Multiple-Description Coding for Overlay Network Streaming

Guijin Wang, Satoshi Futemma, and Eisaburo Itakura
Sony, Japan

We consider the problem of distributing video data from one sender to a population of interested clients. Deploying an overlay network with an application-layer multicast, we propose a scalable forward-error-correction-based, multiple-description coding packetization scheme to achieve optimal performance for each client.

In recent years many have studied how to efficiently distribute video data from one sender to a population of interested receivers. For example, Japan NHK considered delivering TV content to thousands of homes via broadband Internet (see http://www.nhk.or.jp/strl/group/net/net_03.html) and Microsoft's CoopNet project conducted research on distributing media content via peer-to-peer (P2P) communication.[1]

Without widespread support for Internet Protocol (IP) multicasting, we can sort most solutions into two categories: overlay/P2P network and media-processing techniques. Generally the sender and all the receivers (which we call internodes or leaves in Figure 1) make up one logical streaming network in the application layer. Such application-layer multicasting takes advantage of distributed network resources and greatly alleviates the network's burden. As opposed to lower-layer network elements such as routers, each node in the overlay has capabilities far beyond basic operations like storing and forwarding.[2] It follows, then, that more operations can occur in these internodes to improve the system's overall performance.

Media processing is related not only to source compression technology, but also packetization and channel coding to combat network packet loss. In this article, we focus on packetization technology for overlay network streaming and introduce our optimized scheme for this—a scalable forward-error-correction (FEC)-based multiple description coding—that we call SMDC.

Video packetization is closely connected with channel coding. Along with the information bits, parity must be generated and accompanied to help the receiver recover the packet loss. Many techniques have been proposed to address the transmission of the scalable bitstream over packet erasure networks. For a brief overview of these techniques, see the "Other Techniques" sidebar.

We give a brief description elsewhere[3] of SMDC's framework for overlay network streaming. In this article, we provide a more detailed explanation and extensive analysis of this SMDC scheme. Specifically, between the source bit and packet, we introduce one data structure that we call a coded unit (CU).

Our framework transforms this scalable bitstream into a CU stream in each layer only once. The framework can then pack this CU stream into output packets many times for different heterogeneous receivers by using additional operations, such as assembling and truncating. In our simulations, we found that the speed of the CU packetizer can sometimes be 32 times faster than that of channel encoding. Compared to a traditional end-to-end multiple-description FEC (MDFEC) coding scheme, our combined SMDC and overlay scheme can adapt to and achieve optimal performance for each heterogeneous receiver. (To see what work we primarily referenced when deciding the best approach for our SDMC scheme, see the "Referenced Work" sidebar, page 76.)

Scalable multiple-description coding

In describing the construction of an SMDC packetization scheme, we can basically regard SMDC as a channel-coding embedded packetizer. At first, the packetizer transforms each layer of the compressed bitstream to n_i CUs by the (n_i, k_i) erasure channel codec. According to our observed network conditions, the packetizer picks up some CUs to form the group of packets (GOP). The distinguished feature of MDFEC—its scalability—results from the CU stream being generated only once, yet it can be packed many times into different GOPs for heterogeneous network conditions. Thus, it provides an easy way for an internode in the overlay network to adapt its output GOP to meet each child. Before further

1070-986X/07/$25.00 © 2007 IEEE

Published by the IEEE Computer Society

Other Techniques

In McCanne et al.,[1] a layered protection scheme was proposed together with the scalable bit streams to tackle the problem of heterogeneity and ensure graceful quality degradation. However, the performance is sensitive to the packet loss' position. To overcome this problem, recently researchers[2-4] have studied priority encoding transmission, which assigns a different protection degree to different layers based on their importance.

Majumdar et al.[5,6] used a min–max regret criteria to formulate the problem of real-time video multicasting with one stream. However, this approach doesn't offer a good tradeoff for the distortions seen by clients. Combining the concept of layered protection and multiple-description, forward-error-correction coding (MDFEC), Chou et al.[7] explored constructions of layered multiple-description coding (MDC) to achieve robustness to unreliable channels and adaptivity to heterogeneity. In their construction, high-bandwidth users pay a performance penalty relative to a nonlayered MDC case. We summarize the transmission procedure for this method in Figure A.

As Figure A shows, at first the source bitstream is packed into packets in a data-packing module; then the system generates parity packets in a channel-coding module. Both modules are controlled by the same vector $(N, \mathbf{K})$, where N is the number of total packets and $N - \mathbf{K}$ denotes the protection degree of each layer. Such a packetizer is quite sensitive to network conditions. Given the fixed average packet length, N is decided by the available network bandwidth while the degree of error protection of each layer $N - \mathbf{K}$ is affected by the network's packet loss ratio. That is, the packetizer must handle channel coding and the data-packing process for each different network condition. Thus, such a packetizer might not be friendly to any emerging overlay network streaming, where the sender or internode might have several child nodes with different network conditions.

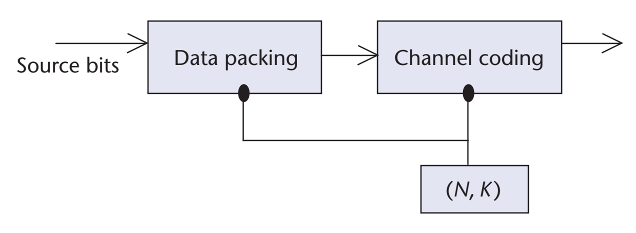


Figure A. Illustrative diagram of a traditional packetizer.

