

UC San Diego

UC San Diego Previously Published Works

Title

Prioritized Packet Fragmentation for H.264 Video

Permalink

<https://escholarship.org/uc/item/2xv8b2gb>

Authors

Kambhatla, K R

Kumar, S

Cosman, P

Publication Date

2011-09-01

Peer reviewed

Prioritized Packet Fragmentation for H.264 Video

Kashyap K. R. Kambhatla

Electrical and Computer Engineering
San Diego State University, San Diego
Email: kashyap3881@gmail.com

Sunil Kumar

Electrical and Computer Engineering
San Diego State University, San Diego
Email: skumar@mail.sdsu.edu

Pamela Cosman

Electrical and Computer Engineering
University of California, San Diego
Email: pcosman@eng.ucsd.edu

Abstract—We introduce a cross-layer priority-aware packet fragmentation scheme to enhance the quality of H.264 compressed bitstreams over bit-rate limited error-prone links in packet networks. The H.264 slices are prioritized in the encoder based on their cumulative mean square error (CMSE) contribution towards the received video quality. Specifically we derive the optimal fragment size for each priority level which achieves the maximum expected weighted goodput at different encoded video bit rates and slice sizes. The packet fragmentation scheme uses slice discard in the buffer. Simulation results show that the proposed scheme provides considerably improved video quality.

Index Terms—H.264, video compression, packet fragmentation, priority-aware, video quality.

I. INTRODUCTION

Lost video packets induce different levels of quality degradation due to temporal and spatial dependencies in the compressed bitstream. This problem has led to the design of error-resiliency features such as flexible macroblock ordering (FMO), data partitioning and error concealment schemes in H.264 [1]. Packet segmentation and reassembly can be carried out at the transport layer of the source and gateway nodes to comply with the maximum packet size requirements of intermediate networks [2]. Van der Schaar et al. [3] demonstrated the benefits of the joint APP-MAC-PHY approach for transmitting video over wireless networks. Since the channel statistics and network information form efficient interface parameters between the MAC and PHY layers, the MAC layer can efficiently take into account the network congestion and transmission opportunities.

Lately there has been increasing effort to adopt packet fragmentation techniques for enhancing H.264 compressed video transmission over wireless networks [4]–[7]. Fallah et al. [4] fragment application layer units formed using dispersed FMO and ‘foreground with left over’ H.264 modes. This is extended in [5] to 3G UMTS networks uplink and downlink. Unlike data applications such as email and FTP file transfers, real-time video can tolerate the loss of some packet fragments and still provide good quality. Data partitioning in H.264 is used to map the 802.11e MAC access categories to different partitions in [6]. Fallah et al. extend this and [4] and employ controlled access phase scheduling [7].

Packet fragmentation at the MAC layer aims to adapt the packet size to the channel error characteristics to improve the successful packet transmission probability and reduce the cost of packet retransmissions. MAC layer fragmentation and retransmission in wireless networks also avoids the costly

retransmissions from transport layers [8], [9]. It is an integral part of the IEEE 802.11 MAC layer [10]. The fragmentation threshold is optimized to maximize the system throughput. This technique calls for a trade-off between reducing the number of overhead bits by adopting large fragments and reducing the transmission error rate by using small fragments. However maximum throughput does not guarantee minimum video distortion at the receiver because lost video packets induce different amounts of distortion in the received video.

In this paper, we propose a cross-layer fragmentation scheme for streaming of pre-encoded H.264 video data. Under known link conditions, we address the problem of assigning optimal fragment sizes to the individual priority packets within the channel bit-rate limitations. The objective is to maximize the expected weighted goodput which provides higher transmission reliability to the high priority packets by using smaller fragments, at the expense of (i) allowing larger fragment sizes for the low priority packets, and (ii) discarding low priority packets when necessary. The Branch and Bound (BnB) algorithm along with an interval arithmetic method is used to find the maximum expected weighted goodput and derive the optimal fragment sizes.

Section II-A introduces the proposed cross-layer video priority packet formation. Priority-agnostic fragmentation is discussed in Section II-B and Section II-C formulates the expected weighted goodput maximization problem. The comparison between priority-aware and priority-agnostic fragmentation appears in Section III. Section IV concludes.

