Multiple Description Coding with Feedback Based Network Compression

Jesper H. Sørensen*, Jan Østergaard†, Petar Popovski*, Jacob Chakareski†,
*Aalborg University, Department of Electronic Systems, E-mail: {jhs, jo, petarp}@es.aau.dk
†École Polytechnique Fédérale de Lausanne, Signal Processing Institute, E-mail: jakov.cakareski@epfl.ch

Abstract—This paper concerns multi path video streaming using adaptive multiple description coding. The adaptation leverages on the fact that multiple descriptions are correlated. Thus if an intermediate node gets feedback telling that another path is likely to deliver a description, this node can compress its description and forward it. Such a compression can also be done already at the source node; however, the feedback arrives more timely and reliably to intermediate nodes that are closer to the final receiver. In this paper we investigate the performance of such adaptation at the source node and an intermediate node, respectively. A trade-off exists between reducing the delay of the feedback by adapting in the vicinity of the receiver and increasing the gain from compression by adapting close to the source. The analysis shows that adaptation in the network provides a better trade-off than adaptation at the source. Schemes which provide simple solutions to adaptation both at the source and in the network are proposed, analyzed, simulated and compared to non-adaptive reference schemes in scenarios that involve last hop that is wireless. The results reveal that the proposed compression schemes offer significant benefits in streaming scenarios.

I. INTRODUCTION

Internet video streaming requires high bandwidth and low delay, such that major challenges are dropped packets and late arrivals. A viable approach to combat these problems is streaming along multiple paths [1], which increase the likelihood of successful data delivery. Multiple description coding [2] (MDC) encodes a media stream into multiple independent descriptions and is particularly suited for multi path streaming. Each description allows a low-quality content reconstruction, while every additional description improves the quality. As the descriptions are self-sufficient, they introduce a large redundancy, which is wasteful when no descriptions are lost. This redundancy is justified if the server has no information about the network quality or channel state information (CSI) of a given path. Different from MDC, layered coding [3] (LC) encodes the media into prioritized layers. A layer is only decodable if all other layers of higher priority have already been decoded. Layers are not self-sufficient, hence, the redundancy in LC is limited. In general, MDC is preferable over LC in error-prone networks. However, the lack of redundancy in LC decreases the delay, which is critical in time-constrained systems.

In this work, the receiver sends feedback about network condition, which is useful when deciding between MDC and LC. The scenario is multi path streaming with MDC from a single source to a single receiver. Our starting observation is that the descriptions are correlated and therefore hold a compression potential: if a node that needs to relay description 1 receives feedback telling that description 2 is likely to be received from another path, then this node can compress description 1 before forwarding it, thus effectively creating a layered source code.

Two variants of the adaptive scheme are proposed. One where the compression decision is performed at the source and one where the decision is performed at an intermediate point in the network. These schemes differ in performance due to the different delay of the feedback. When the source needs to do the compression, end-to-end feedback is required, such that it arrives with a much larger delay as compared to an intermediate network node. This creates the problem of a low correlation between the feedback information and the actual network state at the time of compression decision. Hence, the relationship between feedback delay and network dynamics is an important issue in this work. The hypothesis is that an informed decision in the network outperforms an informed decision at the server due to decreased delay of the feedback. Moreover, an informed decision in the network outperforms an uninformed decision at the server due to better adaptation to network conditions.

Multi path streaming has been the focus of much work over the recent years. In [4] the use of MDC in multi path networks is investigated and compared to a single description code transmitted along a single path. It is shown that at higher network loads the gain from path diversity outweighs the loss from the increased redundancy in MDC. [5] considers single description code in a multi path network. Here focus is on finding the optimal transmission policy for individual packets, i.e. whether or not to transmit the packet, when to transmit it and which path to use. A similar problem is considered in [6], although for single path networks. Here the decision process has been moved to a proxy, which requests data from the server and forwards it to the receiver.

