Cross-Layer Adaptive ARQ for Uplink Video Streaming in Tandem Wireless/Wireline Networks*

Antonios Argyriou
Philips Research
Eindhoven, The Netherlands
Email: antonios.argyriou@philips.com

***Abstract*— In this paper, we focus on improving the robustness of packetized multimedia streaming in tandem-connected wireless LANs and wireline packet switched networks. To this aim we initially develop an analytical model that expresses the end-to-end packet loss rate and latency, as a function of the retransmission-based error control mechanisms employed both at the application and wireless link layers. The developed model is the basis of an algorithm that dynamically identifies the optimal number of retransmissions at each protocol layer, so that the overall effective packet loss rate is minimized. Realistic video streaming experiments show considerable quality improvements in terms od PSNR, by avoiding the overall number of retransmissions.**

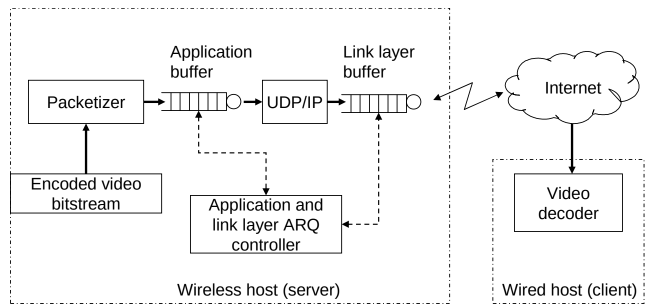

Fig. 1. The proposed architecture for multimedia streaming with cross-layer ARQ control.

I. Introduction

Wireless local area networks (WLANs) such as IEEE $802.11a, b, g$, are used extensively for easy and low-cost Internet access. The typical scenario that is fairly common nowadays, includes a tandem connection of a WLAN link and a wireline IP-based backbone. In this configuration, the unreliable transmission channel of the wireless link, and the CSMA/CA access protocol used in the WLAN, can cause considerable packet loss and high delay that can degrade application performance. Even with new MAC layer enhancements (e.g. 802.11e), packet losses that are caused by wireless errors are unavoidable.

For wireless multimedia streaming systems, robustness to bit errors and packet losses, is critical for the system performance due to the nature of the encoded video [1]. Error control methods generally include error resilient source coding, channel coding with forward error correction (FEC) or automatic repeat request (ARQ), and error concealment. Application layer ARQ, has been studied for video applications in a wide range of error conditions. However, recently there has been interest in using cross-layer information from the protocol stack. For example in the work reported in [2], the authors propose an algorithm for adjusting retransmissions at the link layer so that unequal error protection can be employed for scalable encoded video. In other works that also follow a cross-layer philosophy, it is suggested the use of both application layer ARQ and FEC, for achieving optimal error control for streaming encoded video [3], and for real-time encoding [4].

While in the previous works cross-layer ARQ and FEC systems have been demonstrated as an effective way to combat packet erasures, thorough analyses of the interactions between the different error control strategies are limited. In this paper we consider error control that is implemented through retransmissions (ARQ), both at the application and link layers. The proposed algorithm, supported by an analytical model, dynamically selects the optimal number of retransmissions both at the application and wireless link layers, so that the overall packet loss rate is minimized.

* This work was performed before the author joined Philips Research.

II. Multimedia Streaming System

Fig. 1 depicts the main components of a wireless multimedia streaming system engineered to work with the proposed cross-layer protection framework. The system consists of a media source, which can be either a real-time encoder or a pre-compressed media file. Source packets are packetized with the real-time transport protocol (RTP) [5], and are placed in the application layer transmission buffer. A copy of each packet is also stored in the retransmission buffer in case it needs to be retransmitted. Each packet is assigned maximum number of application layer retransmissions N_a^m. When a retransmission request arrives through a negative acknowledgment (NACK), the requested packet that is stored in the retransmission buffer, has priority over the normal transmission buffer, and is transmitted as soon as the network allows it. Then, a UDP/IP header is added by the network layer. This packet is finally sent to the 802.11 MAC layer which appends a header, and creates an MAC protocol data unit (MPDU), i.e. frame, for wireless transmission. Each frame can be retransmitted up to N_w^m times by the 802.11 link layer. Once the packets are correctly transmitted in the physical layer, they are processed at the access point in similar way, and they either arrive at the client after a specific delay, or they are lost in the wireline channel.

