A Transparent ARQ Scheme for Broadband Wireless Access

Özgür Gürbüz
Faculty of Engineering and Natural Sciences
Sabanci University
Orhanli-Tuzla, Istanbul 34956, TURKEY
ogurbuz@sabanciuniv.edu

Ender Ayanoglu
Center for Pervasive Communications and Computing
Department of Electrical Engineering and Computer Science
The Henry Samueli School of Engineering
University of California
Irvine, CA 92697-2625, USA
ayanoglu@uci.edu

***Abstract*— In this paper, we propose an Automatic Repeat reQuest (ARQ) technique for enhancing the reliability and throughput of broadband wireless Internet access applications, and provide examples of its performance. ARQ is a method for making a link and flow more robust to errors. Other than enhancing link quality, ARQ provides early error recovery, so that the transport layer does not lower the data throughput. The cost of these enhancements is additional overhead. In this work, we have extensively analyzed our proposed ARQ scheme along with realistic wireless channel and protocol stack models. We have quantified the performance gains of our ARQ scheme as reduced end-to-end delay, increased throughput, SNR gain and fade margin, also providing the overhead costs. We have proposed and analyzed enhanced features, namely, Segmentation and Reassembly (SAR) and Bitmap Compression. We showed that these features reduce the overhead, the extent of which depends on the nature of the errors.**

I. Introduction

Wireless links suffer from impairments such as path loss, multipath effect, shadowing, fading, noise and multiple access interference, which cause channel errors. Error control techniques are applied in terms of Forward Error Correction (FEC) or ARQ to recover from these errors. Since in an FEC scheme the throughput efficiency is set by the code rate, it is constant regardless of the channel conditions. On the other hand, the throughput efficiency of ARQ schemes depends mainly on the number of requested retransmissions, i.e., the channel quality. Powerful FEC schemes may be preferred to ARQ schemes in the presence of delay constraints, and ARQ schemes are widely employed when delay is tolerable [1]. Hybrid approaches combine the advantages of both techniques [2]. Eventually, it is crucial to have a good quality link to achieve high rate, i.e., broadband services over wireless systems.

Most data services use transport control protocol (TCP) for reliable delivery. TCP is primarily designed for wired networks for error and flow control. Errors in wired networks occur due to congestion, which occurs in heavy traffic conditions. Transport or TCP layer error recovery enforces flow control to slow down the connection, which in turn reduces the data throughput. Several works in the literature have addressed the behavior of TCP in wireless channel errors, and its interactions with ARQ [3], [4], [5]. All imply that, other than enhancing link quality, ARQ is essential for early error recovery, so that the transport layer does not detect any errors and keeps the data throughput the same.

An example of broadband systems is fixed wireless access that can allow high data rate access to subscriber homes, and small and medium businesses. These systems provide point-to-multipoint connectivity with end-user applications such as high-speed Internet access, premium streaming audio, video and voice. The multiple access environment we consider is a fixed wireless system that uses VOFDM [6], [7]. The chosen MAC layer, DOCSIS, is originally designed for hybrid fiber coax cable networks. Due to the high througput, low delay performance of DOCSIS and due to its strengths in supporting multimedia traffic with Quality of Service, DOCSIS was adapted for the VOFDM air interface, as the MAC protocol of the broadband system [8], [9]. The point-to-multi point system consists of a headend router that is wirelessly connected to wireless modems, i.e., the subscribers. The headend router accesses the subscribers over the downstream channel, and subscribers share the upstream channel. The upstream and downstream channels are separated by Frequency Division Duplexing (FDD). On the uplink, DOCSIS defines contention and contention free periods for channel allocation. On the downlink, it is essentially a TDMA-like system, orchestrated with an intelligent headend scheduler. The headend scheduler informs the subscribers for the schedule of future time slots, assigning some of them as contention slots and some of them as contention-free grants to subscribers. Subscribers send tranmission requests, and contend to get a grant from the scheduler. The transmission request involves the number of requested slots. The initial version of DOCSIS was designed for best effort data, utilizing the channel up to 80%, and the latest version involves all the mechanisms for supporting multimedia services and QoS [9], [10]. Since DOCSIS does not have any error correction functions, we needed additional error control mechanisms to improve the link quality for better range and availability and to maintain high data rates without throttling TCP. The goal is to design a simple algorithm that provides powerful error correction, while causing minimal air overhead and minimal changes to the system.

This paper proposes an ARQ protocol based on Selective Repeat for the point-to-multipoint broadband wireless system in [6], [7]. The proposed algorithm is presented as a design case, and its performance is evaluated through extensive simulations. The rest of the paper is organized as follows. Section II presents the basic ARQ algorithm and enhanced features. Section III explains the simulation models and results, presenting the performance gains

 0-7803-8344-3/04/$20.00

of the ARQ scheme. Section IV includes our conclusions.

