Error Control in Wireless ATM Network

Jianfeng Pu

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Yilu Liu, Chair
Hugh F. Vanlandingham
Robert P. Broadwater
Shu-Ming Sun
Robert J. Donaghey

Appril 24, 2000
Blacksburg, Virginia

Keywords: ATM Network, Wireless, Error Control

Error Control in Wireless ATM Network

Jianfeng Pu

(ABSTRACT)

Asynchronous Transfer Mode (ATM) protocol was designed to support real-time traffic steams over high quality links like fiber optics where the transmission error is extremely low. ATM performs poorly in an error-prone environment such as wireless communications. The purpose of this research is to investigate error control schemes in wireless ATM (W-ATM) to support real-time service, such that the physical layer error conditions are handled in lower layers under ATM transport layer.

Automatic Repeat reQuest schemes (ARQ) and Forward Error Correction (FEC) have been widely used for reliable data transmissions. However, the current existing ARQ schemes can potentially introduce unbounded delay in high error rate environments like W-ATM network due to the lack of delay control mechanism. As a result, they are not appropriate for real-time data communications in which there are strict packet delay requirements. In this dissertation, we explored the issues related to W-ATM area. Adaptation of FEC, specifically Reed-Solomon code, to channel error conditions in W-ATM is investigated. The quality-of-service (QoS)-aware error control algorithm is originated and its performance is evaluated. The algorithm is further simplified to make it more suitable for practical applications.

The requirements of ARQ applicability for real-time communication environment like W-ATM is extensively analyzed. An ARQ scheme, called D-bit protocol, is developed to satisfy the real-time requirements. The scheme supports reliable packet discarding while allowing retransmissions without compromising user-level QoS for real-time stream applications. Simulations show the effectiveness and liveness of the protocol.

Table of Contents

List of Figures

Chapter 1: Introduction

This chapter serves as general introduction of some network concepts that are related to our research in this dissertation. The current research status of each area is briefly discussed.

1.1 Data Communication Networks

Even telecommunications can be dated back to over a hundred of years ago, the first large-scale, general-purpose data communication network was APPANET, which developed by Bolt Beraneck and Newman (BBN) under the sponsorship of U.S. Department of Defense in 1971. APPANET geographically interconnected distributed computer systems and users in some defense research institutions, universities and government agencies. After that, commercial offerings of packet-switched services rapidly followed. For example, TELNET, the first public packet-switched network, linking host computers and dial-up terminal users. Since then, data networks have been exponentially growing. A primary example is the Internet evolved from APPANET, which provides connectivity to many universities and national telecommunications networks. According to some research, the Internet traffic doubles every 3-5 months in the last few years. The information on the Web is so abundant, even most sophisticated web searching engine can only cover less than 10%. And this information base is still growing rapidly and expect to continue. The network infrastructure will drastically changes out society, our economy, our working habits, and the way we live.

1.1.1 Network Architectures

There are many ways to classify network architectures. Topologically, there are three types of network architectures: fully connected point-to-point networks, switched point-to-point networks and broadcast networks.

In fully connected point-to-point networks, any node has at least a direct connection with any other node in the network. So there is no switching action performed when a node communicating with any other nodes. However, the number of links required for such a network of N node is N(N-1)/2, which is O(N*N). This is obviously prohibitive for large-scale network. This architecture is used only for small number of nodes.

In the switched point-to-point network, each endpoint is connected to a switch by a single point-to-pint link. A switch may have multiple endpoints connecting to it. Among the switches, there are at least one path (consisting of one or more physical links and switches) that can be found between any switches. The main functions of the switch is to forward information received from each input port destined for a particular receiving endpoint(s) to the right path(s) across the network to reach that desired endpoint(s). The physical links within the network are shared among different pairs of communicating endpoints using a multiplexing mechanism. In other words, the physical links are a shared resources, so it is more efficient in utilization as opposed to the fully connected network architecture where links are dedicated to each pair of endpoints no matter whether the link is used or not.

In broadcast networks architecture, all endpoints shares the same transmission media and a result, when one party transmits, all other parties can receive. However, only one party can transmit at any time instant, otherwise, collisions occur. To avoid such collisions, a medium access control (MAC) protocol is required to coordinate the transmissions on the networks. An important property of broadcast shared medium networks is that each endpoint transmits on the network at the same bandwidth equal to capacity of the broadcast shared medium.

1.1.2 Network Protocol Architecture

Network communication protocol is a set of rules for exchanging information among the communicating parties. It specifies the way of representing the information, controlling the information flow and starting and stopping communication sessions in the network etc. Protocol architecture is a framework into which protocol functionality is made to fit. Such architecture has been of great value in reducing the conceptual complexity inherent in the end-to-end communication task. Most protocol architectures are based on the concept of layering. In such architecture, an end-to-end communication task is accomplished bay successively and incrementally "adding value" in each a protocol layer.

A layered architecture can be regarded as a hierarchy of nested modules. Each given layer in the hierarchy regards the next lower layer as one or more black boxes, which provides specified service to the given higher layer.

The open system interconnection (OSI) reference model (Figure 1) is the most prominent layered architecture. The OSI reference model was developed as an international standard for data networks by the International Standards Organization (ISO) in 1978.

The OSI model divides the communications process into seven layers with each layer wrapping the lower layers and isolating them from the higher layer. Layering thus divides the total communications problem into smaller functions. It also ensures independence of each layer by defining services provided by layer to the next layer, independent of how these services are performed.

The lowest layer (layer 1) in the OSI model is the physical layer. It provides the physical medium for the information flow. In other words, it covers the physical interface between the devices and is concerned with transmitting raw bits over the communications channel. Thus, it is responsible for activating, maintaining, and deactivating the physical circuit between the sender and the informing layer 2 of the loss of a physical connection or electrical power.

The layer 2 is the data link layer. This layer is responsible for converting the unreliable bits pipe offered by the physical layer into a high-level communication link that appears free of transmission errors to the next higher layer, the network layer. Data link layer breaks the upper layer data into frames, then transmits those frames sequentially. In case of presence of transmission errors, retransmission may be performed. In a shared broadcast network, An extra set of functionality is needed to coordinate the access to the shared media, which is called Medium-Access Control (MAC) layer. MAC is considered as a sub-layer of the data link layer.

The third layer is the network layer. The purpose of the network layer is to provide the functional and procedural means to set up and terminate a session, to route data, and to control data flow across the network. The network layer module uses the destination information in the upper layer data, along with its own stored information, to generate the packet header in accordance with the protocol between peer network modules. An exception for network layer is that it can generate its own control packet. These control packets are used to create or tear down a session, exchange route information and inform link congestion and failure etc. To connect sub-networks, on the top part of network layer is sometimes called internet sublayer. Several subnets are connected through special nodes called gateways. The internet sublayer module handles packets forwarding across subnets.

The next higher layer is transport layer (layer 4). Transport layer provides transparent transfer of data between communication endpoints, so it is end-to-end layer. It performs a number of functions. First, it breaks the upper layer messages into packets at the transmitting end and reassembles the packet into messages. Transport layer may multiplex several sessions, which are from the same source to destination, into one session at network layer. Transport layer may require ensuring reliable data transmission and end-to-end flow control.

The session layer is the fifth layer in the OSI model hierarchy. It handles the interactions two endpoints in setting up a session and provides the means for cooperating presentation entities to organize and synchronize their dialog and manage their data exchanges.

The main functions of presentation layer (layer 6) are provision of a mechanism for presenting data in a manner that can be understood by both the sending and receiving application processes or devices. It also performs data encryption, data compression etc.

The highest layer (layer 7) is the application layer, which directly serve the end user. Each application requires its own software module. Examples are FTP, Telnet etc.

The Figure 1 shows the OSI model architecture. The data flow line describes the path of data traveling from application layer at host A to application layer at host B. Note that the two nodes in the middle do not have all of the protocol layers. This is because the functionality of intermediate nodes are to forward the data the right next node leading to the final destination of the data. This is exactly the network layer function, so only the lower 3 layers with network layer at the top are needed to perform such a task.

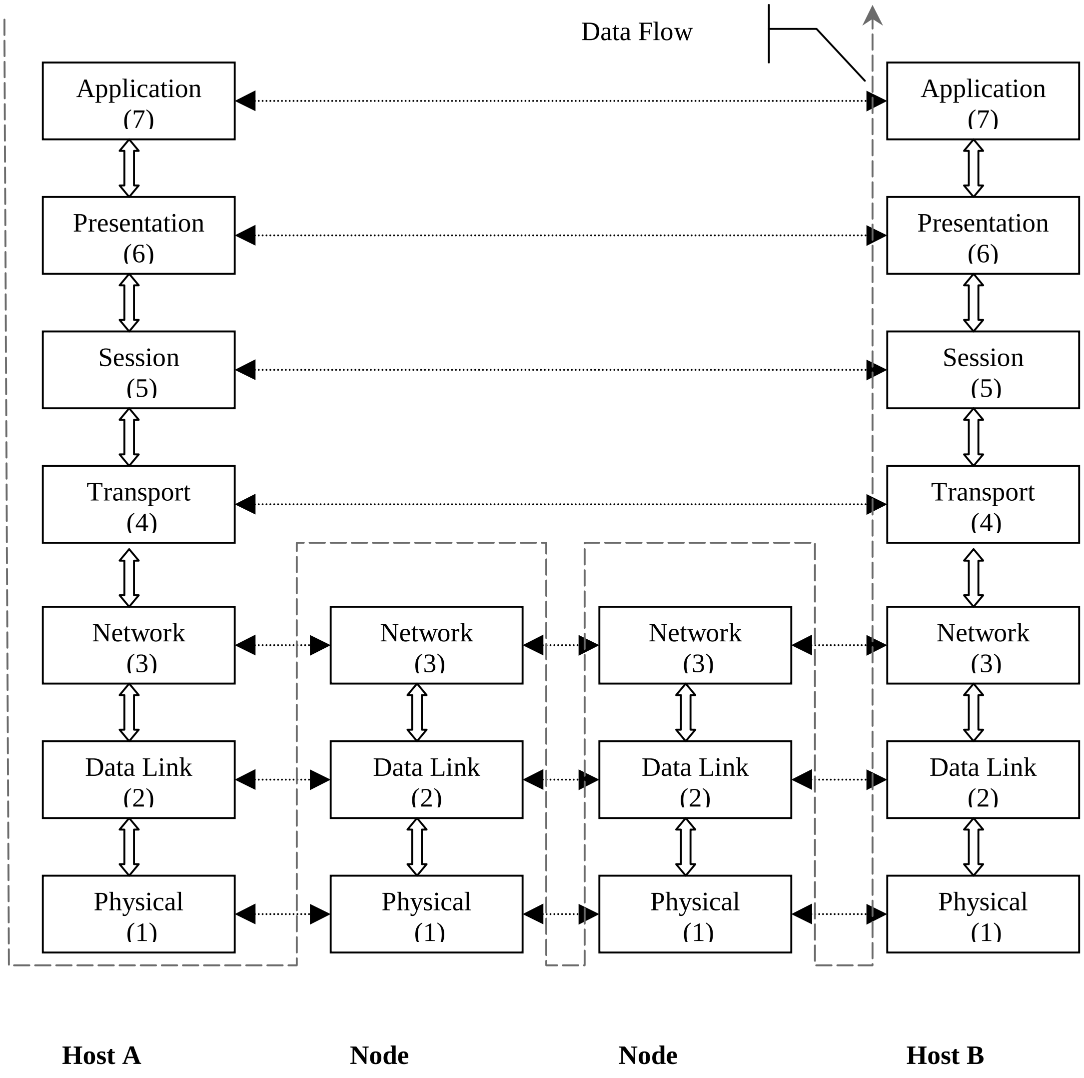

Figure 1. The Seven-Layer OSI Reference Model

1.2 Popular Network Protocols

Varieties of data networks are based on different underlying technologies and support different types of applications. In general, packet switched network can be categorized into two, i.e. connection-less and connection-oriented networks.

1.2.1 TCP/IP

The primary example of connection-less is the very popular IP-based Internet. The packet switches in the Internet that forward and route packets are commonly known as routers or gateways. The packet in the Internet is called an IP (Internet Protocol) packet. The IP packet header contains both its destination and source IP address. The router uses the destination IP address to select the route and forward each IP packet to the next hop that would lead to its final destination. So far, IP routers in general provide the same level of service to all packets – best effort service. Hence, there is no guarantee for bandwidth, delay or loss rates in the Internet. Furthermore, since it is a connections-less packet switched network, packets between two communication parties of the same session can travel along different paths in general. Therefore, packets can arrive out-of-order and there is no guarantee of reliable data receipts either. However, an upper layer (transport layer) module, specifically Transmission Control Protocol (TCP), is needed for those applications that requires the orderly and reliable-reception of its packets. Due to lack of delay and bandwidth assurance, currently the Internet is mainly used for email, file transfer and information downloading like web browsing. However, currently, there is tremendous industry push for IP carrying real-time data like voice. This can be realized by adding Quality-Of-Service (QoS) provision in IP protocol.

1.2.2 ATM

The primary example of connection-oriented packet-switched network protocol is Asynchronous Transfer Mode (ATM). The term "ATM" was coined to contrast it with synchronous transfer mode (STM) which is time-division based multiplexing. In ATM, each time slot carries exactly one ATM cell. Time slots are first allocated to those connections to satisfy their QoS requirements. ATM is a connection oriented, packet switching network protocol. ATM supports multiple class of services including data and real-time applications and provides guarantees for bandwidth, delay jitters and packet loss ratio. It has been adopted in Broadband Integrated Service Network (B-ISDN) that is designed to carry data, voice, image and video. In an ATM network, when a user requests a connection, the user provides the network with traffic description and QoS parameters (QoS parameters include maximum delay, maximum cell loss rate, etc). The network invokes its admission control algorithm and decides if the network has adequate resources to satisfy the requirements of the connection without degrading current existing connections, so network resources over-commitment is avoided. As a result, ATM can guarantee QoS to various traffic sources. This is one of the most attractive features that distinguishes ATM from other protocols and is extremely important for real-time traffic like voice and video since they are very sensitive to delay and cell loss rate. Specifically, ATM supports four classes of services [BERT92]: (a). Constant-bit-rate traffic (CBR). Examples of this are 64 kbit/sec voice, fixed-rate video, and leased lines for private data network. (b). Variable-bit-rate (VBR) packetized data that must be delivered with fixed delay. Example of this are packetized voice or video; (c). Available-bit-rate (ABR) traffic of connection-oriented. Examples of this include all the conventional applications of data networks. (d). Unspecified-bit-rate connectionless data (UBR) similar to UDP datagram service. Examples include the conventional applications of data networks in cases where no connection is set up.

In an ATM network, the data is divided into small, fixed length units called cells. The cell is 53 bytes long. Each cell contains a 5-byte header and 48-byte the actual data (payload). The header carries identification, control priority and routing information. At the end of the cell header is an 8-bit CRC checking code for the cell header. This error checking code has the capability of detecting errors and correcting a single error in the header. The error correction function is effective only when the transmission media has a relatively low error rate and the error pattern tends not to be in burst like optical fiber. ATM does not perform link-by-link payload data error control to reduce the processing load among the intermediate switch node. It passes such error control to the endpoints that may handle the error data differently depending on the requirements of applications.

1.3 Wireless Data Networks and Wireless ATM

The fast growing demand on cellular phone and laptop PCs in both consumer and commercial market proved that users are willing to pay significant value on portability which brings much convenience to our daily life. Many of us are beginning to depend on mobile computers and wireless modems of various form and function. Mobile data networks and wireless LANs empower these devices by connecting them to each other and to the information we need wirelessly.

Wireless data networks can be categorized into two types. Wireless Wide-Area Networks (WAN) and wireless Local Area Networks (LAN). The examples of wireless WAN include packet networks such as RAM/Mobitex and ARDIS/Modacom, paging networks, data over cellular, and data over satellite channels. Wireless LANs includes Industrial Scientific and Medical band (ISM band) LANs, Infrared LANs and "Unlicensed PCS" LANs. Wireless LANs cover a smaller geographic area than wireless WAN does, but provides much higher bandwidth. Currently, Wireless LAN products are based on spread spectrum technique, both frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS). Several MAC alternatives are possible for handling the special requirements of Wireless LANs. Power management is critical and may require special provision in the MAC protocol for mobile applications. Interconnection with backbone networks allows roaming within a campus environment as will as in the wide area. The IEEE 802.11 wireless LAN standard provides asynchronous, connection-less service. This service is aimed at providing short response time. A time-bound, connection-oriented service was postponed to a later phase of the standard. The IEEE 802.11 protocol is contention-based, as result, it more or less is a wireless version of IEEE 802.3 plus some power management. It is fundamentally unsuitable to support any QoS provisioning.

ATM has been accepted as the standard transport protocol in B-ISDN that is designed to carry multimedia traffic. As the wireless network is being expanded rapidly in recent years, there are tremendous demands for the transparent integration of wireless ATM terminals into fixed ATM networks during the last a few of years. In 1995, ATM-Forum and ETSI have established special wireless ATM groups, which are currently investigating requirements and system architectures for a wireless extension of ATM networks. In general, the users of wireless ATM terminals request the same functionality and Quality of Service as users of wired terminals. Thus, the protocol stack at the ATM air interface has to behave similarly as a fixed ATM multiplexer. This virtual ATM multiplexer around the air interface has to coordinate the access to the shared radio resources in such a way that the QoS of all ATM service classes can be provisioned.

As discussed previously, terrestrial (wired) data networks like B-ISDN will support real-time connections for some multimedia applications, it is natural to extend those real-time connections to wireless network so that mobile devices are multimedia-capable. Wireless data networking has been advancing rapidly in recent years. Many new wireless network based products are emerging. For example, wireless email terminal, wireless voice/data terminal and laptop and PDAs with wireless networking, wireless LAN products etc.. However, wireless data communication is still in its infancy stage because most of available products in the market have low bandwidth and do not provide integrated service and do not support user level QoS. To support QoS based integrated service, improvements in system wide architecture will be needed. From network protocol side, ATM will be adopted as the basis for the transport architecture of next generation of wireless data network [GILM96b]. The reasons behind are that it facilitates the wireless network to transparently and seamlessly interface with terrestrial ATM-based B-ISDN network ATM provides flexible bandwidth allocation and service type selection for a range of applications and efficient multiplexing of traffic from bursty data/multimedia sources. For instance, the second-generation High Performance Radio Local Area Network (HIPERLAN) is ATM-based and provides transparent access to B-ISDN. A typical wireless ATM network integrated with B-ISDN network is given in Figure 1.