For the aforementioned streaming system, our goal is to express analytically, the overall effective packet loss rate, and the overall latency for each transmitted source packet. Based on the developed model, we will develop an algorithm that calculates the optimal level of protection by adjusting N_a^m, N_w^m.

III. Source and Channel Models

1) Description of a Generic Media Source: Since the proposed framework is media-aware but not media-type specific, we have to formalize the fundamental properties of the payload carried by media packets. A packet l is considered to be available for transmission at time $t_{l,a}$, it is transmitted at time $t_{l,s}$, while its prescribed playback deadline is $t_{l,d}$. We assume that the source generates data packets of size S_l at at a rate of R_s packets per second. Therefore, the maximum allowable delay for this source packet is denoted as $\tau_l = t_{l,d} - t_{l,a}$.

2) Hybrid Wireless/Wireline Network Packet Loss Model: In this paper, we define a joint model between the wireless and the wireline networks where each of them is modeled by a two-state Markov chain. Each of the two Markov chains, is characterized by a good (S_0, U_0), and a bad state (S_1, U_1). The wireless link of the end-to-end connection path is essentially modeled as a two-state discrete Markov chain known as Gilbert model [6]. We must note here that we also consider MAC frame erasures in the wireless network, since if the errors are detected by the link layer frame check sequence, each MAC frame with one or more errors is discarded [3]. If the packet loss rate of the wireless and wired links are equal to p_w and p_o respectively, then the total end-to-end probability of packet erasures will be:

$$\pi = 1 - [1 - p_w][1 - p_o] \tag{1}$$

Recall that N_a^m and N_w^m are the maximum allowable retransmission attempts at the application and links layers respectively. Since p_w and p_o are the packet loss rate of the wireless and wired links respectively without considering retransmissions, the packet loss rate for N_w^m-truncated ARQ will be equal to $P_w^{arq} = p_w^{N_w^m+1}$. In addition, the total end-to-end packet loss rate will be equal to $\pi^{N_a^m+1}$. Therefore, Eqn. 1 will become

$$P_{e2e}^{arq^2} = \pi^{N_a^m+1} = (1 - [1 - p_w^{N_w^m+1}][1 - p_o])^{N_a^m+1}, \tag{2}$$

if we employ both application and link layer ARQ. This equation will be proven a very useful tool for exploring tradeoffs for different combinations of application and link layer ARQ.

IV. Analysis

Having described the streaming system, and both the source and channel models, we will now calculate the end-to-end delay, and the effective packet loss rate for the hybrid wireless/wireline network.

For the system configuration that we described in section II, the one-way end-to-end delay for a video packet l will be:

$$L_{l,o} = L_{enc/dec} + L_{pkt} + L_w + L_N \tag{3}$$

$L_{enc/dec}$ is the encoding and decoding delay, while L_{pkt} is the packetizer and de-packetizer delay. These two delay components are generally constant, and have minimum effect in the overall delay [4]. Therefore, we ignore them in our calculations for the rest of this paper. The wireless link and wireline network delays are denoted as L_w and L_N respectively. In this section we will derive the expression for these two delay components. Recall from the previous section, that when we calculated the end-to-end packet loss rate $P_{e2e}^{arq^2}$, we only considered the packet loss due to packet erasures both at the wireline and wireless links along with the different levels of application and link layer ARQ. However, the delay model that considers ARQ, will give us the ability to estimate the number of packets that were not subject to an erasure, but they were late for their prescribed playback deadline. Therefore, the overall effective packet loss rate in that case will be

$$\epsilon_l = P_{e2e}^{arq^2} + (1 - P_{e2e}^{arq^2}) P_r\{L_{l,o} > \tau_l\}, \tag{4}$$

where $P_r\{L_{l,o} > \tau_l\}$ expresses the probability that source packet l is late for the prescribed playback deadline.

