header is used to carry optional information that is examined by every node along a packet's delivery path. Instead, the Destination Options header is used to carry optional information that needs to be examined only by a packet's destination node.

C. ICMPv6

ICMPv6 [5] is a simple protocol that relies directly on IPv6. It is used by IPv6 nodes to report errors encountered in processing packets, and to perform other Internet functions. ICMPv6 is an integral part of IPv6 and must be fully implemented by every IPv6 node. This requirement is very important for its applicability into the PHOENIX framework for transferring signalling data in a backward compatible way.

ICMPv6 messages are grouped into two classes: error messages and informational messages. Error messages are used for a one-way communication, usually for error status notification, while query/reply messages are used for asking and retrieving information.

In PHOENIX project we can employ ICMPv6 to transfer control/signalling information between the entities involved in the JSCC/D chain by defining new message types and/or message codes.

D. Ad-hoc Signalling Protocols

This approach is based on the employment of an already existing control or signalling protocols designed to optimally and effectively transport either the specific information needed by a JSCC/D entity or conceived from the beginning to carry information of generic type. In the latter case, just the structure and the transport features of the protocol are strictly defined, while the delivered upper layer data content can be defined anyhow (a level of adaptation may be needed though).

A meaningful example of such an option that could be even employed into the PHOENIX framework is the Real-time Transport Protocol/Real-time Transport Control Protocol RTP/RTCP [6] that actually includes two strictly coupled protocols that provide transport for both data (RTP) and related signalling information (RTCP) of a multimedia session.

E. Direct socket-by-socket communications

Direct socket-by-socket communications are referred to end-to-end communications, in which the operating system level sockets are opened for data communication for some specific protocol and for some specific use. For example, when using the TCP/IP protocol stack, the system level socket can be reserved and opened e.g. for TCP [7] and UDP [8] end-to-end protocols and data transmission is performed through the reserved socket using the protocol in question. As one optional solution for delivering some of the JSCC/D signalling information, the direct socket-by-socket communication can be considered. This implies that the JSCC/D signalling information delivery is performed with end-to-end protocol level by opening additional protocol sockets for JSCC/D signalling. The solution offered by direct socket-to-socket communication does not require any modification of the Internet protocols or definition of new options. It also allows working totally on application layer.

IV. SIGNALS AND CONTROL INFORMATION

In the PHOENIX proposal, there are various signalling and control information that need to cross the network from the generation point to the target destination(s), mainly:

- Source significant information (SSI)
- Channel state information (CSI)
- Decision reliability information (DRI)
- Source a-priori information (SRI)
- Source a-posteriori information (SAI)
- Network state information (NSI)

Each of these has its own nature, characteristics, frequency, size, path to traverse, etc., therefore one or more of the mechanisms that can implement the concept of the Network Transparency could be the most suitable for each of them.

Hereafter, a brief description of each mentioned control and signalling information is provided.

1) SSI: The SSI is generated by the source coder and it represents the information on the sensitivity of source bits to channel errors. The SSI is strictly related to the data stream and needs to be synchronised with it. For this reason, the communication mechanisms that are more suitable to be deployed for the transmission of SSI are: encapsulation inside the IP video packet header, or an ad-hoc signalling protocol. With encapsulation inside the video packet header, SSI is automatically coupled with the data fragment it refers to. More reasonably, this employs the Hop-by-Hop options, because SSI must be available for different entities of the JSCC/D chain before reaching the destination terminal(s). However, an encapsulation into packet payload could also be appropriate. On the other hand, for an ad-hoc signalling protocol, it is necessary to deal with synchronisation to the associated video data stream, which could result in a considerable effort.

2) CSI: The CSI represents the actual conditions of each wireless channel, through which the media stream is directed. CSI is generated by the radio receiver node and effectively exploited by all the JSCC/D protocol levels on the transmitting side, at the radio interface and at the source coder. The CSI signal travels in the reverse direction with respect to the video data packets; hence it is not strictly synchronised with them. Because the CSI frequency should be much lower than the packet rate, it is considered almost negligible in terms of additional overhead. The communication mechanisms that are more suitable to be deployed for the transmission of CSI are: encapsulation inside IP video packet payloads, or ICMP messages. In both cases, an end-to-end communication can be realised as the concatenation of multiple connections between JSCC/D-aware network nodes.

