Experiment No. -09

Experiment Title: Study of ALOHA Protocol in Data Communication

Objective: The main objective of this experiment is to study the Pure ALOHA protocol and analyze its performance characteristics in the context of data communication. Pure ALOHA is a simple, contention-based protocol used for media access control in shared communication channels. Through this experiment, we aim to understand the efficiency, collision characteristics, and throughput of the Pure ALOHA protocol under different network conditions.

Equipment and Materials:

1. Computers or devices with network interfaces
2. Network simulation software (e.g., Cisco Packet Tracer, GNS3)
3. Ethernet cables
4. Switches and routers (if not using simulation software)
5. Data communication devices (such as PCs or laptops)
6. Stopwatch or timer
7. Protocol analyzer tool (e.g., Wireshark)

Theory:

The ALOHA protocol was first developed at the University of Hawaii in the early 1970s for packet radio networks. However, it can be used in any situation where multiple devices share a common communication channel. This protocol allows devices to transmit data at any time, without a set schedule. This is known as a random access technique, and it is asynchronous because there is no coordination between devices. When multiple devices attempt to transmit data at the same time, it can result in a collision, where the data becomes garbled. In this case, each device will simply wait a random amount of time before attempting to transmit again. The basic concept of the ALOHA protocol can be applied to any system where uncoordinated users are competing for the use of a shared channel.

What is Pure ALOHA:

- Pure ALOHA refers to the original ALOHA protocol. The idea is that each station sends a frame whenever one is available. Because there is only one channel to share, there is a chance that frames from different stations will collide.
- The pure ALOHA protocol utilizes acknowledgments from the receiver to ensure successful transmission. When a user sends a frame, it expects confirmation from the receiver. If no acknowledgment is received within a designated time period, the sender assumes that the frame was not received and retransmits the frame.
- When two frames attempt to occupy the channel simultaneously, a collision occurs and both frames become garbled. If the first bit of a new frame overlaps with the last bit of a frame that is almost finished, both frames will be completely destroyed and will need to be retransmitted. If all users retransmit their frames at the same time after a time-out, the frames will collide again.
- To prevent this, the pure ALOHA protocol dictates that each user waits a random amount of time, known as the back-off time, before retransmitting the frame. This randomness helps to avoid further collisions.

- The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations (2 x Tp).

- Let all the packets have the same length. And each requires a one-time unit for transmission (tp). Consider any user to send packet A at a time. If any other user B has generated a packet between time (to), and (to + tp), the end of packet B will collide with the beginning of packet A. Since in a pure ALOHA packet, a station does not listen to the channel before transmitting, it has no way of knowing that the above frame was already underway.

- Similarly, if another user wants to transmit between (to, +tp) and (to +2tp) i.e. packet C, the beginning of packet C will collide with the end of packet A. Thus if two packets overlap by even the smallest amount in the vulnerable period both packets will be corrupted and need to be retransmitted.