II. The Proposed ARQ Scheme

A. Basic ARQ

The throughput efficiency of Stop-and-Wait and Go-back-N schemes are low for high error rates which can occur in wireless channels. Unnecessary retransmissions such as those in the Go-back-N scheme need to be avoided, since the radio resources are shared and scarse. Besides, these schemes are better suited for point-to-point links, where round trip delays are low. For point-to-multipoint systems, the round trip delay can be high due to collisions and access delays.

The ARQ protocol proposed in this paper is based on Selective Repeat. It is a transparent scheme designed as a layer inserted between Ethernet and DOCSIS layers (Figure 1). The sender and receiver operations are the same for both the upstream and downstream paths: The sender adds an ARQ header with sequence numbers to MAC frames and keeps a copy of the transmitted frames in the transmitter window until it receives an acknowledgement from the receiver. On the other end, the receiver buffers the frames upon detecting an out-of-sequence packet, until all missing frames are received. Figure 2 shows the sender/receiver window. W_t indicates the size of the transmit window, which is the maximum number of outstanding packets, and the receive window size, W_r, shows the maximum number of packets that can be received in sequence. F_t is the sequence number of the first unacknowledged packet, while F_r is the sequence number of the first non-received packet. L_t is the sequence number of last outstanding packet, and L_r is the largest sequence number received. The receiver confirms the received frames and notifies the missing ones via the acknowledgement (ACK) messages, sent periodically or upon request from the sender. The ACK messages contain the sequence number of the first missing packet (F_r) and a bitmap of the packet positions in the receiver's window (up to L_r). A "1" in the bitmap indicates the presence, and a "0" indicates the absence of a packet at the corresponding position. Having received an ACK message, the sender advances its window, releases all successful frames and retransmits only the missing frames requested by the receiver. For full reliability, the sender can retry for retransmission of missing packets and acknowledgement requests for infinite number of times. Partial reliability can also be implemented by setting finite values to the limit of retries, in which case, the sender gives up on retries and releases packets out of its window.

The parameters of the protocol are timers, buffer sizes, and maximum number of retransmissions: The sender's timer, *RequestACKTimer*, (T_r), is used to ask for ACKs from the receiver; the receiver's *PeriodicACKTimer*, T_a, is the regular timer that produces the ACK messages periodically. The *FrameFlushTimer*, T_f, is employed at the receiver in the case of mismatched sequence numbers. Buffer sizes are W_t and W_r as explained above. R_t indicates the maximum number of retries for transmitting a packet, and R_r stands for the maximum number of retries for a RequestACK message. A separate ARQ process is applied for each flow. Therefore, the flows can be classified and ARQ parameters can be chosen optimally, according to different traffic classes so that the system resources can be most efficiently utilized.

B. Segmentation and Reassembly (SAR)

Basic ARQ operates on MAC PDUs. For flows with bulk data transfer, such as FTP of large files, MAC PDUs are as large as 1500 bytes. In the case of errors, the entire PDU has to be retransmitted, even though only a portion of it is corrupted. With SAR employed at the sender, a MAC PDU is divided into smaller fixed size segments of size *S*. Each segment is encapsulated with Ethernet header and CRC, before the ARQ data header is appended (Figure 3). The receiver collects segments and requests retransmission of missing segments via ACK messages and bitmaps. When all the segments are received, the segmented PDU is reassembled and released to the application. With the SAR feature, error detection and recovery is on a segment basis. Hence, retransmission overhead and error recovery delay can be reduced. The drawback is increased overhead.

C. Bitmap Compression

The ACK messages introduce overhead on the feedback or receive path. This overhead is variable depending on the nature of the errors since a simple compression is applied in basic ARQ: The ACK message starts with the first missing packet (i.e., no ACK is sent for packets received in sequence) and the bitmap length is variable as it spans the buffer positions until the last sequence number received. However, with heavy traffic and high error rates, the receiver buffer can get full and the bitmaps can get long, increasing the overhead. When the errors are bursty, the bitmap can again get long with a sequence of zeros. To reduce the feedback channel overhead, further compression is needed. As an enhancement to basic ARQ, a run-length coding like compression technique is proposed. Our compression rule exploits the changes in the bitmap sequence pattern. Using bitmap compression, the bitmap again starts with a missing packet, i.e., zero. Then the location of the next change of information content, i.e., change from zero to one, is included instead of sending all the individual positions (bits) in the bitmap. By only encoding the changes, the number of bits can be reduced, and the bitmap can be shortened.

III. Performance Analysis

Performance of ARQ schemes has been studied analytically in the literature for point-to-point links [11]. These results cannot be directly applied to the point-to-multipoint case since the performance depends on the applied media access control scheme and the physical layer characteristics. Moreover, when the performance of ARQ is to be considered along with TCP applications, analytical formulation becomes impossible and unrealistic due to the operation and interaction of many protocols. Therefore, simulation is the most appropriate way to assess the performance in the presence of all protocols in the stack and the wireless channel.