II. PROPOSED CROSS-LAYER FRAGMENTATION SCHEME

A. H.264 slice and video packet formation

In this paper, we consider videos which are pre-encoded using H.264 AVC with fixed slice size configuration; macroblocks are aggregated into a slice such that their accumulated size does not exceed a pre-defined size. The network limits the number of bytes that can be transmitted in a single packet based on the MTU bound. Slices formed at the encoder are aggregated into a video packet for transport over IP networks; each packet is appended with a 40-byte RTP/UDP/IP header. This aggregation helps to control the amount of overhead. When slices are prioritized (in Section II-C), slices of different priority classes are separately aggregated into packets. We use a binary symmetric channel $BSC(p_b)$ where p_b is the BER. The video packets are fragmented at the data link layer using channel BER information (continuously updated from

the PHY layer) and slice priority (from the application layer). Each fragment is attached with MAC and PHY layer headers.

H.264 slices are prioritized based on their distortion contribution to the received video quality. The total distortion of one slice loss is computed using CMSE which takes into consideration the error propagation within the entire GOP. All slices in a GOP are distributed into two priority levels based on their pre-computed CMSE values.

B. Priority-agnostic fragmentation

In conventional packet fragmentation, the data link layer at the receiver expects that erroneous packet fragments will be re-transmitted; the entire packet is discarded if any of its fragments is not received properly. However, such retransmission may not be feasible for real-time video streaming. Since the video bitstream is tolerant of packet losses, the decoder reconstructs the lost packets or fragments using error concealment. Video traffic can also tolerate some slices being discarded to accommodate more fragmentation overhead. In this section, we discuss priority-agnostic fragmentation. The fragment size cannot be smaller than the target slice size and each fragment contains one or more slices in their entirety.

A measure of the reliable transmission of packets over error-prone channels is *goodput*. We define the goodput G as the expected number of successfully received video bits per second (bps) normalized by the target video bit rate R bps. G depends on the fragment success rate (fsr) which is a function of the fragment size (y) and the channel BER (p_b). Though real slice sizes vary, we assume that each slice is x bits long in our theoretical formulation. A fragment is successfully received iff all its bits are received without error. The fsr is:

$$fsr = (1 - p_b)^y, y = nx + h \quad (1)$$

Here, the fragment size is y bits, containing n slices (nx payload bits) and h MAC and PHY header bits. We define F_{TX} as the total number of fragments transmitted during a one second and F_{RX} as the corresponding expected number of successfully received fragments. F_{RX} is computed as $F_{RX} = (fsr)(F_{TX})$. We assume that the channel bit rate is R_{CH} bps, video bit rate is R bps, and $N = R/x$ slices are generated every second. The number of payload bits in a fragment can vary from x to P bits, where P represents the MTU size. Therefore, the feasible number of slices in each fragment varies from 1 to $\frac{P}{x}$. The expected goodput G is computed, after excluding the header bits associated with each fragment, as

$$G = \frac{F_{RX}(y - h)}{R} = \frac{F_{TX}(1 - p_b)^y(y - h)}{R} \quad (2)$$

Here, the objective is to find the optimal fragment size y such that G is maximized:

$$y = \operatorname{argmax}_y G = \operatorname{argmax}_y \frac{F_{TX}(1 - p_b)^y(y - h)}{R} \quad (3)$$

$$F_{TX} = \begin{cases} (\frac{N}{n}); & (\frac{N}{n}) \leq \frac{R_{CH}}{y} \\ \frac{R_{CH}}{y}; & (\frac{N}{n}) > \frac{R_{CH}}{y} \end{cases}$$

If $(\frac{N}{n}) \leq \frac{R_{CH}}{y}$ then sufficient bits are available to allocate headers to all the fragments generated in one second. If $(\frac{N}{n}) > \frac{R_{CH}}{y}$ then headers cannot be allocated to all fragments for a fragment size y . One can compute the number of application layer packets that would be discarded and the corresponding number of discarded slices.

