This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Context-aware Receiver-driven Retransmission Control in Wireless Local Area Networks

Dzmitry Kliazovich, Nadhir Ben Halima, and Fabrizio Granelli
DISI – University of Trento
Via Sommarive 14, I-38050 Trento, Italy
E-mail: [kliazovich,nadhir,granelli]@disi.unitn.it

Abstract—**Automatic Repeat reQuest (ARQ) techniques employed by leading wireless technologies aim at compensating the high error rates due to radio impairments, but do not offer any differentiated levels of protection. In this work, we propose to enable per-packet differentiation of link layer ARQ protection (in terms of no. of retransmissions) driven by requirements of the end applications as well as of communication protocols implemented on the mobile terminal. Experimental results demonstrate the potential benefits deriving from the proposed strategy, both on TCP data flows and MPEG-4 video streams.**

Keywords-retransmission, ARQ, distributed protocol stacks

I. INTRODUCTION

Wireless technologies represent a networking sector growing at unprecedented rate. They enable fast and cost-effective network organization, deployment and maintenance and allow users to easily join, leave or switch to other networks in a simple and possibly transparent way.

Nowadays wireless technologies are mostly considered in the last mile – connecting end-users to the core of the network, while leaving transport of data in the core to wired architectures. As a result, characteristics of wireless links often determine the performance of entire system in terms of capacity and time-varying characteristics having a relevant impact on the user experience.

One of the crucial limitations of wireless networks is related to the high Bit Error Rate (BER) on the radio link, considerably reducing the performance of the widely used Transmission Control Protocol (TCP) [5].

In order to counteract high BERs, most of the wireless network technologies employ Automatic Repeat reQuest (ARQ) protocols at the link layer [4]. Such ARQ protocols are commonly based on a stop-and-wait error control strategy, requiring the sender to wait for a positive acknowledgement for the transmitted frame before continuing with the transmission of the subsequent frame. Lack of positive acknowledgement triggers retransmission.

Such choice of ARQ protocols is mainly motivated by their advantages over Forward Error Correction (FEC) techniques in terms of flexibility and low bandwidth overhead, i.e. ARQ protocols consume bandwidth only for retransmission of erroneous frames, while FEC schemes are typically tuned to the worst case introducing constant bandwidth overhead – also in case of correctly transmitted frames.

The performance of stop-and-wait ARQ schemes is determined by *retry limit* parameter, which specifies the maximum number of retransmission attempts taken for a single packet delivery. If the retry limit is reached, the frame is discarded.

In most of wireless network technologies like WiFi, WiMAX, or cellular networks, the *retry limit* parameter is fixed to a default value, computed to provide a certain radio link BER improvement tuned to an average case (e.g. to compensate typical levels of signal fading and interference for predefined packet sizes).

To counteract the resulting non-optimal performance deriving from such approach, researchers proposed several techniques [1-3] for making ARQ protocols adaptive to the radio link conditions and type of modulation employed. This is achieved by increasing ARQ strength for noisy channels and decreasing it for errorless channels. Such solutions are able to keep the BER provided to upper layers (i.e. above the link layer) of the protocol stack stable.

Summarizing, the use of ARQ protocols, in most of the wireless technologies deployed nowadays as well as in the research proposals available in the literature, aims at compensating radio link bit errors up to the levels in the range of 10^{-6} – 10^{-8} required for enabling a reasonable performance of the TCP/IP protocol stack.

In this work, we argue that further performance improvement can be achieved by tuning ARQ strength based on the application requirements and protocol stack operation on the mobile terminal. Specifically, in the framework of an IEEE 802.11 network, the proposed method allows the mobile terminal to control ARQ strength through specifying the retry limit on a per-packet basis by using a feedback channel. This feedback channel is transparently encapsulated into IEEE 802.11 MAC protocol – avoiding the requirement for modifications of the standard.

The rest of the paper is organized as follows: Section II provides a detailed description on the proposed approach; Section III presents performance evaluation results for real-time multimedia and TCP-based data transfer scenarios; Section IV concludes the paper final remarks.

II. Proposed Approach

The proposed Context-aware Receiver-driven Retransmission Link-Layer Control in Wireless Local Area Networks (CORREC) approach enables the mobile receiver to tune the *retry limit* parameter used at the link layer of the Base Station (BS) for the transmission of the next frame.

