RECOMMENDATION ITU-R BT.1789

A method to reconstruct received video using transmission error information for packet video transmission

(Questions ITU-R 44/6 and ITU-R 109/6)

(2007)

Scope

This Recommendation specifies a method for a service provider to reconstruct received video in order to monitor video quality at a receiver using transmission error information for packet video transmission. This Recommendation applies to video services where two-way digital communications are available.

The ITU Radiocommunication Assembly,

considering

a) that the traditional evaluation of video quality has been performed subjectively by a number of evaluators;

b) that, although the subjective test is considered to be the most accurate method, it has many limitations since it is time-consuming and expensive;

c) that it is desirable that the service provider monitors video quality at the receiver;

d) that some objective methods for video quality measurement require additional bandwidth for transmitting parameters;

e) that bandwidth is a valuable and expensive resource in many multimedia services;

f) that the communication paths of most multimedia applications will be completely digital;

g) that transmission errors and their effects on received video are readily identified when video data is transmitted in packets;

h) that a certain type of receiver is able to detect the occurrence of transmission errors;

j) that the receiver is able to send such transmission error information to the head-end[1] in some multimedia applications with return channel capability,

noting

a) that it is possible for the head-end to efficiently monitor the received video quality by using the reconstructed video sequence, together with other available information including the source sequence for the packet video transmission,

recommends

1 that the method described in Annex 1 should be used for the head-end to reconstruct the video seen at any receiver, in order to monitor the video quality at that receiver.

---

[1] The head-end includes a transmitter, a received video estimation unit and a video quality estimation unit, in accordance with Recommendation ITU-R BT.1683. It may also include an encoder.

Annex 1

1 Introduction

Objective video quality measurement methods are classified into three categories: full-reference (FR) models, reduced-reference (RR) models, and no-reference models. Generally, the accuracy of no-reference models is inferior to that of the FR and RR models. Figure 1 shows a block diagram of full-reference models and Fig. 2 shows a block-diagram of reduced-reference models. The full-reference model, which takes two input video sequences (source video sequence and processed video sequence), produces a video quality metric (VQM) of the processed video sequence.

As can be seen, both the FR and RR models require source video and processed video sequences for video quality assessment. On the other hand, for some broadcasting services, monitoring of received video quality is important. If a FR model is to be used, the source video sequence should be available at the receiver (Fig. 3) or the processed video sequence (impaired video) should be available at the head-end (Fig. 4). According to this Recommendation, the received video quality, as seen at the receiver, may be evaluated at the head-end. This requires the source video sequence, or features extracted from the source video sequence, to be available to the head-end. If an RR model is to be used, the features extracted from the source video sequence are required to be available at the receiver (Fig. 5) or the features extracted from the processed video sequence (impaired video) are required to be available at the head-end (Fig. 6). Since bandwidth is a valuable and expensive resource in many multimedia applications, it is desirable to avoid transmission of this additional data.

FIGURE 1

A full-reference model

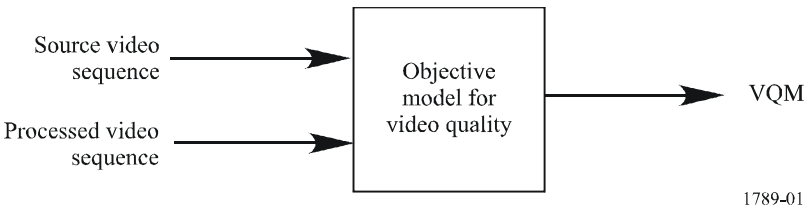


FIGURE 2

A reduced-reference model

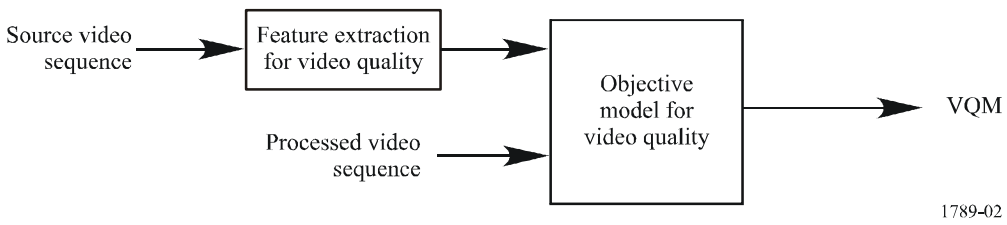


FIGURE 3

A block diagram for the receiver computing the video quality of the received video using an FR model

