On the reduction of scalable video base-layer packet loss rate on FIFO/Drop-tail queues

Guilherme D. G. Jaime[1], Edmundo de Souza e Silva[1], Rosa Maria Meri Leão[1], José Roberto Boisson de Marca[2]

[1]COPPE/PESC - Federal University of Rio de Janeiro (UFRJ)
P.O Box 68511 . Zip 21941-972 . RJ . Brazil

[2]CETUC/PUC - Pontif. Catholic University of Rio de Janeiro
R. Marquês de S. Vicente 225 . 22453-900 . RJ . Brazil

{gdjaime,edmundo,rosam}@land.ufrj.br, jrbm@cetuc.puc-rio.br

Abstract. *In this work we develop a simple, yet robust, application-layer adaptation technique based on scalable video codecs. The proposed technique can adapt to the varying capacity available to a connection and, delivers a superior quality of received video. We take advantage of both scalable video properties and the characteristics of the packet dropping random process at the last link's wireless router's queue. By fine-tuning the inter-packet transmission time at the server side, we show that the resulting packet loss pattern yields a superior quality of experience (QoE) without any changes in the wireless router's packet scheduling or drop policies. In other words, by varying the per frame packet inter-spacing time while maintaining a given frame rate (i.e., by varying the transmission* burstiness*) one can control the QoE. We develop an analytical model to study the loss pattern of individual packets in a n-packet burst that is the foundation of our proposal.*

1. Introduction

Video streaming services over wireless networks are rapidly becoming very popular. This is due to two recent developments: (i) the strong deployment of wideband wireless access technologies, such as 802.11 [Vassis et al. 2005, Xiao 2005], EVDO [3rd Generation Partnership Project 2 2001], HSDPA [Holma and Toskala 2006] and WIMAX [Li et al. 2007], and (ii) the availability of efficient video coding standards such as MPEG-4 [Richardson 2003] and its scalable version [Radha et al. 2001, Li 2001] which allows for a graceful quality degradation in the presence of the hostile and time varying mobile radio channel.

There are several performance issues that must be dealt with to offer video streaming services over a wireless channel with reasonable quality. For instance, one has to cope with the channel bit rate fluctuations due to interference, fading, shadowing and others. Modern wireless technologies generally apply time varying adaptive modulation/rate selection in order to keep the packet error rate (PER) under a desired threshold (i.e. 1%).

Another issue is to provide fairness among users, and packet scheduling algorithms have been proposed to deal with this problem (e.g. [Kang and Zakhor 2002, Andrews and Zhang 2004, Ryu et al. 2005]). Although packet scheduling techniques can provide good performance to different applications such as web browsing, e-mail, and file transfer services, it was observed in [Jaime et al. 2008] that fading can still cause individual user's achievable transmission rate to vary by orders of magnitude. This is because wireless technologies usually adaptively switches from several modulation/transmission rate in order to keep the physical layer packet error rate (PER) under control (i.e. 1%). However, this may drastically affect fairness.

During the transmission time of a video, variations of the wireless channel effective transmission rate are not negligible and the channel may incur periods of very low transmission rates. In this scenario, traditional (non-scalable) video streaming suffers from severe levels of losses and possibly playout buffer underflow, resulting in unacceptable degradation on the video quality experienced by the user. Scalable video then emerges as a alternative to mitigate the effects of high channel capacity variations.

Roughly, scalable video allows for the selection of a wide range of bit rate options from a single data stream. This is very appealing specially in wireless environments since, for instance, the application may adapt the transmission rate to the current bandwidth available in the source-destination path.

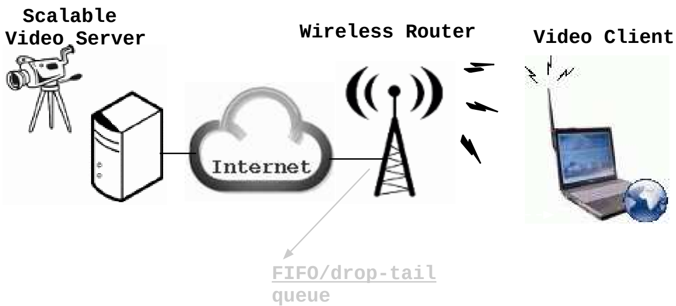


Figure 1. General scenario considered in this paper

Figure 1 shows a scenario where a client user is streaming video from a server. The wireless channel is the last hop and we assume that this is the bottleneck of the server-client path. Three main issues have been addressed in recent studies in the literature and below we cite some of these studies.

In the first set of works, the sending rate from the server to a client adapts to variations of the client's wireless channel bit rate, aiming at improving the user perceived video quality (*quality of experience* - QoE). Basically, the proposals in this subset combine scalable streams, such as MPEG-4 FGS [Radha et al. 2001, Scaglione and van der Schaar 2005] and H.264/SVC [MPEG 2005] with application-layer adaptation techniques [Shan 2005, Haratcherev et al. 2006]. One drawback of these proposals [Chen and Chen 2004], is that they only allow for coarse bandwidth adaptation capability. Unpredictable channel rate fluctuations caused, for instance, by fast-fading or medium sharing may cause increased data loss which may considerably affect the QoE.