3) DRI: The DRI information provides further elements related to the channel decoding process. The DRI is generated by each radio receiver and it is collected by the destination terminals in order to better tune the source decoding process and hence improve the resulting QoS. DRI must be strictly synchronised with the video stream. Considering this synchronisation requirement, the most natural solution to carry DRI could be the encapsulation in IP extension headers (Destination option, in particular), but it is necessary to cope with fragmentation, which is very likely to happen. For this reason, it is better to use a dedicated IP packet flow. In principle, such a flow could consist of ICMP messages or upper layer PDUs transporting a service data unit containing DRI. A technique to synchronise data with corresponding DRI is to exploit either sequence numbers or time stamps contained in the RTP headers of the video datagrams.

4) SRI: The SRI is additional information produced by the source coder that is exploited at the destination side and possibly also by the other entities concerned in the JSCC/D chain along the data path at the radio transmitter nodes, in order to optimise the QoS resulting from the decoding process of the video stream. The SRI is synchronised with the associated video stream and is generated by and targeted to the same JSCC/D devices as SSI. However, the amount of delivered SRI information is lower than SSI data; therefore, ICMP messages appear more attractive, but for synchronisation requirements with the video data (actually, not so stringent), and the delivery to all the JSCC/D entities that need to read the SRI. The first issue can be solved as explained for DRI, while the latter for example, by configuring appropriate filters along the communication path as needed. A dedicated IP datagram flow (direct socket-by-socket communication) could be employed, but the relative amount of overhead introduced could not be negligible. The encapsulation into an IP option could also be feasible and would solve the synchronisation problem.

5) SAI: The SAI results from the analysis of the decoding process of the video stream. It is generated by the destination terminal and exploited at the radio receiver to set the working parameters of the channel decoder and demodulator module. In a further design step it can be exploited also at the transmitting terminal in order to improve the performance of the channel coding and modulation and the resulting QoS. The SAI is not strictly synchronised with the video data packets. However, the relation between SAI and the concerned video fragment must be enforced. If the frequency and size of SAI are of the same order or even higher than the video packets, a new IP datagram flow, addressed to the source terminal, should be the most appropriate solution. Otherwise, ICMP is suggested, because it introduces less overhead and the needed control messages are issued within a reasonable interval. The encapsulation in IP extension headers, such as the Hop-by-Hop options, could

be foreseen only when a video stream on the reverse path exists.

6) NSI: NSI reports about the availability of network resources across the data path. Such information can be represented by QoS performance parameters. Therefore, NSI can be effectively exploited at the source coder to better tune the amount of the generated rate and coding parameters in general, as well as at each radio transmitter node. The NSI information goes towards the source terminal and it is not synchronised to the media stream. However, the NSI reports must be frequent enough and thus an automatic scaling mechanism with respect to the number of destination terminals is required in order to well accomplish large multicast sessions without significantly loading the network, especially in the uplink direction. By the above observations and requirements, RTP/RTCP looks fine. It already has an automatic scaling algorithm based on the size of the concerned multicast group and sender/receiver(s) reports contain a fairly complete set of information about the provided network QoS. Another possibility is to deploy ICMP, because it introduces a small overhead and the control messages can be triggered whenever required. IP encapsulation either in a datagram payload or in an extension header is practically feasible, but not recommended for efficiency reasons.

V. SIMULATION RESULTS

Goal of the simulation analysis is mainly to compare the different design choices for the transferring of JSCC/D control/signalling information. The study has been carried out by a well-known simulation tool, namely OPNET [9]. Simulation results show and compare the overhead due to the transmission of control/signalling data with the communication mechanism described in the previous section.

Fig. 2 depicts the simulation scenario composed by a single JSCC source that sends unicast video flow to a JSCC destination. At the source side, a wired IP network connects the source with a wireless transmitter. At the destination side, the wireless receiver node is connected to the destination node through a further wired IP network.

