Analysis of SR ARQ Delays Using Data-bundling over Markov Channels

Iffat Ahmed⋆, Leonardo Badia†, Andreas Petlund⋆, Carsten Griwodz⋆, Pål Halvorsen⋆
⋆ Simula Research Laboratory, University of Oslo, Norway
†Department of Information Engineering, University of Padova, Italy
E-mail: {iffat,apetlund,griff,paalh}@simula.no, badia@dei.unipd.it

***Abstract*—Data-bundling is a useful technique that decreases the delivery delay of packet streams when they are transmitted over noisy channels and are subject to retransmission-based error control. In this paper, we investigate the packet delay statistics for a fully reliable selective repeat automatic repeat request (SR ARQ) where a data-bundling mechanism is employed. In more detail, we discuss a model for data-bundling to analyze the SR ARQ mechanism over wireless channels based on Markov chains. We evaluate various channel error distributions and analyze the buffer occupancy to check if the data-bundling mechanism provides efficient results. We further analyze the queueing, delivery and overall delay statistics at link layer. We found that using data-bundling can improve the delay performance of the SR ARQ mechanism, especially when bursty channels with heavily correlated errors are considered. Thus, this technique can bring useful improvements for real-time services, multimedia, and other delay-sensitive applications over wireless networks.**

***Index Terms*—Data-bundling, multimedia networks, error control, automatic repeat request, real-time services.**

I. Introduction

Next Generation Networks will be facing a tremendous increase in the exchange of multimedia content [1], which are delay-sensitive in nature. This issue gets more important when error-prone channels as provided by wireless networks are considered, since multimedia applications are sensitive to channel losses and delays. This implies the need for effective error control techniques that can significantly impact on delays, bandwidth usage and data reliability. Careful design of effective error control mechanisms can significantly impact on reducing latencies for such applications.

Automatic repeat request (ARQ) is a commonly known error control mechanism besides forward error correction (FEC). In the selective repeat (SR) ARQ mechanism, the sender retransmits only the packets that are negatively acknowledged. The transmission is then resumed from the last packet sent so far [2]. Since the packets must be released *in-order*, the packets experience delays caused by head-of-line blocking.

Data-bundling decreases the likelihood of experiencing such delays by aggregating multiple packets between the same source and destination. It does this by exploiting the redundancy which is naturally present in the packets, for example in the header, and/or by making use of the fact that multimedia applications generate packets whose size is only a fraction of the maximum wireless frame size. In these situations, data-bundling can combine multiple packets into a single one, thereby reducing the delay, which is discussed in more detail in the subsequent sections.

This technique has been explored as a useful tool for reducing latency in wide area networks (WANs) for many years [3]–[5]. In this paper, we apply the principle of data-bundling to a single-hop scenario of a wireless network, at the link layer. If retransmission-based error techniques are employed, retransmissions of erroneous packets is generally prioritized over new packets, and forwarding of new packets can experience delays. However, if bundling is possible, retransmitted packets can be fused with newer ones so as not to waste a transmission opportunity for the retransmission only. Clearly, this technique has to be carefully evaluated against any added resource consumption incurred by the mechanism: for example, a flow using bundling in this way consumes more bandwidth (unless it is slotted) and end systems are required to distinguish between normal and bundled packets. However, the latter is generally within the capabilities of modern wireless terminals.

To model and investigate this scenario, we use Markov chains which provide a good representation for wireless channels [2], [6], [7], and allow for general and conceptually simple descriptions. From the modeling standpoint, only one transmission event is considered. We remark that most of the delay investigations over wireless networks assume an independent and identically distributed (*iid*) error process, since it is a relatively easy choice that only depends on one parameter (the channel error probability). However, such a characterization may be imprecise as errors are often correlated, and correlation heavily impacts on the SR ARQ delays, as shown by [8]. Such an aspect can be well included in our Markov-based analysis.

In the literature, several contributions focused on some aspects of the delay statistics for SR ARQ or similar techniques. However, to the best of our knowledge, features such as accounting for possible data-bundling has never been considered before, which makes our analysis entirely original. For example, the authors of [7] investigate the impact of variable arrival rate on the SR ARQ delivery process. This paper also discusses the effect of channel error bursts, as opposed to an *iid* error process, however it does not focus on the data-bundling mechanism.

