# Assignment

| | |
|---|---|
| **Course Code** | CSC303A |
| **Course Name** | Computer Networks |
| **Programme** | B.Tech |
| **Department** | CSE |
| **Faculty** | FET |


| | |
|---|---|
| **Name of the Student** | Satyajit Ghana |
| **Reg. No.** | 17ETCS002159 |
| **Semester/Year** | 5/2019 |
| **Course Leader(s)** | Mr Nithin Rao R |

# Declaration Sheet

| | | | |
|---|---|---|---|
| Student Name | Satyajit Ghana | | |
| Reg. No | 17ETCS002159 | | |
| Programme | B.Tech | Semester/Year | 5/2019 |
| Course Code | CSC303A | | |
| Course Title | Computer Networks | | |
| Course Date | | to | |
| Course Leader | Mr. Nithin Rao R | | |

**Declaration**

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

| | | | |
|---|---|---|---|
| Signature of the Student | | Date | |
| Submission date stamp (by Examination & Assessment Section) | | | |

| Signature of the Course Leader and date | Signature of the Reviewer and date |
|---|---|
| | |

# Contents

# 1 Question 1

Solution to Question No. 1

## 1.1 Introduction

ARQ Mechanisms detect and correct errors that occur in the transmission of frames. The aim of ARQ is to turn an unreliable conceptual point-to-point data link into a reliable one. Three standard ARQ methods are

- Stop-and-wait ARQ
- Go-back-N ARQ
- Selective repeat ARQ

When one host sends traffic to another it is desirable that the traffic should arrive in the same *sequence* as that in which it is dispatched. It is also desirable that a data link should deliver frames in the order sent. At any instant of time the sender maintains a list of consecutive sequence numbers corresponding to frames it is permitted to send. These frames are said to fall within the *sending window.* Similarly, the receiver maintains a *receiving window* corresponding to frames it is permitted to accept.

The size of the window relates to the available buffers of a receiving or sending node at which frames may be arranged into sequence.

At the receiving node, any frame falling outside the window is discarded. Frames falling within the receiving window are accepted and arranged into sequence. Once sequenced, the frames at the left of the window are delivered to the host and an acknowledgement of the delivered frames is transmitted to their sender. The window is then rotated to the position where the left edge corresponds to the next expected frame, RN.

Whenever a new frame arrives from the host, it is given the next highest sequence number, and the upper edge of the sending window is advanced by one. The sequence numbers within the sender's window represent frames sent but as yet not acknowledged. When an acknowledgement comes in, it gives the position of the receiving left window edge which indicates what frame the receiver expects to receive next. The sender then rotates its window to this position, thus making buffers available for continuous transmission.

Cyclic Sequenced numbering of the outgoing frames with cycle length $L = 2^n - 1 \geq 1$

## 1.2 Effect of channel noise on DLL sliding window protocols

### 1.2.1 Go-Back-N

Considering the Sender Window Size $w_S \geq n$ and the Receiver Window Size $w_R = 1$

Sender may send multiple frames up to $w_S$, The Receiver buffers only one received frame. It cannot receive frames *out-of-order*. Frames are passed in order to the Network Layer.

Consider two nodes A and B communicating with each other using Go-Back-N ARQ.

Damaged Frame:

- B detects error in frame(i): B send REJ(i). When A receives REJ(i) it must resend (i) and all subsequent frames.
- B does not detect frame(i) [frame is lost] and A already send frame(i+1): B receives frame(i+1) out of order and send REJ(i).
- Frame(i) is lost and A does not send anything after frame(i): A timeouts and resend frame(i).

Damaged ACK:

- ACKs are cyclic cumulative. Id an ACK(i) gets lost and a subsequent higher ACK(i+n) is received, then this ACK will account for intervening ACKs.
- If peer A timeouts without ACK: A requests retransmission of the cumulated ACK.

Performance:

- More complex than Stop-And-Wait: $w_S$ buffers/timers at the transmitter, 1 buffer/timer at receiver.
- More efficient than Stop-And-Wait: Allows for pipelining, but inefficient for noisy channels.

### 1.2.2 Selective Repeat/Reject

In this form of ARQ, $w_S = w_R$, the receiver buffers al the received frame. It can receive frames *out-of-order* and only NACKed (Negative Acknowledged) frames are retransmitted. Frames are passed in order to the Network Layer.

Performance:

- More complex than Go-Back-N: $w_S$ buffers/timers both at the transmitter and receiver.
- More efficient than Go-Back-N for noisy channels.

