Architecture Considerations for Video Conferencing in the Internet with Wireless Links

Frank Lyonnet *Walid Dabbous*

INRIA Sophia Antipolis
2004 route des Lucioles, BP-93
06902 Sophia Antipolis Cedex (France)
e-mail: Frank.Lyonnet@inria.fr, Walid.Dabbous@inria.fr

Philippe Perrot

Laboratoires d'Electronique Philips
22 Avenue Descartes, BP-15
94453 Limeil Brevannes Cedex (France)
e-mail: perrot@lep.research.philips.com

Abstract

Multimedia applications are already supported over the Internet with application level adaptation mechanisms. However, wireless links have specific characteristics requiring that these adaptation mechanisms to be revisited. In this paper we discuss the feasibility of an end-to-end solution to support video conferencing over the Internet with wireless links. This solution is based on receiver driven layered multicast with hierarchical data coding and transmission, and on application level forward error correction mechanisms. It is being implemented within the Rendez-Vous video conferencing application.

1. Introduction

The range of the basic services provided by the Internet is already wide: e-mail, newsgroups, Web, file transfer, remote access, etc. With the development of new real-time transport protocols (RTP/RTCP) that are adapted to the current best effort service of the Internet, multimedia applications such as video conferencing, Internet telephony, shared white boards, etc. are now made feasible and broaden the scope of available services.

In parallel, Internet access means are getting more and more diversified, thus offering different environment transport conditions. This is particularly true with the increasing development of mobile networks. Some mobile data services are already being commercialized (GSM data, CDPD) and on-going normalization is taking place to specify new data services, like in ETSI for example where a whole range of data profiles for the DECT are being defined.

In this context, LEP (Laboratoires d'Electronique Philips) has established a collaboration with INRIA in order to work on the performance issues of Internet video conferencing applications when a wireless environment was considered. The bad conditions in terms bit error rates introduce new parameters that need to be taken into account for a efficient video conferencing over the Internet including wireless links.

In this paper, we describe an end-to-end solution to perform video conferencing over the Internet with wireless links. In section 2, we will make a brief description of how multimedia applications are supported on the Internet today. In section 3, we will discuss the wireless link characteristics and their impact on multimedia data coding and transmission control. In section 4, we detail the proposed solution and we discuss its feasibility. Section 5, concludes the paper and present future work.

2. Real time multimedia on the Internet

Supporting multimedia applications over the Internet is not a trivial task as one has to face varying conditions in terms of delay, jitter and packet loss directly affecting the subjective quality rendered to the users. In fact, multimedia data transmission is not a simple extension of the classical text data transmission. The reason is simply that multimedia applications needs are different from needs of classical data transfer applications (e-mail, ftp, telnet). Most classical applications does not involve more than two users: a source and a destination. Moreover, the transmission delay is often not a serious problem for a user: who cares that its outgoing e-mail takes 5 or 10 seconds to reach its destination? In the opposite, multimedia applications may involve more than two users (a 10-person videoconference for example). Furthermore, these applications may have specific needs such as low delay and jitter in order to ensure interactivity and smooth playout of data. Specific protocols and mechanisms at the transport and application level are therefore required.

Those mechanisms already exist and multimedia applications are today a reality in the Internet. Applications such as *vic* [McCanne95] and *vat* [Vftp] and *wb* are used regularly to transmit seminars on the network. Other applications such as RendezVous [RVweb], FreePhone [FPweb] and Rat [Rweb] are also available. All these applications are based on three key elements : a multicast network service, an adequate transport protocol and specific transmission control mechanisms integrated within the application.

2.1. IP Multicast

One of the main factors enabling multimedia application support is the IP multicast service. Of course, a multicast transmission, can be emulated by opening as many connections as possible destinations. But clearly this is not advisable as it waste much network resource and also as it implies that the source knows about each destination. The multicast extensions of the IP connectionless best effort service solve this problem [Deering91]. From the application point of view the Internet offers the ability to transmit to a multicast group, defined with a group identity called multicast address. If one wants to transmit to this group it does as usual but with this multicast address as a destination address. And if one wants to receive what is sent to a group it simply sends a group membership request through the local network interface. Efficient transmission of multicast packets is achieved via the Internet using a multicast routing algorithm. The Internet overlay that does support multicast routing facilities is called the MBone (Multicast backBone).

2.2. RTP

Classical transport protocols such as TCP are not suitable for the transmission of real time multimedia flows. As TCP is designed for a reliable transport of data, this protocol uses a repeated retransmission scheme which unfortunately leads to an increase of the end to end delay in case of congestion. As in real-time flows the data frames have a limited lifetime, such retransmissions may be useless. In addition TCP does not support multicast transmission. UDP connection-less multicast service seems therefore more appropriate. However, multimedia applications need a framing protocol in order to provide end-to-end delivery services for data with real-time characteristics, such as interactive audio and video. Those services include payload type identification, sequence numbering, timestamping and delivery monitoring. The Real-time Transport Protocol (RTP) [rfc1889] was designed to provide these services. RTP supports data transfer to multiple destinations using multicast distribution if provided by the underlying network. Typically, RTP runs on top of UDP/IP multicast in the Internet. RTP does not guarantee delivery or prevent out-of-order delivery, nor does it assume that the underlying network is reliable and delivers packets in sequence. The sequence numbers included in RTP allow the receiver to reconstruct the sender's packet sequence.

RTP consists of two parts, a data part (referred as RTP) and a control part (referred as the Real-time Transport Control Protocol, RTCP). RTP data packets start with a 12-byte header preceding the data payload consisting of video frames, a sequence of audio samples or even distributed interactive game data. The type of the payload is contained in one byte of the RTP header. The RTP header carries also the sequence number of the packet and a 32-bit timestamp representing the generation instant of the payload contained in the packet. Finally, the packet header contains a randomly generated 32-bit scalar that uniquely identifies the packet source. RTCP conveys information about the participants in an on-going session, namely the network loss and delay jitter perceived by each participant. This is achieved by periodic transmission of specific RTCP packets (referred as Sender Report - SR and Receiver Report - RR). Other RTCP packets include information such as participant descriptors (name, e-mail, phone number) or application specific control information.

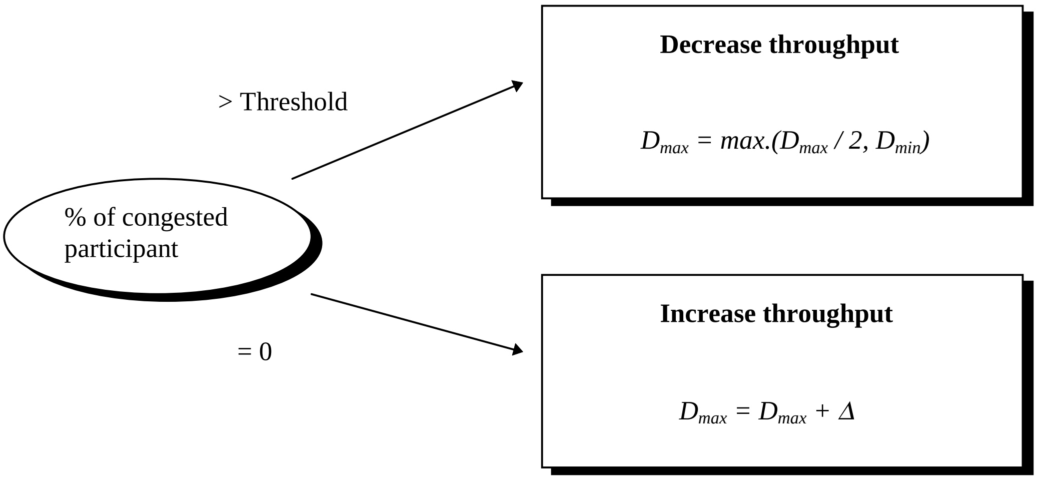
In summary, RTP/RTCP defines the generic headers and control messages for real time multicast applications. It allows the application to have a good estimation of the network conditions and to optimize the handling of a given real time media through high level adaptation mechanisms.

2.3. Adaptive applications

As previously cited, multimedia applications have specific delay, jitter and packet loss requirements. However, the current best effort service does not offer any garantees in terms of delay and packet loss. Two different approaches are possible: the first one is to change the Internet to provide multiple services (best effort, garanteed, etc). This is the goal of some working groups in the IETF such as the RSVP working group. Another approach is to consider the Internet as it is today and try to use it optimally, that is minimizing the negative impact of this best effort service on the transmitted multimedia data. The idea is to implement specific transmission control mechanisms at the application level to cure from high jitter and packet losses (not too much can be done to cure from high transmission delay in interactive non predictable applications). We will focus hereafter only on mechanisms to cure from packet losses (see [Vftp] for a description of a playout mechanism to smooth the jitter). These mechanims try to reduce the *number* of packet losses and to reduce the *impact* of remaining packet losses.