Rosberg et al. [9] analyzed the resequencing delay and buffer occupancy for SR ARQ. However, this paper is based on average buffer occupancy, which can be translated into average delay estimations through Little's law, but no exact analysis

of the full statistics of either delay term has been performed.

Similarly, Rossi et al. [6] studied the delay statistics of SR ARQ over a Markov channel and tried to approximate the respective delivery delays, but they did not consider the effect of data-bundling on the SR ARQ delivery process. Another related contribution is presented in [2], where the accurate approximation of packet delay statistics is discussed. In [10], SR ARQ was investigated in the presence of feedback errors, and it was discussed how this affects the packet delays.

Various ARQ techniques have been analyzed and delay statistics have been compared with respect to varying packet error and packet arrival rate in [11]. This work provides the average results for delay faced by various ARQ techniques, and no exact analysis has been performed. Further, the authors do not consider data-bundling or different channel error distributions.

Our approach to characterize the delay statistics can especially be applied by targeting the application of SR ARQ to scenarios where the packet size is very small and packet interarrival time is (relatively) short, for example IP telephony and audio conferencing [12] and other examples in [13]. In these cases, data-bundling can effectively be used to reduce the latency for the flow. In this context, we analyze and compare the performance of SR ARQ with and without data-bundling over a wireless channel represented via a discrete time Markov chains (DTMCs), and investigate the buffer occupancy, queueing delay as well as delivery and overall delay statistics. The results highlight the practicality of the data-bundling approach, which performs well especially when correlated errors are present. Also, our approach may be used to derive concrete guidelines for the setup of bundling in real contexts.

The rest of the paper is organized as follows: Section II presents the SR ARQ queueing model and related Markov processes. The channel model is then defined. In Section III, the data-bundling functionality is introduced and the complete delivery processes is discussed in detail. The analytical results are discussed in Section IV, and finally, Section V concludes the findings and enlists future work.

II. SR ARQ Queueing Model and Markov Processes

We model a wireless channel as a Markov chain with SR ARQ over a slotted time where a packet transmission takes place at each time slot. We assume the round trip time as m which also refers to the SR ARQ window size. We assume $m > 1$ so that the packet transmission outcome is known after m slots. Packets are transmitted continuously as far as there are available packets in the queue.

We consider the transmission of packets between two entities (that is, a transmitter and a receiver) over a wireless noisy link with an unlimited number of retransmissions. For error recovery, the SR ARQ technique is used, and one time slot corresponds to one packet transmission and a feedback packet arrives after a full round-trip-time (RTT) (m slots), containing either an acknowledgement (ACK) or negative acknowledgement (NACK) message. The sender transmits packets in arrival order as long as ACKs are received. Once a NACK is received, a retransmission is scheduled. Since NACKs can only be received after a full RTT, a retransmission is only scheduled after m slots. The data packets are only released when the lower identifiers have been acknowledged, that is, *in-order* to higher layers. Generally for SR ARQ, packets that are correctly received but not yet released to the higher layer on the receiver side due to the *in-order* requirement are kept in the sender side buffer until successfully delivered and ACKed.

For the packet arrival process, we use a Bernoulli distribution with an arrival rate $\lambda < 1$, which refers to the probability of a packet arrival (and thus $1 - \lambda$ is the probability of no packet arrivals). Other more complicated arrival processes may be considered within our proposed framework, if needed. However, the Bernoulli distribution offers the advantage of conceptual simplicity and easier tractability (the parameter λ can be regulated to simulate different load conditions: the larger the λ, the heavier the load).

A discrete time Markov chain (DTMC) is used to represent a wireless channel so as to consider error correlation as well. The temporal slot of chain is equal to packet transmission time. We assume two states, G and B, representing Good and Bad, respectively. For the sake of simplicity, we assume G has packet error probability 0 and B has packet error probability 1, that is, in good state transmission always succeeds and the bad state is always erroneous. Of course, it is just a modeling choice and we can change these probabilities as being low ε for G and high ε for B, which will be straightforward. The transition from one state to another (or itself) is made with a probability which is referred to as p_{ij} for transition from state i to j. These values can be collected into the transition matrix $\mathbf{P} = \{p_{ij}\}$, such that $i, j \in \{G, B\}$.