II. SYSTEM MODEL

The targeted scenario is video streaming to a single client along two paths, both having wired links and a wireless last hop. The network between the source and the wireless transmitters is not further specified, but will be treated as a network cloud. Fig. 1 shows the topology of the network.

Each link incurs a delay of the video stream, which is denoted $t_{i}^{w}$ for wired links and $t_{i}^{w}$ for wireless links, where $i$ is the path number. It is assumed that no errors occur on
state time. Choosing higher values of $\tau$ where at this purpose a finite state Markov chain (FSMC) is used. The wireless links use a simple stop-and-wait ARQ with the packet is dropped if there was no successful transmission. Moreover, the wireless links use adaptive modulation.

A. Fading Model

It is expected that video streaming most frequently occurs in urban areas with many scattering objects and no line of sight, hence Rayleigh fading is assumed. This means that the distribution of the instantaneous SNR $\gamma$ is $f_\gamma(\gamma) = \frac{1}{\bar{\gamma}} \exp\left(-\frac{\gamma}{\bar{\gamma}}\right)$, where $\bar{\gamma}$ is the mean SNR. The proposed schemes rely on feedback information on the instantaneous SNR, hence the fading model is an integral part of these schemes. Channel state is discretized in order to enable reporting through feedback. For this purpose a finite state Markov chain (FSMC) is used.

In the FSMC the SNR spectrum is divided into $N$ intervals with boundaries $\Gamma_k$ for $k = 0, 1, \ldots, N$. Each interval is represented by a state in the FSMC. Note that $\Gamma_0 = 0$ and $\Gamma_N = \infty$. Boundaries in between are found using the method proposed in [8], where a uniform stationary probability distribution is ensured. Hence, $\pi_k = \frac{1}{N}$ for $k = 1, 2, \ldots, N$, where $\pi_k$ is the stationary probability of being in state $k$.

For the FSMC we define a state transition probability matrix, $P$, where $P_{i,j}$ denotes the probability of a transition from state $i$ to $j$. It is assumed that only transitions to neighboring states and the current state itself occur, hence $P_{i,j} = 0$ for $|i - j| > 1$. The probabilities for transitions to neighboring states, $P_{k,k+1}$ and $P_{k,k-1}$, can be approximated by [9]

$$P_{k,k+1} \approx \frac{\xi(\Gamma_{k+1})}{\pi_k} \tau, \quad k = 1, 2, \ldots, N - 1,$$

$$P_{k,k-1} \approx \frac{\xi(\Gamma_k)\tau}{\pi_k}, \quad k = 2, 3, \ldots, N,$$

where $\xi(\Gamma_k)$ is the level crossing rate of the received SNR at $\Gamma_k$ and $\tau$ is the time step size of the FSMC. The value of $\tau$ is chosen arbitrarily to get a suitable model granularity in time. Choosing higher values of $\tau$ means that a state transition models the passing of more time, while the converse holds for lower values of $\tau$. The level crossing rate [10] is given by $\xi(\Gamma_k) = \sqrt{\frac{2n_0}{\pi}} f_m \exp(-\frac{T}{2})$, where $f_m$ is the maximum Doppler shift. The probability that the channel stays in the current state, $P_{k,k}$, is found as $P_{k,k} = 1 - P_{k,k+1} - P_{k,k-1}$.

Having found the elements of $P$, it is possible to calculate the state distribution after $n$ state transitions from a given known initial state, $S_0$. This is done as follows:

$$S_n = (P^T)^n \cdot S_0,$$

where $S_n$ is a vector whose $k$'th element is the probability of being in state $k$ after $n$ transitions, and $S_0$ is a vector whose $k$'th element is $1$ if $S_0 = k$ and zeros elsewhere. Each state is associated with a modulation scheme, optimized for that SNR interval, as described next.