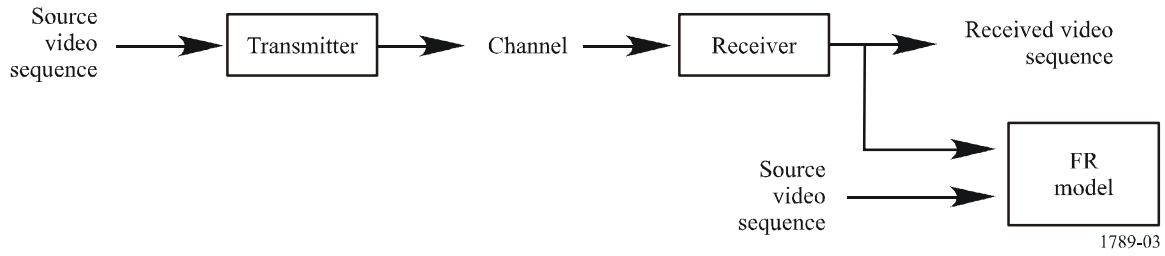


FIGURE 4

A block diagram for the head-end computing the video quality of the received video using an FR model

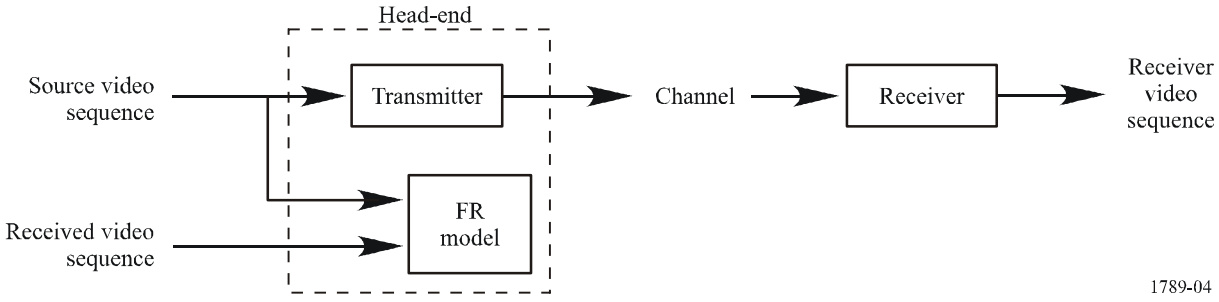


FIGURE 5

A block diagram for the receiver computing the video quality of the received video using an RR model

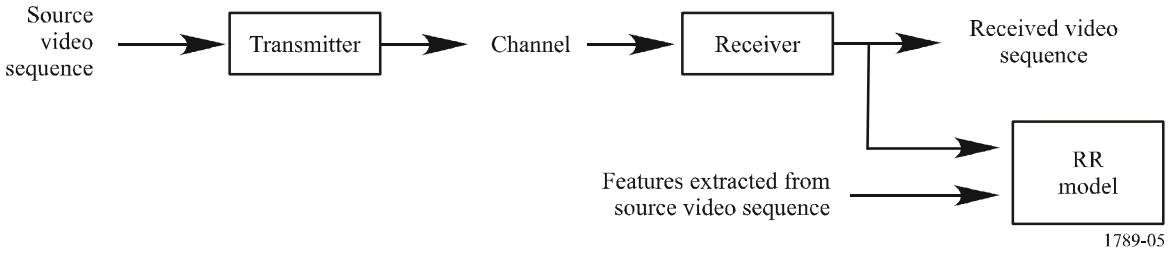
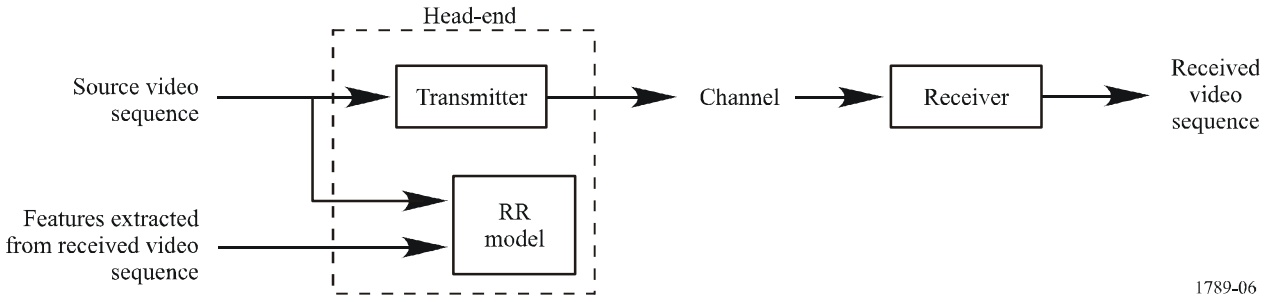