References

1. S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-Driven Layered Multicast," *Proc. ACM Special Interest Group in Comm.*, ACM Press, 1996, pp. 117-130.
2. Q. Zhang, G. Wang, et al., "Robust Scalable Video Streaming over Internet with Network-Adaptive Congestion Control and Unequal Loss Protection," *Proc. PacketVideo Workshop*, 2001.
3. A.E. Mohr, E.A. Riskin, and R.E. Ladner, "Graceful Degradation over Packet Erasure Channels through Forward Error Correction," *Proc. Conf. Data Compression*, 1999, pp. 92-101.
4. R. Puri et al., "An Integrated Source Transcoding and Congestion Control Framework for Video Streaming in the Internet," *IEEE Trans. Multimedia*, vol. 3, no. 1, 2001, pp. 18-32.
5. A. Majumdar et al., "Multicast and Unicast Real-Time Video Streaming over Wireless LANs," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 6, 2002, pp. 524-534.
6. A. Majumdar et al., "Robust Video Multicast under Rate and Channel Variability with Applications to Wireless LANs," *Proc. IEEE Int'l Symp. Circuits and Systems*, IEEE CS Press, vol. 1, 2002, pp. 49-52.
7. P.A. Chou, H.J. Wang, and V.N. Padmanabhan, "Layered Multiple Description Coding," *Proc. PacketVideo Workshop*, 2003; http://research.microsoft.com/~pachou/publications.htm.

discussing the details of how this works, we list frequent notations in Table 1 (page 76).

SMDC architecture at the sender

Figure 2 (page 77) shows how the SMDC structure at the sender generates an output GOP from the original compressed bitstream. On the sender's side, first the media data generator compresses the raw media signal into a layered, scalable, compressed bitstream. Then the scheme partitions each layer into k_i source units, which it can expand into n_i CUs in a CU-generation module. Both modules are controlled by a vector parameter $(\mathbf{n}, \mathbf{k})$. This vector determines the CU number and network conditions that the generated CUs can cover. Figure 2's shadow region is the packing part for each directed child node. The CU data are multiplexed into N packets, which are controlled by the rate allocation parameters $(N', \mathbf{K})$. The vector $(N, \mathbf{K})$ depends on the observed network status between each receiver and the sender. We assume that the network condition can be well estimated[4] and focus on the main modules of SMDC construction:

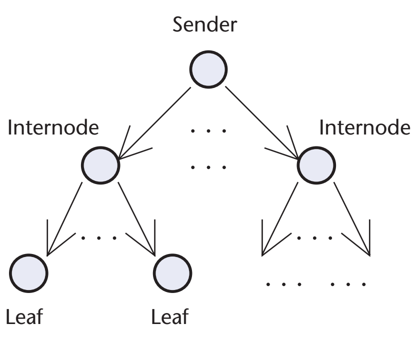


Figure 1. Illustrative topology of an overlay network.

Referenced Work

Many methods of multiple-description coding have been developed over the years. One particularly efficient method is multiple-description, forward-error-correction coding (MDFEC) proposed in Mohr et al.[1] and Puri et al.[2] that combines layered source coding with unequal erasure protection. As Figure B illustrates, this packetization scheme partitions layered bit streams into groups of packets (GOP). Each layer i is then separated into K_i source blocks, and K_i source blocks are expanded into N blocks using an (N, K_i) channel error-erasure code across the blocks.

In Figure B, note the boundaries of layer i in the encoded bitstream by R_{i-1} and R_i. For the GOP packetized by MDFEC, if any K out of N packets are received, then the scheme can recover the first i layers where

$$i = \arg\max_j \{K_j \le K\}$$

Meanwhile, the latter layers are lost, resulting in distortion $d(R_i)$. In this sense, each packet holds the same importance; the number of received packets determines the reconstructed quality. By using unequal error protection, MDFEC generates N multiple descriptions, with each packet constituting one description.

The MDFEC developed in Puri et al.[2] performs well for one-to-one streaming applications. Extended by Majumdar et al.'s approach,[3] we propose a min–max criteria to address the problem of multicasting video:

$$\Delta(R) = \max_m (E[D_m(R)] - E[D_m]_{\min})$$

where R is the channel capacity, $E[D_m]_{\min}$ is the minimal expected distortion for the mth-class client achieved by using optimal scheme when it's the only class, and $E[D_m(R)]$ is the expected distortion for the particular coding scheme being used. Such quality criterion minimizes the maximum penalty that any client suffers.

References

1. A.E. Mohr, E.A. Riskin, and R.E. Ladner, "Graceful Degradation over Packet Erasure Channels through Forward Error Correction," *Proc. Conf. Data Compression,* 1999, pp. 92-101.
2. R. Puri et al., "An Integrated Source Transcoding and Congestion Control Framework for Video Streaming in the Internet," *IEEE Trans. Multimedia*, vol. 3, no. 1, 2001, pp. 18-32.
3. A. Majumdar et al., "Multicast and Unicast Real-Time Video Streaming over Wireless LANs," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 6, 2002, pp. 524-534.