A. Delay in the Wireline Network

In this work, we model the one-way Internet delay L_N as a Gamma distribution, since this distribution captures the main reason of Internet packet delays, which is the buffering in the routing infrastructure [7]. Note that this delay distribution is used for both the forward and backward wireline paths. Now consider the case where a packet erasure happens, and a negative acknowledgement is immediately sent from the client. The delay for unsuccessful transmission (including the immediate notification with NACK) is given by the joint distribution of the forward and backward trip times, i.e. $\int_0^\infty f_F(t) * f_B(t) dt$. Subsequent retransmissions will introduce additional delay both in the forward and backward paths. For i application layer retransmissions, if we consider only the latency component of the wireline path, the probability that a packet l is late for its deadline will be:

$$P_r\{L_{l,N}^{(i)} > \tau_l\} = \int_{\tau_l}^{\infty} f_F^{(1)} * ... f_F^{(i+1)} * f_B^{(1)} * ... f_B^{(i)} dt \tag{5}$$

B. Delay in the 802.11 WLAN

The next step in our analysis is the derivation of the transmission delay for every 802.11 frame. Generally, this delay depends on the number of contending users at the MAC layer, which is a parameter that may change dynamically. However, in this paper we consider a single transmitting user where packet losses are only caused by wireless link errors. Even though this assumption may look limiting, it is not since when the 802.11e MAC protocol is used, it is able to provide bandwidth guarantees, and effectively a delay bound. Regarding the detailed operation of the 802.11 MAC protocol, the interested reader is referred to [8]. In this section we will explain the protocol behavior when a single link layer frame is transmitted correctly and when it is not.

1) Correct Transmission: Let us denote the transmission delay for a single frame during a loss and no-loss event as $L_{1,w}$ and $L_{2,w}$ respectively, S_o the combined header overhead for the packet l, and R_m the raw data rate for the selected PHY mode. With no other user active, when the wireless station transmits a data packet, it receives an acknowledgment after a duration of $SIFS\ \mu sec$ [8]. When it receives the ACK frame, the sender transmits a new data frame after another period equal to $SIFS$. Therefore the transmission delay is simply equal to $L_{2,w} = SIFS + \frac{S_l+S_o}{R_m} + \frac{S_{ack}}{R_m} + SIFS$.

2) Incorrect Transmission: For a single transmitting user, there are two cases that will require a retransmission of a link layer frame. It will happen either with the loss of the data frame or the loss of the acknowledgment frame. When a single data frame transmission fails, the sender does not get an ACK after $SIFS\ \mu sec$ as expected. According to the 802.11 MAC protocol, the sender retransmits the data frame after $PIFS$ μsec [8]. Therefore, the transmission delay in this case will be $L_{1a,w} = 2\frac{S_l+S_o}{R_m} + PIFS + SIFS + \frac{S_{ack}}{R_m}$. Even though an ACK frame from the receiver fails, the sender is able to sense the failed ACK transmission (busy medium). When this transmission ends and the medium is free, the sender waits for $SIFS$ and retransmits the data frame. The transmission delay in this case will be equal to $L_{1b,w} = 2\frac{S_d+S_o}{R_m} + 3SIFS + 2\frac{S_{ack}}{R_m}$. The combined packet loss probability for these two cases is given by $p_w = p_{1,w}p_{2,w}$. The combined delay caused by any of the two frames loss events is $L_{1,w} = p_{1,w}L_{1a,w} + p_{2,w}L_{1b,w}$.

3) Average Delay: Given that the maximum number of link layer retransmissions is N_w^m, the probability that i retransmissions are needed for successful transmission, out of the N_w^m allowable is given by:

$$\pi_w(i, N_w^m) = \frac{(1-p_w)p_w^i}{1-p_w^{N_w^m+1}} \tag{6}$$

Also, given N_w^m, and the wireless packet loss probability p_w, we can calculate the average number of retransmissions at the link layer as:

$$\bar{N}_w = \frac{1-p_w^{N_w^m+1}}{1-p_w} \tag{7}$$

The same equation will hold for application layer ARQ. If we add the frame transmission duration for the case where there was a delivery failure even after the N_w^m-th retransmission, we obtain the average link layer layer frame transmission delay for N_w^{max}-truncated ARQ:

$$L_{l,w} = \sum_{i=0}^{N_w^m} \pi_w(i, N_w^m)[iL_{1,w} + L_{2,w}] + (N_w^m + 1)L_{2,w} \tag{8}$$