Two other strategies, namely packet scheduling and packet dropping have been proposed in recent works. Their objective is to try to minimize the QoE degradation caused by losses at the last link's wireless router's queue. This is achieved by adopting either elaborate packet scheduling schemes [Zhang et al. 2009], [Fiandrotti et al. 2008], or last link packet dropping disciplines such as, for example, UPP - Unequal Packet-Loss

Protection, or one of its variants [Zhang et al. 2009, van der Schaar and Radha 2001]. Among the disadvantages of the proposed schemes are its implementation complexity which requires processing of the upper layer packets header and the intrinsic requirement of changing standards/hardware.

Different from the works in the literature, our main goal is to avoid changes in the wireless routers. For that we develop a simple, yet robust, application-layer adaptation technique. We take advantage of both scalable video properties and the characteristics of the packet dropping random process at the wireless router. By fine-tuning the inter-packet transmission time at the server side, we show that the resulting packet loss pattern yields a superior QoE without any changes in the wireless router's packet scheduling or drop policies. In other words, by varying the per frame packet inter-spacing time while maintaining a given frame rate (i.e., by varying the transmission *burstiness*) one can control the QoE.

It is well known that the packet loss process is sensitive to the transmission burstiness and, by increasing the burstiness, one may adversely affect the expected packet loss rate. What is intriguing is that, by using scalable video, an increase in burstiness may favor the quality of the transmission. We develop analytical models to help understand this phenomenon.

As mentioned above, the relationship between packet losses and packet spacing/pacing (burstiness) have already been studied in the past. Some of the existing several works focus on multimedia content distribution [Feng et al. 2002], while others do not [Sivaraman et al. 2006, Cai et al. 2009]. Usually, the measure of interest is the packet loss rate [Feng et al. 2002, Sivaraman et al. 2006] or queuing length related metrics [Cai et al. 2009]. The effect of the loss burst size on the mean-squared error distortion of non-scalable video sequences has also been addressed in [Liang et al. 2003] using simulation. However, the authors did not consider scalable video coding and, in addition, did not assess the *intra-burst* packet loss process.

To the best of our knowledge, our work is the first one to assess the loss process of individual packets in a burst and show how to improve the quality of a transmission that uses a scalable video codec. In summary, our contributions are: (a) an analytical model that is employed to study the loss process of individual packets in a n-packet burst; (b) propose a simple application-level mechanism based on scalable video codecs that can adapt to the varying capacity available to a connection and, delivers a superior quality of received video without any changes on standards or the scheduling of wireless routers. Our results are based on a general scenario which covers a wide variety of possible applications, such as wireless or even wireline Internet access technologies.

The remaining of this paper is organized as follows. In section 2, after briefly introducing the scalable stream packetization model considered in this paper, we develop an analytical model that is the foundation of our proposal. The chosen performance metrics are also presented in section 2. Our simple proposal is described in section 3 as well as the results from the model that supports our ideas. Section 4 concludes the paper.

2. The Analytical Model

We start the section by presenting a brief overview of scalable video. Then we develop an analytical model that provides the foundation for the proposed approach.

2.1. Brief overview of fine granularity scalability

We consider FGS - Fine Granularity Scalability [Li 2001, Radha et al. 2001] coded video. As shown in Figure 2, the FGS structure consists of only two layers: the base layer (BL) and the enhancement layer (EL).

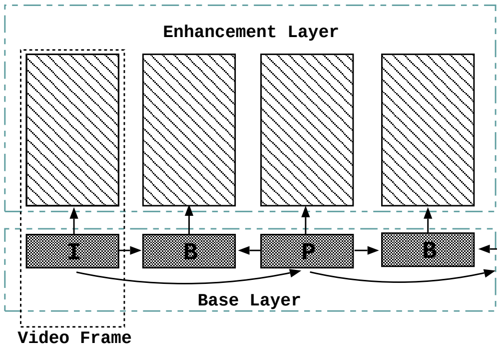


Figure 2. Example of FGS scalability structure

The BL uses traditional nonscalable coding. The EL is encoded with the difference between the original picture and the BL reconstructed picture [Radha et al. 2001]. Consequently, the player must possess a BL in order to decode its associated EL. The EL is organized from the most significant bits (i.e. bitplane) to the least significant one. This means that the EL may be truncated at any point, and the EL video quality is proportional to the number of bits decoded for each picture.

The adaptive bit rate is achieved by playing the BL and any truncated part of the EL. Clearly, if only the BL is played, a minimum bitrate R_{min} is achieved. Likewise, the maximum bitrate R_{max} is obtained when the BL plus the complete EL is played.

2.2. Analytical Model

In what follow we develop an analytical model whose objective is to calculate the *intra-burst* packet loss distribution. Our model considers the scenario described in Figure 1 and provides the main results that supports our proposal.

We acknowledge that the analytical model is simple and does not include neither details of the wireless networks nor traffic characteristics. However, we develop a realistic and detailed simulation model. The simulation model provides the same kind of results obtained from the analytical model and we chose not to include it in the paper for conciseness. We opt for using a simple analytical model to try to isolate the causes underneath the counter-intuitive observations, of our work. Roughly, we want to show that controlling the video traffic burstiness over a queue with finite buffer size in the presence of cross traffic may improve the video quality at the receiver.

Figure 3 presents the main building blocks of our model. The module labeled *Bursty_Source* represents the traffic sent by the video server. The module *Last_Queue* is an abstraction of the FIFO/Drop-tail queue at the last hop that could be a wireless link, for instance. We assume that the link at the last hope is shared by other traffic. This additional cross-traffic is modeled as a Poisson source.

