Flow Control and ARQ

ITS323: Introduction to Data Communications
CSS331: Fundamentals of Data Communications

Sirindhorn International Institute of Technology
Thammasat University

Prepared by Steven Gordon on 3 August 2015

ITS323Y15S1L06, Steve/Courses/2015/s1/its323/lectures/flow-control-and-arq.tex, r3920

Contents

Protocols and Protocol Analysis

Flow Control

Error Control with ARQ

Example Data Link Layer Protocols

Protocols and Protocol Analysis

A communication protocol is a set of rules for how two (or more) entities exchange data, defining:

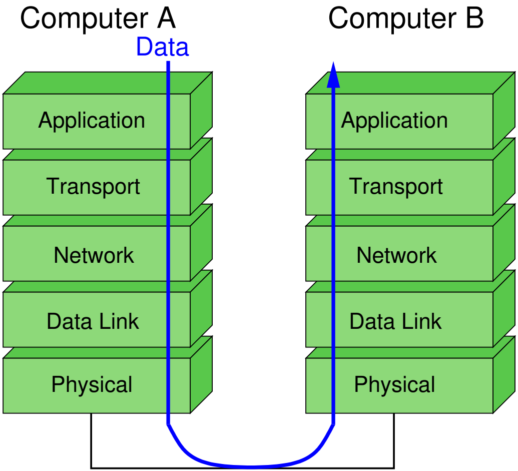
- Entities involved
- Types and formats of messages
- Assumptions about underlying communications mechanism

Designing protocols can be difficult; many techniques to support design

- Correctness: clear descriptions, formal techniques
- Performance: mathematical, simulation, experiments
- Security: informal and formal reasoning

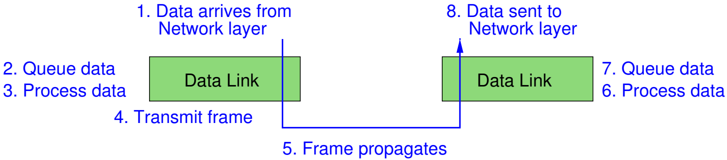
We will use simplified message sequence charts (time sequence diagrams) in this course

Analysing and Illustrating Protocols

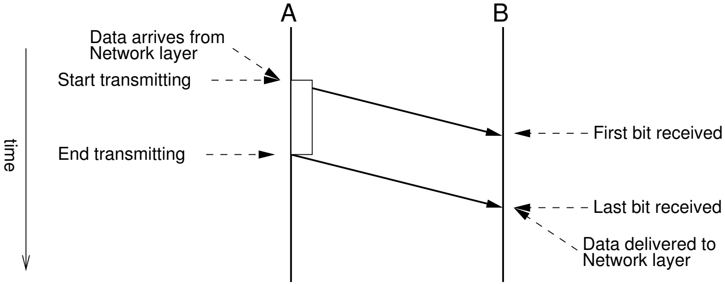


Processing and queuing at each layer
Transmission by physical layer
Propagation across link

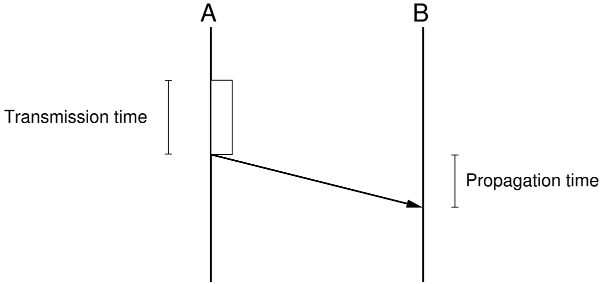
Analysing and Illustrating Protocols



Analysing and Illustrating Protocols



Analysing and Illustrating Protocols



Contents

Protocols and Protocol Analysis

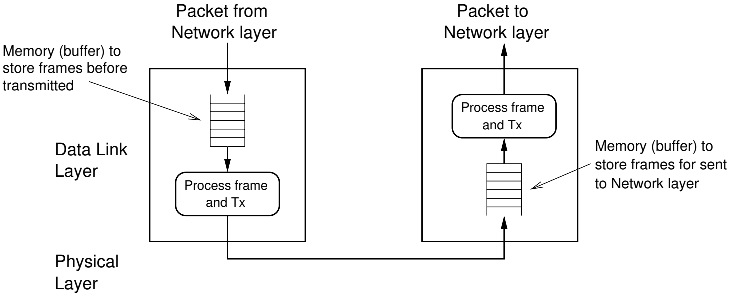
Flow Control

Error Control with ARQ

Example Data Link Layer Protocols

Flow Control

Receivers typically have a finite amount of memory (buffer space) to store received data before processing


