Media communications over IP networks – An error correction scheme for IPTV environment

LÁSZLÓ LOIS, ÁKOS SEBESTYÉN

Budapest University of Technology and Economics, Department of Telecommunications
{lois, sebestyen}@hit.bme.hu

Keywords: IPTV, webTV, quality of service, error correction

Video transmission over IP networks has been gaining more and more popularity recently. One of the crucial problems of video transmission over IP networks through unreliable links is the susceptibility to errors in the transmission path.
Packets lost or discarded by the NIC due to CRC errors must be somehow regenerated. Regeneration can be done by requesting a retransmission, or the packet can be recalculated provided that some redundancy is introduced in the transmitter side.
After a general description of media transmission over IP links, the paper describes a method that can be used for forward error correction in IPTV applications.

1. Introduction

For seamless playback of digital audio and video contact frames must arrive in the decoder at the pace of the frame frequency. Moreover, as the misalignment of audio and video produces degradation of the perceived quality, synchronization between audio and video must be maintained. In an IP environment, however, audio and video frames are transmitted in packets. These packets travel independently within the network thus suffering from timing and alignment problems. It is the decoder that, through buffering operations, tries to resolve timing inaccuracies and presents accurately synchronised content to the viewer.

Another important issue is the error prone transmission of information. The IP network provides only very basic error protection, which, in some cases, is not adequate to keep up the quality of the service. Hence, for certain services more robust forward error correction and quality control schemes need to be introduced.

The paper is organised around the aforementioned two topics and is structured as follows: Section 2 gives an overview on the architecture of the different IP applications and reveals their timing requirements. The next Section deals with the most dynamically evolving application area, IPTV, and focuses mainly on the different error correction schemes that can be used in an IPTV environment. Section 4 presents an implemented approach based on Reed-Solomon encoding and erasure decoding [1] for reducing the number of retransmission requests together with some measurement data. Finally, Section 5 concludes our paper with some possible utilization of the implemented forward error correction scheme.

2. Media communication applications

2.1 Architecture [2]

The quality of the service offered by a media communication application is determined by the following factors:

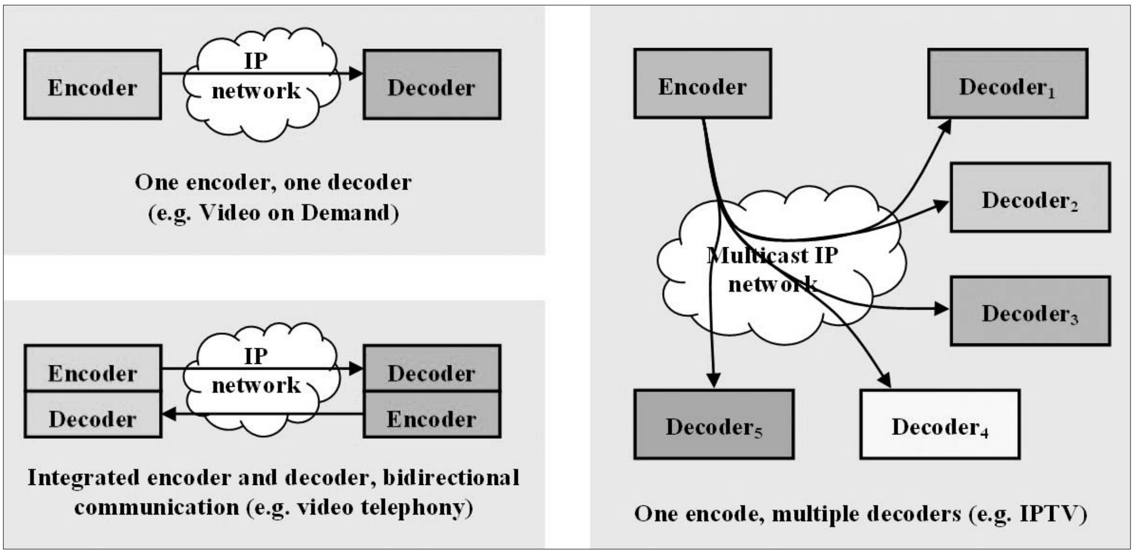


Figure 1. Basic layout of IP applications

- Quality requirements as set forth by the consumer and the service provider
- User terminal equipments
- Devices and media formats used by the service provider
- Quality of service parameters of the network or the interlinked heterogeneous networks

