MEDIA-AWARE RETRANSMISSION TIMEOUT ESTIMATION

Ali C. Begen and Yucel Altunbasak

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA USA

ABSTRACT

Developing error-control and error-resiliency methods for transmitting delay-sensitive media content over the best-effort networks poses several challenges. Due to the lack of QoS guarantees in the conventional Internet as well as in emerging wireless networks, these methods must continuously monitor the characteristics of the underlying network and try to infer the incipient network conditions so that they can take the necessary actions on time. This is utmost important for enhancing the end-user quality, particularly in low-delay multimedia applications. In this study, we tackle this problem from an error-control method perspective and develop an innovative framework that optimizes the retransmission decisions based on the urgency and importance of the media packets.

Index Terms— Packet-switched networks, packet delay, jitter, delay prediction, timeout estimation, video dependency structure.

1. INTRODUCTION

In a recent study [1], we developed an adaptive retransmission timeout (RTO) estimation method for low-delay Internet video applications. This method consisted of two main steps: delay prediction and delay-boundary prediction. In the first step, we exploited the temporal dependence among the packet delay samples and used an adaptive linear delay predictor to produce the best estimate in terms of the mean-squared error criterion. This predictor computed the required predictor filter coefficients on the fly and did not use any fixed coefficients. This way, we were able to carry out delay prediction in an optimal fashion regardless of the source video transmission rate and time-varying network conditions. For delay-boundary prediction, on the other hand, we used a controller that optimally managed the trade-off between the amount of overwaiting and spurious retransmissions by regulating the bias to be added to the estimate produced in the first step. The goal was to compute the shortest timeout duration, and hence, to maximize the chance of on-time error recovery such that the redundant retransmission rate did not exceed a desired threshold. Our overall approach merely used the delay samples observed at the client side.

In packetized video applications, however, timely delivery of a packet does not guarantee successful decoding. This is because many video coding standards, *e.g.*, MPEG-x and H.26x, use motion-compensated prediction to gain in coding efficiency at the expense of inducing a dependency structure among the encoded video frames. This dependency structure renders video frames unequally important. For example, a predicted frame can only be decoded after all the frames to which this particular predicted frame is referenced (called *ancestor* frames) are received and decoded. This implies that a frame missing during decoding not only causes errors or a freeze during its display time, but also impedes the successful decoding of all frames that are dependent on it (called *descendant* frames). The resulting error propagation continues through all dependent frames and usually decays slowly. It is therefore essential to optimize the error control for each video packet/frame based on its importance.

In this study, we develop a *media-aware* RTO estimation method that computes the timeout estimates by jointly considering the interdependency relations and the decoding deadlines of video frames. The architecture of media-aware RTO estimation is sketched in Fig. 1. Naturally, we should select a shorter timeout duration for packets belonging to more important and urgent frames than it is for packets belonging to less important and non-urgent frames. If the retransmission capability is severely limited due to scarce bandwidth, we may even opt not to request a retransmission for less important packets and save the retransmission opportunities for more important packets. This prescient discrimination helps us achieve a higher rendering quality of video at the client side without any additional increase in the total transmission rate.

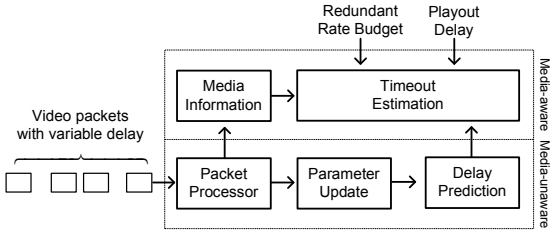

Fig. 1. Architecture of client-driven media-aware RTO estimation.

In the literature, a large number of studies recently explored the problem of rate-distortion optimized media transmission in a variety of setups (See the references in [2]). Inspired by the work of Chou and Miao [3], these studies proposed solutions to find the optimal transmission and/or error-control policies by solving a Markov decision process (MDP) framework. However, to make the analysis tractable and obtain a manageable solution, the original MDP framework ignored the correlation between consecutive packet delay samples, and adopted the assumption of no dependency between the packet loss events and packet delays. These assumptions may hold true for low-bitrate video transmission where the packets are transmitted at large intervals and the delay/loss correlation between the packets is rather insignificant. However, as we will discuss in the next section, these assumptions may not hold if the packets are transmitted at small intervals. In this case, packet delays and loss events will be correlated, and ignoring this correlation will produce sub-optimal transmission policies.