Packet from Network layer
Packet to Network layer
Memory (buffer) to store frames before transmitted
Data Link Layer
Process frame and Tx
Process frame and Tx
Memory (buffer) to store frames for sent to Network layer
Physical Layer

Flow Control

Flow control aims to ensure sending entity does not overwhelm receiving entity

- If sender sends too fast for receiver, then buffer may overflow
- Result of buffer overflow: data is lost, possibly need to retransmit, which reduces performance
- Flow control tries to prevent buffer overflow
- Assume no errors but varying delays
- Flow control is also used in transport layer, e.g. TCP

Stop-and-Wait Flow Control

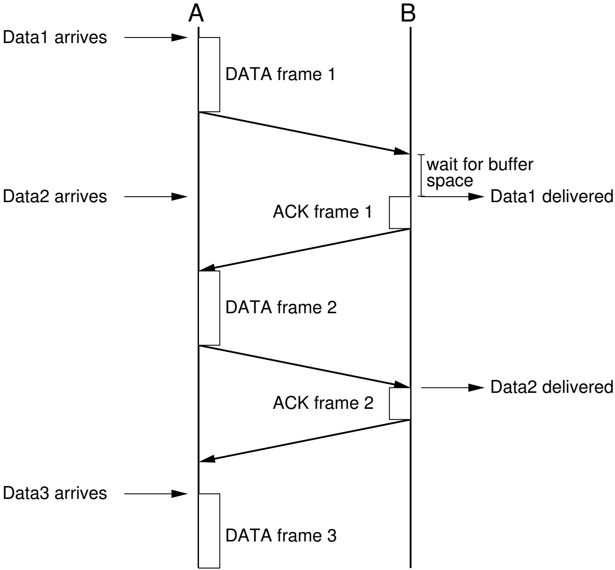
Frame Types

- DATA: contains information to be sent
- ACKnowledgement: acknowledges receipt of data

Rules

- Source transmits a DATA frame
- Source waits for ACK frame before sending next DATA frame
- Destination receives DATA frame and replies with an ACK if ready for more data
- Destination can stop flow of data by not sending ACK

Stop-and-Wait Flow Control



Example of Stop-and-Wait Performance

Source has 3 × 1000 Byte messages to be sent immediately. Destination takes 1μs to process each frame. DATA frame contains 1000 Bytes data plus 20 Byte header. ACK frame is 20 Bytes. Link is 2km, 1Mb/s and velocity of 2×10^8m/s. What is the throughput?

Efficiency of Stop-and-Wait Flow Control

- Best-case efficiency, η, for stop-and-wait flow control:

$$\eta = \frac{t_{\text{Payload}}}{t_{\text{Payload}} + t_{\text{Hdr}} + t_{\text{Ack}} + 2 \times t_{\text{Prop}}}$$

- Efficient when data transmission time is much larger than propagation time
- Inefficient for links with very high data rate (e.g. optical), large distance (e.g. satellite) or small data frames

What Size Frames To Use?

- Protocols often limit size of packets (frames), i.e. maximum number of bytes of data or payload
- Large frames minimise header overheads
- Small frames:
  - Allow more data to be sent when receive buffers are limited
  - Introduce small overhead if a retransmission is required
  - Allow fair sharing amongst multiple users
- Optimal packet size depends on overheads, and desired throughput and delay performance

Sliding-Window Flow Control

- Stop-and-wait allows only 1 frame to be in transit at a time
- Sliding-window flow control allows multiple frames to be in transit at a time

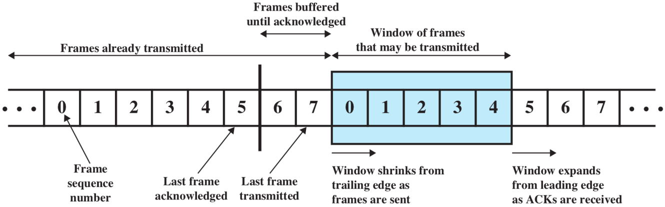
Sequence Numbers

- Each frame header contains k-bit sequence number (wraps back to 0 after $2^k - 1$)
- Keep track of frames sent and acknowledged