Figure 1(a) shows the variation in expected goodput G for different fragment sizes and channel BERs for a video encoded at $R = 960$ Kbps with 150-byte slices. The channel bit rate R_{CH} is 1Mbps for all cases in this paper. The maximum video data in a fragment is $P = 1500$ bytes. For a fragment of 1500 bytes, the maximum value of G is 55% for $p_b = 5 \times 10^{-5}$ which increases to 98% for a lower channel error rate $p_b = 10^{-6}$, because fsr increases as channel BER decreases. The expected goodput also depends on the number of slices discarded. More slices are discarded as the fragment size decreases since the requirement for header bits increases. For a fragment size of 150 bytes, though fsr is higher than that for larger fragment sizes, the corresponding G is lower. The system can achieve a higher value of G at this BER when the encoding bit rate is lower, as shown in Figure 1(b) for the 720 Kbps video bit rate. There exists an optimal point in each case which trades off losses due to channel errors with packet discards. For example, the maximum value of G is achieved at fragment sizes of 300 and 750 bytes for $p_b = 5 \times 10^{-5}$ and 10^{-5} , respectively.

For three different encoded video bit rates at $p_b = 5 \times 10^{-5}$, Fig. 1(b) illustrates the variation in G for different fragment sizes. For $R = 720$ Kbps, sufficient bits are available to allocate headers to each fragment, and no slices are discarded. Every slice of the video packet can be transmitted independently in a fragment with maximum $G = 93\%$. As R increases, more slices are generated every second. The maximum achievable G decreases as R increases and approaches or exceeds R_{CH} , because fewer bits are available for fragment headers. More header bits can only be accommodated by discarding slices. As a result, the maximum value of G decreases to 77% and 69% for $R = 960$ Kbps and 1080 Kbps, respectively, when each fragment contains two slices. When $R = 1080$ Kbps, exceeding R_{CH} , 14.1 Kbytes worth of slice data is discarded every second even for a 1500-byte fragment size. Though one may be inclined to choose a large fragment size to reduce the number of discarded slices, it also decreases fsr as discussed earlier.

C. Priority-aware fragmentation

We now extend the fragmentation scheme to make it adaptive to packet priorities, assigning smaller fragment sizes to high priority packets to increase their transmission success. The link layer scheduler transmits all high priority fragments before low priority ones every second. We define the expected weighted goodput G_W to be a linear combination of individual priority goodput:

$$G_W = w_1 g_1 + w_2 g_2 \quad (4)$$

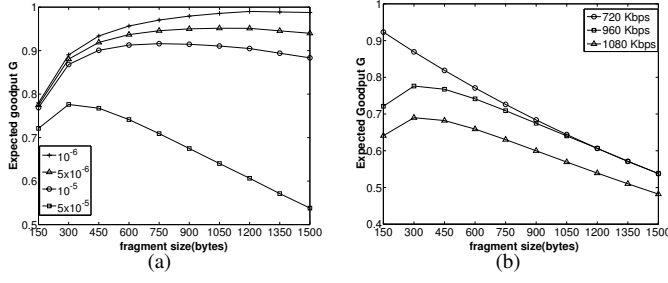


Fig. 1: Expected Goodput G vs. fragment size y . (a) $R=960$ Kbps and $p_b = 10^{-6}$, 5×10^{-6} , 10^{-5} , and 5×10^{-5} for the curves from top to bottom, (b) $p_b = 5 \times 10^{-5}$ and $R=720$, 960, and 1080 Kbps for the curves from top to bottom.

