# A Markov Analysis of Selective Repeat ARQ with Variable Round Trip Time

Leonardo Badia, *Senior Member, IEEE*

***Abstract*—Many papers analyze selective repeat automatic repeat request schemes by means of a Markov chain representation for the channel and, by extension, the whole transmission system. This Markov approach precisely characterizes the queueing behavior and the statistics of several delay terms. In the vast majority of the investigations, the round-trip time of the channel, which determines the instant of reception of the feedback from the receiver, is taken as a fixed value. This letter explores the relaxation of this assumption, still framing the system within a Markov chain. The main conclusion is that the impact of a variable round trip time on the delay statistics is rather limited. The used approach can be promptly applied to any similar analysis of retransmission-based error control systems.**

***Index Terms*—Queueing analysis, automatic repeat request, Markov processes, error analysis.**

## I. Introduction

AUTOMATIC retransmission request (ARQ) is a widely employed technique for error control. Its implementation through a selective repeat (SR) scheme consists of transmitting data as a continuous and ordered stream of packets, which is preempted by retransmissions of packets that were in error at their previous transmission attempt. Retransmissions can be performed when the transmitter is notified of the errors by the receiver, i.e., after one *round-trip time* (rtt) from the first transmission. If no retransmission needs to be performed, normal transmissions are resumed from the last packet sent [1].

For each received packet, the receiver sends back to the transmitter either a positive or a negative acknowledgement, and a new transmission or a selective retransmission is performed accordingly. This procedure can trigger error protection on-demand, i.e., only when errors are present, in contrast with proactive techniques such as forward error correction (FEC). With unlimited retransmission attempts, the achieved reliability can be arbitrarily high. However, there are some prices to pay. First, a separate feedback channel is needed. Second, SR ARQ also requires a resequencing buffer to store pending packets that have been received out-of-order. Finally, and most importantly from the performance standpoint, the packet delays may grow, since retransmissions are involved. It is worthwhile noting that even the so-called *hybrid ARQ* schemes [2], which are a mixture of ARQ and FEC, still retain the rationale of ARQ, i.e., they perform retransmissions and therefore potentially increase the packet delay. The characterization of packet delays in SR ARQ is an important topic, and has been the subject of many research papers [3]–[10].

Manuscript received June 24, 2013. The associate editor coordinating the review of this letter and approving it for publication was S. De.

L. Badia is with the Dept. of Information Engineering (DEI), University of Padova, via Gradenigo 6B, Padova, Italy (e-mail: badia@dei.unipd.it).

Digital Object Identifier 10.1109/LCOMM.2013.092813.131446

In the literature, several delay terms are defined. The existing taxonomy is here summarized by considering two main delays, i.e., the *queueing* and the *delivery* delay, which together form the *overall* packet delay. The former is the time from the arrival of a packet in the queue until its first transmission. The latter is the further time spent by the packet before being *released* from the receiver's resequencing buffer, which implies the correct reception of not only this packet, but also all those with lower sequence numbers [5].

In the aforementioned studies, these delay terms have been studied with many techniques, always considering a fixed rtt. One notable exception is [6], giving an *approximate* queueing analysis under Bernoulli packet service; to some extent, this randomness in the service process can be seen as an indirect consequence of a variable rtt, although the authors do not make such a connection and only consider a fixed rtt. Instead, the present letter develops an *exact* Markov analysis, where both queueing and delivery delays are evaluated by *directly* considering a variable rtt for each packet, which is shown to lead to interesting conclusions. The used approach is fairly general and can be applied to any Markov analysis of SR ARQ, e.g., [3], [11], [12], and even those considering different types of ARQ [7], [13], or hybrid ARQ [2], [14].

## II. Notations and Assumptions

Throughout this letter, a bold lower-case letter, such as $\mathbf{y}$, denotes a column vector, whose length is written as $\mathcal{L}_{\mathbf{y}}$; the same letter in italic with an index refers to an element, i.e., $y_i$ is the $i$th element of $\mathbf{y}$, where $1 \leq i \leq \mathcal{L}_{\mathbf{y}}$. If $\mathbf{y}$ only contains non-negative integers, $\mathbf{y}^-$ is obtained by decreasing all non-zero elements of $\mathbf{y}$ by one, i.e., $\mathbf{y}^- = [\mathbf{y} - \mathbf{1}]^+$, where $\mathbf{1}$ is an all-one vector and $[\cdot]^+ = \max(\cdot, 0)$ is applied element-wise.