The basic building blocks of a media communication network are the encoder and the decoder. The encoder is responsible for converting the video, audio and data content to a format that can be transmitted through the network infrastructure, whereas the task of the decoder is to process such video, audio, and data content, restore their synchronism, and present them to the viewer.

Depending on the number and layout of the encoders and decoders, several basic network structures are possible *(Figure 1)*. The most simple media communication application consists of a single encoder (server) and a single decoder (client) which together make up a very basic peer-to-peer video on demand system (Fig. 1, top left). If both peers possess some encoding and decoding capabilities, they both can act simultaneously as a server and a client. Hence, encoded information can travel in both directions, and a video telephony system can be set up (Fig. 1, bottom left). The same architecture allows for conferencing services provided that more peers are allowed to join the telephony session.

A more sophisticated approach is when the IP infrastructure supports multicast transmission (Fig. 1, right hand side). In a multicast session, a single encoder supplies multiple decoders with a common output stream. The packets of this stream are duplicated (multiplied) and routed by special network components thus reducing the burden on the network.

The communication between the server and the client follows a hierarchical approach, in which each layer has its own task as depicted in *Figure 2*.

2.2 Timing accuracy in media communication

Media communications can also be characterised by the timing accuracy. The quality of timing accuracy is basically determined by the seamlessness of playback and the delay introduced by transmission and processing. Based on these quality factors three schemes can be distinguished. The properties of each are summarised in *Table 1*.

Regardless of the scheme used, the primary aim of transmission is to supply the user with as high an image and sound quality as can be ensured by the particular network in a reasonable time. A key point in that is choosing an appropriate buffering strategy which not only decreases jitter and ensures seamless playback, but also allows for either the possibility of retransmission or the correction of packet by FEC and interleaving.

An offline service is usually implemented by TCP/IP or HTTP protocols and is intended for downloading content to a temporary or final storage. In an offline service, the terminal equipment or a neighbouring network component has a storage capacity to store the whole content. The primary aim is safe delivery of content, transmission delay is only a secondary factor if relevant at all.

An online service such as a video telephony application tries to minimize the annoying delay of transmission and presentation. As any buffering operations increase the delay, the size of the transmitter and receiver buffer is usually limited to a couple of frames. As a result, there is no time to interleave the content in the transmitter side and perform FEC encoding (as that would also require buffering), or to request the resending of information in the receiver side. Because re-sending is not possible, unreliable protocols, such as UDP [3], are preferred.

Most of the current streaming services belong to the near-line scheme. Due to the lack of interactivity, basic streaming applications can tolerate longer delay. Buf-