FIGURE 6

A block diagram for the head-end computing the video quality of the received video using an RR model



However, in some multimedia applications, video data is transmitted in packets. During transmission, various errors might occur, which include packet loss, overflow, underflow and delay. These errors can produce frame freezing, frame skipping, block errors, jitter, delay, etc. in the received video. In digital communications, all these transmission errors and their effects can be exactly identified when video data is transmitted in packets. Furthermore, in digital video transmission, if there is no transmission error, the received video quality will be identical to the transmitted video quality.

Therefore, if the receiver sends transmission error information, which includes information on packet loss and delay in packet video transmission, back to the head-end, the head-end can exactly reconstruct the received video as seen at the receiver.

It is necessary that the service provider and the receiver interwork in order for the receiver to provide all the necessary information to the service provider. In other words, all information on the decoder and post-processing techniques used in the receiver must be available to the service provider for the service provider to exactly duplicate the video sequence at the receiver. With this information, the method can be used with any codec and communication channels, including internet and wireless communications. Since video quality assessment is performed at the service provider, where the video source is available, it is possible to use any model which includes full-reference and reduced-reference models.

1.1 Application

This Recommendation provides a method to reconstruct received video for video quality monitoring for video services where return channels are available when video data is transmitted in packets. The applications for the method described in this Recommendation include, but are not limited to:

– monitoring of received video quality, as seen at the receiver, with minimum consumption of additional bandwidth;

– real-time received video quality monitoring at the head-end.

1.2 Limitations

The method presented in this Recommendation describes a procedure to reconstruct video sequences as seen at the receiver, using transmission error information and transmitted packet video data. The method in this Recommendation requires that each packet can be traced and identified. Some packet transport protocols such as RTP (real-time transport protocol) and ATM (asynchronous transfer mode)/AAL (ATM adaptation layer) have this feature. The method also requires a return channel so that the receiver can send transmission error information to the service provider. In order to evaluate video quality at the receiver, the method needs to be used with an objective model for video quality measurement. It is suggested that a standardized objective model for video quality measurement method be used.

2 The method

Figure 7 illustrates the procedure. The head-end transmits packet video data to the receiver. It is noted that the source video is first encoded and then the compressed video data is arranged into packets. The receiver has a transmission error detection unit which detects the occurrence of transmission errors. If transmission errors occur, the transmission error detection unit sends the transmission error information, which includes packet loss and delay along with their effects such as frame freezing, frame skipping, block errors, jitter, etc., back to the head-end. Table 1 shows typical transmission error information. Then, the received video estimation unit in the head-end emulates the receiver and estimates the received video as seen at the receiver, using the

transmission error information and the packet video data produced by the encoder. Finally, a video quality evaluation unit computes video quality scores at the receiver using the source video and the estimated received video. Figure 8 shows an example of the method when an FR model is used. The estimated received video in Fig. 8 is produced by the received video estimation unit (Fig. 7). In cases where the source videos are not available at the head-end (service provider), it is also possible for the head-end to use an RR model provided that feature parameters are available.

