An introduction to coding theory

Adrish Banerjee

Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh
India

Mar. 13, 2017

Lecture #19: Automatic Repeat reQuest (ARQ) schemes

Outline of the lecture

- Introduction

Outline of the lecture

- Introduction
- Automatic Repeat reQuest (ARQ) and error detecting codes

Outline of the lecture

- Introduction
- Automatic Repeat reQuest (ARQ) and error detecting codes
- Protocols for Automatic Repeat Requests

Outline of the lecture

- Introduction
- Automatic Repeat reQuest (ARQ) and error detecting codes
- Protocols for Automatic Repeat Requests
- Performance Analysis of ARQ systems

Outline of the lecture

- Introduction
- Automatic Repeat reQuest (ARQ) and error detecting codes
- Protocols for Automatic Repeat Requests
- Performance Analysis of ARQ systems
- Hybrid ARQ

Introduction

Forward Error Correction (FEC):

- In *one-way* system, transmission takes place only in one direction, i.e. from transmitter to receiver.

Introduction

Forward Error Correction (FEC):

- In *one-way* system, transmission takes place only in one direction, i.e. from transmitter to receiver.
- The error correcting codes used in such a system are referred as *forward error correction (FEC)* codes.

Introduction

Automatic Repeat reQuest (ARQ):

- In *two-way* system, there exist a feedback path from the receiver to the transmitter.

Introduction

Automatic Repeat reQuest (ARQ):

- In *two-way* system, there exist a feedback path from the receiver to the transmitter.
- Error correction can be achieved for two-way system using error detection and retransmission, also known as *automatic repeat request (ARQ)*.

ARQ and error detecting codes

- Automatic repeat request schemes use an error detecting code to check whether certain parity constraints in the receiver are violated.

ARQ and error detecting codes

- Automatic repeat request schemes use an error detecting code to check whether certain parity constraints in the receiver are violated.
- Most commonly used error detecting codes are cyclic redundancy check (CRC) codes.

ARQ and error detecting codes

- Automatic repeat request schemes use an error detecting code to check whether certain parity constraints in the receiver are violated.
- Most commonly used error detecting codes are cyclic redundancy check (CRC) codes.
- A (n, k) CRC code is linear block code defined by a generator polynomial of degree $n - k$.

ARQ and error detecting codes

- Automatic repeat request schemes use an error detecting code to check whether certain parity constraints in the receiver are violated.
- Most commonly used error detecting codes are cyclic redundancy check (CRC) codes.
- A (n, k) CRC code is linear block code defined by a generator polynomial of degree $n - k$.
- Let the n bits in a frame be $[c_{n-1}, c_{n-2}, \cdots, c_1, c_0]$ and $n - k$ low order bits (c_0 through c_{n-k-1}) be the parity bits, then the parity bits are chosen such that the polynomial $c(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \cdots + c_1 x + c_0$ is divisible by g(x).

ARQ and error detecting codes

- Automatic repeat request schemes use an error detecting code to check whether certain parity constraints in the receiver are violated.
- Most commonly used error detecting codes are cyclic redundancy check (CRC) codes.
- A (n, k) CRC code is linear block code defined by a generator polynomial of degree $n - k$.
- Let the n bits in a frame be $[c_{n-1}, c_{n-2}, \cdots, c_1, c_0]$ and $n - k$ low order bits (c_0 through c_{n-k-1}) be the parity bits, then the parity bits are chosen such that the polynomial $c(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \cdots + c_1 x + c_0$ is divisible by g(x).
- Since c(x) and g(x) are polynomials with binary coefficients, all operations are defined modulo two.

Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to coding theory

CRC code: an example

- Consider the generator polynomial $g(x) = x^4 + x + 1$ for a (15,11) code.

CRC code: an example

- Consider the generator polynomial $g(x) = x^4 + x + 1$ for a (15,11) code.
- Let the information bits be [1 0 0 1 1 0 0 0 1 1 0], then the parity bits are chosen such that codeword polynomial $c(x) = x^{14} + x^{11} + x^{10} + x^6 + x^5 + c_3x^3 + c_2x^2 + c_1x + c_0$ is divisible by g(x).

