17 GHz Wireless LAN: Performance Analysis of ARQ Based Error Control Schemes

Giuseppe Razzano\textsuperscript{1,2}, Luca Cecconi\textsuperscript{1}, and Roberto Cusani\textsuperscript{1}

\textsuperscript{1} INFOCOM Dpt., University of Rome “La Sapienza”, Italy
\textsuperscript{2} Telecommunication Research Center Vienna (fkw.), Austria
\{razzano, cusani\}@infocom.uniroma1.it

Abstract. The paper presents the development of ARQ schemes applied to a 17 GHz single-hop ad hoc network, providing very high bit rate. In particular, the paper aims to analyze and compare the performances of different ARQ protocols, in order to outline the most suitable one for a very high bit rate wireless system. Simulation trials show that a reliable error control protocol, realized by means of a proper retransmission strategy, can sensibly reduce the number of transmission errors, thus improving the wireless network overall performances.

1 Introduction

It is well known that wireless communications are exposed to high probability transmission errors. This is even more challenging when the aim is to develop a wireless LAN working at very high frequency (17 GHz) and with tight QoS requirements, in terms of error probability and transmission delay.

In this work we consider a novel wireless LAN developed in the framework of the WIND-FLEX (WF) project \cite{1}, funded by the European Information Society Technology (IST) program. We investigate in particular the performance of error control (EC) handling protocols at data link layer, to deal with the noisy radio channel. The EC protocol uses an Automatic Repeat reQuest (ARQ) that works on a per connection basis.

ARQ scheme is just an aspect of a wide strategy to improve the performance of the error-prone air interface. At physical layer, the system employs Forward Error Correction (FEC) scheme that improves the receiver capacity to detect and correct garbled bits \cite{2}. In this paper we deal with Protocol Data Units (PDUs) at the data link layer, after they have been processed and, when possible, corrected by means of the the FEC scheme.

For what concerns the upper layers, the interaction between TCP and data link layer ARQ over wireless links has been extensively studied in many papers in the past years, especially considering ARQ persistence consequences on TCP behavior. Different conclusions are drawn about this subject. According to some works (e.g. \cite{3}) not-fully persistent ARQ strategies should be employed at the data link layer, while other authors claim that a completely reliable ARQ scheme improves TCP performance (e.g. \cite{4}, \cite{5}). In this work, we do not analyze the
interaction between TCP and the data link layer ARQ. We adopt a non-fully persistent ARQ scheme, adapting the maximum number of retransmissions to the QoS requirements (especially in terms of maximum tolerable delay) of the applications. As showed in the simulation results, this approach leaves to the TCP layer a small residual packet loss percentage, when, after some retransmission, the packet cannot still be correctly delivered within a certain delay threshold.

The rest of the paper is organized as follows: Section 2 presents the ARQ protocols, while Section 3 describes the wireless channel model. Simulation results are reported in Section 4. Finally some conclusions are drawn in Section 5.

2 ARQ Protocols

As already mentioned, the analysis of the paper focuses on three ARQ protocols, namely Stop-and-Wait (SW), Go-Back-N (GBN), and Selective-Repeat (SR). The following section provides a short description of the algorithms, pointing out the way they have been adapted to the WF system.

2.1 Stop and Wait (SW)

SW is the simplest ARQ scheme: the sender waits for a positive acknowledgment (ACK), or a negative one (NACK) from the receiver, after every packet transmission. Depending on the reply, the sender either transmits the next packet (ACK reception), or retransmits the last one (NACK reception). If the packet is lost or the reply is lost or corrupted, when a timer expires, the sender retransmits the not acknowledged packet. Using SW scheme no buffering for transmitted packets is required (both transmission and reception sliding windows are unitary).

The simplicity makes the SW algorithm attractive in many situations, but given its scarce utilization of the available bandwidth, the algorithm is not suitable for a network characterized by very high bit rate and thus expected to support high traffic loads. As simulation results confirm (see Section 4), the network performances are definitely scarce, when either traffic load or Packet Error Rate (PER) increase.