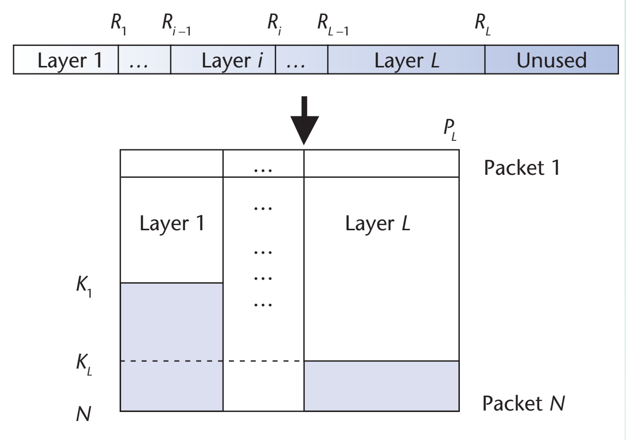


Figure B. Traditional packetization with multiple-description, forward-error-correction coding (MDFEC).

Table 1. List of notations used.

Notation	Description
U_{ij}	jth source unit in the ith layer
CU_{ij}	jth coded unit (CU) in the ith layer
n_i	Total number of CUs in layer i
k_i	Total number of source units in layer i; (n_i, k_i) are the erasure channel coding parameters
(**n**, **k**)	Vector of erasure channel parameters of all layers
real_n_j	Number of delivered CUs in layer i
N_1	Number of packets in the GOP
r	Number of lost packets
K_1	Number of information blocks in layer i $\mathbf{K} = \{K_1, \ldots K_i, \ldots\}$
L	Number of layers to be delivered
R_i	Length up to the end of layer i
B_i	Length of layer $i = (R_i - R_{i-1})$ R bit budget

- the source unit and CU generator,
- CU packetizer, and
- rate allocation.

Source generator and coded-unit generation

For input data, we used a layered scalable bitstream, like MJPEG-2000 (encoding each frame with JPEG-2000)[5] or MPEG4-FGS.[6] We used bit-plane entropy coding to generate an embedded stream with several quality layers. In this way, the data correlates in each layer. To be specific, the higher-layer information relies on the corresponding ones in the lower layers.

Figure 3 explains the details of how to generate CUs in each layer, where 1 is the first symbol and B_i is the last symbol of layer i. The symbols are put

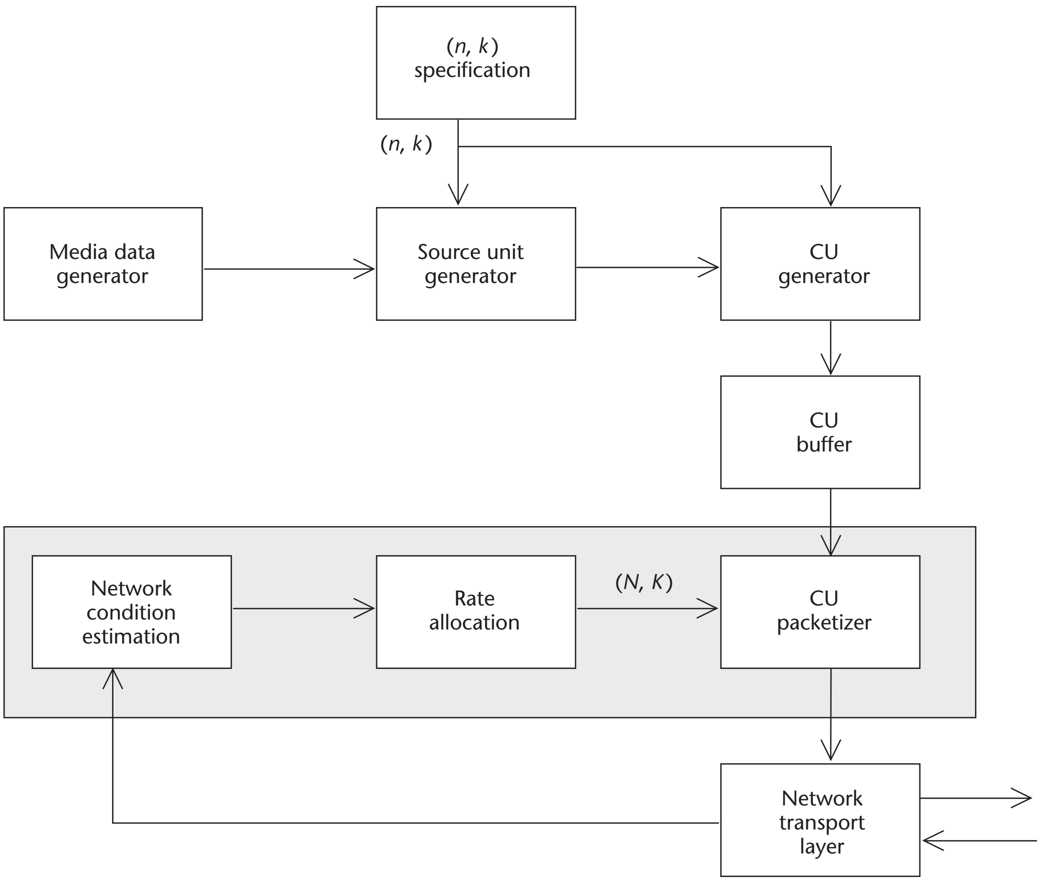


Figure 2. Scalable, forward-error-correction-based multiple-description coding (SMDC) packetizer at the sender. (CU stands for coded units.)

across the k_i units one by one. Then, we generate n_i CUs by applying (n_i, k_i) erasure channel codes along the column (one symbol for each unit).