2. PROBLEM FORMULATION

We solve the problem of media-aware timeout estimation within a finite-horizon optimization framework: at each decision epoch a set of frames are considered, and the optimal timeout durations are computed for each packet/frame. Let $\mathcal{S}$ denote a set of frames and assume that the frames within this set have well-defined

interdependency relations that are known by the client. The critical step in media-aware RTO estimation is to develop an expression for evaluating the expected video quality of set $\mathcal{S}$ in terms of the packet decodability probabilities. As it will be clear shortly, the decodability of a packet depends on its on-time delivery probability, therefore, on the amount of its timeout duration, as well as the decodability of the packet(s) on which this packet is dependent.

In our derivations, we quantify the video quality by the average rendered frame rate. We preferred this metric over more sophisticated ones since this metric does not require the knowledge of per-packet distortion information (which can only be extracted during the encoding process). This feature makes it a practical and easy-to-work-with metric. By definition, the achieved frame rate for set $\mathcal{S}$ is computed by

$$\mathcal{Q}_{\mathcal{S}} = f_0 \times \frac{\eta_{\mathcal{S}}^+}{\eta_{\mathcal{S}}}, \tag{1}$$

where f_0, $\eta_{\mathcal{S}}^+$ and $\eta_{\mathcal{S}}$ are the original frame rate, the number of decodable frames and the total number of frames in set $\mathcal{S}$, respectively. Generally, a video frame is packetized into one or more equal-sized packets. Thus, without loss of generality, we assume that the decodable fraction of frame $\mathcal{F}_u$ is given by the ratio of the number of decodable packets in frame $\mathcal{F}_u$ (denoted by υ_u^+) to the total number of packets in frame $\mathcal{F}_u$ (denoted by υ_u). Hence, we have

$$\eta_{\mathcal{S}}^+ = \sum_{u=1}^{\eta_{\mathcal{S}}} \frac{\upsilon_u^+}{\upsilon_u}. \tag{2}$$

Since f_0 and $\eta_{\mathcal{S}}$ are constants in (1), our goal reduces to computing the optimal timeout duration for each packet in set $\mathcal{S}$ such that $\eta_{\mathcal{S}}^+$ is maximized while the expected redundant retransmission probability does not exceed the desired limit. With our notation listed in Table 1, we formalize our optimization problem as follows:

Given: A set of frames, $\mathcal{S}$.

Objective: Find the optimal timeout for each packet in set $\mathcal{S}$.

$$\boldsymbol{\tau}_{opt} = \arg\max_{\boldsymbol{\tau}} \eta_{\mathcal{S}}^+ \tag{3}$$

Subject to: Expected redundant retransmission probability stays within the required limit.

$$\frac{\sum_{u=1}^{\eta_{\mathcal{S}}} \sum_{n=1}^{\upsilon_u} \boldsymbol{p}_f[n]}{\sum_{u=1}^{\eta_{\mathcal{S}}} \upsilon_u} \leq p_f \tag{4}$$

Given a set of frames, the optimization problem defined in (3) and (4) requires the delay prediction for R future packets, where $R = \sum_{u=1}^{\eta_{\mathcal{S}}} \upsilon_u$. That is, if n^* denotes the last successfully-received packet, we need to predict the delays for packets n^*+1, n^*+2, ..., n^*+R. For this purpose, we use the multi-step version of the 2^{nd}-order autoregressive, denoted by AR(2), delay predictor that we proposed in [1]. The r-step AR(2) predictor is defined as follows:

$$\tilde{\mathbf{s}}_2^r[n] = E\left\{\mathbf{s}[n] | \mathbf{s}[n-k], r \leq k \leq r+1\right\} \quad 1 \leq r \leq R. \tag{5}$$

In the following discussion, we develop the mathematical framework for media-aware RTO estimation based on our earlier work [1], and illustrate the relation of the observed delay samples, timeout estimates, playout buffer size and retransmission round-trip times to the video quality. The following equations are provided in a generalized form, however, it should be noted that for packet n, the corresponding r-step predictor filter and error statistics are used, where $r = n - n^*$.

