Flow Control Protocols - Study Notes

Overview

Flow Control Protocols are used in the **Data Link Layer** of Computer Networks. The three main protocols are:

1. **Stop & Wait**
2. **Go Back N**
3. **Selective Repeat**

---

1. Stop & Wait Protocol

Key Characteristics

- **Transmission Method**: Transmits **one frame at a time**
- **Waiting Mechanism**: Waits for acknowledgment before sending next frame
- **Frame Limit**: Only 1 frame is transmitted at a time

Window Sizes

- **Sender Window Size**: 1
- **Receiver Window Size**: 1
- **Reason**: Since only 1 packet can be sent at a time, window size > 1 makes no sense

Efficiency

Formula: `Efficiency = 1/(1+2x)`

- Where `x = Propagation delay / Transmission time`

Alternative Formula: `Efficiency = Transmission Time / (Transmission Time + Round Trip Time)`

- **Useful Time**: Transmission time for 1 frame
- **Total Time**: Transmission time + Round trip time
- **Round Trip Time**: 2 × Propagation delay

Performance Characteristics

- **Efficiency**: **Very Poor** (lowest among all three protocols)
- **Reason**: Waiting for long periods reduces bandwidth utilization

- **Key Point**: If asked "which has least efficiency?" - Answer is always **Stop & Wait**

Retransmission

- **When it occurs**:
  - Frame gets lost on the way
  - Acknowledgment gets lost
  - Packet gets corrupted
- **Number of retransmissions**: Only **1 packet** (the same packet that was lost)
- **Process**: Wait for timeout timer, then retransmit

Implementation

- **Complexity**: Very **Easy**
- **Reason**: No searching or sorting algorithms required

Sequence Numbers Required

Formula: `Sender Window Size + Receiver Window Size`

- For Stop & Wait: 1 + 1 = **2 sequence numbers**

Example Scenario

Think of PayTM transactions - until acknowledgment comes, you feel the transaction is incomplete. Similarly, sender waits for acknowledgment before proceeding.

---

2. Go Back N Protocol

Key Characteristics

- **Protocol Type**: Sliding Window Protocol
- **N**: Window size
- **Transmission Method**: Sends **entire window** (multiple packets) at once

Window Sizes

- **Sender Window Size**: `2^k - 1` (where k = number of bits to represent window size)
- **Receiver Window Size**: **1**

Understanding k (Number of Bits)

Important: If "window size is represented by 3 bits" ≠ window size is 3

Calculation:

- If k = 3 bits
- Available sequence numbers = $\boxed{2^3 = 8}$ (from 000 to 111, i.e., 0 to 7)
- **Sender window size** = $\boxed{2^3 - 1 = 7}$
- **Total sequence numbers** = 8

Efficiency

Formula: $\boxed{\text{Efficiency} = (\text{Sender Window Size}) \times 1/(1+2x)}$

- Where $\boxed{x = \text{Propagation delay / Transmission time}}$
- **Example**: If sender window size = 7, then efficiency = $\boxed{7 \times 1/(1+2x)}$

Acknowledgment System

- **Type**: **Cumulative Acknowledgment**
- **Rule**: Acknowledgment is always for **next expected frame**

Example:

- Sender sends frames: 1, 2, 3
- Receiver receives all three
- Acknowledgment sent: Frame 4 (next expected)

Advantage: Reduces network traffic (one acknowledgment for multiple packets vs. individual acknowledgments)

Out-of-Order Packets

- **Can accept out-of-order packets**: **NO**
- **Reason**: Receiver window size = 1
- **Rule**: Packets accepted only **in order**

Retransmission

- **Amount**: **Highest** among all three protocols
- **Reason**: If any frame is lost, entire window must be retransmitted

Scenario Example:

1. Sender sends: 1, 2, 3, 4

2. Frame 1 gets lost
3. Receiver wants frame 1 first
4. Frames 2, 3, 4 reach but are **rejected** (receiver only accepts in order)
5. **Entire window** (1, 2, 3, 4) must be retransmitted