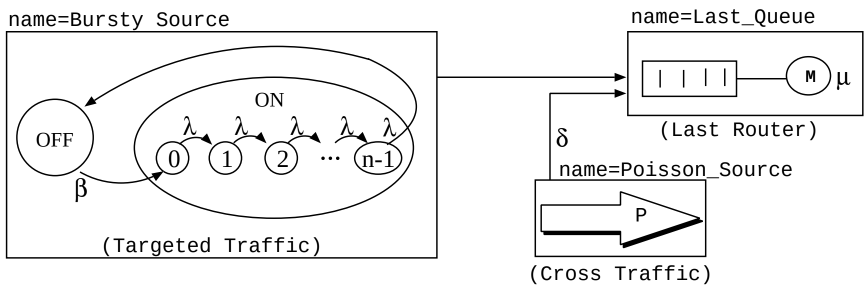


Figure 3. Analytical model

The *Bursty_Source* module is defined as follows. If the *Bursty_Source* is in the OFF state, no packets are generated. We assume that the OFF period is exponentially distributed with mean $\frac{1}{\beta}$. Once the *Bursty_Source* module state changes to ON, a fixed number of packets is generated. Each packet carries a piece of the coded frame produced at the server. The inter-packet transmission time is assumed exponentially distributed with mean $\frac{1}{\lambda}$. Note that, although the packets at the video source are generated at constant intervals, the inter-arrival interval at the last hop is far from a deterministic random variable due to the jitter introduced by the Internet when the packets travel from source to the destination. Therefore, it is not unreasonable to assume that the video packets inter-arrival time are exponential random variables.

Let f be the video frame rate (FPS). From the model, the average duration of the time spent in both ON and OFF states is the interval between two consecutive frames and equal to $\frac{1}{f}$. Clearly, $\frac{1}{\beta} + \frac{n}{\lambda} = \frac{1}{f}$, and β can be written as a function of both λ and f:

$$\beta = \frac{f\lambda}{(\lambda - (nf))} \qquad \text{for } \lambda > nf. \tag{1}$$

The *burstiness* (b) is defined as the ratio between the targeted traffic packet generation rate nf and the intra-burst video packet rate (λ): $b = \frac{\lambda}{nf}$ ($b > 1$). Therefore,

$$\beta = \frac{bf}{(b-1)}. \tag{2}$$

Note that it is easy to control the burstiness of the video stream. If we fix the frame generation rate f, the burstiness can vary by changing λ. From Equation 1, β can be obtained as a function of the burstiness.

The *Poisson_Source* module generates Poisson traffic which represents the aggregated cross-traffic [Cao et al. 2003] sharing last link resources with the video traffic. We define δ as the average cross traffic packet generation rate.

A state S of the model is the concatenation of three state variable (s_1, s_2, s_3), where s_1 indicates the state of the source (ON (1) or OFF (0)), s_2 counts the number of packets generated in the current burst ($(0, 1, ..., n-1)$) and s_3 is the current number of packets in the buffer of the *Last_Queue*. The buffer size is equal to q.

Note that the set of states with $s_3 = q$ contains all the states with a full buffer. Thus, a transition

$$(1, s_2, q) \rightarrow (1, (s_2 + 1 \; mod \; n), q) \tag{3}$$

indicates the loss of the $(s_2+1)^{th}$ video packet generated by the source.

Let $\pi_{(s_1,s_2,s_3)}$ denote the fraction of time the system remains in state (s_1,s_2,s_3). Define γ as the average number of video packet losses per time unit. (In what follows the time unit used is seconds.) Since our model is Markovian γ is equal to:

$$\gamma \;=\; \sum_{i=0}^{n-1} \lambda \pi_{1,i,q} \qquad \text{for } 0 \;\leq\; \gamma \;\leq nf$$

In addition, let γ^i be rate of losses of the $(i+1)^{th}$ packet generated. Then

$$\gamma^i \;=\; \lambda \pi_{1,i,q}, \qquad \text{for } i = 0, ..., n-1.$$

We define F^i as the ratio between the $(i+1)^{th}$ intra-burst video packet loss rate and the $(i+1)^{th}$ packet generation rate. We also define F as the ratio between the overall video packet loss rate and the packet generation rate. Thus, F^i and F can be written as:

$$F^i \;=\; \frac{\gamma^i}{f} \quad \text{and} \quad F \;=\; \frac{\gamma}{nf}. \tag{4}$$

Let ρ be the system load, which is equal to the ratio between the total traffic (video + cross traffic) $\delta + nf$ and the service rate μ:

$$\rho \;=\; \frac{\delta + nf}{\mu}. \tag{5}$$

It is important to note that ρ does not vary with λ because we set β using Equation 1 in order to maintain fixed the packet generation rate (nf).

Figure 4 illustrates three examples of λ and β values used in Section 3. Note that β increases with λ in order to keep the packet generation rate and system load constant.