The analysis can be easily extended to a higher number of states and similar results can be obtained qualitatively, that is the trend of results will be same. The main advantage of using Markov chains for a wireless channel is the possibility of tuning parameters to get channel correlation. The channel transition probability matrix can be written as:

$$\mathbf{P} = \begin{pmatrix} p_{gg} & p_{gb} \\ p_{bg} & p_{bb} \end{pmatrix} \tag{1}$$

The steady-state channel error probability is given by:

$$\varepsilon = \frac{p_{gb}}{p_{gb} + p_{bg}} \tag{2}$$

and the average error burst length is $B = 1/p_{bg}$. If $B = \varepsilon^{-1}$, then the channel errors are i.i.d, otherwise (which usually means $B > \varepsilon^{-1}$) errors are correlated. The transition probability matrix at step i can then be written as:

$$\mathbf{P}(i) = \mathbf{P}^i = \begin{pmatrix} p_{gg}(i) & p_{gb}(i) \\ p_{bg}(i) & p_{bb}(i) \end{pmatrix} \tag{3}$$

The queueing system is stable if $\lambda < 1 - \varepsilon$. Otherwise, instability ensues; such a condition may still be used to evaluate the system in the "Heavy Traffic" condition, meaning that in the long run a packet will always be present in the queue [6].

ACK/NACKs are assumed to be error-free and an unlimited queueing buffer size at transmitter-side is used for this model. These are very common assumptions for the sake of clarity and they do not change the analysis substantially [10]. Further, we assume that the link layer is fully reliable, that is, every packet is transmitted (and retransmitted if needed) until it is received correctly.

Exploiting the data-bundling mechanism for retransmissions gets more interesting when considering bursty channels where errors are correlated. For such scenarios, data-bundling can be beneficial in terms of reducing delays. In the subsequent section, we discuss in detail the method for redundantly bundling data on retransmission and the delivery process.

One of the motivations is to investigate the buffer occupancy in case of small-packet streams, and analyze if the data-bundling mechanism is advantageous or not. We further investigate the distribution of the Markov channel burstiness and its effect on queueing, delivery and overall delay statistics.

III. Data-bundling and Delivery Process

When the packets transmitted are relatively small, it may be possible to fit more than one packet into a single transmission unit of the network. If it is possible to redundantly send unacknowledged data from the send queue, it is in many cases possible to avoid the delay that is incurred by waiting for feedback from the receiver that triggers a retransmission. For the transport layer, this was discussed by Evensen et al. in [4]. In this way, a bundling mechanism reduces the recovery latency when loss occurs at the cost of spending more bandwidth resources. We have devised a model for the delivery process of a wireless network that includes redundantly bundling one segment if possible. Of course, we can bundle more than two packets but eventually the size of the bundled packets would exceed the maximum frame size, for the sake of simplicity we consider data-bundling for two packets only, further bundling would be straightforward.

As far as the delivery process is concerned, an m sized data-bundling and retransmission window is used to track the status of past m transmissions. We use m sized vectors for data-bundling and the retransmission window, $\mathbf{b}$ and $\mathbf{c}$, respectively, with elements $b_i, c_i \in \{0,1\}, 1 \leq i \leq m$. The m^{th} bit represents the slot under transmission at time t. The bits c_j, b_j, $1 \leq j \leq m-1$ refer to the transmission and data-bundling status information at time $t-m+j$, respectively. We track the data-bundling status information because we restrict our system to bundle the data only once. That is, if an already bundled packet is lost and needs to be retransmitted, it will not be bundled with the third packet, it will be transmitted as is.

For the vector $\mathbf{b}$, the value 0 refers to no data-bundling performed at time t, whereas 1 refers to the state where a new packet was bundled with a packet that needs to be retransmitted. On the other hand, for vector $\mathbf{c}$, the value 0 refers to a successful transmission of the packet and 1 refers to a failed transmission. Further, we need to track the number of packets in queue q and the channel state s at time t. As per our approach the channel can be either Good or Bad, which we represent by $\{0,1\}$, respectively. It is sufficient to keep track of state s at time t, due to the Markovian nature of the channel.

