Performance Analysis of Stop-and-Wait ARQ for Wireless Channels

Stijn De Vuyst
Supervisor(s): Herwig Bruneel, Sabine Wittevrongel

Abstract—ARQ (Automatic Repeat reQuest) is used to ensure a reliable communication between transmitter and receiver over an error-prone channel. When the transmitter sends a data packet over the channel to the receiver, a transmission error may occur depending on the condition of the channel. The receiver checks the integrity of the packet and notifies the transmitter by returning a NACK in case of erroneous transmission or an ACK otherwise. If necessary, the packet is retransmitted a number of times until it is received correctly.

We present the analysis of the Stop-and-Wait ARQ protocol in case the channel produces errors in a bursty, correlated manner. Indeed, especially in the case of wireless communications with mobile transmitters and receivers, it is expected that the conditions of the wireless medium are not static but change over time. Our results show that the performance of the transmitter buffer rapidly deteriorates for higher error correlation.

Keywords—Discrete-time queueing, ARQ, wireless communication, error control

I. INTRODUCTION

WHENEVER packets of data need to be transmitted from point A (the transmitter) to point B (the receiver), there is always a chance that something bad happens to them while they move through the medium between A and B (the channel): some packets may be corrupted under the way or even lost entirely. To cope with this, various ARQ (Automatic Repeat reQuest) protocols have been proposed and implemented in order to provide a more reliable way of communication between transmitter and receiver.

Fig. 1. Operation of the transmitter queue under SW-ARQ, with fixed feedback delay $s$.

In this study, we focus on the performance of the so-called Stop-and-Wait ARQ protocol (SW-ARQ), illustrated in Fig. 1. In the usual setting of layered digital networks, one or more users situated in some higher network layer offer packets to the transmitter side with the specific request of transporting them to the receiver without errors. To provide this service, the transmitter requires some temporary storage facility or queue, not only because the packets ‘arrive’ randomly and therefore may have to wait before they can be sent into the channel, but also to allow the transmitter to retransmit packets at a later time if necessary. With regard to this transmitter queue, we provide explicit results for the equilibrium distribution

(i) of the number of packets in the queue, and
(ii) the delay experienced by an arbitrary packet.

Specifically, in case of SW-ARQ, the transmitter sends a packet available in its queue and then simply waits until it receives a feedback message for that packet. This message is either a negative (NACK) or a positive acknowledgement (ACK), depending on whether the receiver respectively detected an error in the transmitted packet or not. If an ACK is returned, then everything is ok and the next packet waiting in the queue is transmitted. Otherwise, if a NACK is returned, the same packet is retransmitted over the channel. Note that the transmitter remains inactive after sending a packet for a period of time which is called the feedback delay. This is the time required for the packet to travel through the channel, to be processed by the receiver and subsequently, for the feedback message to travel back. Compared to other ARQ schemes, SW-ARQ is simple to implement and ensures that packets are received in the same order as they arrived to the transmitter. Interestingly, our model distinguishes itself from previous studies in that we allow the errors occurring in the channel to be correlated in time. This complication is inspired by the observation that wireless communication channels rarely have the same error sensitivity during their whole time of operation. Factors such as user mobility, electromagnetic interference and channel fading are mostly time-varying in nature.

II. MODEL OF THE TRANSMITTER QUEUE

We propose and analyse a model of the transmitter queue under SW-ARQ in a discrete-time setting, i.e. we assume time to be divided in fixed-length slots whereby one slot is the time required to transmit one packet from the queue into the channel. The numbers of packets arriving to the system during slots $k = 0, 1, 2, \ldots$ are assumed to constitute a sequence $a_k$ of independent and identically distributed (iid) random variables with common probability generating function (pgf) $A(z)$. Instead of assuming that the probability of an erroneous packet is static in time, the channel alternates between two states which could be termed the GOOD (0) state and the BAD (1) state. These states reflect different conditions with regard to the error probability. The channel state process is modelled as a two-state Markov chain with a fixed error probabilities $e_i$ in either state $i = 0$ or $i = 1$, resulting in what is known as the Gilbert-Elliott (GE) model illustrated in Fig. 2, where usually $e_0 < e_1$.