In packet video transmission, the effects of transmission errors can be described as follows:

– video degradation due to packet loss;

– lost frames due to packet loss, delay, overflow and underflow;

– delayed frames due to transmission errors.

Therefore, if the receiver sends information on lost or impaired packets, lost or skipped frames and delayed frames to the head-end, the head-end can reconstruct the received video as seen at the receiver.

FIGURE 7

A method for a head-end to monitor video quality at a receiver using transmission error information

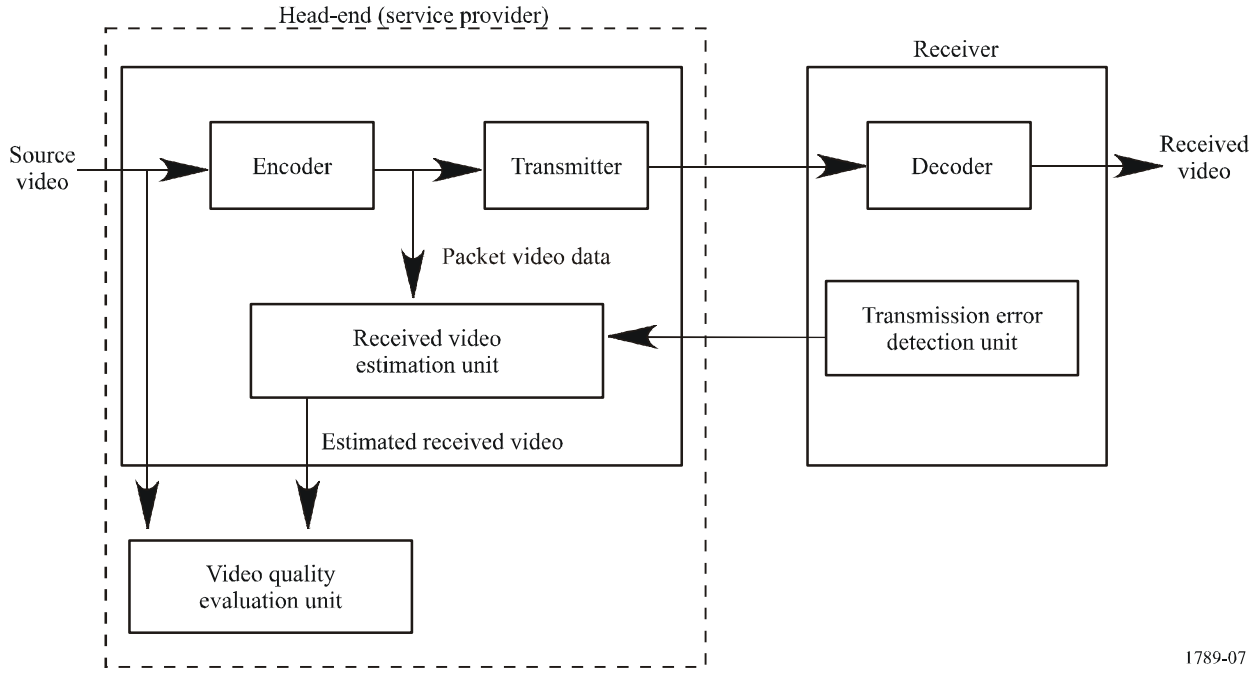


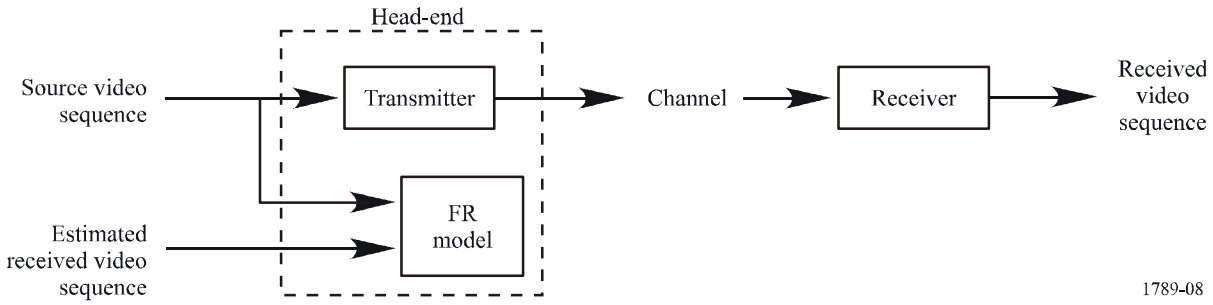
TABLE 1

Transmission error information

Type of transmission errors	Contents of transmission information
Information on lost or impaired packets	Corresponding packet indexes
Information on delayed frames	Amount of delayed time and indexes of delayed frames
Information on skipped or lost frames	Skipped or lost frame indexes