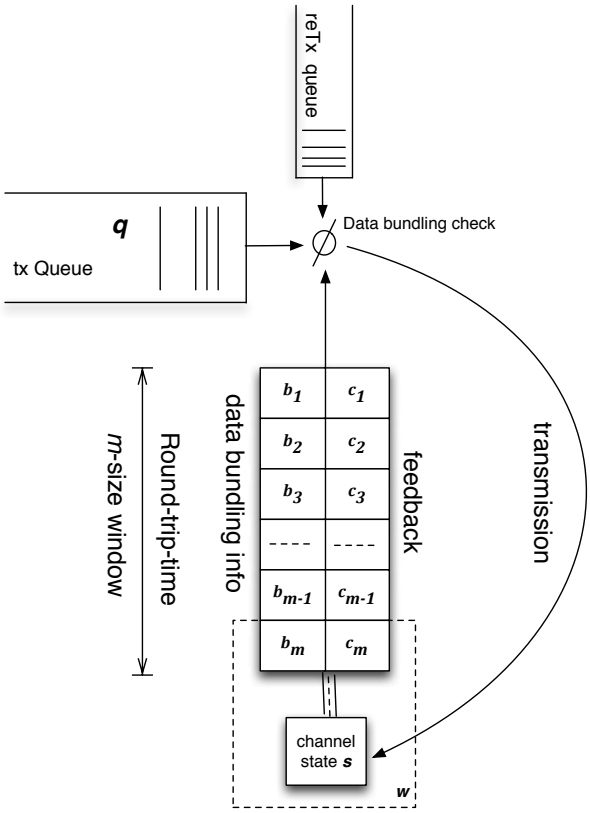

Fig. 1. SR ARQ and data-bundling Transmission Process

The full state of the delivery process can now be described by the quadruple $(q(t), \mathbf{c}(t), \mathbf{b}(t), s(t)) = (q, c_1, c_2, \ldots, c_m, \mathbf{b}, s)$. Here, c_m and s are not independent, as the retransmission is scheduled ($c_m = 1$) only when the channel is bad ($s = 1$). It is not possible that ($s = 0$) and ($c_m = 1$), but the reverse does not hold. That is, with $s = 1$ it can be that $c_m = 0$, which represents the case when there is no packet in the transmitter's queue. Thus, we can use another tertiary variable w to represent the following three cases. It is important to distinguish between case 1 and case 3 of the following because, both represent no-retransmission scheduling, but for different states. In case $w = 0$, the state is good and no retransmission scheduled, in case $w = 1$, a state is bad and a retransmission is scheduled, and finally, in case $w = -1$, the state is good but no packet is transmitted.

The Markov chain $X(t) = (q(t), \mathbf{c}(t), \mathbf{b}(t), s(t))$ can now be rewritten as $X(t) = (q, c_1, c_2, \ldots, c_{m-1}, \mathbf{b}, w)$. Note that we omit the time t here for the simplicity of expression. Now, by considering every possibility for channel transition, we can determine the value of $X(t+1)$ based on $X(t)$. Therefore, we can derive the transition matrix $\mathbf{T}(\mathbf{P}, \lambda)$ which is a function of matrix $\mathbf{P}$ and the arrival rate λ.

A diagram of the delivery process is shown in Fig. 1. For each time slot, a packet is transmitted on the channel. It can be either a new packet transmission or a retransmission. In the

latter case, the data-bundling function is enabled. For which a packet that needs to be retransmitted is bundled with a new packet and then transmitted over a channel depending upon the status information from bundling vector b_i, $1 \leq i \leq m$, that is, if it is a retransmission of already bundled packet, then no further bundling will be done.

IV. Evaluation and Results

We evaluate the buffer occupancy by considering the variable arrival rate and different channel error processes. We investigate how the data-bundling functionality can be useful, so that retransmission delays are reduced. For this purpose, we used Monte-Carlo simulations over a large number of times so that we can have sufficient empirical measurements. Our main contribution is to analyze the data-bundling mechanism when SR ARQ is used as an error control technique. We have used various values of channel error probability ε, retransmission window size m and packet arrival rate λ. The results we obtained are discussed below.