2.3.1. Source throughput adaptation

A possible way to reduce packet losses is to perform source level adaptation. In such a scheme, each data source gets feedback on the reception conditions of the destinations. If no congestion is detected (the number of congested destinations is zero) the source increases its maximum throughput (D_{max}) in order to probe for available network resources [Turletti94A][Bolot94]. If the number of congested destinations is nonzero but below a certain threshold, no action is performed. Otherwise, the source considers that *its* output flow is partly responsible for the congestion and reduces its maximum throughput (D_{max}). The INRIA Videoconference System (IVS) [Turletti94B] uses such a scheme. The used algorithm is depicted in Figure 1.



Decrease throughput
$D_{max} = max.(D_{max} / 2, D_{min})$
> Threshold
% of congested participant
= 0
Increase throughput
$D_{max} = D_{max} + \Delta$

FIGURE 1. : The IVS source level adaptation scheme

2.3.2. Receiver-driven layered multicast

The source throughput adaptation may be satisfactory in the case of unicast transmission but have less interest in a multicast transmission. In this case any action taken to reduce the consequences of congestion will affect all the destinations. This means that for a congestion occurring in an isolated node of the multicast session we may decrease the user quality of all the destinations by reducing the source throughput. To avoid this problem, it has been proposed by [Turletti94A], [McCanne96] to transmit hierarchically encoded data. Each source splits its output in several spatially or temporally complementary flows. For spatially complementary flows we handle one output containing the base data (coarse resolution) and the others complementing the base flow with higher resolution details. For temporally split flows we may for example transmit odd and even numbered frames on seperate flows (2 flows split). Network level mechanisms are used to send to each destination only the flow(s) that it can handle while avoiding network congestion. In the Internet, this may be achieved by using a separate multicast group for each flow and by ensuring that each destination subscribe to the group(s) corresponding to the flow(s) of interest. As the multicast routers support pruning [IGMP], this scheme prevents the transmission of multicast flows on sub-trees that do not contain receivers that indicate explicitly their interest to receive these flows. This mechanism that engages the receiver to detect its optimal reception conditions is called receiver-driven layered multicast (see Figure 2).

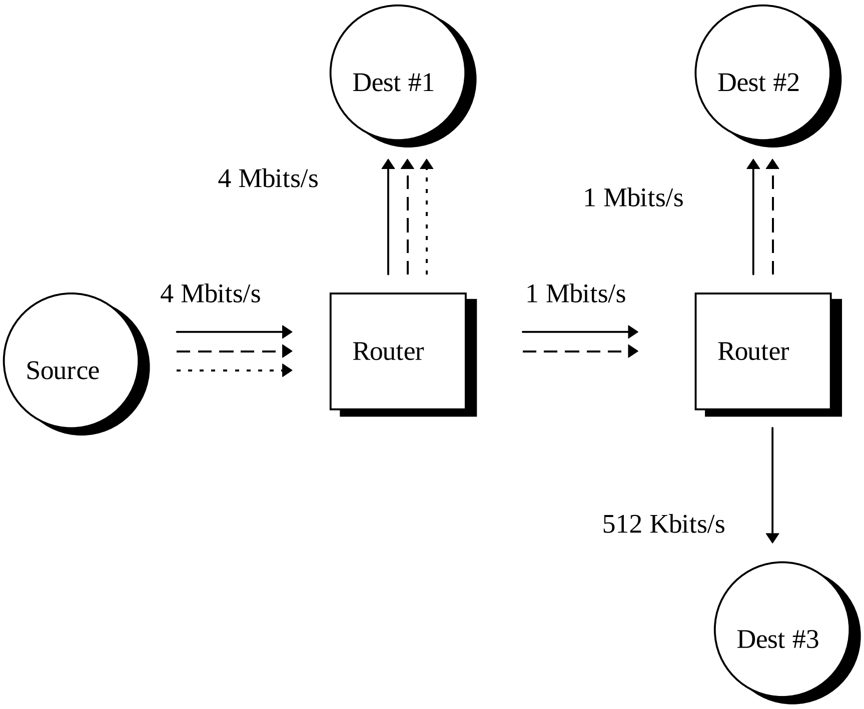


FIGURE 2. : Receiver driven layered multicast

2.3.3. Forward Error Correction

Even if multicast congestion control algorithms are supported either by the source or the receivers, the transmitter’s packets may still be lost in the network due to the traffic generated by other sources. We should find a way to cure packets losses. The retransmission of lost packets could be envisaged only in cases where the end-to-end delay is sufficiently small to retransmit lost packets while subsequent packets are queued in the playout buffer. However, if the end-to-end delay is large, retransmission will not solve the problem as data frames have a limited lifetime. Redundancy mechanisms should be used to overcome the packet loss problem. In [Huitema96], a generic packet level Forward Error Correction (FEC) scheme is described. This scheme shows interesting results in several cases and in particular for multicast data transmission. This «transport level» scheme is based on transmitting N-M additional packets constructed by XORing properly the original M packets. It allows to reconstruct the M packets even if up to N-M packet loss occurs during the transmission.

However, if we consider the semantic of the packets payload we can go even further. [Bolot96] proposed a perceptual redundancy mechanism where a main audio stream is complemented with another time-shifted audio stream of redundant semantic value. For example we can have a PCM encoded audio stream complemented with a GSM stream (see Figure 3). A preliminary study on this mechanism for real time audio conferencing has been done at INRIA and presented very satisfactory results in certain packet loss conditions. We will describe in more detail a similar mechanism for video applications.

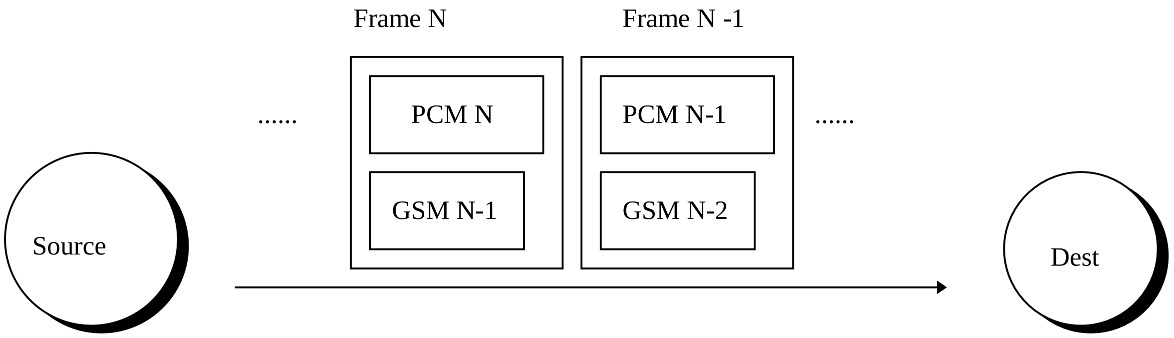


FIGURE 3. : Example audio redundancy scheme

2.3.4. Adaptation to the available computing power

The heterogeneity of the Internet is not only in terms of link capacity but also in terms of computing power of the hosts connected to it. We then have to take actions to control the application level QOS according to the available computing power [Fall95]. In other words, this means we need to allow graceful degradation of the application level QOS with CPU overload. This becomes quite an important issue if we consider multicast video/audio conferencing where costly compression / decompression methods are used. A lack of computing power currently leads to arbitrary packet losses in the kernel of the destination hosts.

Classical adaptation mechanisms can thus be applied. They rely on packet loss (in the network or in the kernel) to reduce the incoming bandwidth. Fortunately a reduced input bandwidth means less needs in computational power and then permits to the stream to be gracefully degraded. But adaptation schemes such as receiver layered multicast have a limited granularity and dynamic join and leave have a cost. Its is in fact not suitable to use these mechanisms to handle slight variations in terms of available bandwidth or CPU power. Complementary mechanisms have to be developed.

An arbitrary packet loss can be very damageable to this accuracy depending of the video compression method in use. For example using H261/H263 encoding, we practically see that a packet loss as a lot of consequences especially if differential coding is used between frames. But if we don't have the time to fully decode a packet we certainly can take advantage of the information we have inside this packet to use it at a later time. Commonly, a video decompression process can be split into three main phases : the entropy decoding phase, the reverse transform phase and the final colorspace conversion/rendering phase. After each of this phases we get a consistent (a semantically correct) video image expressed in some space (for example with H261/H263 method this is DCT space, YCbCr space and finally RGB space). If differential coding is used between frames the 2D image representation is complemented with a logical representation (for example a simple boolean matrix with H261 in intra only mode). At this stage, image degradation can be limited to only a lower frame rate if we have enough CPU time to go through the first entropy decoding phase, which typically cost a third of the total process time of a frame. This may be done by giving the highest priority to entropy decoding of the incoming packet. Upper phases are only engaged if we have enough time to realize them (i.e. without causing any kernel level packet losses). This mechanism, can be combined with a priority mechanism between flows giving a great flexibility to the whole system. An implementation of this mechanism has been incorporated to the Rendez-Vous videoconference tool and it shows excellent results on old low power workstations with a sustained subjective quality.