Figure 2. Functional model of the media encoder and decoder

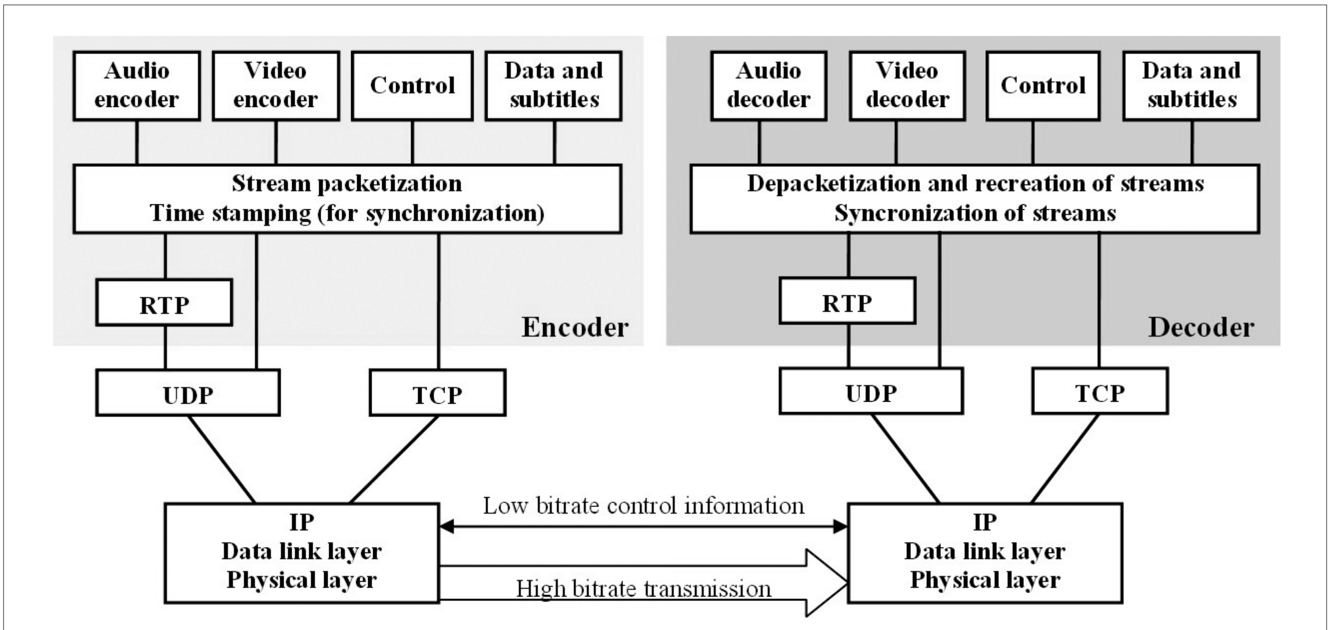


Table 1. Characterization of media applications by timing accuracy

	Offline scheme	Near-line scheme	Online scheme
Primary aim	Full and error free download	Seamless playback	Immediate playback with minimal delay
Seamlessness	Irrelevant	Yes	Yes
Presentation of each frame after reception	Irrelevant	After buffering delay	Immediately after reception
Buffering	Irrelevant	To ensure error correction and seamless playback	Minimal
Replacement of lost packets	Error correction or packet resending	Error correction or packet resending if allowed for by buffering	No resending of packets, only minor error correction
Typical application	Media file download	Media streaming	Video telephony, video conference

fering for a couple of seconds is common to streaming applications. The near-line scheme usually uses unreliable but more complex protocols to transmit information. In most cases a control channel is used through which control and link state information can be exchanged. A typical near-line service using real-time transport protocol (RTP [4]) is depicted in *Figure 3*.

The audio, video, or data sub-streams (depicted by a simple UDP block in Fig. 3) are either treated separately or combined to form a multiplexed stream. While in the former case separate UDP sessions and control channels are created for each individual sub-stream, in the later case the sub-streams are multiplexed and only one aggregate control channel is set up.