CRC code: an example

- Consider the generator polynomial $g(x) = x^4 + x + 1$ for a (15,11) code.
- Let the information bits be [1 0 0 1 1 0 0 0 1 1 0], then the parity bits are chosen such that codeword polynomial $c(x) = x^{14} + x^{11} + x^{10} + x^6 + x^5 + c_3x^3 + c_2x^2 + c_1x + c_0$ is divisible by g(x).
- In this case $c_3 = c_1 = 1$ and $c_2 = c_0 = 0$.

CRC code: an example

- Consider the generator polynomial $g(x) = x^4 + x + 1$ for a (15,11) code.
- Let the information bits be [1 0 0 1 1 0 0 0 1 1 0], then the parity bits are chosen such that codeword polynomial $c(x) = x^{14} + x^{11} + x^{10} + x^6 + x^5 + c_3x^3 + c_2x^2 + c_1x + c_0$ is divisible by g(x).
- In this case $c_3 = c_1 = 1$ and $c_2 = c_0 = 0$.
- We can verify that $c(x) = g(x) \cdot (x^{10} + x^2 + x)$ and this choice of the parity bits is unique in the sense that results in a multiple of g(x).

CRC code: an example

- Consider the generator polynomial $g(x) = x^4 + x + 1$ for a (15,11) code.
- Let the information bits be [1 0 0 1 1 0 0 0 1 1 0], then the parity bits are chosen such that codeword polynomial $c(x) = x^{14} + x^{11} + x^{10} + x^6 + x^5 + c_3x^3 + c_2x^2 + c_1x + c_0$ is divisible by g(x).
- In this case $c_3 = c_1 = 1$ and $c_2 = c_0 = 0$.
- We can verify that $c(x) = g(x) \cdot (x^{10} + x^2 + x)$ and this choice of the parity bits is unique in the sense that results in a multiple of g(x).
- At the receiver, the received polynomial is checked whether it is divisible by g(x). If it is, the frame is declared error free, else it is declared as erroneous.

CRC codes in practice

- Examples of some popular CRC codes

Code	$g(x)$	n_{max}	d_{min}
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	2,047	4
CRC-ANSI	$x^{16} + x^{15} + x^2 + 1$	32,767	4
CRC-CCITT	$x^{16} + x^{12} + x^5 + 1$	32,767	4

Table: Properties of Three Standard CRC Codes

Protocols for ARQs

- In ARQ protocols, frames that are received in error needs to be retransmitted.

Protocols for ARQs

- In ARQ protocols, frames that are received in error needs to be retransmitted.
- Three types of ARQ protocols

Protocols for ARQs

- In ARQ protocols, frames that are received in error needs to be retransmitted.
- Three types of ARQ protocols
  - Stop and Wait ARQ

Protocols for ARQs

- In ARQ protocols, frames that are received in error needs to be retransmitted.
- Three types of ARQ protocols
  - Stop and Wait ARQ
  - Go-Back-N ARQ

Protocols for ARQs

- In ARQ protocols, frames that are received in error needs to be retransmitted.
- Three types of ARQ protocols
  - Stop and Wait ARQ
  - Go-Back-N ARQ
  - Selective Repeat ARQ

Protocols for ARQs

- In ARQ protocols, frames that are received in error needs to be retransmitted.
- Three types of ARQ protocols
  - Stop and Wait ARQ
  - Go-Back-N ARQ
  - Selective Repeat ARQ
- A positive acknowledgment (ACK) is sent by the receiver to indicate that an error free frame has been received.

Protocols for ARQs

- In ARQ protocols, frames that are received in error needs to be retransmitted.
- Three types of ARQ protocols
  - Stop and Wait ARQ
  - Go-Back-N ARQ
  - Selective Repeat ARQ
- A positive acknowledgment (ACK) is sent by the receiver to indicate that an error free frame has been received.
- A negative acknowledgment (NACK) is sent by the receiver to indicate that an erroneous frame has been received.

Protocols for ARQs

- An ACK or NACK is usually protected by an error detecting/correcting code.

Protocols for ARQs

- An ACK or NACK is usually protected by an error detecting/correcting code.
- If ACKs or NACKs are not received in a specified amount of time, the transmitter adopts a timeout strategy where it assumes as if the the frame is not correctly received.

Protocols for ARQs

- An ACK or NACK is usually protected by an error detecting/correcting code.
- If ACKs or NACKs are not received in a specified amount of time, the transmitter adopts a timeout strategy where it assumes as if the the frame is not correctly received.
- We assume that the frames transmitted on the forward link and the acknowledgments transmitted on the return link are all received in order in which they are sent.