In IEEE 802.11 WLAN standard, each data frame transmitted at the link layer should be positively acknowledged by the receiver. The acknowledgement frame (presented in Fig. 1) has a reserved portion of 12 bits, which provides a suitable place for organizing a feedback channel between the mobile node and the base station without introducing any additional overhead and avoiding modifications of the MAC protocol.

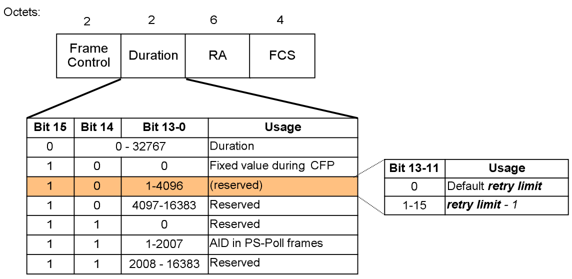

Figure 1. IEEE 802.11 ACK frame.

In fact, only 4 bits are required for the specification of maximum retry limit equal to 14, while the value equal to 0 is used to signal the BS to use the default value for the retry limit.

A. Packet Importance Metric

For the purpose of the paper, we extend the definition of packet importance, *Imp*, given in [8] for VoIP flows to the following:

"The importance of a given packet corresponds to the level of quality reduction for a given flow in case this packet is lost during transmission or corrupted at the receiver."

The notion of flow quality depends on end-to-end application requirements. For example, for VoIP flows, the quality can correspond to Mean Opinion Score (MOS) metric of the flow, for video flows it is commonly represented by Peak Signal-to-Noise Radio (PSNR), and for file transfer applications the quality may simply correspond to the average throughput achieved during the transfer.

B. File Transfer Applications

Most of the data flows transferred on the Internet are TCP-based. Currently, all the packets produced by TCP layer (connection setup packets, data segments, and acknowledgements) are treated by the link layer equally, i.e. with the same level of error protection (or same value of *retry limit*).

The proposed CORREC approach assigns packets different levels of importance with an algorithm that captures TCP semantics, leading to significant performance improvement.

Fig. 2 presents congestion window evolution in TCP New Reno and the corresponding proposed variation of the packet importance metric. Specifically, the proposed approach assigns the highest importance (*High Imp*) to TCP segments produced right after each window reduction and decreases it down to the *Low Imp* threshold following linear or any other function.

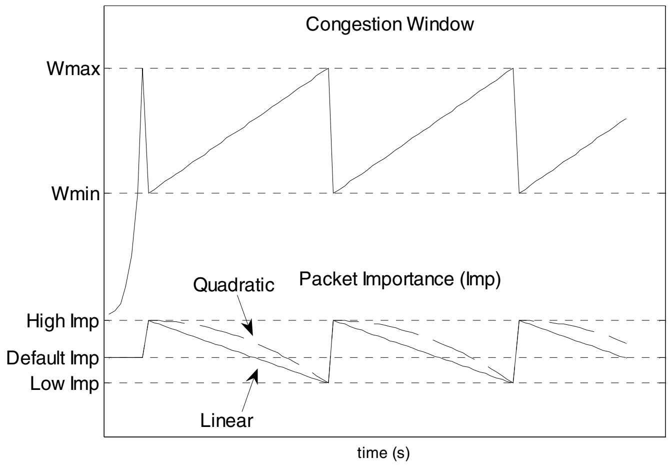

Figure 2. Packet importance metric for TCP-based data flow.

Each value of the *Imp* parameter has a direct correspondence to the *retry limit* parameter configured at the base station for packet transmission at the link layer. In this paper, we use *retry limit = 7* for packets with *High Imp* and *retry limit = 0* for packets with *Low Imp*.

The main idea behind the proposed approach is to provide higher protection on the radio link (and more retransmission attempts) when congestion window is small and lower protection for high window values. Indeed, when the congestion window is small, any link error will trigger window reduction to its half – unnecessarily reducing the throughput of the TCP flow. In the opposite case, the impact of the link error becomes less significant, since the window will be possibly reduced due to congestion-related losses.

C. Multimedia Applications

CORREC has several applications, ranging from optimizing TCP-based traffic, to VoIP and real-time multimedia traffic. In order to evaluate our scheme for the multimedia traffic case, we selected MPEG-4 video flows which represent the current coding and transport standard for video delivery over the Internet.