However, there are tons of iss\ues in wireless networking that become active R&D topics in many organizations worldwide since the concept of extending standard ATM protocol over wireless network interface was proposed in 1994 [RAYC94]. In mobile ATM area, there are issues like handoff control [ACHA96][RAMJ98], location management and routing/QoS control [CAMP96]. In radio ATM area, issues include multiple access control, data link control and wireless control and physical layers [GILM96a] [CAIN97]. While researchers have paid a lot of attentions on ATM extension to wireless network, Error control in lower layer in W-ATM has not been adequately investigated. We will further illustrate in the following chapter that error control in the impaired wireless environment is extremely important. This motivates us to propose some research in this area.

A typical network structure is shown in Figure 2. The core of a wired B-ISDN network consists of fixed ATM switches with high-speed transmission trunks linked together. The data exchanges between ATM switches are through ATM protocol. On the edge of B-ISDN cloud, some wired end system and mobile terminals are attached. The point-to-point and point-to-multipoint links between any terminals including both wired and mobile can be dynamically established and torn down on demand.

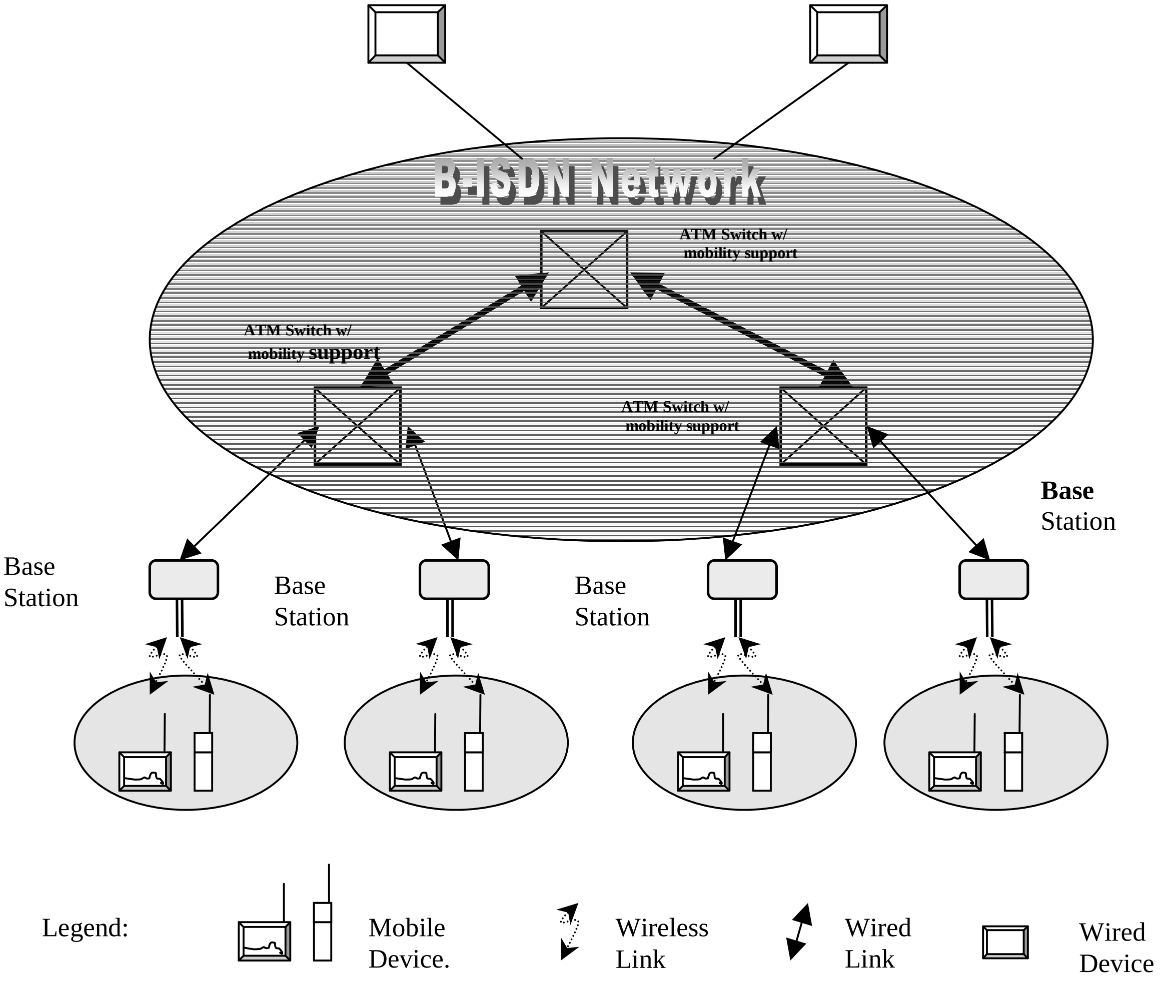

Figure 2. Wireless ATM Network Structure

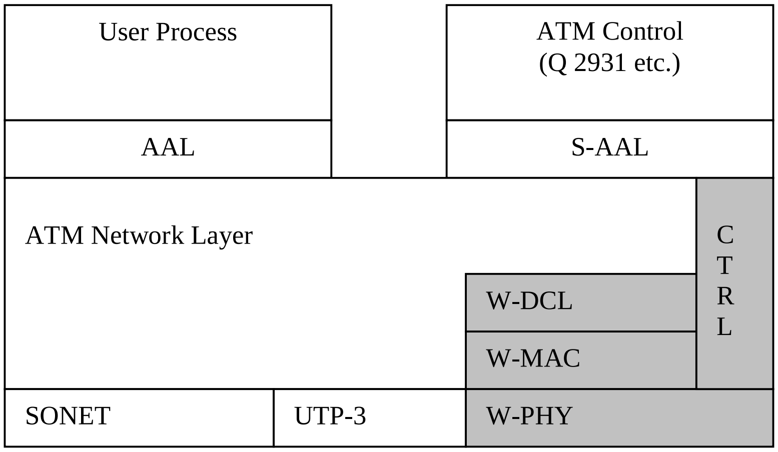

Figure 3. Air Interface protocol stack for Wireless ATM Network

1.4 Automatic Repeat Request Protocol (ARQ)

A communication system typically consists of a number of nodes that have both transmitting and receiving capabilities. Information flows from one node (source node) to another node (destination node) through links between the nodes. Information of large size is segmented into small pieces called data frames, which are transmitted sequentially. A data frame may be corrupted during the transmission process due to channel errors and transceiver errors. As a result, some error control mechanisms are needed to ensure that every data frame is successfully received with error free. Automatic Repeat Request (ARQ) schemes are the most commonly used schemes for error control and recovery in data communication systems. The schemes are simple and provide highly reliable data delivery between nodes. There are three basic types of ARQ schemes, namely, the Stop-and-Wait (SW), the Go-Back-N (GBN) and the Selective-Repeat (SR).

In a Stop-and-Wait ARQ data transmission system, the transmitter sends a data frame to the receiver and wait for an acknowledgment to that frame from the receiver (so called stop and wait). A positive acknowledgment (ACK) signals that the data frame has been successfully received, and transmitter sends the next new data frame. A negative acknowledgment (NAK) from the receiver indicates that the data frame has been detected in error, and the transmitter re-sends the data frame and waits for the acknowledgment again. The stop-and-wait ARQ scheme is simple to implement. However, it is inherently inefficient in bandwidth utilization and incurs extra delay due to the idle time spent waiting for an acknowledgment for each transmitted data frame.

In a go-back-N ARQ system, data frames are transmitted continuously. The transmitter does not wait for an acknowledgment after sending a data frame. As soon as it has completed sending one, it begins sending the next data frame. The acknowledgment for a data frame arrives at transmitter after a round trip delay. The round trip delay is defined as the time interval between the transmission of a data frame and the receipt of an acknowledgment for the data frame. During this interval, N-1 other data frames were transmitted. Here N is a protocol parameter. When a NAK is received, the transmitter stops sending new data frames. It backs up to the data frame that is negatively acknowledged and re-sends that frame and N-1 succeeding frames. At the receiver side, the N-1 received data frames following an erroneously received data frame are discarded regardless of whether or not they are error-free. Due to the continuous transmission and retransmission of data frames, the go-back-N ARQ is more effective than the stop-and-wait ARQ. Its throughput efficiency is high where the data rate is not too high and the round trip delay N is not too large. Another advantages is that the receiver buffer size can be small because the receiver insists on orderly receipt.

In a selective-repeat (SR) system, data frames are transmitted continuously. The difference from GBN is that when transmitter receives a NAK, it only re-sends the data frame that was unsuccessfully received. Receiver buffers those out of order frames according to the sequence numbers. Apparently, it is more efficient in bandwidth usage since it only retransmits those frames incorrectly received. But SR is more complicated and requires more buffering space on receiver side than GBN and SW.

In all of the three basic ARQ schemes, time out and retransmission are used to handle data frame loss and acknowledgment loss. The transmitter starts a timer immediately after the conclusion of a data frame transmission. Once this timer exceeds a threshold called the time-out value, the transmitter retransmits the data frame.

1.5 Forward Error Correction (FEC)

Shannon 's channel coding theorem opened the door of forwarding error correction world. We quote the theorem here as the following.

"*If we take increasingly long sequences of source digits and map them into correspondingly long transmission waveforms, the error rate in the data delivered can be brought arbitrarily close to zero, as long as we do not attempt to transmit data at a rate higher than channel capacity. Therefore, at any nonzero level of channel signal-to-noise ratio S/N, there is some nonzero information transfer rate below which arbitrarily accurate communication can in principle be achieved.*"

The essence of Shannon's result is that noise in the channel does not inherently limit the accuracy with which communication can be achieved, but only the rate at which information can be reliably transmitted. For a coded communication system, sequences of information bits are mapped into long codewords by the error-control encoder and then into long digital waveforms by the modulator. The demodulator and decoder then utilize all the received signal energy during the transmission of a codeword in the decision-making process. Several practical issues need to be dealt with, however. First, the coding theorem provides no means for constructing effective codes. Second, requirements of every low error probabilities will compel the use of very long codes, and this in turn will lead to very complete decoding operations. Because of these, much of the research in the coding field over the last forty decades has dealt with two key problems: finding classes of codes that yield good performance over wide ranges of lengths, and designing decoding algorithms that realize the intrinsic code performance without prohibitive complexity.

There are two important types of codes: Block codes and convolutional (or recurrent) codes. In block codes, a block of k data digits is encoded by a code word of n digits. For each sequence of k data digits, there is a distinct code word of n digits. The k data digits are accumulated and then encoded into an n-digit code word. In convolutional codes, the coded sequence of n digits depends not only on the k data digits but also on the previous N-1 data digits (N>1). Hence, the coded sequence for a certain k data digits is not unique but depends on N-1 earlier data digits. The coding is done on a continuous or running basis rather than by accumulating k data digits.

1.5.1 BCD Code

Bose-Chardhuri-Hocquenghem (BCD) codes are efficient multiple-error-correcting codes defined binary and non-binary symbol alphabets. BCD code is a type of block code family, specifically, a cyclic code. So, it can be defined in terms of its generator polynomial.

The primitive element of a finite field. An important property of finite fields is that every finite field GF(q) contains at least one primitive elements.

1.5.2 Reed-Solomon Code

RS code is a subset of non-binary BCD codes, obtained by choosing the locator field to be the same as the symbol field. When constructing an arbitrary cyclic code, there is no guarantee as to the resulting minimum distance.

Chapter 2: Issues and This Research

In this chapter, we first briefly summarize the research work having done previous in the area of error control in data communications. Then we highlight the characteristics of error control in wireless ATM networks and identifies the areas that need further research. Finally, we outline the research topics on which this dissertation will focus.

2.1 Previous work

There are two major approaches in error control in data communications: ARQ and FEC. ARQ has been an attractive research subject since 1970’s. Basically, the majority of the research falls into one of following categories.

- Modification of ARQ schemes to improve/optimize the throughput efficiency and channel utilization with channel error pattern and channel characteristics. [WELD82] [BRUN86][KIMS93] [YAOY95][CAMR97] [FANT97].
- Multiplexed ARQ performance analysis.[CAMR98][SAEK82]
- ARQ protocol and channel error modeling and performance evaluation. [LEET92] [CHOY94] [LEUN88] [MILL81][ZORZ96] [LUDL93]
- Analysis of queuing behavior and buffer requirement of ARQ schemes. [TOWS79B] [ANAG86]
- ARQ protocol parameter optimization.[MORR79]
- ARQ protocol stabilization. [AFEK93][GOUD85][SPIN97]
- Broadcast ARQ protocols. [AMMA92][GOPA84]
- Hybrid ARQ (with FEC).

ARQ protocol optimization has attracted a lot of research interest. In [JMOR79], the throughput efficiency was optimized with respect to the block length for traditional ARQ schemes. This is based on the observations that it is desirable to select the largest block length to minimize the time wasted in acknowledgments and associated delays. On the other hand, it is desirable to select the smallest possible block length to minimize the block error probability and to minimize the time wasted in block retransmissions. For a high error and long propagation delay environment, [DTOW79] described a modification to GBN scheme to improve the efficiency. During the periods of time the channel world normally be idle under GBN, the transmitter repeatedly sends the last unacknowledged message, if any, residing in the transmission queue. There were several variations of basic selective repeat schemes. [JMOR78], following [ASAS75], proposes to repeat blocks S times rather than just once when a NACK is received, where S is the number of blocks stored in the data link. This virtually eliminates buffer overflow, but because blocks are repeated many times performance suffers. For high-error conditions, [JLIN80] improved on Morris's scheme by having the transmitter repeat the requested block a number times and also repeat subsequent NACKed blocks. The same authors analyze a variation of SR in [PYU21] and low bound its throughput. [EWEL82] proposed a new strategy. The basic idea of the strategy is to repeat NACKed blocks multiple times with the number of repeats increasing as the receiver buffer approaches overflow.

There are recent research on wireless ATM systems [NAKA97] [CAIN97] [AYAN95]. In [NAKA], A stronger error detection/correction code, shortened BCH codes, are selected in replace of standard ATM HEC to better protect ATM header from errors. The author also discussed how to reduce false-synchronization using FEC. It was shown that the proposed approach can reduce the probability of mis-routing errors due to corruption of ATM header, however, the payload data also needs certain protection to support ATM real-time service models. E. Ayanoglu etc. proposed an algorithm to combat transmission error in multitone wireless network in [AYAN95]. The receiver iteratively requests that the transmitter retransmit additional parity bits to the receiver until the number of parity bits transmitted is sufficient to correct the data in the data in the erroneous data block. The algorithm does not address the maximum delay associated with each packet in a real-time stream. J. Cain studied the error control architecture for wireless ATM network [CAIN97]. The research focuses on architecture issues rather than applicable concrete error control schemes.

2.2 Issues

Previous research on ARQ has been done with application of conventional data service, i.e. they are designed to ensure error-free and reliable data delivery through retransmission mechanisms. For the research in protocol optimization, their objectives are exclusively the throughput efficiency. They are apparently not appropriate for W-ATM network based on the following observations.

- For real-time virtual connections in W-ATM, the network must provide bounded end-to-end delay guarantee. This is a network commitment to its service subscribers.
- Another important QoS parameter is the cell loss ratio, which must be satisfied to deliver promised service under service level agreement (SLAs).
- On the ATM network technology side, the mobility support on existing protocol structure needs to be addressed.
- Wireless transmission media is error-prone comparing with fiber optics. This brings a lot of issues in W-ATM. We shall elaborate them in the following sections.

One of the most important issues in wireless ATM is how wireless medium supports ATM protocol while keeping ATM API intact to permit network-based multimedia application developers to use a uniformed ATM API with end-to-end QoS control for both fixed and mobile service platforms. This requires the wireless ATM and physical layer provide transparent service equivalent to the wired ATM. There are several challenges with wireless networks due to the following characteristics of wireless networks.

2.2.1 Multi-access, Mobility Management and Handoff

In a wired ATM network, the links are point-to-point between the nodes. However, wireless mediums like radio are shared resource. Multi-access protocol is needed to coordinate the access to radio channels with the capability of supporting multi-service with dramatically different performance requirements.

Wired ATM protocol for connection setup assumes that a terminal's address implicitly identifies its attachment point ("location") to the network. However, with mobile terminals, the location of such a terminal may no longer be deduced from its endpoint address. Additional addressing schemes and protocols are needed to locate and track mobile terminals, along with suitable modifications to the connection setup process.

After a connection has been setup, the connection path in a network with fixed terminals does not change during the connections' lifetime (other than due to link/switch failures). This assumption is invalidated when the endpoint is mobiles. When a mobile terminal moves from a radio-port to another, connection needs to be re-routed and cell loss while re-routing the data path.

2.3 Issues in Error Control

As we described in the previous chapter, the ATM protocol was designed for high quality of transmission media such as fiber optics whose bit error rate (BER) is as low as 1-E9. As a result, the ATM does not provide any error detection operations on the user payload inside the cell, and also provides no retransmission services. However, even in indoor radio environment, the BER is typically up to 1-E3. ATM protocol will perform very poorly under such as a high error condition. The solution is to mitigate the error condition in lower layers. This requires lower layers, namely data link layer and physical layer, to dramatically improve those error conditions of radio links. There are two major approaches for error control and correction in the two lower layers: Forward Error Correction (FEC) and ARQ schemes. FEC employs error-correcting codes to combat bit errors by adding redundancy to information frames before they are transmitted. Receiver uses the redundancy to detect and correct errors. ARQ, as described previously, deals with errors through retransmission. A lot of research has been done in the area of ARQ schemes. Nearly all of them focus on conventional data services (non-real time) that are characterized as insensitivity to delay and best effort service. In W-ATM environment, the scenario is very different for the real time services like CBR and ABR services that provide guaranteed QoS. Among the major parameters of QoS are maximum cell delays and maximum cell loss ratio (CLR). How will these conventional ARQ schemes behave in W-ATM network? What are the problems? And how the problems can be solved?

We propose the following research topics**.**

2.3.1 Applicability and Feasibility of ARQ in WATM: Are ARQ protocols needed in real time wireless ATM environment.

All of current existing ARQ schemes are used in non real-time data transmissions. In those scenarios, the delays in the transmission process are not a concern (or more accurately, a requirement) to both ends, as long as the information frames are received correctly and orderly. Since W-ATM supports real time service like CBR and VBR traffic which have strong requirements of QoS. Apparently, ARQ can reduce cell loss ratio by retransmission, but retransmission introduces extra delays. This was one of primary reasons that ARQ was not incorporated into ATM protocol that was designed for B-ISDN over high quality physical links. So, what justify that wireless ATM of using ARQ schemes? There are several perceived justifications.