If $\mathbf{y}$ contains $\ell$ elements equal to 1, for any $0 \leq k \leq \ell$ denote with $\mathbf{y}^{(k)-}$ the vector where all the elements larger than 1 and the first $k$ elements equal to 1 are decreased by 1:

$$(y^{(k)-})_i = \begin{cases} y_i - 1 & \text{if } y_i > 1 \\ y_i - 1 & \text{if } y_i = 1 \text{ and } i \leq \text{pos}_1(\mathbf{y}, k) \\ y_i & \text{if } y_i = 1 \text{ and } i > \text{pos}_1(\mathbf{y}, k) \\ y_i & \text{if } y_i < 1 \end{cases} \quad (1)$$

where $\text{pos}_1(\mathbf{y}, k)$ is the position of the $k$th "1" in $\mathbf{y}$. Finally, $\mathbf{e}_i$ denotes the $i$th element of the canonic basis, i.e., the $i$th column of the identity matrix $\mathbf{I}$.

The analysis considers a slotted time where identical packets are transmitted from a first-in-first-out (FIFO) buffer, unless retransmissions need to be performed, which, in the SR ARQ scheme, have priority over normal transmissions. The rtt is denoted by $M$, taken without loss of generality as an integer value (otherwise, it is rounded up). Thus, $M$ is the number of slots elapsed between the transmission of a packet and the

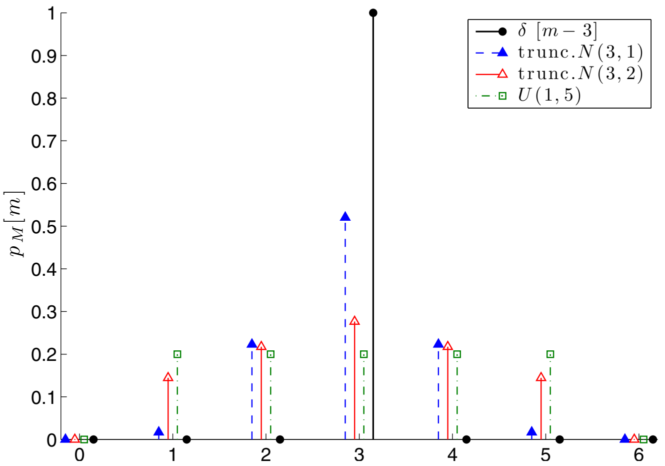


Fig. 1. Examples of $p_M[m]$: from left to right, two truncated Gaussian distributions with different standard deviations and a uniform distribution. The right-most one is the reference deterministic case, where $M = 3$.

notification of its outcome back at the transmitter's side. In most of the literature, $M$ is assigned a constant value $\bar{m}$,[1] also called the *ARQ window* [1], i.e., the number of packets that are still pending acknowledgement. For a Markov description of the system, the number of states is exponential in $\bar{m}$ [9].

This letter considers a *variable* rtt $M$ with discrete distribution $p_M[m] = \Pr\{M{=}m\}$. For the sake of analytical tractability, let $p_M[m]$ be zero outside $\{m_{\min}, \ldots, m_{\max}\}$ and rtt values of different packets be independently drawn. The assumption of *ideal ARQ* [5], [13] implies that $p_M[m]{=}\delta[m{-}1]$, i.e., equal to 1 for $m{=}1$ and 0 elsewhere. A larger, but still constant, rtt is considered by [1], [3], [4], [7]–[10], which means that $p_M$ is still a delta-function, i.e., $p_M[m] = \delta[m{-}\bar{m}]$, centered on $\bar{m} > 1$. The present analysis instead can be adapted to any $p_M$ of choice. The numerical evaluations use the functions shown in Fig. 1, all with $m_{\min}{=}1$, $m_{\max}{=}5$, and mean $\bar{m}{=}3$: a truncated Gaussian distribution with standard deviation $\sigma{=}1$, another truncated Gaussian with $\sigma{=}2$, and a uniform distribution.