In brief, a MPEG-4 video is composed of Groups of Pictures (GOPs). Each GOP includes video frames of three types. The I-Frames (Intra coded frames) are encoded without reference to any other frame in the sequence, and are inserted every 12 to 15 frames as well as at the beginning of a sequence. Video decoding can start only at an I-frame. P-Frames (Predicted frames) are encoded as differences from the last I- or P-frame. The new P-frame is first predicted on the basis of the reference I- or P-frame through motion compensation and then the prediction error is encoded. B-Frames (Bidirectional frames) are encoded as the difference from the previous or following I- or P-frames. B-frames use prediction as for P-frames but for each block either the previous or the following I- or P-frame is used.

Due to the correlation property of P- and B-frames, the effective impact deriving from the loss of an I-frame is much higher than that of P- or B-frame. In addition, the loss of one packet may generate error propagation. While the loss of a B-frame does not affect the quality of the consecutive frames, the loss of an I-frame may disable correct decoding of subsequent P and B frames. This leads to the conclusion that I-frames are more important than P-frames, which are more important than B-frames.

As in wireless LANs a relevant packet loss is expected, we propose that the portion of the multimedia stream which is crucial to the overall quality is to be retransmitted with a higher *retry limit*, i.e. packets belonging to an I-frame are retransmitted with retry limit = 7, while packets belonging to a P-frame and B-frames are retransmitted with a retry limit equal to 4 and 2, respectively.

Similar reasoning is clearly applicable for H.263 and H.264 encoded video streams.

III. Performance Evaluation

A. Scenario Definition

To test the CORREC approach, NS-2 (version 2.31) network simulator [9] is used. Figure 3 illustrates the considered simulation wireless-cum-wired scenario, where a Mobile Node (MN) is receiving a data flow from a server (the Fixed Host, FH) located on the fixed Internet. The IEEE 802.11b specification parameters, with 11 Mbps physical data rate, characterize the wireless link between the Base Station (BS) and the MN. The FH is connected to the Base Station (BS) using a link providing a data rate of 10 Mbps and one-way delay of 200 ms. The bottleneck buffer located at the BS is of FIFO (First-In-First-Out) and chosen to be limited to 700 packets.

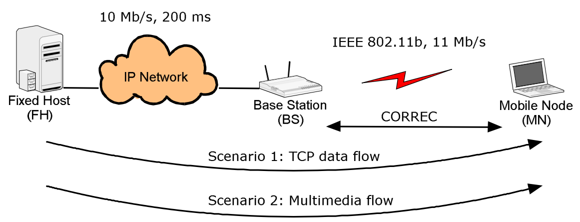

Figure 3. The simulation scenario.

In the first scenario, an FTP connection (over TCP protocol) is established between the MN and the FH. TCP/IP datagram size is 1500 bytes which corresponds to the most commonly used Ethernet MTU.

In this second scenario, the FH is a video server transmitting video streams, while the video receiver located at MN is connected using the wireless link. Several video traces are used in the experiments. However results are presented in the paper only for the "Foreman" video sequence, using MPEG-4 video coding. Video format is Quarter Common Intermediate Format (QCIF, 176 x 144). GOP structure is IBBPBBPBBPBBPB. The video trace is composed of 300 frames (10 I-frames, 102 P-frames and 188 B frames) further divided into IP packets. For our simulation, the employed integrated simulation tool consists of a MPEG-4 encoder, a video sender (VS), Network Simulator 2 (ns-2), a MPEG-4 decoder, an evaluate trace (ET) program and a PSNR calculation program. The integrated environment methodology was proposed and developed within the framework of EvalVid [10], enhanced as in [11] for including into NS-2.

The video clip, stored in YUV format, is introduced into an MPEG-4 encoder, which generates an encoded video stream. The open-source ffmpeg MPEG-4 encoder and decoder [12] are used for our experiments. The encoded video stream is read by the Video Sender (VS) for generating a trace file, which contains information such as frame type, size, etc. for each video frame. The trace file is then introduced to the streaming server in the ns-2 simulator to produce video streams in the network simulation platform (ns-2). The effect of streaming video over the network is captured in a streaming client log file, which is generated by NS-2. The log file contains information such as timestamp, size and ID of each packet, both for the client and the streaming server. The trace file and the log files are used by the Evaluate Trace (ET) program to generate the corresponding corrupted video files as a result of transmission over the considered network. The corrupted video file is then processed by the Peak Signal to Noise Ratio (PSNR) calculation module to evaluate end-to-end video quality.

Average throughput of TCP connection is chosen as the main metric for evaluation of the proposed scheme in application data transfer scenario, while PSNR and the achieved goodput are the metrics for the second scenario, to evaluate the advantages enabled by the proposed approach if applied for multimedia flows.