Let p_n denote the probability of packet n being received by its decoding deadline, $t_D[n]$[1]. In our problem scenario, each packet has one initial transmission and one retransmission opportunity. We first examine these cases separately and then combine them together to compute p_n.

[1]Here, the decoding deadline represents the difference between the transmission time at the server and the decoding time at the client.

$\mathcal{A}_n$	Set of the ancestor packets for packet n
$\boldsymbol{\epsilon}_2^r[n]$	Prediction error for packet n
$F_{\boldsymbol{\epsilon}_2^r}$	Cumulative density function of $\boldsymbol{\epsilon}_2^r$
$\mathcal{F}_u$	Frame u
f_0	Frame rate of the original video
$\mathbf{I}[n]$	Retransmission indicator function for packet n
$\eta_{\mathcal{S}}$	Total number of frames in set $\mathcal{S}$
$\eta_{\mathcal{S}}^+$	Number of decodable frames in set $\mathcal{S}$
$\boldsymbol{p}_f[n]$	Pre-mature timeout probability for packet n
p_f	Desired probability of timing out pre-maturely
p_n	Probability of on-time delivery for packet n
p_n^1	Probability of on-time initial transmission for packet n
p_n^2	Probability of on-time retransmission for packet n
P_n	Decodability probability for packet n
$\mathcal{Q}_{\mathcal{S}}$	Video quality of set $\mathcal{S}$
$\mathbf{r}[n]$	Retransmission round-trip time for packet n
$\mathcal{S}$	Set of frames considered in the optimization
$\mathbf{s}[n]$	Observed delay for packet n
$\tilde{\mathbf{s}}_2^r[n]$	Predicted delay for packet n
$\boldsymbol{\tau}[n]$	Additional amount of waiting for packet n
$t_D[n]$	Decoding deadline for packet n
υ_u	Total number of packets in frame $\mathcal{F}_u$
υ_u^+	Number of decodable packets in frame $\mathcal{F}_u$

Table 1. List of the notation for the optimization problem.

The first step is to calculate the probability of on-time initial transmission for packet n. Due to the correlation between the delay samples, this probability is given as follows:

$$p_n^1 = P\left\{\mathbf{s}[n] \leq t_D[n] | \mathbf{s}[n^*], \mathbf{s}[n^*-1]\right\}. \tag{6}$$

Expressing this conditional probability in closed form, however, is difficult. Instead, we can avoid the conditions by substituting $\mathbf{s}[n]$ with $\tilde{\mathbf{s}}_2^r[n] + \boldsymbol{\epsilon}_2^r[n]$. The conditional in (6) can be now expressed as an unconditional probability of the random variable $\boldsymbol{\epsilon}_2^r$:

$$p_n^1 = \begin{cases} F_{\boldsymbol{\epsilon}_2^r}\left(t_D[n] - \tilde{\mathbf{s}}_2^r[n]\right), & \text{if } \mathbf{s}[n] < \infty; \\ 0, & \text{if } \mathbf{s}[n] = \infty. \end{cases} \tag{7}$$

Note that p_n^1 is still conditioned on whether packet n is lost or not since the prediction error is only defined for non-lost packets. Thus, we also need to compute the loss probability of packet n.

In order to understand the relation between the packet loss and delay, we plot the loss probability of packet n^*+r as a function of the delay of packet n^*. Fig. 2 shows that the loss probability for packet n^*+1 is negligible if packet n^* experienced a delay smaller than 220 ms. However, if packet n^* experienced a delay between 220 and 260 ms, the chance of being lost for packet n^*+1 increases up to 50%. Clearly, there is a strong dependence between the loss probability of packet n^*+1 and the delay of packet n^*. More importantly, a noticeable dependence also exists between the loss probability of packet n^*+r and the delay of packet n^* for $r \leq 10$. Ignoring this dependence and merely using the average packet loss rate (2.2% for this particular trace) would result in either an overrated or underrated packet loss probability. Therefore, it is important that we express the loss probability of packet n as a conditional on the last observed delay sample.

$$P\left\{\mathbf{s}[n] = \infty\right\} = P\left\{\mathbf{s}[n] = \infty | \mathbf{s}[n^*]\right\} \tag{8}$$

In practice, the conditional loss probability distribution can be generated on the fly. A closed form expression is not essential for media-aware RTO estimation.