A. Wireless Channel Models

In this work, two types of wireless channel impairments were modeled to test the performance of our ARQ schemes. The first type is the additive white gaussian noise (AWGN) channel that characterizes the impairments due to thermal noise and interference. Our AWGN channel model produces random errors with a uniform distribution, which can be configured with different average error rate values. Average Codeword Error Rate (CER) is used as the benchmark of errors. A frame is considered as a collection of codewords (OFDM bursts), and the frame is dropped if at least one of its codewords is in error. Our second channel model aims to represent the fading characteristics of the wireless channel. We have taken the models of fixed wireless channels presented in [12] as a baseline. We used a Matlab simulation to generate the time samples of a multi-path Rayleigh fading channel [12] and passed it through a filter, defined by the data rate. The received power is sampled once every ten codewords, and these power samples are converted to codeword errors by a threshold comparison: A codeword error is assumed to occur any time the received power, P_r, is below the threshold power P_{thr}. A level of "1" in the graph indicates that the power is above the threshold, i.e., there are no errors, while a "0" indicates a codeword error. Figure 4 shows a sample power timeline and the channel status obtained by threshold comparison. Again, average CER is used as the benchmark of errors and it is calculated for the fading channel, as a function of the threshold value as, $CER = OccurrenceRate(P_r <= P_{thr})$. Different channel status timelines are obtained for different power thresholds, which correspond to different average CER values. In this type of channel, the errors occur consequtively, in a burst.

B. The Protocol and Network Models

The OPNET simulation tool was used to model the entire fixed wireless system. DOCSIS MAC layer, Ethernet, ARQ and the layers of the TCP/IP stack were modelled. Figure 5 shows the headend and subscriber nodes. The MAC layer implements DOCSIS functionality and air interface, including the headend scheduler. The ARQ module between the Ethernet and DOCSIS MAC layers implements the proposed ARQ algorithm, basic and enhanced features. The wireless channel models are implemented as tx-channel and rx-channel modules for upstream and downstream channels respectively. The AWGN and fading channel errors are modeled in these blocks: For the AWGN channel, the output of the uniform error generator is applied to check for the codeword errors in the MAC frames. The average codeword error rate is specified as an input to the simulation. For the fading channel, the error timeline data obtained from Matlab channel simulations is used. Each time a packet is to be transmitted (or received) over the physical interface, the error timeline data is referred to check for channel status and find out if any codeword errors occur during its transmission. In both cases, a packet is considered to be in error, and is dropped when at least one codeword error occurs during its transmission time.

C. Results

In the simulation experiments, a File Transport Protocol (FTP) application was considered. The end-to-end performance for the cases with and without the ARQ feature were evaluated in AWGN and fading channels. The system data rates were selected as 14.7 Mbps for the downstream channel, and 3.2 Mbps for the upstream channel [7]. The simulation scenario included 20 users, performing FTP transactions simultaneously, over the upstream channel. The file transfer took place from the subscribers to the headend. The average file transfer rate was chosen as 100 files/hour, with an exponential interarrival distribution. The average file size was 500 kilobytes, with a truncated normal distribution (most files were sized 450-550 kilobytes). The traffic parameters were set such that the upstream channel was loaded by 70%. The errors were modeled for the upstream channel only; the downstream channel was error-free[1]. The performance of the following systems were compared in AWGN and fading channels: No ARQ, Basic ARQ, ARQ with SAR (Segment Size = 400 bytes) and ARQ with Bitmap Compression. The following performance metrics were observed: Average and Cumulative Distribution Function (CDF) of FTP Download Response Time (end-to-end FTP delay), Average Throughput, Average ARQ overhead (Packet , retransmission and feedback channel overhead), and the gain in Signal to Noise Ratio (SNR).

We first present the AWGN channel results: Figure 6 illustrates the average FTP download delay as a function of average CER of the channel. FTP download delay accounts for the time interval from an FTP session request is made, until the entire file is downloaded and the session is closed. Figure 7 shows the throughput of the upstream channel, measured at the DOCSIS MAC layer, again as a function of average CER. Both figures clearly indicate the performance degradation, decreasing throughput and increasing delay, when ARQ is not employed. This is due to TCP's flow control mechanism that assumes that errors are caused by congestion of the buffers, and slows down the traffic by reducing its window size. When ARQ is employed, the errors are recovered earlier than the TCP layer, so that congestion control does not take in effect. Both figures show that the performance difference between ARQ and No ARQ becomes more severe (almost an order of magnitude difference) as the error rate is increased. It can also be seen that with SAR, the delay is reduced to half of basic ARQ, while the throughput increase is around 10%. The performance enhancement with SAR is due to faster error recovery, since only the segment in error needs to be retransmitted, while in basic ARQ the entire frame is to be retransmitted. The cost of ARQ is added overhead which is added both in the transmission path (upstream channel in this case) and the receive path (downstream channel). The upstream overhead is due to retransmissions, added packet overhead to the frames, and control frames, while the downstream overhead is due to acknowledgements. Figure 8 depicts the upstream overhead of basic ARQ and ARQ with SAR, as a function of error rate in the AWGN channel simulations. The plot on the top compares the overhead due to retransmissions. It can be seen that SAR reduces the retransmission overhead by up to 75%, since only the errored segments of a frame are retransmitted. As opposed to this, the ARQ packet overhead is increased since the number of packets are increased with segmentation as shown in the middle plot. The plot at the bottom indicates that for error rates less than 1%, the total ARQ overhead of ARQ with SAR is larger than basic ARQ, with only slight enhancements in throughput and delay. For error rates larger than 1%, SAR starts to become advantageous in terms of reduced total ARQ overhead, better throughput and delay. Figure 9 shows the downstream overhead as a function of error