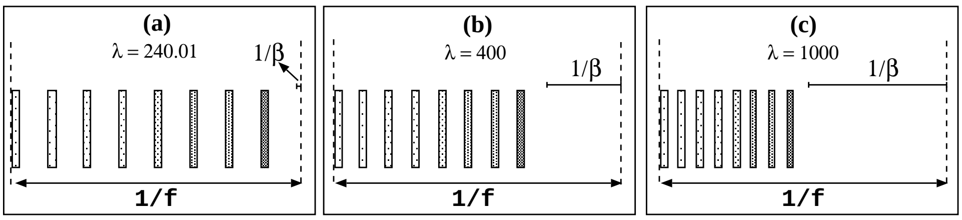


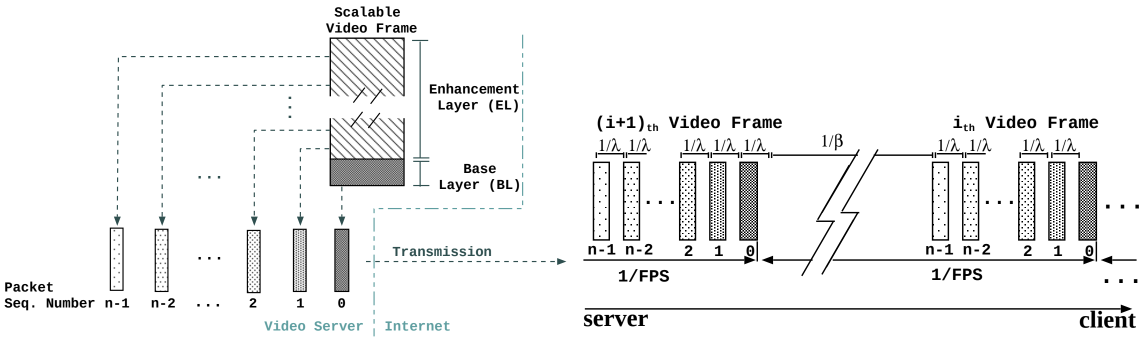
Figure 4. Examples to illustrate the relation between λ and β.

It is not difficult to see that the model's infinitesimal generator matrix (Q) has a Quasi-Birth-Death (QBD) structure [Latouche and Ramaswami 1999]. Thus efficient solutions exist to solve such class of models and the metrics of interest (equation 4) can then be easily calculated. For describing and solving the model we used the Tangram-II [de Souza e Silva et al. 2009] modeling environment.

3. Proposal, Numerical Results and Discussion

As mentioned in previous sections our main objective is to propose a simple application-layer technique that could improve the loss of the most important packets and yet would not require any changes in the wireless router. We recall that the scenario considered is shown in Figure 1, and we assume that the bottleneck link is wireless and at the last hop.

We refer to Figure 5 to explain our proposal. The idea is simple, and requires only the packetization and subsequent transmission of the bytes of a scalable video frame in a certain manner. Each video frame is divided into a sequence of packets such that the BL bits fits into the first packet to be transmitted (packet 0). The following packets, from 1 to $n-1$, contain the EL bits *ordered* according to their relevance for the decoding process (Figure 5(a)). Since the EL bytes of a scalable video frame are generated in order according to their relevance, the packetization is trivial: the first x EL bytes generated by the encoder are included in the payload of the second packet to be transmitted (packet number 1), bytes $x+1$ to $2x$ are included in packet number 2, etc. Packets are transmitted in increasing order of payload relevance, i.e., the first packet of the burst is packet 0, the second is packet 1, etc. (see Figure 5 (b)).



(a) (b)

Figure 5. (a) Video Frame Packetization; (b) Packet transmission

Needless to say the packetization process is done in the natural way one would perform such a task. The key point is the manner by which packets containing a scalable video frame are transmitted. All packets containing a frame are sent within a burst and the variable to be controlled is the interval between two consecutive packets. Figures 5(b) and 4 illustrate how the burst is generated. The duration of a burst of packets corresponding to a single scalable video frame is equal to $(1/FPS - 1/\beta)$, where we recall that FPS is the video frame rate (in frames per second). The interval between the end of a burst and the beginning of the next is equal to $1/\beta$. Evidently, no packets are emitted during this period.

We have not explained yet how the very simple procedure outlined above may achieve our final objective to provide a better video quality at the client. In what follows we use the model of section 2.2 to show how this can be accomplished. We emphasize that we have only a single control variable: the transmission burliness or, equivalently, the interval between the generation of packets in a burst.

We base our analysis on a subset of parameter values that are in agreement with the values encountered in a real streaming video application and common wireless channel capacities. For all the analysis performed in this section the video frame rate and bit rate are chosen equal to $f = 30$ FPS and 2.80 Mbps, respectively. The above value for bit rate was selected in order to provide a good video quality for the client. The chosen frame and bit rates imply that the targeted (video) traffic generation rate is equal to 240 packets per second. In addition, considering packets with length 1500 bytes, the number of packets generated per frame is 8, which is equal to the burst size.

The capacity of the wireless router is set to 5.80 Mbps which approximates the maximum data rate of either EVDO or HSDPA 3G Internet wireless access technologies. Therefore, the service rate of the queue in the model of Figure 3 is $\mu = 484$ packets per second. We choose two values for the queue buffer size: $q = 25$ and $q = 150$. The first ($q = 25$) is a common value used in the literature. The second value $q = 150$ represents a very high buffer size, larger than any reasonable value. This was done in order to show that our results are valid over a wide range of buffer sizes.

We vary the burstiness b from $(1 + \epsilon)$ [1] to 83.33. (Equivalently, since $b = \lambda/nf$, λ vary from 240.01 to 20000.)

The cross-traffic arrival rate δ varies from 99 to 295 packets per second. From Equation (5) ρ varies from medium to very high load values (0.7 to 1.1). This was done to generate non-negligible packet losses.

For convenience, in Table 1 we summarize all parameter values used in the studies shown in the sequel.