3. Streaming services, IPTV

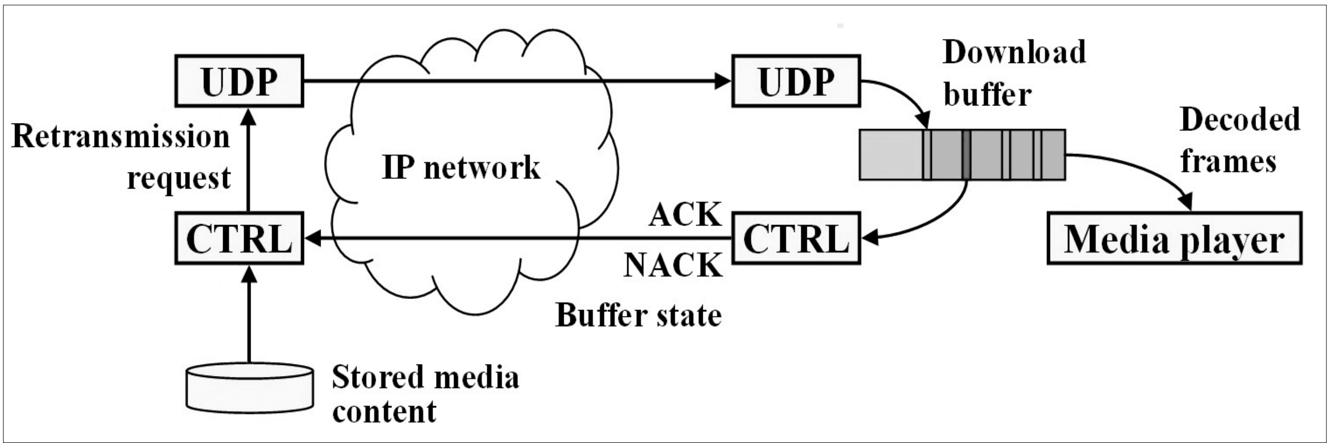
Streaming services usually use UDP/IP or RTP protocols. In a streaming environment not only error free transmission but also timing accuracy is a concern. A delay of some seconds is tolerable at the start of playback, but once started, playback must be uninterrupted and seamless. Although a short delay is allowed, it has to be minimized as much as possible. Since the delay introduced by the UDP protocol is minimal, it is considered to be a favourable choice.

Typical streaming services include applications like television through the Internet (webTV) and Internet Protocol Television (IPTV) services wich have been gaining more and more popularity recently. The difference between the two is while webTV is implemented on the public Internet, IPTV uses an IP infrastructure which is a private and closed system maintained by a service provider. It is this service provider who is responsible for providing the content, managing user access through access control mechanisms, and, in many cases, supplying the users with terminal equipments (so-called set-top-boxes). As the network is closed, the service provider has freedom to prioritise packets carrying television streams over other packets. (This is certainly not possible in webTV applications.)

IPTV is often bundled with other services like Voice over IP (VoIP) telephony, and supplementary Internet access. These three together form a service often referred to as TriplePlay.

The IPTV network is a switched digital video (SDV) architecture, in which only streams requested by the viewers are present. Programs that are not being viewed do not appear on the network or on the network segment. This approach, on the one hand, is very economic in the sense that it saves valuable bandwidth for the service provider which can be used to provide value-added ser-

Figure 3. A typical near-line streaming service based upon the RTP protocol



vices like Video on Demand (VoD). On the other hand, it allows for building an access network with only as much bandwidth as required by the subscription of each user. Therefore, the bandwidth of the access network for each user is determined by the number of programs that can be simultaneously accessed by them. If a user has access to only one standard definition (SD), MPEG4-AVC [5] encoded stream at a time, then the data rate can be as low as 2.4 Mbps. (Although 2.4 Mbps is considered to be a minimal data rate for standard definition AVC, due to technological reasons this is often reduced to 2 Mbps.)

The most typical consumer behaviour in an IPTV environment is watching live television streams. Since every program is watched by many users, the streams are transmitted in multicast mode. In a multicast session the network and the additional architecture take care of both packet multiplication and multicast group management.

If a program change is requested by the user, the user is moved from their previous multicast group corresponding to the program they were watching to a new multicast group corresponding to the newly requested program. Since multicast group switching is a slow process, and an additional buffering delay is introduced by the near-line scheme, a common method is to supply the user with packets of the newly requested stream through a VoD-like unicast connection of higher data rate. As soon as the multicast group switching is complete and the receiver buffer is full, the terminal equipment can switch to the normal multicast data stream and continue receiving that.

3.1 Error correction in IPTV networks

As far as the network and the data link layers are concerned, IP transmission features no forward error correction. The only error resiliency method used in a normal IP environment is the insertion of a CRC code into the packet that can be used for error detection. If a packet fails the CRC error check in the receiver, then it is discarded. How the system behaves in the case of packet loss is essential from the point of view of the service.