Protocols for ARQs

- An ACK or NACK is usually protected by an error detecting/correcting code.
- If ACKs or NACKs are not received in a specified amount of time, the transmitter adopts a timeout strategy where it assumes as if the the frame is not correctly received.
- We assume that the frames transmitted on the forward link and the acknowledgments transmitted on the return link are all received in order in which they are sent.
- We assume that there is no undetected error in the feedback channel.

Stop-and-Wait ARQ

- In stop and wait ARQ the transmitter waits until it gets an acknowledgment on the sent frame before sending a new frame is transmitted.

Stop-and-Wait ARQ

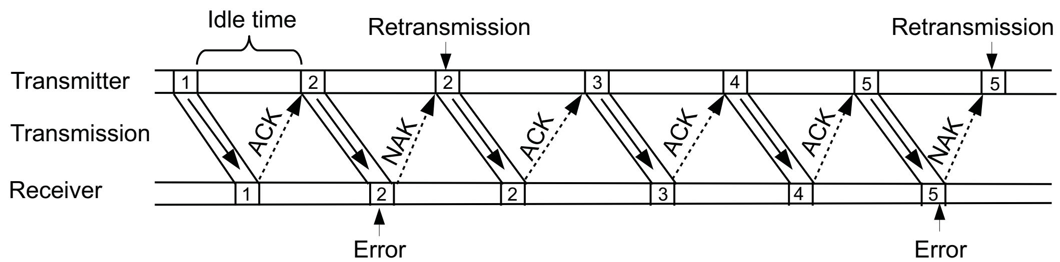
- In stop and wait ARQ the transmitter waits until it gets an acknowledgment on the sent frame before sending a new frame is transmitted.
- If a NACK is received the same frame is retransmitted else a new frame is transmitted.

Stop-and-Wait ARQ

- In stop and wait ARQ the transmitter waits until it gets an acknowledgment on the sent frame before sending a new frame is transmitted.
- If a NACK is received the same frame is retransmitted else a new frame is transmitted.
- In this protocol, the transmitter and receiver take turns in sending frames and acknowledgments respectively.

Stop-and-Wait ARQ

- In stop and wait ARQ the transmitter waits until it gets an acknowledgment on the sent frame before sending a new frame is transmitted.
- If a NACK is received the same frame is retransmitted else a new frame is transmitted.
- In this protocol, the transmitter and receiver take turns in sending frames and acknowledgments respectively.
- To avoid potential confusion caused by lost frame, a sequence number is often included in the header of the frame and acknowledgments.



Go-Back-N ARQ

- Unlike Stop and Wait ARQ, in Go-Back-N ARQ, the transmitter doesn't have to wait for an acknowledgment before sending another frame.

Go-Back-N ARQ

- Unlike Stop and Wait ARQ, in Go-Back-N ARQ, the transmitter doesn't have to wait for an acknowledgment before sending another frame.
- Transmitter can send all the frames in a window maintained at the transmitter.

Go-Back-N ARQ

- Unlike Stop and Wait ARQ, in Go-Back-N ARQ, the transmitter doesn't have to wait for an acknowledgment before sending another frame.
- Transmitter can send all the frames in a window maintained at the transmitter.
- If the first frame in the window is acknowledged, the transmitter slides the window one position, dropping the acknowledged frame and adding a new frame that it then transmits.

Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to coding theory

Go-Back-N ARQ

- Unlike Stop and Wait ARQ, in Go-Back-N ARQ, the transmitter doesn't have to wait for an acknowledgment before sending another frame.
- Transmitter can send all the frames in a window maintained at the transmitter.
- If the first frame in the window is acknowledged, the transmitter slides the window one position, dropping the acknowledged frame and adding a new frame that it then transmits.
- If the first frame in the window is negatively acknowledged, the transmitter backs up and resends all the frames in the window starting with the received erroneous frame.