C. Overall Delay and Effective Packet Loss Rate

In the previous section we showed that a packet will always experience an average link layer delay $L_{l,w}$ as given by Eqn. 8. Since we derived a formula for the average $L_{l,w}$, and not the actual distribution of the random variable, the sender continuously re-calculates this value. Therefore, after i application layer retransmissions, the probability that the application packet l missed the playback deadline is:

$$P_{r,l}^{(i)} = P_r\{L_{l,N}^{(i)} + iL_{l,w} > \tau_l\} \tag{9}$$

The last expression captures the probability that the packet missed its deadline after i retransmissions. Since $\tau_l^{'} = \tau_l - iL_{l,w}$ is constant for given i, the pdf of the previous equation is a shifted version of the pdf that we derived in Eqn. 5. Now, we have to calculate the probability that i application retransmissions are needed for a single application packet. In the same way we derived this expression for the link layer retransmissions, we can write for this probability:

$$\pi_a(i, N_a^m) = \frac{(1-P_w^{arq})(P_w^{arq})^i}{1-(P_w^{arq})^{N_a^{max}+1}} \tag{10}$$

So the probability for an application layer transport packet to miss the playback deadline, for any possible number of used retransmissions, will be:

$$P_{r,l}\{L_{l,o} > \tau_l\} = \sum_{i=0}^{N_a^m} \pi_a(i, N_a^m) P_{r,l}^{(i)} \tag{11}$$

The final equation that expresses the effective packet loss rate is obtained after we substitute in Eq. 4, the results that we derived in Eqns. 2 and 11. Finally, from our previous discussion, we can see that channel rate consumed for retransmissions for a source of R_s packets per second, will be equal to $R_s Q \bar{N}_a \bar{N}_w$, if Q link layer frames are used for a single application packet. Therefore, if R_T is the available channel rate, the Shannon's theorem dictates that:

$$R_s + R_s Q \bar{N}_a \bar{N}_w \leq R_T \tag{12}$$

This is essentially one of the constraints that a selected error protection strategy must adhere to.

V. Optimal Retransmission Adjustment

Due to space limitations we only highlight the cross-layer adaptive ARQ algorithm. However, the basis of the algorithm is the analytical model that we described in the previous sections. Therefore, it is obvious that the most important function of a model-supported algorithm is the correct estimation of the needed parameters in real-time. For example, the packet loss rate due to wireless link layer erasures $p_{1,w}$ and $p_{2,w}$, is obtained from the underlying MAC layer which maintains this information in the Linux $/proc/net/wireless$ filesystem. Subsequently, the wireless host calculates the link layer delay L_w according to Eqn. 8, and the wireline packet loss rate according to Eqn. 1. The wireline network delay is calculated by measuring the parameters of the Gamma model through RTCP receiver reports [9]. The algorithm tries to minimize the effective packet loss rate ϵ_l for a specific packet l, subject to the maximum allowable end-to-end delay τ_l, and the available channel rate R_T (Eqn. 12). This task is achieved by gradually increasing the protection level for each packet being considered for transmission. After the algorithm derives the optimal values for N_a^m and N_w^m, it sets the actual value for N_w^m with the Linux $iwconfig$ tool.

VI. Experiments

The experimental setup includes a wireless video streaming server, equipped with the Atheros AR5005G wireless network adapter, and a client located in the wireline network. We set wireless link data rate with $iwconfig$ to 12Mbps, while the distributed coordination function (DCF) of the IEEE 802.11a MAC protocol is also activated. The timing parameters were set as follows SIFS=16μsec, PIFS=25μsec, DIFS=34μsec. The overhead that is added to the packet size from RTP/UDP/IP/MAC (application, transport, network, mac) is 68 bytes. Occasionally, we inserted cross-traffic in the wireline network, that did not generated packet losses but resulted in delay fluctuations. The QCIF sequence Claire was encoded at 30fps with the H.264/AVC JM12.2 software. We used only I and P frames and we set the GOP size 32 frames, while no source coding error-resilience tools were used. The QP was set equal to 14 in order to get constant video quality, at an average bitrate of 512Kbps. Furthermore, the maximum source video packet size at 1000 bytes. With this frame size, the encoded P frames are packetized into three source packets that have the same decoding deadline. For obtaining comparative results we implemented a hybrid scheme where FEC is applied at the application layer, while ARQ is applied at the wireless link layer. We name this system ARQ/FEC, while the proposed system is denoted as ARQ^2.

