Errors, Error Detection, and Error Control

Learning Objectives

- Identify the different types of noise commonly found in computer networks
- Specify the different error prevention techniques and be able to apply an error prevention technique to a type of noise
- Compare the different error detection techniques in terms of efficiency and efficacy
- Perform simple parity and longitudinal parity calculations, and enumerate their strengths and weaknesses
- Cite the advantages of cyclic redundancy checksum, and specify what types of errors cyclic redundancy checksum will detect
- Differentiate the three basic forms of error control, and describe under what circumstances each may be used
- Follow an example of stop-and-wait ARQ, go-back-N ARQ, and selective-reject ARQ.

Introduction

- All transmitted signals will contain some rate of errors (>0%) because noise is always present
- If a communications line experiences too much noise, the signal will be lost or corrupted
  - Once an error is detected, a system may perform some action, or may do nothing but simply let the data in error be discarded
- Popular error detection methods:
  - Simple Parity (add a 1 or 0 to the end of each seven bits)
  - Longitudinal Parity or Longitudinal Redundancy Check (LRC)
  - Cyclic Redundancy Checksum (CRC), or called Polynomial checking

What’s an “error”?

- **Human errors:**
  - **Incorrect IP address assignment, or subnet mask, or input errors etc.**
  - **Not what we consider in this class**
- **Network errors:**
  - **Lost data**
  - **Corrupted data (received, but garbled)**
    - **Misinformation is worse than no information!**

Line Noise and Distortion Errors

Source	Likely cause
Line outage	Storm, accident
White noise	Movement of electrons
Impulse noise	Random spikes of power
Cross-talk	Guardbands, wires too close
Echo	Reflective feedback
Jitter	Timing irregularities
Delay distortion	Propagation speed
Attenuation	Wires too long

Noise and Errors

Noise and Errors – White Noise

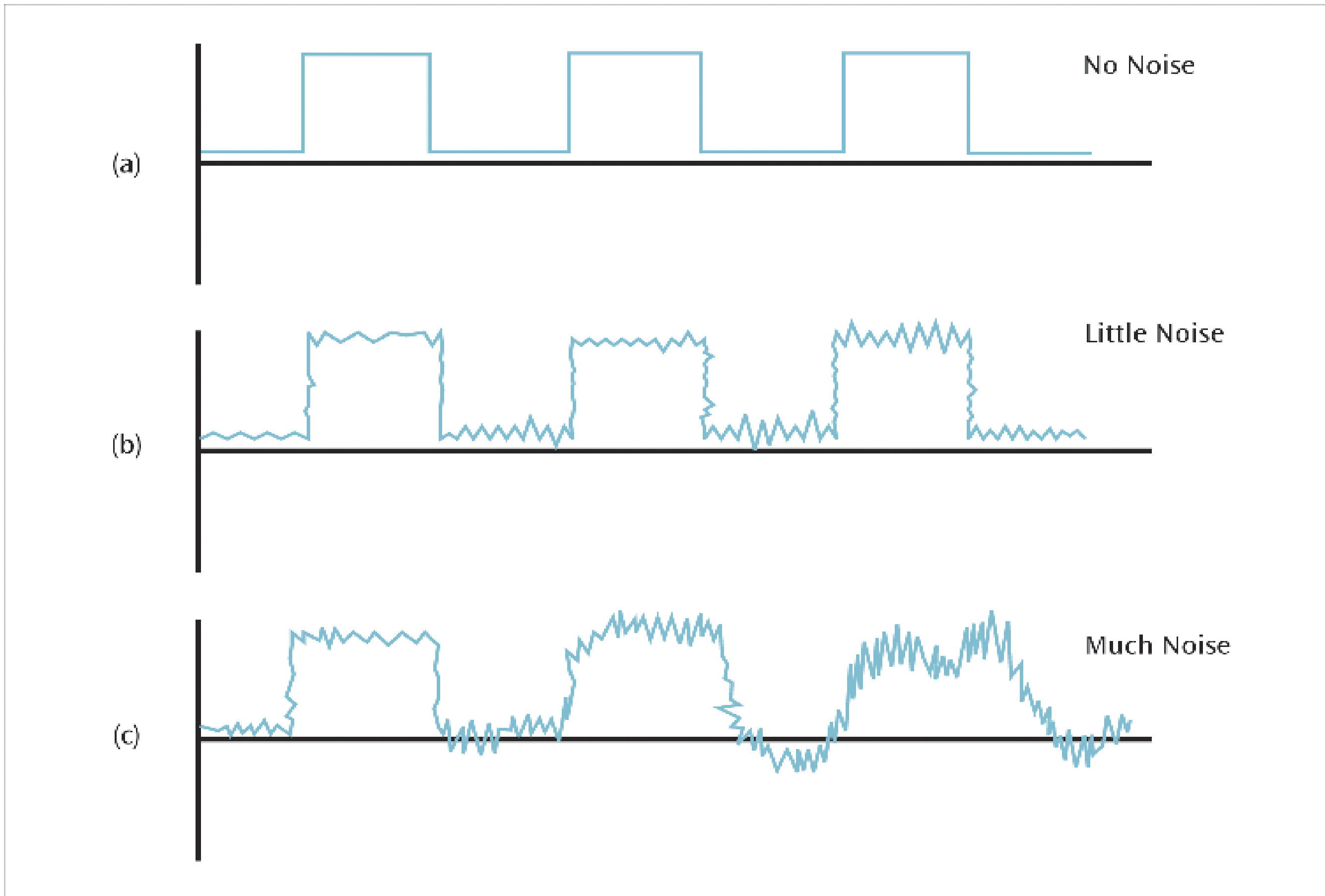
Also known as thermal or Gaussian noise

Relatively constant and can be reduced.