FIGURE 8

A block diagram for the head-end (service provider) computing the video quality of the received video using the estimated received video (FR model)



3 Messages for transmitting transmission error information

In this method, the head-end (service provider) and the receiver interwork to ensure that the receiver provides the necessary transmission error information to the service provider. It is also noted that all information on the decoder and post-processing techniques used in the receiver must also be provided to enable the service provider to exactly estimate the video sequence at the receiver.

In order to estimate the received video sequence at the head-end, the required information on transmission errors is summarized in Table 1. For each type of transmission error, a message is transmitted. Such messages consist of two or three fields: type and binary numbers. A number of messages can be combined and then transmitted.

3.1 Messages for decoder information (receiver model information)

In order to exactly estimate the received video sequence, the head-end needs information on the decoder and post-processing techniques used at the receiver. For this purpose, at the beginning of transmission, the receiver needs to transmit a model identification message. It is assumed that the head-end has a database and can obtain all the necessary information on the decoder and post-processing techniques of the receiver from the model identification message.

3.2 Source identifier

In broadcasting and multicast environments, when the head-end receives transmission error messages, it needs to identify the corresponding source video. For this purpose, the receiver needs to transmit a source identification message. The source information is available in packets.

3.3 Messages for lost packets

For a lost packet, a lost packet index needs to be transmitted.

When burst errors occur, a number of consequent packets are lost. In this case, a starting packet index and an ending packet index of the lost packets need to be transmitted.

3.4 Messages for delayed frames

For a delayed frame, a delayed frame index and the amount of time delayed need to be transmitted.

3.5 Messages for skipped frames

For a skipped (lost) frame, a skipped frame index needs be transmitted.

When burst errors occur, a number of consequent frames may be lost. In this case, a starting frame index and an ending frame index of the skipped frames need to be transmitted.

3.6 Hand shaking and error handling

Due to transmission errors, these messages can also be lost or corrupted. On the other hand, most two-way communication systems employ some error detection and handling mechanisms, which can be used to ensure the delivery of the messages. The error messages can be transmitted in real-time or may be transmitted in a batch-mode.

Table 2 summarizes the error message description. Figure 9 illustrates the received video estimation unit. Examples of error message formats are provided in Appendix 1.

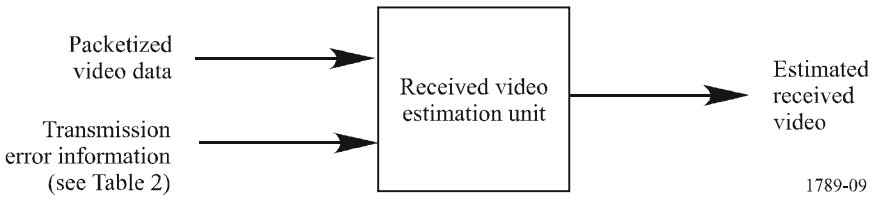
TABLE 2

Message description

Type of transmission errors	Message descriptions
Information on the receiver	A model identification message
Source identifier	A source identification message
Information on a lost packet	A lost packet index
Information on lost packets	A starting packet index and an ending packet index of the lost packets
Information on a delayed frame	A delayed frame index and the amount of delayed time
Information on a skipped frame	A skipped frame index
Information on skipped frames	A starting frame index and an ending frame index of the skipped frames

FIGURE 9

Inputs and output of the received video estimation unit



Appendix 1

In this Appendix, an example of a message format capable of sending information on transmission errors is described.