In this article, we adopted Reed-Solomon codes[7] as the channel codes, which can be replaced by any other error-erasure codec like LT codes.[8] When the network experiences a lower packet loss ratio (PLR), the system doesn't need to deliver all the CUs. The actual transmitted number of CUs is less than n_i.

On the other hand, when less bandwidth is available, we can also truncate the CU at the *j*th symbol to fit the constraints of the channel's capacity. This module is controlled by the pair (n_i, k_i), which decides the domain of network conditions that the generated CUs can deal with. Here k_i is the least CU number to restore the source bits of layer *i*, and $n_i - k_i$ bounds the generated SMDC codes' protection capability.

After establishing a fixed PLR and averaged packet size, the maximal bandwidth that SMDC can deal with is bounded when the minimal ratio k_i/K_i among all the layers equals 1. With the available bandwidth fixed, the worst PLR to be covered is limited when any layer runs out of its redundant $n_i - k_i$ CUs.

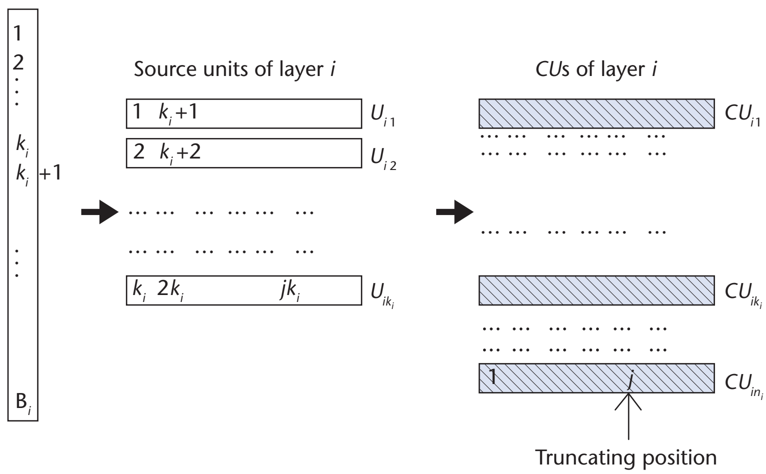


Figure 3. Generating CUs.

As we mentioned previously, the scalable grain of layer ***i*** depends on the CU length, where truncation occurs only at established $\lceil B_i/k_i \rceil$ points in each column. Taking both factors (the scalable grain and domain of network conditions) into account, it's better if the CU length lies in the range of 10 to 100 bytes. If B_i is too small, the scheme can merge layer ***i*** with the next layer to get one new layer bitstream. If B_i is too large, it can break this layer into two layers accordingly.

Rate allocation

As Figure 2 shows, the greatest benefit of our SMDC scheme is that the bit allocation process can be independent from channel coding (by using the CU generator) so that it's possible for the internode or the sender to reuse the CUs. Of course, we achieve this flexibility at the cost of additional bits when k_i is not the integer times K_i.

For the constrained resource budget R, we pick a fixed average packet length P_L and decide the GOP size $N = R/P_L$. If any K out of the N packets are available, then we can recover all the layers i where $K_i \le K$, resulting in a distortion $d(R_i)$. For the lost packets, we first see if we can recover them in the first layer, then check the second layer, and so on. The distortion can be the sum of the distortion with each packet loss pattern:

$$D(R) = \sum_{i=0}^{L} \{ \sum_{r=N-K_{i+1}+1}^{N-K_i} P(N,r) d(R_i) \} \tag{1}$$

where $P(N, r)$ is the loss probability of r out of N packets, R_i is the number of bits at the end of layer i, and $d(R_i)$ is the source distortion when the scalable bit stream is truncated at the point of R_i. For simplicity, we adopt Mallat's and Falzon's distortion model $y = Cx^{1-2y}$ for calculation,[9] where C is a positive number and y is of the order of 1, and x is the bit rate by the unit bit per pixel (bpp). The overall distortion expression in Equation 1 is constrained by the network capacity

$$R_L + R_c = R_L + \sum_{i=1}^{L} \frac{R_i - R_{i-1}}{k_i} \left(N - K_i\right) \left\lceil \frac{k_i}{K_i} \right\rceil \le R \tag{2}$$

where $(R - R_{i-1})/k_i$ is the data length of CU in layer i, and R_c is the bits used for the redundancy. Observe that in the distortion formulation R_i $(i < L)$ is the constant, so Equation 1 can be reexpressed in Equation 3 as