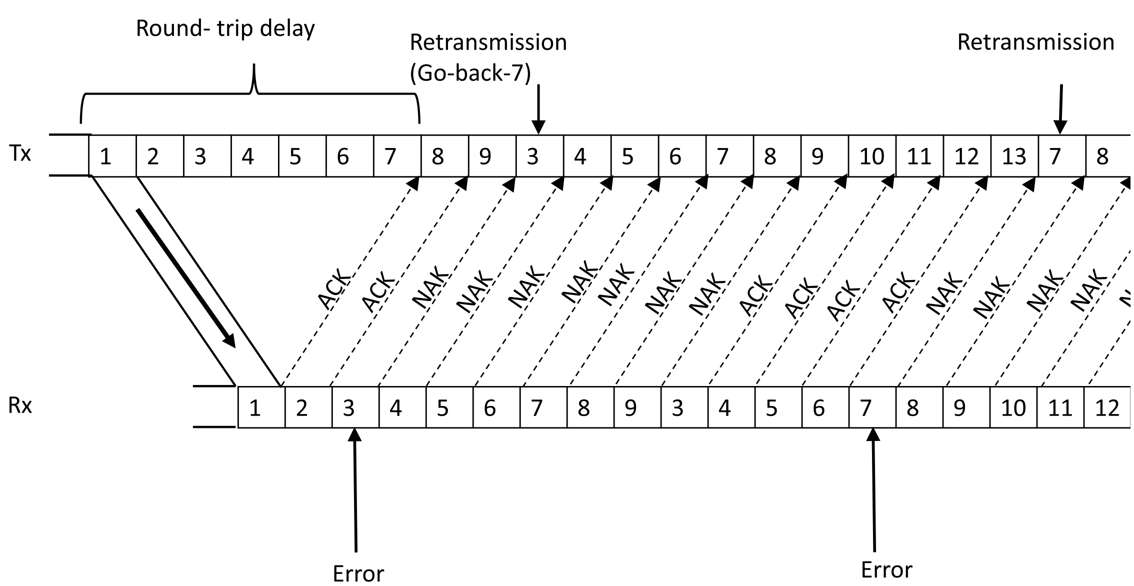
Go-Back-N ARQ

- Unlike Stop and Wait ARQ, in Go-Back-N ARQ, the transmitter doesn't have to wait for an acknowledgment before sending another frame.
- Transmitter can send all the frames in a window maintained at the transmitter.
- If the first frame in the window is acknowledged, the transmitter slides the window one position, dropping the acknowledged frame and adding a new frame that it then transmits.
- If the first frame in the window is negatively acknowledged, the transmitter backs up and resends all the frames in the window starting with the received erroneous frame.
- Go-back-N is often referred as sliding window protocol because the transmitter maintains a sliding window of active frames that have been transmitted but not yet acknowledged.

Go-Back-N ARQ

- Unlike Stop and Wait ARQ, in Go-Back-N ARQ, the transmitter doesn't have to wait for an acknowledgment before sending another frame.
- Transmitter can send all the frames in a window maintained at the transmitter.
- If the first frame in the window is acknowledged, the transmitter slides the window one position, dropping the acknowledged frame and adding a new frame that it then transmits.
- If the first frame in the window is negatively acknowledged, the transmitter backs up and resends all the frames in the window starting with the received erroneous frame.
- Go-back-N is often referred as sliding window protocol because the transmitter maintains a sliding window of active frames that have been transmitted but not yet acknowledged.
- Appropriate choice of N is governed by the propagation delay of the system and the length of the frames.

Go-Back-N ARQ

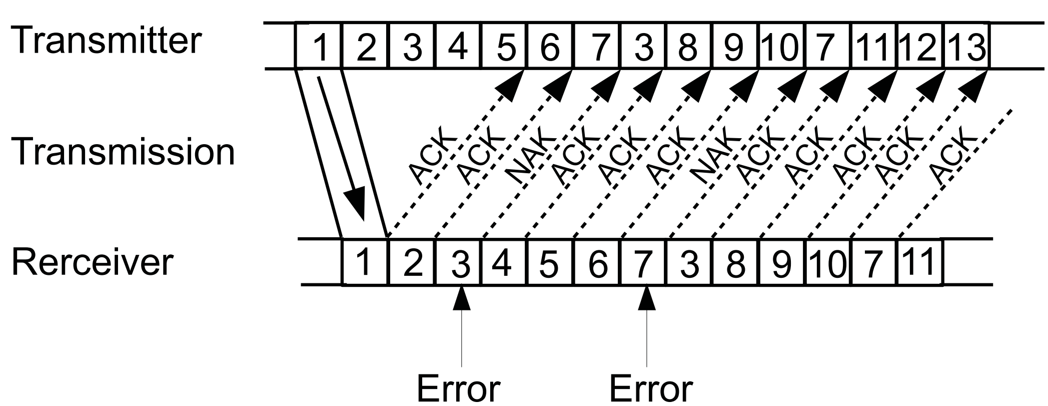


Selective Repeat ARQ

- Unlike Go-back-N ARQ, where all the frames in a window following a frame received in error are retransmitted, in selective repeat ARQ only those frames that are received in error are retransmitted.