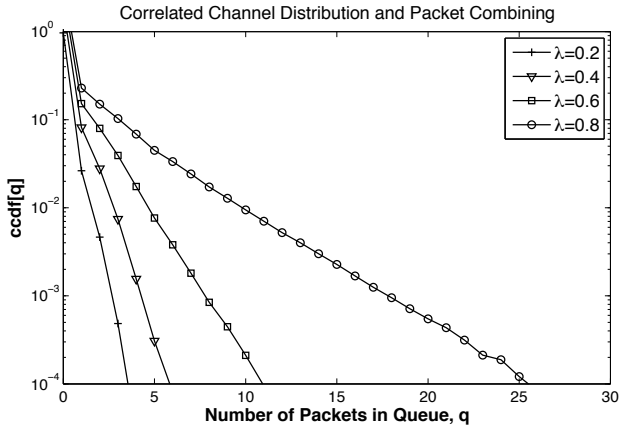

Fig. 2. Buffer Occupancy using data-bundling, B=5, m=4, ε=0.1

The probability of having buffer occupancy is defined as:

$$P[q] = \sum_{\mathbf{c} \in C} \sum_{w=-1}^{1} \pi\left(q, \mathbf{c}, \mathbf{b}, w\right) \tag{4}$$

where $\pi\left(q, \mathbf{c}, \mathbf{b}, w\right)$ is a stationary probability of a generic state defined earlier $X(t) = (q, c_1, c_2, \ldots, c_{m-1}, \mathbf{b}, w)$ and $C \in \{0,1\}^{m-1}$.

Fig. 2 presents the complementary cumulative distribution function of the queueing buffer occupancy for the correlated channel error process with burst size $B = 5$, and round-trip-time $m = 4$. As expected, the higher the packet arrival rate (λ), the higher the probability that the queueing buffer is occupied.

For data-bundling on retransmission using an *iid* error process, the results are depicted in Fig. 3. It shows that by using data-bundling, the queueing buffer occupancy can be reduced to some extent. But, as explained earlier, we must not neglect the correlated channel error process, which can be more realistic in wireless environment. Fig. 4 shows that using data-bundling can be beneficial for lower as well as for higher

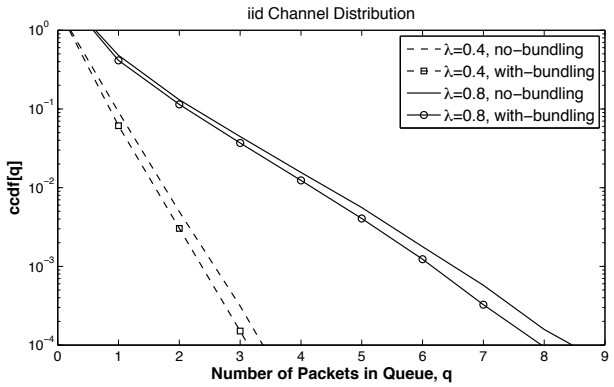

Fig. 3. Queue occupancy with and without bundling using an i.i.d. channel error process, B=5, m=4, ε=0.1

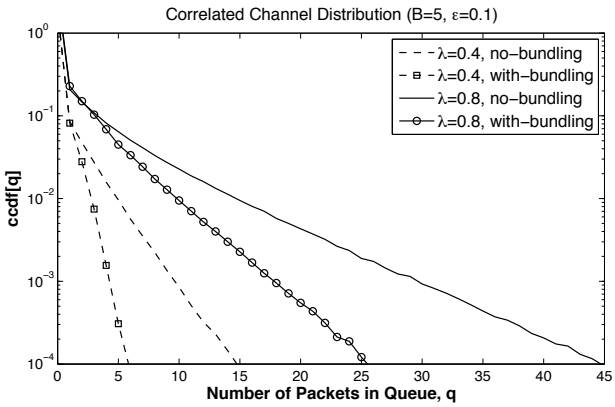

Fig. 4. Queue occupancy with and without bundling using a correlated channel error process, B=5, m=4, ε=0.1

packet arrival rates, but the gain is higher when the arrival rate is higher.

To evaluate the overall delay for a packet, we assume that the packet arrived in the buffer in the previous slot, and after its arrival, we turn off the arrival process so that we can compute the queueing delay τ_Q and the delivery delay τ_D. To compute τ_Q, the system has to reach the state where $q = 0$, we call this condition $\mathcal{Q}$. The delivery delay is then defined as the difference between τ_Q and the overall delay τ_G. We further define a condition $\mathcal{G}$ where not only $q = 0$ but also the entire system is empty, that is, all the values in $c_i = 0$, $1 < i \leq m-1$ and $q = 0$. Thus, formally, if we have the probabilities $f_{(q,\mathbf{c},\mathbf{b},w)\mathcal{Q}(t)}$ and $f_{(q,\mathbf{c},\mathbf{b},w)\mathcal{G}(t)}$ that the first passage time [14] from state $(q, \mathbf{c}, \mathbf{b}, w)$ takes t timeslots to reach the absorbing sets $\mathcal{Q}$ and $\mathcal{G}$. The queueing delay τ_Q statistics can now be computed as

