A Model for Error Avoidance and Error Correction in Peer-to-Peer Networks

Christian Spielvogel, Sabina Serbu, Pascal Felber, Peter Kropf
University of Neuchâtel, Switzerland
{firstname.lastname}@unine.ch

Abstract

Video streaming over best effort networks remains a challenging task. Video quality decreases with the number of frames that are corrupted, lost or received after the playback time. In order to deliver videos in high quality, a model for selecting the proper network-error treatment in a peer-to-peer overlay is presented. Based on the model, the selection is between error avoidance, error correction and a combination of both approaches. The model is evaluated by using the network simulator NS-2 and a modified version of EvalVid. The solution is presented in the context of an indexing-cache overlay, which has two interesting properties for our selection model: it efficiently locates even rare videos, and the search procedure returns multiple locations for a requested video.

1 Introduction

Delivering videos in the desired quality over best effort networks remains an important challenge. If a video is streamed from an arbitrary server to an arbitrary client, the perceived quality typically varies in an unpredictable way.

The commonly used methods for handling packet loss are Forward Error Correction (FEC) and Automatic Repeat Request (ARQ). FEC adds redundancy before transmitting the data, ARQ is used to retransmit lost packets. The problem about FEC is that redundant information requires additional bandwidth and causes even more packet losses in case of crowded links. While ARQ retransmission increases the packet delay it is not applicable to data with real-time constraints for playback. The alternative to *error correction* is *error avoidance*. Error avoidance can be achieved by using Multiple Description Coding (MDC) [7]. MDC encodes the signal into a number of separate bit streams called descriptions. Each description has roughly the same size and equally influences the media quality. Any of the descriptions is able to provide some base quality, but the more descriptions are available the better is the quality. The best quality is obviously achieved when all descriptions are available. The main drawback of Multiple Description Coding is that end-users may not always be satisfied with the quality of the stream received. This for example is the case when only one out of four descriptions arrives.

Existing error correction and avoidance approaches (e.g., [1], [2], [3] and [8]) consider either network or stream characteristics, but none of them considers both. The main contribution of this paper is to present a model that is able to select either one or a combination of both, in dependency of the coding characteristics of the stream and the probabilistic link behavior of the overlay. In the presented approach we use a peer-to-peer overlay network for locating videos. The location mechanism is based on indexing-caches that contain information about a part of the videos in the network. Once a video is found in one or more locations, we apply our model for selecting between error correction, error avoidance and a combination of both. The selection depends on the coding characteristics of the media stream and the probabilistic link behavior of the network. We have evaluated our model using the NS-2 [11] network simulator and an extended version of the EvalVid plug-in [5]. We have extended EvalVid to support the evaluation of multiple sub-streams (descriptions) that are delivered within NS-2.

The rest of the paper is structured as follows: Section 2 presents related work, Section 3 describes the peer-to-peer overlay and the efficient way we use to locate videos. Section 4 introduces the error model, followed by Sections 5 and 6 that present the evaluation and conclusions respectively.

2 Related Work

Approaches relying on error avoidance and error correction are not new. One of the first documents discussing such ideas is [10], where error avoidance is based on the principle of reducing the packet loss probability by sending parts of the data either (1) from different sources or (2) over parallel paths. One possibility for realizing error avoidance is Multiple Description Coding. In [3] it is shown that Multiple Description Coding in combination with Multiple Source Streaming is able to deliver video streams in much better quality than the classical server client approach.

Forward error correction is based on the principle of reconstructing data that has been lost during network transmission. A good overview about forward error correction can be found in [1], [2] and [8].

3 Peer-to-Peer Overlay

The purpose of the peer-to-peer overlay is to efficiently find the peer(s) that store and provide a certain video. Typically, in a peer-to-peer overlay the less popular videos will be available from only few peers, while the more popular ones will be provided by many peers. However, the demand for videos is not always equal to their popularity. Mechanisms should be provided in order to also efficiently locate rare (unpopular) videos.