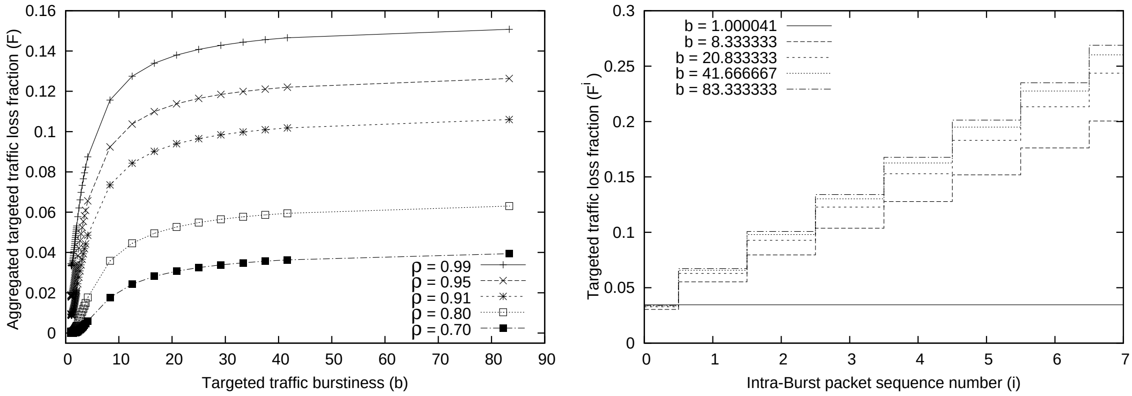
Parameter	Value/Range	Explanation	Corresponds to...
f	30	video frames per second (FPS)	30 FPS video
n	8	fixed burst size (ON state)	8 packets per burst
nf	$8 \times 30 = 240$	average video packet rate	2.88Mbps
μ	484	average service rate	5.88Mbps
q	25,150	buffer size in packets	max. queue delay of 0.052s and 0.31s
δ	99 to 295	cross traffic of 1.18 to 3.54Mbps	0.70 to 1.10 load (ρ)
λ	240.01 to 20000	b from 1 (smooth) to 83.33 (bursty)	2.88Mbps to 240Mbps

Table 1. Considered analytical model parameters

Figure 6(a) shows the fraction of video traffic (also called *targeted* traffic) that are lost as a function of the burstiness parameter b. As expected, for a fixed value of ρ, the loss fraction sharply increases in the lower 20% burstiness values. The increase in the loss fraction as a function of the burliness is not a new result. Later we comment on this.

Figure 6(b) shows the fraction of lost packets for each individual packet (F^i) transmitted in a burst generated for each frame. The number in the x-axis is the order of the transmitted packet, 0 being the first. (Recall that packet 0 carries the base-layer of the coded video, and the remaining the enhancement-layer.) This figure shows that, except when the value of the burstiness is close to 1, the loss fraction increases with the order of

[1]Since b must be greater than 1 so that β is positive (see equation (2)), ϵ was chosen a small quantity, 0.45×10^{-5}

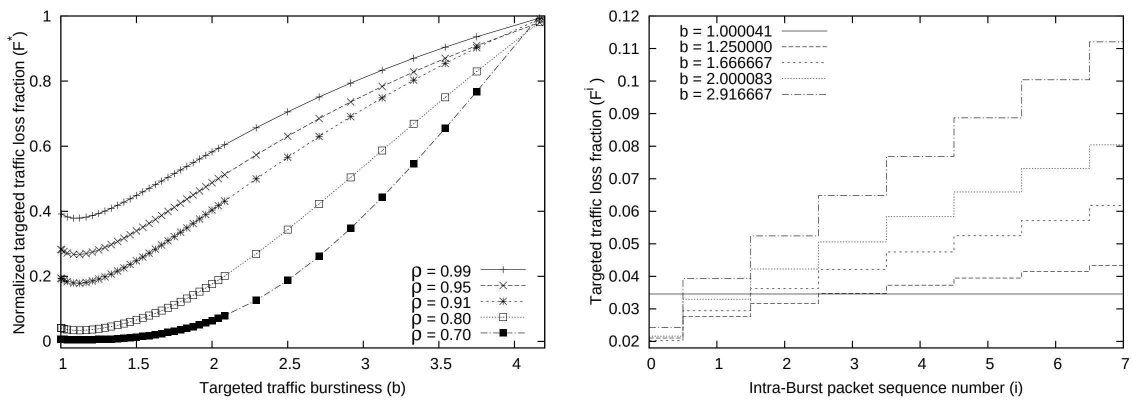


(a) (b)

Figure 6. (a) Targeted Traffic loss rate, q=25; (b) Intra-burst Loss rate distribution, for $\rho = 0.99$ and q=25

the packet transmitted in a burst. In order words, when $b \approx 1$ (which means that the video packets are sent smoothly during the interval between the generation of two consecutive frames), the loss process is approximately the same for all packets in a burst. However, when b increases, $F^0 < F^1 < \ldots, F^{n-1}$, and the first packets transmitted in a burst experience less losses in comparison to the remaining packets.

A few works in the literature [Floyd and Jacobson 1992, Floyd 1993, Towsley et al. 2000] show that the packet loss rate increases with the $burstiness$, similar to what is shown in Figure 6(a). This result suggests that a smoothed traffic ($b = 1$) is a good option for video transmission over packet switched networks. However, based on Figure 6(b), the next set of results show that it is possible to adjust the value of b in order to control the packet loss rate of individual packets in the burst. This is the key to show that one can increase the perceived quality of the video stream by varying b.