3. Realtime multimedia over wireless Internet links

At the opposite of the wired links of a classical Internet, wireless links are not free of bit errors. Fully protecting the data flow at the link layer is often too much expensive in terms of bandwidth. Consequently this new parameter has to be introduced in the realtime multimedia over Internet equation. It effectively affects the way transmission control is achieved over the Internet, the way multimedia data is coded and finally it pushes away the Internet heterogeneity even further.

3.1. Wireless aware adaptation mechanisms

Data flows can be corrputed on wireless links. This first means that in classical transport protocols such as TCP a bit error can result in a packet loss. TCP/IP stacks handle two checksums: one for the IP header and one for the TCP header and data. Corrupted TCP packets are discarded at the kernel level and considered as «lost». In addition, this packet «loss» will result in the reduction of the congestion window, as it will be incorrectly interpreted by TCP as a congestion signal. In fact it has been shown in [Bakre95] that TCP performances are very poor over wireless links and that modified TCP algorithms have to be used. It is in fact essential for an Internet adaptation mechanism to distinguish packet loss and packet corruption.

Furthermore, the cell based design of most of the wireless systems, via the latency of handover procedures, introduces also packet losses which are not due to congestion nor packet corruption. This also has to be distinguished for a transmission control mechanism to be efficient. The adaptation mechanisms should therefore distinguish losses due to corruption or handoff from loss due to congestion.

3.2. Wireless aware multimedia coding

We have seen in section 2.3 that for some data types such as real-time audio and/or video, it is desirable to pass the handling of errors to the application. In a wireless environment, with an high bit error rate, application level error handling could be even more interesting. In videoconferencing applications we can certainly achieve more efficient forward error correction regarding the network overhead if we consider the bitstream as issued from a video or audio signal. Such considerations are commonly depicted as joint source / channel encoding techniques. The image coding is involved into the process of correcting the channel errors.

3.2.1. Adding robustness to a video codec

In order to transmit video or audio signal, a compression is performed to reduce the used bandwidth. This compression is achieved by reducing - ideally removing - any redundancy in the signal. Depending on the compression methods, it can be (but not always) decomposed into three main phases: image transform, quantization and entropy coding. This last step, entropy coding, can first be the place for «wireless optimized» redundancy coding. In [Redmill93], an entropy coding method resilient to bit error is described. This is the first step of a what we can call signal level redundancy ensuring FEC functionality.

But we can also apply the FEC mechanism at the image transform step. To do this the transform must generate a signal that can be decomposed into as small as possible independently processable parts (for example simple block decomposition of an image using transform such as DCT on each block). This is essential in order to reduce the consequences of an erroneous bit inside the data stream (the ideal but unachievable smallest part in our case is of course a part coded on a single bit). It should be also pointed out that using such transform is not sufficient: we must have a way to detect erroneous parts. Specific coding technics should therefore be used in order to efficiently identify wrong parts that can be in the form of erroned or shifted bits.

3.2.2. Design issues for a wireless aware video codec

In classical compression mechanisms, we generally choose an image transform that is close to optimality in terms of image decorrelation. The optimal transform of an image is the Karunen-Loeve transform and the more commonly used transform is the Discrete Cosine Transform which is near to optimality. Optimality in terms of image decorrelation means that all the natural redundancy of an image is removed. In [Normand96], a new image transform called Mojette transform is described. This transform can be adjusted in order to achieve a given amount of redundancy in the transformed signal and also it can be decomposed into small independent parts. Ideally we could go even further with compression methods that could be decomposed into parts of equal importance in terms of subjective quality impact, with a controlled amount of redundancy. This unfortunately is not so trivial.

Usually the compressed signal does not contain data of equal importance in terms of impact on the subjective quality (on the basis that the human eye is more responsive to low spatial frequencies). But we can typically create a video stream compressed with a spatial frequency domain transform and associated with a hierarchical decomposition in several subflows (see Figure 4). The obtained subflows effectively have different importance in terms of impact on subjective image quality. We can then protect the different subflows unequally in function of their importance on the whole image quality. This can be for example done using a modulated bit level redundancy on each of the flows. Such schemes are called Unequal Protection Schemes [Horn][Sadka] [Belzer]. In such hierarchical and unequally protected flows, a bit error pattern has an equal impact on the different flows regarding the subjective image quality. But this method has a big disadvantage: the different flows are not independent. We effectively need to receive the base flow in order to make the other flows usable.

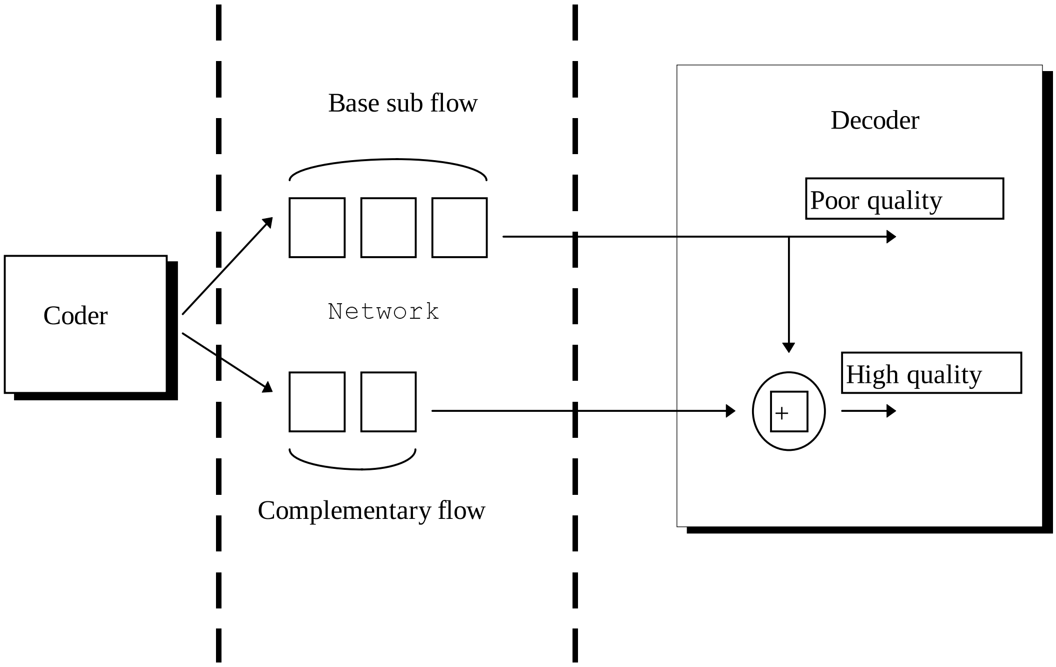


FIGURE 4. : Hierarchical flow transmission

3.3. How to overcome hetherogeneity

We've seen that wireless Internet links characterisitics require new features to be added to a video conferencing application. If now we consider the wireless links as part of the global Internet we have to find efficient solutions to allow videoconferencing that involve participants on both wired and wireless hosts. The main question is about the need to split the transmission control over the two network parts (wireless and wired) or to keep an end to end transmission control. The first possibility can be implemented using more or less complex gateways and the second using hiearchical transmission. The rest of this section is dedicated to the presentation of these different interoperability scenarii.

3.3.1. Gateways

The first solution is to use an application level gateway that could be placed near the IP gateway of the considered wireless subnetwork. Several solutions can be considered and their key point is their complexity.

A high complexity solution would consist of *transcoders* (see Figure 5) i.e. two codecs working head to head each of them generating an encoded flow suitable to the part of the Internet it is connected to (wireless or wireline), like e.g. an MPEG-H.261 gateway. This offers certainly the best mean to efficiently transmit the flow on both the wireless and wireline parts of the Internet. But this introduces additional delay due to the overhead of successive decoding and coding. Another drawback is the cost of such a system. Real time video compression requires a lot of computational power, thus raising the cost of he whole infrastructure.