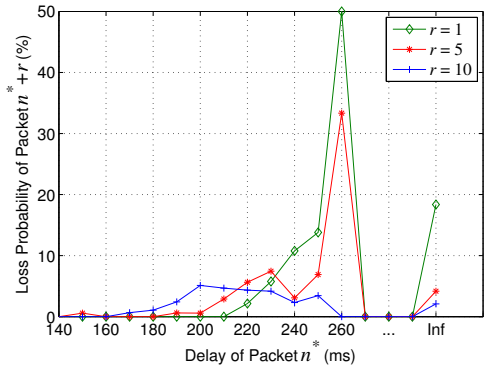

Fig. 2. Relation between the packet delay and loss probability. Produced by streaming an H.264-encoded test sequence (300 Kbps) over an Internet topology [4] in *ns*-2 environment [5].

Having computed the first step of the on-time delivery probability, we now compute the probability of the retransmission for packet n being received before the decoding deadline. Recall that when a packet is received, the client predicts the delay for the subsequent packet and estimates the timeout duration. If the expected packet is still not received within this time, a retransmission request is sent to the server. Assuming that the request is immediately processed by the server, the probability of on-time retransmission equals

$$p_n^2 = P\left\{\tilde{\mathbf{s}}_2^r[n] + \boldsymbol{\tau}[n] + \mathbf{r}[n] \leq t_D[n]\right\}, \tag{9}$$

where $\mathbf{r}[n]$ is the round-trip time of the retransmission. It is important to note that in (9) we do not impose any condition on the previous delay samples. The reason is that the correlation between $\mathbf{r}[n]$ and $\mathbf{s}[n^*]$ is usually insignificant. Thus, p_n^2 can be computed from the empirical distribution of $\mathbf{r}$ in a straightforward manner.

Once we have computed p_n^1 and p_n^2, it is easy to express p_n as

$$p_n = p_n^1 + \mathbf{I}[n] \times (1 - p_n^1) p_n^2, \tag{10}$$

where $\mathbf{I}[n]$ is an indicator function: $\mathbf{I}[n] = 1$ if a retransmission is requested for packet n, and 0 otherwise. Considering that the chance of a retransmission arriving earlier than the initial transmission is negligible, (10) reduces to

$$\begin{aligned} p_n = &\left(1 - P\left\{\mathbf{s}[n] = \infty\right\}\right) F_{\boldsymbol{\epsilon}_2^r}\left(t_D[n] - \tilde{\mathbf{s}}_2^r[n]\right) + \\ &\mathbf{I}[n] \times P\left\{\mathbf{s}[n] = \infty\right\} P\left\{\tilde{\mathbf{s}}_2^r[n] + \boldsymbol{\tau}[n] + \mathbf{r}[n] \leq t_D[n]\right\}. \end{aligned} \tag{11}$$

As mentioned previously, packet n can only be decoded if all of its ancestor packets were decoded successfully. Thus, the decodability probability of packet n equals the following product:

$$P_n = p_n \times \prod_{n' \in \mathcal{A}_n} p_{n'}, \tag{12}$$

where $\mathcal{A}_n$ denotes the set of the ancestor packets for packet n. Here, we observe how the dependency structure of the streamed video explicitly factors in the video quality. More implicitly, we also notice that as more of its descendant packets are received by the client, an ancestor packet becomes more important since its successful delivery would enable the decoding of several packets. Note that in (12), we are able to express P_n as the product of individual packet decodability probabilities since any existing correlation is already taken into account while computing p_n.

Given the packet decodability probabilities, we compute the expected number of decodable packets in frame $\mathcal{F}_u$ from $\upsilon_u^+ = \sum_{n=1}^{\upsilon_u} P_n$. Finally, the expected video quality of set $\mathcal{S}$ is calculated by plugging υ_u^+ into (2).

The last step in our optimization problem is to calculate the redundant retransmission probability for each packet. We compute the pre-mature timeout probability for packet n as follows:

$$\begin{aligned} \boldsymbol{p}_f[n] &= \mathbf{I}[n] \times P\left\{\tilde{\mathbf{s}}_2^r[n] + \boldsymbol{\tau}[n] < \mathbf{s}[n]\right\} \\ &= \mathbf{I}[n] \times \left(1 - F_{\boldsymbol{\epsilon}_2^r}(\boldsymbol{\tau}[n])\right). \end{aligned} \tag{13}$$

A solution to (3) is feasible only if the expectation of the redundant retransmission probability over all packets in set $\mathcal{S}$ satisfies the constraint given in (4).