B. Adaptive Modulation

Selection of modulation for a given state is done based on the achieved throughput for transmissions with no error control. Hence, we need to optimize:

$$M(\gamma) = \max_M \int_{\Gamma_{k-1}}^{\Gamma_k} f_\gamma(\gamma) \log_2(M)(1 - p_{mtu}(\gamma))^d \, d\gamma,$$

where $M$ is the constellation size of the applied modulation scheme and $p_{mtu}(\gamma)$ is the error probability for the maximum transmission unit (MTU) of the wireless link, which has a length of $\delta_{mtu}$ in bits. $\theta = \frac{\delta_{mtu}}{\bar{\gamma}}$ and is the number of MTUs necessary for the transmission of a description. The constellation size is adapted to a state and not a specific instantaneous SNR, hence the expectation over the SNR region covered by the $k$'th state. The possible modulation schemes are $M$-ary QAM with even $\log_2(M)$. For these schemes the, per carrier, symbol error probability, $p_c'$, is $p_c' = \left(1 - \frac{1}{\sqrt{M}}\right) \text{erfc} \left(\sqrt{\frac{3\gamma}{\sqrt{2}\alpha}}\right)$, where $\text{erfc}(x)$ is the complementary error function. Since we need to detect two carriers per symbol in QAM and we need $\alpha = \frac{\delta_{mtu}}{\log_2(M)}$ symbols, $p_{mtu}(\gamma)$ is found as follows:

$$p_{mtu}(\gamma) = 1 - (1 - p_c')^{2\alpha}.$$

Thus (5) becomes an optimization problem as a function of $\gamma$.

III. THE INVESTIGATED SCHEMES

This section describes the details of the reference schemes and the proposed schemes. Before describing the individual schemes, the applied variants of source coding are explained, since these are the basis for all schemes.

A. Source Coding

The target application in this work is video streaming. Usually a video stream is partitioned into Groups Of Frames, GOF, of a certain size, each of which are treated individually by the source encoder. Without further specifying the source encoder, we assume that it creates a layered representation of the source consisting of a base layer and a refinement layer. The SVC extension of the MPEG4 codec is an example of an encoder...
using this approach [11]. The base layer is independent and can be decoded separately, while the refinement layer is only decodable if the base layer has already been received. The layers are assigned individual rates which result in cumulative rates denoted by \( r_z \), where \( z \) indicates the number of decoded layers, \( z = 0, 1, 2 \). This means that a rate of \( r_1 \) is allocated to the base layer, while the refinement layer is allocated \( r_2 - r_1 \).

Associated with a rate, \( r_z \), is a distortion, \( d_z \). In this work we do not evaluate the schemes for specific video content, but as we do not evaluate the schemes for specific video content, but rather the theoretical upper bound from rate-distortion theory [12], which says that for an i.i.d., zero-mean, unit-variance Gaussian source the distortion-rate function is:

\[
d(r_z) = 2^{-2z},
\]

where \( \kappa \) is samples pr. GOF. The scaling by \( \frac{1}{\kappa} \) is necessary here since \( r_z \) is defined as bits pr. GOF, while general rate-distortion theory uses bits pr. symbol.

The layered representation of the GOF is transcoded into descriptions using a principle called MDC-FEC [13][14]. The idea of this principle is to use Forward Error Correction, FEC, on each layer, where the code rate is \( \frac{1}{z} \) for the \( i \)th layer counting from the base layer, \( i = 1, 2, ..., L \). The resulting data for each layer is divided into \( L \) parts of equal size which are distributed across \( L \) descriptions. A description is an aggregation of parts from all layers. Fig. 2(a) illustrates this principle for the example of three layers and three descriptions.

Due to the error correction property of the FEC, only a ratio of \( \frac{1}{z} \) of the bits from layer \( i \) is needed in order to decode that layer. Hence, for decoding the base layer we only need a single description. In general, in order to decode the \( i \)th layer we need any \( i \) descriptions.

In our case we wish to transcode two layers into two descriptions. This is easily done with MDC-FEC as shown in Fig. 2(b). Switching to a layered representation of the GOF is done by removing the redundant base layer data in one of the descriptions. Such a compression makes the description less susceptible to errors and it will decrease its delay, but it also makes the client dependent upon the base layer of the description, since the refinement layer is useless by itself. Hence, whether compression is advantageous depends on the network conditions. Compression is used to switch between a layered representation and multiple descriptions in the proposed adaptive schemes, which are presented in section III-C.