$$\begin{aligned} D(R) = d(R_L) &+ \sum_{r=N-K_L+1}^{N} P\left(N,r\right)\left(d\left(R_{L-1}\right) - d\left(R_L\right)\right) \\ &+ \sum_{r=N-K_i+1}^{N} P\left(N,r\right)\left(d\left(R_{i-1}\right) - d\left(R_1\right)\right) + \ldots \\ &+ \sum_{r=N-K_i+1}^{N} P\left(N,r\right)\left(d\left(R_0\right) - d\left(R_1\right)\right) \end{aligned} \tag{3}$$

where the first item denotes the source distortion. The left i item,

$$d_c(i) = \sum_{r=N-K_{i+1}}^{N} P\left(N_1,r\right)\left(d\left(R_{i-1}\right) - d\left(R_i\right)\right)$$

represents the channel distortion related to the protection degree of the ith layer. In this expression, there are $L + 1$ variables: the end of the last layer's boundary and the number of source blocks $K_1 \ldots K_{L_1}$ $(K_0 = 0)$. The problem is to arrive at the optimal variable sets to obtain the minimal distortion (Equation 3) based on the network capacity constraints (Equation 2).

This optimal problem can be solved with an iterated Lagrange algorithm. Here we would provide another suboptimal-but-fast method to decide the rate allocation parameters. Let's analyze the first order derivative of $D(R)$ on K_i

$$\frac{\partial(D(R)}{\partial K_i} = \frac{\partial d\left(R_{L_1}\right)}{\partial K_i} \times \frac{\partial R_{L_i}}{\partial K_i} + \frac{\partial d_c(i)}{\partial K_i}, i = 1, \ldots, L_1 \tag{4}$$

If the derivative in Equation 4 is larger than 0, it means that a greater protection degree $(N - K_i)$ is needed to reduce the overall distortion. That is, when the derivative is less than 0, the protection of layer i is robust to the packet loss. By adding one upper bound of

$$\left|\partial d\left(R_L\right)/\partial R_L\right|, \left|\partial d\left(R\right)/\partial R\right|$$

we can arrive at the suboptimal solution:

$$\begin{aligned} \hat{K}_i = \underset{\max\left(K_i^l\right)}{\arg} &\left(\left| \frac{\partial d_c(i)}{\partial K_i} \times \frac{\partial K_i}{\partial R_{L_i}} \right| \le \left| \frac{\partial d(R)}{\partial R} \right|, \ \forall K_i \le K_i^* \right) \\ &\text{subject to } \hat{K}_i \le \hat{K}_j, \qquad \forall i \le j \end{aligned} \tag{5}$$

It's easy to confirm that the derivative $\partial(D(R)/\partial \hat{K}_i^l$ is less than 0. The resource allocation problem is reduced to several one-dimension search problems. We can figure out the optimal solution $\hat{K}_i$ as the maximum number fit (Equation 5). And the final information bytes R_L can be directly calculated from Equation 2.

CU packetizer

The CU packetizer processes each child node. Figure 4 depicts the illustrative diagram of a GOP after using the CU packetizer. In Figure 4, there are four data structures: CUs, blocks, packets, and GOPs. Similar to the MDFEC, all the layers are multiplexed into GOPs with resource allocation parameters (N, K_1, …, K_L). N is the number of packets in the GOP and $N - K_i$ is the maximal packet loss that layer i can bear.

Each packet is separated into L blocks, one block per layer. In layer i, k_i CUs are packed into K_i blocks. The CU number of layer i in one block would be either $\lceil k_i / K_i \rceil$ or $\lfloor k_i / K_i \rfloor$, where $\lceil \alpha \rceil$ is the upper-floor function, and $\lfloor \alpha \rfloor$ is the under-floor function of float number α. In the left $N - K_i$ gray block, the CUs number picks up the larger number $\lceil k_i / K_i \rceil$. Thus, in Equation 6, the number of CUs of layer i in the mth block/packet, NU_{im}, equals

$$NU_{im} = \begin{cases} \lceil k_i/K_i \rceil & 0 < m \le (k_i \bmod K_i) \\ \lfloor k_i/K_i \rfloor & (k_i \bmod K_i) < m \le K_i \\ \lceil k_i/K_i \rceil & K_i < m \le N \end{cases} \quad (6)$$

where m is the index of the block/packet.

We can consider the CU packetizer as a transcoder, transcoding the CUs to different GOPs according to the network conditions of each child node. The transcode operation here refers to deciding the number of real delivered CUs, real_n_i, and the length of CUs in each layer. The real_n_i equals $K_i + (N - K_i) * \lceil k_i / K_i \rceil$. The CU of the last layer must then be shortened to meet the available network bandwidth.

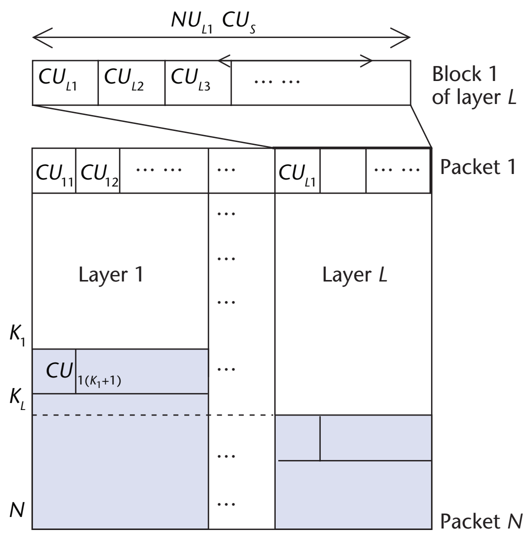


Figure 4. An SMDC group of packets (GOP) after accessing the CU packetizer.