- Wireless media transmission error rate is quite high, bursty and time-varying. It is well known that most of FEC coding schemes do not work desirably in situations where the channel errors are bursty. The joined efforts of FEC and ARQ in physical and data link layer may be needed to mitigate media error condition to some level so that the upper ATM layer can achieve acceptable performance.
- For the concern that ARQ retransmissions introduce extra delays. As long as ARQ frames satisfy maximum delay parameter of QoS, the upper layer is absolutely beneficial from ARQ since it can reduce cell loss ratio (CLR). There is some leeway for retransmission in term of delay requirement. For instance, for 25Mbps rate, 1 ms is equivalent to the transmission time of over 100 ATM cells. As the bandwidth of W-ATM increase, the leeway will also increase.

2.3.2 Performance evaluation and optimization of conventional ARQ schemes in W-ATM environments.

So far, most of the research in evaluating the ARQ protocol performance has been done in conventional data service. The performance index includes only throughput efficiency. The real time traffic sources in W-ATM have delay and cell loss ratio requirement. It is practically important to analyze the protocol performance including frame loss ratio, delay and throughput efficiency in the W-ATM real time environment and compare the cell loss ratio and delay characteristics among the various ARQ schemes. The research will help identify the protocols and their variants that are more appropriate to W-ATM than others and what improvements could be achieved with those ARQ schemes.

Some researchers proposed several ARQ schemes in which the protocol parameters are adaptively adjusted according to current channel error conditions to optimize the protocol performance [BRUN86][YAOY95][WELD82]. However, nearly all of them restricted the throughput efficiency as the optimization objective. In W-ATM, the delay performance and cell loss ratio are just as much important as throughput performance and even more important for some service categories. In fact, they are the constraints of optimization for real-time service categories. As a result, a weighted performance index with delay as a constraint would be more appropriate for adaptive optimization in W-ATM.

2.3.3 Real Time ARQ. What ARQ schemes are appropriate for W-ATM?

Conventional ARQ protocol families were designed for reliable data delivery with no real time requirement in mind. For instance, if a data frame is received in error, the sender will retransmit it. If the data frame is received in error again, the sender will retransmit the data frame again. … In fact, the process will go on forever until the frame is received in error free. For an impairment data link like wireless, the number of retransmission for a data frame can be huge. When a data frame gets finally received in error free, it may be useless because the delay between the instant when the data frame arrived at sender and the instant when the data frame is successfully received is larger than maximum delay parameter of QoS of real time connections. The fundamental problems of the conventional ARQ protocols lie in the lack of delay control mechanism when used in real time environment like W-ATM. Delay control mechanism can ensure every ARQ frames arriving at the receiving side are within the frame's time to live (TTL) and no over-aged (The age of an ARQ frame is the time elapsed since the frame is submitted for transmission) frames are transmitted. Therefore, a real time ARQ must have the delay control mechanism through which ARQ is frame ages-aware and is capable of discarding those over-aged frames. So far, the author has not seen such a real-time ARQ scheme with both error control and delay control functionality on any literature. This motivates us to research on design of such a real time ARQ for W-ATM with following wish list.

- Capable of retransmitting an incorrectly received frame.
- Capable of tracking the ages of outstanding frames.
- Capable of discarding over-aged frames.
- Minimum protocol overhead.
- Minimum system resource requirement.
- Liveness and stability
- High performance.

2.3.4 Placement of ARQ scheme in W-ATM

An architecture issue is placement of ARQ schemes. Non-real time data link protocols use a single ARQ instance for all traffic sources of a link. The ARQ frames are delivered in order of their arrivals. For W-ATM, it would be problematic to differentiate the frames of different service classes that have different priorities and QoS requirement if we use a single ARQ instance for all channels of all service classes. What is the best strategy for placing ARQ schemes among different service classes and different channels within a service class? How to choose ARQ type for channels of each service class? Those are obvious questions to be answered during the course of our research.

2.3.5 Correctness of Real Time ARQ: Safety and Liveness Proof

While it is relatively easier to come up with a communication protocol, its proof of correctness can be very difficult. Proof of the correctness of a protocol's algorithm can be broken into two parts [BERT92], characterized as safety and liveness. An algorithm is safe if it never produces incorrect result. An algorithm is live if it can continue forever to produce results, i.e., if it can never enter a deadlock condition from which no further progress is possible.

2.3.6 Performance analysis and evaluation of the protocol.

After the protocol's correctness is proved, Its performance needs to be evaluated. There are several key criteria indices. Protocol overheads, throughput efficiency, delay performance and cell loss ratio etc. Markov chain can be employed to model error burstness of wireless media [COXD77]. Analytical expression of the key performance indices may be derived through renewal theory [ROSS85]. Another approach is to use signal flow methodology [LUDL93]. Section 3 will highlight those theories. Simulation will be used to validate the theoretical results.

2.3.7 Combination of ARQ and FEC at data link layer (hybrid ARQ).

Forward Error Correction (FEC) employs error-correcting codes to combat bit errors by adding redundancy to information frames before they are transmitted. Receiver uses the redundancy to detect and correct errors. Although the use of FEC decreases the frame error rates (as observed by the receiver, the increased number of parity bits) decreases throughput. If the bit error rates are sufficiently small then ARQ based error control schemes can provide throughputs that are significantly larger than those the can be achieved when FEC is used. If the bit error rates are sufficiently high then the increased frequency of retransmissions can result in a significant waste of transmission capacity. Conceivably, the combination of FEC and ARQ can achieve a better performance over wide range of error conditions. Some degree of adaptation to link error condition could further enhance the performance.

Besides, since wireless link error tends to be bursty and most of FEC algorithm work undesirably under bursty error, it is important to choose appropriate FEC algorithm. On the other hand, channel error can be randomized through some intentional measures like channel interleaving.

In case that high error rate and long round trip delay, FEC may have advantage of less delay than ARQ.

2.3.8 Slot scheduler and ARQ schemes.

W-ATM network has a star topology. Due to the sharing nature of wireless media, time can be divided into equal length of small segments called time slots. Each slot is assigned to a pair of nodes. During the slot time, only the two nodes can exchange information. Slot scheduler decides which slot is assigned to the nodes in the network based on some scheduling strategy (e.g. round robin). This multi-access scheme is called Time Division Multi-access (TDMA). TDMA is deterministic in the sense that each slot is deterministically associated with the specific nodes. There are some contention-based multi-accesses schemes like IEEE 802.11. It is conceivable that they are inappropriate for time-bounded connections in W-ATM since it is very difficult to guarantee delays. TDMA seems to be the right choice for W-ATM. It is easy to see that the slot scheduler will play a key role, and scheduling strategy is very important to fulfill QoS commitment to real time traffic connections. Some research has been done in this area. [CHAN97] [STAM97], but channel error conditions are not taken into account in their strategies. This is partially because there is no error information easily available to slot scheduler. In our proposed architecture, ARQ schemes can provide slot scheduler with handy error information for each slot. Conceivably, scheduler can make full use of this information to adaptively make or adjust its scheduling.

2.3.9 Impact of service class mix to QoS guarantee in time-vary channel

Priority of service classes can be from high to low in the order of CBR, VBR, ABR, and UBR. As part of admission control, the relationship between channel error condition and various mix of CBR and VBR traffic with ABR and UBR traffic.

2.3.10 Multi-Access Control (MAC) protocols.

Wireless media is a shared resource in nature. With multiple mobile terminals share a same radio frequency, a MAC is needed to coordinate those competing terminals. The major challenges for designing such a MAC for wireless ATM are high throughput, low delay, ability to support handoff/roaming between service areas, fairness, power consumption, ability to support asymmetrical traffic and maximum number of nodes.

2.4 Outline of Research of this Dissertation

We will focus two of most important issues in wireless ATM network in this dissertation. Adaptive coding and real-time ARQ protocol. It is believed that FEC code rate must adapt to the channel error conditions to achieve a better performance due to the time and space varying nature of wireless links, and at the same time, the QoS of real-time connections are satisfied. We will use Reed-Solomon code with different code rates combining with selective and repeat ARQ scheme to satisfy the QoS requirements of wide-range of real-time applications in a wireless ATM environment. The algorithm will be defined and evaluated.

In real-time communications, retransmission schemes are attacked for their introduced delays, as a result, some researchers are in favor of FEC approach. I would argue that ARQ protocol has significant value in real-time applications in wireless ATM networks based on the following observations.

- ARQ techniques provide a way to trade off delay jitters for packet loss rate. In contrast, FEC schemes typically allow for the tradeoff between data rate and packet loss rate. The jitters/error tradeoff is at the heart of managing packet-switched streaming service such as voice and video.
- In a virtual circuit oriented network like ATM, ARQ makes it easier to implement differential services policies for each virtual connection. One-time protocol parameter negotiation per connection can realize different packet level treatment for each connection. FEC unequal error protection for each connection will be more computationally intensive and energy consuming.
- ARQ can potentially effectively discard packet (no such a scheme exists yet). For example, in a wired network, the control policy for packet discarding might allow congested interior nodes to selectively discard packets from certain layers of a layer-encoded image. Such a policy is hard to carry over to a wireless link where errors due to outages are not controlled. ARQ can regain control by selectively retransmit some of the packet while allowing other to expire.

We will create an ARQ protocol, which can handle packet discarding and manage the packet delay. We call it as real-time ARQ or rt-ARQ in contrast with conventional ARQ protocol that are designed for reliable data transmission for conventional data service. Its performance and liveness will be evaluated through simulation.

Chapter 3: Adaptive Hybrid ARQ in Real Time Communications

In this chapter, we investigate adaptation of FEC code rate, specifically Reed-Solomon code, to channel error conditions in a real time communication environment like wireless ATM network. In particular, the protocol throughput performance of error control system is optimized through adaptation of FEC code rate to channel error condition, and at the same time, the delay and cell loss ratio of real time data are also satisfied. The algorithm allows retransmission where possible and honors maximum delay constraint. Reed-Solomon code rate is adaptively selected so that (i). Packet loss probability are within cell loss ratio specified by user QoS parameter, (ii). Packet is delivered to the destination within the maximum delay specified by user QoS parameter. (iii). Protocol performance is optimized without compromising delay and cell loss ratio.

3.1 System Model

3.1.1 The Throughput Performance of ARQ Protocol

Selective Repeat is the most efficient of three basic ARQ schemes. The throughput efficiency is given by

$$\eta = R(1 - P_{pe})$$

Where R is FEC code rate and P_{pe} is the probability of an ARQ packet with uncorrectable error(s).

3.1.2 Reed-Solomon Error Correction performance model

Assume the elements in the code-word symbol alphabet $GF(q^m)$ are to be transmitted, using an q^b-ary constellation of channel symbols. The channel symbols are transmitted across a symmetrical and memory-less channel with probability of channel symbol error of P_{ce}.

The upper bound of the probability of uncorrectable error occurrence may be computed as [WICK95]

$$P_{pe} \le 1 - \sum_{j=0}^{\lfloor (d_{\min}-1)/2 \rfloor} \binom{n}{j} p_{se}^{j} (1-p_{se})^{n-j}$$

Where, the probability of FEC code symbol error $P_{se} = 1-(1-p_{ce})^{m/b}$ and $d_{\min}$ is the minimum distance. For Maximum Distance Separable (MDS) code (n, k) like Reed-Solomon code,

$$d_{\min} = n-k+1$$

Without the loss of generality, let $q = 2$ (Zierler code). So $n = 2^{m} - 1$

3.1.3 Model of Quality of Service.

There are two major constraints associated with real time data: i.e. maximum delay D and tolerable data loss ratio ε. These are the constraints that must be satisfied by the adaptation of FEC code rate to the channel error condition. Hence, it comes to a need to incorporate the two QoS parameters into the adaptation process. The maximum delay can be mapped to the maximum number of possible transmissions (N) through the round trip delay.

In a real time ARQ scheme, the transmitter stop re-transmitting an ARQ in the following two situations.

A). The residual life of the packet is not long enough for another transmission.

B). The receiver has positively acknowledged the reception of the packet. Packet loss could occur in both situations.

The probability of a block is received at the nth transmissions is

$$P(n) = (1-P_{pe})P_{pe}^{n-1}$$

The probability of packet loss P_d in case A can be quantified as.

$$P_d = P(n > N) = 1 - \sum_{n=1}^{N} (1 - P_{pe}) P_{pe}^{n-1}$$

Where N is the maximum number of transmissions allowed for a data block bounded by delay.

In case B, a packet accepted by the receiver might contain the error(s) that are not detected by FEC code. From the point of view of application, it can be considered as a packet loss. For Reed-Solomon code, the undetected error probability of per packet transmission is given by

$$P_{u1} \le 1 - \left[\sum_{j=0}^{d_{\min}-1} \binom{n}{j} P_{se}^{j} (1 - P_{se})^{n-j} \right]$$

A packet may be transmitted multiple times bounded by N. the average number of transmissions per packet is $\frac{1}{1 - P_{pe}}$. So the probability of a packet received with undetected error is

$$P_u = 1 - (1 - P_{u1})^a$$

Where $a = \min(N, \frac{1}{1 - p_{pe}})$

It is easy to see that the two sources of packet loss are mutual exclusive and in either case, one packet will be lost, therefore, the packet loss ratio constraint can be expressed as

$$P_{CLR} = 1 - (1 - P_u)(1 - P_d) \le \varepsilon$$

The problem of our adaptation is that, for any measured channel symbol error, find an optimal Reed-Solomon code rate so that the throughput efficiency is maximized while the requirement of data packet delay and packet loss ratio is satisfied.

3.1.4 Problem Formulation

It can be formulated as a convex program

$$\begin{cases} Maxmize\ f(x)\ subject\ to \\ \qquad g(x) \le \varepsilon \end{cases}$$

The detailed form is as follows.

$$\begin{cases} Maximize\ \dfrac{k}{n}(1-P_{pe}),\ subject\, to \\ 1-\sum_{i=1}^{N}(1-P_{pe})P_{pe}^{i-1}\left(\sum_{j=0}^{d_{\min}-1}\binom{n}{j}P_{se}^{j}(1-P_{se})^{n-j}\right)^{\min(N,\frac{1}{1-P_{pe}})} \le \varepsilon \\ where\ \ P_{pe}=1-\sum_{j=0}^{\lfloor (d_{\min}-1)/2\rfloor}\binom{n}{j}P_{se}^{j}(1-P_{se})^{n-j}\ \ and\ \ N=D/RTD. \end{cases}$$

While it is difficult to obtain an analytical solution in closed-form, using nonlinear programming optimization approaches [PERE88] due to the complexity of f(x) and g(x), such a close form may not have too much practical meanings since it would computationally be too complicated to implement on a real wireless communication systems that usually have the limited computation resource and power.

Next, we shall evaluate the performance of Reed-Solomon code in real-time communications, then we’ll derive a look-up table consisting of a series of Reed-Solomon codes with the indices of maximum delay, maximum packet loss ratio and channel error probability

3.2 Performance Analysis

In this section, we analyze the behavior of various RS codes in a real time communication environment

3.2.1 Throughput Performance versus SR Code Rate.

The throughput efficiency of three ARQ encode schemes are given in Figure 4. When error rate is lower, the pure ARQ has higher throughput efficiency than those of the FEC encoded ARQ. This is intuitive since errors rarely occur, the redundancy introduced by FEC is a waste of bandwidth because there is rare error to correct. As the error rate increases, more and more ARQ frames are received with error. In the pure ARQ case, a single error in an ARQ block makes the whole block useless because there is no error-correcting capability. As a result, the throughput efficiency of pure ARQ degrades logarithmically. In case of the RS-coded ARQ, the maximum achievable throughput efficiency is dictated by the RS code rate. This maximum throughput efficiency can be maintained as long as the channel error rate is low enough so that the number of erroneous bits in an ARQ block is less than the maximum number of errors that FEC is capable of correcting. As the channel error rate increases, the lower code rate SR shows advantages over high code rate SR due to its redundancy. It can be seen in Figure 4 that there is no an ARQ which provides best throughput efficiency throughout all channel error conditions. In other words, each ARQ has a higher throughput performance than the others only in certain range of error conditions. As mentioned previously in this paper, the channel conditions in wireless communication system are violently fluctuating. Therefore, it is necessary to adapt SR code rate to channel error conditions to improve throughput efficiency of ARQ protocols.

3.2.2 QoS Cell Loss Ratio (CLR) Performance versus RS Code Rate.

Throughput efficiency is the sole optimization objective of ARQ protocols in conventional data service. However, in real time communications, data has to be received correctly by destination node within the maximum delay and data loss ratio (QoS parameters) imposed by applications. It is meaningful to achieve higher throughput performance only under the condition of the QoS requirements being satisfied. Hence, there is a need to incorporate the user QoS requirements into the optimization process of throughput efficiency. From the Figure 5, we can see that the pure ARQ suffers the highest cell loss ratio among the three schemes. As FEC code rate increases, the cell loss ratio decreases due to its increasing error correcting capability. For a typical cell loss ratio of 1e-5 for real-time communications, the pure ARQ is not an appropriate choice even it is attractive in term of throughput efficiency in certain error conditions.

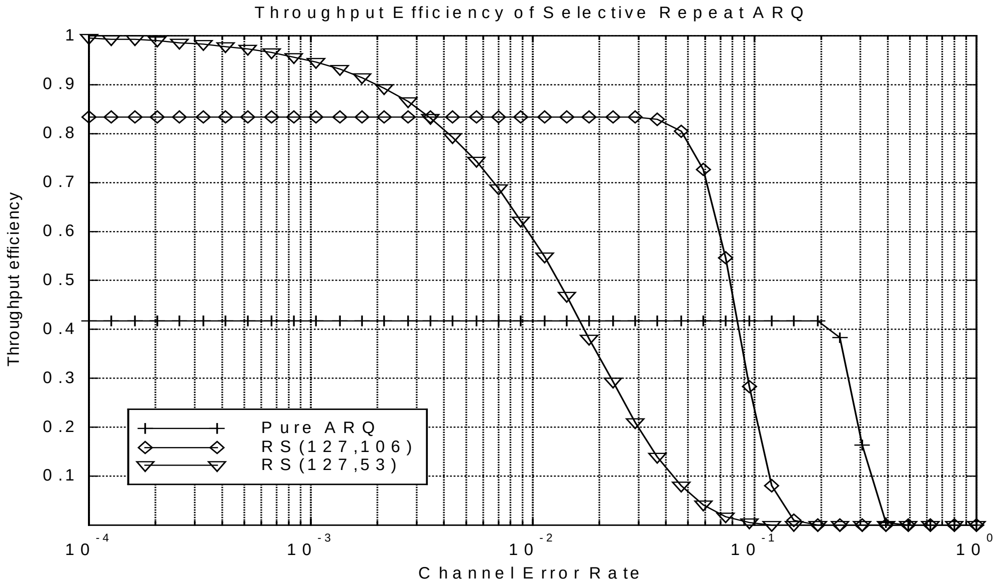

Figure 4. Throughput efficiency of selective ARQ with and without FEC .