The weights w_1 and w_2 capture the relative distortion contribution per bit from the individual slice priorities. w_1 is computed as the ratio of the mean CMSE of high priority slices to the mean CMSE of all slices in the pre-encoded video, and $w_2 = 1 - w_1$. The CMSE threshold for assigning priority to each slice is the median of all slice CMSE values. The weights depend on this threshold, video content and encoding parameters such as R and slice size x . We define $n_1, n_2 \in [1 \frac{P}{x}]$ as the number of slices that can be aggregated into each fragment of the high priority and low priority packets. The corresponding fragment sizes are $y_1 = n_1 x + h$ and $y_2 = n_2 x + h$ bits. Let N be the fixed total number of slices generated per second, and l_1 and l_2 be the corresponding numbers of slices in the high and low priority packets. So $l_1 + l_2 = N$. If video has high motion activity, there will be more high priority packets. We assume l_1 is uniformly distributed over $[0 N]$. Now we find $\bar{y} = [y_1 y_2]$ which maximizes G_W averaged over all possible values of l_1 from $[0 N]$,

$$\operatorname{argmax}_{\bar{y}} G_W = \operatorname{argmax}_{\bar{y}} \sum_{l_1} p(l_1) (w_1 g_1 + w_2 g_2) \quad (5)$$

The individual priority goodputs g_1 and g_2 are therefore computed using the expected goodput formula expressed in Equation 2.

$$g_1 = \begin{cases} (1 - p_b)^{y_1}; & \left(\frac{l_1}{n_1}\right) \leq \frac{R_{CH}}{y_1} \\ \frac{\left(\frac{R_{CH}}{y_1}\right)(y_1 - h)(1 - p_b)^{y_1}}{R}; & \left(\frac{l_1}{n_1}\right) > \frac{R_{CH}}{y_1} \end{cases} \quad (6)$$

$$g_2 = \begin{cases} (1 - p_b)^{y_2}; & \left(\frac{l_2}{n_2}\right) \leq \frac{\left(R_{CH} - \left(\frac{l_1}{n_1}\right)y_1\right)}{y_2} \\ \frac{\left[\frac{\left(R_{CH} - \left(\frac{l_1}{n_1}\right)y_1\right)}{y_2}\right](y_2 - h)(1 - p_b)^{y_2}}{l_2 x}; & \left(\frac{l_2}{n_2}\right) > \frac{\left(R_{CH} - \left(\frac{l_1}{n_1}\right)y_1\right)}{y_2} \end{cases} \quad (7)$$

The low priority goodput g_2 is computed from the bits remaining after all high priority fragments have been transmitted during each second. Condition (a) in Equations 6 and 7 implies that sufficient bits are available to allocate fragment headers

when high and low priority fragments are transmitted at sizes y_1 and y_2 . Condition (b) in Equation 6 implies that all the low and some high priority slices should be discarded in order to meet the demand for fragment overhead while transmitting at size y_1 . Condition (b) in Equation 7 implies that there are sufficient bits to transmit all high priority fragments at size y_1 , but not for transmitting all low priority fragments at size y_2 . Therefore, some low priority slices are discarded. Combining Equations 5, 6 and 7 and substituting $l_2 = N - l_1$, we get the objective function to find \bar{y} which maximizes G_W . We use the branch and bound (BnB) technique to solve the priority-aware expected weighted goodput optimization problem. BnB is a global optimization technique used for non-convex problems; it reduces the number of times the expected weighted goodput values have to be computed compared to the exhaustive search case.

Figure 2 shows G_W during one second for a video encoded at $R = 960$ Kbps at $p_b = 10^{-5}$. The weights $(w_1, w_2) = (0.89, 0.11)$ used were derived for the CIF Foreman sequence. The mean CMSE of the high priority slices contributes 89% of the received video distortion whereas the mean CMSE value of low priority slices contributes only 11%. The optimal fragment sizes are determined in terms of the number of 150 byte slices that can be aggregated into a fragment. In Fig. 2, $(n_1, n_2) = (3, 5)$ and $[(y_1, y_2) = (450, 750) + h]$ are the optimal high and low priority fragment sizes which achieve the maximum goodput of 0.93. This is achieved at the cost of discarding 36 low priority slices per second. As fragment size decreases, f_{sr} increases but the number of discarded slices also increases. When $(n_1, n_2) = (1, 1)$, more than 160 slices are discarded and the corresponding G_W decreases to 0.88.