B. Reference Schemes

The two reference schemes are static with respect to source coding. One scheme is using the uncompressed source code. In this scheme one description is transmitted on each path of the network and the wireless transmitters just forward the received data. This scheme is referred to as Static MDC. The other scheme uses the compressed source code and the compressed description is sent through path 1 and the uncompressed one through path 2. Again, the wireless transmitters just forward the data. This scheme is referred to as Static LC.

C. Proposed Schemes

The proposed schemes are both adaptive in the sense that they choose whether to compress one of the descriptions or not based on feedback. The feedback is the current state of the wireless channels at the time of creation of the feedback. It is assumed that the receiver is able to obtain this information through estimation, as in e.g. [15]. Time will pass between the creation of this feedback and the next use of the wireless channels, hence the channel states at the time of next use will follow certain distributions biased by the feedback. These distributions are found using (4). For a certain distribution pair the adaptive schemes evaluate the expected performance of all possible compressions and select the optimal one. The options are not to compress, to compress the description transmitted on path 1 and to compress the description transmitted on path 2 and the associated optimization is presented in Section IV.

The difference between the adaptive schemes is where the compression decision is made. One scheme makes the decision at the source, which means that feedback must be transmitted from client to source. This scheme is referred to as Adapt@Source. The other scheme makes the decision in the network, i.e. in a distributed fashion where the relays perform the compression. This requires only feedback from the client to the relays. This scheme is referred to as Adapt@Network. For this scheme it is assumed that both relays receive feedback for both wireless channels and use the same decision algorithm. In this way both relays are able to perform joint optimization for the two paths and carry out the compression required in their individual paths. Adaptation of the source code in the network is normally a difficult task, however, in Adapt@Network it is merely a matter of dropping a part of a description.

The two adaptive schemes represent two options in an interesting trade-off. On the one hand it is preferable to make the compression decision as soon as possible, i.e. at the source, since this makes it possible to benefit from the decreased description size end-to-end. On the other hand, it is preferable to postpone the decision as far as possible, i.e. to the wireless last hop, since this makes it possible to utilize the most recent feedback which will act as a stronger bias in the state distribution from (4).

IV. Analysis

The goal of this analysis is to derive the necessary equations in order to compare the proposed schemes to the reference schemes. There is only a small difference in the analysis of the two proposed schemes, hence only the derivations for Adapt@Source will be presented followed by a description of the difference.
A. Analysis of Adapt@Source

The performance metric applied in the comparison of the schemes is expected distortion, which is defined as follows for a specific compression decision indicated by $c$ and a specific wireless channel state $S_n$:

$$E[d|c, S_n] = \sum_{z=0}^{2} p^c_z \cdot d(r_z),$$

where $d(r_z)$ is given in (7) and $p^c_z$ is the probability of having $z$ decodable layers. The values of $r_z$ are determined in the design of the source code, which is not within the scope of this paper. The possible values of $c$ are $c0$, $c1$ and $c2$, indicating no compression, compression of description 1 and compression of description 2, respectively. Note that the expression in (8) is for a given instantaneous SNR, which $p^c_z$ depends on. When the transmitter makes a compression decision, there will be uncertainty about the instantaneous SNR, because time have passed since the feedback was transmitted. Hence, when evaluating the expected normalized throughput, this uncertainty must be taken into account. This is done by calculating the expectation over the state distribution found in (4). Hence, if $F_i$, $i = 1, 2$, indicates the state information fed back for path $i$, then for a specific feedback combination, $F_1, F_2$, the expected distortion is

$$E[d|F_1, F_2] = \min_c \sum_{S_n^1=1}^{N} \sum_{S_n^2=1}^{N} S_n^1(S_n^2) \cdot S_n^2(S_n^1) \cdot \sum_{z=0}^{2} p^c_z \cdot d(r_z),$$