Thus, the feedback of a packet is known at the transmitter's side after $m$ slots with probability $p_M[m]$. After $j{<}m$ slots the packet is pending acknowledgement, which will be received after $m{-}j$ slots. A packet transmitted $j$ slots ago whose rtt is equal to $m$ will be said to have a *feedback delay* equal to $m{-}j$ at present time. For discussion ease, the following work assumptions, quite common for ARQ, are made; it would be immediate to remove them at the price of a longer and tedious, but substantially equivalent, analysis. Packet errors are independent and occur with probability $\varepsilon$, and packet arrivals are independently distributed, following a Bernoulli distribution, i.e., at most one packet arrives at any slot with probability $\lambda$ (the system is stable if $\lambda < 1 - \varepsilon$). Also, consider error-free feedback and unlimited retransmission attempts. The effects of correlation, feedback errors, and limited retransmissions was investigated in [11], [12], [14], respectively; yet, none of these papers considers a variable rtt as done here. The reader is referred to them for an in-depth discussion on these points.

[1]This may be justified by a stringent feedback time-out. The present analysis can also be seen as a relaxation of this hypothesis.

## III. Analysis

A Markov analysis of SR ARQ systems [8] frames the queue, the arrival process, the packet error process, and the outcome of the past packets in the state of a chain whose transitions are described by a matrix $\mathbf{T}(\lambda)$, depending on $\lambda$. For the problem at hand, the system state is $\mathbf{s} = (q, \mathbf{d})$ with $q$ being the queue length, which can only increase/decrease by 1, or stay unchanged, at each slot. The $(m_{\max})$-sized vector $\mathbf{d}$ contains the feedback delays for the packets pending acknowledgement. Any entry $d_i$ represents a feedback due in $d_i$ slots; thus, there can be at most $m_{\max}$ pending feedbacks.

Zero-valued elements may be present in $\mathbf{d}$, e.g., if some of the past $m_{\max}$ time slots were without arrivals. In the case of fixed rtt, all non-zero elements of $\mathbf{d}$ are different, as the instantaneous value of the feedback delay is different for each past packet. If $M$ is variable, instead, there may be duplications. However, it is immediate that, for any meaningful state $\mathbf{s}$, there is always at least one element in $\mathbf{d}$ that is either 0 or 1. Denote with $\zeta$ the lowest index of such elements, i.e.,

$$\zeta = \arg \min_{i=1 \ldots \mathcal{L}_{\mathbf{d}}} (d_i \leq 1) \, . \tag{2}$$

For $\mathbf{s} = (q, \mathbf{d})$, define $\mathbf{u}$ as the list of indices $j$ for which $d_j = 1$. Thus, $\mathcal{L}_{\mathbf{u}}$ is the number of feedbacks due to arrive in the next time slot and the probability that none of them is negative is $\alpha = (1 - \varepsilon)^{\mathcal{L}_{\mathbf{u}}}$. Also, for any $k$ in $\{1, \ldots, \mathcal{L}_{\mathbf{u}}\}$, $a_k = \varepsilon(1 - \varepsilon)^{k-1}$ is the probability that the lowest index of a negative feedback is $u_k$. Trivially, $\alpha + \sum_{k=1}^{\mathcal{L}_{\mathbf{u}}} a_k = 1$.

For a transition from $\mathbf{s} = (q, \mathbf{d})$ to $\mathbf{s}' = (q', \mathbf{d}')$, it is important to represent the case where simultaneous feedbacks reach the transmitter at the same time, which happens when $\mathcal{L}_{\mathbf{u}} > 1$. The precise mathematical description of this situation depends on the nature of acknowledgement packets in ARQ. Hereafter, it is assumed that simultaneous reception of multiple acknowledgements is possible within the same time slot (think of acknowledgements as much shorter than the data packet sent in one slot). However, it is surely not possible to have two or more simultaneous *retransmissions* in the same slot.

The transitions when $\mathcal{L}_{\mathbf{u}} > 1$ are as follows. If all packets are acknowledged, which happens with probability $\alpha$, a new packet is transmitted, if available; or, if all incoming feedbacks are positive, the queue is empty, and no packet arrives, the system stays idle until the next slot. Else, incoming feedback messages are scanned in order of their index within vector $\mathbf{d}$. As long as those in $u_1, u_2, \ldots$ are correct, they are set to 0. If the first negative acknowledgement is in $u_k$ (probability $a_k$), the corresponding packet is retransmitted and the evaluation of other incoming feedbacks is deferred to the next time slot.