$$P\{\tau_Q = t\} = \sum_{q=0}^{\infty} \sum_{\mathbf{c} \in C} \sum_{w=-1}^{1} \Lambda\left(q, \mathbf{c}, \mathbf{b}, w\right) f_{(q,\mathbf{c},\mathbf{b},w)\mathcal{Q}(t)} \tag{5}$$

Note that we use Λ to represent the conditional probability of being in state $(q, \mathbf{c}, \mathbf{b}, w)$ at time t, given that a packet arrived at time $t-1$. Fig. 5 shows the complementary cumulative distribution function for τ_Q for the comparison of data-bundling functionality. It shows that there is little

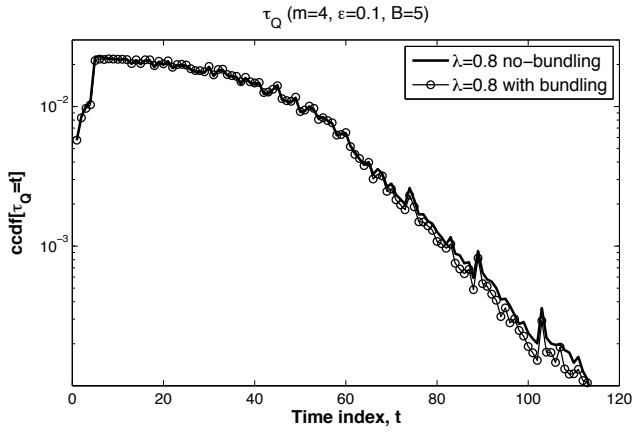

Fig. 5. Queueing delay τ_Q with and without bundling, $\lambda = 0.8, B{=}5, m{=}4, \varepsilon{=}0.1$

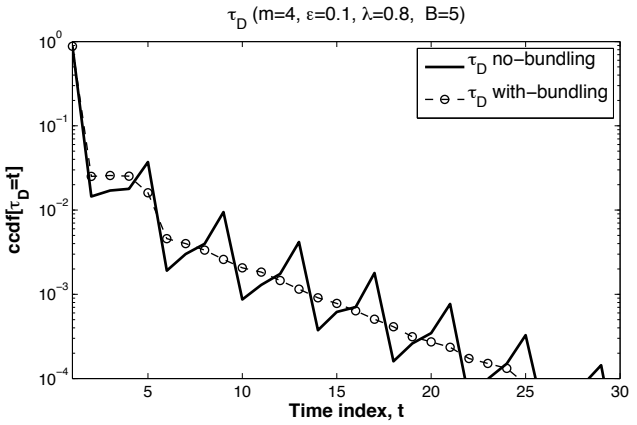

Fig. 6. Delivery delay τ_D with and without bundling, $\lambda = 0.8, B{=}5, m{=}4, \varepsilon{=}0.1$

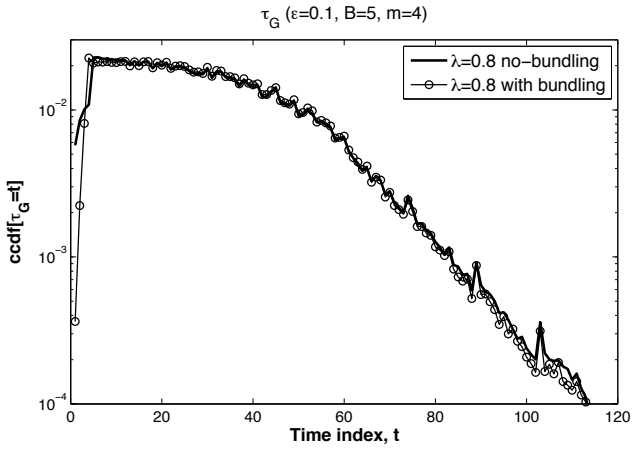

Fig. 7. Overall delay τ_G with and without bundling, $\lambda = 0.8, B{=}5$, $m{=}4$, $\varepsilon{=}0.1$