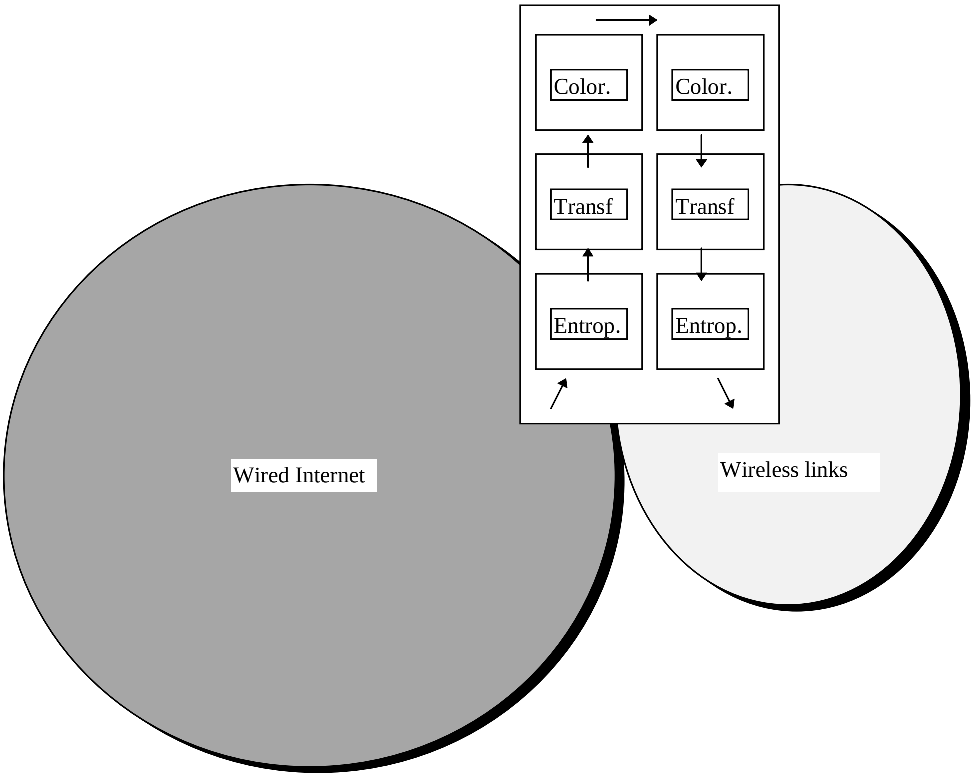


FIGURE 5. : A transcoder gateway

Another solution is to sacrifice a part of the efficiency of the gateway in order to bring its complexity down, for example by only adding bit level FEC to the flow being processed (see Figure 6). This does not require a complex and delay costly implementation, as it could be processed by the wireless network interface hardware. However, as this gateway does not perform source code conversion, the wireless part should have a bandwidth at least equal to the bandwidth of the incoming flow from the wireline part. Intermediate solutions exist between these two extrems (transcoder and simple gateway introducing bit level FEC). However, these solutions are not complient with the «end-to-end» argument i.e. the Internet philosophy of reducing the complexity of the network in order to make it easily deployable.

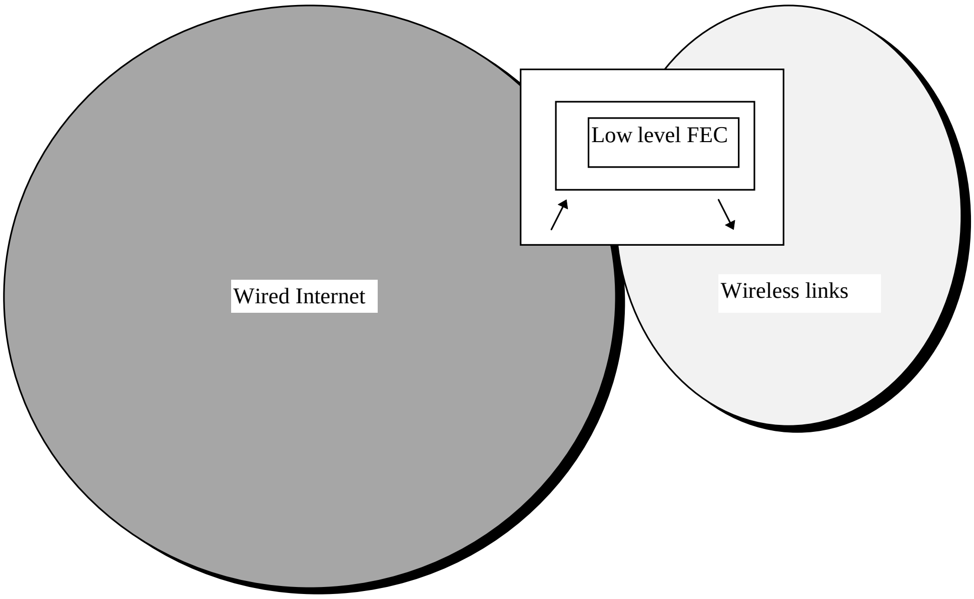


FIGURE 6. : A simpler gateway

3.3.2. Hierarchical transmission

An end to end approach i.e. without any gateway within the network is preferred. This solution is more elegant from an architectural point of view. It consists in splitting the source flow into parts transmitted on different multicast groups as described in section 2.3.2. Users join only the selected groups. In our case, we could then consider a base flow distributed to all the participants to the conference, some complementary flows for high bandwidth wired hosts and other complementary flows containing redundancy dedicated to efficient transmission on wireless links. This solution is possibly less optimal than those based on a gateway but its flexibility and very low cost make it an essential research axis.

4. Feasibility of an end-to-end solution

As a part of an experimentation phase to test the different solutions and scenarii the Rendez-Vous software is being used as an evaluation platform. In order to test the efficiency of an end-to-end solution to the problem of video conferencing over Internet with wireless links we are developing a layered DCT codec enhanced for transmission over wireless links with only small overhead compared to classical compression methods.

4.1. DECT simulation

The wireless links can either be DECT ones (which corresponds to LEP interest) or any wireless LAN. Waiting for the availability of a DECT hardware supporting the multimedia (unprotected) profile, we have modified an Ethernet device driver by introducing DECT error patterns in order to simulate this hardware. These error patterns are generated by DECT generative models i.e. low-complexity algorithms that reproduce the equivalent statistics at the network layer of those of real or simulated channels. These generative models were obtained by simulating a DECT transmission chain. The multipath channel characteristics (mainly the short term characteristics) have been derived from the work of the COST 207 group on mobile channel characterization. Corresponding error patterns for worst conditions at different speeds and for different environments have been produced and have served as training patterns for the generative models.

4.2. An end to end solution in a bit erroned context

A full end to end solution is by definition a solution where no transport level error correction is done through the whole path. This is true for packet loss in the wireline part but also for bit errors on the wireless part. Packets may be corrupted in the wireless part as link level FEC may be very costly. As seen in a previous section, the IP header and the UDP header and payload are protected by a checksum. This means that corrupted packets will be discarded. In fact, all the packets with at least one erroneous bit are discarded inside the kernel because the UDP checksum will fail. However, we would like to be able to exploit the non corrupted information within the packet. This means that we need to disable the UDP checksum (and not the IP header checksum as a packet with a corrupted header cannot be sent to its destination). Fortunately this UDP checksum can be easily disabled with a slight modification of Internet stack implementations. For even more efficiency it would be advisable to protect all the (Mac, IP, UDP and also RTP) headers with a bit level FEC. A Mac address protection is often done in wireless link, RTP header protection can easily be done at the application level. As for the IP header protection it could be done inside the IP over DECT profile without breaking the end to end philosophy. But for the UDP header protection it is not so easy : UDP is not known by the IP over DECT layer nor by the application layer.

4.3. A unit based approach

In a classical Internet context, videoconferencing applications must be robust to packet loss. A seen in a previous section, a way to achieve this robustness is to design the video codecs in a way a single packet loss would have a minimal impact on the consistency of the whole data flow. For example with the H.261 coding method we can arrange the basic blocks of data (the group of blocks) to be fully contained inside only one packet. With this approach each packet is independantly decodable thus reducing a loss consequences. But it should be pointed out that such packetization is not achievable with all the coding methods. With H.261 this could only be applied with a subset of the features of the coding method. If the coded data contains too large dependancies it is impossible to arrange them into independant parts. In H.261 this is achievable in pure Intra mode (without temporal compression other than block change detection) but not in Inter mode (macroblocks are coded relative to their preceeding value).

In our bit erroned context, we have to extend this concept. Here this is not only the packets that are being lost but we have also to deal with erroneous bits inside a packet. To reduce the consequence of an erroned bit we can apply the same methods as for packet loss. To be the most efficient as possible we have to find out the smallest basic block of the choosen compression method, let’s say a *unit*. To let units being independant regarding the bit errors we have to protect them independantly. We can for example apply a checksum over each of them individually (see Figure 7). If we do so we can safely decode all the bit error free (with a consistent checksum) units of a packet and simply throwing away the erroned units.