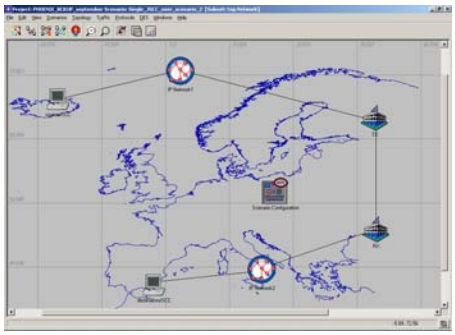

Fig. 2. Single JSCC/D scenario

The traffic is generated by an MPEG4 video source [10] at 370kbps. The control/signalling overhead is independent from the wireless technology (e.g. 802.11b [11] or UMTS [12]) so, we conducted simulation only on a WiFi scenario.

Hereafter, the simulation results of each mentioned control/signalling information is provided.

A. SSI

Fig. 3 depicts the traffic overhead for SSI control/signalling information. The SSI has been fixed into multiples of 2 bytes (3 bits for the code and 13 bits for the size). Lower overhead is for encapsulation into the hop-by-hop IPv6 extension header of the video packet. In this case the SSI average traffic is less than 3 KB/sec. On the other hand, if the SSI uses an out-of-band scheme (both ICMPv6 message and IPv6 payload encapsulation), the overhead is significantly higher and the average is about 16 KB/sec or 18 KB/sec respectively.

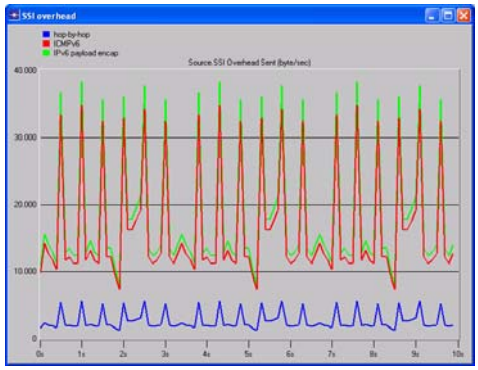

Fig. 3. SSI overhead with different control/signalling schemes

Results show therefore that the hop-by-hop encapsulation introduces significantly lower overhead in the network if compared to out-of-band schemes.

It is interesting to evaluate the SSI overhead with different MPEG4 video sources. We simulate four different cases with the hop-by-hop encapsulation mechanism deployed.

Simulation results (Tab. 1) show that the SSI overhead is proportional to the video codec rate.

video source	SSI overhead
MP4 370kbps	3 KB/sec
MP4 192kbps	1.5 KB/sec
MP4 125kbps	1 KB/sec
MP4 64kbps	500 B/sec

Tab. 1 SSI overhead for different coding rates

Obviously, choosing a source with a lower rate means sending lower SSI overhead on the network.

B. CSI

Fig. 4 depicts simulation results for CSI overhead related to the ICMPv6 and IPv6 payload encapsulation schemes. The CSI information has been fixed to 8 bytes. To compare the overhead for different transmission schemes, we chose a 200 msec CSI refreshing period. Results show that overhead is very similar in both cases.

Fig. 4. CSI overhead

For IPv6 payload encapsulation, the CSI overhead is about 280 byte/sec while for ICMPv6 the overhead is about 260 byte/sec. The little difference is plausible because with the

IPv6 payload encapsulation the overhead is higher than in ICMPv6 (the UDP header is 8 bytes while the ICMPv6 header is 4 bytes).

An important parameter to evaluate is the CSI frequency because it influences the reaction time of the JSCC/D system to wireless channel conditions changes. A high frequency means a higher overhead but also a better knowledge of the channel condition, particularly useful when it changes rapidly.

C. DRI and SAI

Fig. 5 depicts the overhead due to DRI. The graph shows that DRI is very bandwidth consuming and introduces a high overhead (about 180 KB/sec, higher than the video source code rate of 370 Kbps).

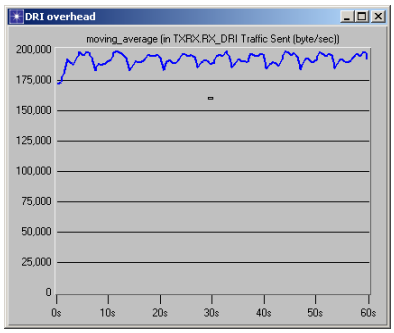

Fig. 5. DRI overhead

When the destination of the video data is not the wireless receiver (Fig. 2), the DRI can reduce the video quality at the destination side (especially, when a bottleneck between the wireless RX node and the destination node is present). In terms of overhead, simulation results for SAI lead to the same conclusion as for DRI.