Selective Repeat ARQ

- Unlike Go-back-N ARQ, where all the frames in a window following a frame received in error are retransmitted, in selective repeat ARQ only those frames that are received in error are retransmitted.
- However, here one has to store the arriving frame in a buffer and reorder them in the correct order.



Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to coding theory

Reliability analysis of an ARQ system

- The reliability of an ARQ scheme can be evaluated using frame error rate.

Reliability analysis of an ARQ system

- The reliability of an ARQ scheme can be evaluated using frame error rate.
- The coded sequences can be broken up into blocks of data *frames*.

Reliability analysis of an ARQ system

- The reliability of an ARQ scheme can be evaluated using frame error rate.
- The coded sequences can be broken up into blocks of data *frames*.
- A frame error occurs if any information bit in that data frame is in error. The decoded FER is the percentage of frames in error.

Reliability analysis of an ARQ system

- The reliability of an ARQ scheme can be evaluated using frame error rate.
- The coded sequences can be broken up into blocks of data *frames*.
- A frame error occurs if any information bit in that data frame is in error. The decoded FER is the percentage of frames in error.
- When a frame is transmitted, one of the following things can happen

Reliability analysis of an ARQ system

- The reliability of an ARQ scheme can be evaluated using frame error rate.
- The coded sequences can be broken up into blocks of data *frames*.
- A frame error occurs if any information bit in that data frame is in error. The decoded FER is the percentage of frames in error.
- When a frame is transmitted, one of the following things can happen
  - Frame is received error free.

Reliability analysis of an ARQ system

- The reliability of an ARQ scheme can be evaluated using frame error rate.
- The coded sequences can be broken up into blocks of data *frames*.
- A frame error occurs if any information bit in that data frame is in error. The decoded FER is the percentage of frames in error.
- When a frame is transmitted, one of the following things can happen
  - Frame is received error free.
  - Frame is received in error and it is detected by the error detecting code.

Reliability analysis of an ARQ system

- The reliability of an ARQ scheme can be evaluated using frame error rate.
- The coded sequences can be broken up into blocks of data *frames*.
- A frame error occurs if any information bit in that data frame is in error. The decoded FER is the percentage of frames in error.
- When a frame is transmitted, one of the following things can happen
  - Frame is received error free.
  - Frame is received in error and it is detected by the error detecting code.
  - Frame is received in error, however the error detecting code is unable to detect it.

Reliability analysis of an ARQ system

- Let P_d be the probability that a frame is received in error and correctly detected by the error detecting code.

Reliability analysis of an ARQ system

- Let P_d be the probability that a frame is received in error and correctly detected by the error detecting code.
- Let P_u be the probability that a frame is received in error and the error detecting code is unable to detect it.

Reliability analysis of an ARQ system

- Let P_d be the probability that a frame is received in error and correctly detected by the error detecting code.
- Let P_u be the probability that a frame is received in error and the error detecting code is unable to detect it.
- Assuming there is no limit on retransmissions, the frame error rate is given by

$$\begin{aligned}\text{FER} &= P_u + P_d P_u + P_d^2 P_u + P_d^3 P_u \cdots + \\ &= P_u \sum_{i=0}^{\infty} P_d^i = \frac{P_u}{1 - P_d}\end{aligned}$$

Reliability analysis of an ARQ system

- Let P_d be the probability that a frame is received in error and correctly detected by the error detecting code.
- Let P_u be the probability that a frame is received in error and the error detecting code is unable to detect it.
- Assuming there is no limit on retransmissions, the frame error rate is given by

$$\begin{aligned}\text{FER} &= P_u + P_d P_u + P_d^2 P_u + P_d^3 P_u \cdots + \\ &= P_u \sum_{i=0}^{\infty} P_d^i = \frac{P_u}{1 - P_d}\end{aligned}$$

- This analysis is true for any ARQ protocol.