B. Results: TCP flows

In this scenario, a TCP data flow is considered. Prediction of the congestion window (*cwnd*) of the server is performed by simply counting the number of TCP packets successfully received between two consequent events of error detection at the receiver (with three duplicate ACKs). This measure provides the length of the linear increase phase between two window reductions. Dividing this length by the difference between the maximum and the minimum value of the *retry limit*, a packet interval is obtained, after which the retry limit is decreased by 1 from retry limit max = 7 to retry limit min = 0.

Fig. 4 provides a comparison between the achieved TCP throughput using CORREC approach and the legacy scenario for variable link error rates. The main advantages come for PERs higher than 10^{-4} due to non-congestion related reduction of window size and unnecessary degradation of TCP throughput.

C. Results: Multimedia traffic

In the second scenario, a MPEG-4 video flow is transmitted on the topology presented in Fig. 3. The GOP structure is given by n (the repetition rate of I-frame in data sequence) and m (the repetition of anchor frames – I- or P-frames). For instance, a GOP structure of IBBPBBPBBPBBPBB has n = 15 and m = 3.

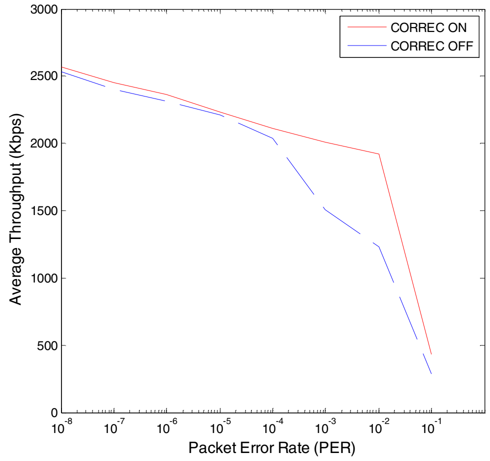

Figure 4. Performance of CORREC in terms of TCP throughput against packet error rate.

Fig. 5 compares the achieved PSNR against wireless link errors in 3 different cases:

1) with no link layer retransmissions performed (corresponds to with the retry limit equal to 0);

2) with retry limit constantly set to be equal to 3 for all the outgoing packets; and

3) with dynamic retry limit varied according to the proposed CORREC approach.

The results demonstrate significant performance improvement for the latter case (i.e. when CORREC is used). The quality of video flow remains stable for PERs of up to 10^{-2}.

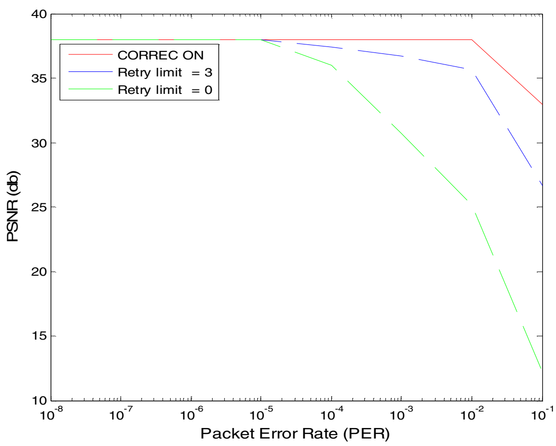

Figure 5. PSNR against wireless link errors in case of static settings and with CORREC enabled.

Fig. 6 shows superposed bars which are presented to underline the difference between packets successfully received by MN's decoder using standard scheme (with retry limit equal to 3) and CORREC. It also underlines that the gain in video quality depends on the selective protection of I- and P-frames. The ratio between the packets sent and packets received is highlighted in Fig. 7.

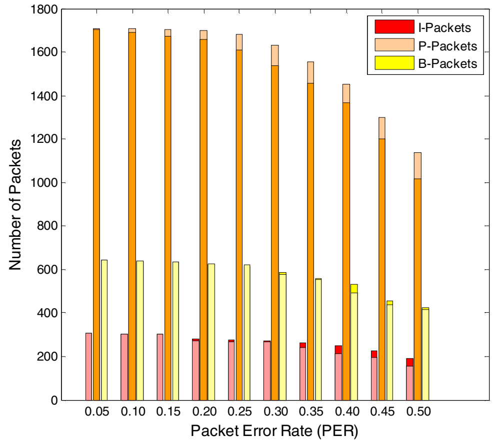

Figure 6. Number of I-, P- and B- packets correctly received against PER on the wireless link.