3. SOLUTION APPROACH & IMPLEMENTATION ISSUES

Depending on the complexity of the video dependency structure and the horizon of the optimization, the solution to our optimization problem can potentially require a large number of multiplications and additions. In practice, however, solving the system given in (3) and (4) is less complicated than it may seem. For example, when the network conditions are not severe and packet delays are below a certain threshold, the client can safely skip computing a timeout estimate for the subsequent packets based on the knowledge that the loss probability for those packets is negligible. The delay traces we collected reveal that the majority of the packets usually experience a non-critical delay, implying that the computational load of the RTO estimation on the client is often minimal.

It is, however, critical to solve (3) and (4) for the client when it infers an incipient congestion. An important issue in this optimization is the selection of the optimization horizon and the granularity of the timeout durations. Suppose that we have R packets and we need to select a timeout duration for each of them from a set of H quantized values. In this case, our solution has a complexity of $O(H^R)$. Due to the exponential relation, the optimization horizon R cannot be chosen arbitrarily large. Furthermore, the predictive accuracy of the multi-step delay predictor degrades with R. Fig. 3 shows that the prediction-error standard deviation doubles at step four and triples at step 10 for all three delay traces collected at different streaming rates. Since a poor prediction has no practical use, we suggest that the optimization horizon should not exceed 10. On the other hand, the value of H depends on the maximum complexity tolerable by the client. In this study, we selected the timeout durations among seven different values from the set $\mathcal{H} = \{0 \text{ ms}, 20 \text{ ms}, 40 \text{ ms}, 60 \text{ ms}, 80 \text{ ms}, 100 \text{ ms}, \infty\}$.

For R-step prediction, we require R sets of the predictor filter coefficients, $\alpha_{1,2}^r$ and $\alpha_{2,2}^r$. To compute these coefficients, we use a window-based approach. The window size W is chosen short enough to ensure the pseudostationarity of the input data over the length W. Our tests indicate that $W = 20$ is a good choice. Given this window size, the predictor filter coefficients are computed by solving the Yule-Walker equations, which requires the knowledge of the sample autocorrelations for the first two lags in our case [1]. When a new sample is observed, the oldest sample is removed from the window and the other samples remain unchanged. Thus, the sample autocorrelations can be updated in an efficient manner. Furthermore, the filter coefficients vary over time, but due to the pseudostationarity of the data, we observe that $\alpha_{1,2}^r + \alpha_{2,2}^r = 1$. Thus, it is sufficient to compute only one of the coefficients for each r.

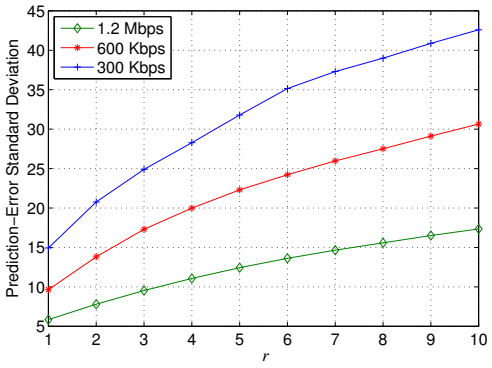

Fig. 3. Variation of the prediction-error standard deviation with r.

4. SIMULATION RESULTS

In this section, we analyze the performance of the media-aware RTO estimator. We denote this estimator with $\mathrm{RTO}_{\mathrm{Media-aware}(R)}$, where R is the optimization horizon. Naturally, $\mathrm{RTO}_{\mathrm{Media-aware}(1)}$ performs exactly the same as the media-unaware RTO estimator that was proposed in [1], which we denote by $\mathrm{RTO}_{\mathrm{AR}(2)}$. To better illustrate the impact of R on the performance of $\mathrm{RTO}_{\mathrm{Media-aware}(R)}$, we compare the on-time arrival rates of the individual frames in a GOP. For this purpose, we encoded a test sequence with a standard H.264 codec [6] at 300 Kbps and 20 frames per second. The adopted GOP structure was one I-frame plus nine P-frames. We streamed this video multiple times between two end-points in a moderately-congested Internet topology, where the forward-path packet loss rate averaged 5%.