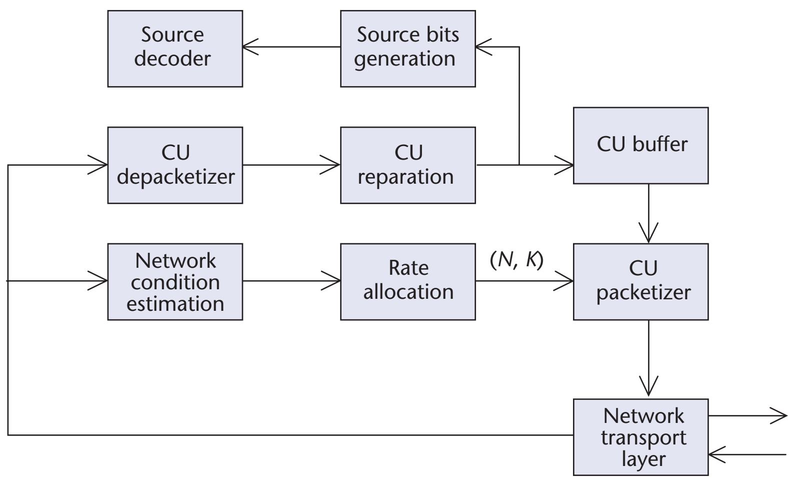


Figure 5. Illustrative SMDC framework at the internode.

SMDC architecture at the internode

In the overlay network, the internode serves two tasks. As the receiver, the internode reconstructs the video signal from the received packets. It's also responsible, though, for forwarding the video to its child nodes.

Figure 5 shows the SMDC architecture at the internode side. At first, the internode unpacks the received packets into the CUs of each layer and recovers lost CUs. Then the procedure divides into two paths. Along one path, the restored CUs reorganize into a compressed bitstream for the source decoder. Along the other transcode path, the internode truncates and repacks CUs based on resource allocation parameters (N, $\mathbf{K}$) of each child node, which it can decide quickly with our suboptimal (but fast) rate allocation algorithm. Each receiver can take advantage of local network resources and enjoy high video performance.

Simulation results

We tested the SMDC extensively to demonstrate its performance. We transformed raw images, with dimensions of 720 × 576 pixels, into seven scalable quality-layer bitstreams, compressing the first layer at 0.0625 bpp. We made this compression using the following Kakadu codec (more information on the Kakadu codec is available at http://www.kakadusoftware.com) as a command option:

```
kdu_compress –i ∞@xxx.raw -o xxx.j2c
Clayers=7 -rate -,0.0625 Sdims={576,720}
Ssigned=no Sprecision=8 Cuse_eph=yes
Cuse_sop=yes
```

We conducted the simulations under the total bits budget varying from 100 to 400 kilobits (Kbits). We set the average packet size as 512 bytes.

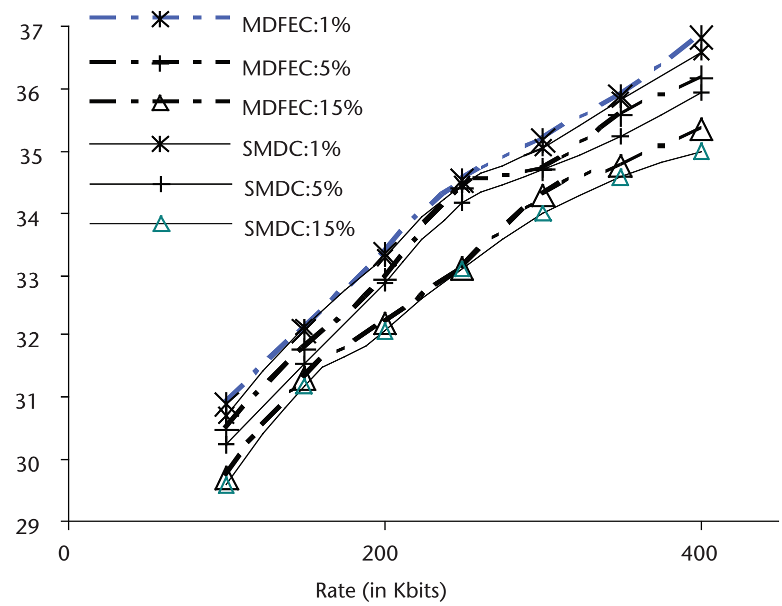


Figure 6. Average peak signal-to-noise ratio (PNSR) for an image under different channel conditions.

We modeled the network as an independent packet erasure channel. Taking into account practical Internet scenarios, we set the PLR as three numbers—0.01, 0.05, and 0.15—denoting low, middle, and high packet loss cases, respectively. We obtained the following results by averaging the peak signal-to-noise ratio (PSNR) of the reconstructed image over 200 Monte Carlo simulations.