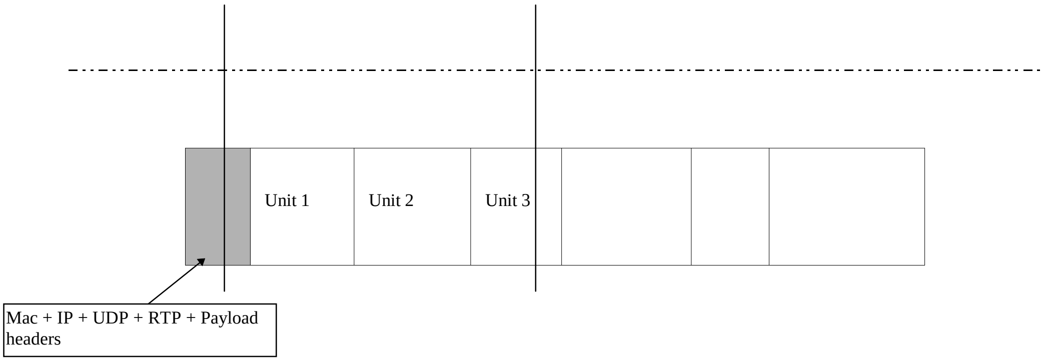


FIGURE 7. : Packet CRC check vs Unit CRC check

We have made some simulations to test the validity of such an approach on an Internet including DECT links. With have measured the unit error rate of a simulated data stream obtained with a theorical unit based video coder. We consider that this unit error rate is the inverse of the subjective quality of the decoded stream. This is though somewhat optimistic but a codec permitting this is not unachievable, depending on the unit size. Figure 8 shows interesting results concerning the unit loss being obtained with three different settings : a packet level checksum, a unit based checksum and a unit based checksum with (Mac + IP + UDP + RTP) headers protection using bit level FEC (in this case, we assume that the protection will always permit to reconstruct the headers if they are corrupted). We see that the unit error rate tends to be very high with a packet based CRC check, growing naturally with the MTU. Using a unit based CRC check gives a 30 % drop in the loss rate at the minimal MTU (576 bytes) and achieve a relative independence of the loss rate from the MTU. Protecting the headers lowers even more the loss rate that then represents a third of that with packet level CRC check at the minimal MTU.

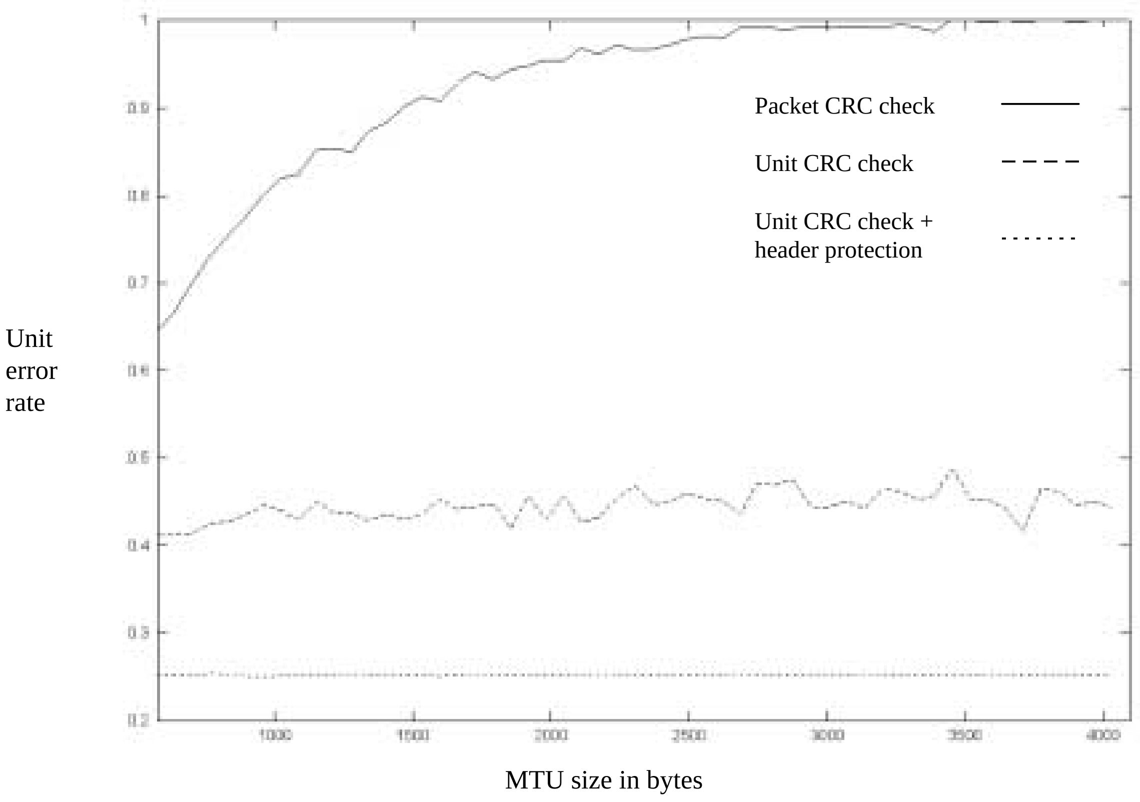


FIGURE 8. : Unit error rate with a DECT error pattern

4.4. A unit based layered DCT codec

The goal was to develop a scalable and robust codec efficient over an Internet with both wireline and wireless links.

4.4.1. Spatial layers

The Discrete Cosine Transform provides an easy way to split video data into several complementary subflows that we call layers [Amir96]. Robustness is provided by a unit based data packetization and can be complemented by bit level or higher (signal) level FEC. Figure 9 shows the bloc diagram of our codec. A video frame is split into 8 by 8 blocks. The DCT phase applied on each block is followed by a scalar quantization phase. At this point the quantized coefficient are split into a parametrable number of layers. Each layer is then split into units containing 4 blocks of luminance with 2 blocks of chrominance (as we use a YCbCr 4 :1 :1 colorspace). Each unit is entropy coded using an Huffman tree scheme. Then a checksum is computed for each unit. Units are then grouped into packets of size mapped to the underlying MTU. A unit is in no case split across two packets. The codec uses an interframe coding based on a bloc change detection between the last two grabbed frame. Thus only the modified blocks are transmitted in a frame. The codec has also support for a signal FEC based on repetition of the modified blocks of previous frames in addition to the modified blocks of the current frame.