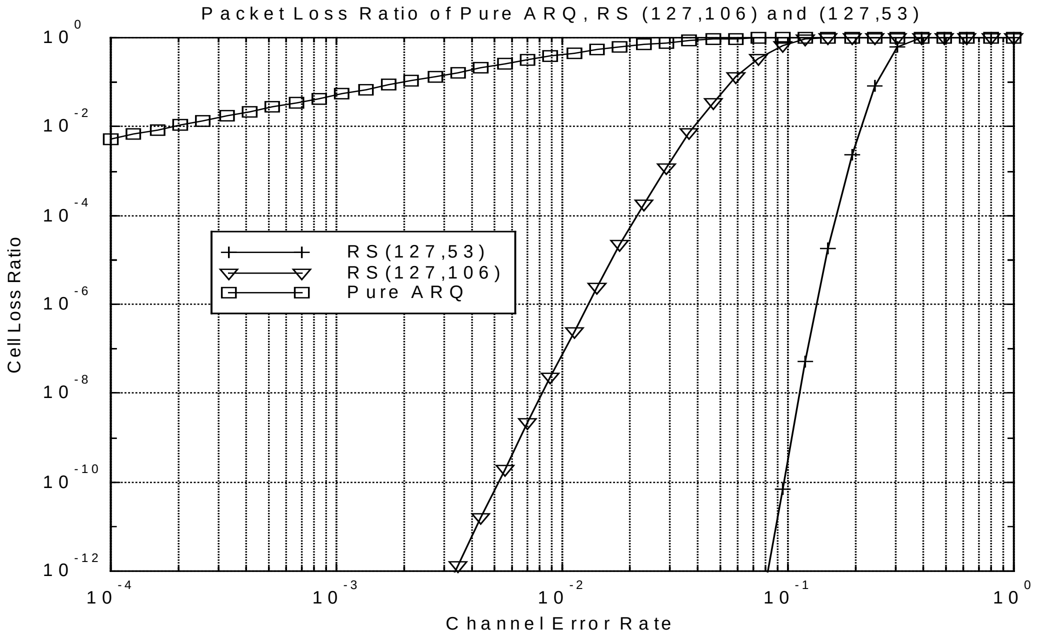

Figure 5. Cell Loss Ratio versus Channel Error Conditions

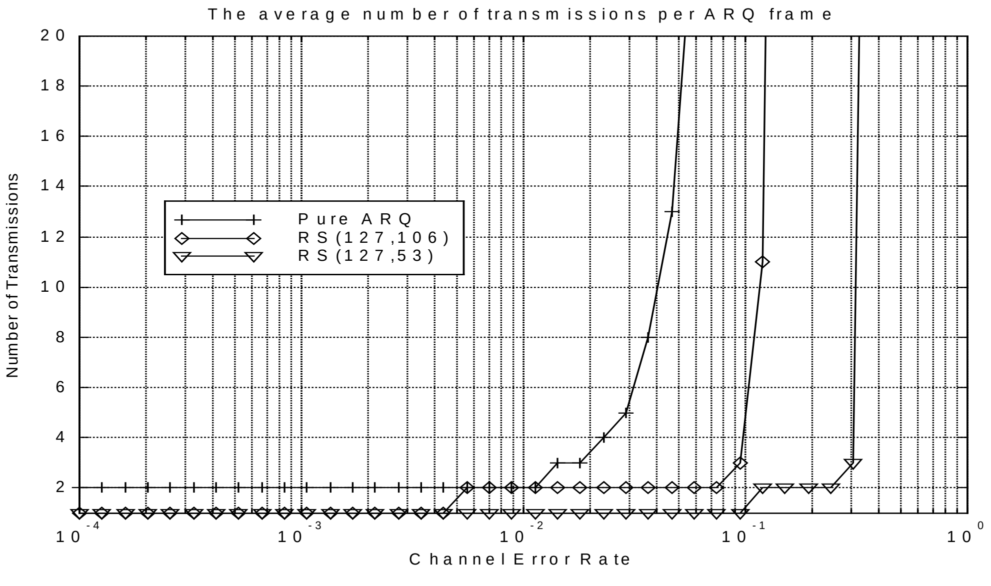

Figure 6. Cell Loss Ratio versus Channel Error Conditions

3.2.3 Delay Performance versus RS Code Rate.

RS code rate has a significant direct impact on the cell delay performance. The Figure 6 gives the average number of transmissions per cell of three ARQs under various error conditions. The pure ARQ suffers more delays than the other two. The average delay increases very sharply when the channel error rate exceeds certain values (break points) for the FEC coded ARQ cases while it is quite stable before the break points. The Figure 6 also shows that the larger delays allow more choices of different code rates and then more likely higher throughput efficiency can be achieved.

3.2.4 Delay Requirement versus Cell Loss Ratio and RS Code Rate.

Intuitively, the retransmissions can reduce cell loss probability, but to what extent? The

Figure 7 gives the visual answer to that. When only one retransmission is possible, the pure ARQ is not usable for most real time communications with CLR of 1e-5 even that the pure ARQ has a higher throughput performance in certain lower error conditions. With the maximum delay equivalent to 5 transmissions, the pure ARQ can satisfy the CLR of 1e-5 with the channel error rate lower than 2e-3. Within that error range, the pure ARQ has a throughput efficiency of over 0.9 that is higher than the other two ARQs.

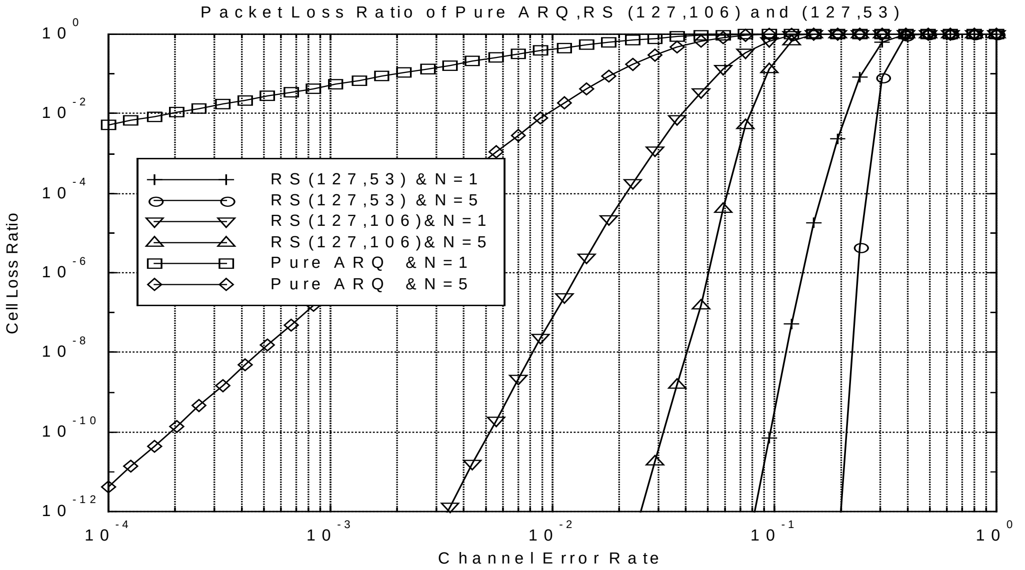

Figure 7. Cell Loss Ratio with different Code Rate

3.3 Adaptive Hybrid ARQ

In this section, we'll design an adaptive algorithm for the ARQ protocol that can be applied to wireless ATM network. Without loss of generality, a set of link layer and MAC layer protocol parameters are chosen to demonstrate the effectiveness of the adaptive algorithm. The adaptive ARQ protocol supports user level QoS and can optimize throughput efficiency in a wide range of channel error conditions. The simulation results are given at the end.

3.3.1 Adaptive ARQ Protocol Packet Format.

ATM cells are 53 bytes long, consisting of 48 byte payload data and 5-byte header. The header is composed of 28-bit address (VPI+VCI), 8-bit CRC checksum, 2-bit payload type, a priority bit and a reserved bit. With 28-bit address space, the ATM cell format allows for over 268 millions sessions to share a link. This is much more than enough for wireless ATM environment, considering that a base station can only cover a limited territory (micro-cell) and the number of mobile terminals has to be limited by the fact that all mobile terminals in a micro-cell share the limited bandwidth of the base station. A 12-bit address field, allowing 9600 sessions, should be enough for wireless micro-cells. Accordingly, HEC field is reduced into 4 bit to provide error correction and detection capability equivalent to that of regular ATM cells. Four bit ARQ sequence number is added. As a result, the ARQ packet has a header of 3 byte and 51 bytes in total as described as in Figure 8.

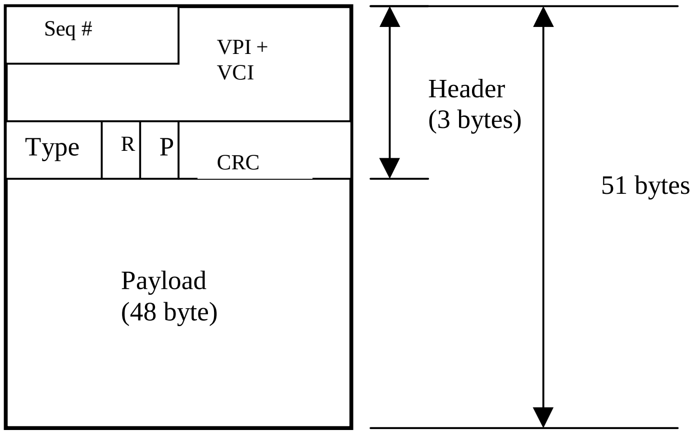

Figure 8. Adaptive ARQ Packet Format.

3.3.2 MAC Protocol and ARQ Packet Framing.

There are two basic types of multi-access protocols for wireless network, random access like IEEE802.10 and controlled access like TDMA. Due to the difficulties in scheduling and controlling delays with random access protocols, the controlled multi-access protocol of TDMA type seems more favorable for real time communications. In a TDMA-based network, the frame size is fixed.

3.3.3 Selection of SR Codes for Adaptation.

There are several criteria for selecting SR codes.

- An FEC code word needs to fit in a MAC frame of fixed-size to minimize protocol conversion overhead.
- SR code Goalie field of power of 2 facilitates the transmission of binary numbers.
- A code word can hold the integral number of ARQ packets to avoid any bandwidth waste.

- Certain variety of SR codes of different redundancy is provided for adaptation. Too few codes to select will limit the efficiency of adaptations. Too many codes will cause too often switching among codes as channel condition fluctuates violently.

The fixed code word length of 255 bytes satisfies the criteria 1, 2 and 3 above. The set of SR codes consists of (255, 51), (255,102), (255, 153), (255, 204) and (255, 255). The last one corresponds to pure ARQ. Without loss of generality, let's assume that a TDMA frame carries one code word. Since one code word can contain 1 to 5 ARQ packets, TDMA frame header contains the information about SR code type and the number of ARQ packets contained in the frame.

Two bits: number of packets in the frame.

Three bits: FEC code types.

One byte TDMA frame header.

3.3.4 Adaptation Algorithm

Mobile wireless devices are characterized as low power, light-weighted, small-sized device. These characteristics require the FEC adaptation algorithm performed on mobile devices must be simple and efficient in computing.

The adaptation algorithm is outlined in three steps. The result is a small array of switching points through which a RS code is adaptively selected so that the best throughput can be gained with the QoS requirements are satisfied.

Step 1: The SR code rate switch tables are pre-computed with the cell loss ratio and maximum number of retransmissions as index. The switch table contains the code rate and channel error rate mapping can be stored into a ROM chip for mobile device use. A computer program is developed to generate such a table. The table size can be controlled through choice of error resolutions. This step is done on a desktop PC.

Step 2: During the channel connection time, a connection-specific switch table is generated using the QoS parameters provided by applications by pulling out related information from the switch table generated in step 1. The result is a few switching points between the five SR codes and pure ARQ. Note that this only needs to do once for the whole lifetime of a connection. The computation complexity is linear to the table size and equal among all connections with different QoS requirements.

Step 3: Based on the switching points from step 2 and current channel error conditions, the decision of which RS code to use is made for next MAC frame. So, only minimal switching point-lookup is involved in the normal operation process.

3.3.5 Performance Evaluation

The performance of the adaptive ARQ scheme is evaluated and compared with some fixed SR-coded and pure ARQ schemes.

The Figure 9 to Figure 14 are associated with the QoS parameter of the cell loss ratio 1e-05 and delay of one transmission. The Figure 9 gives the throughput efficiency of our adaptive real-time ARQ scheme. The discontinuities are caused from the RS code switching as the channel error conditions change indicated in the Figure 10. The Figure 11 shows the effectiveness of the adaptive algorithm in satisfying user's cell loss ratio QoS requirement in a wide range of channel error conditions as high as 0.27736. The throughput performance gain over the pure ARQ and RS (255,51) coded ARQ are plotted in the Figure 12. It shows the significant throughput efficiency gain over the RS (255,51) coded ARQ in various channel conditions. In comparison to the pure ARQ, the throughput efficiency in the low error conditions is less. This is because the pure ARQ can not satisfy the QoS requirement in those channel conditions, which are shown in the Figure 13. This also demonstrates that throughput compromise needs to be made in a real-time communications in which users pay more values to get higher QoS service. The Figure 13 also shows that the adaptive ARQ apparently outperforms the pure ARQ in higher channel error conditions in throughput performance.

With delay of five transmission and the same cell loss ratio of 1e-5 as above, the adaptive real-time ARQ has a higher throughput performance as shown in the Figure 14.

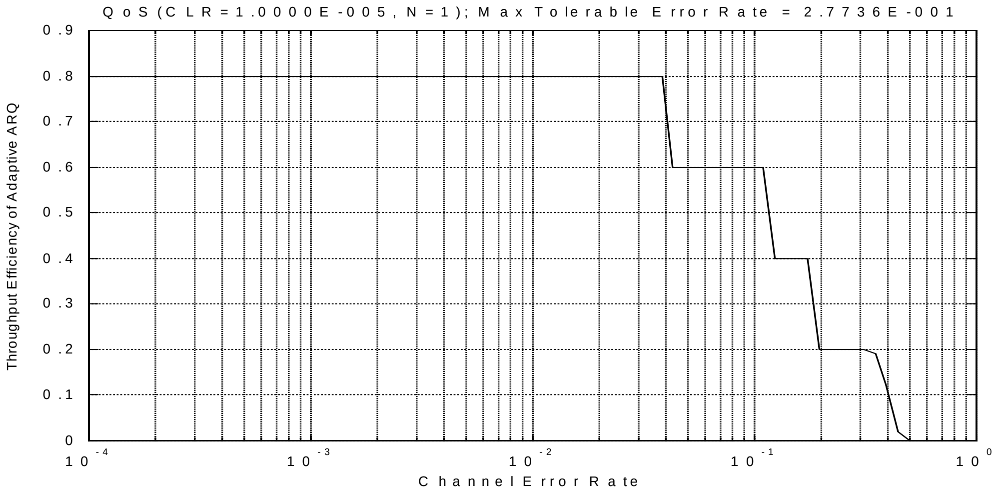

Figure 9. Throughput Efficiency of the Adaptive Real-time ARQ

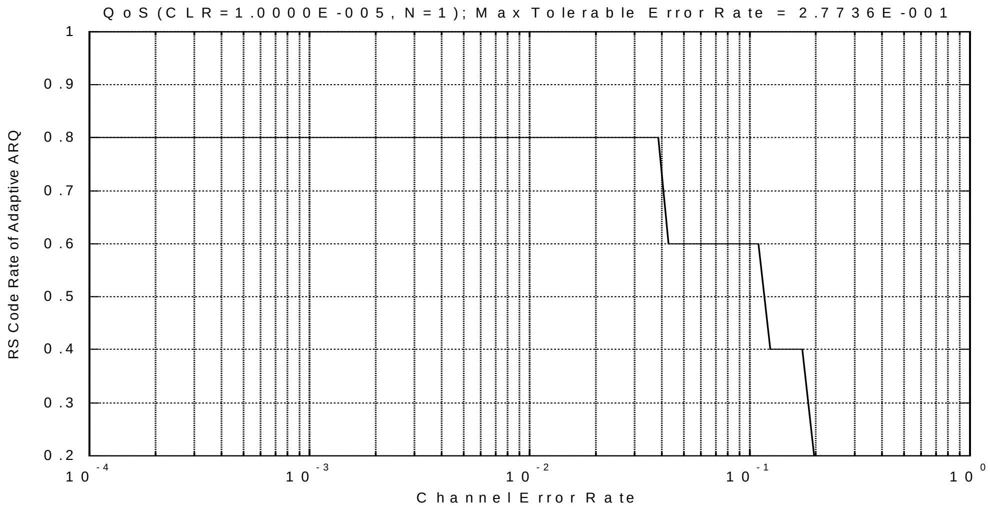

Figure 10. Reed-Solomon Code Switching Points of the Adaptive Real-time ARQ.

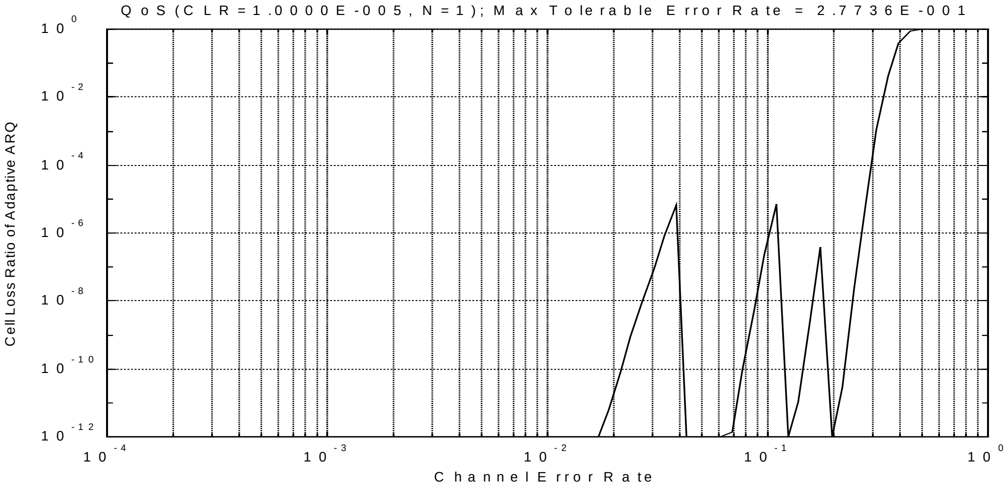

Figure 11. Cell Loss Ratio of the Adaptive Real-time ARQ.