Rate-distortion performance

Because of space limitations, Figure 6 shows our SMDC rate-distortion performance for an image under three lossy channels. Similar results can be observed for other images that we tested.

We compared our results to the scheme, where the channel coding and packetization process must be made for each channel condition with the optimal rate allocation parameters. In the SMDC scheme, the error-erasure codec (n_i, k_i) transforms each layer into CUs only once. In this work, we paired the parameters of all layers, fixing and setting them at (255, 153), which would deal with all the loss cases in the simulations. We obtained similar performances when we tried other pair parameters, like (255, 150), (255, 160), and so on. Based on the optimal rate allocation, the scheme packs and truncates CUs accordingly for each network condition (in terms of the bit budget and PLR).

On the rate-distortion performance side, SMDC can achieve comparable PSNRs with MDFEC, with no more than 0.3 decibels (dB) of degradation under any PLR and bit budget. The CU packetizer induces the cost when k_i isn't the integer times K_i. But it should be noted that in the SMDC scheme, the CU packetizer transforms the same generated CU data for each tested case.

Table 2 shows the running time of channel encoding in the MDFEC and transcoding in the SMDC on a Pentium IV 3.0-GHz computer. In the SMDC, the transcoding time keeps nearly constant and low (at around 0.62 ms) no matter what the bit budget and PLR are. Meanwhile, for the MDFEC the running time of channel encoding[7] increases with the bit-budget and PLR. At 100 Kbits and a 1-percent PLR, the running time of the MDFEC is nearly three times that of the SMDC; and the ratio can be up to 32 times faster, as is the case with 400 Kbits and a 15 percent PLR.

Table 3 reveals more details of the SMDC packetization procedure. The set of coding parameters includes the allocation (N, K_i), the actual delivered number of CU real_n_i, the CU length of each layer, and the CU number per block. The three GOPs are constructed by the same copy CUs, but with different allocation parameters (N, K_i). For instance, at 100 Kbits with a PLR of 5 percent, the first layer utilizes 207 out of 255 CUs and has nine CUs in one block; meanwhile, at 400 Kbits with a PLR of 10 percent, the first layer utilizes more CUs (231 out of 255) because of the high PLR and has three CUs in one block because of the large size of the GOP. By looking at Table 3, then, we can discern the original CU length of each layer (for instance, the CU length of the third layer is 45 bytes). To fit the constraint of the bit budget, however, the third layer is truncated to 18 bytes.

Table 2. Running time in milliseconds (ms) of channel encoding in the MDFEC and transcoding in the SMDC with packet loss rates (PLRs) of 1 and 15 percent.

	SMDC (ms)		MDFEC (ms)	
Size (in Kbits)	PLR: 1%	PLR: 15%	PLR: 1%	PLR: 15%
100	0.62	0.62	1.72	7.82
400	0.62	0.62	3.91	20

Heterogeneity performance

We can now examine the SMDC performance with heterogeneous receivers. Figure 7 illustrates one simple overlay network. In the gray region, the first hop 1 → 2 experiences a 10-percent PLR and the second hop 2 → 3 experiences a 5-percent PLR. If we consider the compared MDFEC scheme without overlay streaming, the bit-budget equals the lower one of the two hops (200 Kbits) and the

Table 3. Coding parameters of the SMDC of an image.

Layer	100 Kbits N = 24 packets, PLR = 5%				200 Kbits N = 48 packets, PLR = 5%				400 Kbits N = 97 packets, PLR = 10%			
	K_i	real_n_i	CU Length (Bytes)	$\left\lceil \frac{k_i}{K_i} \right\rceil$	K_i	real_n_i	CU Length (Bytes)	$\left\lceil \frac{k_i}{K_i} \right\rceil$	K_i	real_n_i	CU Length (Bytes)	$\left\lceil \frac{k_i}{K_i} \right\rceil$
1	18	207	22	9	37	208	22	5	71	231	22	3
2	19	198	23	9	39	189	23	4	73	225	23	3
3	20	185	18	8	40	185	45	4	77	193	45	2
4	—	—	—	—	42	177	42	4	77	193	90	2
5	—	—	—	—	—	—	—	—	80	187	72	2

experienced PLR becomes 14.5 percent (1 – 0.9 * 0.95), assuming that the end-to-end path between 1 and 3 is the combination of the two hops.

With the MDFEC optimized for 200 Kbits with a PLR of 14.5 percent, nodes 2 and 3 are each expected to achieve 32.32 dB. In the SMDC overlay streaming, we take advantage of the overlay distributed streaming and the SMDC packetization scheme. Table 3 lists the coding parameters of nodes 2, 3, and 4.

Node 2 acts as the internode. It unpacks received packets into CUs of each layer and recovers the lost CUs. Along the transcoding path, the restored CUs are truncated and packed again into the new GOP for node 3. In the first four respective layers, the scheme extracts the following CUs: 208 out 231, 189 out of 225, 185 out of 193, and 177 out of 193. The CUs in the last layer are shortened from 77 to 42 bytes.

Because node 2 transcodes the larger GOP with 97 packets into smaller GOPs with 48 packets, we correspondingly adjust the CU number in one block. For instance, the number in one block of the first layer is set as five in the second hop, different from three in the first hop. Taking advantage of overlay streaming, the SMDC scheme can perform at up to 34.30 dB for node 2 and 32.88 dB for node 3, achieving 1.98 dB (for node 2) and 0.56 dB (for node 3) in gain compared to the MDFEC scheme.

