RAMAIAH UNIVERSITY OF APPLIED SCIENCES

ASSIGNMENT

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

Name of the Student	SHIKHAR SINGH
Reg. No	17ETCS002168
Semester/Year	05/2017
Course Leader/s	Mr Nithin Rao R

<table>
<tr><th colspan="6">Declaration Sheet</th></tr>
<tr><td>Student Name</td><td colspan="5">Shikhar singh</td></tr>
<tr><td>Reg. No</td><td colspan="5">17ETCS002168</td></tr>
<tr><td>Programme</td><td colspan="2">B.Tech</td><td>Semester/Year</td><td colspan="2">05/2017</td></tr>
<tr><td>Course Code</td><td colspan="5">CSC303A</td></tr>
<tr><td>Course Title</td><td colspan="5">Computer Network</td></tr>
<tr><td>Course Date</td><td></td><td>to</td><td colspan="3"></td></tr>
<tr><td>Course Leader</td><td colspan="5">Mr Nithin Rao R</td></tr>
<tr><td colspan="6">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.</td></tr>
<tr><td>Signature of the Student</td><td colspan="3"></td><td>Date</td><td></td></tr>
<tr><td>Submission date stamp
(by Examination & Assessment Section)</td><td colspan="5"></td></tr>
<tr><td colspan="2">Signature of the Course Leader and date</td><td colspan="4">Signature of the Reviewer and date</td></tr>
<tr><td colspan="2"></td><td colspan="4"></td></tr>
</table>

Faculty of Engineering & Technology			
Ramaiah University of Applied Sciences			
Department	**Computer Science and Engineering**	Programme	**B. Tech.**
Semester	**5th**		
Course Code	**CSC303A**	Course Title	**Computer Networks**
Course Leader	**Dr Rinki Sharma, Mr Nithin Rao R, Gp Capt N Rath VSM**		

Assignment			
Register No.	**17ETCS002168**	Name of Student	**SHIKHAR SINGH**

Sections		**Marking Scheme**	**Max Marks**	**First Examiner Marks**	**Second Examiner Marks**
Question-1	1.1	Introduction	01		
	1.2	Effect of channel noise on DLL sliding window protocols	03		
	1.3	Stance taken and Justification	01		
		Question 1 Max Marks	**05**		
Question-2	3.1	Introduction	01		
	3.2	Comparison of ALOHA and CSMA	03		
	3.3	Conclusion	01		
		Question 2 Max Marks	**05**		
		Total Assignment Marks	**10**		

Course Marks Tabulation				
Component- 1(B) Assignment	**First Examiner**	**Remarks**	**Second Examiner**	**Remarks**
1				
2				
Marks (Max 10)				
Signature of First Examiner				Signature of Second Examiner

Question No. 1

Solution to Question No. 1:

1.1 Introduction:

The physical layer receives the data in form of streams of bits without any regard to meaning or structure, it is up to the data link layer to create and recognize frame and their properties. One common way of doing this is attaching special bit patterns to the beginning and end of the frame (frame delimiters). Also, if these patterns occur in data, special care must be taken to make sure these patterns are not incorrectly interpreted as frame delimiters.

An important aspect of the functionality and the topic of the debate is flow control. In a situation where the sender transmits the frames faster than the receiver can actually accept. If the sender keeps pumping out the frames at higher rate, at one point the receiver will get swamped by the tremendous amount of incoming data and start losing frames. To prevent this, we introduce flow control mechanism to inform the sender when it should transmit the next frame.

A common protocol which ensures flow control is **sliding window protocol**.

- Sliding window protocol is a flow control protocol.
- It allows the sender to send multiple frames before needing the acknowledgements.
- Sender slides its window on receiving the acknowledgements for the sent frames.
- This allows the sender to send more frames.
- It is called so because it involves sliding of sender's window.

Also, an important property is to be noted for this: -

Maximum number of frames that sender can send without acknowledgement

= Sender window size

A significant problem with this protocol is that if a frame is lost or an erroneous frame is received, the receiver may simply discard all subsequent frames and it will not send any acknowledgement for the discarded frames. This will result in the sender waiting for acknowledgement and re-transmitting the entire set of frames. Which will be a wastage of resources. Hence, Go-Back-N and Selective Repeat protocols are ARQ techniques are implemented. These are further advanced protocols developed by modifying sliding window protocol.

The GO-Back-N and selective repeat are ARQ protocols that are two-way transmission protocols used to send frames in DLL. ARQ (automatic repeat query) methods are the ones where the sender waits for a specific time for the acknowledgement of the frames sent by the receiver. If that is not received, the sender will re-transmit the frames.

For a noisy channel i.e. higher probability of frame damage or frame loss, will lead to more re-transmissions, hence there's a need to determine which protocol holds and advantage over the other.

1.2 Effect of channel noise on DLL sliding window protocols:

Considering **Go-Back-N protocol,** we know that the window size of the receiver is fixed to 1, and the senders window size is kept variable. Now, if the frame is lost or damaged, the receiver will send either no acknowledgement or a negative acknowledgement respectively for the issues mentioned. To resolve this issue, the sender window will go back to the first frame and send all the frames from the first frame to the last frame sent. (the lost frame.)