So, at most one negative feedback, the $u_k$th one, is considered at a time, while all $d_{u_j}$s with $j = k{+}1, \ldots, \mathcal{L}_{\mathbf{u}}$ are kept to 1 and evaluated at the next transition. The delay evaluations are not affected, because, if all the feedbacks at positions $u_{k+1}, u_{k+2} \ldots, u_{\mathcal{L}_{\mathbf{u}}}$ are positive, they will be removed at the next transition, with no impact on the delays. Conversely, if at least one of them is negative, there will be a retransmission, thereby increasing the delay, but not before the next slot, as the current time slot already has one (that described by $u_k$).

Finally, consider the case where a new feedback delay, either of a retransmission or a new packet, is stored in $\mathbf{d}'$. To simplify the matrix derivation, assume that this new value is stored in position $\zeta$, which guarantees to be available. In fact, the value of $d_\zeta$ is either 0, which means no feedback delay is recorded, or 1, i.e., the corresponding feedback arrives in this time slot and a new feedback (possibly even of the same packet if it is retransmitted) can be recorded instead. As per (2), $\zeta$ is a valid position even when multiple simultaneous feedbacks arrive.

Thus, the transition matrix $\mathbf{T}(\lambda)$ can be written by considering the following transition terms from $\mathbf{s}$ to $\mathbf{s}'$, for all $j \in \{m_{\min}, \ldots, m_{\max}\}$, $k \in \{1, \ldots, \mathcal{L}_{\mathbf{u}}\}$:

$$\lambda\, \alpha\, p_M[j] \quad \text{if } \mathbf{s}' = (q,\, \mathbf{d}^{-}{+}j\mathbf{e}_\zeta) \tag{3}$$

$$(1{-}\lambda)\, \alpha\, p_M[j] \quad \text{if } \mathbf{s}' = (q{-}1,\, \mathbf{d}^{-}{+}j\mathbf{e}_\zeta) \text{ and } q{>}0 \tag{4}$$

$$(1{-}\lambda)\, \alpha \quad \text{if } \mathbf{s}' = (0,\, \mathbf{d}^{-}) \text{ and } q{=}0 \tag{5}$$

$$\lambda\, a_k\, p_M[j] \quad \text{if } \mathbf{s}' = (q+1,\, \mathbf{d}^{(k)-}{+}j\mathbf{e}_\zeta) \tag{6}$$

$$(1{-}\lambda)\, a_k\, p_M[j] \quad \text{if } \mathbf{s}' = (q,\, \mathbf{d}^{(k)-}{+}j\mathbf{e}_\zeta)\ . \tag{7}$$

Some equations may describe the same transition $\mathbf{s} \to \mathbf{s}'$, for instance (3) and (7) if $k = \mathcal{L}_{\mathbf{u}}$, thus the left-hand terms needs to be summed in the corresponding element of $\mathbf{T}(\lambda)$.

The transition in (3) corresponds to all incoming feedbacks being positive and the arrival of a new packet in the queue, which happens with probability $\lambda\alpha$. A packet is taken from the queue, whose size remains the same, and is placed in $\zeta$, i.e., the first available position of $\mathbf{d}$; this is the result of summing $j\mathbf{e}_\zeta$. The value of the new feedback delay is $j$ with probability $p_M[j]$. All the non-zero delay values are decreased by 1, which is represented by $\mathbf{d}$ transiting to $\mathbf{d}^{-}$. Identical reasonings apply to (4) and (5), which assume instead no packet arrival, therefore $\lambda$ is replaced by $1{-}\lambda$. Either $q > 0$, as in (4), then the queue is decreased by 1 and the new feedback delay is placed in position $\zeta$, or the queue is empty, so all non-zero delays in $\mathbf{d}$ are just decreased by 1, which is described by (5). The two remaining equations report instead the transitions when one of the arriving feedbacks, precisely the $k$th one, is negative, which happens with probability $a_k$. Again, either a new packet arrives with probability $\lambda$, as per (6), or none arrives with probability $1{-}\lambda$, as per (7). This time, no packet is taken from the queue, as the retransmission is prioritized, therefore the queue size is either increased by 1 or kept unchanged, respectively. As per the discussion above, the evolution of $\mathbf{d}$ is to $\mathbf{d}' = \mathbf{d}^{(k)-}$, i.e., only the first $k$ elements equal to 1 are decreased, the remaining ones are kept at 1 and evaluated at the next time slot, plus $j\mathbf{e}_\zeta$ is added as before.

