Lecture 4: Fundamentals of Communication Protocols
09/16/2009

Lecture Outline

1. Introduction
2. The Idle RQ Protocol
3. Continuous RQ Protocol
4. Sliding Window Protocol

1. Introduction

(1) What is a communication protocol?

(2) Errors detected must be resolved.

(3) The collection of conventions for detecting, correcting, and controlling errors to ensure reliable transmission of information over unreliable communication media are called **data link control protocols**.

(4) Error control

a. Basic concept

(a) Error control and in typing: from keyboard, users use *manual error control* – use backspace key and retyping to control errors.

(b) Error control in communications are not that simple and straitforward

b. **ARQ** (**automatic repeat request**): when a transmission error occurs, the computers must be able to resolve the problem by (sender's) performing a timeout and retransmission. Acknowledgment can be sent (from the receiver to the sender) to acknowledge good frames.

2. The Idle RQ Protocol

(1) Basic Ideas (Fig.1.23, p.53):

a. Error free: Fig.1.23(a)

(a) The sender sends out one information frame, and then waits for acknowledgment message from the receiver;

(b) The receiver, upon receiving the message, sends out an acknowledgment message back to the sender;

(c) The sender sends the next information frame upon receiving the acknowledgment frame.

b. Corrupted information frame: Fig.1.23(b)

(a) When a corrupted message is received, the receiver will send back a *negative acknowledgment* (**NAK**) message, instead of the normal ack.

(b) Retranmission is needed.

c. Corrupted acknowledgment: Fig.1.23(c)

(a) When an acknowledgment is corrupted, the receiver will not be able to interpret the acknowledgment.

(b) Timeout is needed to prevent deadlocks.

d. Other cases:

(a) Loss of information frame: similar to the case of corrupted acknowledgment.

(b) Loss of acknowledgment: also similar to the case of corrupted acknowledgment.

(2) Link Utilization

a. Total time of a frame transmission: Fig.1.24, p.55

(a) Two components of propagation delays: one for info frame and another for ack;

(b) Two components of transmission delays: one for info frame and another for ack;

(c) Two components of processing time: one for info frame and another for ack;

b. Link utilization U:

$$U = \frac{T_{ix}}{T_t}$$

where T_{ix} is the time needed for a transmitter to transmit a frame, and T_t is the T_{ix} *plus* any time the transmitter spends waiting for the ack. T_t in general consists of six components:

$$T_t = T_{ix} + T_p + T_{ip} + T_{ax} + T_p + T_{ap}$$

c. Usually T_{ip} (frame processing time in the receiver) and T_{ap} (ACK processing time in the sender) are very small hence negligible (compared with T_{ix} and T_{ax}). We have

$$U = \frac{T_{ix}}{T_{ix} + 2T_p}$$

or

$$U = \frac{1}{1 + 2T_p/T_{ix}}$$

i.e.

$$U = \frac{1}{1 + 2a}$$

d. **Example** (pp.56-57, Example 1.6): Frame size: 1000 bits. Determine the link utilization for data transmission rate of (a) 1kbps and (b) 1Mbps. The velocity of propagation of the link is $2 \times 10^8 ms^{-1}$. The bit error rate is negligible.

(i) a twisted pair cable of 1km in length;

(ii) a leased line of 200 km in length;

(iii) a satellite link of 50,000 km.

Answer:
By definition, the time taken to transmit a frame is given by:

$$T_{ix} = \frac{\text{Number of bits in frame, } N}{\text{Bit rate, } R\text{, in bps}}$$

At 1 kbps:

$$T_{ix} = \frac{1000}{10^3} = 1s$$

At 1 Mbps

$$T_{ix} = \frac{1000}{10^6} = 10^{-3}s$$

$$T_p = \frac{S}{V}$$

i. $T_p = \frac{10^3}{2 \times 10^8} = 5 \times 10^{-6}$s
(i) $a = \frac{5 \times 10^{-6}}{1} = 5 \times 10^{-6}$. Hence $(1 + 2a) \approx 1$ and $U = 1$
(ii) $a = \frac{5 \times 10^{-6}}{10^{-3}} = 5 \times 10^{-3}$. Hence $(1 + 2a) \approx 1$ and $U = 1$

ii. $T_p = \frac{200 \times 10^3}{2 \times 10^8} = 1 \times 10^{-3}$s
(i) $a = \frac{1 \times 10^{-3}}{1} = 1 \times 10^{-3}$. Hence $(1 + 2a) \approx 1$ and $U = 1$
(ii) $a = \frac{1 \times 10^{-3}}{10^{-3}} = 1$. Hence $(1 + 2a) = 3$ and $U = 1/3$

iii. $T_p = \frac{50 \times 10^6}{2 \times 10^8} = 0.25$s
(i) $a = \frac{0.25}{1} = 0.25$. Hence $(1 + 2a) = 1.5$ and $U = 1/1.5 \approx 0.67$
(ii) $a = \frac{0.25}{10^{-3}} = 250$. Hence $(1 + 2a) = 501$ and $U = 1/501 \approx 0.002$

e. Effect of propagation delay and transmission rate: Fig.1.25,p.58.

(a) Three cases in Example 1.6 are illustrated.

(b) Longer distances means larger propagation delay. Higher transmission rate means each bit occupies smaller interval between the two DTEs.

f. The effect of transmission errors on link utilization: non-zero BER.