3.1.1 Retransmission of lost or erroneous packets

Packets can be lost due to congestion in the network, or they can be discarded by the network layer due to an invalid CRC code. Either way, missing packets of multimedia services, if not recovered, produce visual artefacts.

To reduce the quality degradation, missing packets can be recovered by requesting their retransmission. Retransmission makes sense only if the round trip time T_{RTT} that consists of the time needed to send the retransmission request plus the time the retransmitted packet arrives is less than the time T_{buf} the packet spends in the buffer till it is either presented or used as a reference to decode other frames:

$$T_{RTT} \leq T_{buf}$$

To satisfy the above condition the following approaches can be followed:

- RTT must be reduced by placing the server as close to the clients as possible. An IPTV network usually features one server at a predefined location and cannot be freely relocated. The problem of relocation, however, can be overcome by installing so-called secondary caching servers. The caching servers store a well defined portion of the streams that pass through them. Upon a retransmission request from a client, they are ready to resend the packet through a unicast connection.
- The buffering time T_{buf} must be increased. Since in an IPTV environment it means that the STB must be equipped with more memory, it is usually not a plausible approach.

3.1.2 Replacement of lost or erroneous packets

Missing packets can be replaced if enough redundancy is introduced in the system, and this redundancy can be used to regenerate the packets that have been lost. The redundancy information can either be inserted in the packets themselves, or it can be transmitted as a separate correction stream. As in the second approach the redundancy information can be discarded by receivers which do not need or do not support them, and there is no need to recalculate the CRC code of the original packets, this is a more plausible method.

The information can be recovered by utilising appropriate encoding (like systematic Reed-Solomon encoding) provided that the number of erroneous symbols remains below a well defined upper bound. The method for such encoding is the following:

(1) The transmitter appends N-K parity symbols to the K source symbols.
(2) The K source symbols and the appended N-K parity symbols together form the encoded word having a data length of N symbols.
(3) Out of the N encoded symbols E symbols are corrupted during transmission and N-E remain intact.
(4) The receiver receives the N encoded symbols, and recovers the E erroneous ones provided that:

$$2 \cdot E \leq N - K$$

if the error locations are unknown (normal error correction), or:

$$E \leq N - K$$

if the error locations are known (erasure error correction).

Both normal and erasure error correction means a trade-off between the channel capacity and error correction capability. To be able to correct E symbols out of the N received ones, $2E$ or E out of the N symbols must be parity information. A well-known systematic FEC encoding scheme that satisfies the above conditions and can be implemented quite easily is the Reed-Solomon encoding. In a test bed which is described in the next section, Reed-Solomon encoding and erasure decoding was used to provide a means to regenerate missing data.

4. Error correction as implemented in a real-world IPTV testbed

The method described in the previous section cannot be applied directly to the Ethernet packets, as that would imply an enormous calculation burden on the system. To reduce calculation complexity a so-called interleaving approach is followed, which is depicted in *Figure 4*.

4.1 Encoder

The encoder reserves two memory areas referred to as Network Data Table and RS Data Table. Each position within the reserved memory areas can hold one byte of information. The Network Data Table consists of *K* columns and 1346 rows, and is used to store incoming packets that need to be Reed-Solomon encoded. The RS Data Table has *N-K* columns and holds the parity information, a generated header and some additional information.

The Reed-Solomon encoder works above GF(8), therefore the upper bound for *N* is 255. K can be any arbitrary odd number between 1 and *N*. The difference *N-K* determines the error correction capability of the erasure Reed-Solomon decoder as seen in Section 3.1.2.

First, a copy of the Ethernet packets arriving at the encoder is buffered in the Network Data Table in a column-wise direction. If the packet length is less than the maximum packet length of 1340 bytes (corresponding to seven 188-byte transport stream packet plus the RTP header), the empty positions within the respective column are zero padded.