The vector $\boldsymbol{\sigma}$ of the steady-state probabilities satisfies $\mathbf{T}(\lambda)\boldsymbol{\sigma} = \boldsymbol{\sigma}$ and $\sum_i \sigma_i{=}1$. To solve numerically, one needs to set a sufficiently high maximum queue size $Q_{\max}$, for which $Q_{\max}{+}1$ is replaced by just $Q_{\max}$. The number of states, i.e., the size of $\mathbf{T}(\lambda)$, is a finite value $N$, which is upper bounded by $Q_{\max}(m_{\max}+1)^{m_{\max}}$. Then,

$$\boldsymbol{\sigma} = \begin{bmatrix} \mathbf{T}(\lambda) - \mathbf{I} \\ \mathbf{1} \end{bmatrix}^{-1} \mathbf{e}_{N+1}\ . \tag{8}$$

Despite $N$ being large and exponential in $m_{\max}$, matrix $\mathbf{T}(\lambda)$ is sparse, see (3)–(7), thus (8) is relatively simple to compute.

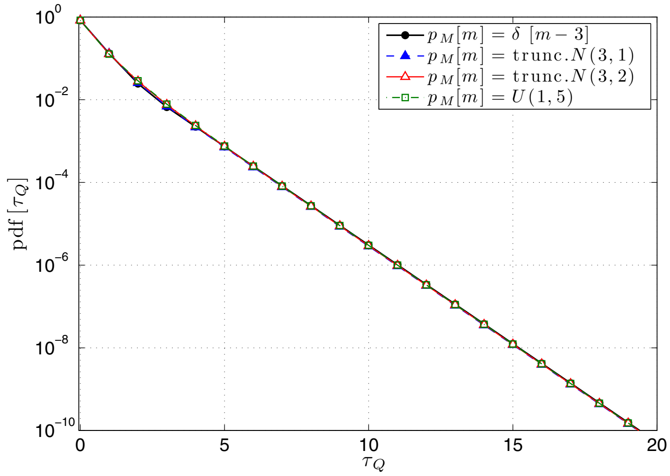


Fig. 2. Distribution of queueing delay $\tau_Q$, for $\lambda = 0.4$ and $\varepsilon = 0.2$.

After deriving $\boldsymbol{\sigma}$, the statistics of the delays can be found by proceeding as sketched here (for more details, see [11]). The probability that a packet arrives when the system is in a given state $\mathbf{s}$ is the $\mathbf{s}$th entry of vector $\mathbf{T}(1)\boldsymbol{\sigma}$. Note that this is not the steady-state probability of $\mathbf{s}$; the system was in steady-state one step before the arrival (due to the Markov property) and the last slot has surely seen an arrival, thus $\lambda = 1$. After the arrival of this packet, none of the subsequent arrivals will affect the delays [8], [15]. Thus, $\lambda$ can be turned off and the system evolves according to matrix $\mathbf{T}(0)$. The queueing delay $\tau_Q$ is the first passage time through any state with an empty queue ($q = 0$) and the overall delay $\tau_G$ is the first passage time to the state without pending packets, i.e., $q = 0$ and $\mathbf{d} = \mathbf{0}$ (all-zero vector). The delivery delay $\tau_D$ is the difference $\tau_G - \tau_Q$.[2]

## IV. Numerical Results

Through the analysis, the distribution of the queueing delay $\tau_Q$ can be derived, as shown in Fig. 2, for $\lambda = 0.4$ and $\varepsilon = 0.2$. Remarkably, the four distributions for various choices of $p_M$ almost coincide. Although not shown due to space limitations, the same holds for other choices of $\lambda$ and $\varepsilon$, meaning that the distributions are very similar, even though they all depend on the load factor $\lambda/(1-\varepsilon)$. The first point, i.e., $\Pr\{\tau_Q = 0\}$ has the exact same value for any choice of $p_M$. This follows from the probability of finding an empty queue being identical for every $p_M$ with the same average of $M$. However, the curves match with striking similarity also for higher values of $\tau_Q$.