In order to assure a high success rate when searching for both unpopular and popular videos while keeping the overlay maintenance costs low and not using flooding, we use a simple and dynamic structure where the peers periodically collect information about the videos in the overlay, by updating their neighborhoods. This information is kept at each peer in an *indexing-cache*.

Each peer collects in its indexing-cache up-to-date information about the videos of the neighbors. When a peer updates its neighborhood, it replaces an existing neighbor with a new one. The information from the existing neighbor is still kept in the indexing-cache, however, now with an increasing age associated to it, and the information from the new neighbor is added as up-to-date information. Whenever the limit of the indexing-cache is reached, the information with the highest age is removed.

Given that each peer has an accurate knowledge of the videos stored in its neighborhood and (possibly outdated) information about videos from further peers, we use *random walks* of finite length for the search procedure. This method is expected to perform in practice as well as a TTL-limited flooding. To search for a video, a peer sends a video request to a random neighbor that checks its indexing-cache for the video, and if not found, it will repeat the process by sending the video request further to a random neighbor.

The overlay deals easily with peer failure. When a peer from the indexing-cache fails, its corresponding entries are discarded. When a neighbor fails to respond, it is simply replaced with another peer from the overlay.

In order for the search result to contain multiple peers that have the requested video, the indexing-cache can contain several locations for a video; also, the random walk can finish only when a certain number of locations have been found without exceeding the random walk maximum length. If needed, several random walks can be issued. More than two peers having the same video are necessary to enable multiple source streaming (in case of error avoidance), as well as delivering correction streams from alternative sources (in case of error correction).

4 A Model for Handling Network Errors

When using the indexing approach described in Section 3, the network bandwidth between the sender and the receiver is not taken into consideration. In order to deliver the content in the desired quality, it might be necessary to apply (1) error correction, (2) error avoidance or (3) a combination of both approaches. In this section we present a model to select between these three alternatives based on current network and content characteristics. The model combines two measures called Quality Probability and Network Probability. The combination of Quality Probability and Network Probability is called Success Probability.

$$SuccessProbability = QualityProbability * \prod_{i=1}^{M} NetworkProbability_i \quad (1)$$

where M is the number of streaming peers and $NetworkProbability_i$ represents the probability of successfully sending packets between peer i and the receiver. $QualityProbability$ represents the probability that network errors are propagated within the video stream. The combination of both (i.e., the success probability) can take values between 0 and 1.

4.1 Network Probability

Network probability is used to select between alternative peers based on the available bandwidth to the receiver and the bit rate of the video stream. We calculate the available bandwidth between the sender and the receiver based on the "TCP-friendliness" formula obtained from [6]:

$$AvailableBandwidth = \frac{s}{t_{RTT}\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)} \quad (2)$$

where s is the packet size, t_{RTT} is the round-trip time, p is the packet loss probability and t_{RTO} is the TCP retransmit timeout value.

Network Probability is calculated as the ratio between the required bit rate and the available bandwidth:

$$NetworkProbability = min(1, \frac{AvailableBandwidth}{RequiredBandwidth})$$

where $AvailableBandwidth$ is the TCP-friendly available bandwidth (Equation 2) between the sender and the receiver and $RequiredBandwidth$ is the bit rate of the (partial) video stream. Network Probability can take values between 0 and 1. In case the available bandwidth from the sender to the receiver is sufficient to deliver the content without loss, Network Probability has the value of 1.

4.2 Quality Probability

Quality probability expresses the probability that all video frames that arrive at the receiver can be decoded successfully. This probability depends on (1) the number of lost packets and (2) the type of frames affected by the packet loss. MPEG coded video streams [9] consist of three main frame types, I, P and B [4]. I-frames (Intra-coded frames) have the advantage of being self-contained and allowing random access. They have the disadvantage that the compression rate is usually much lower compared to P- or B-frames. P-frames (predictive-coded frame) have a better compression ratio than I-frames but encoding and decoding requires information from the previous I- or P-frames. The third type of frames are Bi-directionally predictive-coded frames (B-frames). The advantage of B-frames is that they have the highest compression ratio compared to I- and P-frames but they additionally depend on one preceding and one succeeding frame in the Group of Pictures (GOP). The preceding can be an I- or P-frame, the succeeding is always a P-frame. So quality probability is used to consider the structure of the stream additionally to the loss rate of the network (NetworkProbability). As another example, a stream encoded using only I-frames and losing many packets, usually results in a better quality in case that the same content is streamed with a lower bit rate but encoded using I-,P- and B-frames. Different packet losses have different effects on the media quality and thus error handling has to be adapted to the relative importance of the frames.