[1]The upstream and downstream channel errors are uncorrelated due to FDD operation. Since the downlink channel has higher power and the critical link is the uplink, this assumption was made.

rate. It can be seen that the overhead for ARQ with SAR is larger than basic ARQ due to increased number of packets (segments) in the system since larger size bitmaps are needed to encode larger number of packets. Next, the performance is observed with the bitmap compression. It is concluded that bitmap compression reduces the receive path (feedback channel) overhead by 50% for basic ARQ. For ARQ with SAR, the feedback channel overhead is reduced by bitmap compression up to an error rate of 2.3%. For larger error rates, bitmap compression becomes unnecessary. This is possibly due to a large number of uniformly distributed errors in large bitmaps, which cannot be well compressed by the proposed run-length like compression scheme.

Next, we present the results for the fading multipath channel: Figure 10 illustrates the average FTP download delay as a function of average CER. In this case, the average CER is varied by considering different power thresholds, and different channel status timelines are obtained for each level, as explained in Section III.A. Figure 11 shows the throughput of the upstream channel, measured at the DOCSIS MAC layer, again as a function of average CER. The cases No ARQ, basic ARQ and ARQ with SAR are compared. Both figures indicate the performance degradation in the No ARQ case, as compared to the results with basic ARQ. It can also be seen that ARQ with SAR scheme outperforms No ARQ case, but it performs worse than basic ARQ. The reason is the bursty characteristics of errors in this type of channel. When an error occurs, consecutive segments are lost, so all of them have to be retransmitted. Figure 12 depicts the upstream overhead of basic ARQ and ARQ with SAR as a function of error rate. It can be inferred that SAR reduces retransmission overhead, however the difference between basic ARQ and SAR cases is not as big as the AWGN channel, and it gets smaller as the error rate is increased. oreover, the packet overhead is increased due to increased number of packets. Therefore, the overall overhead of ARQ with SAR is larger than the overhead of basic ARQ scheme. Figure 13 depicts the effect of bitmap compression on the feedback channel overhead. Due to the bursty nature of the errors, bitmap compression reduces the downstream channel overhead.

Another performance metric we examined for the fading channel is the SNR gain. In addition to the advantages of faster error recovery and increased throughput, ARQ also enhances the link quality. With the help of ARQ, the system performs as if it is operating at a higher SNR than its actual SNR level. The difference between the two SNR levels is the SNR gain that is provided by ARQ. In order to quantify the SNR gain of ARQ, the CDF of the FTP download response time is observed for different CER values, for No ARQ and ARQ with SAR cases, as shown in Figure 14 (the results of basic ARQ were similar to ARQ with SAR, so only one set of results is provided). First, a target delay performance of 20 seconds is determined as the 90 percentile delay for the FTP session. Then, the CER levels below which the end-to-end delay is less than 20 seconds with 90% probability are determined for each case. Examining the first plot with No ARQ, we found that the delay is less than 20 seconds 90% of the time for $CER \leq 10E-3$. The same target delay can be achieved with $CER \leq 5*10E-2$ with basic ARQ and ARQ with SAR. Looking up for the power threshold values to obtain the above CER values, we determined that $P_thr = -13.4dB$ for No ARQ, $P_thr = -6.5dB$ for the two ARQ cases. This means that ARQ introduces an SNR gain of almost 7 dB, since the system performs as if its SNR is higher, as if $CER = 10E-3$, when the error rate is actually $CER = 5*10^-2$. The ARQ SNR gain obtained for the fading channel can be considered as a gain in the fade margin of the system. For power link engineering purposes, a 7 dB gain is very significant for a fixed broadband wireless system.