(a) Assume that to transmit a single frame successfully, an average N_r transmissions are required. Hence:

$$U = \frac{T_{ix}}{N_r T_{ix} + 2N_r T_p}$$

or

$$U = \frac{1}{N_r(1 + \frac{2T_p}{T_{ix}})}$$

(b) Assume that BER is P and a frame has N_i bits. The probability that a frame is received without a single bit error is:

$$(1 - P)^{N_i}$$

Hence the probability P_f that a frame is received with at least a single bit error is:

$$P_f = 1 - (1 - P)^{N_i} \simeq N_i P \text{ if } N_i P \ll 1$$

(c) Example (p.188). Assume that $P = 10^{-4}$, $N_i = 1000$. By the exact formula for P_f,

$$P_f = 1 - (1 - P)^{N_i} = 1 - (1 - 10^{-4})^{1000} = 0.095$$

while by the approximation formula:

$$P_f = N_i P = 1000 * 10^{-4} = 0.1$$

(d) $1 - P_f$ is the probability that a frame will be received without error. Hence, N_r, the number of transmissions that have to be made to successfully transmit a frame is given by:

$$N_r = \frac{1}{1 - P_f}$$

(e) Substitute this value for N_r to the formula for U, we have:

$$U = \frac{1}{N_r(1 + \frac{2T_p}{T_{ix}})} = \frac{1 - P_f}{1 + 2a}$$

3. Continuous RQ

(1) Motivations: improve the link utilization

a. Idle RQ is a stop-and-go protocol. Why should we stop sending the next frame?

b. The performance (link utilization) is very poor when the distance is long or the transmission rate is high.

(2) Basic ideas: Fig.1.26, p.59

a. The sender sends I-frames without waiting the ACK-frame returned.

b. The receiver returns an ACK-frame for each I-frame received.

c. Each I-frame contains a unique identifier which is returned in the corresponding ACK-frame.

Link utilization theoretically could reach 100%.

(3) Dealing with errors:

a. The sender maintains a copy of each I-frame transmitted in a *retransmission list* that operated in a FIFO queue manner. This list is used for possible retransmissions in case of errors.

b. The receiver maintains a list – the *receive list* – which contains the ids of the last n correctly received I-frames. The list is used to filter out duplicates.

c. On receipt of an ACK-frame, the corresponding I-frame is removed from the retransmission list by the sender.

(4) How to treat corrupted I-frames: **go-back-N** and **selective repeat**.

a. **Go-back-N**: when the receiver detects an corrupted I-frame, it requests the sender to retransmit all outstanding I-frames starting from the last correctly received, hence acknowledged I-frame;

b. **Selective-repeat**: when the receiver detects an corrupted I-frame, it requests the sender to retransmit only those corrupted I-frames.

(4) **Selective-Repeat** (Fig.1.27, pp.61)

a. Basic ideas: hold out of sequence frames in buffers instead of discarding them. But does not ack them.

b. Implications: acknowledge a frame with seq. number N implies all frames with seq. numbers smaller than N have been received.

c. Variations: how to handle back frames: ack; ignore; send nack.

(5) **Go-back-N** (Fig.1.28, p.63)

a. Basic ideas: discard out of sequence frames instead of holding them in buffers them.

b. Variations: how to handle back frames: ack; ignore; send nack.

4. <u>Sliding Window Protocol</u>

(1) The need of *flow control*

a. The need of storage at the receiver: each frame has to be buffered in a buffer, checked against errors.

b. The amount of buffers at each receiver is limited.

(a) With the previous continuous RQ protocols (go-back-N or selective-repeat), the fast sender may flood the slow receiver.

(b) In other words, the sender's window size and the receiver's window size are related and should be controled.

c. Flow control: control of the amount data sent to a receiver before the latter is out of usable buffers.

(2) Sender's windows and receiver's windows in continuous RQ

a. The sender has a limit on the maximum number of frames it can send before it must stop: Fig.1.29(a), p.65

(a) It maintains a list of sequences numbers with which frames can be sent.

(b) The list decreases by one in size everytime an information frame is sent.

(c) If the list becomes empty, the sender has to stop sending new information frame.

(d) Receiving ack to a previously sent information frame will create one more space in the list.

b. Similarly, the receiver has a limit on the maximum number of frames it will receive.

(a) It maintains a list of sequence numbers. It only receives info frames with sequence numbers in that list.

(b) The list will increase by one if the receiver sends an ack (for the first time) for an info frame.

(c) The receiver will stop receiving new info frame if the list size becomes zero.

(3) Window size for different protocols: Fig.1.29(b)

a. For Idle RQ, both send and receive windows have size 1.

b. For selective-repeat, both send and receive windows have size $K \geq 1$.

c. For go-back-N, the send window size is K and the receive window size is 1.

(4) Sequence numbers

a. In theory, the sequence number increases monotonically, to infinity.

b. In practice no implementation can allow infinite sequence numbers.

c. In fact, there is no need of using infinite seq numbers: Fig.1.30,p.66

  (a) For Idle RQ: 2;

  (b) For selective-repeat: 2K+1;

  (c) For go-back-n: K+1.

(5) Piggybacking

a. In practice, there are few pure senders or receivers. In many cases, a sender is also a receiver.

b. A receiver that is also sending an info frame can place ack info as part of the info frame. This is called *piggybacking*.