In Fig. 4, we plot the average on-time arrival rates of the individual frames when the playout buffer is 500 ms. Under the adopted simulation settings, we observe that $\mathrm{RTO}_{\mathrm{AR}(2)}$ could deliver approximately 30% of the retransmissions on time, and as expected, this success rate did not vary much among the frames. In contrast, $\mathrm{RTO}_{\mathrm{Media-aware}}$ was able to deliver as much as 40% of the I-frame retransmissions on time at the expense of least important P-9 frames. In other words, $\mathrm{RTO}_{\mathrm{Media-aware}}$ recovered more of the important video content by not increasing the streaming rate, but by relinquishing the recovery of the less important content. It is important to note that as we increased the optimization horizon, the optimization gain improved.

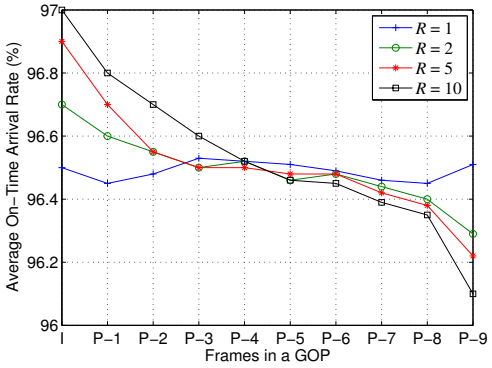

Fig. 4. Variation of the frame on-time performance with the optimization horizon when the playout buffer is 500 ms.

Next, we plot the average on-time arrival rates when the playout buffer is relaxed to 600 ms. In this case, $\mathrm{RTO}_{\mathrm{AR}(2)}$ delivered 90% of the retransmissions on time. Naturally, the on-time retransmission probability improved with an increase in the playout buffer size. Although $\mathrm{RTO}_{\mathrm{Media-aware}}$ still produced better quality video, the performance gap between the media-aware and media-unaware approaches reduced with respect to the previous case. Thus, we conclude that the media-aware RTO estimation becomes more crucial under low end-to-end delay requirements, and as the delay requirement relaxes, its performance will converge to that of the media-unaware RTO estimation.

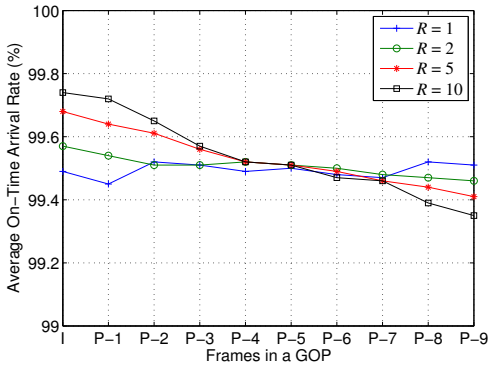

Fig. 5. Variation of the frame on-time performance with the optimization horizon when the playout buffer is 600 ms.

5. CONCLUSIONS

Previously, we proposed an autoregression-based adaptive RTO estimation method for low-delay Internet video applications. This method substantially outperformed existing estimators such as Jacobson's algorithm and recursive weighted median filtering. In this study, we furthered our approach and developed a media-aware RTO estimator. This RTO estimator computes the optimal timeout duration for each packet such that the rendered video quality is maximized for a given retransmission rate budget. Our simulations show that media-aware RTO estimation provides a significant quality improvement over its media-unaware counterpart, particularly when the application requires a low end-to-end delay.

Acknowledgments– This work is supported by NSF under NSF award CCF-0430907.

6. REFERENCES

[1] A. C. Begen and Y. Altunbasak, "Redundancy-controllable adaptive retransmission timeout estimation for packet video," in *ACM NOSSDAV*, 2006.

[2] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

[3] ——, "Rate-distortion optimized streaming of packetized media," *Microsoft Research Technical Report MSR-TR-2001-35*, 2001.

[4] E. W. Zegura, K. L. Calvert, and M. J. Donahoo, "A quantitative comparison of graph-based models for Internet topology," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 770–783, 1997.

[5] S. McCanne and S. Floyd. Network simulator. [Online]. Available: http://www.isi.edu/nsnam/ns

[6] H.264 AVC reference software. [Online]. Available: http://iphome.hhi.de/suehring/tml/download