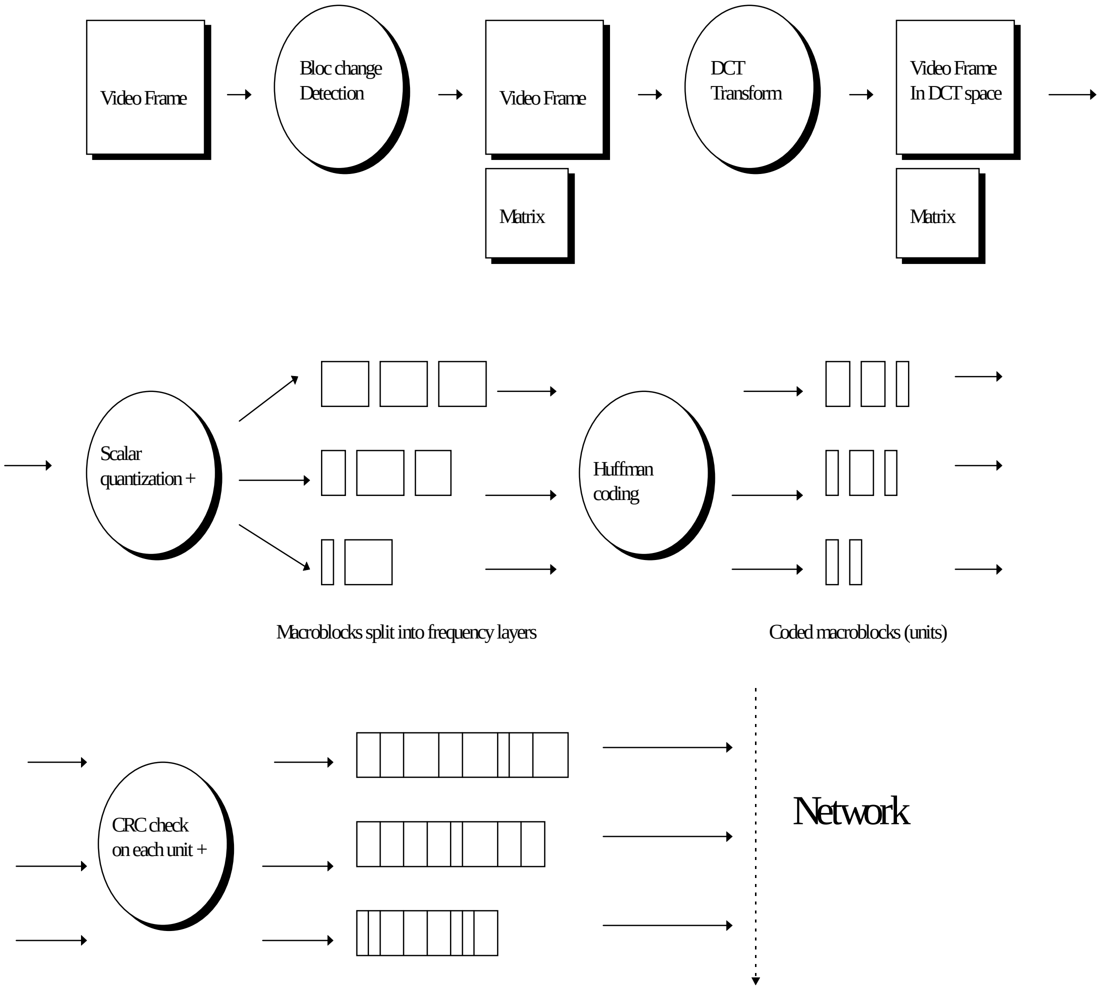


FIGURE 9. : A unit based layer DCT codec

4.4.2. Temporal layers

We can also provide a split of the flow into temporal layers. We can effectively split the frames into a number of flow each containing a fraction of the frame sequence. But as each frame contains only the blocs modified between the frame and its preceding one this lead to non consitent reconstructed images if a receiver does not receive all the layers. Fortunately in a video conference scenario, the source often consist of scenes where a single person stands in a front of a unchanging background. This means that modified blocks from frame to frame are often correlated thus leading to an somewhat consistent image.

Of course we can easily combine the spatial and temporal layers thus giving more granularity. Each temporal layer decomposes into spatial layers the same way as if applied on the complete flow.

4.5. High level Forward Error Correction

We made a statistical study on the bit errors patterns obtained with the DECT simulator. Figure 10 shows the distribution of the unit loss bursts inside a DECT flow. We can see that the burst range is very sparse and we can make a number of statements. Classical Internet FEC such as found in the FreePhone tool (see section 2.3.3) is not applicable : the burst distribution is too large. A frame to frame FEC is certainly more adequate.

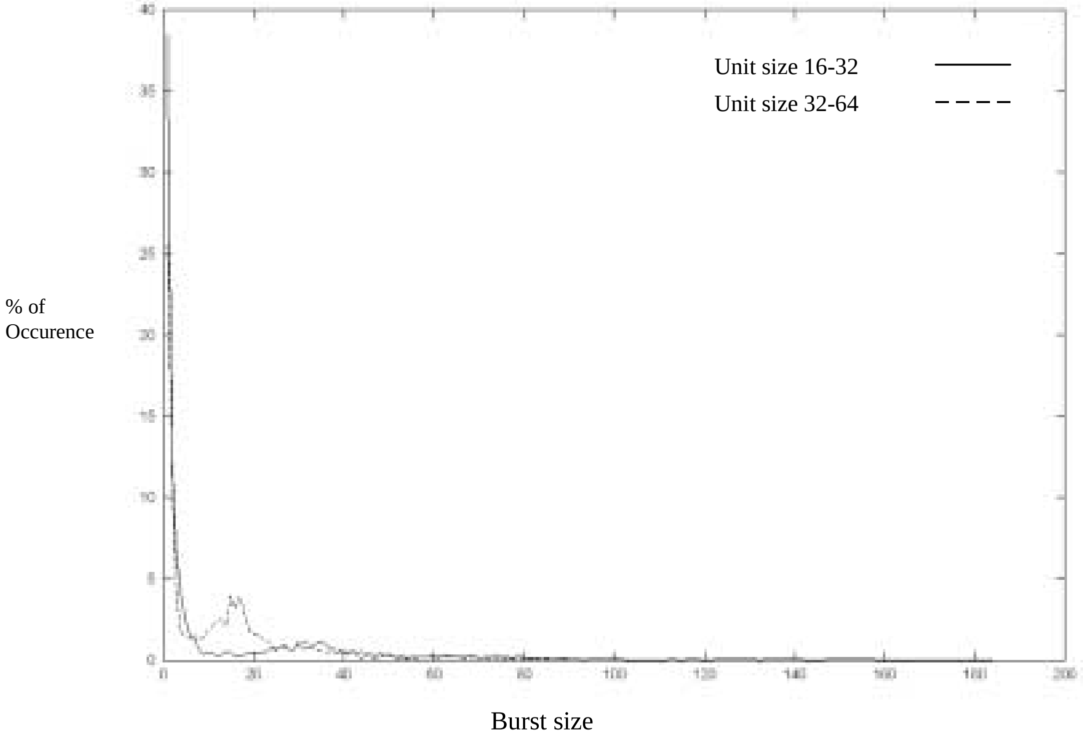


FIGURE 10. : Unit error burst distribution

4.5.1. Block based FEC

We can do FEC at the unit level easily. Using a memory of the past modified blocks, we can achieve a variable amount of redundancy by applying a logical OR between the K levels of previous block matrix with the one of the current frame (see Figure 11). We call this scheme FEC of depth K, i.e. FEC computed by ORing modified blocks in the K previous frames with the current one.

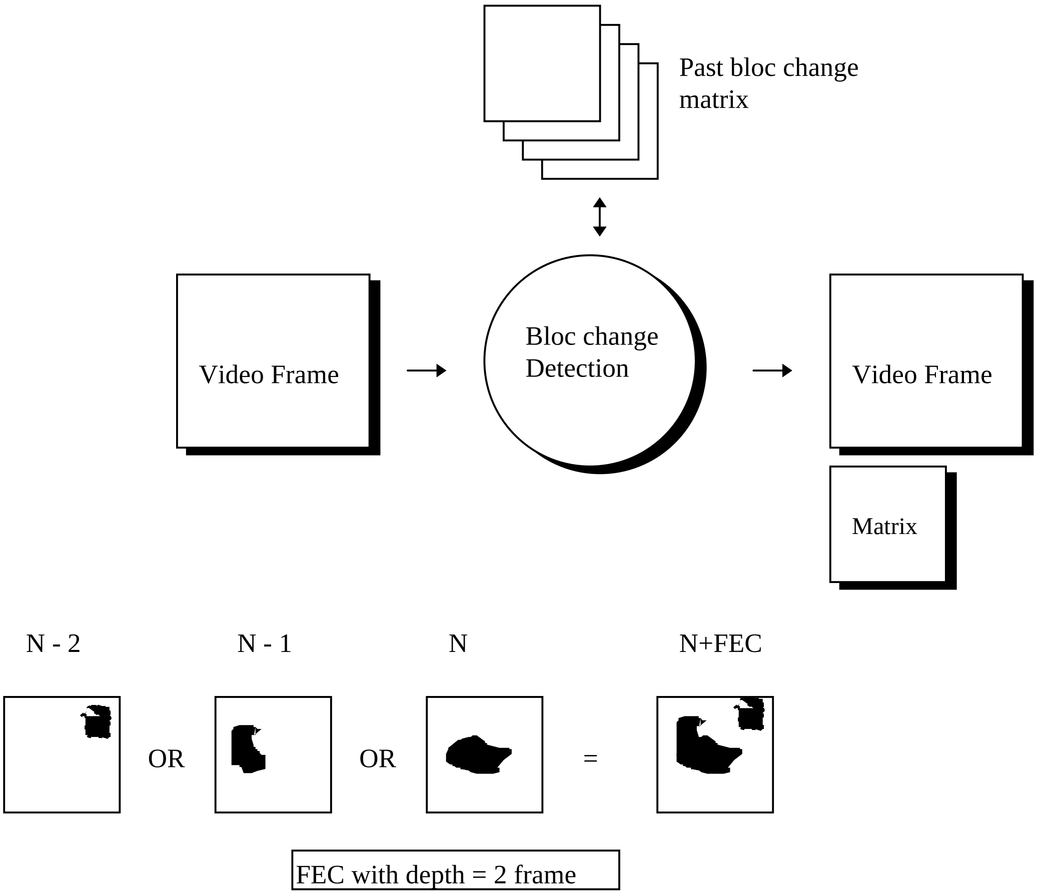


FIGURE 11. : Bloc based forward error correction

As said before there is a lot of correlation between the succeeding changed block matrix. This means that such an FEC is not so much costly in terms of bandwidth.