Among nodes 2, 3, and 4, 3 and 4 have a different available bit budget (100 and 200 Kbits, respectively) and experience the same PLR of 5 percent. That is, each user can receive at most 24 or 48 packets, respectively, given that the packet size is 512 bytes.

Similarly, we can use a comparative scheme called the weighted MDFEC. In this scenario, node 2 transcodes the received packets to the appropriate GOP and sends them to each receiver. Nodes 3 and 4 can achieve 32.88 dB and 30.25 dB. In this compared scheme, the rate allocation parameters are averaged as the candidate and the same GOP is sent to nodes 3 and 4. To meet this heterogeneity, the scheme only delivers 150 Kbits (36 packets). Because of the channel capacity, node 4 can at most receive 24 packets, and can't reconstruct any image data. On the other hand, node 3 receives 36 packets, doesn't fully use the available network resource of 48 packets, and this results in a 1.11-dB degradation compared to our SMDC overlay streaming scheme.

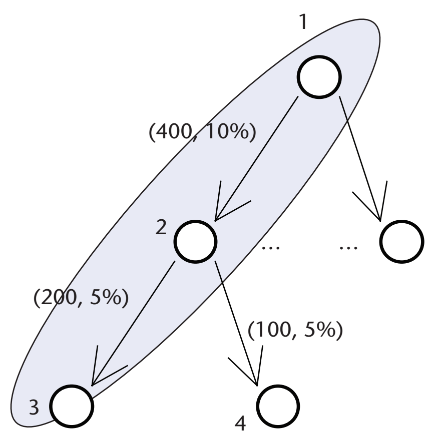


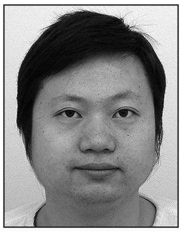
Figure 7. Simple overlay network with heterogeneity.

Future work

In this article, we proposed a scalable FEC-based multiple-description coding (SMDC) architecture for distributing video data from one sender to a population of interested receivers. Compared with traditional technologies, our scheme favors overlay network streaming, where the internode can easily reorganize the received bitstream for its child nodes so as to achieve optimal performance for each client. In the future, we'll consider how to apply our SMDC coding scheme in a wireless mesh network. **MM**

References

1. V.N. Padmanabhan, H.J. Wang, and P.A. Chou, "Resilient Peer-to-Peer Streaming," *Proc. IEEE Int'l Conf. Network Protocols*, IEEE CS Press, 2003, pp. 16-27.
2. Y. Zhu, B. Li, and J. Guo, "Multicast with Network Coding in Application-Layer Overlay Networks," *IEEE J. Selected Areas in Comm.*, vol. 22, no. 1, 2004, pp. 107-120.
3. G. Wang, S. Futemma, and E. Itakura, "FEC-Based Scalable Multiple Description Coding for Overlay Network Streaming," *Proc IEEE Consumer Comm. and Networking Conf.*, IEEE CS Press, 2005, pp. 406-410.
4. J.-C. Bolot and T. Turletti, "Adaptive Error Control for Packet Video in the Internet," *Proc. IEEE Int'l Conf. Information Processing,* IEEE CS Press, vol. 1, 1996, pp. 25-28.
5. W. Yu, R. Qiu, and J. Fritts, "Advantages of Motion-JPEG2000 in Video Processing," *Proc. Int'l Soc. for Optical Eng.* (SPIE), *Electronic Imaging 2002,* SPIE Press, 2002; http://www.enee.umd.edu/~weiyu/homepage/research.html.
6. W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, 2001, pp. 301-317.
7. S. Lin and D.J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, 1983.
8. M. Luby, "LT Codes," *Proc. 43rd Ann. IEEE Symp. Foundations of Computer Science*, IEEE CS Press, 2002, pp. 271-280.
9. S. Mallat and F. Falzon, "Analysis of Low Bit Rate Image Transform Coding," *IEEE Trans. Signal Processing*, vol. 6, no. 4, 1998, pp. 1027-1042.



Guijin Wang is an associate professor in the Electronic Engineering Department at Tsinghua University, China. His current research interests include wireless ad hoc networks, mesh network streaming, overlay networks, and video coding and processing. Wang received a PhD in electronic engineering from Tsinghua University and is an IEEE member.

Satoshi Futemma is a researcher at the Information Technology Laboratories at Sony. His research interests include audiovisual communications among wired and wireless networks, error control in real-time communications, and high-quality video communications. Currently, he's also very interested in JPEG-2000 video communications. Futemma received his PhD in information science from the Nara Institute of Science and Technology in Nara, Japan.

Eisaburo Itakura is currently a researcher and senior manager in the IPELA division of Sony. He's also working on his doctorate at the Graduate School of Informatics at Kyoto University, Japan. His research interests include audiovisual communication systems over the Internet, and combining quality-of-service and codec technologies.

Readers may contact Guijin Wang at wangguijin@tsinghua.edu.cn.

For further information on this or any other computing topic, please visit our Digital Library at http://www.computer.org/publications/dlib.