If white noise gets to strong, it can completely disrupt the signal.

Noise and Errors

Figure 6-1
White noise as it interferes with a digital signal



Noise and Errors

Noise and Errors – Impulse Noise

One of the most disruptive forms of noise.

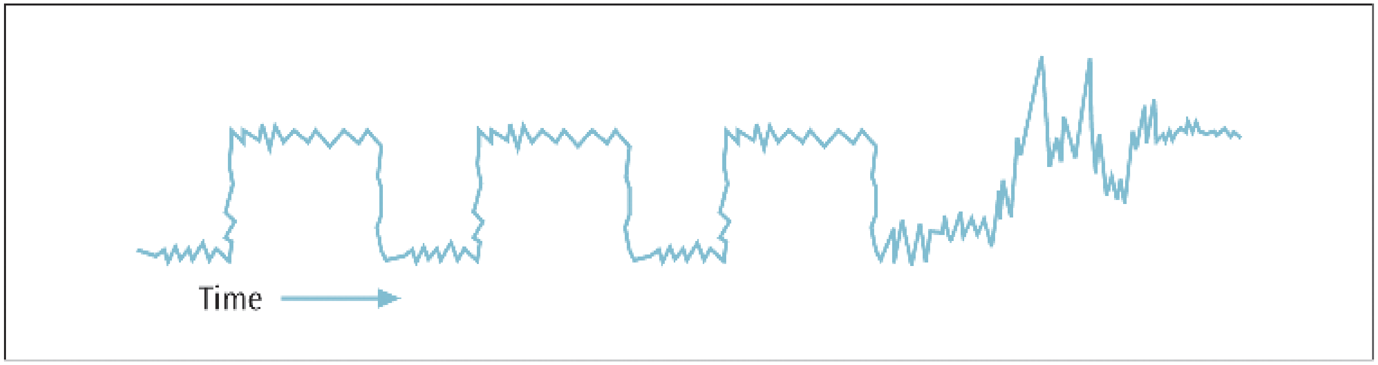
Random spikes of power that can destroy one or more bits of information.

Difficult to remove from an analog signal because it may be hard to distinguish from the original signal.

Impulse noise can damage more bits if the bits are closer together (transmitted at a faster rate).