4.5.2. FEC as a complementary layer

Furthermore the introduced FEC can be easily isolated of the main stream. If we OR only the K previous levels we effectively obtain the FEC part of the frame. This means that our block based FEC can be completely independent of the main flow and can be transmitted as an independent layer. The FEC layer and the main layer can be recomposed at the receiver end. By definition the FEC alone is not correlated with the main flow. Thus this separation has no cost in terms of bandwidth other than the small overhead inherent to the handling of a new flow.

An independent FEC is crucial as we consider videoconference scenarii involving both wireline connected and wireless connected hosts. Using a receiver based layered multicast algorithm extended to detect the profile of the wireless link it is connected to, we can realize an efficient transmission on both the wireless and wireline parts of the network (see Figure 12). With this system, only the path between the source and the destination 2 carry the FEC flow.

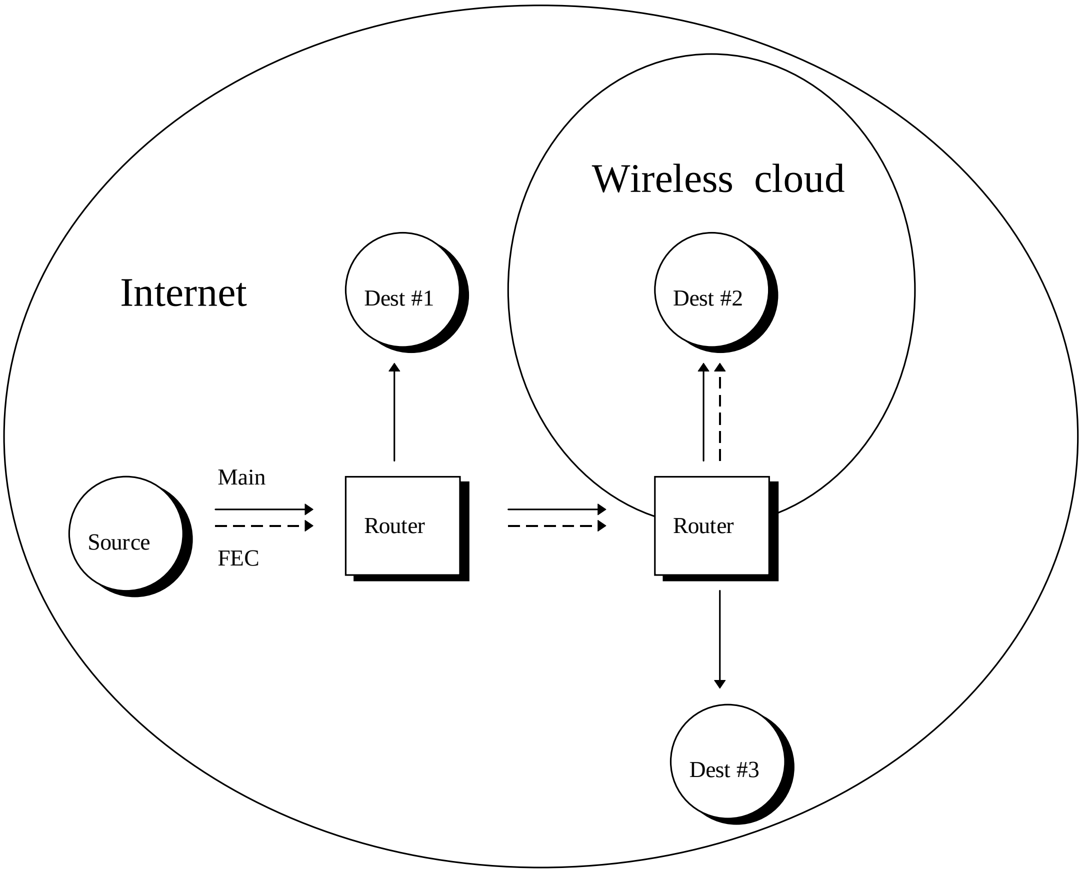


FIGURE 12. : Transmitting FEC separately

4.5.3. Block based FEC with a hierarchical codec

When considering hierarchical main flows the FEC flow must also be transmitted hierarchically. This is straightforward with frequency layers, the FEC flow decomposes into frequency in the same way as the main flow without any loss in terms of bandwidth. A receiver can subscribe to N flow layers and from 0 to K FEC layers depending of the amount of FEC it needs.

But with temporal layers, we raise again the problem of the inconsistency induced by a temporal split when using a coding method based on bloc change detection. For the FEC to be consistent with an isolated layer its depth must be greater than N (where N is the number of layers). Figure 13 gives an example in the case where two temporal layers are used, one for odd numbered frames and the other for even numbered frames. It should be pointed out that this FEC is the cure for the artifacts induced by a temporal split in layers alone we described in a previous section. In both uses, this FEC cannot be used at level lower than the number of layers.

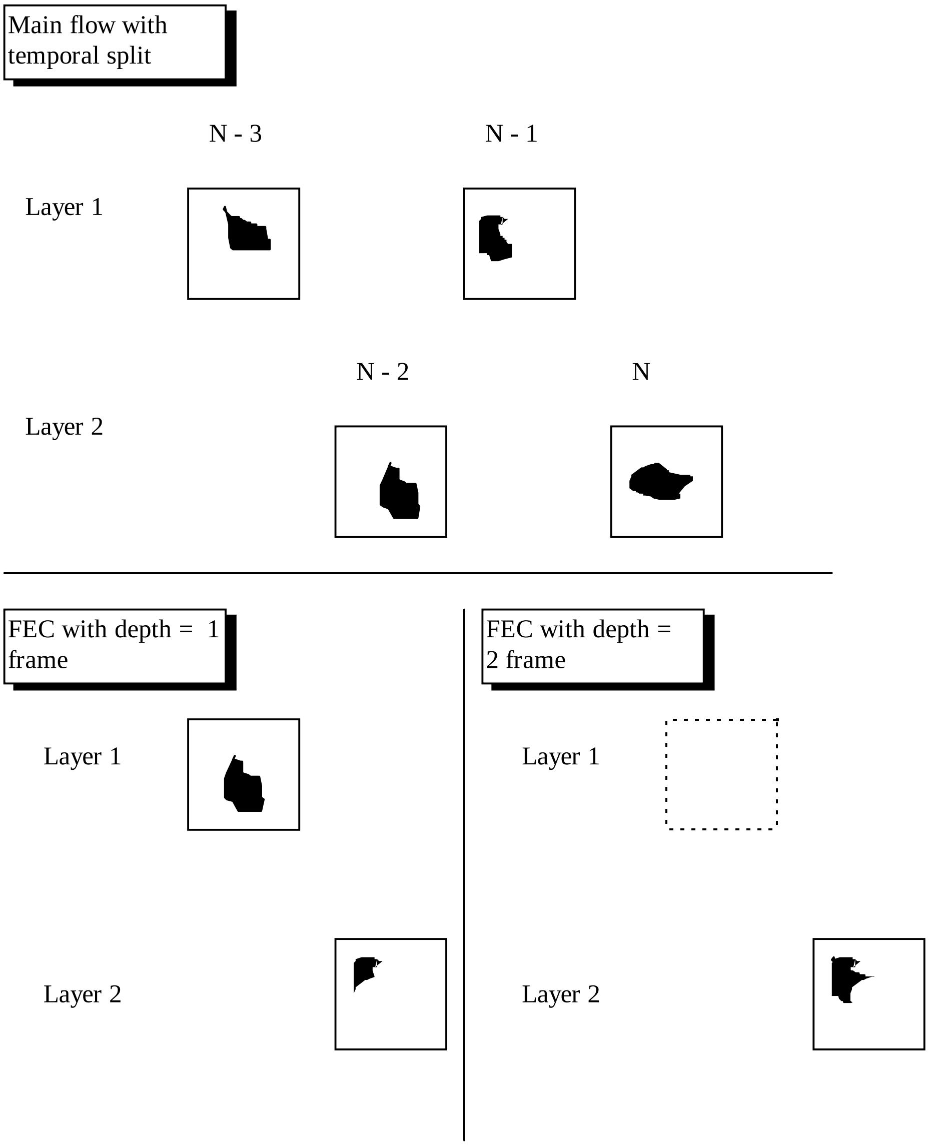


FIGURE 13. : Consistent FEC with temporal split of the flow

Anyway, it is also possible to separate the FEC temporally, in the same way we separate temporally the main flow, without bandwidth loss. FEC of depth K is obtained by subscribing to K layers of FEC. Again we can combine the spatial and temporal FEC layers leading to a wide range of complementary FEC layers to be offered.

4.5.4. Hierarchical FEC

A wide range of complementary FEC means the possibility to adapt more precisely to a given wireless link. But it also means that a single source can send an FEC customized for different wireless links into several complementary flows as shown in Figure 14.