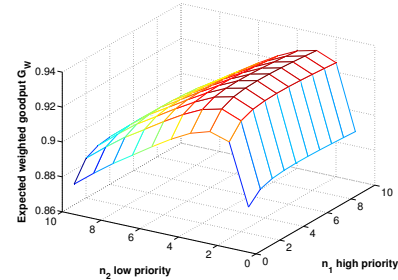


Fig. 2: Expected weighted goodput for a video encoded at $R=960$ Kbps and $x=150$ bytes.

III. PERFORMANCE COMPARISON

We now compare the performance of priority-aware and priority-agnostic fragmentation. As shown in Fig. 3(a), priority-aware fragmentation achieves a goodput gain of 12% over priority-agnostic fragmentation at $R = 960$ Kbps and $p_b = 10^{-4}$, even when it discards 8.25 Kbytes of additional data during every second as shown in Fig. 3(b). However, the performances of priority-agnostic and priority-aware fragmentation converge as channel BER decreases from 10^{-4} to 10^{-6} . Fig. 3(a) also shows a goodput gain of 18% at $R=1080$ Kbps and $p_b = 10^{-4}$. We discard 8.7 Kbytes of

additional data to achieve this gain as shown in Fig. 3(b). Unlike $R=960$ Kbps, priority-aware fragmentation achieves a goodput gain of 8% over priority-agnostic fragmentation at lower BER ($p_b = 10^{-6}$) for $R=1080$ Kbps.

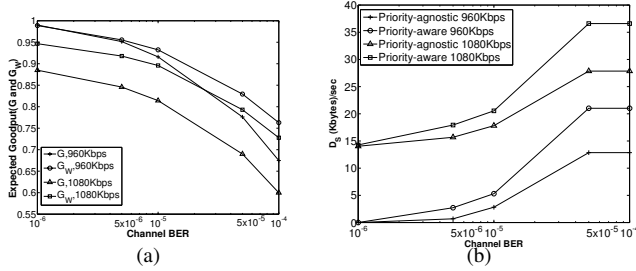


Fig. 3: Comparison between priority-aware and priority-agnostic fragmentation: (a) Expected Goodput, (b) Slice discard.

Though the priority-aware fragmentation provides goodput gain by increasing the transmission reliability of higher priority packets, we have also investigated if this G_W gain corresponds to better video quality. Table I shows the video quality improvement in dB achieved by priority-aware over priority-agnostic fragmentation for the Foreman CIF sequence. The PSNR gain achieved by priority-agnostic fragmentation over the baseline system is shown in brackets. This gain increases as channel BER increases for a given slice size. Increasing slice size decreases the flexibility in choosing fragment sizes as each fragment contains one or more slices in their entirety. For example, the fragment size can be either 600 or 1200 bytes for a 600-byte slice. A 900-byte slice allows us only 1 slice/fragment at 1500 bytes MTU. This restricts the gain that can be achieved over the baseline system. Priority-aware fragmentation provides further gain over priority-agnostic fragmentation and more so at $R = 1080$ Kbps by transmitting the high priority packets with higher fsr at the expense of discarding low priority packets. A maximum gain of 7.8 dB at 960 Kbps and 7.1 dB at 1080 Kbps is achieved by the priority-agnostic fragmentation over the baseline system at a BER of 10^{-4} for 150-byte slices. Similarly a maximum additional gain of 1.2 dB at $R = 960$ Kbps and $p_b = 10^{-4}$ and 1.6 dB at $R = 1080$ Kbps and $p_b = 10^{-6}$ is achieved by 150-byte slices in priority-aware over priority-agnostic fragmentation.

For the Foreman CIF video sequence encoded at 960 Kbps with a slice size of 150 bytes and transmitted with BER = 10^{-5} , the expected goodput values for $n_1 \in [1, 10]$, $n_2 \in [1, 10]$ were shown in Fig. 2. Our results show that higher G_W indeed corresponds to higher video PSNR values. For example, the highest value of $G_W = 0.93$ was obtained for $(n_1, n_2) = (3, 5)$ which corresponds to the highest PSNR of 30.81dB. The lowest value of $G_W = 0.88$ was obtained for $(n_1, n_2) = (1, 1)$ which corresponds to the lowest PSNR of 25.71 dB. We observed similar behavior for other video encoding rates.