Sliding-Window Flow Control

The Sender

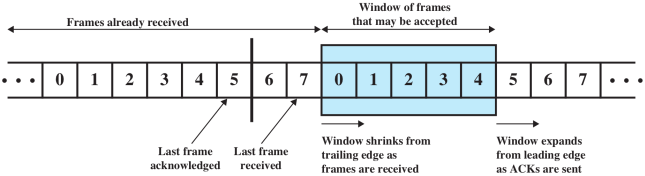
- Sender is allowed to send up to W frames without receiving ACK
- Sender records:
  - Last frame acknowledged
  - Last frame transmitted
  - Current window size


Frames buffered until acknowledged
Window of frames that may be transmitted
Frames already transmitted
• • •
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
• • •
Frame sequence number
Last frame acknowledged
Last frame transmitted
Window shrinks from trailing edge as frames are sent
Window expands from leading edge as ACKs are received

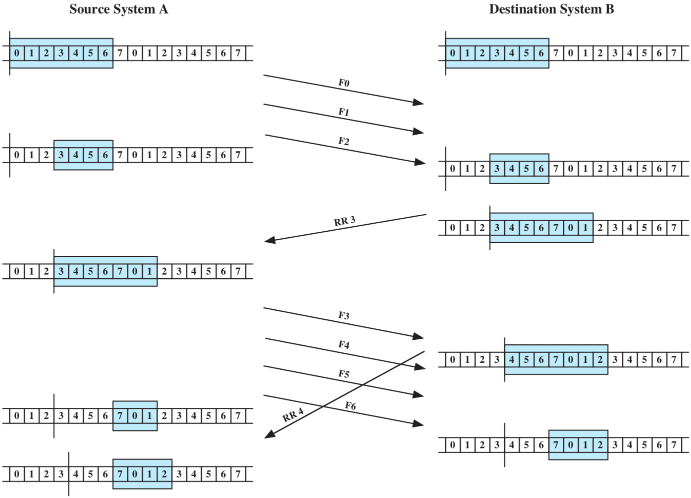
Sliding-Window Flow Control

The Receiver

- Receiver has buffer space for W frames
- Receiver records:
  - Last frame acknowledged
  - Last frame received
  - Current window size
- Receiver sends an ACK (or Receiver Ready, RR) frame
- ACK contains sequence number of next expected DATA frame


Frames already received
Window of frames that may be accepted
• • •
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
• • •
Last frame acknowledged
Last frame received
Window shrinks from trailing edge as frames are received
Window expands from leading edge as ACKs are sent

Example of Sliding Window Protocol



Example of Sliding Window Protocol

Source always has data ready to send; DATA frame contains 9000 bits payload, 1000 bits header; ACK frame is 1000 bits; data rate 100kb/s; propagation 200ms. What is efficiency if using 2-bit sequence number? 3-bits?

Additional Features of Sliding Window

- Receive Not Ready frame: acknowledges received frames but does not allow any more data
- Piggybacking: DATA frame header contains sequence number of DATA and sequence number of ACK (acknowledgement number)
- If no DATA to send, normal ACK is transmitted
- If no new ACK, previous ACK number is repeated in DATA frame