Once all *K* columns in the Network Data Table are completely filled, a 6-byte supplementary information consisting of the packet length and a CRC code is appended to the end of each column. Since this supplementary information can be regenerated in the receiver side provided that the packet it corresponds to does arrive, this is information is not transmitted.

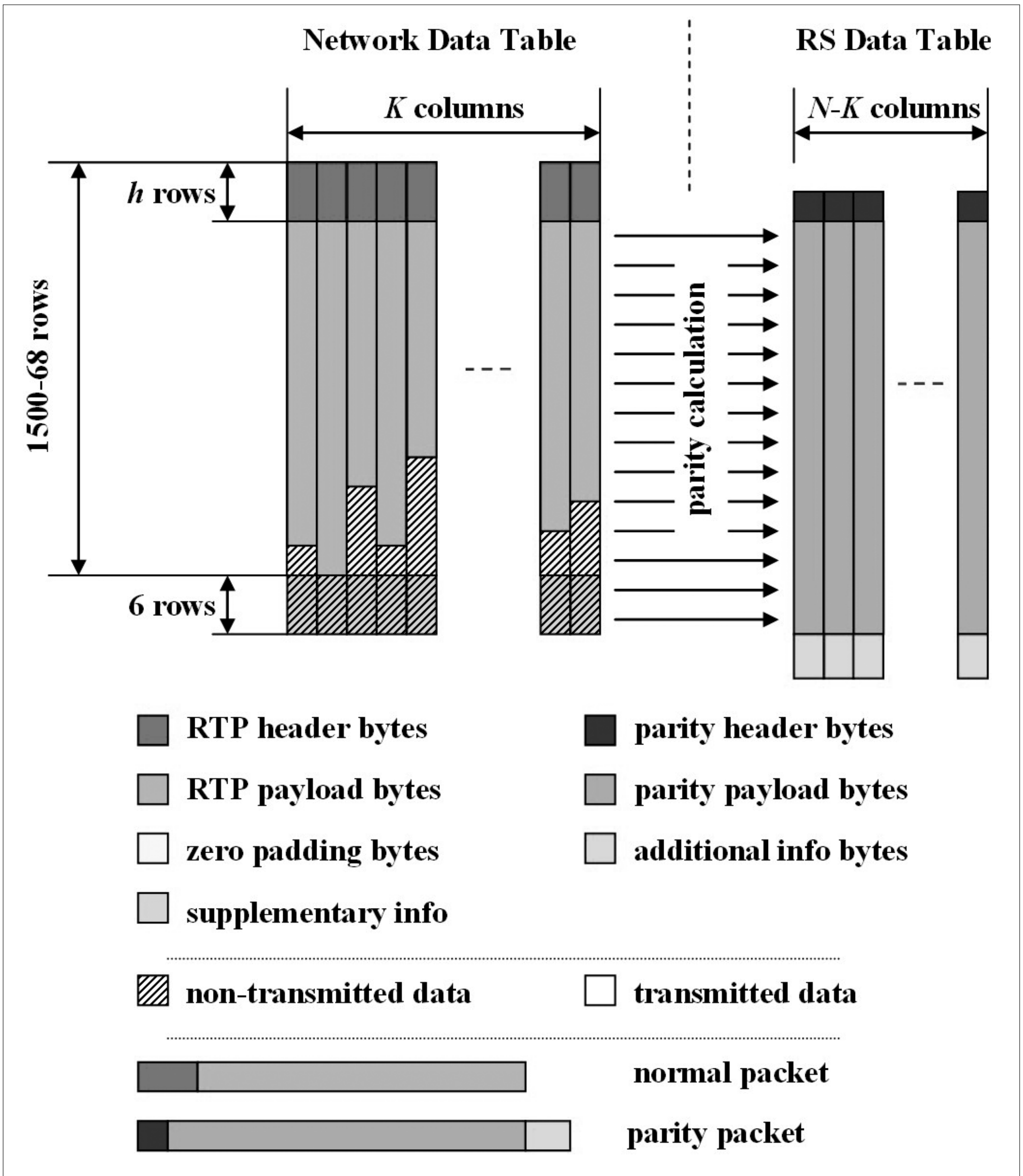


Figure 4. Network Data Table and RS Data Table used for interleaved encoding and decoding

In the next step, the useful payload of the packets (excluding the header), plus the zero paddings (if any) and the supplementary information are Reed-Solomon encoded row wise. The calculated parity information is stored in the appropriate row of the RS Data Table.