D. NSI

To evaluate NSI overhead two transfer schemes have been considered: ICMPv6 messages and ad-hoc signalling. For ad-hoc signalling we simulated RTCP encapsulation. RTCP reports are, by default, sent by destination (RTCP receiver report) and source (RTCP sender report) every 5 seconds. A value of 5 seconds may not be suitable to recover from highly variable network conditions. A lower transmission period can be more suitable.

Fig. 6. NSI overhead

The NSI refreshing period influences the NSI overhead linearly. Fig. 6 shows simulation result for NSI timer of 100 msec. Sending the NSI with RTCP report requires more bandwidth than ICMPv6 message. In the first case, the overhead rate is about 1.1 KB/sec while, in the latter about 750 B/sec. An NSI refreshing period of 100 msec could be considered a good compromise in order to recover from highly variable network conditions.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have investigated the several promising mechanisms to transport control and signalling information through an IPv6 network, i.e. supporting the Network Transparency concept, in order to enable JSCC/D system deployment.

The architecture proposed in the framework of the PHOENIX IST project encompasses a complete transmission chain from application level source coding to wireless and wired channel models, requiring JSCC/D control/signalling information exchanges. Thus, we have specified some solutions to effectively deliver JSCC/D control data across the network and the protocol stack. We have tackled the issue of the Network Transparency, which entails both the internal cross layer communications and the control data transfer through heterogeneous networks, in a transparent manner.

Therefore we have considered various signalling and control information that needs to cross the network from the generation point to the target destination(s), and the more effective mechanisms that could implement the concept of the Network Transparency for each of them.

Accordingly to the collected simulation results, encapsulating the SSI into IPv6 header by hop-by-hop option could be the most suitable way for carrying it. The CSI and NSI signalling entail lower overhead when they are sent by ICMPv6 messages, while for DRI and SAI a dedicated IP flow could be the most reasonable solution. However, DRI and SAI should not be delivered on the network due to their heavy overhead.

Concerning future work, it is worthwhile to further develop the overall JSCC/D network architecture and subsequently simulate more complex scenarios, with the goal to optimise the transmission of multimedia over wireless IP 4G Networks.

REFERENCES

[1] J. L. Massey, “Joint source and channel coding,” in *Communication Systems and Random Process Theory, NATO Advanced Studies Institutes Series E25*, J.K. Skwirzynski editor, pp. 279-293, Sijthoff & Noordhoff, Alphen aan den Rijn, The Netherlands, 1978.

[2] PHOENIX: FP6 IST European project http://www.ist-phoenix.org/

[3] S. M´erigeault and C.Lamy, “Concepts for exchanging extra information between protocol layers transparently for the standard protocol stack,” in *Proceedings of IEEE ICT’03*, Tahiti, French Polynesia, Feb. 23–Mar. 1st. 2003.

[4] S. Deering et al., “Internet Protocol, Version 6 (IPv6) Specification,” RFC 2460, December 1998.

[5] A. Conta and S. Deering, “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification,” RFC 2463, December 1998.

[6] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” IETF RFC 1889, January 1996.

[7] J. Postel, “Transmission Control Protocol,” RFC 761, January 1980.

[8] J. Postel, “User Datagram Protocol,” RFC 768, August 1980.

[9] OpNet Modeler by OPNET Technologies Inc. http://www.opnet.com

[10] MPEG-4 Video Group, “Overview of the MPEG-4 Standard,” ISO/IEC JTC1/SC29/WG11 N3444, Geneva, May-June 2000.

[11] “IEEE 802.11 – 1999 edition, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Standard 802.11, 1999.*

[12] H. Kaaranen, A. Ahtiainen, L. Laitinen, S. Naghian and V. Niemi, “UMTS Networks: Architecture, Mobility and Services,” John Wiley and Sons, Chichester, England, 2001.