2.2 Go Back N (GBN)

When GBN protocol is used, the transmitter sends several packets consecutively. Transmitted packets are stored in a retransmission buffer, until they are positively acknowledged (individually or cumulatively). The number of packets, sent consecutively without acknowledgment, cannot exceed a transmission window, whose length is one of the algorithm parameters and mainly depends on the available HW resources. The receiver instead does not use any buffer, packets are acknowledged as soon as they are received.

If a gap in the reception sequence is detected, meaning that a packet is lost, the receiver suspends accepting packets and sends a NACK to request a
retransmission of the missing one. The sender receiving a NACK restarts transmission from the missing packet and proceeds in sequence. The N packets that were transmitted after the corrupted (or missing) one are discarded even if they reached the receiver without errors.

Contrary to SW algorithm, GBN does not force the transmitter to remain inactive waiting for ACK/NACK messages after every packet transmission, therefore it is able to achieve a greater efficiency. Nevertheless, whenever a packet is lost or gets damaged, up to N redundant retransmissions are made, which results in a considerable resource waste.

2.3 Selective Repeat

When SR schemes are employed only corrupted or lost packets are retransmitted. In particular the algorithm considered for the WF system is Selective-Repeat-with-Partial-Bitmap (SRPB), which makes use of an optimized bit mask field in the acknowledgment messages, in order to reduce the amount of overhead [7]. The protocol works as follows. The transmitter sends a series of PDUs, sequentially numbered within the predefined window, making full use of the available bandwidth. Transmitted PDUs are stored until they are not acknowledged; for each PDU a timer is set. Also at the receiver side, a window is used to buffer correctly received packets. For each packet of the window, the status of the packet (positively received or not) is stored.

The acknowledgment is done sending a packet, whose format is reported in Fig. 1. The TYPE field is used to distinguish among the three kinds of packets used: data packets, acknowledgment packets, and data plus acknowledgment packets (which allow the transmission of ACK message in a piggybacking fashion). The CONNECTION ID field carries the connection number to which the ACK message is related, i.e. the connection used to send data packets that are being acknowledged. The Flow Control (FC) field is set to 1 if the receiver window is full, in order to suspend the transmission of other data packets, until a new communication (FC bit set to 0) is received by the transmitter. We employed transmission and reception sliding windows, whose length is $W_s=512$, thus the sequence numbers supported are at least twice the buffer size, representing numbers from 0 to 1023. Data packet acknowledgment is done considering

![Fig. 1. SRPB acknowledgement packet format](image-url)
blocks of packets and then specifying packets (PDUs), belonging to the block, which have or have not been received correctly. In particular, as reported in Fig. 1, we decided to use ACK messages containing up to 3 bitmap blocks, where each block (BM), identified by a 7-bit bitmap block number (BMN), is an 8-bit bitmask. Every packet belonging to a block is acknowledged with a bit set to 1 (correct packet) or 0 (corrupted or missing packet). For every ACK, the first 0 in the bitmask (corresponding to the first corrupted packet of the acknowledged blocks) suspends the progress of both the receiver and transmitter sliding windows, forcing the sender to retransmit the requested packet.

To construct blocks for the ACK message, groups of 8 packets to be acknowledged are considered. In the BLOCKNUM field it is reported the current number of bitmap blocks (BMs). When the last BM is used to acknowledge less than 8 packets, the MASKLEN field indicates the effective length of the block, thus allowing to dynamically allocate the necessary bitmap blocks, in order to reduce protocol overhead. The Cumulative ACKnowledgment (CACK) field is used to grant multiple packets, whose sequence number is lower than the one of the first packet in the first bitmap block. Finally the CRC field is used to detect transmission errors. Obviously, in presence of feedback errors, such as the lost of an ACK message, all packets related to the ACK blocks are retransmitted, after the expiration of the timers associated to the packets at transmitter side.