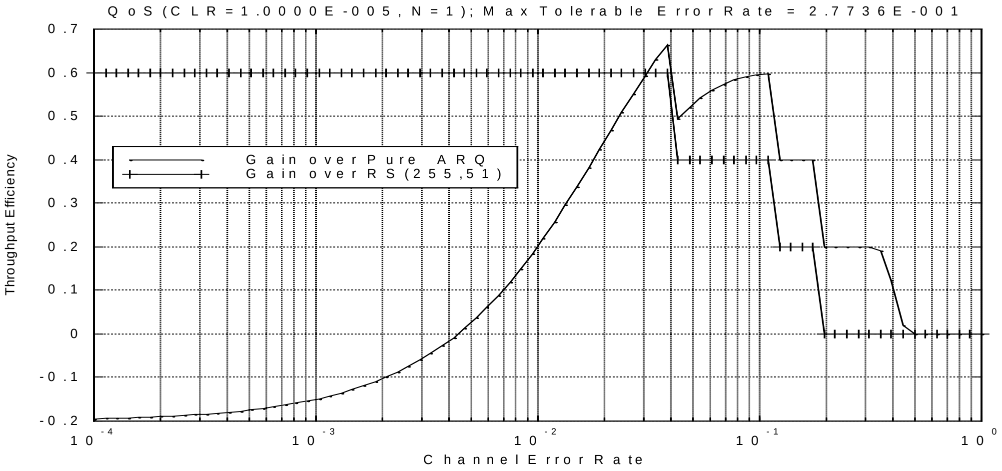

Figure 12. Throughput Efficiency Gain of the Adaptive Real-time ARQ over the Fixed ARQs

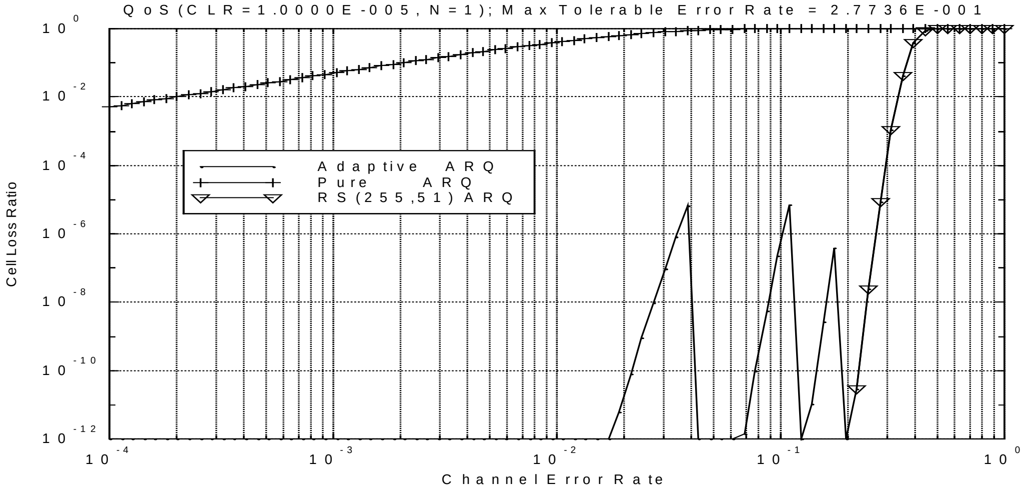

Figure 13. Cell Loss Ratio of the Adaptive ARQ and the Fixed ARQs

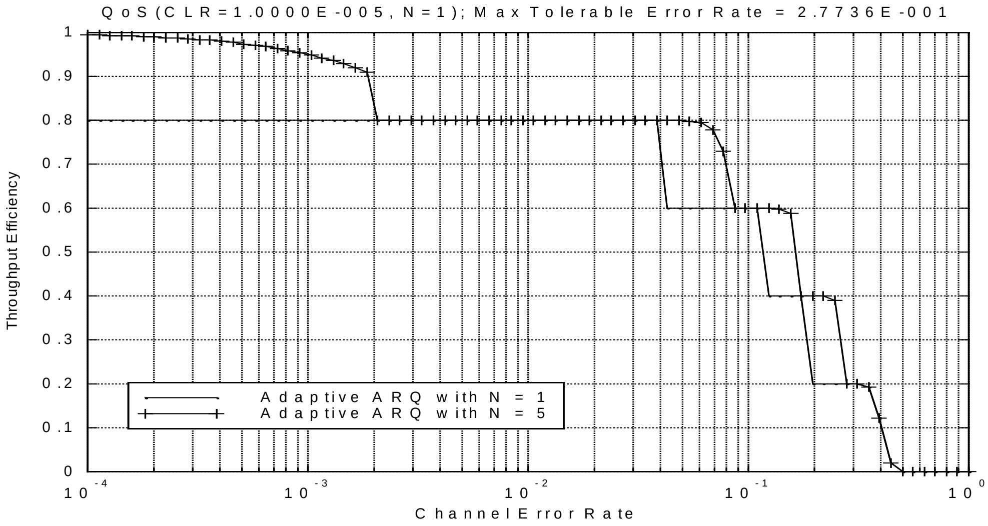

Figure 14. Throughput Efficiency of Adaptive ARQ with Different Delay QoS

Chapter 4: Real-time ARQ.

In this chapter, real-time ARQ is discussed. The importance of ARQ in error control, especially for lower layers, is highlighted. The applicability of ARQ in a real-time communication environment is explored and the special requirements in such an environment are also specified. Based on these requirements, an ARQ scheme for real-time communication channel is proposed and its efficiency is analyzed.

4.1 Automatic Repeat reQuest Schemes

Automatic Repeat reQuest (ARQ) schemes are widely used in data communication to control errors occurring from physical media. As described in the previous chapter 1, it is through retransmission of corrupted packets to achieve reliability of delivery. Sender transmits data packets along with its error detection code. Receiver checks the incoming packet against the error detection code. If there is no error in the packet, receiver returns an acknowledgment packet to inform the successful receipt of the packet. Otherwise, the receiver ignores the packet. There are several different schemes about how sender and receiver synchronize such a sending and acknowledging process. Details are given in the previous chapter.

All of these schemes are designed in early time of data communications when almost all applications are non-time-critical in nature, like email, ftp etc. As a result, the sole goal for ARQ schemes at that time is to reliably deliver all data packet to other end. The latency is not a requirement, rather than best efforts. In next few sections, we shall first discuss the requirements of general real-time applications, then the challenges for ARQ schemes.

4.2 Quality of Service Requirement for real-time applications

Quality of service for real-time application consists of delay and error requirement. Absolute delay requirement is measured by from the time a packet is generated by the application to the instant at which the entire packet is presented to the target user(s). The total end-to-end delay is the sum of the following components: packetization delay, transmission delay, propagation delay, queuing delay and processing delay. Packetization delay occurs at source end as a result of waiting for enough data samples to fill up the payload of the packet before transmitting the packet. Transmission delay and propagation delay account for the time to get the packet through the transmission media and get successfully received by the destination end. If the packet passes multiple intermediate nodes, some queuing delay may incur as the aggregate buffering delay. Processing delay is the sum of the processing time needed at each switch along the path and at the endpoints both at source and destination. All the components are relatively fixed for a specific path except transmission delay that depends on the error condition of the transmission media. Especially, for wireless media, the error condition is highly volatile. This is where an ARQ scheme works.

Another QoS requirement is error. Real-time streaming applications are typically tolerant to some error, because they are typically for human consumption. Due to the limitation of humans to perceive certain error conditions, the applications can be designed to take advantage of this to allow limited errors introduced by the network. Error requirement can be specified at the application, network and physical layers respectively. At network layer, errors may occur when packets are dropped due to buffer overflow and when packets are mis-routed due to corrupted header etc. However, it is physical layer that contributes the most to the errors. While the errors in higher layer can be deterministic and adequate network design and network management can control those error at a very low level, the errors in physical layer is very difficulty or even impossible to predict, such as in wireless media.

4.3 An Ideal real-time ARQ scheme

The detailed discussions about the requirements of real time ARQ schemes are given in this section. Note that they are ideal properties of a real time ARQ. Not all of them are hard requirements.

4.3.1 An ideal real-time ARQ should never retransmit an over-aged packet.

Each real-time connection has an absolute delay constraint. A packet of the connection does not get successfully received within the maximum delay is called over-aged. These over-aged packets are ignored by the application at the receiving end because their late arrivals make them useless for real-time applications. Sender should never attempt to transmit any packet that is to be received as over-aged. As a result, any retransmission of such a useless packet is purely a waste of system bandwidth and transceiver power. So an ARQ system should avoid this. There are several sub-requirements for implementing such a scheme. First of all, the sender needs a way to keep track of residual lifetime of each outstanding packet.

However, it is not trivial to satisfy such a requirement. Several complications are given here. First, at ARQ level, sender must keep track of residual life of all packets that have not been positively acknowledged by receiver side yet. For high-speed connections, the number of outstanding packets could be huge. Managing the lifetime of all these packets could be a serious challenge given that the system resource may be limited.

4.3.2 Safety and Liveness

This is a mandatory requirement for any communication protocol. A protocol should never produce incorrect result and never get into deadlock situation in which it stops producing correct results. What does that means for a real-time ARQ? a). ARQ scheme continuously accepts packets from its upper layer, b). A packet always gets chance to be transmitted within finite time after it reaches the data link layer while whether the packet gets received error free or not depends on the channel condition. Since these retransmitted packets will eventually be over-aged, and then discarded, the sender window will always move forward. As a result, the liveness is a less-stringent issue in a real time ARQ scheme. However, the safety issue should be paid more attention in designing a mechanism to handle the synchronization of the communication ends when discarding an over-aged packet, given that any packet could be lost including those packets carrying packet-discarding information. Unilateral packet discarding will result in loss of synchronization without proper self-restoration capability.

When two communication entities temporarily lost the synchronization, it is mandatory that ARQ scheme can regain synchronization on its own in finite time. The feature is called self-stabilizing. This requires the ARQ scheme be capable of recovering from the sequence mismatching resulting from previous packet losses or decoding errors. Put it in a simple way, a packet that carries the information of packet discarding should also carry the information about the packet(s) discarded at earlier time. This redundant information will facilitate an ARQ scheme to automatically correct previous loss of information about packet discarding.

4.3.3 Minimum System Resources is needed

System resources may come to be an issue for some wireless terminals (I.e. handle held computing device, wireless terminals etc.) with low processing capability as result of power consumption and size constraints. The requirement for system resources to implement an ARQ scheme should be minimized. For instance, system timers for implementing time out for each outstanding packet may be a problem since the number may be prohibitively large. Another important system resource is memory. The cost for memory is not an issue since its price has been declining in recent years. The issue is the power consumption for some low power mobile devices with battery. The other issue with memory is the size limitation for some compact packet devices. The major memory consumption comes from buffers in ARQ scheme, which implies the window size of sliding window protocol should be limited for high-speed connections.

4.3.4 Low Overhead

Any communication protocol involves overhead. The overhead of a communication protocol consists of the following components.

- Transmission bandwidth. Since all protocol needs a header carrying protocol specific information in comparison to the payload data from upper layer, this header is prefixed to any payload data and is transmitted together with the payload data. So it does take certain percentage of system bandwidth for transmitting the header. Shorter header means the fewer overheads ARQ scheme takes.
- Processing load. Enforcement of a communication protocol means that some computation will be needed. Depending on complexity of a protocol, it may take

varying processing time. In a mobile system with low-end microprocessor, it could be overloading to process any complicated protocols.

- Power consumption. In a wireless communication environment where some mobile devices are powered with battery, low power consumption in such a system is always desirable, and a system requirement. While the electronic components in the system play a significant part of power consumption, a communication protocol should be designed so that low power mode should be possible. For instance, system may enter sleep mode when there is no connection is active. In light load, system may periodically doze to save battery life. These require the communication protocols are capable of handling on-off service and interrupting.

4.3.5 High Performance in Throughput and Low Latency

This is the performance aspect of a communication protocol. Higher throughput and lower latency is always desirable characteristics of any protocol.

4.3.6 Fairness

Each packet from the same service category should get equal chance to be transmitted. This is a fairness issue in scheduling among the outstanding packets. When channel is under a high error condition, retransmission requests will increase. Usually, higher priority is given to those packets that need to be retransmitted. It is possible that all assigned bandwidth is occupied by retransmissions and new packet never get chances to transmit until those retransmitted packets are over-aged, by that time, some new packets may have so little residual time that is not enough even for a single transmission. In real-time communications, this translates into a chunk of packets missing in a data stream. It is undesirable since some smooth-type filter can not applied to such a stream to recover from a batch of packet losses. In a real time video and audio application, a chunk of data loss will result in service blackout and obvious interruption that human can observe. In case of high error conditions, on one hand, more bandwidth should be requested. On the other hand, policy for deciding which packet get the next slot in a connection should give all outstanding packets adequate chance to transmit.

4.4 Clarification of acknowledgment

Acknowledgment information can be carried in designated acknowledgment packet in which the correctly received packets are identified through sequence numbers. Acknowledgment information can also be carried with data packets flowing from receiver to sender through piggyback. While the second way seems more attractive from utilization of bandwidth, but it is not always possible in an unbalanced data transfer. In which case, the receiver still needs to send designated acknowledgment packets, the same way as the first one. This adds more complexity to protocol, as a result, piggyback acknowledgment mechanism is rarely used in data communication protocols.

There are two kinds of acknowledgments a sender may get from the receiver, namely positive acknowledgment and negative acknowledgment. A positive acknowledgment packet contains the information that one or more packets have been successfully received. When receiver gets an error-free data packet, it returns the sender a acknowledgment packet via feedback channel. By comparison, it is much more complicated for negative acknowledgment due to potential errors in both forward and feedback channel. In most systems, if a packet gets errors in forward channel, receiver would not know the sequence number of the packet since the packet sequence number field in packet header, may be also in error. In this case, the receiver can not send any negative acknowledgment due to the lack of erroneous packet identification information. So, sender will not get any acknowledgment packet at all. If a packet is received with error-free, but the acknowledgement packet gets error in feedback channel, the sender would not know what packet the acknowledgement packet tries to acknowledge. In either case, the consequences are same, i.e. sender does not know whether a packet is successfully received or not if it is not positively acknowledged.

However, there are some systems in that the receiver knows the identification of erroneous packets in certain situations.

For example, in some protocols, packet header has a separate checksum or even heavily protected FEC code. The whole packet including packet header and payload data has a different checksum. In this case, if the packet checksum is good. The receiver knows the sequence number of erroneous packet. So it can send a negative acknowledgment.

Here is another example. In a dynamic token allocated TDMA based network, every slot is pre-allocated. The dynamic slot distributor, which usually located in base station, assigns slots for each mobile device (node) based on the demand of each node. The slot assignment information is carried in beacon, which is transmitted before data slot as a short signal. The slot assignment information includes node identification, transmission direction and type of data (data packet or acknowledgment packet) etc.

So if the sender resides on base station, it would know which slot is given for which mobile node and transmission direction (sending or receiving slot). In this case, the sender may know an acknowledgment for which packet is expected a receiving slot if the sender knows the exact round trip delay. This may be pre-determined from TDMA hardware design based on the maximum possible delay. However, the maximum delay may be too loose to satisfy the minimum QoS delay in some circumstances. For wireless LAN, this may be acceptable. Obviously, these conditions do not hold in most situations in real world. First of all, it is very difficult to estimation the exact round trip delays given that there may be multiple applications running on modern mobile devices, so the processing time for a specific data packet may be random. TDMA requires the mobile device follows the rule in timely fashion, otherwise the assigned slots are wasted. There may be occasional occurrences of such a slot loss. As a protocol, it must be able to handle all situations including those occasional abnormal ones. In other words, there may be some situations the sender can infer the missed acknowledgment, but it can not rely on it as a protocol. If the sender resides on mobile device, it also has the information about the slot that was allocated by base station and its transmission direction. It is in same shoe as sender is in the base station.

In W-ATM, there may exist a number of virtual channels between base station and mobile terminals (or devices). An acknowledgment must bear node and channel identifier. A shorten VPI and VCI combination may be used to identify a specific channel within a mobile terminal. This channel identification may be alternatively combined with slot allocation schemes, in which case, acknowledgment slot announcement contains the channel identification information. For instance, the information could be put into beacon if it does not take reasonable amount of bandwidth.

In a non-dynamic slot allocation scheme, the channel identifier must be contained in an acknowledgment packet itself.

4.5 Analysis of TDMA-based Multi-access with dynamic slot allocation

When slot manager assigns a slot to a mobile node, it also assigns an accompanying acknowledgment slot to that mobile node. This acknowledgment slot is designated for receiver to acknowledge the data packet. If the sender resides on the same station as slot manager does, it would know which data slot corresponds to which acknowledgment slot. So, conceivably, sender knows that what packet should get acknowledged in a specific acknowledgment slot no matter whether the sender gets a valid acknowledgment packet during the acknowledgment slot or not. However, this is based on two assumptions.

1. Slot manager has the exact round trip delay information for the data packet when it assigns a data slot to a mobile node.
2. The mobile node can timely process a data packet and send an acknowledgment in the assigned acknowledgment slot corresponding to the data packet.

For the assumption 1, the difficulties lie in estimation of the round trip delay at such a high accuracy in a high-speed network without introducing extra delay due to possible choice of upper bound delay value. For the assumption 2, in a TDMA network, all nodes must keep synchronized. So timing is a hard requirement for all nodes. The question is how does the receiver side know what acknowledgment slot should be used to acknowledge which received data packet. The slot manager may put some information in the beacon to indicate that what slot after the data slot should be used to send the acknowledgment for the data packet. Or a better solution could be the acknowledgment slot is always a fixed distance after the data slot. This distance is proportional to the time a mobile node needed to process a data packet and prepare acknowledgment packet. Since processing speed varies from mobile device to mobile device, the distance can be a configuration parameter that is set during the initialization phase through handshaking. We call such a distance as processing distance.

There are some new difficulties. Sometimes, due to the conflict with other mobile node, it may be impossible for slot manager to assign a acknowledgment slot to a mobile so that the distance between two slots is exactly the distance established during the initial phase if there are multiple mobile device with different processing distances. When that happens, the first available acknowledgment slot after the processing distance is assigned instead. In that case, receiver sends the acknowledgment of a data packet in the first available slot beyond the processing distance.