The model requires knowing the number of network packets belonging to each video frame as well as the loss probability of the network path. The loss probability can be expressed as the ratio of transmitted and received packets:

$$LossProbability = \frac{Packets_{Received}}{Packets_{Transmitted}} \tag{3}$$

The number of received packets ($Packets_{Received}$) is determined by sending test packets from the sender to the receiver. The number of packets to be transmitted can be calculated by parsing the structure of the stream.

The *LossProbability* takes values between 0 and 1: 0 means that all packets are received, while 1 means that all packets are lost. Knowing the loss probability (p) of the path and the number of packets (T) belonging to a video frame, the arrival probability (ap), which is the probability for successfully receiving one single frame, can be calculated using the statistical binomial distribution as follows:

$$ap(T,F,p) = \sum_{i=T}^{T+F} \binom{T+F}{i} * p^i * (1-p)^{T+F-i} \tag{4}$$

where T is the number of network packets, F is the number of forward error correction packets and p is the loss probability of the path (defined in Equation 3).

Computing the arrival probability (ap) for one single frame is not sufficient for selecting between streams from alternative peers. Videos have playback times ranging from several seconds to hours and thus analyzing the complete structure would take too long. However the fact that video streams are organized in subsequent groups of pictures (GOPs) can be used to simplify calculations. In the test streams used in our experiments each GOP follows the same frame pattern, "IBBP...", providing sufficient information to make predictions about the complete video.

In order to model packet loss for a group of pictures, I-, P- and B-frames must be analyzed separately as they have different sizes and dependencies:

$$\begin{aligned} ap_I &= ap(N_I, F_I, p) \\ ap_P &= ap(N_P, F_P, p) \\ ap_B &= ap(N_B, F_B, p) \end{aligned} \tag{5}$$

where ap_I, ap_P, ap_B are the probabilities that I-,P- and B-frames are not lost. N_I, N_P, N_B are the numbers of packets for each type of frame, F_I, F_P, F_B are the numbers of forward error correction packets used and p is the network loss probability (defined in Equation 3).
The probability (QualityProbability) for being able to successfully decode all frames belonging to the GOP is defined as:

$$QualityProbability = ap_I * ap_P^{C_P} * ap_B^{C_B}$$

where C_P is the total number of P-frames and C_B is the total number of B-frames. In order to be able to determine the required amount of forward error correction packets (F_I, F_P, F_B) for the I-, P- and B-frames, we compute the arrival probability for each frame separately. The necessity for doing so is explained by giving an example with two frames. Consider that an I-frame has an arrival probability of 50% and the depending P-frame an arrival probability of 100%. Taking into account the dependency between the I-frame and the P-frame, the P-frame can also only be used with a probability of 50%. Sending correction packets for the B-frame would be useless but by using the equations that are explained in the rest of this section, it can be seen that it is the I-frame that needs to be protected.

The computation of the arrival probability (R_I) for the I-frame is simple because no dependencies need to be considered:

$$R_I = ap_I \tag{6}$$

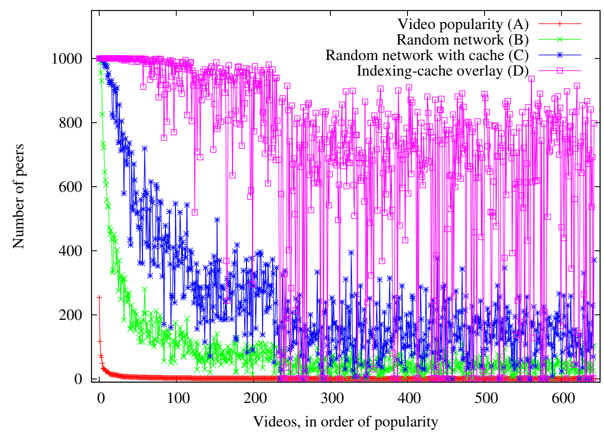

Figure 1. Request success per video.