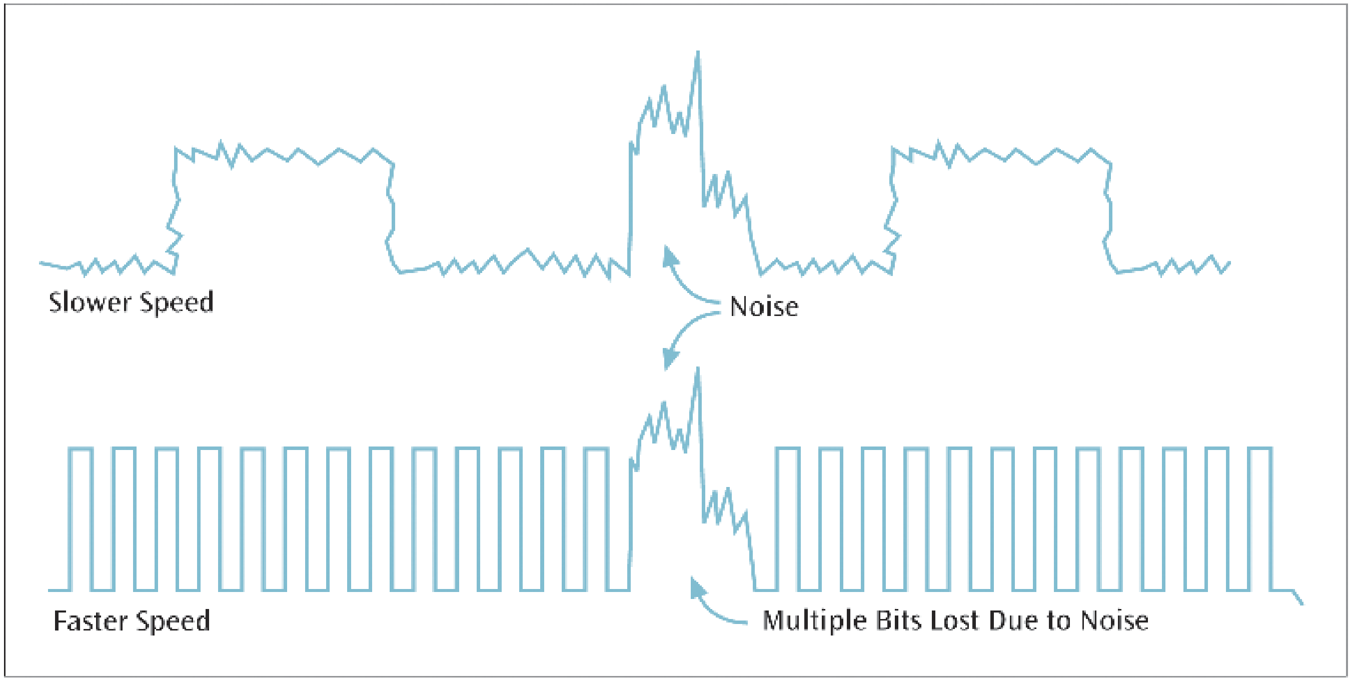
Noise and Errors

Figure 6-2
The effect of impulse noise on a digital signal



Noise and Errors

Figure 6-3
Transmission speed and its relationship to noise in a digital signal



Noise and Errors

Noise and Errors - Crosstalk

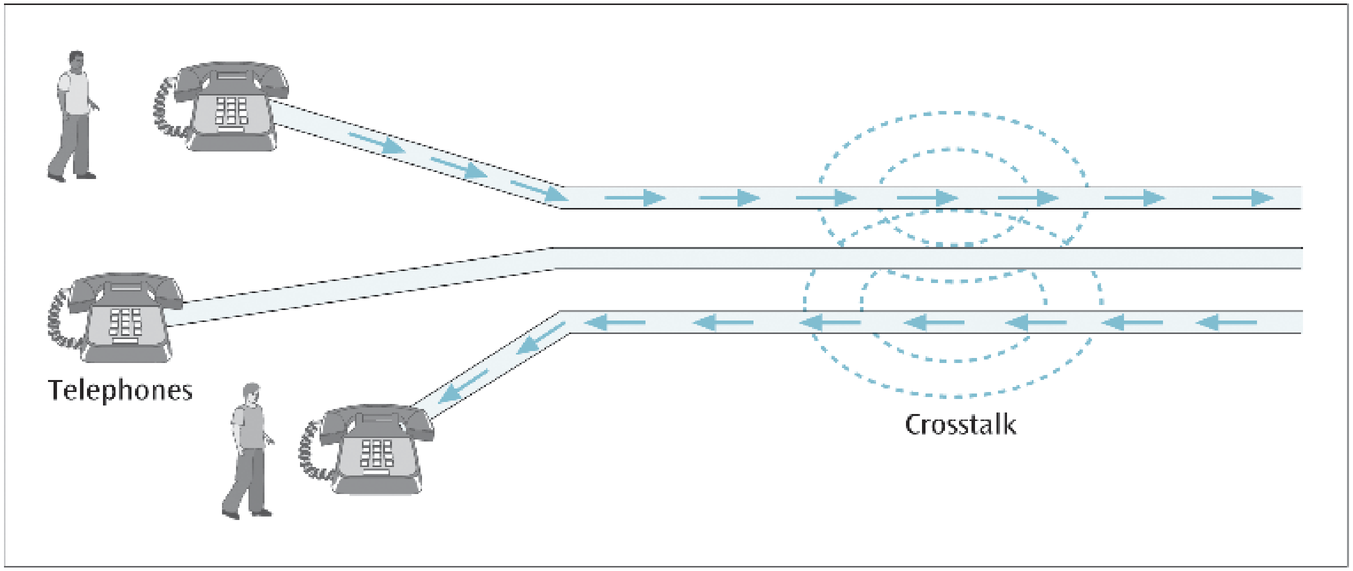
Unwanted coupling between two different signal paths.

For example, hearing another conversation while talking on the telephone.

Relatively constant and can be reduced with proper measures.

Noise and Errors

Figure 6-4
Three telephone circuits experiencing crosstalk



Noise and Errors

Noise and Errors - Echo

The reflective feedback of a transmitted signal as the signal moves through a medium.

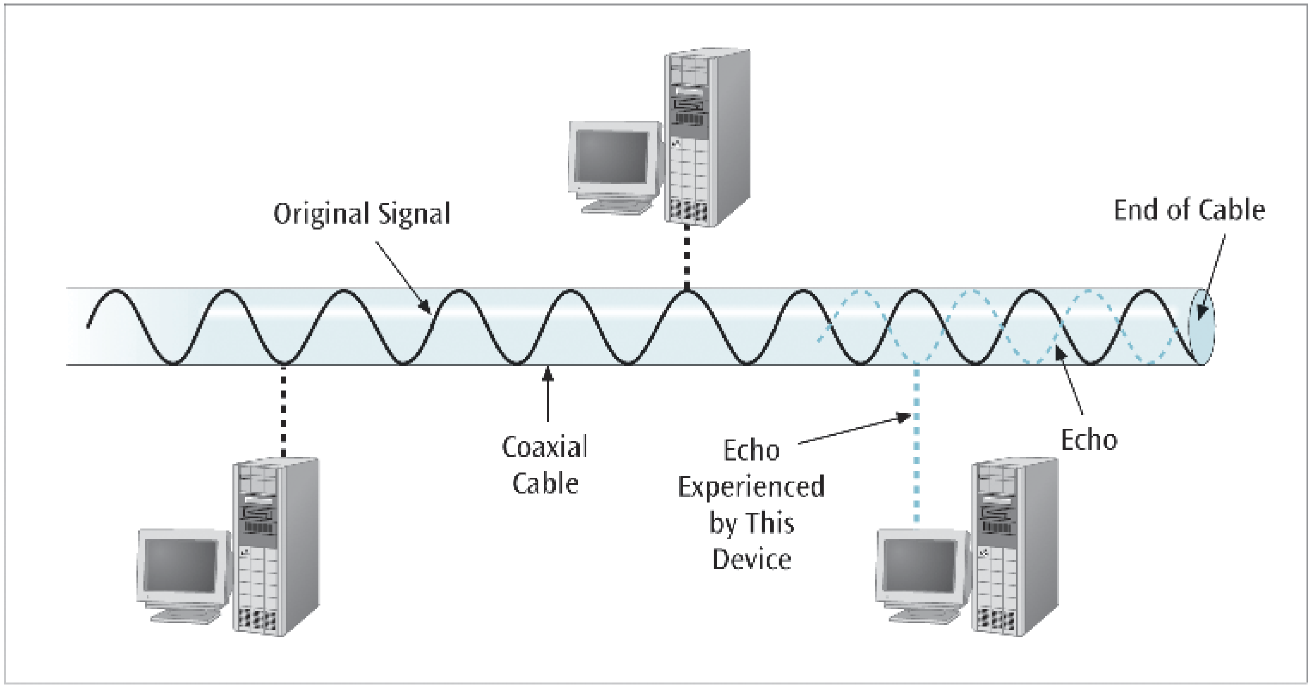
Most often occurs on coaxial cable.