Chapter 5: D-Bit Real-time ARQ

In this chapter, a real-time ARQ scheme, called Discard-Bit (D-Bit), is presented. As discussed previously, real-time connections have stringent delay requirement. Over-aged packets are useless for delay-bounded real-time applications. From the ARQ protocol point of view, there should be a way of discarding those over-aged packets. I.e. when a packet is over-aged, ARQ protocol should ensure the over-aged packet are not be transmitted. Designing an efficient packet discarding mechanism in ARQ is not a trivial task. The major challenges lie in the synchronization of two communication entities in a distributed computing environment with unreliable communication links. The second challenge is to achieve high throughput efficiency and to keep protocol overhead reasonably low.

5.1 Definitions:

Next, we shall describe a light-weighted ARQ protocol for real-time communications. To facilitate description, a few definitions are given below.

- **ARQ Packet**: ARQ packet is the protocol data unit of an ARQ protocol. There are two types of ARQ packet. I.e. data packet and acknowledgment packet. A data packet is the ARQ packet that carries application data (information). Acknowledgment packet is the packet used by receiver to inform the transmitter of the status of data packet reception. An acknowledgment packet can indicate that the data packet is successfully received (positive) or unsuccessfully received (negative). It also can be used to indicate the status of a group of packets and receive window states as we shall see later.
- **Outstanding Pakcet**: A data packet is outstanding if it was transmitted but has not been acknowledged and has not been timed out.
- **Sequence number**: Each ARQ packet is given a number by the transmitter. This number is unique among the outstanding ARQ data packets waiting for acknowledgment from the receiver. Sequence number is used by the receiver to

distinguish any duplicate copy of data packets and packets assembly for orderly delivery to upper layer. And it is also used to match the data packet with an acknowledgment packet on the transmitter side.

- **NT packet**: A data packet is non-transmittable (NT) if at least one of the following conditions is satisfied
  (a) It has been positively acknowledged.
  (b) It has not been positively acknowledged within its maximum delay. I.e. it has been over-aged.
  (c) Its residual life is not long enough for another transmission.
  Note that NT packet is a concept specific to the transmitter.
- **D-bit**. A single bit in ARQ packet header. This bit is used by transmitter to inform the receiver that all the outstanding packets that are older than the packet that carries the D bit will not be transmitted no matter whether these packets have been successfully received or not. So named as discard bit (D-bit). The D bit in acknowledgment packet must be set if it acknowledges a data packet whose D bit is set.
- **D-packet**: A packet with D-bit in the packet header is set. A D-packet may be a D-data-packet or D-ACK packet.
- **ANT packet**: Active Non-Transmittable (ANT) packets are those NT packets, of which sender has notified the receiver through D packet, but have not been acknowledged with D ACK packet. Among the NT packets in the sender window, some of them may be announced to the receiver through D packets while some of them are not because some younger packets have not been acknowledged. ANT packets refer to the first category.
- **YNT packet**: the Youngest Active Non-Transmittable (YNT) packet is the packet arriving at transmitter in the latest among all active non-transmittable packets.
- **ONT packet**: the Oldest Active Non-Transmittable (ONT) packet is the packet arriving at transmitter in the earliest among all active non-transmittable packets.
- **Sliding window**: a primitive structure used by both the transmitter and receiver to control transmission flow to limit the buffer size on receiver side. Window here consists of a set of data packets with consecutive sequence numbers. The smallest sequence number is called the low bound of the window. And the largest sequence

number is called the high bound of the window. The difference between the low bound and high bound is usually a fixed value called window size. When the packet with sequence number of low bound is received (for receiver) or acknowledged (for transmitter), the new low bound of the window is the sequence number next to the low bound. As a result, the high bound is also moved to a higher sequence number. This is why it is called sliding window. Transmitter always picks up a packet whose sequence number is within the send window to transmit. On the other end, the receiver only receives a data packet whose sequence number is within the receive window. Since the transmitter can only have at most the number of outstanding data packets equal to the send window size, the receiver buffer size is limited to the window size. Packets with logically smaller sequence number are placed at the left-hand side of the window.

- **Next packet to send**:
- **Youngest Packet Transmitted (YPT)**: This is an attribute parameter of send window. Among all packets having been transmitted, YPT is the packet arriving at ARQ layer most lately. Note that YPT packet may not be acknowledged. As a matter of face, in most cases, it is not.
- **Oldest packet to receive (OPTR)**: This is a concept specific to the receiver. The oldest packet to receive is the leftmost packet having not been received in the receiver window. It is the oldest packet that the receiver expecting to receive.

5.2 D-bit Processing and Observations

D packet processing is a kind of three way hand-shaking. In error free environment, the processing of D packet is straightforward. Transmitter decides if the D-bit of a data packet should be set or not. The rule is simple. If all of the packets in the send window older than the packet to be transmitted are NT packets, the D-bit of the packet is set, otherwise, the D-bit of the packet is cleared. The new D-packet is YANT. When the receiver receives the D packet, it returns acknowledgment of the D packet with the D-bit set. It also passes all data packets from the low bound of the receive window up to and including the packet to upper layer. It moves the low bound of the receive window to the packet next to the newly received data packet.

However, the low bound of the send window can not move.

Observation 1:

Let P(t) denotes a data packet arriving at transmitter at time instant of t. If t1>t2 and P(t2) is over-aged, then P(t1) must also over-aged.

In a real-time stream, maximum delay QoS index is applied to each packet. Packet is generated earlier will expire earlier.

Observation 2:

By successfully receiving a D-packet of sequence number n, it means that the packet with sequence number of n is successfully received and all packets that are older than packet n are NT packets.

5.3 Protocol State Transition

Following diagram gives the state transition of the real-time protocol.

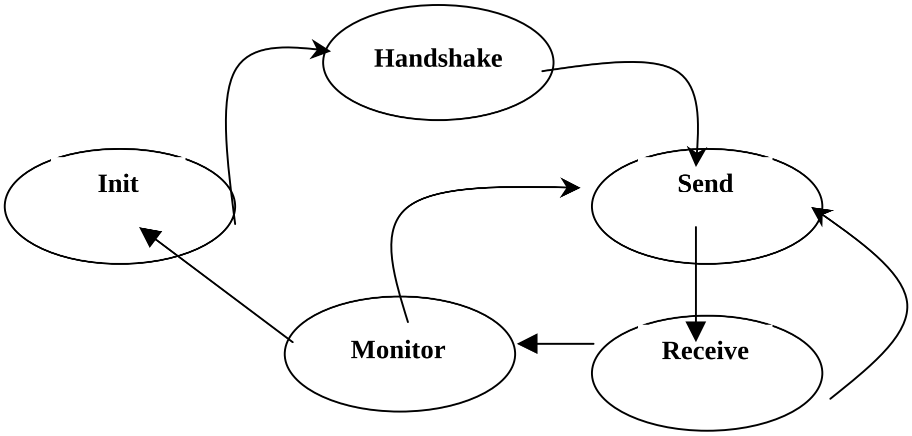

Figure 15. Protocol State Transition Diagram of the Transmitter

From the top level, the Discard Bit Scheme can be described as the state transition diagram above on the transmitter side. In the Init (Initialization) state, the transmitter configures its protocol data including allocating buffer and timer etc. There is no packet exchanges with the receiver during this state. After the initialization done, the protocol enters the Handshaking State. In this state, the transmitter will negotiate with the receiver to configure some protocol parameters like window size etc. This state also serves as the synchronization of the initialization process of both sides in case that one side finishes earlier that the other side. The next state is the Send State in which the transmitter sends one or more data packets to the receiver. During the state, the transmitter decides which packet(s) to send and in which slot. In an error-free environment, it is straightforward. However, it is much more complicated in error-prone transmissions. We will discuss the details of this state later. In non-burst mode, transmitter sends one data packet, then immediately enters the receive mode to receive acknowledgment. Depending on what type of acknowledgment received, the transmitter may change the protocol internal data accordingly in this state. The transmitter may enter Send State or Monitor State as a result of processing the acknowledgment. In Monitor State, the transmitter tries to detect if any synchronization between the two ends has occurred. If yes, the transmitter will reset and starts over.

On the receiver side, the state transition is alike with minor difference.

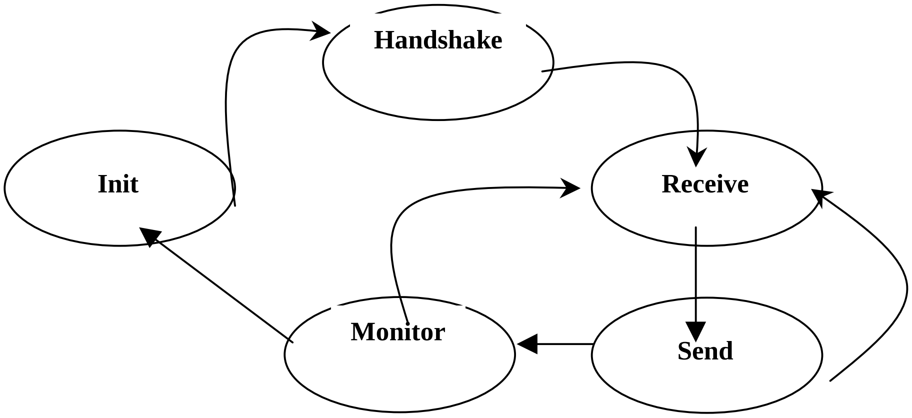

Figure 16. Protocol State Transition Diagram of the Receiver

The difference is the position of the Receive and Send States exchange. This is because the receiver always tries to receive data packet first, then decides what type of acknowledgment is to be sent.

5.4 Protocol Data Unit and Protocol Variables

As there are two types of ARQ packets with the Discard Bit scheme: Data packet and acknowledgment packet.

Type	D	Seq	0
Seq			1
VCI			2
VCI	PLT	CLP	3
Data			4-51
CRC			52
CRC			53

Figure 17. ARQ data packets format.

Type	D	Seq	0
Seq			1
CRC			2
CRC			3

Figure 18. ARQ acknowledgment packets format.

Fields:

Type = subtypes of the packet (2 bits).

D = discard bit (1 bit).

Seq = sequence number (13 bits).

VCI = Virtual Channel Identifier (12 bits).

PLT =Payload type (2 bit) and reserved bit (1 bit).

CLP = Cell Loss Priority (1 bit).

CRC = Cyclic Redundancy Code (16 bits).

In the designed rt-ARQ, the control part of the protocol is mainly on the transmitter side. I.e. it is the transmitter that dictates the packet exchange process. Specifically, the transmitter side performs the following major functions.

- Initialize ARQ sessions.
- Manage the ages of data packets
- Make retransmission and packet discarding decisions
- Monitor protocol states

The following diagram shows the variables of both send and receive window and their relationship.

The transmitter sends D packet to move the lower bound of the receiver window, which incurs the packet delivery to upper layer. The lower bound of the send window is forwarded through the receiption of an ACK to a D data packet. The objective of using YNT is to ensure that the transmitter does not send any over-aged packet and acknowledged packet (NT packets). YNT is re-assigned when either the current YNT is acknowledged or over-aged. The YNT may leap forward in a pace of multiple packets when a type 2 group acknowledgment is received. On the receiver side, OPTR is used for group acknowledgment. This is important when the feedback channel is noisy, in which case, some positive acknowledgment may be lost, so unneccesary retransmissions may occur. However, due to the nature of selective repeat retransmission scheme, it is not always possible to send a group acknowledgment in the proposed rt-ARQ protocol.

The sender window can not move until a D acknowledgment is received. Otherwise, the sender window and receiver window may lose synchronization due to some D-data or D acknowledgment packet losses, in which case, unilateral window movement occur.

The send window keeps track of the lower bound of the receive window, which is learned from the receiver through the type 1 and type 3 acknowledgment. This is necessary for two reseaons. First, normally, the lower bound of the reciever window is always following the YNT of the sender window as a result of D packet receiption on the receiver side. When the difference between rcvLow nad YNT is large, the sender can conclude that the receiver window is torward full due to heavy D packet loss. So the sender can make intelligent decision as to resend YNT packet to move the receiver window low bound to avoid window full situation. Secondly, in a very low error environment, due to the infrequency of D packet, sender may need to explicitly send a D packet purely for the purpose of moving receiver window.

The transmitter periodically checks the distance between the YNT and the receive window lower bound. It may send a special data packet with the YNT sequence number inside just to move the receiver window if the difference is above certain threshold value.

The lower bound of the receive window is always between the the lower bound of the send window and the YNT of the send window. This is because that the rcvLow moves only when a D packet is received according to the protocol specification of the receiver and a D packet never younger than the YNT of the send window. The OPTR may be at most one packet younger than the YPT of the send window.

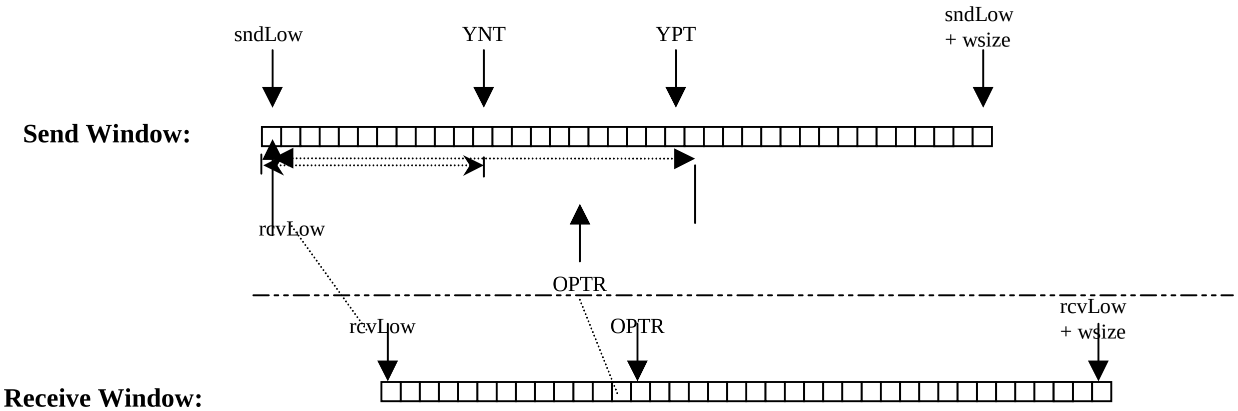

Figure 19. Window variables and their relationship

5.5 Protocol Description

5.5.1 Receiver

The logic on the receiver side is more straightforward than that on the sender side since it only passively processes the data packet and generates ACK accordingly. The receiver replies the sender an ACK deliberately with the following objectives.

- To inform the status of data packets as much as possible with a single ACK
- To inform the status of receive window to help synchronization
- To reduce the number retransmissions due to corrupted ACK

There are three types of acknowledgments the receiver may send.

Type 1 ACK:

It is a normal positive acknowledgment in a conventional selective repeat ARQ protocol. It is designed to acknowledge only a single packet.

Type 2 ACK:

This acknowledgment packet is designed for announcement of the oldest packet expecting to receive. When sender receives this type of acknowledgment, it assumes all the packets older than this packet and including this packet are successfully received. So, it is a type of group acknowledgment.

Type 3 ACK:

It is used for receiver to announce of the low bound of the receive window to achieve re-synchronization with the sender.

Depending on the status of the data packet and its internal status of the receive window, the receiver replies acknowledgment accordingly. The type 1 acknowledgment is sent when all of the following conditions hold with the received data packet.

- It is error free.
- It is within the receive window.
- It is not the OPTR.

The sole purpose of type 1 acknowledgment is to inform the sender of successful reception of the data packet.

The type 2 acknowledgment is sent when all of the following conditions hold with a received data packet.

- It is error free.
- It is within the receive window.
- It is the OPTR

In which case, receiver needs to locate the new OPTR and sends type 2 acknowledgment with the sequence number of that of the new OPTR. In comparison to type 1 acknowledgment that acknowledges a single packet, the type 2 acknowledges multiple packets, so called group acknowledgment. This can potentially reduce the number of unnecessary retransmissions caused by the lost acknowledgments due to feedback channel errors.

The purpose of type 3 acknowledgment is for announcing the low bound of the receive window. This type acknowledgment is sent when the following conditions satisfied:

- It is error free
- It is out of the receive window.

The motivation is to inform the sender of the span of receive window and let sender take measures to re-synchronize its window with the receiver’s. This is because the reception of out of window data packet is a strong indication that two sides may have lost the synchronization. Such as out-of-synchronization may occur like false acknowledgment etc., which is an extremely low probability event with CRC-16 error detection code.

The following is the high-level pseudo-code of protocol processing on receiver side.

```
Constant:
  WSIZE: Window size in packets.
Variables:
  optr: The oldest packet to receive
  seq(P): the sequence number of packet P
  dbit(P): the d bit value in the packet header of packet P
Procedures:
Receive(P)
Do CRC-16error-checking on P
If P passes the error-checking
  If seq(P) is within the receive window
    If P is not a duplicate
      Insert P into the receive window
    End
    If P is OPTR
      Find new OPTR
      Reply a type 2 ACK with seq of the new OPTR and set D-bit
    Else
      Reply a type 1 ACK with seq(P) and copy the dbit(P)
    End
    If dbit(P) is set
      Deliver all packets before and including P to upper layer
      Update internal management data
    End
  Else
    Reply a type 3 ACK
  End
Else
  Reply a type 2 ACK (OPTR) with D-bit set.
End
```

5.5.2 Transmitter

```
Constant:
  WSIZE: Window size in packets.
  SND_THRESHOULD: Maximum distance between nYNT and rcvLow

Variables:
  sndLow: The lower bound of the send window
  rcvLow: the lower bound of the receiver window
  nYNT: Next to the YNT packet
Procedures:
INIT():
  sndLow := 0; rcvLow := 0; nYNT := 0;
ProcessDbit():
  Update sndLow
  Update rcvLow
  Free the buffers of this packet and all older packets
SetNextToYNT():
  Mark the D bit
  Re-generate checksum
PickNextToYNT():
  If nYNT is not set
    Set the window head packet as nYNT
    Mark it as D packet
    Re-generate the CRC-16 checksum
  End
  If (nYNT – rcvLow > SND_THRESHOULD)
    Send a special packet with seq of nYNT
    Return
  End
  If the nYNT is not outstanding
     Send the nYNT
    Return
  Else if the special packet is not outstanding
    Send it with seq of nYNT
    Return
```

```
  Else if (the oldest not outstanding data packet is found)
    Send it
    Return
  Else
    Send a new data packet
    Return
  End
PickNextPacketToSend():
    If the window is not full
      Send a new data packet
      Update internal management data
    else
      PickNextToYNT()

ProcessAck():
    If (A type 1 ACK is received)
      If (It is a D-ACK)
        ProcessDbit()
      Else
        If it acks the nYNT
          Set a new nYNT
        Else
          Mark the acked packet as NT packet
        End
      End
      PickNextPakcetToSend()
    Else if (A type 2 ACK is received)
      Mark as NT all packets that are older than announced OPTR
      Update nYNT
      PickNextPakcetToSend()
    Else if (A type 3 ACK is received)
      Update the mirror variables of the receive window
      Remove all packets that are older than the announced rcvLow
      Update nYNT
      PickNextToYNT()
    End
  Else
    Send the most recent time out packet
```

```
End
```

5.6 A Scenario of Protocol Execution

The following tables show a case of the protocol execution. Protocol variables values are given for time slots.