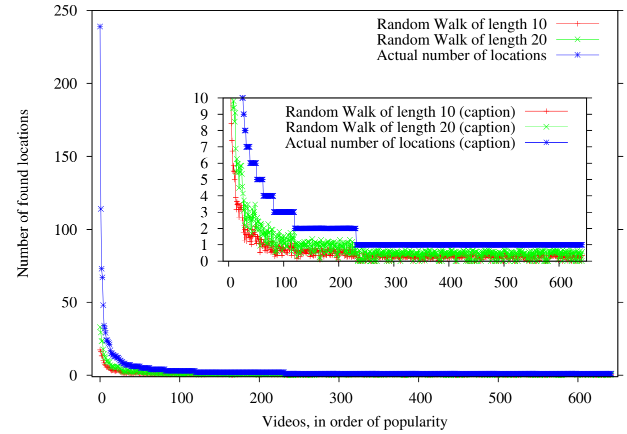

Figure 2. Average number of found locations for each movie.

The dependencies of P- and B-frames are considered in the rest of this section. When computing the arrival probability for P-frame i, the dependencies to the I- and all previous P-frames have to be considered:

$$R_{P(i)} = \begin{cases} R_I * ap_p & \text{if } \ i = 1, \\ R_{P(i-1)} * ap_p & \text{if } \ i > 1 \end{cases} \tag{7}$$

where $P(i)$ is the i^{th} P-frame in the GOP. In case of the first P-frame in the GOP ($i = 1$), only the probability of successfully decoding the I-frame and the P-frame itself is considered. In case that $i > 1$ also the dependencies to all previous P-frames are included.
As B-frames depend on I- and P- frames, the probability of arrival of a B-frame at position j is calculated as:

$$R_{B(j)} = R_{P(k)} * ap_B \tag{8}$$

where $B(j)$ is the j^{th} B-frame in the GOP and $P(k)$ is the immediate successor frame that is referenced.

5 Evaluation

We first evaluate the efficiency of our mechanisms to locate videos (i.e., the indexing-cache overlay) in Section 5.1 — then we present two scenarios for streaming the content in Section 5.2.

5.1 Searching the overlay

In order to evaluate the success rate of both unpopular and popular videos, we have executed a small experiment with 1,000 peers and 643 videos, where each peer has 5 neighbors and an indexing-cache of up to 50 entries. In the indexing-cache there can be up to 2 entries per movie. Each entry of the indexing-cache specifies a location, i.e., a peer that provides the movie. The association of videos to peers follows a Zipf distribution with α=1. Each peer issues a search request for all videos in the form of two random walks, each one with a maximum length of 20 hops. The search procedure stops when at least one location for the requested movie has been found.

The results are presented in Figure 1, where we show the request success for each video. The horizontal axis represents the videos, in order of popularity (most popular movies at the left side). The vertical axis shows the number of peers that successfully find the specified movie. For comparison purposes, we have included in the figure the success rate of the search procedure of the following cases (the legend, from top to bottom):

(A) local-search only, which is actually the Zipf distribution of the videos (i.e., number of peers that have a certain video);

(B) random walks in a random network, no notion of cache;

(C) random walks in a random network, where each peer stores locally, in a cache, the results (i.e., locations) of the video requests that it had issued;

(D) random walks in the indexing-cache overlay; the information from the neighbors (not the search results as before) is cached.

The caches of (C) and (D) have the same size and they use the same aging process as replacement policy. The particularity of these two cases is that whenever a random walk containing a video request arrives at a peer, if the peer does not own the video, it searches for the video in the cache.

The results for the *indexing-cache overlay* (D) show that the popular videos are always found and moreover, most of the unpopular videos are found by at least half of the peers.