If echo bad enough, it could interfere with original signal.

Relatively constant, and can be significantly reduced.

Noise and Errors

Figure 6-5
A signal bouncing back at the end of a cable and causing echo



Noise and Errors

Noise and Errors - Jitter

The result of small timing irregularities during the transmission of digital signals.

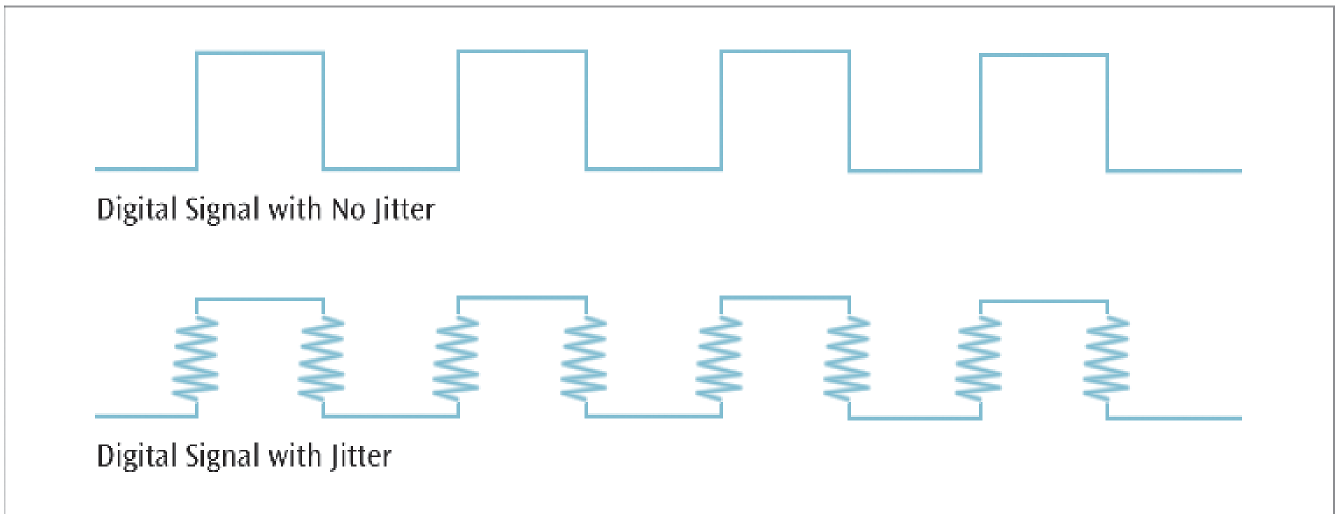
Occurs when a digital signal is repeater over and over.

If serious enough, jitter forces systems to slow down their transmission.

Steps can be taken to reduce jitter.

Noise and Errors

Figure 6-6
Original digital signal and digital signal with jitter



Noise and Errors

Noise and Errors – Delay Distortion

Occurs because the velocity of propagation of a signal through a medium varies with the frequency of the signal. Can be reduced.

Attenuation (not an error, but indirectly affects error)

The continuous loss of a signal’s strength as it travels through a medium.

Error Prevention

- To prevent errors from happening, several techniques may be applied:
  - Proper shielding of cables to reduce interference
  - Telephone line conditioning or equalization
  - Replacing older media and equipment with new, possibly digital components
  - Proper use of digital repeaters and analog amplifiers
  - Observe the stated capacities of the media

Error Prevention

Table 6-1

Summary of errors and error-prevention techniques

Type of Error	Error-Prevention Technique
White noise	Filters for analog signals; signal regeneration for digital signals
Impulse noise	Special filters for analog signals; digital signal processing for digital signals
Crosstalk	Proper shielding of cables
Echo	Proper termination of cables
Jitter	Better-quality electronic circuitry, fewer repeaters, slowing the transmission
Delay distortion	Added circuitry that equalizes the transmission speeds of the different frequencies
Attenuation*	Amplification of analog signals; regeneration of digital signals

* Not a type of error, but indirectly affects error

Error Detection Methods

- The only way to do error detection and correction is to send extra data with each message
- Two common error detection methods:
  - Parity checking
    - Simple parity
    - Longitudinal parity or Longitudinal Redundancy Check (LRC)
  - Polynomial Checking
    - Cyclic Redundancy Checksum (CRC)

Simple Parity

- Occasionally known as vertical redundancy check
- Add an additional bit to a string of bits:
  - Even parity
    - The 0 or 1 added to the string produces an even number of 1s (including the parity bit)
  - Odd parity
    - The 0 or 1 added … an odd number of 1s (including the parity bit)

0	1	0	1	0	1	0	1	Even parity
							0	Odd parity

Example

Letter	7-bit ASCII							Parity bit
D	1	0	0	0	1	0	0	1
A	1	0	0	0	0	0	1	1
T	1	0	1	0	1	0	0	0
A	1	0	0	0	0	0	1	1

Even Parity or Odd Parity?