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Ont											2				6				8
Ynt							2	3	4	5	6	6	7	7	8	8	9	9	10
Tx	1	2	3	4	5	6	2d	7	4d	8	6d	9	7d	10	8d	11	12	13	14
Ack					1	-2	3	-4	5	-6	2d	-7	-4d	8	6d	9	-7d	10	8d
Rx	x	x	1	2	3	-4	5	6	2d	7	-4d	8	6d	9	7d	10	8d	11	12
NT									2				6		7		8		

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
	10	11	12	13	14	15	16	17	18	18	19	20	21	22	23	24	25	26	27
Tx	15	16	17	18	14d	19	20	21	22	23	24	25	26	27	28				
Ak	11	12	13	-14	15	16	17	18	-14d	19	20	21	22	23	24	25	26	27	28
Rx	13	-14	15	16	17	18	14d	19	20	21	22	23	24	25	26	27	28		
							14												

Table 1. A case study of protocol execution

5.7 Simulation Model and Performance Analysis

A simulator is built to evaluate the performance of the proposed D-Bit real-time protocol and prove the liveness and safety properties. Extensive simulations have been done with the simulation. The architecture of the simulation model is described and key performance index is given in this section.

5.7.1 Simulation Model

The D-Bit ARQ protocol is simulated on WIN32 platform. The simulator is decomposed into the following four layers.

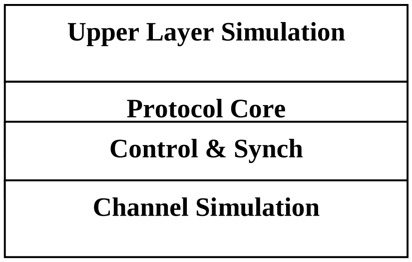

Figure 20. The rt-ARQ Simulation Model Structure

- Physical channel transport simulation: This is the lowest layer, two WIN32 file pipes are used to simulate the forward and feedback channel, of each has a scrambling error generator attached. The transceivers on both the transmitter and receiver side are simulated through two WIN32 tasks that interact with WIN32 pipes. The built-in file lock mechanism in WIN32 platform enforces the synchronization of the concurrent access of the transceiver and error generator tasks. This layer provides asynchronous data exchanges between the two peers.
- Synchronization and delay control is the second lowest layer: The round trip delay and slot alignments are realized in this layer. It also provides buffering of incoming and outgoing packet. This layer presents a synchronous transport API to the upper layer, which is protocol core.

- Protocol core: All protocol processing logic are implemented in this layer. The sliding window structure is also located inside.
- Upper layer simulation basically generates traffic and passes down to protocol layer on the transmitter side. It absorbs data streams and maintains statistics on the receiver side.

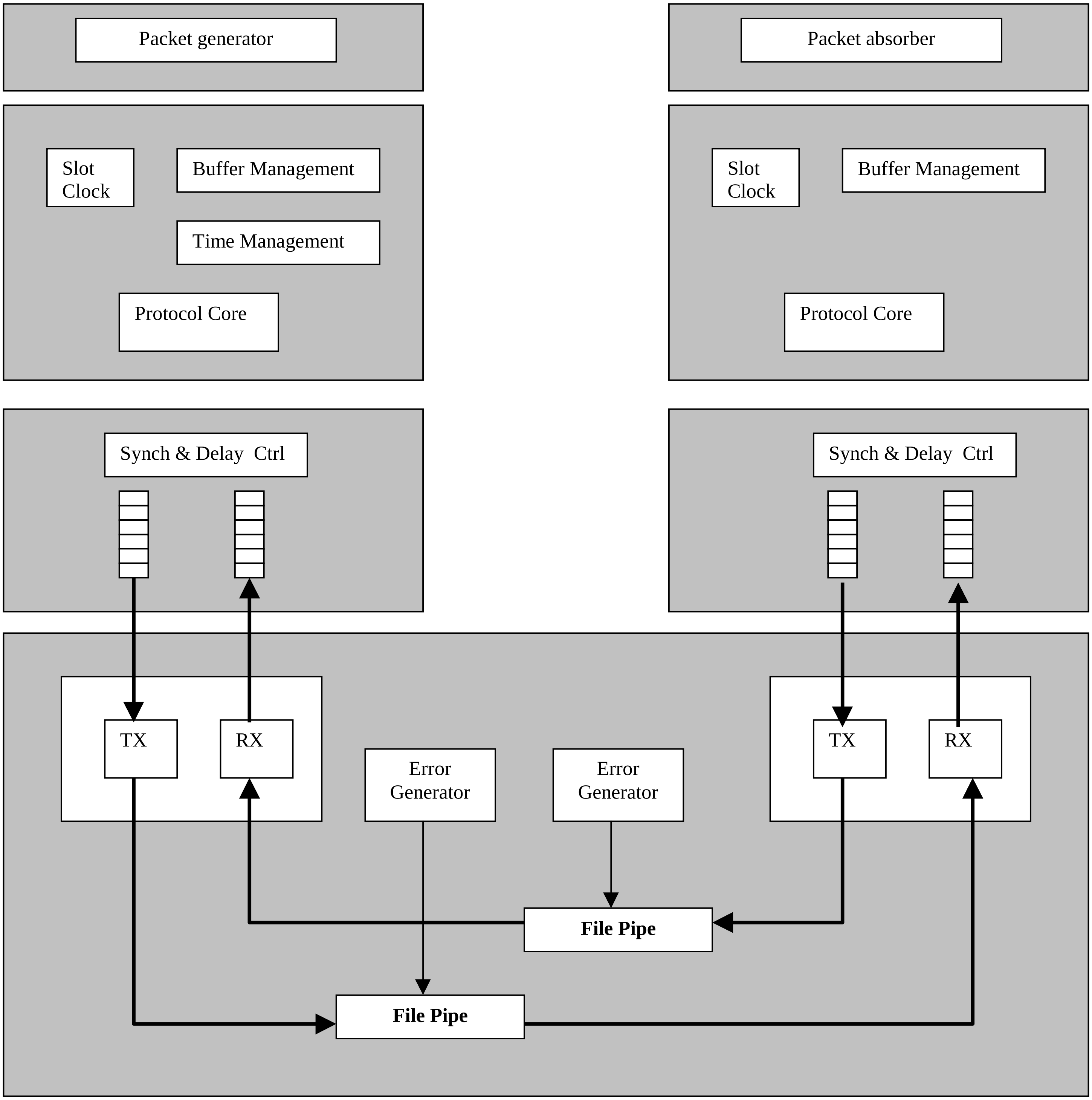

Figure 21. The structure and data flow of rt-ARQ simulator

5.7.2 Performance Analysis

The average packet loss rate of the proposed rt-ARQ is given in Figure 22. As packet error rate increases, the packet loss also increases. Quantitatively, when packet error rate is lower than 1E-2, the protocol can maintain a low packet loss rate that can satisfy a large family of real-time applications. The latency performance shown in Figure 23 is measured as the elapse time from the time instant of the first transmission to the time instant of either acknowledged or over-aged of all data packets in the unity of single packet transmission time or slots. When the packet error rate is very low, the latency converged to 5, which is the channel propagation delay. This can be intuitively induced since all packets have a high probability of being successfully received with a single transmission. On the other hand, when the packet error rate is high, nearly all packets will be over-aged after the maximum number of transmissions. So the latency will approach to the maximum delay allowed with each packet, which is 10 as Figure 23 confirms. Figure 25 shows also the latency of those packets that are successfully acknowledged. As expected, it is smaller than the latency of all packets since some of these packets take less than maximum delay to be successfully received. In both cases, it is shown that the packet latency is never larger than the required packet maximum delay. This is because the proposed rt-ARQ can fundamentally guarantee that all packets will be transmitted within their lifetime and there is no bandwidth waste to transmit any over-aged packet.

The throughput efficiency drops as the packet error rate becomes higher in Figure 24. The protocol overhead is reflected in the Figure 24 with lower packet error rate where the maximum achievable throughput efficiency is a little less than 1. The overhead is on the acceptable level of less than 5% on an error-free channel.

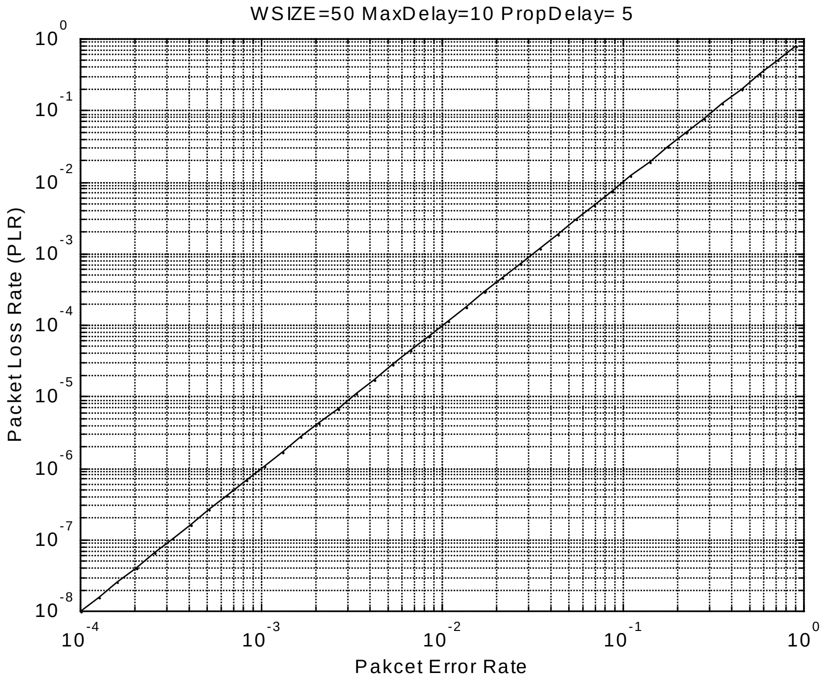

Figure 22. The performance of packet loss rate

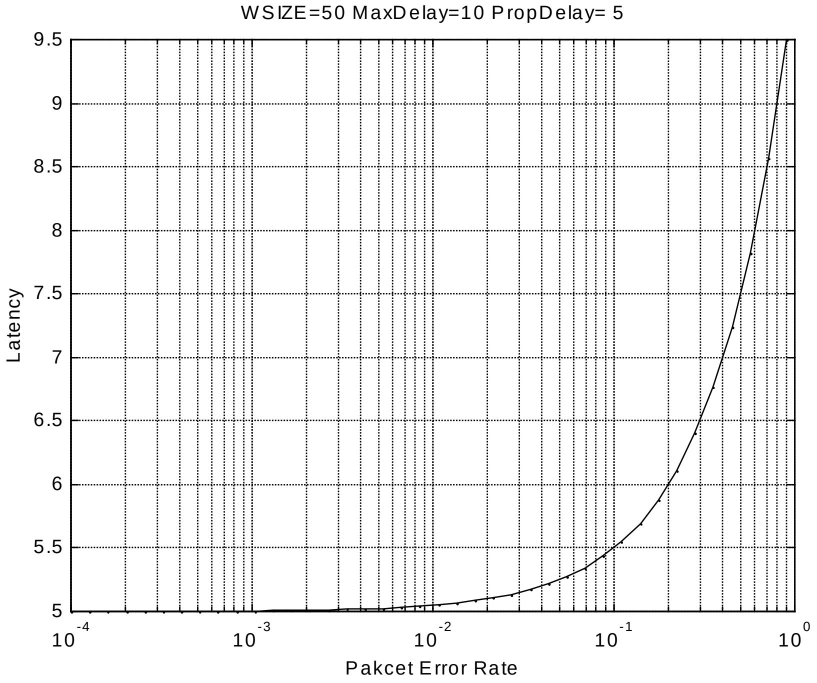

Figure 23. The performance of average packet latency

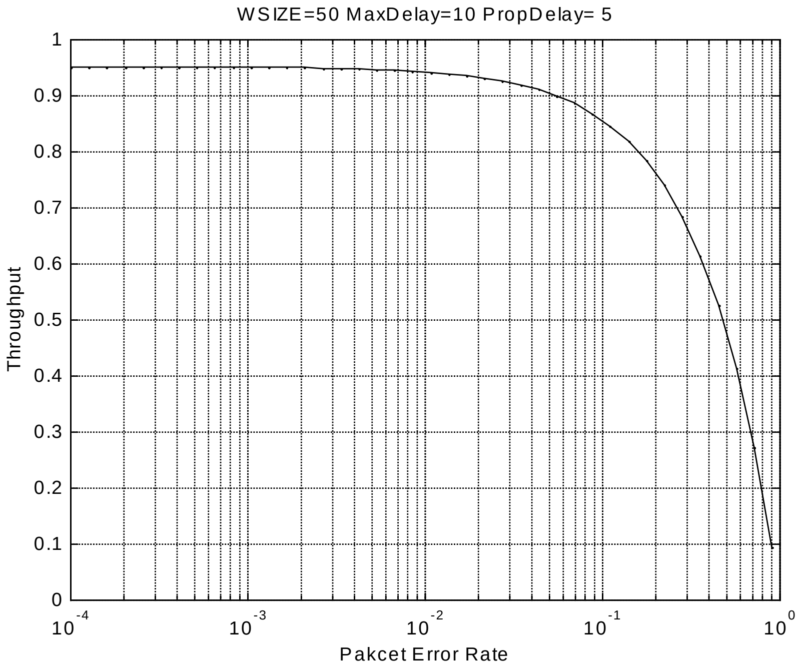

Figure 24. The performance of Throughput

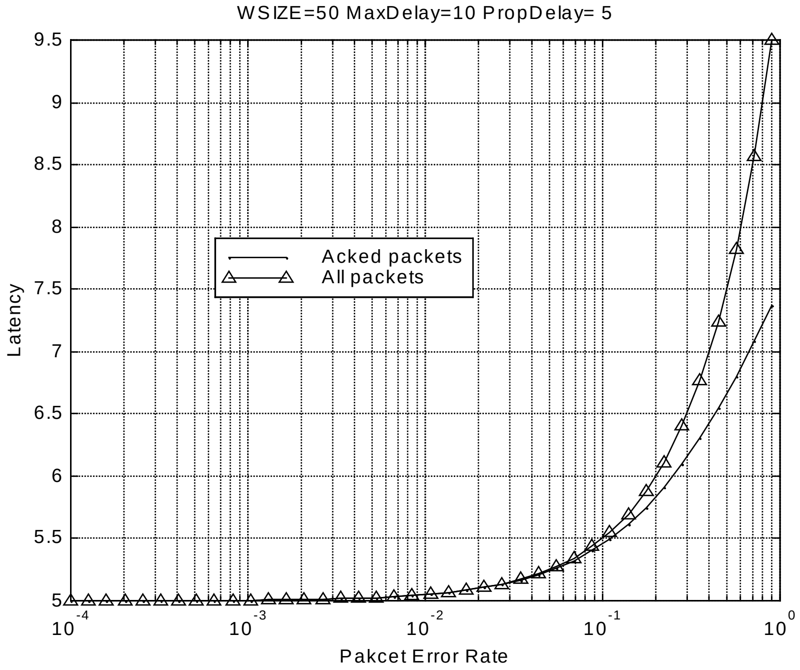

Figure 25. The performance of average packet latency

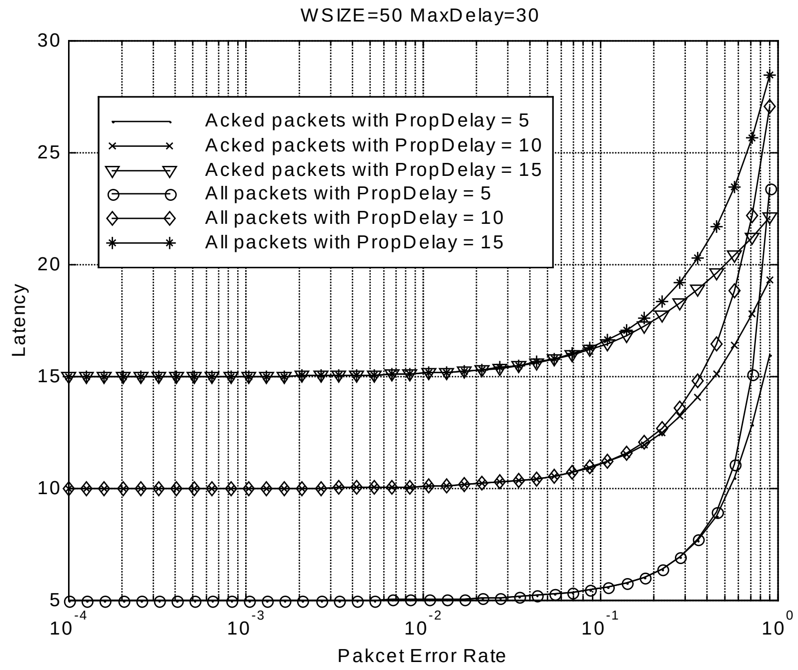

Figure 26. Latency as the propagation delay varies

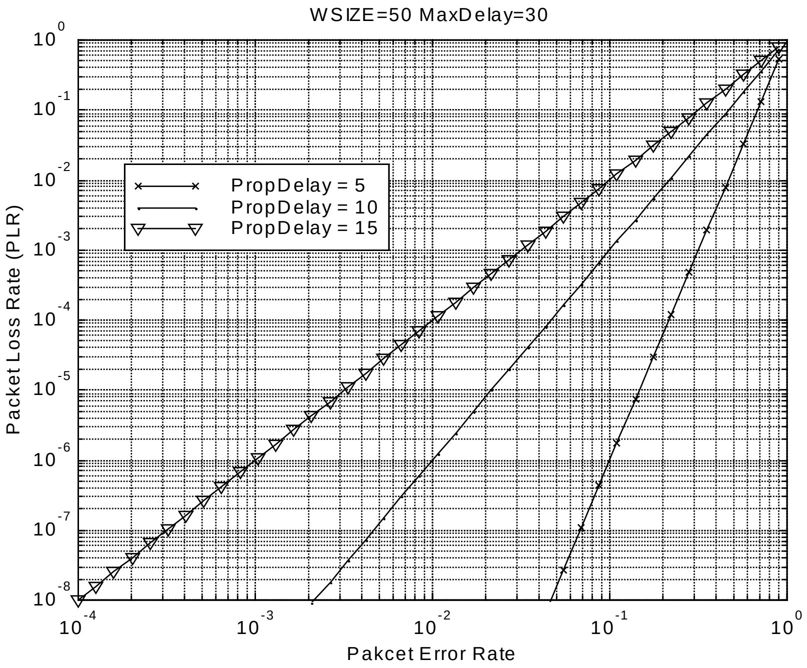

Figure 27. Packet loss rate as the propagation delay varies

With different propagation channel delay, the latency and packet loss rate are shown in **Figure 26** and Figure 27. With small propagation delay, the system can achieve a better throughput efficiency since it allows more retransmissions with the same maximum delay requirement. The latency always converges to the propagation delay as packet error rate goes low and bounded with the maximum delay as the packet error rate goes high.