#### 1.2.3 Performance Comparison of the DLL Protocols

Channel Utilization as a function of the relative channel length $a$ and frame error probability $P_{fr}$

Go-Back-N:

$$U = \begin{cases} \dfrac{1-P_{fr}}{1+2aP_{fr}} & w_S \geq 2a+1 \\ \dfrac{w_S(1-P_{fr})}{(2a+1)(1-P_{fr}+w_SP_{fr})} & w_S \leq 2a+1 \end{cases}$$

Selective-Repeat

$$U = \begin{cases} 1-P_{fr} & w_S \geq 2a+1 \\ \dfrac{w_S(1-P_{fr})}{(2a+1)} & w_S \leq 2a+1 \end{cases}$$

Where,

Single Bit Error Probability (or bit error rate), $BER = \frac{n_{ERR}}{n_t}$

Frame Error Probability $P_{fr} = 1 - \ 1 - BER\ ^{S_{fr}} \approx [for\ BER^{-1} \gg S_{fr}] \approx S_{fr}BER$

Frame Size $S_{fr}$

### 1.3 Stance taken and justification

As discussed above in 1.2, in Go-Back-N algorithm if a frame is damaged, the entire window has to be resent in order to maintain the order of the frames. Taking a case of extremely noisy channel where most of the frames are damaged, here this algorithm will have to retransmit the entire window every time the frame is damaged. But in case of Selective Repeat, the order in which the frames are received does not matter, the receiver has to sort the frames once they are all received. Also looking at the formula, the channel utilization as discussed in 1.2.3 is low for Go-Back-N. Hence, I do not agree with the statement that Go-Back-N is better in a noisy channel compared to Selective-Repeat.

# 2 Question 2

Solution to Question 2

## 2.1 Introduction

The CSMA family of protocols, including ALOHA, comes in two variants: slotted and unslotted. In slotted CSMA the time is divided into time intervals, called slots, and a transmission can only start at a slot boundary. Timing information is included in the transmissions or beacons can be used to synchronize the different nodes in the network. It can be shown that the performance is better with slotted CSMA because the vulnerable period, i.e. the time interval in which two or more nodes can start to transmit simultaneously, is reduced due to the alignment of the time slots.

**ALOHA:** ALOHA is a seminal random-access protocol that became operational in 1971. In ALOHA, nodes transmit packets as soon as these are available, without sensing the wireless carrier. As a result, wireless packets may collide at a receiver if they are transmitted simultaneously. Hence, successful packet reception is acknowledged by transmitting a short acknowledgment packet. If an acknowledgment is not received timely enough, then the data packet is resent at a later instant determined, e.g., by binary exponential backoff.

**CSMA/CA:** Carrier Sense Multiple Access with Collision Avoidance is an improved random-access scheme, according to which wireless nodes first sense the wireless medium before transmitting their data packets. If the medium is sensed busy, then transmissions are deferred, e.g., according to a binary exponential backoff. Collision avoidance is enabled by:

i. waiting for an interframe spacing (IFS) duration after the channel has been sensed idle.

ii. transmitting only after a certain number of (not necessarily contiguous) sensed idle time slots, chosen randomly from the contention window (i.e., an adaptive range of possible backoff durations).

iii. exchanging Request-to-Send and Clear-to-Send frames (RTS and CTS). Out of these three methods, this example models the first two (IFS and contention window). CSMA/CA has been employed in Ethernet, IEEE® 802.11, and IEEE 802.15.4, among other standards.

## 2.2 Comparison of ALOHA and CSMA

### 2.2.1 ALOHA

ALOHA has a single parameter, the backoff time, which is a random period of time before the transmitter resends the data. In case of Slotted ALOHA time is divided into slots of equal length greater or equal to average frame duration $\tau_f$, and frame transmission can only start at beginning of a time slot.