Reliability/Efficiency

- What happens if the character 10010101 is sent and the first two 0s accidentally become two 1s?
- Thus, the following character is received: 11110101.
- Will there be a parity error?
- Problem: Simple parity only detects odd numbers of bits in error
  - Isolated single-bit errors occur 50-60% of the time
  - Error bursts occur 10-20% of the time
- Efficiency: 1:7 (parity:data) w/ mediocre error-detection

Longitudinal Parity

- Sometimes called LRC or Horizontal Parity
- Add Block Check Character (BCC) after a block of characters:
  - Perform simple parity checking on each row of bits
  - Then create a row of parity bits, or a BCC: each parity bit in this last row is a parity check for all the bits in the column

	0	1	0	1	0	1	0	1
	1	0	0	1	1	1	0	0
BCC	1	1	0	0	1	0	0	1

Even Parity or Odd Parity?

Example

Letter	7-bit ASCII							Parity bit
D	1	0	0	0	1	0	0	1
A	1	0	0	0	0	0	1	1
T	1	0	1	0	1	0	0	0
A	1	0	0	0	0	0	1	1
BCC	1	1	0	1	1	1	1	1

Even Parity or Odd Parity?

Reliability/Efficiency

- Longitudinal parity is better at catching errors
  - But requires too many check bits added to a block of data
- But still…

Table 6-3

The second and third bits in Rows 1 and 2 have errors, but longitudinal parity does not detect the errors

	Data							Parity
Row 1	1	~~1~~ 0	~~0~~ 1	1	0	1	1	1
Row 2	1	~~1~~ 0	~~1~~ 0	1	1	1	1	1
Row 3	0	1	0	1	0	1	0	1
Row 4	0	0	1	1	0	0	1	1
Parity Row	0	1	0	0	1	1	1	0

- Efficiency: (n+8) : 7n (check bits: data) w/ a little bit better than mediocre error-detection
- We need a better error detection method

Polynomial Checking

- Adds a character to the end of the data based on a mathematical algorithm:
  - Checksum
    - E.g., Sum the data values and divide the sum by 255. The remainder is the checksum

D	68
A	65
T	84
A	65

$$\frac{282}{255} = 1 \quad \text{remainder} \quad 27$$

Checksum	0	0	1	1	0	1	1	1

Cyclical Redundancy Check (CRC)

- CRC error detection method treats packet of data to be transmitted as a large polynomial
- Transmitter
  - Using polynomial arithmetic, divides polynomial by a given generating polynomial
- Quotient is discarded
  - Remainder is “attached” to the end of data
- Data (with the remainder) is transmitted to the receiver
- Receiver divides the (data – remainder) by the same generating polynomial
- If a remainder of zero results → no error during transmission
- If a remainder not equal to zero results → error during transmission

Example: CRC

7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	1

Message polynomial

$x^5+x^4+x^2+x^1+x^0 \equiv x^5+x^4+x^2+x+1$

Generating polynomial

ATM CRC $x^8 + x^2 + x + 1$

CRC-16 $x^{16} + x^{15} + x^2 + 1$

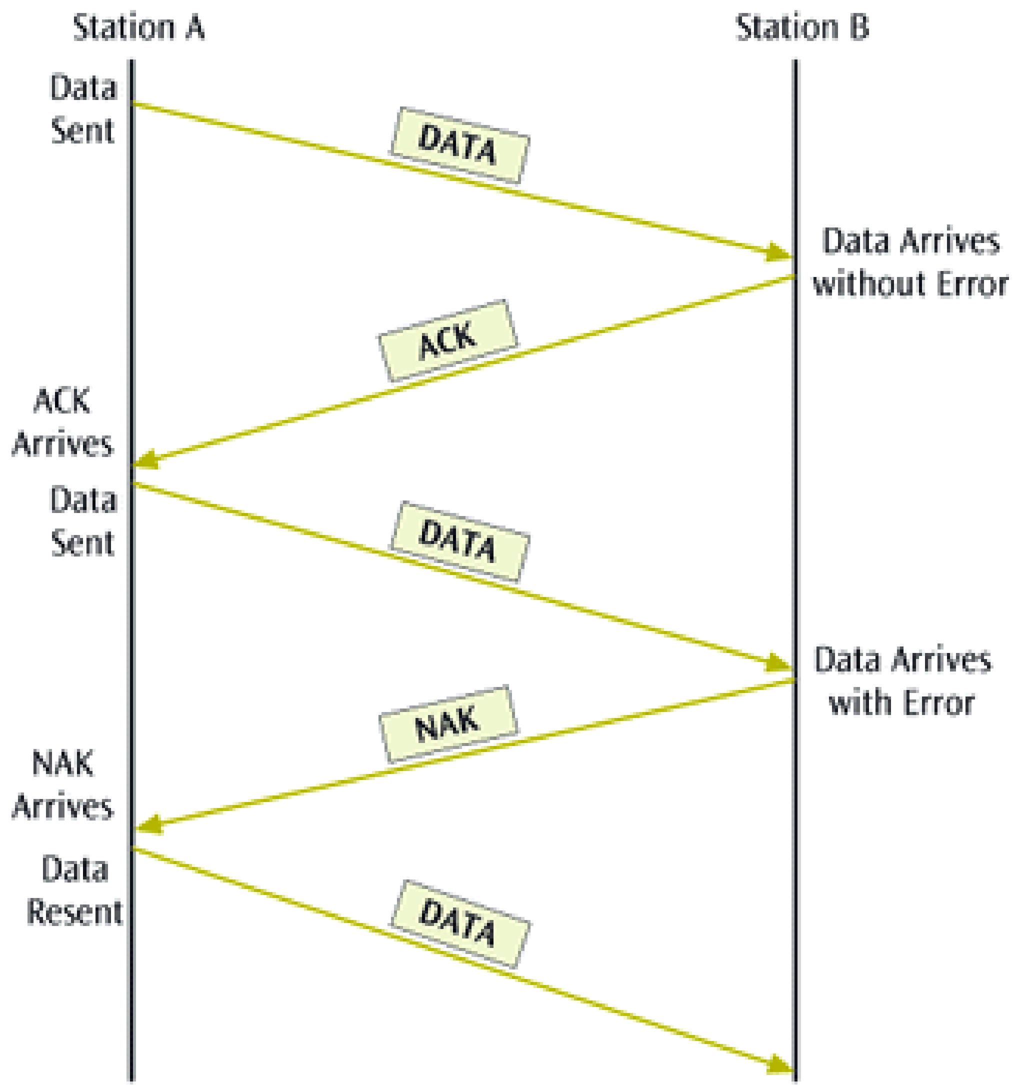
Error Control

- Once an error is detected, the receiver can:

1. Do nothing
   - Some newer systems such as frame relay perform this type of error control
2. Return an error message to the transmitter
   - Stop-and-wait ARQ (Automatic Repeat reQuest)
   - Go-back-N ARQ
   - Selective-reject ARQ
3. Fix the error with no further help from the transmitter

Stop-and-wait ARQ

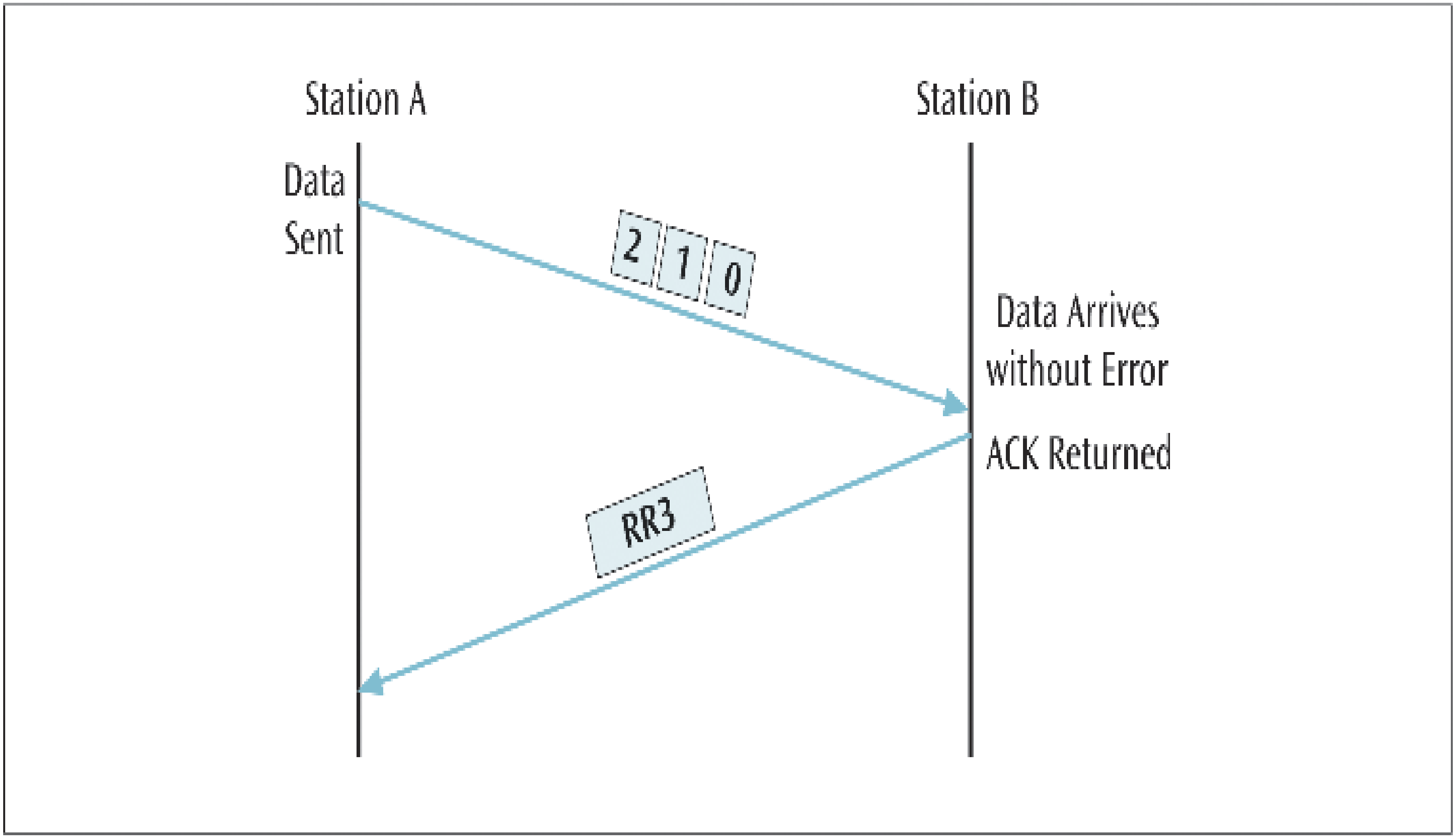
- A transmitter sends a frame then stops and waits for an acknowledgment
- If a positive acknowledgment (ACK) is received, the next frame is sent
- If a negative acknowledgment (NAK) is received, the same frame is transmitted again



Sliding Window Protocol

- **Go-back-N ARQ and Selective-reject are more efficient protocols**
  - **They assume that multiple frames are in transmission at one time (sliding window)**
- **A *sliding window protocol* allows transmitter to send up to the window size frames before receiving any acknowledgments**
- **When a receiver does acknowledge receipt, the returned pack contains the number of the frame expected *next***