IV. Conclusions

In this paper, we have proposed, modeled and extensively analyzed a selective repeat ARQ scheme for the wireless broadband system specified in [7]. Our performance analysis involves realistic wireless channel models, as well as the modeling of the entire protocol stack. We have evaluated the end-to-end performance for a typical application, FTP, in the presence of channel errors due to AWGN and fading channel impairments. We showed that without ARQ, TCP's congestion avoidance results in severe throughput reduction and huge delays, which is unacceptable for broadband services. Link layer error correction is essential for enhancing the link quality and allowing earlier error recovery, so that high throughput can be provided to the end users. The performance gains of the ARQ scheme are reduced end-to-end delay, increased throughput, and increased range and fade margin due to the SNR gain obtained. There is a cost to these enhancements, which is the introduced overhead, both on the transmit path and receive path. Enhanced features such as SAR and bitmap compression reduce the overhead, the extent of which depends on the nature of the errors. It is also worth to note that this ARQ feature is to be applied to non-real time applications only.

References

[1] M. Zorzi, R. R. Rao, "ARQ Error Control for Delay-Constrained Communications on Short-Range Burst Error Channels," *Proceedings of IEEE VTC'97*, vol. 3, pp. 1528-1532, May 1997.

[2] M. Zorzi, R. R. Rao , "On the Statistics of Block Errors in Bursty Channels," *IEEE Transactions on Communications*, vol.45, pp. 660-667, June 1997.

[3] A. Chockalingam, M. Zorzi, R. R. Rao, "Performance of TCP on Wireless Fading Links with Memory," *Proceedings of IEEE ICC 1998*.

[4] A. Chockalingam, M. Zorzi. V. Tralli, "Wireless TCP Performance with Link Layer FEC/ARQ," *Proceedings of IEEE ICC 1999*.

[5] A. F. Canton, T. Chahed, "End-to-end Reliability in UMTS: TCP over ARQ," *Proceedings of IEEE Globecom 2001*.

[6] E. Ayanoglu, VK Jones, G. G. Raleigh, J. Gardner, D. Gerlach, and K. Toussi, "VOFDM Broadband Wireless Transmission and Its Advantages over Single Carrier Modulation," Proc. IEEE ICC 2001, June 2001, Helsinki, Finland.

[7] E. Ayanoglu, VK Jones, G. G. Raleigh, J. Gardner, D. Gerlach, and K. Toussi, "VOFDM Broadband Wireless Transmission and Its Advantages over Single Carrier Modulation," IEEE-ISTO Broadband Wireless Internet Forum White Paper, Doc. WP-1 TG-1, December 2000.

[8] E. Ayanoglu, "Media Access Protocols: Circuit Switching to DOCSIS," IEEE-ISTO Broadband Wireless Internet Forum White Paper, Doc. WP-2 TG-1, December 2000.

[9] N. Golmie, F. Mouveaux, D. Su, "A comparison of MAC protocols for hybrid fiber/coax networks: IEEE 802.14 vs. MCNS," Proceedings of IEEE ICC'99, vol. 1, pp. 266-272.

[10] Data-Over-Cable Service Interface Specifications (DOCSIS) 1.1, Radio Frequency Interface Specification, SP-RFIv1.1-I10-030730.

[11] A. Leon-Garcia, I. Widjaja, *Communication Networks*, Mc-Graw Hill, 2002.

[12] V. Erceg and et al., "A Model for the Multipath Delay Profile of Fixed Wireless Channels," *IEEE Journal on Selected Areas in Communications*. Vol. 17, pp.399-409, March 1999.

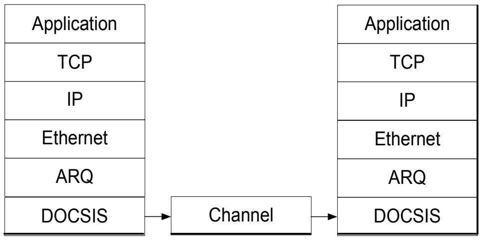


Figure 1: Protocol stack

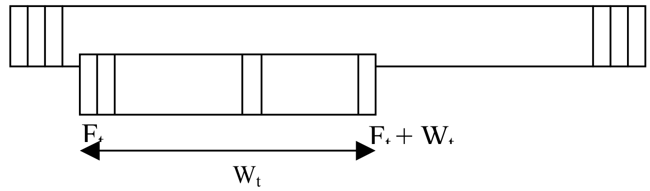


Figure 2: Sender and receiver ARQ windows

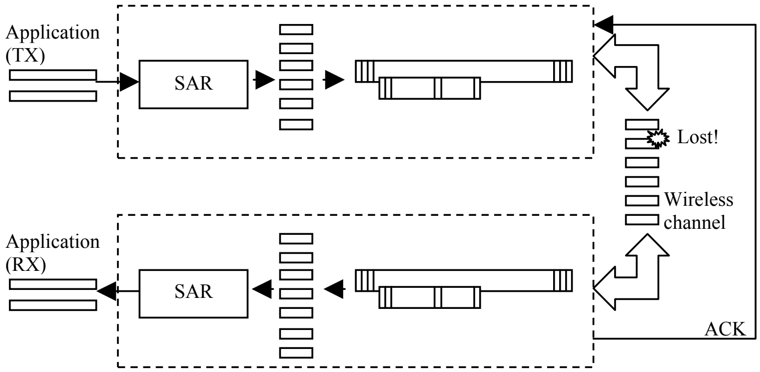


Figure 3: Segmentation and Reassembly (SAR)