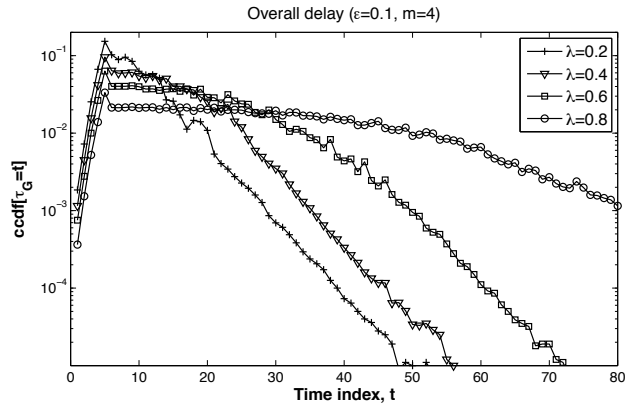

Fig. 8. Overall delay τ_G vs. λ, $B{=}5, m{=}4, \varepsilon{=}0.1$

gain in queuing delay when we do data-bundling. However, we can see significant improvements for buffer occupancy when data-bundling is performed, as shown in Fig. 4. For example, average buffer occupancy is reduced by 25% when data-bundling is performed.

Now, to find the overall delay statistics, it is sufficient to replace the condition $\mathcal{Q}$ with $\mathcal{G}$. Recall that for condition $\mathcal{G}$, not only the queue must be empty but the entire system should be empty. Thus, overall delay is computed as

$$P\{\tau_G = t\} = \sum_{q=0}^{\infty} \sum_{\mathbf{c}\in C} \sum_{w=-1}^{1} \Lambda\left(q, \mathbf{c}, \mathbf{b}, w\right) f_{(q,\mathbf{c},\mathbf{b},w)\mathcal{G}(t)} \quad (6)$$

Computing the delivery delay τ_D from τ_G and τ_Q is straightforward: $\tau_D = \tau_G - \tau_Q$, the difference between overall delay and queueing delay [7].

Fig. 6 and 7 shows the complementary cumulative distribution function for τ_D and τ_G, respectively. In Fig. 6 delivery delay statistics are presented. It can be seen from the figure that data-bundling mechanism provides smoother curves. The overall delay (τ_G) statistics show that there is little gain if data-bundling is performed. On the other hand, Fig. 8 represents the impact of variable packet arrival rate on the overall delay statistics for a bursty channel with $B = 5$ and $\varepsilon = 0.1$. It illustrates that when the packet arrival rates get high, the overall delay increases.

Finally, Fig. 9 represents the comparative analysis of various error rates over the Markov channel and shows how the data-bundling performs for both the *iid* and correlated channel distributions. We take $\varepsilon = 0.1$ and 0.01 with retransmission window size $m = 7$ and packet arrival rate $\lambda = 0.4$. It shows that with respect to error rates, the bursty channel, the data-bundling seems to be beneficial for emptying the queue earlier. Of course, there is a trade-off between applying data a bundling mechanism and memory usage to keep track of the status of bundled packets and the bandwidth consumption is higher if the actual system is not slotted.

V. Conclusions

Data-bundling is considered useful for latency-critical traffic consisting of small packets, like the traffic of many interactive multimedia applications, to reduce retransmission delays for end-to-end transfers by local measures. In this study, we have applied the bundling principle to the link layer of a lossy wireless network. We have analyzed the impact of data-bundling by using the SR ARQ retransmission process for variable packet arrival rate over Markov channels. Not