Reliability analysis of an ARQ system

- Let P_d be the probability that a frame is received in error and correctly detected by the error detecting code.
- Let P_u be the probability that a frame is received in error and the error detecting code is unable to detect it.
- Assuming there is no limit on retransmissions, the frame error rate is given by

$$\begin{aligned}\text{FER} &= P_u + P_d P_u + P_d^2 P_u + P_d^3 P_u \cdots + \\ &= P_u \sum_{i=0}^{\infty} P_d^i = \frac{P_u}{1 - P_d}\end{aligned}$$

- This analysis is true for any ARQ protocol.
- Reliability of an ARQ protocol depends only on the error detecting capability of the underlying error detecting code and is independent of how many corrupted frames are subsequently retransmitted.

Efficiency analysis of an ARQ system

- Efficiency of an ARQ scheme is strongly related to the type of protocol used for ARQ.

Efficiency analysis of an ARQ system

- Efficiency of an ARQ scheme is strongly related to the type of protocol used for ARQ.
- Efficiency of an ARQ scheme is measured by its throughput.

Efficiency analysis of an ARQ system

- Efficiency of an ARQ scheme is strongly related to the type of protocol used for ARQ.
- Efficiency of an ARQ scheme is measured by its throughput.
- The throughput of an ARQ scheme is defined as the ratio of the average number of information bits accepted by the receiver per unit time to the number of bits that can be transmitted over the channel per unit time.

Efficiency analysis of an ARQ system

- Efficiency of an ARQ scheme is strongly related to the type of protocol used for ARQ.
- Efficiency of an ARQ scheme is measured by its throughput.
- The throughput of an ARQ scheme is defined as the ratio of the average number of information bits accepted by the receiver per unit time to the number of bits that can be transmitted over the channel per unit time.
- In selective-repeat ARQ each time a frame is transmitted, it has a probability $P = P_c + P_u$ of being accepted by the receiver where P_c denotes the probability that the frame is delivered error free to the receiver and P_u denotes the probability that the frame is in error but the error detecting code is unable to detect it.

Efficiency analysis: SR-ARQ

- Let T_{SR} be a random variable equal to the number of frame transmissions required for SR ARQ until a particular frame is accepted, then $Prob(T_{SR} = 1) = P, Prob(T_{SR} = 2) = P(1 - P), Prob(T_{SR} = 3) = P(1 - P)^2$ and so on.

Efficiency analysis: SR-ARQ

- Let T_{SR} be a random variable equal to the number of frame transmissions required for SR ARQ until a particular frame is accepted, then $Prob(T_{SR} = 1) = P, Prob(T_{SR} = 2) = P(1 - P), Prob(T_{SR} = 3) = P(1 - P)^2$ and so on.
- Thus T_{SR} has a geometric distribution with mean $E[T_{SR}] = 1/P$.

Efficiency analysis: SR-ARQ

- Let T_{SR} be a random variable equal to the number of frame transmissions required for SR ARQ until a particular frame is accepted, then $Prob(T_{SR} = 1) = P, Prob(T_{SR} = 2) = P(1 - P), Prob(T_{SR} = 3) = P(1 - P)^2$ and so on.
- Thus T_{SR} has a geometric distribution with mean $E[T_{SR}] = 1/P$.
- If we use a (n,k) error detecting code, the receiver accepts k information bits in the amount of time (on average) it takes to transmit n/P over the channel.

Efficiency analysis: SR-ARQ

- Let T_{SR} be a random variable equal to the number of frame transmissions required for SR ARQ until a particular frame is accepted, then $Prob(T_{SR} = 1) = P, Prob(T_{SR} = 2) = P(1 - P), Prob(T_{SR} = 3) = P(1 - P)^2$ and so on.
- Thus T_{SR} has a geometric distribution with mean $E[T_{SR}] = 1/P$.
- If we use a (n,k) error detecting code, the receiver accepts k information bits in the amount of time (on average) it takes to transmit n/P over the channel.
- Thus the throughput for selective-repeat ARQ is given by

$$\eta_{SR} = \frac{k}{n}P$$

Efficiency analysis: Go-back-N ARQ

- Let $P = P_c + P_u$ be the probability of being accepted by the receiver where P_c denotes the probability that the frame is delivered error free to the receiver and P_u denotes the probability that the frame is in error but the error detecting code is unable to detect it.