We present PSNR results in Fig. 2 for wireline and wireless packet loss rate equal to $p_o = 10\%$ and $p_w = 5\%$ respectively. The packets that are lost, after the successful transmission of I frames, belong to frames 8 and 49. In the first case, ARQ^2 is able to retransmit on-time the lost packet, while in the later case the proposed system defers from retransmitting. However, with this experiment, an interesting situation arises since there are are subsequent packet losses before the correct transmission of the next I frame. Packets that belong to frame 25 are lost, and even ARQ^2 avoids retransmitting, since the current end-to-end delay estimate, exceeds the allowable delay. However, the important observation is that quality is not degraded as bad with ARQ^2 as with ARQ/FEC, since the later system suffers already from error propagation in the decoding loop. In the second event of packet losses, frame 49 is not recovered by any of the two systems on-time. However, with ARQ^2, we are able to retransmit the lost packet of frame 55, and keep quality at a better level (nearly 40 dB). From this preliminary experimental evaluation, we see that the important benefit of our system is that it enables correct decisions regarding retransmissions, that are very crucial for the quality of the decoded video sequence.

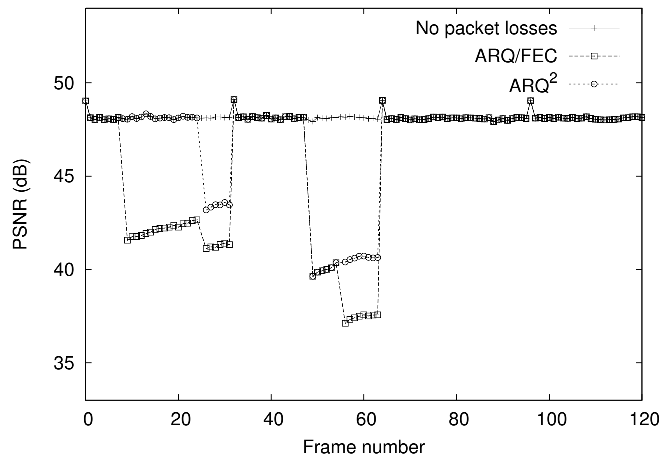

Fig. 2. Instantaneous PSNR for uplink video streaming. Delay constraint is set at 200ms, and $R_T = 1.5R_s$.

VII. Conclusions

In this paper, we presented a cross-layer error control mechanism for low-delay and robust video transmission in tandem-connected wireless LANs and wireline packet switched networks. This mechanism was developed by observing the inadequacy of current proposed schemes to capture all the possible ARQ strategies that exist both at the application and link layers. We initially developed an analytical model that expresses the packet loss rate and delay, as a function of both the application and link layer ARQ. Based on these model, we developed an algorithm that determines the optimal protection level given the delay and channel rate constraints. Our preliminary experiments indicate that the proposed algorithm is able to identify protection strategies that improve substantially the video quality at the client.

References

[1] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Prentice Hall, 2002.

[2] Q. Li and M. van der Schaar, "Providing adaptive qos to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 278–291, April 2004.

[3] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE Journal on Selected Areas in Communication*, vol. 21, no. 10, pp. 1752–1763, December 2003.

[4] F. Zhai, "Optimal cross-layer resource allocation for real-time video transmission over packet lossy networks," Ph.D. dissertation, Northwestern University, 2004.

[5] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction," RFC 2733, December 1999.

[6] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ error control for fading mobile radio channels," *IEEE Transactions on Vehicular Technology*, vol. 46, pp. 445–455, May 1997.

[7] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," University of Pennsylvania, Tech. Rep. MS-CIS-92-83, Tech. Rep., 1992.

[8] "Part 11: Wireless LAN Medium Access Control and Physical Layer (PHY) Specifications," IEEE 802.11-1999 Standard, August 1999.

[9] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *Microsoft Research Technical Report MSR-TR-2001-35*, 2001.