While implementing this protocol in a noisy channel, we may have to re-transmit all the frames from first to last lost frame every time a damaged packet arrives or that packet is lost. This increases the total time required and exhausts the entire bandwidth.

For a **selective repeat protocol**, the window sized is kept equal for both the sender and the receiver. If the frame to be sent is lost or damaged, then that particular frame itself is sent after receiving a negative or no acknowledgement. This causes the frames being sent to be unordered. Hence, all the frames are stored in a buffer by the receiver, which, when receives all the frames, sorts all the frames it receives in the buffer. Therefore, due to buffer storage and sorting algorithm, it is considered as a complex protocol to be implemented.

If this protocol is implemented in noisy channel, number of frames being re-transmitted shall be comparatively less than Go-Back-N but if the sorting is to be done every time the frame is corrupted, then it becomes even more complex and thus it is costly to implement. Number of re-transmissions per packet drop is just 1.

In terms of bandwidth and time efficiency, selective repeat is better than Go-back-N and in terms of implementation complexity and overall cost analysis, Go-back-N is better than selective repeat.

1.3 Stance taken and justification:

My stance will be in favor of the argument “Go-Back-N protocol is a better choice compared to Selective Repeat for a noisy channel” because in a noisy channel, the rate of frame drop and corruption will be high and thus if the disadvantages and advantages of both the protocols are taken into account, then Go-back-N will outperform selective repeat in terms of implementation complexity and cost effectiveness as it doesn’t need a buffer nor a sorting algorithm at both the ends in order to make two nodes communicate in a network. Hence, making it easier to implement.

In terms of time complexity in a noisy channel, the Go-back-N will be slower than selective repeat and also it wastes the bandwidth to a higher degree because it resends the entire set of frames every time the frames are lost or damaged.

But in real-life conditions, the cost efficiency and implementation complexity outweigh the time complexity and bandwidth wastage, hence easily making Go-Back-N a better choice than selective repeat.

Solution to Question No. 2:

2.1 Introduction:

The CSMA protocol methods (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 (also called 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, random back-off time.
CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance is an improved 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 random back-off time 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 back-off durations), (iii) exchanging Request-to-Send and Clear-to-Send frames (RTS and CTS).

2.2 comparison of ALOHA and CSMA:

ALOHA:

In the case of Aloha, be it pure Aloha or slotted Aloha, we use the concept of vulnerability time, it is the amount of time, the data is prone to collision. Since the protocol has no sensing mechanism, the possibility of collision is very high. The collision occurs if two nodes transmit data at the same time. Even if a single bit is corrupted, then it is collision. For pure aloha, the vulnerability time is '2t' and for slotted aloha it is 't', where 't' stands for the time taken to send one frame. So, the parameter used to measure the performance is time.

Its 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 τ_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 the following relation:

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

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 is λ

Throughput $S = G \times \text{Probability of no collision}$

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

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

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 − p

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.

Here, the node will sense the channel before it sends any data on to the channel. The sender sends RTS (request to send) frame, and the receiver responds with a CTS (clear to send frame). A concept of IFS (inter frame space, the relaxation time for channel and sender while the frame is being processed at the receiver end) is used in this. In CSMA/CA the DFS is used. DIFS stands for DCF (Distributed Co-ordinated Function) IFS. The node will sense the channel, if it is free, it will not send data immediately, it will wait for DIFS, sense channel again and then sends the data. So, the parameter “space” comes into existence because of this. So, the collision depends on both time and space here. Though CSMA/CA should be avoiding collision but having a completely collision free network/channel is not possible, it’s just theoretical.

Hence, the main difference between both the protocols (ALOHA and CSMA) is that ALOHA protocol does not try to detect whether the channel is free before transmitting but the CSMA protocol verifies that the channel is free before data transmission. Thus, avoiding the possibility of future collisions. Due to this preventive measure, CSMA is more suitable for networks such as Ethernet where multiple sources and destinations use the same channel.

With ALOHA one can imagine that they would need to know how many stations there were around him. Because he would have to change that probability that he would transmit in each time slot depending upon how many stations there were. But if he’s not doing any sensing, there's no way he can tell how many stations there are that could be interfering with it. Therefore, he would need some central mechanism, some sensual coordinator to be able to tell that. In order for him to keep the optimal probability of transmitting each time. But with CSMA, we can run all the back off, and keep tallies on what the current back off counter is. just in our own device. And we don't need to really know anything about how the other devices are operating other than the fact that we have to sense them.

2.3 Conclusion

The possibility of distinction mostly deals with the applicative principle that governs the working of these protocols i.e. carrier sensing or no carrier sensing.

In ALOHA, time is the only major factor which indicates anything about the possibility of collision. Whereas, in CSMA, both time and space play a major role in collision detection and avoidance.

Hence, the distinction is based on the performance measuring parameters of these protocols.

1. https://www.cse.iitk.ac.in/users/dheeraj/cs425/lec09.html
2. https://www.gatevidyalay.com/tag/difference-between-go-back-n-and-selective-repeat-protocol/
3. “Performance analysis for a stabilized multi-channel slotted ALOHA algorithm” - 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003.
4. “CSMA with Enhanced Collision Avoidance: A Performance Assessment” - VTC Spring 2009 - IEEE 69th Vehicular Technology Conference