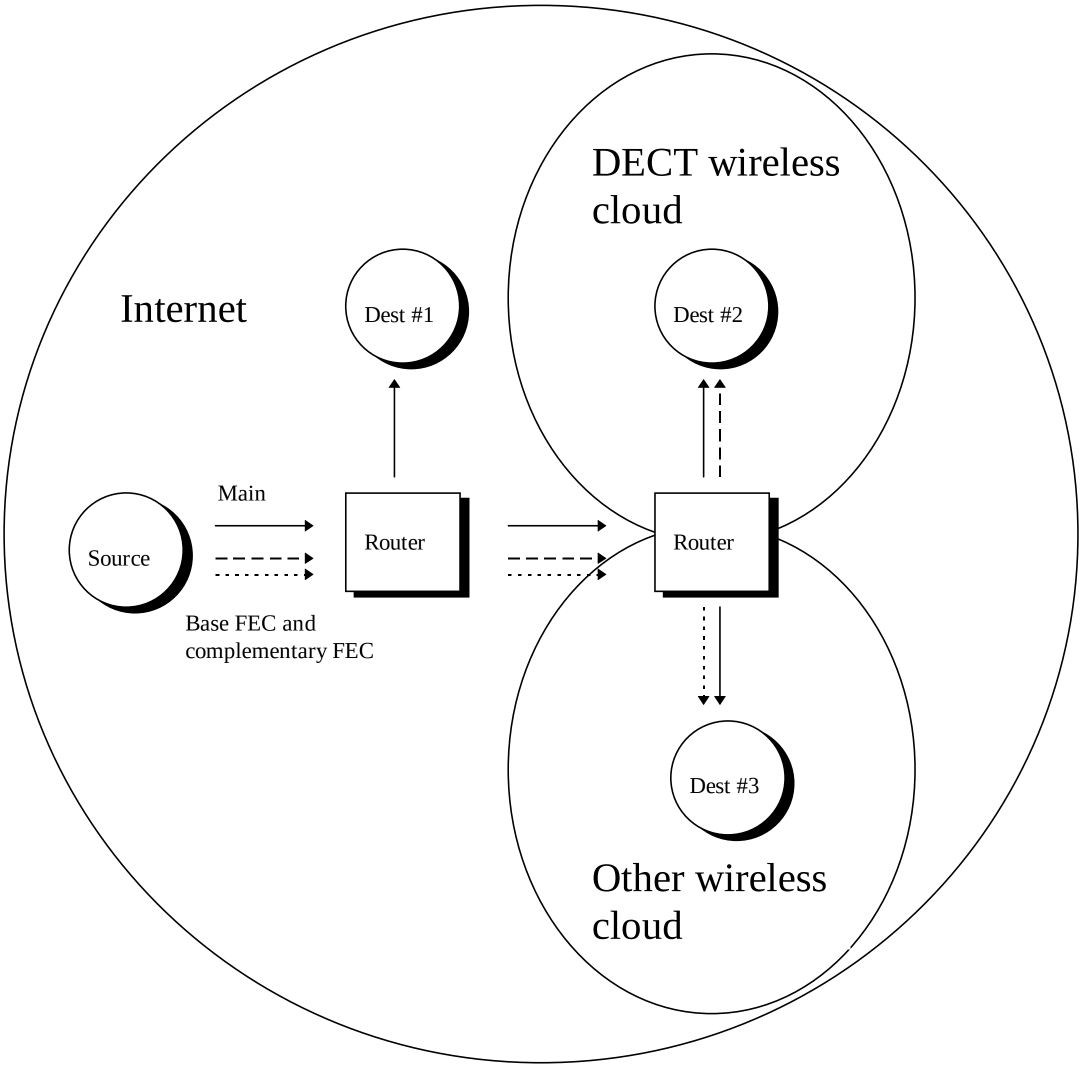


FIGURE 14. : Hierarchical FEC

4.5.5. Limitations of block based FEC

In DECT networks some burst are really too large to be managed by any FEC. Perhaps a bit level FEC to lower the raw BER of the stream could be applied. This possibly could cure a lot of unit losses at a minimal cost. The ideal solution is certainly a mix of low and high level FEC. Bit level FEC is used to «upgrade» to link characteristics to an acceptable level. Then end-to-end FEC mechanisms are applied. We are working on the implementation of these schemes within the RendezVous applications. We expect to perform experiementations soon.

5. Conclusion

Multimedia applications support over the Internet is based on adaptation by either the source or the receiver. Wireless links characteristics have an impact on these adaptation mechanisms. We presented in this paper an end-to-end approach to support video conferencing over the Internet with wireless links. We are currently implementing this approach in the Rendez-Vous video conferencing application. Further work concerns a study of the efficiency of this end-to-end approach from both theoretical and experimental aspects. Error patterns on the wireless links will be combined with existing traces of packet losses over wired Internet. Our goal is eventually to design a high level FEC mechanism that corresponds to the combined error patterns.

References

[Amir96]: "A layered DCT Coder for Internet Video", Elan Amir, Steven McCanne and Martin Vetterli, University of California, Berkeley, IEEE International Conference on Image Processing. September, 1996. Lausanne, Switzerland

[Bakre95]: I-TCP: indirect TCP for mobile hosts, Ajay Bakre, B. R. Badrinath, 15th Int. Conf. on Distributed Computing Systems, (ICDCS), May 1995, DCS-TR-314 and WINLAB TR-89, October 1994

[Belzer]: “Adaptive Video Coding for Mobile Wireless Networks”, Benjamin Belzer, Judy Liao, John D. Villasenor, Electrical Engineering Department, University of California, Los Angeles

[Bolot94]: J-C. Bolot, T. Turletti, I. Wakeman, "Scaleable feedback control for multicast video distribution in the Internet", in Proceedings of ACM/SIGCOMM 94, Vol. 24, No 4, Oct. 1994, pp. 58-67.

[Bolot96]: “Control mechanisms for packet audio in the Internet» J-C. Bolot, A. Vega Garcia», in Proceedings of IEEE Infocom 96, San Francisco, CA, pp. 232-239, April 1996.

[Deering91] : “ Multicast routing in a Datagram Internetwork“, Stephen E. Deering, Ph.D. Thesis, Stanford University, Dec. 91

[Fall 95]: "Workstation video playback performance with competitive process load", Kevin Fall, Pasquale and Steven McCanne, NOSSDAV 95, p 197

[FPweb]: The FreePhone web page, http://www.inria.fr/rodeo/fphone/

[Horn]: “Scaleable Video Coding for Multimedia Applications and Robust Transmission over Wireless Channels”, Uwe Horn, Bernd Girod and Ben Belzer, Telecommunication Institute, University of Erlangen-Nuremberg, Germany and Electrical Engineering Department, University of California, Los Angeles, United States

[Huitema96]: The case for packet level FEC, Christian Huitema, Fifth International Workshop on Protocols for High-Speed Networks, Sophia Antipolis, France, October 96

[IGMP]: "Internet Group Management Protocol, Version 2", W. Fenner, <draft-ietf-idmr-igmp-v2-04.txt>, IETF draft

[McCanne95]: "Vic a flexible framework for packet video", Steven McCanne, Van Jacobson, in Proceedings of ACM Multimedia ’95, San Francisco, CA, Nov 95, pp. 511-522

[McCanne96]: "Receiver-driven Layered Multicast", Steven McCanne, Van Jacobson, ACM SIGCOMM 96 proceedings, Computer Communication Review, vol. 26, no. 4

[Normand96]: Controlled Redundancy for Image Coding and High-speed Transmission, N. Normand, J-P Guedon, O. Phillipe and D. Darba, Proceedings of the S.P.I.E. Vol. 2727, 1996

[rfc1889] : Request For Comment 1889, RTP : a Transport Protocol for Real-time Applications, H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson

[Rweb] : The Rat web page, http://www-mice.cs.ucl.ac.uk/mice/rat/info.html

[Redmill93]: Improving the Error Resilience of Entropy Coded Video Signals, D. W. Redmill and N. G. Kingsbury, Proceedings of the International Conference on Image Processing: Theory and Applications, San Remo, June 1993

[RVweb]: The Rendez-Vous web page, http://www.inria.fr/rodeo/rv/

[Sadka]: “Source and Channel Coding for Mobile Multimedia Communications”, A.H Sadka, F. Eryurtlu, A.M. Kondoz, Center for Satellite Engineering Research, University of Surrey, Guilford, Surrey

[Turletti94A]: “Issues with Multicast Video Distribution in Heterogeneous Packet Networks”, in Proceedings of Packet Video Workshop 94, Thierry Turletti, Jean-Chrysostome Bolot, INRIA

[Turletti94B]: T. Turletti, "The INRIA Video conf