1 Messages for decoder information (receiver model information)

A model identification message can be transmitted using a 32-byte message. The first byte is the ASCII code of character “m” (6D in hexadecimal) representing a model identification. The following 31 bytes are a character string terminated by a null character. For example, if the model number of the terminal is “ABC-1234”, the following message is transmitted:

6D 41 42 43 2D 31 32 33 34 (“mABC-1234”) followed by 23 null characters.

2 Source identifier

A source identification message can be transmitted using five bytes of binary data at the beginning of transmission. The first byte is the ASCII code of character “i” (69 in hexadecimal) representing a source identifier. The other four bytes are used for source identification:

69 XX XX XX XX (hexadecimal).

3 Messages for lost packets

A lost packet index can be transmitted using five bytes of binary data. The first byte is the ASCII code of character “l” (6C in hexadecimal) representing a lost packet. The other four bytes are a long integer (four bytes) representing the lost packet index. For instance, if the 100th packet is lost, the following message is transmitted:

6C 64 00 00 00 (hexadecimal)

where the first byte is the least significant byte in the four-byte long integer (unsigned).

When burst errors occur, a number of consequent packets are lost. In this case, a starting packet index and an ending packet index can be transmitted using nine bytes of binary data. The first byte is the ASCII code of character “L” (4C in hexadecimal). The next four bytes are a long integer (four bytes) representing the starting index of the lost packets. The last four bytes are a long integer representing the ending index of the lost packets. For instance, if the 60th - 90th packets are lost, the following message is transmitted:

4C 3C 00 00 00 5A 00 00 00 (hexadecimal)

where the first byte is the least significant byte in the four-byte long integers (unsigned).

4 Messages for delayed frames

A delayed frame index and the amount of time delayed can be transmitted using seven bytes of binary data. The first byte is the ASCII code of character “d” (64 in hexadecimal) representing a delayed frame. The next four bytes are a long integer (four bytes) representing the delayed frame index. The last two bytes are a short integer (two bytes) representing the amount of time delayed in milliseconds. For instance, if the 60th frame is delayed by 300 ms, the following message is transmitted:

64 3C 00 00 00 2C 01 (hexadecimal)

where the first bytes in the unsigned long integer and unsigned short integer represent the least significant byte.

5 Messages for skipped frames

A skipped frame index can be transmitted using five bytes of binary data. The first byte is the ASCII code of character “s” (73 in hexadecimal) representing a skipped frame. The other four bytes are a long integer (four bytes) representing the skipped frame index. For instance, if the 60th frame is lost, the following message is transmitted:

73 3C 00 00 00 (hexadecimal)

where the first byte is the least significant byte in the four-byte long integer (unsigned).

When burst errors occur, a number of consequent frames may be skipped. In this case, a starting frame index and an ending frame index of the skipped frames can be transmitted using nine bytes of binary data. The first byte is the ASCII code of character “S” (53 in hexadecimal). The next four bytes are a long integer (four bytes) representing the starting index of the skipped frames. The last four bytes are a long integer representing the ending index of the skipped frames. For instance, if the 60th - 90th frames are skipped, the following message is transmitted:

53 3C 00 00 00 5A 00 00 00 (hexadecimal),

where the first bytes in the unsigned long integer and unsigned short integer represent the least significant byte.

Table 3 summarizes the error message formats.

TABLE 3

Error message formats

Type of transmission errors	Transmission error messages in hexadecimal	Descriptions
Information on a lost packet (5 bytes)	6C XX XX XX XX	"l" + packet index in long integer
Information on lost packets (9 bytes)	4C XX XX XX XX XX XX XX XX	"L" + starting packet index in long integer + ending packet index in long integer
Information of a delayed frame (7 bytes)	64 XX XX XX XX XX XX	"d" + frame index in long integer + delay time in short integer
Information on a skipped frame (5 bytes)	73 XX XX XX XX	"s" + frame index in long integer
Information of skipped frames (9 bytes)	53 XX XX XX XX XX XX XX XX	"S" + starting frame index in long integer + ending frame index in long integer
Information of the receiver (32 bytes)	6D + 31-byte string	"m" + 31 byte string
Source identifier (5 bytes)	69 XX XX XX XX	"i" + 4 bytes (32 bits)