Efficiency analysis: Go-back-N ARQ

- Let $P = P_c + P_u$ be the probability of being accepted by the receiver where P_c denotes the probability that the frame is delivered error free to the receiver and P_u denotes the probability that the frame is in error but the error detecting code is unable to detect it.
- Let T_{GBN} be a random variable equal to the number of frame transmissions required in Go-back-N ARQ until a particular frame is accepted, assuming that a full window of N frames must be retransmitted every time a frame is rejected, we have

$$\begin{aligned} E[T_{GBN}] =& 1 \cdot P + (N+1) \cdot (1-P) \cdot P \\ & + (2N+1) \cdot (1-P)^2 \cdot P + \cdots \\ =& 1 + \frac{N(1-P)}{P} \end{aligned}$$

Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to coding theory

Efficiency analysis: Go-back-N ARQ (contd.)

- The receiver accepts k information bits in the amount of time (on average) it takes to transmit $n(1 + N(1 - P)/P)$ bits over the channel, so the throughput is given by

$$\eta_{GBN} = \left(\frac{k}{n}\right)\left(\frac{P}{P + N(1 - P)}\right)$$

Efficiency analysis: SR-ARQ

- Let β denote the amount of time the transmitter is idle measured in frame units.

Efficiency analysis: SR-ARQ

- Let β denote the amount of time the transmitter is idle measured in frame units.
- If a frame is accepted the first time, the delay is $1+\beta$ frame units, if the frame is rejected first time, but accepted the second time, the delay is $2(1+\beta)$ frame units and so on.

Efficiency analysis: SR-ARQ

- Let β denote the amount of time the transmitter is idle measured in frame units.
- If a frame is accepted the first time, the delay is $1+\beta$ frame units, if the frame is rejected first time, but accepted the second time, the delay is $2(1+\beta)$ frame units and so on.
- Let T_{SW} be the random variable that represents the delay in terms of frame units incurred from the time a frame is first sent to the time it is finally accepted, we have

$$\begin{aligned} E[T_{SW}] =& (1+\beta)\cdot P + 2\cdot(1+\beta)\cdot(1-P)\cdot P \\ &+ 3\cdot(1+\beta)\cdot(1-P)^2\cdot P + \cdots \\ =& \frac{1+\beta}{P} \end{aligned}$$

Efficiency analysis: SR-ARQ

- Let β denote the amount of time the transmitter is idle measured in frame units.
- If a frame is accepted the first time, the delay is $1+\beta$ frame units, if the frame is rejected first time, but accepted the second time, the delay is $2(1+\beta)$ frame units and so on.
- Let T_{SW} be the random variable that represents the delay in terms of frame units incurred from the time a frame is first sent to the time it is finally accepted, we have

$$\begin{aligned} E[T_{SW}] =& (1+\beta)\cdot P + 2\cdot(1+\beta)\cdot(1-P)\cdot P \\ &+ 3\cdot(1+\beta)\cdot(1-P)^2\cdot P + \cdots \\ =& \frac{1+\beta}{P} \end{aligned}$$

- So the throughput is given by

$$\eta_{SW} = \left(\frac{k}{n}\right)\left(\frac{P}{1+\beta}\right)$$

Efficiency analysis: SR-ARQ

- Let β denote the amount of time the transmitter is idle measured in frame units.
- If a frame is accepted the first time, the delay is $1+\beta$ frame units, if the frame is rejected first time, but accepted the second time, the delay is $2(1+\beta)$ frame units and so on.
- Let T_{SW} be the random variable that represents the delay in terms of frame units incurred from the time a frame is first sent to the time it is finally accepted, we have

$$\begin{aligned} E[T_{SW}] =& (1+\beta)\cdot P + 2\cdot(1+\beta)\cdot(1-P)\cdot P \\ &+ 3\cdot(1+\beta)\cdot(1-P)^2\cdot P + \cdots \\ =& \frac{1+\beta}{P} \end{aligned}$$

- So the throughput is given by

$$\eta_{SW} = \left(\frac{k}{n}\right)\left(\frac{P}{1+\beta}\right)$$

Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to coding theory

Hybrid Automatic Repeat reQuest (HARQ)

- For channels with feedback, one can use ARQ protocols in combination with FEC to improve system performance. These types of schemes are known as hybrid-ARQ (HARQ) schemes.