III. ANALYSIS

The analysis of the transmitter queueing model relies on viewing the system as a (multidimensional) Markov chain and calculating its equilibrium distribution accordingly. Hence, a description of the system state in the Markovian sense is needed, i.e. a set of random variables such that their distribution in slot
k + 1 depends only on the values of the corresponding variables in slot k (and not on those in previous slots). For this purpose, we choose the set \( \{r_k, m_k, u_k\} \) illustrated in Fig. 3, which is to be interpreted as follows. First, \( u_k \) is the queue content (number of packets) at the beginning of slot \( k \). Next, we also need to know how far a packet has progressed through the channel during slot \( k \) and when we can expect its feedback message. Therefore, we define the residual roundtrip time \( m_k \) as the remaining number of slots at the beginning of slot \( k \) needed to complete the roundtrip of the most recently transmitted packet if \( u_k \geq 1 \), and for which \( m_k = 0 \) if \( u_k = 0 \). Finally, the random variable \( r_k \) indicates the channel state (0 or 1) during slot \( k \), which is a sequence according to the GE channel model. The value of \( r_k \) comes into play during slots with \( m_k = 1 \), where it determines the probability that either an ACK or a NACK is returned, or equivalently, that the packet departs from the queue or is to be retransmitted.

The transitions from slot \( k \) to slot \( k + 1 \) in this three-dimensional Markov Chain \( (r_k, m_k, u_k) \) are described by a set of system equations derived from the model. The remainder of the analysis is largely performed in the transformation domain and results in an expression for the joint pgf

\[
P(x, y, z) = \lim_{k \to \infty} E[x^r y^m z^u],
\]

of the system state distribution during an arbitrary slot in equilibrium. For details we refer to [2–4]. We note that the system will only reach stability for \( k \to \infty \) if the condition

\[
\lambda < \eta = (1 - \sigma) \frac{1 - e_0}{s + 1} + \sigma \frac{1 - e_1}{s + 1},
\]

is satisfied, where \( \lambda = E[u] = A'(1) \) is the mean arrival rate and \( \eta \) is the throughput of the SW-ARQ protocol over the GE channel, i.e. the maximum of packets per slot that can correctly be delivered to the receiver. The system load \( \rho \) then is \( \lambda/\eta \).

Obviously, the pgf \( U(z) \) of equilibrium queue content \( u \) follows from (1) as \( U(z) = P(1, 1, z) \). Also, as described in [3,4], the pgf \( D(z) \) of the packet delay \( d \) can be obtained as well, but requires a substantial additional effort. The moments of either distribution can be obtained simply by invoking the moment-generating property of pgfs. For example, \( E[u] = U'(1) = \lambda E[d] \) (Little’s law). The tail distributions Prob\( [u = n] \) and Prob\( [d = n] \) for large \( n \) are accurately obtained by using a dominant pole approximation in the asymptotic expansion of \( U(z) \) and \( D(z) \) respectively.

IV. DISCUSSION OF SOME RESULTS

In Fig. 4 the distribution of the delay \( d \) is shown in a specific situation and for increasing values of the correlation factor \( K = 1, 10, 20, 50, 100 \). Although the fraction BAD slots is the same for all curves, the delay increases drastically with \( K \).

The simple fact that both the BAD and the GOOD periods last longer, results in a deteriorated queueing performance that gets worse and worse as \( K \) further increases. In Fig. 5 we illustrate the precise circumstances in which such unbounded growth of the capacity requirements occurs. Recall that the system is stable (i.e. reaches equilibrium) if (2) is satisfied. Even so, it is useful to distinguish two different types stability regime, depending on the arrival rate \( \lambda \) compared to a threshold value \( \eta_1 = \frac{1 - e_0}{s + 1} < \eta \). As seen from (2), \( \eta_1 \) is the throughput of the system in the fictitious case that \( \sigma = 1, \) i.e. if the channel would always be in the BAD state. If \( \lambda < \eta_1 \), then the impact of increasing \( K \) saturates for \( K \to \infty \), which was called symmetric stability in [4]. If on the other hand \( \eta_1 < \lambda < \eta \), then the system is in a regime of compensated stability in which \( E[u] \) keeps growing with \( K \).

V. CONCLUSION

We have studied the queueing performance of the SW-ARQ protocol over a Two-state Markovian channel. Exact and explicit expressions are derived for the pgf, the moments and the approximated tail distribution of both the queue content \( u \) and the packet delay \( d \) during equilibrium. The profound impact that the channel correlation may have on the performance of the queue is illustrated.

REFERENCES