Under the same overlay configuration, we have done an experiment that shows the number of locations found for each movie during the search procedure. This time, the search stops only when the maximum random walk length has been reached. (Otherwise, there will be at most 2 hits). Again, each movie is requested from all peers, and we have computed the average number of locations found in the request path using a random walk of length 10 and 20, respectively. The results are shown in Figure 2, where we have also added, for comparison purposes, the number of real locations of each movie in the overlay. As expected, popular movies are found in more locations, while less popular movies are found in a smaller number of locations. The advantage is that the search procedure already returns multiple locations for the movies that are in at least 2 or 3 locations.

5.2 Streaming Scenarios

In this part we show that our model is able to find the best alternative among error correction, error avoidance and a combination of both approaches. To keep the examples comprehensible we pick a small subset of peers in the system and use the same media stream, called Akio for all experiments. The stream consists of two descriptions encoded using MDC in the temporal domain. The full stream has a rate of 1462 Kbit/s when it is encoded using I-frames only and 1081 Kbit/s when it is encoded using I-,P- and B-frames.

The experiments have been performed using the network simulator NS-2 [11] and a plug-in called EvalVid [5]. Data streams from multiple servers are merged and forwarded as one single stream to the player.

5.2.1 Scenario 1

The following example illustrates the necessity of combining Network Probability and Quality Probability. The content is provided by two alternative peers in different qualities (Alternatives A and B, see Figure 3). The question is

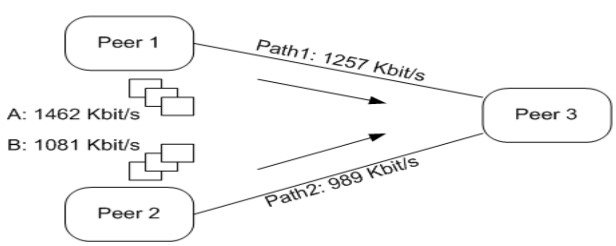

Figure 3. Logical view - Scenario 1.

which peer to select as streaming source. Alternative A is encoded using only I-frames; alternative B is encoded using I-, P- and B-frames. Calculating only Network Probability

Alternative	Bitrate	Avail.-BW	Netw.-Probability
A	1462	1257	0.86
B	1081	989	0.90

Table 1. Stream and Network Characteristics.

(Equation 4.1) yields a better result for alternative B (Table 1). When Success Probability (Equation 1) is calculated

Alternative	Success Probability	MOS
A	0.66	4.02
B	0.5	3.50

Table 2. Success Probability - Scenario 1.

(Table 2) it can be seen that alternative A yields a better result (because of the higher QualityProbability and NetworkProbability). In order to verify the success probability calculation of our model, both decisions are simulated. For comparing the alternative qualities, the Mean Opinion Score (MOS) metric [5] is used. MOS is an objective measure for representing the satisfaction of an end-user receiving a video stream. With this metric the value for the best quality is 5 and for the worst quality is 1.

By sending both streams, it can be seen that considering Network Probability alone is not sufficient and selecting alternative B would have been the wrong decision. The MOS values of alternatives A and B are 4.02 and 3.50, respectively (see Table 2). Alternative B (the one with the lower bit rate) scores worse because of the temporal dependencies to the frames that were lost.

It can be seen that our model is able to select the better alternative. This small example is used to show that computing the ratio between the available bandwidth and the bit rate of the stream is not sufficient to decide between alternative streaming sources, because the structure of the stream has a strong influence on the resulting quality.