(a) (b)

Figure 7. (a) Targeted Traffic average aggregated loss rate (F_l) (normalized), for lower b values and q=25; (b) Intra-burst Loss rate distribution ($\rho = 0.99$), for lower b values and q=25

We refer to figures 7(a) and 7(b) to illustrate that, if we adjust the burstiness parameter b, it is possible to lower the values of F^i for the first packets in the burst in comparison with F^i when $b \approx 1$ (i.e., smooth traffic). Surprisingly, this can be achieved without a significant increase in the value of F, as it will be discussed later. Our proposal for video packetization explores this fact. Recall that, after a video frame is generated, packets as transmitted in decreasing order of significance for the quality of the video image to be decoded. The first packet of each burst contains BL bits while the subsequent packets contains the EL bits. As a consequence, the most important packets in the burst are the first and those should be preserved with higher priority in comparison to the packets in the burst tail.

Figure 7(a) is a zoom from the left side of Figure 6(a). We focus on the burstiness values $1 < b \leq 4.2$, that is, the video (targeted) traffic gradually changes from smooth to slightly bursty. In order to facilitate the comparison among the curves for different loads, we normalize each $F(\rho)$ function for a given ρ by dividing $F(\rho)$ by the maximum value obtained (F^{max}) for that ρ. The normalized values are referred as F^*. This explains why all $F^*(\rho)$ functions reach 1 for $b = 4.2$. As shown in Figure 7(a), there is no significant increase in F^* until $b > 1.2$ (for $\rho > 0.9$) or $b > 1.7$ (for 0.7 < $\rho < 0.9$).

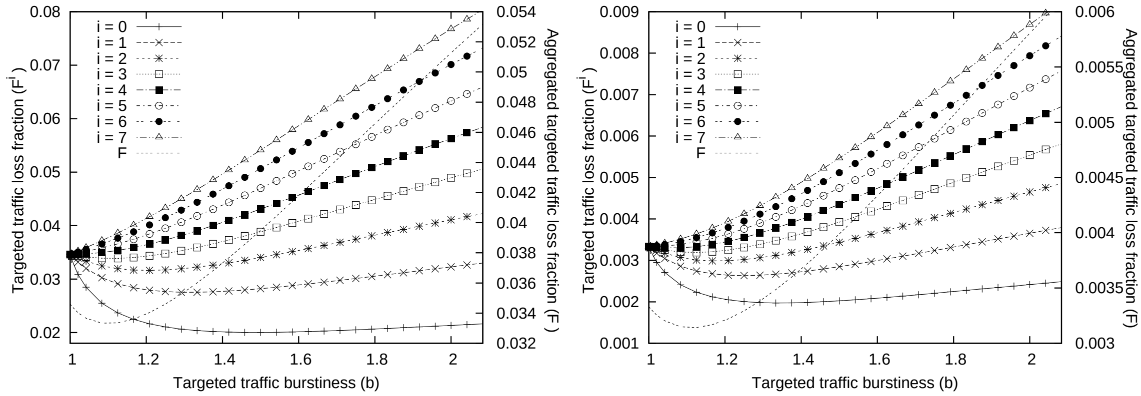
Figure 7(b) shows the fraction of loss F^i for the i-th packets in a burst, for $1 < b < 3$. The key observation is that, in this range, F^0 is smaller than the equivalent metrics when $b \approx 1$. This means that, by properly adjusting the burstiness of the targeted traffic, it is possible to lower the losses of the first packets in a frame burst compared to a smooth transmission. By doing that, one favors the most important packets for decoding. This is achieved at the expense of increasing the loss fraction of the packets that carry the least significant information for decoding the video. In fact, there is a tradeoff for selecting b in the range $1.2 < b < 3$: by losing more packets at the end of the burst, we incur in fewer losses at the beginning of the burst. Then one can improve the video quality since, intuitively, when using FGS encoded video, it is better to loose the least important EL bits.

Figure 8 shows F^i for each individual packet transmitted in a burst, as a function of b. Note that for the 0-th packets (i.e., the first packet in a burst that carries the BL) the lowest value for the loss probability are achieved for $1.2 < b < 1.5$. On the other hand, F^i, for $i \geq 4$, increases (≈ 25% to 30%) in that b range. If b is further increased beyond 2, F^i grows toward the higher values shown in Figure 6(b).

It should be noted that there is a range of burstiness values that can provide a significant improvement on the loss fraction of the base layer (BL) packets without hurting the overall packet loss fraction. For instance, the choice of $b \approx 1.3$ results in a reduction of 40% in the loss fraction of BL packets (F^0), when compared to the $b \approx 1$ case, with only a small impact on the overall packet loss ($F^0 = F = 3.45\%$ when $b \approx 1$; and $F^0 = 2.07\%$, $F = 3.6\%$ when $b = 1.3$) (see Figure 7(a)). This is indeed excellent news, since the BL packets are by far the most important packets in the frame, and should be preserved for decoding.

Figure 8(b) shows that similar conclusions as above can be drawn even for large queue sizes (150 is this example). It is interesting to see that, although the loss fraction decreases with increasing queue sizes, the range of burstiness values that favor the BL packets remains virtually unaltered.