since the scheme makes the decision $c$ which yields the highest expected performance based on the knowledge provided by $F_1, F_2$. The value of $n$ is found by evaluating the number of state transitions between time of transmission of feedback and time of transmission of the next description at the wireless link. This time is assumed to be deterministic and composed of four components. The mean transmission delay of the description and the feedback in the wired link, a constant component covering propagation and queuing delay of the feedback in the wired link and the delay associated with an error free transmission in the wireless link. The assumption of a deterministic delay is justified by the relative small size of a feedback packet, which is also why propagation and queuing delay must be taken into account. For smaller packets, these components have a higher relative contribution, since they are independent of the packet size. Hence, $n = \frac{\mu_{fb} + \mu_{w}}{\mu_{fb} + \mu_{w} + \nu_{wd}}$, where $\mu_{fb} = \frac{\delta}{\nu \log(2) (\pi)}$. $\nu_{wd}$ is the sum of propagation and queuing delay in the wired link and $r_s$ is the symbol rate in the wireless link.

The value of the feedback follows the distribution $\pi$, which must also be included in the evaluation of the expected distortion. Hence,

$$E[d] = \sum_{F_1=1}^{N} \sum_{F_2=1}^{N} \pi_{F_1} \cdot \pi_{F_2} E[d|F_1, F_2]$$

The expression of $p^c_z$ depends on the decision by the scheme on whether or not to compress one of the descriptions. Regardless of whether this decision is made at the source or in the network, the following equations hold:

$$p^c_0 = \bar{p}^1_{l,S_n} \cdot \bar{p}^2_{l,S_n}$$
$$p^c_1 = \bar{p}^0_{l,S_n} \cdot (1 - \bar{p}^2_{l,S_n}) + (1 - \bar{p}^0_{l,S_n}) \cdot \bar{p}^2_{l,S_n}$$
$$p^c_2 = (1 - \bar{p}^0_{l,S_n}) \cdot (1 - \bar{p}^2_{l,S_n})$$

where $\bar{p}^i_{l,S_n}$, $i = 1, 2$, is the mean probability of losing description $i$ when the wireless channel is in state $S_n$. Equations for compression of description 2 are equivalent to those for compression of description 1.

A description can be lost for two reasons. One reason is when the ARQ process in the wireless link fails to deliver the description reliably, which results in an erasure whose probability is denoted $p_e$. This happens when the number of necessary (re)transmissions, $q$, exceeds the allowed seven. The other reason is when the description is received beyond the decoding deadline, whose probability is denoted $P(t_r > t_d)$, where $t_r = t_{wd} + t_{ws}$. Hence, we can express the loss probability for any state as follows:

$$p^c_l = p^c_e + (1 - p^c_e)P(t_r > t_d)q^i \leq 7$$

The derivations of the two components of (13) are key to this analysis and will be treated in the following. Due to symmetry in the network, the derivations of $p^c_1$ and $p^c_2$ are equal. Hence, the superscripts are omitted in the remainder of the analysis.

For the derivation of the expression of $p_e$, we need to find the probability distribution of the number of necessary trials in the ARQ process before success is achieved. This is denoted as $f^m_{\text{mtu}}(q)$, where $q$ is the index of the first successful transmission, $q = 1, 2, \ldots, \infty$. Note that this distribution is for an MTU, which is the data level processed by the ARQ protocol. The distribution is expressed by the geometric distribution:

$$f^m_{\text{mtu}}(q) = p^m_{\text{mtu}} \cdot (1 - p^m_{\text{mtu}})$$

where $p^m_{\text{mtu}}$ is the error probability for an MTU, which was expressed in (6). We allow seven (re)transmissions in the ARQ process before a packet is dropped. Hence, the erasure probability of an MTU, $p^m_{\text{mtu}}$, is

$$p^m_{\text{mtu}} = 1 - \sum_{q=1}^{7} (p^m_{\text{mtu}})^{q-1} \cdot (1 - p^m_{\text{mtu}}).$$