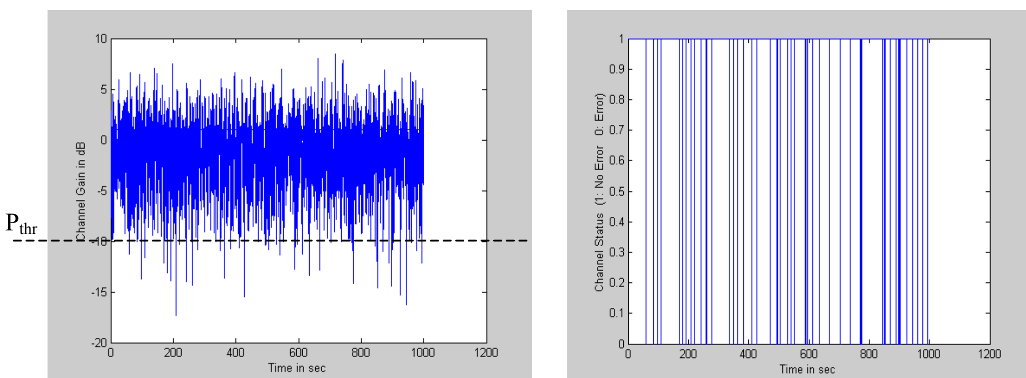


Figure 4: Received power timeline and channel status timeline

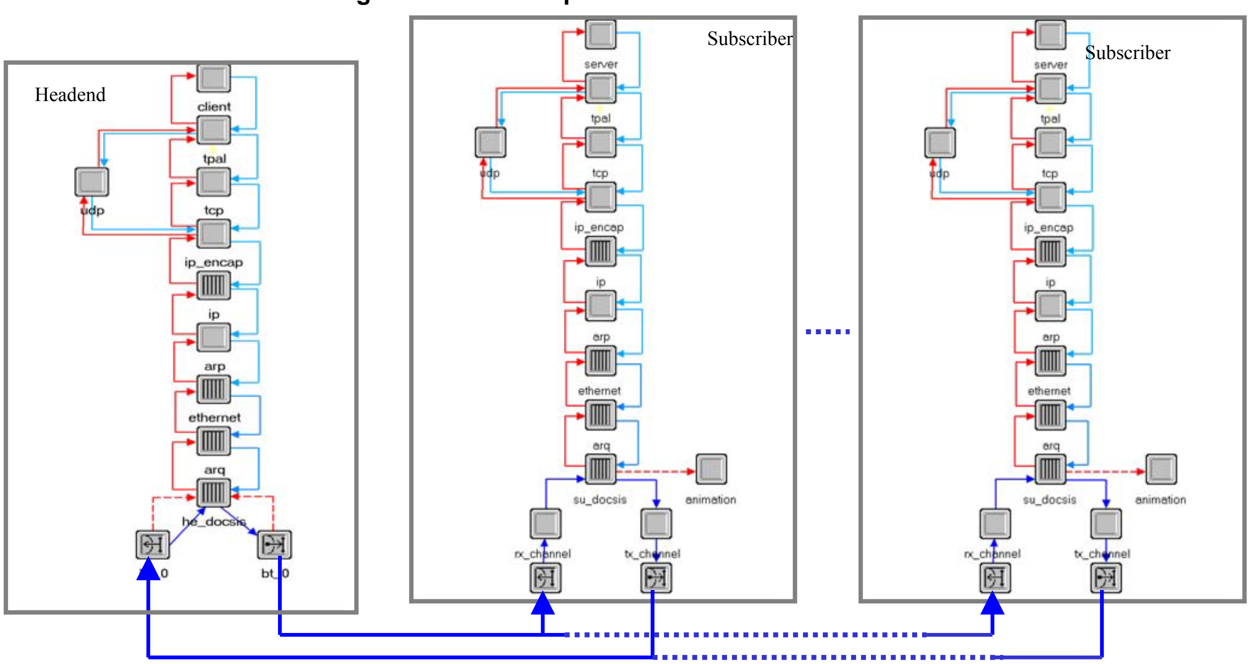


Figure5: Simulation models and framework in OPNET

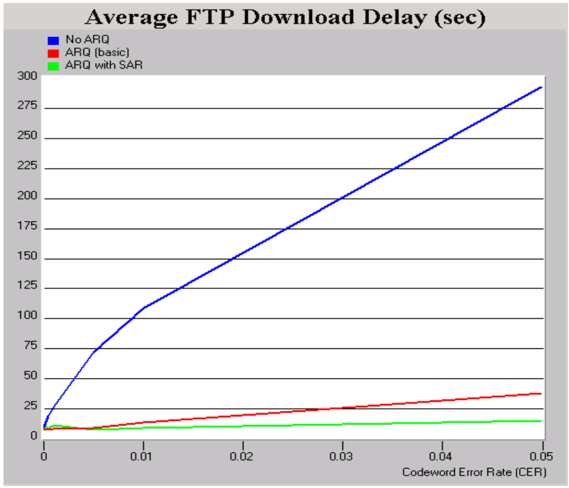


Figure 6: End-to-end FTP delay (AWGN)