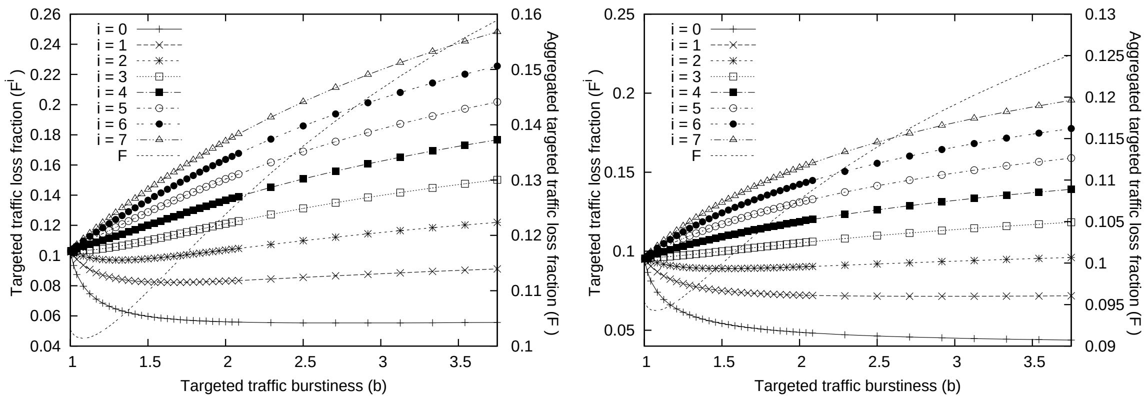
The next set of results shows that the loss behavior perceived in our work is present



(a) buffer = 25 (b) buffer = 150

Figure 8. Loss rate for each intra-burst-Burst packet, as a function of burstiness (b) ($\rho = 0.99$)

for different load values in addition to distinct queue sizes. In Figure 9, we consider $\rho = 1.10$ and in Figure 10, $\rho = 0.89$. In both cases, two buffer sizes are considered: 25 and 150.



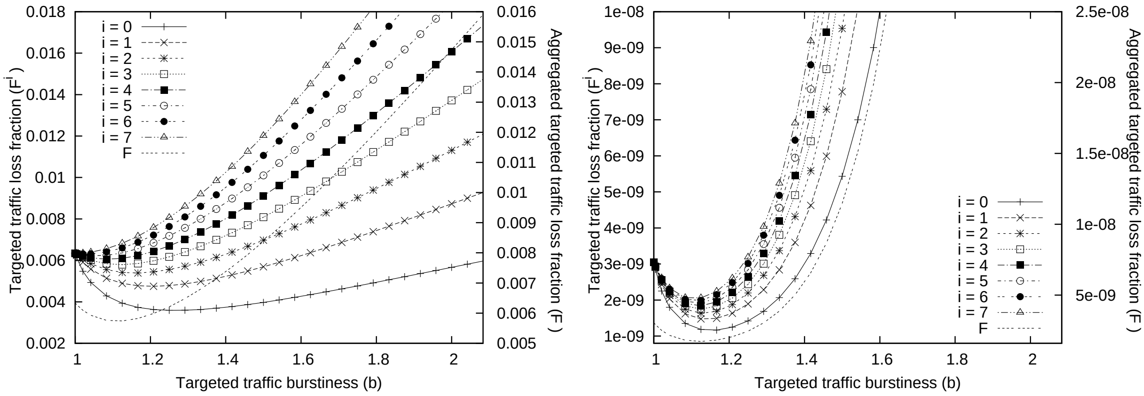
(a) buffer = 25 (b) buffer = 150

Figure 9. Loss fraction for each intra-burst packet (F_l^i), as a function of burstiness (b) ($\rho = 1.105$)

When $\rho = 1.10$ (overloaded system scenario) the loss behavior is very similar to that shown in Figure 8 (where $\rho = 0.99$). Only the loss rate absolute values change.

We can reach identical conclusions as above for $\rho = 0.89$ and $q = 150$ (Figure 10(b)). It is intriguing that, for a wide range of buffer size values and loads, the range of burstiness values that produce smaller BL packet losses with only a small increase in the overall packet loss is virtually unaltered.

As we have mentioned in Section 2.2, we developed a realistic and detailed simulation model. The results obtained from the simulation model show the same behavior for the loss fraction distribution.



(a) buffer = 25 (b) buffer = 150

Figure 10. Loss rate for each intra-burst packet, as a function of burstiness (λ) ($\rho = 0.89$)

4. Conclusions and future work

In this paper we show that, if the drop-tail loss characteristics and packet burstiness of a scalable video stream are handled cleverly, FIFO/Drop-Tail queues can be used to achieve improved video quality. By controlling only the burstiness of the video traffic, the fraction of base layer packet losses can be significantly reduced at the expense of only a small increase in the overall packet loss rate. The improved performance is obtained with no changes in the current standards/protocols, and no additional processing at the wireless routers.

While previous work show that, by decreasing the burstiness of a transmission, the overall loss rate is reduced, we look at the individual loss process of packets in a burst. The motivation for focusing on individual packets in a frame of a scalable video stream is that distinct packets carry data with different levels of importance for the video quality. A simple analytical model was used to show that one can lower the loss fraction of the base layer packets, which are the most important packets in the stream, at the expense of an increase in the loss rate of the least important packets (the packets in the tail of the frame burst). Although our work considers scalable video streaming over wireless networks, the results we obtained are applicable to any multimedia transmission scenarios in which some packets are more important than others with respect to a chosen measure of quality.