5.2.2 Scenario 2

In the second scenario it is assumed that two peers provide the requested content in the same quality. The problem is that the network bandwidth is not sufficient to send any description without loss. Both network paths to the receiver have an average bandwidth of 300 Kbit/s, the required bandwidth for sending description 1 and description 2 are 539 Kbit/s and 542 Kbit/s respectively. The question is either to send one description stream from each of the peers (and accept some loss) or one description stream and one forward error correction stream. When success probability is calculated it can be seen that sending one description plus one forward error correction stream is better than

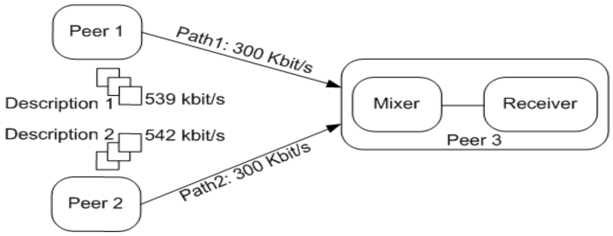

Figure 4. Logical view - Scenario 2.

sending two descriptions (see the higher value of $0,88$ compared to $0,62$ in Table 3). In order to verify the success

Alternative	Success Probability	MOS
2 Descriptions	0,62	2.25
1 Description + FEC	0,88	2.45

Table 3. Success Probability - Scenario 2.

probability (Equation 1), both decisions are simulated. The MOS values from the simulations are also listed in Table 3. It can be seen that the result from sending one description and one forward error correction is 18,2 % better than sending two descriptions (see the higher MOS value of $2,45$ compared to $2,25$). The reason that sending two descriptions leads to a worse result than sending one stream and one forward error correction stream is that none of the two descriptions can be fully received.

6 Conclusions

We have presented a model for selecting between network-error avoidance, network-error correction and a combination of both approaches to deliver multimedia streams over best effort networks in the desired quality. This model was presented in the context of a low-cost indexing-cache overlay that has been shown to deal well with requests for both popular and rare videos and, moreover, to locate multiple peers having the same video.

The error handling model is based on considering network characteristics in combination with stream characteristics. The evaluation has been performed by doing simulations using varying stream and network characteristics within NS-2. The simulation results show that the model can be used to take the decision — error avoidance, error correction or the combination of both — that allows to deliver the stream in the best quality.

References

[1] Bernd Lamparter, Otto Boehrer, Wolfgang Effelsberg and Volker Turau. Adaptable forward error correction for multimedia data streams. Technical Report TR-93-009, University of Mannheim, 1993.

[2] Hang Liu, Hairuo Ma, Magda El Zarki and Sanjay Gupta. Error control schemes for networks: An Overview. *Mobile Networks and Applications*, 2(2):167–182, 1997.

[3] Ivan Lee and Ling Guan. Reliable video communication with multi-path streaming using mdc. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, 2005.

[4] Jill M. Boyce and Robert D. Gaglianello. Packet loss effects on mpeg video sent over the public internet. In *Multimedia '98: Proceedings of the sixth ACM international conference on Multimedia*, pages 181–190, New York, NY, USA, 1998. ACM Press.

[5] Jirka Klaue, Berthold Rathke and Adam Wolisz. Evalvid - a framework for video transmission and quality evaluation. In *Computer Performance Evaluation/Tools*, pages 255–272, 2003.

[6] Jitendra Padhye and Victor Firoiu and Don Towsley and Jim Kurose. Modeling TCP throughput: A simple Model and its Empirical Validation. In *SIGCOMM '98: Proceedings of the ACM SIGCOMM '98 conference Applications, Technologies, Architectures and Protocols for Computer Communication*, pages 303–314, New York, NY, USA, 1998. ACM Press.

[7] John Apostolopoulos, Tina Wong, Wai-tain Tan and Susie Wee. On multiple description streaming with content delivery networks. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1736– 1745, 2002.

[8] Kihong Park and Wei Wang. Qos-sensitive transport of real-time MPEG video using adaptive forward error correction. In *IEEE International Conference on Multimedia Computing and Systems*, volume 2, pages 426–432, 1999.

[9] Mark Claypool and Yali Zhu. Using interleaving to ameliorate the effects of packet loss in a video stream. In *ICDCSW '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*, page 508, Washington, DC, USA, 2003. IEEE Computer Society.

[10] Nicholas Frank Maxemchuk. *Dispersity Routing in Store-and-Forward Networks*. PhD thesis, University of Pennsylvania, 1975.

[11] The Network Simulator NS-2 (v2.1b8a). http://www.ns-2.com, October 2001.