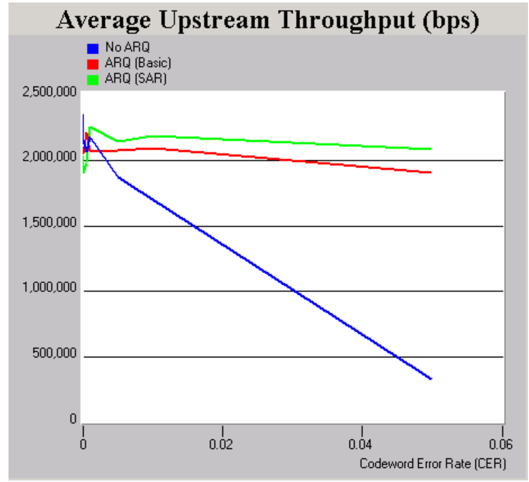


Figure 7: Throughput (AWGN)

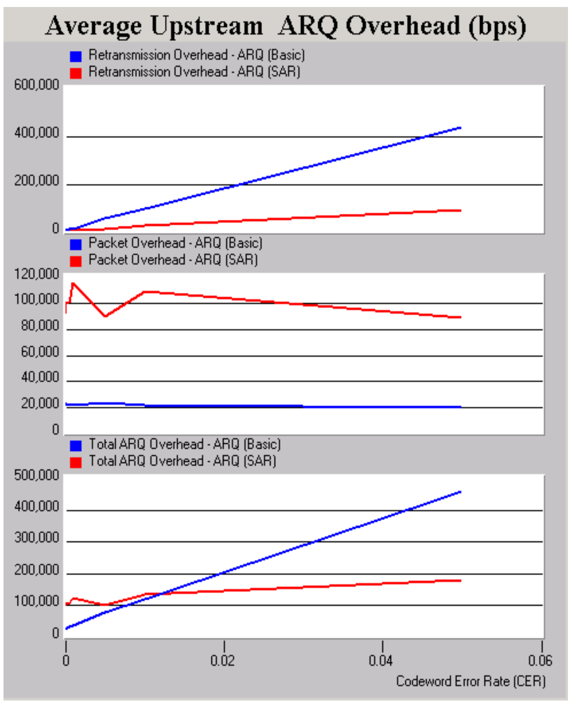


Figure 8: ARQ Overhead - Forward channel (AWGN)

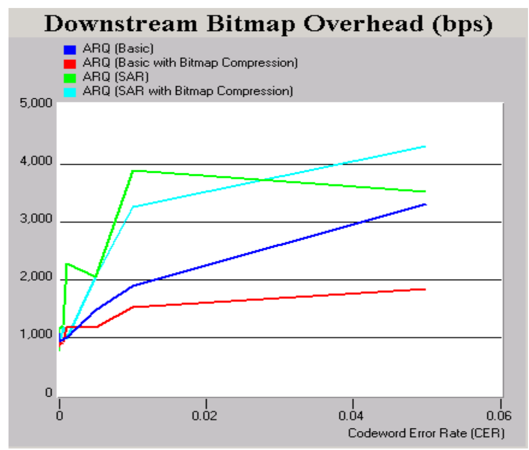


Figure 9: ARQ Overhead – Feedback channel(AWGN)

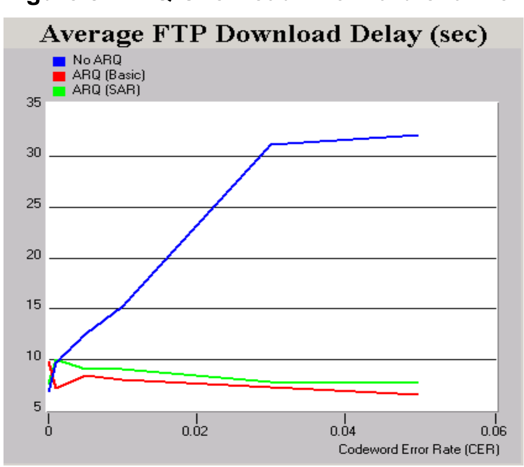


Figure 10: End-to-end FTP delay (Fading)