We have already built a detailed simulation that implements the details of the EVDO protocol and the preliminary results obtained are in agreement with those from the analytical model. In addition, we used a real scalable video stream trace to feed the simulation. Due to the lack of space, we omitted these results. However, we have already observed that the resulting quality of video when the burstiness is fine tuned as compared with that when the video traffic is smoothed is significant. As a future work we will use video quality metrics (i.e. PSNR) in order to quantify the quality gains that are possible to achieve when our proposal is used.

References

3rd Generation Partnership Project 2 (2001). Cdma2000 High Rate Packet Data Air Interface Specification, C.S2002.4, V3.0.

Andrews, M. and Zhang, L. (2004). Scheduling Over non-Stationary Wireless Channels with Finite Rate Sets. In *Proc. of IEEE Infocom*.

Cai, Y., Hanay, S., and Wolf, T. (2009). Practical Packet Pacing in Small-Buffer Networks. *IEEE ICC'09*.

Cao, J., Cleveland, W., Lin, D., and Sun, D. (2003). Internet traffic tends toward Poisson and independent as the load increases. *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG-*, pages 83–110.

Chen, T. and Chen, T. (2004). Fine-Grained Rate Shaping for Video Streaming over Wireless Networks. *EURASIP Journal on Applied Signal Processing*, 2:176–191.

de Souza e Silva, E., Fiqueiredo, D. R., and Leão, R. M. (2009). The TANGRAMII Integrated Modeling Environment for Computer Systems and Networks. *Performance Evaluation Review*, pages 64–69.

Feng, A., Kapadia, A., Feng, W., and Belford, G. (2002). Packet spacing: An enabling mechanism for delivering multimedia content in computational grids. *The Journal of Supercomputing*, 23(1):51–66.

Fiandrotti, A., Gallucci, D., Masala, E., and Magli, E. (2008). Traffic Prioritization of H. 264/SVC Video over 802.11 e Ad Hoc Wireless Networks. In *ICCCN'08*, pages 1–5.

Floyd, S. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking (TON)*, 1(4):397–413.

Floyd, S. and Jacobson, V. (1992). On traffic phase effects in packet-switched gateways. *Internetworking: Research and Experience*, 3(3):115–156.

Haratcherev, I., Taal, J., Langendoen, K., Lagendijk, R., and Sips, H. (2006). Optimized video streaming over 802.11 by cross-layer signaling. In *IEEE Communications Magazine*, pages 115–121.

Holma, H. and Toskala, A. (2006). *HSDPA/HSUPA for UMTS*. Wiley.

Jaime, G., Leão, R., e Silva, E., and de Marca, J. (2008). User Level Fairness of a Multirate 3G Technology (EVDO) With Application to Digital TV Return Channel Technology. *IEEE TRANSACTIONS ON BROADCASTING*, 54(3).

Kang, S. and Zakhor, A. (2002). Packet scheduling algorithm for wireless video streaming. *International Packet Video Workshop*.

Latouche, G. and Ramaswami, V. (1999). *Introduction to matrix analytic methods in stochastic modeling*. Society for Industrial Mathematics.

Li, B., Qin, Y., Low, C., and Gwee, C. (2007). A survey on mobile WiMAX. *IEEE Communications magazine*, 45(12):70–75.

Li, W. (2001). Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Transactions on circuits and systems for video technology*, 11(3):301–317.

Liang, Y., Apostolopoulos, J., and Girod, B. (2003). Analysis of packet loss for compressed video: Does burst-length matter? In *IEEE ICASSP'03*, volume 5.

MPEG, I. (2005). Svc requirements specified by mpeg. jvt-n026. `http://ftp3.itu.ch/avarch/jvt-site/2005_01_HongKong/jvt-n026.doc`.

Radha, H., van der Schaar, M., and Chen, Y. (2001). The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP. *IEEE Transactions on multimedia*, 3(1):53–68.

Richardson, I. (2003). *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons Inc.

Ryu, S., Ryu, B.-H., Seo, H., Shin, M., and Park, S. (2005). Wireless packet scheduling algorithm for ofdma system based on time-utility and channel state. In *ISWSN'05*, volume 7, pages 777–787.

Scaglione, A. and van der Schaar, M. (2005). Cross-layer resource allocation for delay constrained wireless video transmission. In *IEEE ICASSP'05*, volume 5.

Shan, Y. (2005). Cross-layer techniques for adaptive video streaming over wireless networks. *EURASIP journal on applied signal processing*, 2005(2):220–228.

Sivaraman, V., Elgindy, H., Moreland, D., and Ostry, D. (2006). Packet pacing in short buffer optical packet switched networks. In *Proceedings of IEEE Infocom*.

Towsley, D., Firoiu, V., Padhye, J., and Kurose, J. (2000). Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Transactions on Networking (ToN)*, 8(2):133–145.

van der Schaar, M. and Radha, H. (2001). Unequal packet loss resilience for fine-granular-scalability video. *IEEE Transactions on Multimedia*, 3(4):381–394.

Vassis, D., Kormentzas, G., Rouskas, A., and Maglogiannis, I. (2005). The IEEE 802.11 g standard for high data rate WLANs. *IEEE network*, 19(3):21–26.

Xiao, Y. (2005). IEEE 802.11 n: enhancements for higher throughput in wireless LANs. *IEEE Wireless Communications*, 12(6):82–91.

Zhang, H., Zheng, Y., and ohammad A. (Amir) Khojastepour (2009). Scalable Video Streaming over Fading Wireless Channels. *IEEE WCNC'09*.