Since we need $\theta = \frac{\delta}{\nu \log(2) (\pi)}$ MTUs for each description, the description erasure probability, $p_e$, is expressed as

$$p_e = 1 - (1 - p^m_{\text{mtu}})^{\theta}. $$
For the second component, \( P(t_r > t_d|q \leq 7) \), we need to find the distribution of \( t_r \), given that the ARQ process did not fail. The value of \( t_r \) is the sum of two random variables, \( t_{wd} \) and \( t_{ws} \). The distribution of \( t_{wd} \) is already known from (1). The distribution of \( t_{ws} \) is derived in the following. For this purpose we reuse the distribution in (14), although with a modification. In order to condition on the event that the ARQ process did not fail, we truncate and normalize the distribution as follows:

\[
 f^{\text{mtu}}_q(q|q \leq 7) = \frac{p^{(q-1)}_{\text{mtu}} \cdot (1 - p_{\text{mtu}})}{1 - p^{q}_{\text{mtu}}} , \quad q = 1, 2, ..., 7 \tag{17}
\]

We now define a new random variable, \( q' \), which is the total number of (re)transmissions used in the ARQ process for all MTUs needed for a single description. The distribution of this random variable is the \( \theta \)-order convolution, denoted \( f^\theta_q \), of \( f^{\text{mtu}}_q(q|q \leq 7) \). Hence,

\[
 f_{q'}(q') = f^{\text{mtu}}_q(q|q \leq 7) \ast f^{\text{mtu}}_q(q|q \leq 7) \tag{18}
\]

We now have the distribution of the delay in the wireless link, measured in transmissions. This distribution must be expressed in the time domain, which is done by mapping the probabilities through the relationship \( t_{ws} = \mu_{\text{mtu}} q' \), where \( \mu_{\text{mtu}} = \frac{1}{\log_2(4M)} \). This gives us the desired distribution of the delay in the wireless link, \( f_{t_{ws}} \). The distribution of the total delay is then given as follows:

\[
 f_{t_r}(t_r) = f_{t_{wd}} \ast f_{t_{ws}} \tag{19}
\]

The probability of reception past the decoding deadline can now be found as

\[
 P(t_r > t_d|q \leq 7) = \int_{t_d}^{\infty} f_{t_r}(t_r) dt_r \tag{20}
\]

By substituting (16) and (20) into (13) we can express the loss probability as a function of SNR.

\[
 p_l = 1 - (1 - p_e^{\text{mtu}}) \theta + (1 - p_e^{\text{mtu}}) \theta \int_{t_d}^{\infty} f_{t_r}(t_r) dt_r \tag{21}
\]

However, a given state in the FSMC covers a whole range of SNR values. Hence, the mean loss probability in a given state, \( \bar{p}_{l,S_n} \), is

\[
 \bar{p}_{l,S_n} = \int_{\Gamma_{S_n}}^{\Gamma_{S_{n+1}}} p_l \tag{22}
\]

This finalizes the analysis of Adapt@Source.

**B. Analysis of Adapt@Network**

The analysis of Adapt@Network follows the same procedures as for Adapt@Source, although with small changes. The decision is now made at the wireless transmitter, which means that the delay in the wired link, \( t_{wd} \), should be subtracted from \( t_d \) in (20), which now only considers the distribution of the delay in the wireless link. Hence,

\[
 P(t_{ws} > t_d - t_{wd}|q \leq 7) = \int_{t_d - t_{wd}}^{\infty} f_{t_{ws}}(t_{ws}) dt_{ws} \tag{23}
\]

Moreover, an integration over all possible \( t_{wd}^i \), \( i = 1, 2 \), should be performed in (9) in order to account for their impact on the prospective compression decision. Hence,

\[
 E[d|F_1, F_2] = \min_c \int_0^\infty \int f_{t_{ws}}(t_{ws}) f_{t_{wd}}(t_{wd}) dt_{ws} dt_{wd} \tag{24}
\]

Finally, the value of \( n \) from (9) is calculated by only taking the delay in the wireless link into account. Hence, \( n = \frac{\mu_{\text{th}}}{\mu_{\text{th}}} \).