Maximum retransmission: `2^k - 1` packets (entire window size)

Real-World Example

Like sending 100 packets from Chandigarh to Delhi:

- **Better**: Send all 100 at once, get 1 acknowledgment
- **Worse**: Send 1 packet → get acknowledgment → send next → repeat 100 times

---

3. Selective Repeat Protocol

Key Characteristics

- **Transmission Method**: Sends **multiple frames** at once
- **Nature**: Mixed version of Stop & Wait and Go Back N

Window Sizes

- **Sender Window Size**: `2^(k-1)`
- **Receiver Window Size**: `2^(k-1)`
- **Key Point**: Both window sizes are **equal**

Calculation Example:

- If k = 3
- Available sequence numbers = `2^3 = 8`
- Sender window size = `2^(3-1) = 4`
- Receiver window size = `2^(3-1) = 4`

Out-of-Order Packets

- **Can accept out-of-order packets**: **YES**
- **Reason**: Receiver has space for multiple packets (window size > 1)

Scenario Example:

1. Sender sends: 1, 2, 3, 4

2. Frame 1 gets lost, Frame 2 arrives first
3. Receiver accepts Frame 2 and stores it
4. When Frame 1 arrives later, it's placed in correct position
5. **No rejection** of correctly received frames

Retransmission

- **Amount**: **Very Low** (equal to Stop & Wait)
- **Reason**: Only **lost packets** are retransmitted, not entire window
- **Number**: Typically **1 packet** (only the lost one)

Efficiency

Formula: `Efficiency = (2^(k-1)) × 1/(1+2x)`

- Where `x = Propagation delay / Transmission time`
- **Performance**: Good efficiency with low retransmissions

Acknowledgment Types

Three types supported:

1. **Cumulative Acknowledgment**
    - Example: Frames 1, 2, 3, 4 received → Send ACK 5
2. **Independent Acknowledgment**
    - Example: Send separate ACK for each frame (ACK1, ACK2, ACK3)
3. **Negative Acknowledgment (NAK)**
    - **When used**: Frame received but contains errors
    - **Advantage**: Sender doesn't wait for timeout timer
    - **Time saving**: Immediate retransmission instead of waiting for timeout

NAK Example:

- Frame reaches receiver with bit errors
- Instead of waiting for timeout → Send NAK immediately
- Sender retransmits without delay

Implementation

- **Complexity**: **Most Difficult**

- **Reason**: Uses both **searching** and **sorting** algorithms
- **Requirements**:
  - Search for lost packets
  - Sort received packets in correct order

Protocol Comparison Summary

Aspect	Stop & Wait	Go Back N	Selective Repeat
Frames sent at once	1	Multiple (2^k-1)	Multiple (2^(k-1))
Sender Window	1	2^k-1	2^(k-1)
Receiver Window	1	1	2^(k-1)
Efficiency	Poorest	Better	Best
Out-of-order acceptance	N/A	No	Yes
Retransmission	Low (1 packet)	Highest (entire window)	Lowest (only lost packets)
Implementation	Easiest	Moderate	Most Complex
Acknowledgment	Simple	Cumulative	Cumulative + Independent + NAK

Important Formulas to Remember

1. **Available Sequence Numbers**: `2^k` (where k = number of bits)
2. **Stop & Wait Efficiency**: `1/(1+2x)`
3. **Go Back N Sender Window**: `2^k - 1`
4. **Selective Repeat Window**: `2^(k-1)` (both sender and receiver)
5. **Sequence Numbers Needed (Stop & Wait)**: Sender window + Receiver window
6. **General Efficiency**: `(Window Size) × 1/(1+2x)`
7. **x = Propagation delay / Transmission time**

Key Exam Points

- **Least Efficiency**: Always Stop & Wait
- **Highest Retransmission**: Go Back N
- **Out-of-order acceptance**: Only Selective Repeat
- **Window size from bits**: Always do `2^k`, not just k
- **Acknowledgment rule**: Always for next expected frame

- **Implementation complexity**: Stop & Wait < Go Back N < Selective Repeat