3 Channel Model

For modelling the error characteristics of a wireless channel between two stations, is widely used model the "Gilbert-Elliot" model: a two state Markov chain, where the states (Good and Bad) represent the possible behavior of the radio link. According to [8], assuming a flat fading channel and high data rates, such that the duration of a data packet ($\tau$) is smaller than the coherence time of the channel ($f_D$), it is possible to consider as analytical channel model, a Gaussian random process with a given mean and the following covariance function:

$$K(\tau) = J_0(2 \cdot f_D \cdot \tau)$$  \hspace{1cm} (1)

The covariance properties depend only on $f_D \cdot |\tau|$. When this quantity is small (i.e. $< 0.1$) the process is very correlated ("slow" fading). On the contrary, for larger values (i.e. $> 0.2$), two samples of the channel are almost independent ("fast" fading). Note that, for high data rate (small $\tau$), the fading process can always be considered to be slowly varying. Therefore, the dependence between the transmission of consecutive data packets cannot be neglected, and the model for the success/failure process has to take into account this dependence. A general success/failure process model considers samples of the fading process:

$$\alpha_n = (\alpha_1, \alpha_2, ..., \alpha_n), \hspace{1cm} \alpha_i = \alpha(iT)$$  \hspace{1cm} (2)

From a communication point of view, dealing with data link protocols, the aim is to evaluate the binary random process that describes the successes and the
failures of the packet transmissions: $\beta(t) = \phi(\alpha(t))$ (assumed, for simplicity, memoryless). The success or failure of a packet is determined by comparing the signal power to a certain threshold (i.e. the case of power under this threshold stands for a packet failure). For highly correlated fading, it is possible to extend the (approximate) Markovian character of the fading process to the success/failure process. What is now left to verify, adopting a first-order Markov model, is that the success/failure of the transmission in the previous slot summarizes almost all the information contained in the past. The verification is based on [9] and [8], where is considered the average mutual information between the success/failure process $\beta_i$ and the past two transmissions $\beta_{i-1}, \beta_{i-2}$. A measure of the goodness of the first-order Markov approximation can be given in terms of the negligibility of the additional information on $\beta_i$ carried by $\beta_{i-2}$ when $\beta_{i-1}$ is known. For slow fading, this can be demonstrated, validating the first-order Markov approximation to be adequate for packet success/failure process on a fading mobile radio channel.

Given the coherence time $\Delta T_c$ (1 ms) and the high data rates (up to 160 Mb/s) of the WF wireless network, it is possible to confirm the adequacy of the Gilbert-Elliot channel model in representing the considered radio channel. In fact, the bit rate and packet lengths, used in simulation trials (see Section 4), lead to a packet duration in the range $[3.33 \mu s, 50 \mu s]$, which is surely smaller than the coherence time of the channel (1 ms). Moreover, given the coherence time, according to [10], Doppler frequency can be expressed as:

$$f_D = \frac{9}{16\pi \Delta t_c} \Rightarrow f_D \approx 179 Hz$$

(3)

Therefore the product of the Doppler frequency and the packet duration approximately belongs to the following range:

$$f_D \cdot \tau \in [5.96 \cdot 10^{-4}, 8.95 \cdot 10^{-3}]$$

(4)

Being $\tau < \Delta T_c$ and $f_D|\tau| << 0.1$, it is possible to conclude that the Gilbert-Elliot channel is adequate to characterize the behavior of the considered wireless channel.

4 Performance Analysis

The WF network has been simulated via software, using OPNET Modeler 9.0, and several trials have been carried out with different network scenarios and traffic conditions. The presented results refer to a scenario with five (fixed) devices in a 20x20 single-hop cluster. Three classes of service have been considered as representative of WLAN applications: Streaming class (used for audio and video streaming applications), Interactive class (used for web browsing applications), Background class (used for e-mail or FTP applications). The ARQ protocols are applied only on Interactive and Background classes. ARQ algorithms are not applied to Streaming class, whose low delay requirements make ineffective
Table 1. Traffic sources Parameters

<table>
<thead>
<tr>
<th>Class of Service</th>
<th>QoS Requirements</th>
<th>Simulation Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Maximum Delay</td>
<td>ON State</td>
</tr>
<tr>
<td>Background</td>
<td>0.33 s</td>
<td>Exp(2s)</td>
</tr>
<tr>
<td>Interactive</td>
<td>0.125 s</td>
<td>Exp(3s)</td>
</tr>
<tr>
<td>Streaming</td>
<td>0.00275 s</td>
<td>Exp(2s)</td>
</tr>
</tbody>
</table>