Chapter 6: Conclusions and Future Research

6.1 Conclusion

In this dissertation, we investigated error control strategies in the wireless ATM environments. ATM protocol was designed for high-quality physical links like fiber optics. To support such protocol over wireless transmission media where channel conditions are both time and space varying, lower layers of protocol stack must reduce the error rate to certain acceptable level. Since ATM provides multi-level services including real-time applications, the error control schemes must have the capability of controlling both error and delay. The author believes the following contributions have been made in the dissertation.

1. We created an adaptive error control algorithm based on hybrid ARQ scheme using Reed-Solomon code. The protocol throughput performance of error control system is optimized through adaptation of FEC code rate to channel error condition, and at the same time, the delay and cell loss ratio of real time data are also satisfied. The algorithm is mathematically modeled. Based on the model, the exact performance is analyzed, and the results shown the efficiency of the algorithm.
2. The adaptive error control algorithm is further developed into a practical implementation, which is computationally simple and suitable for using on low-power wireless terminals.
3. We created a real-time ARQ protocol. Unlike the existing conventional ARQ protocol, the protocol is capable of selectively discarding packets and controlling packet delays, as a result, it can be used in a real-time communication environment. The protocol also has a self-stabilizing capability. Through simulation, we showed that the protocol provides high throughput and can ensure efficient usage of bandwidth.
4. We also explored the research issues the wireless ATM brings. We found besides error control issues, there are issues like multi-access, hand over and roaming, mobility support with respect to wireless ATM.

6.2 Future work

We think there are some further research efforts deserving to carry on in the line of the research in this dissertation.

1. Investigation of other FEC codes for the proposed real-time ARQ protocol. While Reed-Solomon code is a good choice, it is worthwhile to look into convolutional code with Viterbi decoding. Some researcher claims it has a better performance in a burst error environment.
2. Combination of the adaptive and D-Bit real-time ARQ protocol proposed in this dissertation could yield a more powerful error control scheme for wireless ATM networks.
3. Mathematical model of the proposed real-time ARQ. Some mathematical tools are available like Markov chain and renewal process, flow graph etc.

Reference

[ACHA96] P. Acharya, “Primitives for Location Management and Handoff in Mobile ATM Networks,” ATM Forum/96-1121, ATM Forum Technical Committee.

[AFEK93] Y. Afek, G. Brown, “Self-stabilization over unreliable communication media,” Distributed Computing. Springer-Verloag 1993.

[AMMA92] M. Ammar, L. Wu, “Improving the Throughput of Point-to-Multipoint ARQ Protocols Through Destination Set Splitting.” IEEE INFOCOM’92.

[ANAG86] M. E. Anagnostou, E. N. Protonotarios, “Performance Analysis of the Selective Repeat ARQ Protocol,”, IEEE Trans. Comm., VOL. COM-34, NO. 2, Feb 1986.

[BERT92] D. Bertsekas and R. Gallager, “Data Networks,” Englewood Cliffs, New Jersey: Prentice Hall, 2nd Edition, 1992.

[BRUN86] H. Bruneel, M. Moeneclaey, “ On the Throughput Performance of Some Continuous ARQ Strategies with Repeated Transmissions,” IEEE Trans. on Comm., COL. COM-34, NO. 3, March 1986.

[CAIN97] J. Bibb Cain, D. McGregor, “A Recommended Error Control Architecture for ATM Networks with Wireless Links.” IEEE on SAIC. VOL. 15. NO. 1. Jan. 1997.

[CAMP97] A. Campell, “Supporting AOS Controlled Handoff in Mobiware,” 6th WINLAB Workshop on Third Generation Wireless Information Networks New Brunsiich, New Jersey, March 20-21, 1997.

[CAMR93] R. Cam and C. Leung, “Analysis and optimization od ARQ protocols for imperfect feedback channels,” in Proc. IEEE Veh. Technol. Conf., Secaucus, NJ, May 1993.

[CAMR97] R. Cam and C. Leung, “Throughput analysis of some ARQ protocols in the presence of feedback errors,” IEEE Trans. on Comm., VOL. 45, NO. 1, Jan. 1997.

[CAMR98] R. Cam and C. Leung, “Multiplexed ARQ for time varying channels - Part I: System Model and Throughput Analysis,” IEEE Trans. on Comm., VOL. 46, NO. 1, pp. 41-51 Jan. 1998

[CHAN84] Y. Chang and C. Leung, “On Weldon’s ARQ Strategy,” IEEE Trans. on Comm., VOL. COM-32, NO. 3, pp. 297-300, March 1984.

[CHAN97] C. Chang, et al, “Guaranteed Quality-of-Service Wireless Acdcess to ATM Networks,” IEEE J. on Selected Areas in Communications. BOL. 15, NO. 1, Jan. 1997.

[CHOY94] Y. J. Cho and C. K. Un, “Performance analysis of ARQ error controls under Markovian block error pattern,” IEEE Trans. Comm., VOL. 40, pp. 2051-2061, Feb.-Apr. 1994.

[COOP81] R. B. Cooper, “Introduction to Queuing Theory (2nd ed.)” New York: Elsevier/No9rth_Holland. 1981.

[COXD77] D. Cox, H. Miller, “The Theory of Stochastic Processes,” London New York, Chanpman and Hall, 1977.

[FANT92] R. Fantacci, “Performace Evaluation of Some efficient Stop-and-Wait,” IEEE Trans. on Comm., VOL. 40, NO. 11, Nov. 1992.

[FANT97] R. Fantacci, “Performance evaluation of a finite buffer generalized selective repeat ARQ scheme for satellite communications,” IEEE Trans. on Comm., VOL 45, NO. 2, pp. 140-143.

[GILM96] D. Gilmurray, Alan Jones, “Wireless ATM Access Layer Requirements,” ATM FORUM/96-1057/watm. ATM Forum Technical Committee.

[GILM96] D. Gilmurray, Alan Jones, “Wireless ATM Service Scenraios,” ATM FORUM/96-1056/WATM. ATM Forum Technical Committee.

[GOPA84] I. Gopal, J. Jaffe, “Point-to-Multipoint Communication Over Broadcast Links,” IEEE Transaction on Communications, VOL. COM-32, NO. 9, September 1984.

[GOUD85] M. Gouda, “A Simple Protocol Whose Proof Isn’t: The State Machine Approach,” IEEE Trans. VOL. COM-33, NO. 4, APRIL 1985.

[KIMS92] S. R. Kim and C. K. Un, “Throughput analysis for two ARQ schemes using combined transition matrix,” IEEE Trans. Comm., COL. 40, pp. 1679-1683, Nov. 1992

[LEET92] T. H. Lee, “Throughput performance of a class of continuous ARQ strategies for burst-error channels,” IEEE Trans. on Comm., VOL. 41, NO. 4, pp. 380-386. Nov. 1992.

[LEUN88] C. H. C. Leung, U. Kikumoto, S. A. Sorensen, “ The Throughput Efficiency of the Go-Back-N ARQ Scheme under Markov and Related Error structures,” IEEE Trans. on Comm., VOL. 36, NO. 2, Feb. 1988.

[LUDL93] D. L. Lu and J. F. Chang, “ Performance of ARQ protocols nondependent channel,” IEEE Trans. on Comm., VOL. 41, pp. 721-730, May. 1993.

[MILL81] M. J. Miller, S. Lin, “The analysis of Some Selective-Repeat ARQ Schemes with finite Receiver Buffer,” IEEE Trans. on Comm., VOL, COM-29, NO. 9, Sept. 1981.

[MORR78] Joel M. Morris, “On Another Go-Back-N ARQ Technique for High Error Rate Conditions,” IEEE Trans. on Comm., Vol. COM-26, No. 1, Jan. 1978.

[MORR79] Joel M. Morris, “Optimal Block lengths for ARQ Error Control Schemes,” IEEE Trans. on Comm., Vol. COM-27, Feb. 1979.

[NAND94] Sanjiv Nanda, Richard Ejzak and Bharat Doshi, “A Retransmission Scheme for Circuit-Mode Data on Wireless Links.” IEEE Journal on Selected Areas in Communications, Vol. 12. No. 8, October 1994.

[PERE88]. A. L. Peressini, F. E. Sullivan, J. J. Uhl, “The Mathematics of Nonlinear Programming.” Springer-Verlag, New York, 1988.

[PUJE99]. J.Pu, Y. Liu, “Adaptive Hybrid ARQ in Wireless ATM Networks.” IEEE 50th Vehicular Technology Conference, Spetember, 1999.

[RAMJ98] R. Ramjee, “Performance Evaluation of Connection Rerouting Schemes for ATM-Based Wireless Networks,” IEEE/ACM TRANSACTION ON NETWORKING. VOL. 6, NO. 3, JUNE 1998.

[RAYC94] D. Raychaudhuri, “ATM-based transport architectures for multiservices wireless personal communication networks,” IEEE J. on Selected Areas in Comm., Vol. 12, Oct. 1994. P 1401-1414.

[ROSS85] S. Ross, “Introduction to Probability Model,” Academic Press, Inc. Third Edition. 1985.

[SAAD94] Tarek Saadawi, Mostafa Ammar at el., “Fundamentals of Telecommunication Networks,” John Wiley & Sons, Inc., 1994.

[SAEK82] B. H. Saeki, I. Rubin, "An Analysis of a TDMA Channel Using Stop-and-Wait, Block, and Selective-and-Repeat ARQ Error Control," IEEE Trans. on Comm., VOL. COM-30, NO. 5, May 1982.

[SATR75] A.R.K.Satry, "Improving Automatic Repeat-request (ARQ) Performance on Satellite Channels Under High Error Rate Conditions," IEEE Trans. on Comm., pp. 436-439 April 1975.

[SPIN97] J. Spinelli, "Self-Stabilizing Sliding Window ARQ Protocols," IEEE/ACM Transactions on Networking, VOL. 5. NO. 2. APRIL 1997.

[STAM97] G. Stamatelos, V. Koukoulidis, "Reservation-Based Bandwidth Allocation in a Radio ATM Network," IEEE/ACM Transaction on Netwroking. VOL. 5. NO. 3. June 1997.

[TOWS79a] D. Towsley, "The Shutter Go Back-N ARQ Protocol," IEEE Trans. on Comm., Vol. COM-27, NO. 6, June 1979.

[TOWS79b] D. Towsley, J.K. Wolf "On the Statistical Analysis of Queue Lengths and Waiting Times for Statistical multiplexers with ARQ Retransmission Schemes," IEEE Trans. on Comm., Vol. COM-27, NO. 4, April 1979.

[WELD82] E. J. Weldon, "An Improved Selective-Repeat ARQ Strategy," IEEE Trans. on Comm., VOL. COM-30, NO. 3, March 1982.

[WICK95] S. B. Wicker, "Error Control Systems for Digital Communication and Storage." Prentice Hall, Englewood Cliffs, NJ, 1995.

[YAOY95] Y. D. Yao, "An efficient go-back-N ARQ scheme for variable-error-rate channels," IEEE Trans. on Comm., VOL. 43, NO. 1, pp. 20-23 Jan. 1995.

[YUPS81] P. S. Yu, S. Lin, “An Efficient Selective-Repeat ARQ Scheme for Satellite channels and Its Throughput Analysis,” IEEE Trans. on Comm., COL, COM-29, NO. 3, March 1981.

[ZORZ95a] M. Zorzi and R. R. Rao, “Throughput analysis of ARQ selective repeat protocol with time diversity in Markov channels,” in Proc. IEEE GLOBECOM’95, Nov. 1995.

[ZORZ95b] M. Zorzi and R. R. Rao, “ Performance bounds on the throughput of ARQ Go-Back-N protocol in Markov channels,” in Proc. IEEE milcom’95, Nov. 1995, pp. 1232-37.

[ZORZ96] M. Zorzi and R. R. Rao, “On the use of renrwal theory in the analysis of ARQ protocols,” IEEE Trans. on Comm., VOL. 44, NO. 9, pp. 1077-1081, Sept. 1996.

Appendix A: List of Abbreviations and Acronyms

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
API	Application Program Interface
ARQ	Automatic Repeat reQuest
ATM	Asynchronous Transfer Mode
B-ISDN	Broad-band Integrated Services and Digital Network
BS	Base Station
CAC	Connection Admission Control
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CRC	Cyclic Redundancy Check
DCS	Digital Cellular System
DECT	Digital European Cordless Telecommunications
ER	Explicit Rate
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplex
GCRA	Generic Cell Rate Algorithm
GFC	Generic Flow Control
GSM	Global System for Mobile Communications
HAC	Handover Admission Control
HEC	Header Error Control – Field in ATM cell header
IP	Internet Protocol
ITU	Intelligent Telecommunication Union
ITU-T	ITU Telecommunication standardization Sector
LAN	Local Area Network
LLC	Logical Link Control

MAC	Medium Access Control
MBS	Mobile Broadband System
MCR	Minimum Cell Rate
MONET	MObile NETwork
MPEG	Motion Picture Experts Group
MT	Mobile Terminal
NNI	Network-Node Interface
NSAP	Network Service Access Point
OAM	Operation and Maintenance
OSI	Open Systems Interconnection
PCM	Pulse Code Modulatino
PCR	Peak Cell Rate
PCS	Personal Communication Service
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
P-NNI	Private Network Node Interface
PSTN	Public Switch3ed Telephone Network
PTI	Payload Type Identifier
PVC	Private Virtual Connection
QoS	Quality of Service
RM	Resource Management
SAAL	Signaling AAL
SCR	Sustainable Cell Rate
SDH	Synchronous Digital Hierarchy
SN	Sequence number
SONET	Synchronous Optical Network
SVC	Switched Virtual Connection
TACS	Total Access Control System
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TDMA	Time Division Multiple Access

UBR	Unspecified Bit Rate
UMTS	Universal Mobile Telecommunications Sustem
UNI	User-Network Interface
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VC	Virtual Connection
VCI	Virtual Channel Identifier
VP	Virtual Path
VPI	Virtual Path Identifier
WATM	Wireless ATM
WWW	World Wide Web

Appendix B: Renewal Theory

Consider a stochastic process $\{X_n, n=0,1,2,...\}$ that takes on a finite or countable number of possible values. We suppose that whenever the process is in state i, there is a fixed probability P_{ij} that it will next be in state j. that is

$$P\{X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, .., X_1 = i_1, X_0 = i_0\} = P_{ij}$$

For all states $i_0, i_1, ..., i_{n-1}, i, j$ and all $n \geq 0$. Such a stochastic process is known as Markov chain.

The value P_{ij} represents the probability that the process will, when in state i, next make a transition into state j. We have that

$$P_{ij} > 0,\ i, j \geq 0;\ \sum_{j=0}^{\infty} P_{ij} = 1,\ i = 0,1,....$$

The n-step transition probability P_{ij}^n is the probability that a process in state i will be in state j after n additional transitions. That is,

$$P_{ij}^n = P\{X_{n+m} = j | X_m = j\},\ n \geq 0,\ i, j \geq 0$$

Chapman-Kolmogorov equations provide a method for computing these n-step transition probabilities.

$$P^{(n)} = P^n$$

That is, the n-step transition matrix may be obtained by multiplying the one-step transition matrix by itself n times.

Limiting Probability Theorem: For an irreducible ergodic Markov chain $\lim_{n\to\infty} P_{ij}^n$ exists and is independent of i. Furthermore, letting

$$\pi_j = \lim_{n\to\infty} P_{ij}^n,\quad j \geq 0$$

Then π_j is the unique nonnegative solution of

$$\pi_j = \sum_{i=0}^{\infty} \pi_i P_{ij},\quad j \geq 0$$

$$\sum_{j=0}^{\infty} \pi_j = 1$$

Definition: if the sequence of nonnegative random variables $\{X_1, X_2, ...\}$ is independent and identical distributed, then the counting process $\{N(t), t \geq 0\}$ is said to be a renewal process. Where X_n denotes the time between the (n-1)st and the nth event of the counting process.

Consider a renewal process $\{N(t), t \geq 0\}$ having inter-arrival times $X_n, n \geq 1$, and suppose that each time a renewal occurs we receive a reward. We denote by R_n, the reward earned at the time of the nth renewal. We shall assume that the $R_n, n \geq 1$, are independent identical distributed. However we do allow for the possibility that R_n may (and usually will) depend on X_n, the length of the nth renewal interval. If we let

$$R(t) = \sum_{n=1}^{N(t)} R_n$$

then $R(t)$ represents the total reward earned by time t. Let

$$E[R] = E[R_n], \quad E[X] = E[X_n]$$

Proposition. If $E[R] < \infty$ and $E[X] < \infty$, then with probability 1

$$\frac{R(t)}{t} \to \frac{E[R]}{E[X]} \quad as \quad t \to \infty$$

If we say that a cycle is completed every time a renewal occurs, then the Proposition states that the long-run reward is just the expected reward earned during a cycle divided by the expected length of a cycle.

VITA

JIANFENG PU

I was born and grew up in Hunan province in South China. I got my BS degree in electrical engineering in 1983, then worked for a local power utility company as an assistant electrical engineer. In 1988, I went to the graduate school of Electrical Power Research Institute in Beijing. After graduation, I joined the institute and worked on various power systems research projects. In 1993, I came to Virginia Tech for further graduate study in Electrical Engineering. I studied computer science in University of Massachusetts in Lowell and got MS degree from 1995-1996, then worked for BBN Technologies (part of GTE Internetworking) in Massachusetts on network research. Now I am a senior software engineer in Quarry Technologies, a spin-off startup of BBN, developing multi-service IP QoS router.