After the parity information is calculated for all rows, a parity header is generated for each column and some additional information is recorded in the RS Data Table. The header contains the source and destination addresses and ports as well as the position of the column within the RS Data Table. The additional information includes the sequence numbers of the RS encoded RTP packets, their positions within the Network Data Table plus parity information calculated from the length and CRC of the incoming RTP packets. These data are used in the receiver side for reconstructing the Network Data Table, spotting any missing packets, and for validating the data after reconstruction. Since the sequence numbers and the positions are crucial for the correct operation of the system, they are repeated in every column.

Finally the columns of the RS Data Table are transmitted as user packets.

4.2 Decoder

The decoder works in a similar fashion. It first restores the original order of both the normal and the parity packets, and saves the packets in the Network Data Table and RS Data Table respectively. If a packet, either normal or parity, is found to be missing, then all positions in the respective column are marked as erasures. Once all packets are accounted for (either inserted or marked as erased), the payload of the missing packets can be regenerated by performing erasure RS decoding row wise. After the payload is restored, the RTP header can easily be regenerated, thus all the missing information is regained.

4.3 System architecture and performance

The architecture of the testbed that uses interleaved RS encoding for IPTV streams is depicted in *Figure 5*. Both RS encoding and RS decoding of a predetermined stream were performed by computers with two NIC cards working transparently in a bridged configuration. The appropriate packets were filtered and passed to a user program which then performed RS encoding and RS decoding as described in Section 4. The packet loss in the network was modelled by a uniform distribution. The expected value of the ratio of lost packets and the parameters of the RS forward error correction could be freely chosen. The network traffic after RS decoding was monitored by a measurement device.

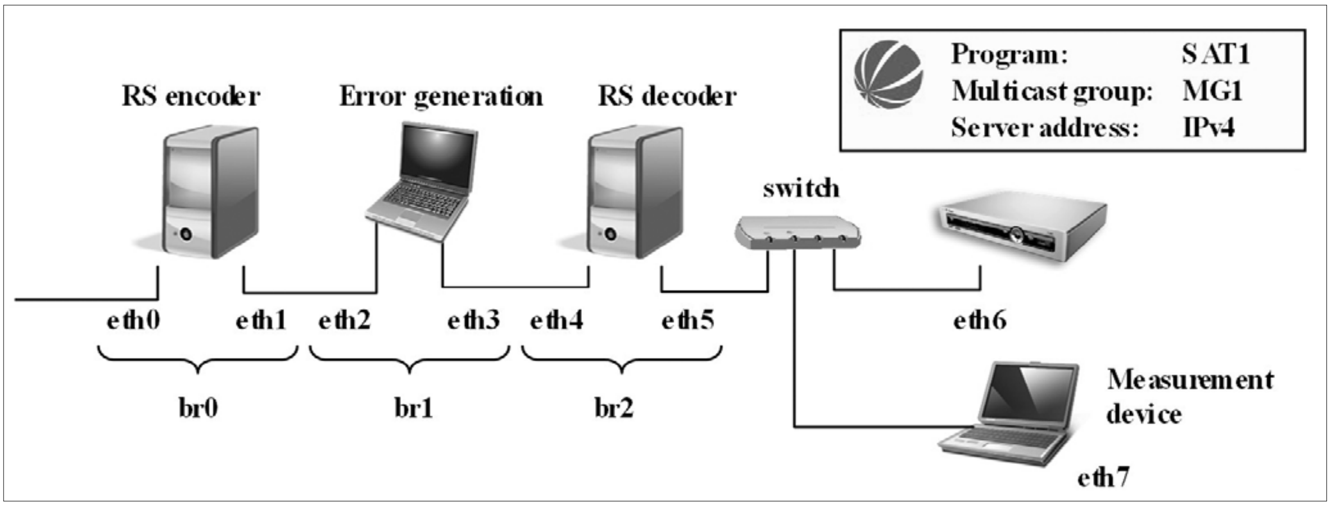
The measurement results for a packet loss ratio of 10% for different RS parameters are summarized in *Table 2*. (The packet loss ratio of 10% means that 10 percent of the packets after RS encoding is dropped by the error generator device.)