Hybrid Automatic Repeat reQuest (HARQ)

- For channels with feedback, one can use ARQ protocols in combination with FEC to improve system performance. These types of schemes are known as hybrid-ARQ (HARQ) schemes.
- In HARQ protocols, the transmitted data is encoded for both error correction and error detection. If the receiver detects an error after decoding, it sends a negative acknowledgment (NACK) to the transmitter, which then retransmits the data.

Hybrid ARQ systems

- Hybrid ARQ schemes can be classified as

Hybrid ARQ systems

- Hybrid ARQ schemes can be classified as
  - Type 1 Hybrid ARQ

Hybrid ARQ systems

- Hybrid ARQ schemes can be classified as
  - Type 1 Hybrid ARQ
  - Type 2 Hybrid ARQ

Type 1 Hybrid ARQ systems

- A simple implementation of Type 1 Hybrid ARQ uses a high rate error detection code and a lower rate error correction code.

Type 1 Hybrid ARQ systems

- A simple implementation of Type 1 Hybrid ARQ uses a high rate error detection code and a lower rate error correction code.
- Information sequence is first encoded using an error detection code and then using an error correction code.

Type 1 Hybrid ARQ systems

- A simple implementation of Type 1 Hybrid ARQ uses a high rate error detection code and a lower rate error correction code.
- Information sequence is first encoded using an error detection code and then using an error correction code.
- At the receiver, the error correcting code first tries to reconstruct the information sequence and then passes the information to the error detection code.

Type 1 Hybrid ARQ systems

- A simple implementation of Type 1 Hybrid ARQ uses a high rate error detection code and a lower rate error correction code.
- Information sequence is first encoded using an error detection code and then using an error correction code.
- At the receiver, the error correcting code first tries to reconstruct the information sequence and then passes the information to the error detection code.
- If the error detection code observes a parity violation, a retransmission request is initiated. Otherwise the packet is accepted.

Type 1 Hybrid ARQ systems

- A simple implementation of Type 1 Hybrid ARQ uses a high rate error detection code and a lower rate error correction code.
- Information sequence is first encoded using an error detection code and then using an error correction code.
- At the receiver, the error correcting code first tries to reconstruct the information sequence and then passes the information to the error detection code.
- If the error detection code observes a parity violation, a retransmission request is initiated. Otherwise the packet is accepted.
- Usually the same information sequence is retransmitted.

Type 2 Hybrid ARQ systems

- Type 2 Hybrid ARQ uses the principle of incremental redundancy.

Type 2 Hybrid ARQ systems

- Type 2 Hybrid ARQ uses the principle of incremental redundancy.
- Information sequence is first encoded using an error detection code.

Type 2 Hybrid ARQ systems

- Type 2 Hybrid ARQ uses the principle of incremental redundancy.
- Information sequence is first encoded using an error detection code.
- At the receiver, the error detection code checks whether there are errors in the received sequence.

Type 2 Hybrid ARQ systems

- Type 2 Hybrid ARQ uses the principle of incremental redundancy.
- Information sequence is first encoded using an error detection code.
- At the receiver, the error detection code checks whether there are errors in the received sequence.
- If the error detection code observes a parity violation, a retransmission request is initiated.

Type 2 Hybrid ARQ systems

- Type 2 Hybrid ARQ uses the principle of incremental redundancy.
- Information sequence is first encoded using an error detection code.
- At the receiver, the error detection code checks whether there are errors in the received sequence.
- If the error detection code observes a parity violation, a retransmission request is initiated.
- During retransmission, parity bits based on the original information bits is transmitted.

Type 2 Hybrid ARQ systems

- Type 2 Hybrid ARQ uses the principle of incremental redundancy.
- Information sequence is first encoded using an error detection code.
- At the receiver, the error detection code checks whether there are errors in the received sequence.
- If the error detection code observes a parity violation, a retransmission request is initiated.
- During retransmission, parity bits based on the original information bits is transmitted.
- Original received packet that was received in error is stored in a buffer and is augmented with the newly received parity bits to form a lower rate more powerful forward error correcting code.