Table 2. Channel parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Good State BER</td>
<td>$e_G = 0$</td>
</tr>
<tr>
<td>Bad State BER</td>
<td>$e_B = 10^{-4} : 10^{-3}$</td>
</tr>
<tr>
<td>Transition State Probability Good-Bad</td>
<td>$P_{GB} = 0.005$</td>
</tr>
<tr>
<td>Transition State Probability Bad-Good</td>
<td>$P_{GB} = 0.04$</td>
</tr>
</tbody>
</table>

a retransmission strategy for such applications. Table 1 describes the main parameters of the simulated traffic sources, and the maximum acceptable delay for the three classes of service. As already explained, we assume that the forward channel is a random-error channel, represented as a Gilbert-Elliot model. The channel status is defined by the BERs of the two states (good and bad) and the transition probability matrix, set as reported in Table 2. The feedback channel is assumed to be an ideal error free link. The considered values for $e_B$, together with the adopted packet size (450 bits), lead to a PER range of 1-10%. Obviously, the PER depends on the relationship between the channel error process and the packet size (longer packets are more likely to be hit by an error). Assuming that the CRC code error detection probability is ideal, a packet is considered garbled when at least one bit is hit by error.

Dealing with a very high bit rate WLAN, our interest was mainly focused on finding the most suitable protocol, to such a system. Out of the parameter considered to evaluate the respect of the QoS requirements, one of the most significant is definitely the maximum delay experienced by the packets, belonging to the three classes of services. As shown in Fig. 2-left, with respect to Background and Interactive classes, SW has a very high percentage (from about 75% to about 90%) of packets discarded due to high delay, mainly caused by the way retransmissions are handled by the protocol. This percentage increases with the traffic load and with PER (although the dependence with PER is weaker). The statistic values are not reported for GBN and SRPB scheme, because the percentage of packets not satisfying the QoS requirements is very low for both the protocols (less than 0.9% for GBN and less than 0.25% for SRPB), for every traffic situation and packet length. The opportunity to retransmit a garbled packet depends on its time to live (TTL), that is the time left which still enables the QoS requirements satisfaction (see Table 1). If a packet has not been received correctly, and its TTL is elapsed, the packet is discarded by the transmitter. The statistic of discarded packets, that cannot be “corrected” by means of the ARQ error control scheme, is shown in Fig. 2-right. As expected, SW is the worst
protocol in recovering the garbled packets. GBN has a very low percentage of unrecoverable packets (from 0% to 3%), while the best results are obtained using SRPB (not reported in the picture), with a not-recovery percentage always lower than 0.002%.

Concerning GBN, it is interesting to show how many packets correctly received, are discarded due to protocol mechanism, which is based on an unitary receiver window. Fig. 3-left reports such percentage. This number grows, with the increasing of traffic load or of PER, reaching very high values and leading to a considerable resource wasting.

When analyzing the performances of SRPB protocol, it is important to evaluate the resequencing delay, which is the time that correctly-received packets spend in the receiver buffer, waiting for corrupted or missing packets to be received correctly. The resequencing delay has to be considered for SRPB protocol, which is the only protocol allowing the reception of out of sequence packets. As can be seen from Fig. 3-right, the receiver buffering time is, in great percentage, concentrated below 10 ms, with a reduction of this percentage when traffic increases, and a very little growth of the percentages related to higher waiting intervals. The resequencing delay is the acceptable price to pay, for SRPB, which avoids retransmission of correctly received out-of-order packets.
5 Conclusions

In this paper we investigated the performances of an EC protocol at DLL, for a very high speed wireless LAN. The EC protocol has been implemented choosing three ARQ schemes: Stop and Wait, Go Back-N and Selective Repeat with Partial Bitmap. We analyzed the performances under different traffic loads, different packet lengths and error rates. On one hand, SW and GBN have outlined great inefficiencies: with regard to the respect of QoS parameters, unrecoverable errors and bandwidth utilization (mainly SW) and overhead, energy waste and bandwidth waste due to useless retransmissions (mainly GBN). On the other hand, SRPB ensured: high efficiency, low overhead, high QoS parameter respect and very low percentage of unrecoverable errors. In particular, the overcoming of Selective Repeat ARQ schemes on the other two protocols, in such a network, comes from considering its Partial Bitmap version. The innovative acknowledging mode, presented in the paper, enables to grant blocks of packets and to dynamically allocate the size of the ACK packet, thus enabling to obtain all the above listed advantages at a reasonable increase of the computational cost.

References