Probability that a frame does not suffer from a collision is given by

$$P_0 = \begin{cases} e^{-2G} & \text{ALOHA} \\ e^{-G} & \text{slotted ALOHA} \end{cases}$$

The throughput/frame time is then

$$S = \begin{cases} G \cdot e^{-2G} & \text{ALOHA} \\ G \cdot e^{-G} & \text{slotted ALOHA} \end{cases}$$

Where,

Normalised channel traffic of old and new frames submitted per frame time is $G = \lambda\tau_f$

Mean Arrival Rate $\lambda$

Throughput is given by $S = G \times \text{Prob(no collision)}$

Maximum throughput of ALOHA

$$\frac{dS}{dG} = 0 \Rightarrow G_{max} = \frac{1}{2} \Rightarrow S_{max} = \frac{1}{2}e^{-1} = 0.1839$$

Maximum throughput of slotted ALOHA

$$\frac{dS}{dG} = 0 \Rightarrow G_{max} = 1 \Rightarrow S_{max} = e^{-1} = 0.3679$$

### 2.2.2 CSMA/CA

If the channel is in use, it must wait. If the medium is idle, it may transmit

- 1-persistent: a user keeps listening to see if channel is free and, as soon as the channel is idle, it transmits
- Nonpersistent: when the channel is busy, it waits for a random period of time before trying to listen again. This is less greedy
- p-persistent: for slotted systems. When the channel is free during current slot, it may transmit with probability p or may defer until next slot with probability $1 - \text{p}$

CSMA/CA uses 2 parameters, IFS and Contention Window, which are as follows

**Interframe Space**

The time interval between frames transmitted is called the Interframe Space (IFS). Two values of IFS are defined in IEEE.802.11-2016: The Short Interframe Space (SIFS) and the DCF Interframe Space (DIFS).

The SIFS interval is the duration of time allowed for a wireless interface to process the received RF signal and its associated frame, and to generate a response frame. The SIFS for IEEE 802.11a, 802.11n and 802.11ac (at 5 GHz) is 16.

The DIFS interval is calculated as: DIFS = SIFS + (2 * Slot time)

A node is required to sense the activity of the wireless medium before transmitting (listen before talk). If it finds that the medium is continuously idle for the duration of a DIFS period, the node is then permitted to start transmission of a frame (after also waiting for an additional random backoff interval). The random backoff waiting time is introduced to avoid synchronisation in this decentralised system. The range of the generated random backoff timer is bounded by the Contention Window. If the channel becomes busy during the DIFS interval, the node is required to defer its transmission until the medium is again found idle for the duration of a DIFS interval.

**Contention Window**

The Contention Window bounds the range of the generated random backoff timer. The initial range is set between 0 and the Contention Window minimum value (CWmin). The CWmin for DCF (in 5 GHz) is specified as 15 slot times (IEEE.802.11-2016).

It is possible that two (or more) nodes happen to (randomly) choose the same value. If this happens, a collision may occur. At this point, the node effectively restarts the algorithm, waiting for the DIFS interval and then selecting a new random backoff value. However, a key difference is that for this subsequent attempt, the Contention Window approximatively doubles in size (exponentially increasing the range of the random value, as in CSMA/CD). Further collisions cause the algorithm to continue to expand the backoff period, until the node reaches the maximum Contention Window size (CWmax). The CWmax for DCF is specified as 1023 slot times, after which the random backoff is not further expanded, but the algorithm may continue to retransmit.

A side effect is that a loaded WiFi access point (with many nodes trying to send at the same time), can result in significant numbers of retransmissions - each failed retransmission adds further delay resulting in jitter for the transmission (and some waste in network capacity due to collisions).

Throughput in case of CSMA

- For CSMA with small p, the method performs very well in terms of throughput at high load (almost 100%). However, for smaller p, users must wait longer (larger delay) to attempt transmission
- In the extreme case: only single user wishes to transmit, expected number of deferring is 1/p. If p = 0.01, at low load, a user will wait an average of 99 time slots before transmitting on an idle line
- For low load, slotted ALOHA is preferred due to its low delay

## 2.3 Conclusion

The performance parameter which is throughput in our case for the two algorithms was calculated, we can clearly see that ALOHA has very low throughput in case of high-load, when compared to CSMA/CA. This should be considering the worst-case scenario of network load. Although in case low load, low delay is preferred, in that case ALOHA is more useful.

The delay caused in CSMA is due to the IFS and the Contention Window overhead, this becomes negligible when the network has high traffic. The IFS reduce the traffic caused in the network just by sensing the carrier continuously.

Thus, we can conclude that CSMA is a better option in general, since it provides less collisions, by using the estimated frame space and the contention window, when compared to ALOHA which only uses a random backoff time to reduce the collisions.

## Bibliography


1. https://www.dsi.unive.it/~franz/reti/dll/Protocolli.html
2. https://erg.abdn.ac.uk/users/gorry/course/lan-pages/csma-ca.html
3. https://in.mathworks.com/help/comm/examples/aloha-and-csma-ca-packetized-wireless-networks.html