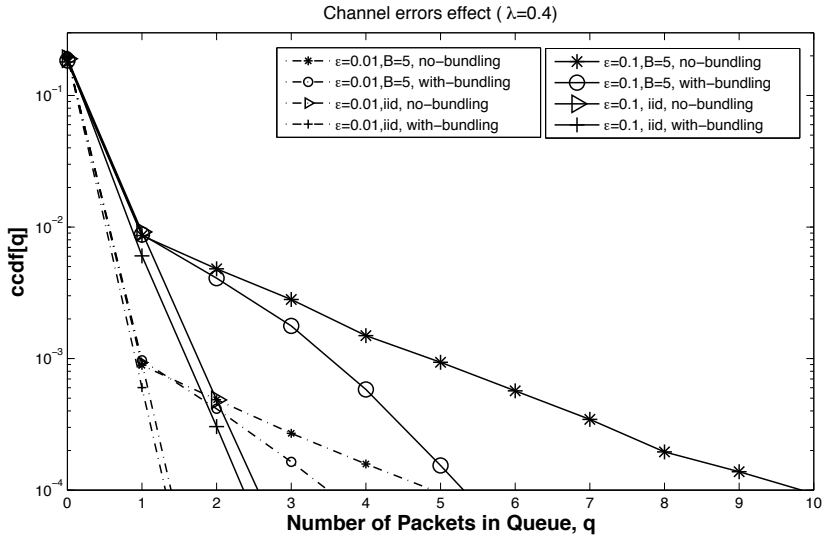

Fig. 9. Channel error process with and without bundling, λ=0.4, m=7

only, we have investigated the *iid* channel error process but also bursty channels with correlated channel error, which indeed provides a better approximation of realistic network behavior. We have analyzed the buffer occupancy both for *iid* and bursty channels. We found that for the *iid* channel distribution, data-bundling provides only very small gains. However, when the correlated channel error distributions with bursts are considered, data-bundling performs far better with respect to buffer occupancy. The trade-off is, of course, that the data for the previous m slots is kept in memory and the that we use more bandwidth than normal ARQ.

For future work, we would like to consider different error probability for bundled packets. To do that we must modify the model so that the error probability increases with the larger packet sizes for the transmission of packets that are bundled. We further plan to investigate and compare the presented mechanism with other error control mechanisms, like hybrid ARQ and FEC.

VI. Acknowledgement

The authors are funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project(ICT-317700). The views expressed are solely those of the authors.

References

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2011-2016," Cisco, May 2012.

[2] M. Rossi, L. Badia, and M. Zorzi, "Accurate approximation of ARQ packet delay statistics over Markov channels with finite round-trip delay," in *Proc. of IEEE WCNC*, 2003, pp. 1767–1772.

[3] C. Holman, K. A. Harras, K. C. Almeroth, and A. Lam, "A proactive data bundling system for intermittent mobile connections." in *Proc. of IEEE SECON*, 2006, pp. 216–225.

[4] K. R. Evensen, A. Petlund, C. Griwodz, and P. Halvorsen, "Redundant bundling in TCP to reduce perceived latency for time-dependent thin streams," *IEEE Communications Letters*, vol. 12, no. 4, pp. 334 – 336, 4 2008.

[5] A. Vulimiri, O. Michel, P. B. Godfrey, and S. Shenker, "More is less: reducing latency via redundancy," in *Proc. of ACM HotNets*, 2012, pp. 13–18.

[6] M. Rossi, L. Badia, and M. Zorzi, "On the delay statistics of SR ARQ over Markov channels with finite round-trip delay," *IEEE Trans. on Wireless Communications*, vol. 4, no. 4, pp. 1858–1868, 2005.

[7] L. Badia, M. Rossi, and M. Zorzi, "SR ARQ packet delay statistics on Markov channels in the presence of variable arrival rate," *IEEE Trans. on Wireless Communications*, vol. 5, no. 7, pp. 1639–1644, 2006.

[8] L. Badia, "On the impact of correlated arrivals and errors on ARQ delay terms," *IEEE Trans. Commun.*, vol. 57, no. 2, Feb. 2009.

[9] Z. Rosberg and N. Shacham, "Resequencing delay and buffer occupancy under the selective-repeat ARQ," *IEEE Trans. on Information Theory*, vol. 35, no. 1, pp. 166–173, 1989.

[10] L. Badia, "On the effect of feedback errors in markov models for SR ARQ packet delays," in *Proc. of IEEE GLOBECOM*, 2009, pp. 1–6.

[11] Y. Qin and L.-L. Yang, "Delay comparison of automatic repeat request assisted butterfly networks," in *Proc. of ISWCS*, 2010, pp. 686–690.

[12] "Long term evolution protocol overview," Freescale Seminconductor, Tech. Rep., 10 2008.

[13] A. Petlund, "Improving latency for interactive, thin-stream applications over reliable transport," Ph.D. dissertation, Simula Research Laboratory / University of Oslo, Oslo, Norway, 2009.

[14] R. Howard, *Dynamic probabilistic systems*. Wiley, 1971, no. v. 2.