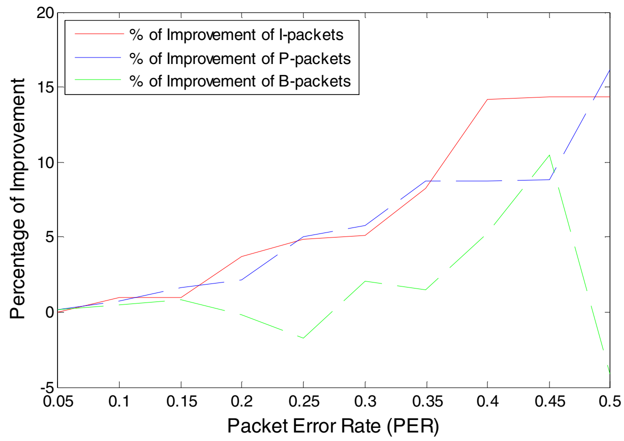

Figure 7. Level of improvement in terms of the ratio between packets sent and packets received of the same type (I, P, and B).

Fig. 8 shows sample snapshots of the "Foreman" video for PER = 0.15, comparing the visual quality in case *retry limit* = 3 and when CORREC is employed. The resulting PSNR is equal to 21,88 db for standard scheme and 33,15 db for CORREC.

IV. CONCLUSIONS

In this work, a flexible and dynamic per-packet differentiation of link layer ARQ protection driven by end application requirements is proposed. The method is applicable to data, voice and video flows. Experimental results demonstrate the potential benefits deriving from the proposed strategy.

REFERENCES

[1] Y. -D. Yao, "An effective go-back-N ARQ scheme for variable-error-rate channels," IEEE Trans. Commun., vol. 43, no. 1, pp. 20-23, Jan. 1995.

[2] H. Minn, M. Zeng, and V. K. Bhargava, "On ARQ scheme with adaptive error control," IEEE Transactions on Vehicular Technology, vol. 50, no. 6, pp. 1426 – 1436, Nov. 2001.

[3] Kurtis J. Guth and Tri T. Ha, "An adaptive stop-and-wait ARQ strategy for mobile data communications," Computer Communications archive, vol. 14, no. 7, pp. 405 – 412, 1991.

[4] D. Kliazovich, F. Granelli, and M. Gerla, "Performance Improvement in Wireless Networks using Cross-layer ARQ," Computer Networks, 2007, vol. 51, pp. 4396 - 4411.

[5] X. Chen, H. Zhai, J. Wang, and Y. Fang, "A Survey on Improving TCP Performance over Wireless Networks," in: Resource Management in Wireless Networking (Springer Network Theory and Applications Series, Vol. 16), edited by M. Cardei, I. Cardei and D.-Z. Du, pp. 657-695, Kluwer Academic Publishers/Springer, 2005.

[6] P. Bucciol, E. Masala, E. Filippi, and J. C. De Martin, “Cross-Layer Perceptual ARQ for Video Communications over 802.11e Wireless Networks,” Hindawi Publishing Corporation, Advances in Multimedia, 2007.

[7] Yufeng Shan, “Cross-Layer Techniques for Adaptive Video Streaming overWireless Networks,” EURASIP Journal on Applied Signal Processing, 2005.

[8] C. Hoene, B. Rathke, and A. Wolisz, "On the Importance of a VoIP Packet", In Proc. of ISCA Tutorial and Research Workshop on th Auditory Quality of Systems, Herne, Germany, April 2003.

[9] NS-2 simulator tool home page, http://www.isi.edu/nsnam/ns/, 2000.

[10] J. Klaue, B. Rathke and A. Wolisz, “EvalVid – A Framework for Video Transmission and Quality Evaluation”, Proceedings of the 13th International Conference on Modeling, Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, 2003. http://www.tkn.tu-berlin.de/research/evalvid

[11] Chih-Heng Ke, Cheng-Han Lin, Ce-Kuen Shieh, Wen-Shyang Hwang, “A Novel Re-alistic Simulation Tool for Video Transmission over Wireless Network,” The IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Com-puting (SUTC2006), June 5-7, 2006, Taichung, Taiwan. Code available in: http://140.116.72.80/~smallko/ns2/Evalvid_in_NS2.htm

[12] ffmpeg multimedia system, http://ffmpeg.mplayerhq.hu/

CORREC OFF (PER = 0.15, Average PSNR = 21.88db)

CORREC ON (PER = 0.15, Average PSNR= 33.15 db)

Figure 8. Visual comparison of achieved performance using the legacy scheme (upper) and the proposed scheme (CORREC, lower).