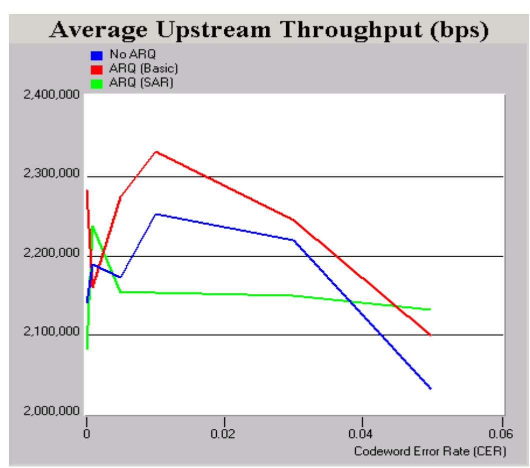
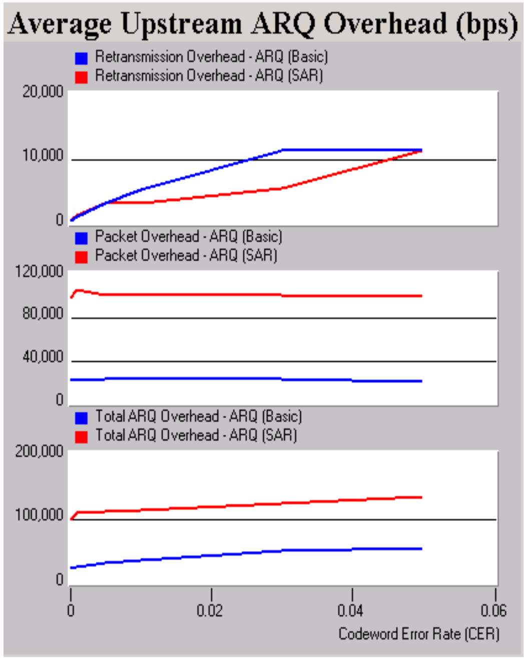


Figure 11: Throughput (Fading)



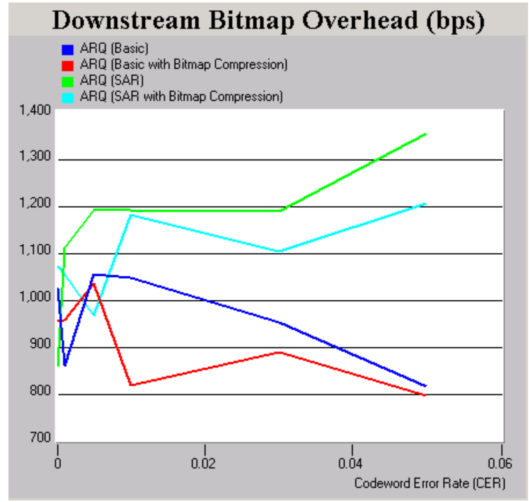


Figure 13: ARQ Overhead – Feedback channel (Fading)

Figure 12: ARQ Overhead - Forward channel (Fading)

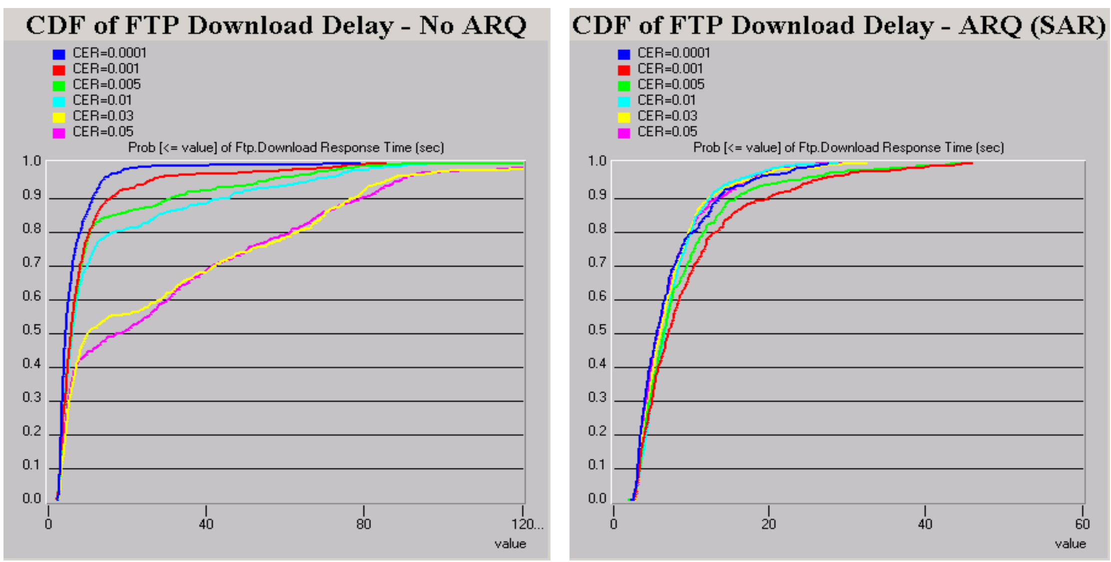


Figure 14: CDF of End-to-end FTP Delay (Fading)