Contents

Protocols and Protocol Analysis

Flow Control

Error Control with ARQ

Example Data Link Layer Protocols

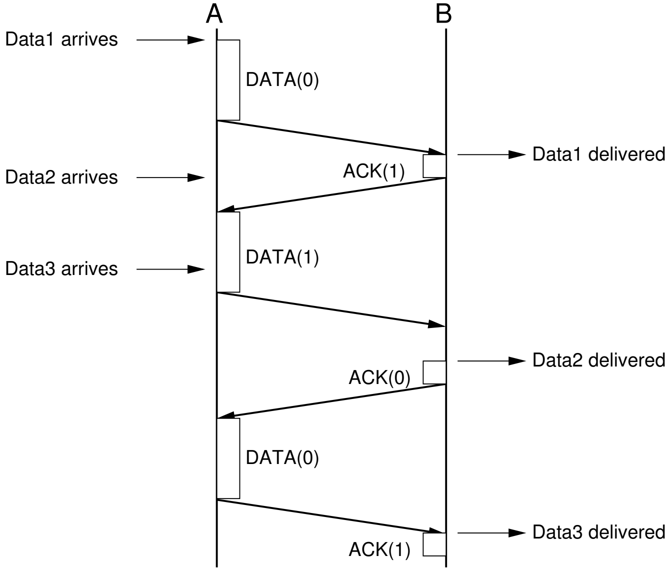
Error Control with ARQ

- Need to detect and correct of errors such as:
  - Lost frames: frame not received
  - Damaged frames: frame received with errors
- Common techniques used:
  - Error detection and FEC (discussed in previous topic)
  - Positive acknowledgment: destination returns a positive ACK after successfully receiving error-free frames
  - Retransmission after timeout: source retransmits a frame that has not been ACKed after predetermined time
  - Negative acknowledgement and retransmission: destination returns negative ACK for frames in which an error is detected
- Last 3 techniques are called automatic repeat request (ARQ). Three versions:
  - Stop-and-wait ARQ
  - Go-back-N ARQ
  - Selective-reject ARQ

Stop-and-Wait ARQ

- Based on Stop-and-Wait flow control
- Source transmits single frame, starts timer and maintains copy
  - If ACK received, stop timer and transmit next frame
  - If no ACK received before timer expires, retransmit copy of frame
- Destination sends ACK if frame received (with no errors); if damaged frame, then discard frame
- Frames have 1-bit sequence number (alternate between 0 and 1)
  - Used for destination to distinguish between duplicate DATA frames in case of damaged ACK
- Stop-and-Wait ARQ is simple, but inefficient

Stop-and-Wait ARQ Example: Normal



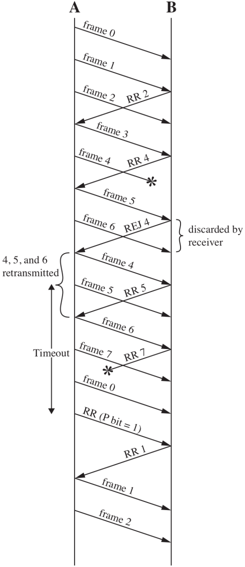
Stop-and-Wait ARQ Example: Lost DATA

Stop-and-Wait ARQ Example: Lost ACK

Go-Back-N ARQ

- Based on Sliding Window flow control
- If no error, ACK as in sliding window (contains sequence number of next expected frame)
- If error detected by Destination, reply with negative ACK (NACK or rejection, REJ)
- Destination will discard that frame and all future frames until error frame received correctly
- Transmitter must go back and retransmit that frame and all subsequent frames
- If no response from Destination after timeout, then Source may send special ACK (ACKRequest or RR(P bit = 1))
  - The ACKRequest from Source to Destination, is a request for an ACK from the Destination
  - Upon receipt of ACKRequest, the Destination sends an ACK
- Maximum window size: $2^k - 1$

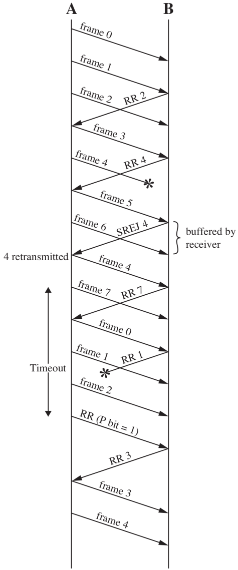
Go-Back-N ARQ Example



Selective-Reject ARQ

- Also called selective retransmission or selective repeat
- Only frames that are rejected or timeout are retransmitted
- Subsequent frames are accepted by the destination and buffered
- Maximum window size: 2^{k-1}
- Minimizes retransmission (GOOD)
- Destination must maintain large enough buffer for frames received out- of-order (BAD)
- More complex logic in transmitter (BAD)
- Not as widely used as Go-Back-N; useful for satellite links with long propagation delays

Selective Reject ARQ Example



How Long Should Timeout Interval Be?

- Long enough to allow destination to process data and return ACK
- But processing and (sometimes) propagation time are variable; difficult for source to estimate optimal timeout interval
- Too long: source waste's time waiting if DATA is lost
- Too short: source retransmits even if DATA is not lost

Contents

Protocols and Protocol Analysis

Flow Control

Error Control with ARQ

Example Data Link Layer Protocols

Example Data Link Layer Protocols

High Level Data Link Control (HDLC)

- Provides frame formats, link establishment procedures, flow and error control; Go-Back-N, Selective Reject
- Mainly used for point-to-point links

Point-to-Point Protocol (PPP)

- Commonly used by Internet Service Providers
- Uses the Link Control Protocol for link establishment, and Network Control Protocol to negotiate information for specific network layer protocols
- No flow control, and error control only via CRC

Local Area Network protocols

- Ethernet, Wireless LAN