TABLE I: Average Video PSNR gain (dB) of priority-aware over priority-agnostic fragmentation (priority-agnostic fragmentation over baseline model) at (a) 960 Kbps and (b) 1080 Kbps.

BER	10^{-6}	5×10^{-6}	10^{-5}	5×10^{-5}	10^{-4}
150 byte	0.1 (1.5)	0.8 (1.4)	1.1 (2.3)	0.9 (6.5)	1.2 (7.8)
300 byte	0 (0.7)	0.9 (1.1)	0.9 (1.9)	0.5 (5.6)	0.1 (7.3)
600 byte	0 (0.6)	0.5 (0.9)	0.8 (1)	0.1 (3.5)	0.1 (3.9)
900 byte	0.1 (0.1)	0.6 (0)	0.2 (0.7)	0.1 (1)	0.1 (0.5)

BER	10^{-6}	5×10^{-6}	10^{-5}	5×10^{-5}	10^{-4}
150 byte	1.6 (5.4)	1.4 (4.2)	1.6 (3.8)	0.7 (5.9)	1.4 (7.1)
300 byte	1.4 (4.6)	1 (3.6)	0.7 (3.4)	0.5 (5.6)	0.4 (6.4)
600 byte	1.2 (4)	0.6 (3.5)	0.7 (3.2)	0.5 (3.3)	0.3 (3.6)
900 byte	0.6 (3.1)	0.5 (3)	0.5 (2.8)	0.4 (1.4)	0.5 (0.5)

IV. CONCLUSION

An efficient priority-aware adaptive packet fragmentation scheme was proposed to improve the quality of pre-encoded H.264 bitstreams transmitted over bit-rate limited and error-prone channels. The fragment sizes for prioritized packets were derived using the BnB algorithm. The cross-layer priority information exchange between the video layer and MAC layer allowed us to selectively discard slices, reducing the impact of lost slices on received quality. The proposed priority-aware fragmentation scheme improves video PSNR over priority-agnostic fragmentation.

REFERENCES

- [1] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard," *Elsevier J. of Visual Communication and Image Representation, Special issue on Emerging H.264/AVC Video Coding Standard*, vol. 17, no. 2, pp. 183–185, April 2006.
- [2] B. A. Forouzan, *Data Communications and Networking*. McGraw-Hill Forouzan Networking Series, 2007.
- [3] M. V. D. Schaar and N. S. Shankar, "Cross-layer wireless multimedia transmission: challenges, principles and new paradigms," *IEEE Wireless Communications Magazine*, vol. 12, no. 4, pp. 50–58, August 2005.
- [4] Y. P. Fallah, K. Darrell, S. Avidel, K. Faizal, and P. Nasiopoulos, "A cross layer optimization mechanism to improve H.264 video transmission over wlangs," in *IEEE CCNC*, 2007, pp. 875–879.
- [5] A. T. Connie, P. Nasiopoulos, V. C. M. Leung, and Y. P. Fallah, "Video packetization techniques for enhancing H.264 video transmission over 3g networks," in *IEEE CCNC*, 2008, pp. 800–804.
- [6] A. Ksentini and M. Naimi, "Toward an improvement of H.264 video transmission over ieee 802.11e through a cross layer architecture," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 107–114, January 2006.
- [7] Y. P. Fallah, H. Mansour, S. Khan, P. Nasiopoulos, and H. Alnuweiri, "A link adaptation scheme for efficient transmission of H.264 scalable video over multirate wlangs," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 875–887, July 2008.
- [8] Y. Sun, I. Sheriff, E. M. Belding-Royer, and K. C. Almeroth, "An experimental study of multimedia traffic performance in mesh networks," in *ACM WiTMeMo*, 2005, pp. 25–30.
- [9] Y. Wang, K. Yu, Y. Liu, and H. Zhang, "Effects of MAC retransmission on TCP performance in IEEE 802.11-based ad-hoc networks," in *59th IEEE VTC*, vol. 4, February 2004, pp. 2205–2209.
- [10] *Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard for Information Technology-Telecommunications and information exchange between systems Std., June 2007.