What is apparent from the figures (apart from the fact that packets with longer parity are more likely to be corrected) is that if the increase in bandwidth (i.e. the code rate) is kept constant, then the ratio of lost packets after RS decoding is decreased with the increase in message length. This agrees with our expectations as the longer the RS codeword, the less probable is that the number of errors per RS codeword will considerably exceed the expected value of the distribution, hence it is more likely that they can be corrected. In case of a codeword length of around 255 and uniform distribution, all packets could be corrected provided that the increase in bandwidth was twice the packet loss ration after RS encoding.

5. Conclusions and future work

In the paper we gave a general description of the two most severe problems of IPTV networks, and presented an approach that can be used for error correction in an IPTV environment. Although the method means an increase in the overall data rate for an IPTV stream (an increase inversely proportional to the code rate), it can be effectively used to decrease the unicast traffic thus making the caching servers obsolete.

Figure 5. The measurement setup



Message length (*K*)	Code word length (*N*)	Code rate	Increase in bandwidth	Ratio of lost packets after RS decoding
10	12	83.33%	20.0%	11.20%
10	14	71.43%	40.0%	0.90%
10	16	62.50%	60.0%	0.00%
20	22	90.91%	10.0%	37.70%
20	24	83.33%	20.0%	8.50%
20	26	76.92%	30.0%	1.20%
20	28	71.43%	40.0%	0.10%
20	30	66.67%	50.0%	0.00%
40	44	90.91%	10.0%	45.00%
40	46	86.96%	15.0%	17.20%
40	48	83.33%	20.0%	4.80%
40	50	80.00%	25.0%	1.00%
40	52	76.92%	30.0%	0.20%
40	54	74.07%	35.0%	0.00%
40	56	71.43%	40.0%	0.00%
40	58	68.97%	45.0%	0.00%

Table 2. Measurement results for a packet loss ration of 10% after RS encoding for different coding parameters

As far as the future enhancements are concerned, the introduced error correction system can be expanded to use low density parity check codes instead of RS encoding.

Authors

LÁSZLÓ LOIS was born in 1971 in Tatabánya. He received his Masters and Ph.D. degree in software engineering from the Budapest University of Technology and Economics (BME) in 1995 and 2005, respectively. He is currently with the Department of Telecommunications at BME, and contributes to the research, development and educational activities. His research interests include source coding applications, media communications and IP television.

ÁKOS SEBESTYÉN was born in 1977 in Budapest. He received his Masters Degree in electrical engineering at the Budapest University of Technology and Economics, Department of Telecommunications in 2002. Now as a member of the staff of the Department, he contributes to the educational and research and development activities. His research areas include digital broadcasting (DVB-T/S/C/H/T2) and IP television.

References

[1] Didier F., "Efficient erasure decoding of Reed-Solomon codes" http://arxiv.org/pdf/0901.1886v1

[2] ITU-T H.264: Advanced video coding for generic audiovisual services, Series H: Audiovisual and multimedia systems, Infrastructure of audiovisual services – Coding of moving video.

[3] RFC 768 – User Datagram Protocol.

[4] RFC1889 – RTP: A Transport Protocol for Real-Time Applications.

[5] ISO/IEC 14496-10:2005: Information technology – Coding of audio-visual objects Part 10: Advanced Video Coding.