Fig. 3 shows the same pdf (zoomed) for $\lambda{=}0.78$ and $\varepsilon{=}0.2$, so that the queue is closer to instability.This plot gives a better view of the (rather limited) differences between the curves. Only the two most diverse curves are plotted, i.e., those for fixed rtt and uniform $p_M$ in $\{1, \ldots, 5\}$. Simulation points are plotted to confirm of the analysis (all the results have been validated by simulation, they are shown only here to avoid too dense graphs). It can be seen that a variable rtt implies a lower value of the distribution for $\tau_Q = 1$ and a higher value

[2]This is evaluated at the transmitter's side. To evaluate it at the receiver's side, also the propagation delay (about half the rtt) must be summed. The derivation would be similar and the conclusions almost identical.

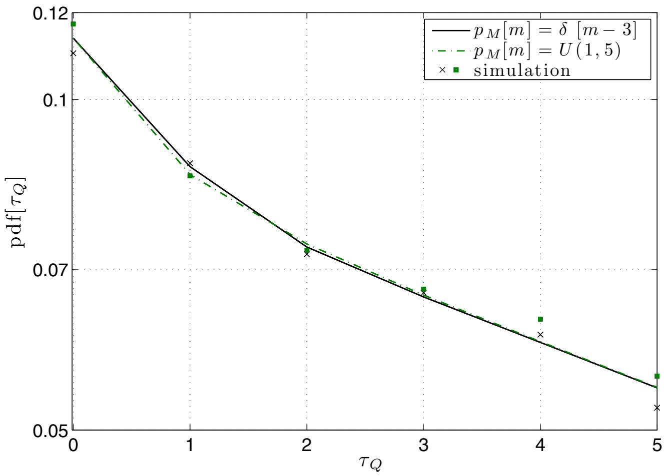


Fig. 3. Distribution of $\tau_Q$, for $\lambda = 0.78$ and $\varepsilon = 0.2$, comparison between fixed and variable rtt, and simulation results.

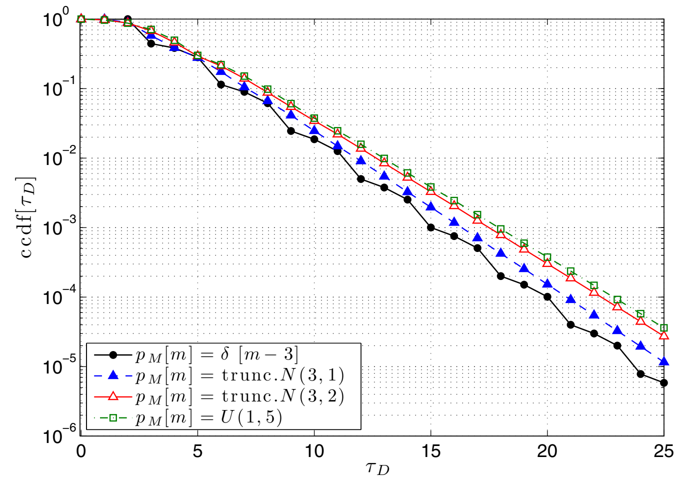


Fig. 4. Complementary cumulative distribution of the delivery delay $\tau_D$, for $\lambda = 0.78$ and $\varepsilon = 0.2$.

for subsequent $\tau_Q$s. However, after $\tau_Q = 6$ the curves become almost impossible to distinguish.

Fig. 4 shows the complementary cumulative distribution of $\tau_D$. A more visible effect is present, i.e., the curves for variable rtt are smoothened. As argued in [11], [15], the statistics with fixed rtt has a quasi-periodic behavior that the variable rtt filters out. Interestingly, these curves would be very similar to those (not shown due to lack of space) considering a different $\lambda$. The fact that $\tau_D$ is almost unaffected by the arrival process was found in [8],[3] but it is now extended to the case of variable rtt. Finally, the curves for variable rtt in Fig. 4 appear to have a slightly higher tail. Hence, the delivery delay for a variable rtt is larger, which was expected, as $\tau_D$ depends on all the delays of past packets. However, the increase is relatively limited. Even in the most variable case, i.e., the uniform distribution between 1 and 5 (which has a 20% chance that the rtt is 67% higher), the average $\tau_D$ is just 12% higher than when the rtt is fixed to 3. Such a difference occurs for a relatively high load factor (97.5%); if $\lambda/(1-\varepsilon)$ is lower, the curves are closer.