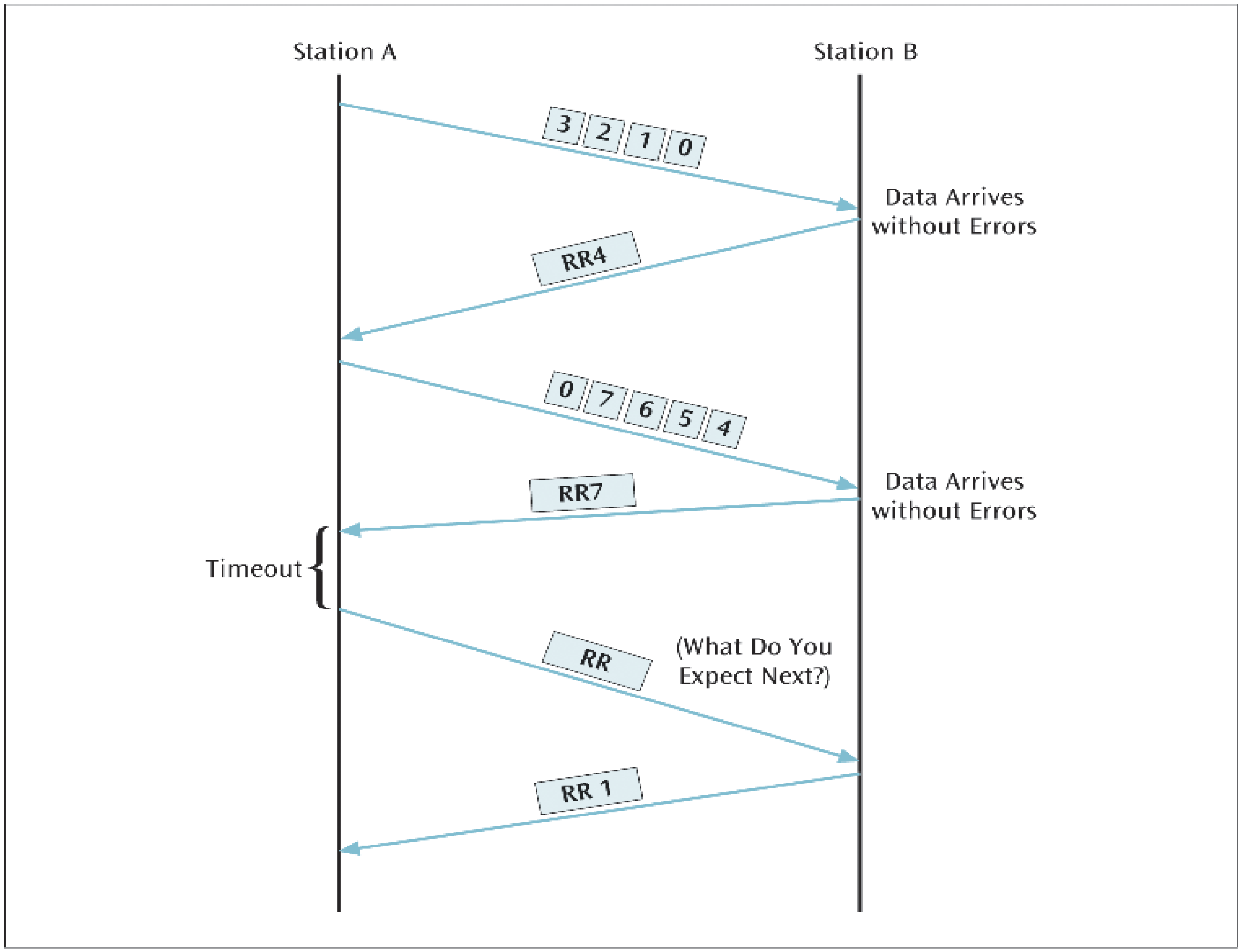
Example of Sliding Window



RR: Receive Ready

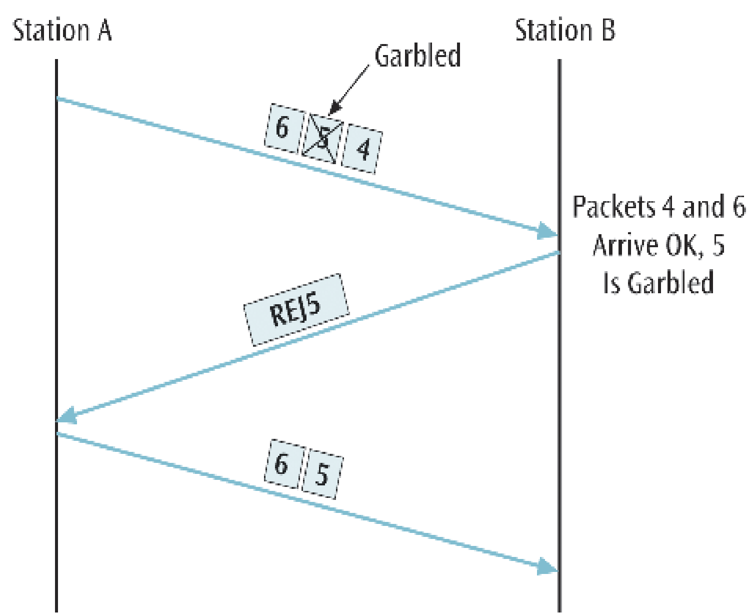
Figure 6-10

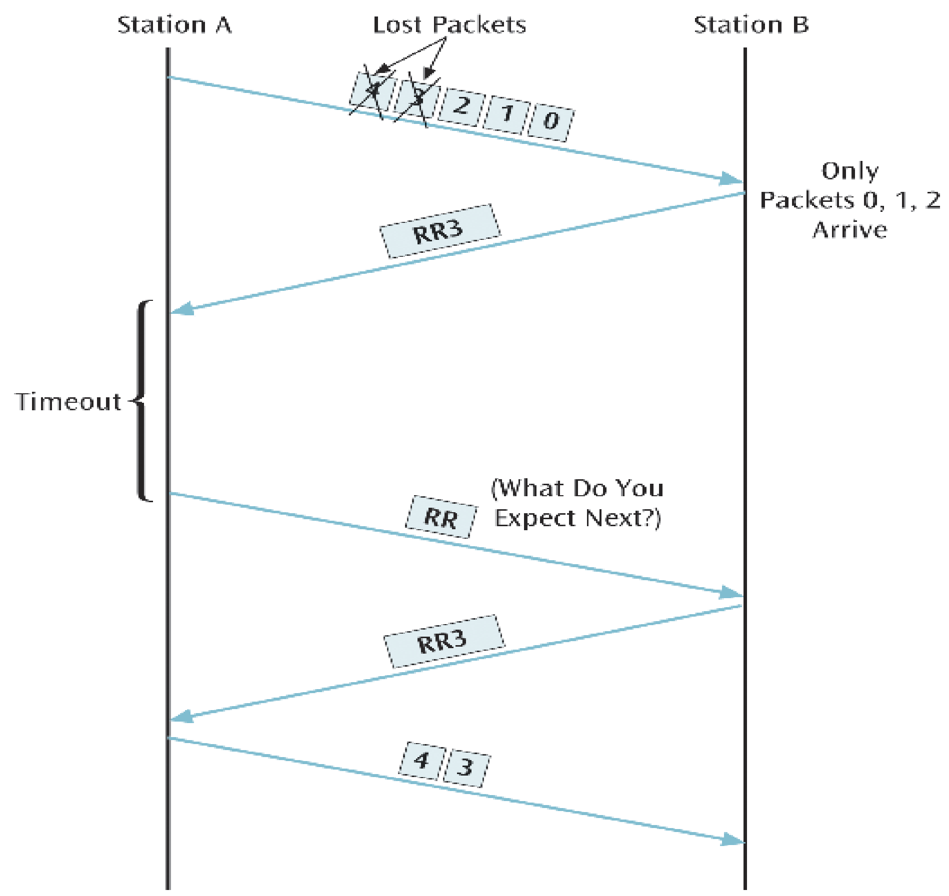
Normal transfer of packets between stations



Go back-N ARQ

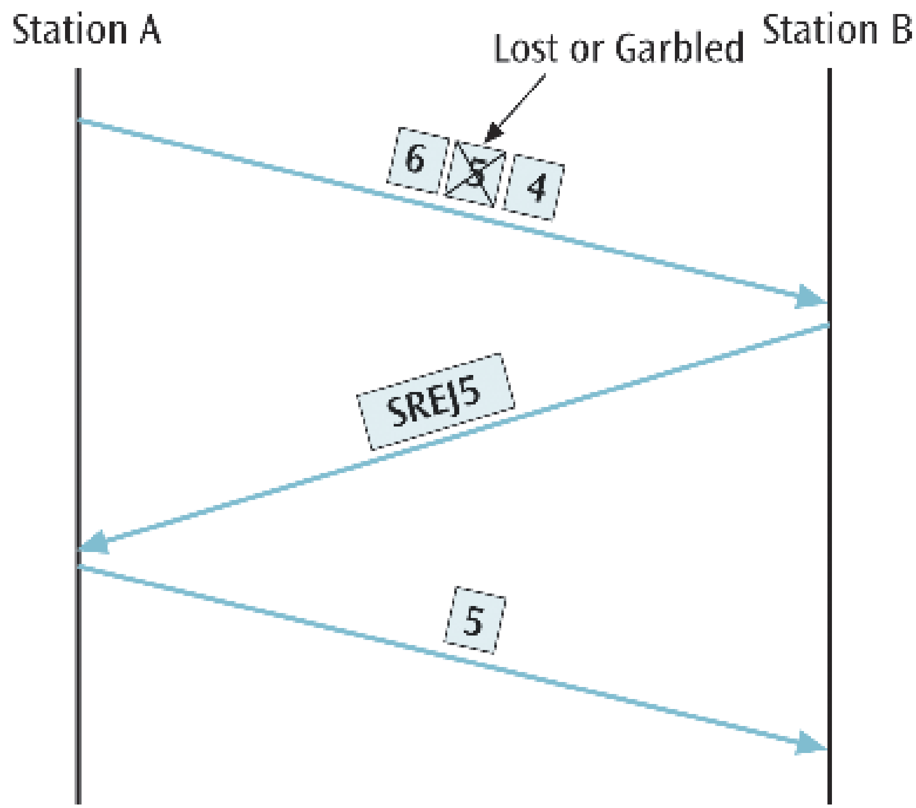
- If a frame arrives in error, the receiver can ask transmitter to go back to the Nth frame, after Nth frame is retransmitted, sender resends all subsequent frames





Selective-reject ARQ

- Most efficient error control protocol
- If a frame is received error, the receiver ask transmitter to resend ONLY the frame that was in error
- Subsequent frames following the Nth fran are not retransmitted



Correct the Error

- **For a receiver to correct the error with no further help from the transmitter requires a large amount of redundant information accompanying original data**
- **This redundant information allows the receiver to determine the error and make corrections**
- **This type of error control is often called**

<u>Forward Error Correction</u>