**V. Results**

For the evaluation of the proposed schemes a simulation has been implemented in Matlab. This serves the purpose of verifying the results found through analysis in section IV. The simulation is based on the models described in section II as the analysis. In this evaluation the transmission of a 1 Mbit/s video stream is simulated. The source encoder allocates 0.5 Mbit/s to both the base layer and the refinement layer, i.e. \( r_1 = r_2 = 0.5 \cdot 10^6 \). The GOFL length is 1 s. and the resolution is 480x320 at 30 frames/s. The wired part of the network is assigned a mean end-to-end bandwidth of 10 Mbit/s and the wireless part a symbol rate of 256-1024 symbols/s. The adaptive modulation chooses between 4QAM, 16QAM, 64QAM and 256QAM, which result in selectable bit rates of 512 kbit/s, 1 Mbit/s, 1.5 Mbit/s and 2 Mbit/s. The MTU of the wireless link is assumed to be 1500 bytes, which is the maximum value in UMTS. A feedback packet is assumed to have a size of 100 bytes. The sum of the propagation delay and the queuing delay of the feedback packet is assumed to be 50 ms. The FSMC model of the wireless channel is created with 8 states and a time step size of 1 ms. A maximum Doppler shift of 10 Hz is used, which corresponds to a walking speed of 6 km/h at a carrier frequency of 1.8 GHz.

The expected distortion measured as the mean squared error (MSE) is shown in Fig. 3 for the analyzed schemes. The figure shows that both adaptive schemes outperform the static schemes at any SNR. This is expected, since taking feedback into account should only improve performance. However, it is an interesting observation that adapting at the source only results in a very small performance gain compared to the Static MDC scheme, whereas moving the adaptation to the network provides a higher gain.

It should be noted that the gain observed for the proposed schemes over Static MDC is actually achieved by decreasing the average consumed rate in the network. This paradox is explained by the increased probability of satisfying the real time constraints in the video stream when compression is performed. For this reason, a more fair performance metric is one which relates the achieved quality to the amount of transmitted bits. We define such a performance metric as \( \text{PSNR}_{\text{bit}} \), where \( \text{PSNR} = 10\log_{10}(\text{MSE}^{-1}) \). The number of bits is measured in the wireless link, since this is the bottleneck, and at the application level, i.e. retransmissions are not taken

\[
 \text{PSNR}_{\text{bit}} = 10\log_{10}(\text{MSE}^{-1}) \tag{25}
\]
Fig. 3. Simulated performance of the two proposed schemes and the two reference schemes.

Fig. 4. Performance of the investigated schemes with PSNR\textsubscript{bit} as performance metric.

Fig. 5. Comparison of simulated results and the results from the analysis.

A comparison of the results from the analysis and the simulation results is shown in Fig. 5. This figure shows that the simulation verifies the expressions derived in section IV. The results in this section confirm the hypothesis that, in the investigated scenario, informed decisions in the network outperform both informed and uninformed decisions at the source. Hence, it is better to be adaptive based on strong feedback over a smaller part of the network, than it is to be adaptive based on weak feedback over the entire network. Adapt@Network is a simple scheme, which provides just that.

**VI. CONCLUSION**

In this paper two schemes for adaptive multi path video streaming has been presented. One scheme adapts at the source based on end-to-end feedback, while the other scheme adapts in the network based on single link feedback. The analysis of the two schemes gives an insight into the trade-off between adaptation over a larger part of the network and adaptation based on stronger feedback information. Results show that there is a gain in moving the adaptation of the source code away from the source. This is usually a task with high requirements to the network nodes performing the adaptation, but the scheme proposed in this paper utilizes a very simple approach based on compression of descriptions. This scheme achieves a considerable gain over the static reference schemes.

**REFERENCES**