## V. Conclusions

This letter gives a Markov analysis of ARQ systems with variable rtt. Methodology aside, a notable quantitative conclusion is that the queueing delay statistics is only marginally affected by the rtt distribution. The results are almost identical (not only in terms of average delay, but for the full distribution of $\tau_Q$) even if the rtt is variable, as long as the average is the same. Therefore, it seems licit, for queueing statistics purposes, to use a fixed rtt equal to the average value. As a caveat, this holds true if the rtt is drawn independently for each packet, some differences may arise in a correlated case.

For the delivery delay, the impact of a variable rtt is more evident but still minor. Especially, a foreseeable smoothening and a slight increase of the curves are observed, but the qualitative behavior is similar. Thus, the approximation induced by assuming a fixed rtt can be regarded as limited.

[3]A recent contribution [16] argues that this may not hold under different assumptions, specifically, a channel that can change even during a packet transmission, and a strictly FIFO queue discipline. Surely this point deserves further investigation, which is left for future studies.

## References

[1] Z. Rosberg and M. Sidi, "Selective-Repeat ARQ: the joint distribution of the transmitter and the receiver resequencing buffer occupancies," *IEEE Trans. Commun.*, vol. 38, no. 9, pp. 1430–1438, Sep. 1990.

[2] C. J. Le Martret, A. Le Duc, S. Marcille, and P. Ciblat, "Analytical performance derivation of hybrid ARQ schemes at IP layer," *IEEE Trans. Commun.*, vol. 60, no. 5, pp. 1305–1314, May 2012.

[3] W. Luo, K. Balachandran, S. Nanda, and K. Chang, "Delay analysis of selective-repeat ARQ with applications to link adaptation in wireless packet data systems," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1017–1029, May 2005.

[4] H. Lee and B. Ngo, "Queuing analysis of the selective repeat ARQ scheme," *IEE Proc. Commun., Speech and Vision*, vol. 138, no. 6, pp. 487–493, Dec. 1991.

[5] J. G. Kim and M. M. Krunz, "Delay analysis of Selective Repeat ARQ for a Markovian source over a wireless channel," *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, pp. 1968–1981, Sep. 2000.

[6] M. E. Anagnostou and E. N. Protonotarios, "Performance analysis of the Selective-Repeat ARQ protocol," *IEEE Trans. Commun.*, vol. 34, no. 2, pp. 127–135, Feb. 1986.

[7] J. Chang and T. Yang, "End-to-end delay of an adaptive Selective Repeat ARQ protocol," *IEEE Trans. Commun.*, vol. 42, no. 11, pp. 2926–2928, Nov. 1994.

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

[9] K. De Turck and S. Wittevrongel, "Receiver buffer behavior for the selective repeat protocol over a wireless channel: an exact and large-deviations analysis," *J. Indust. Manag. Optimization*, vol. 6, no. 3, pp. 603–619, Aug. 2010.

[10] Y. H. Bae, B. Kim, B. D. Choi, and J. Kim, "Tail asymptotic behaviour of resequencing buffer content for selective repeat ARQ," *Oper. Res. Lett.*, vol. 39, no. 4, pp. 272–277, Jul. 2011.

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

[12] ——, "On the effect of feedback errors in Markov models for SR ARQ packet delays," in *Proc. IEEE Globecom*, Dec. 2009.

[13] C. Leung, Y. Kikumoto, and S. A. Sorensen, "The throughput efficiency of the go-back-N ARQ under Markov and related error structures," *IEEE Trans. Commun.*, vol. 36, no. 2, pp. 231–234, Feb. 1988.

[14] L. Badia, M. Levorato, and M. Zorzi, "Markov analysis of selective repeat type II hybrid ARQ using block codes," *IEEE Trans. Commun.*, vol. 56, no. 9, pp. 1434–1441, Sep. 2008.

[15] M. Rossi, L. Badia, and M. Zorzi, "Exact statistics of ARQ packet delivery delay over Markov channels with finite round-trip delay," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1858–1868, Jul. 2005.

[16] L. Huang and T. T. Lee, "Generalized Pollaczek-Khinchin formula for Markov channels," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3530–3540, Aug. 2013.