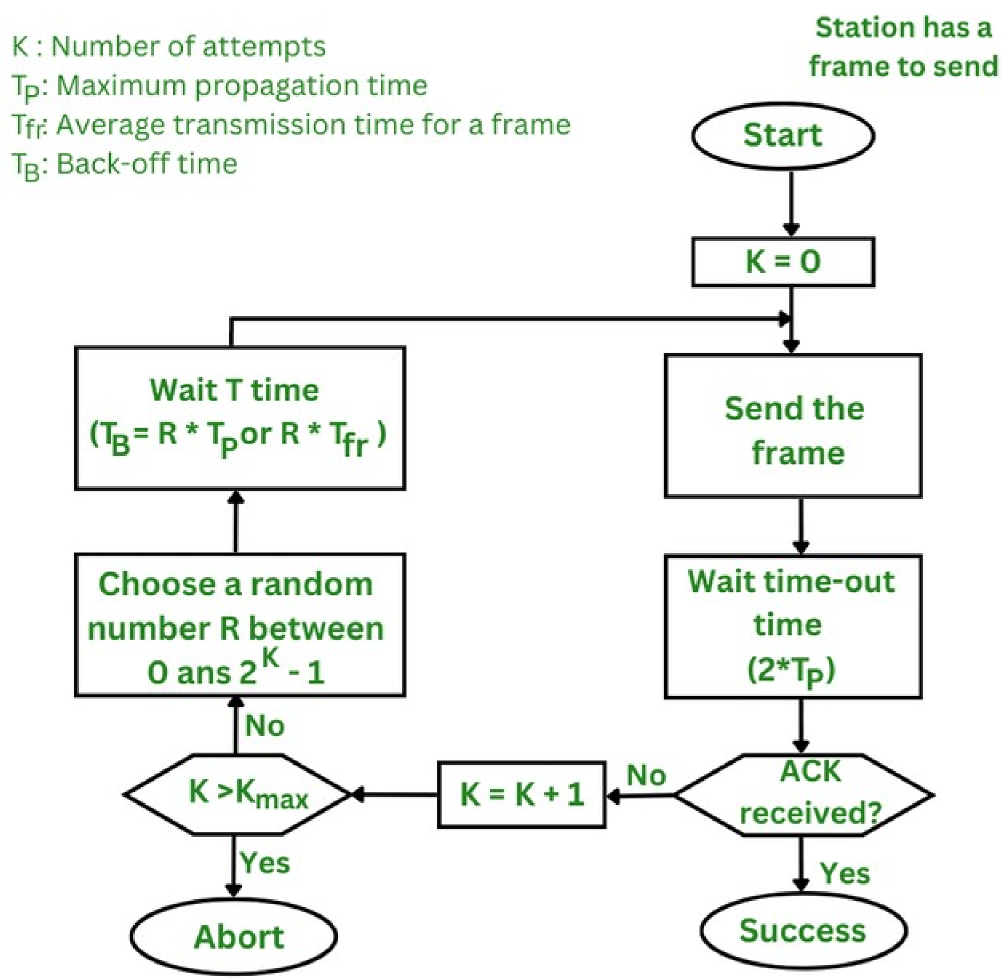

Figure: *Pure ALOHA procedure*

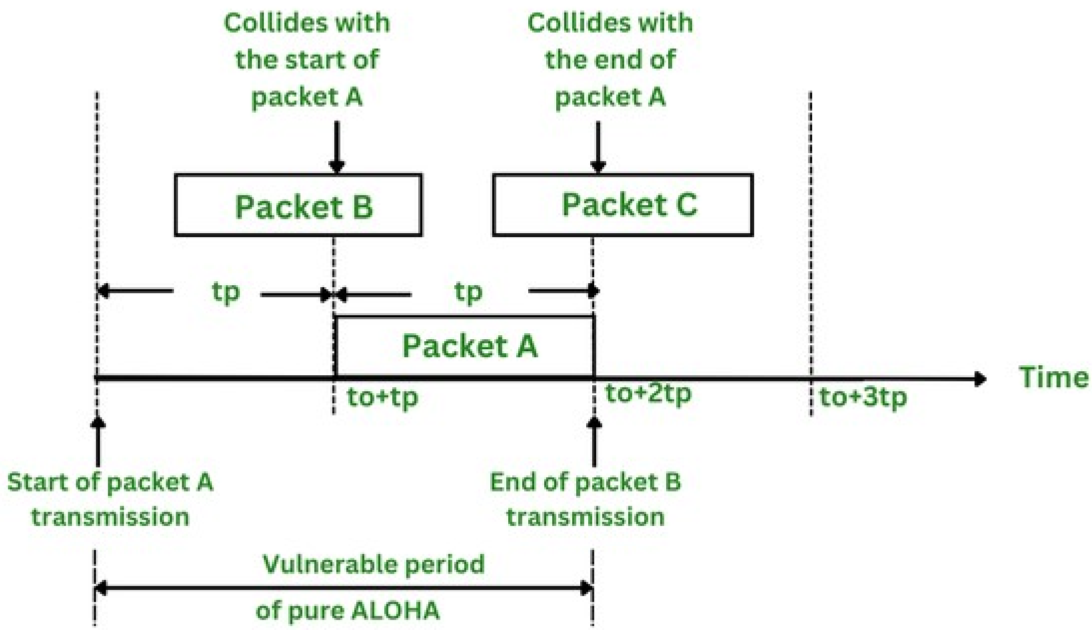

Figure: Vulnerable period for packet A

Key Features of Pure ALOHA:

1. Random Access: Devices can send data whenever they have something to transmit, without needing to wait for a predetermined time slot.

2. Uncoordinated Transmission: Devices do not coordinate with each other before transmitting. They simply attempt to send data whenever they have data to send.

3. Simple Implementation: Pure ALOHA is straightforward to implement, making it suitable for early network experiments and scenarios with low traffic.

4. Persistent Approach: Devices continue to attempt transmission even after a collision, using a form of exponential backoff. This means they introduce random delays before retrying, which helps reduce the chances of repeated collisions.

5. Contention-Based: Since devices transmit without coordination, collisions may occur if two or more devices transmit simultaneously. Collisions are detected through feedback from the receiver or by the transmitting device itself.

What is Slotted ALOHA:

Slotted ALOHA is an improved version of the pure ALOHA protocol that aims to make communication networks more efficient. In this version, the channel is divided into small, fixed-length time slots and users are only allowed to transmit data at the beginning of each time slot. This synchronization of transmissions reduces the chances of collisions between devices, increasing the overall efficiency of the network.

The channel time is separated into time slots in slotted ALOHA, and stations are only authorized to transmit at particular times. These time slots correspond to the packet transmission time exactly. All users are then synchronized to these time slots so that whenever a user sends a packet, it must precisely match the next available channel slot. As a result, wasted time due to collisions can be reduced to one packet time or the susceptible period can be half.

When a user wants to transmit a frame, it waits until the next time slot and then sends the frame. If the frame is received successfully, the receiver sends an acknowledgment. If the acknowledgment is not received within a time-out period, the sender assumes that the frame was not received and retransmits the frame in the next time slot.

Slotted ALOHA increases channel utilization by reducing the number of collisions. However, it also increases the delay for users, as they have to wait for the next time slot to transmit their frames. It's also worth noting that there is a variant of slotted ALOHA called "non-persistent slotted ALOHA" which is a variation of slotted ALOHA, in this variant the station that wants to send data, first listens to the channel before sending the data. If the channel is busy it waits for a certain time before trying again.

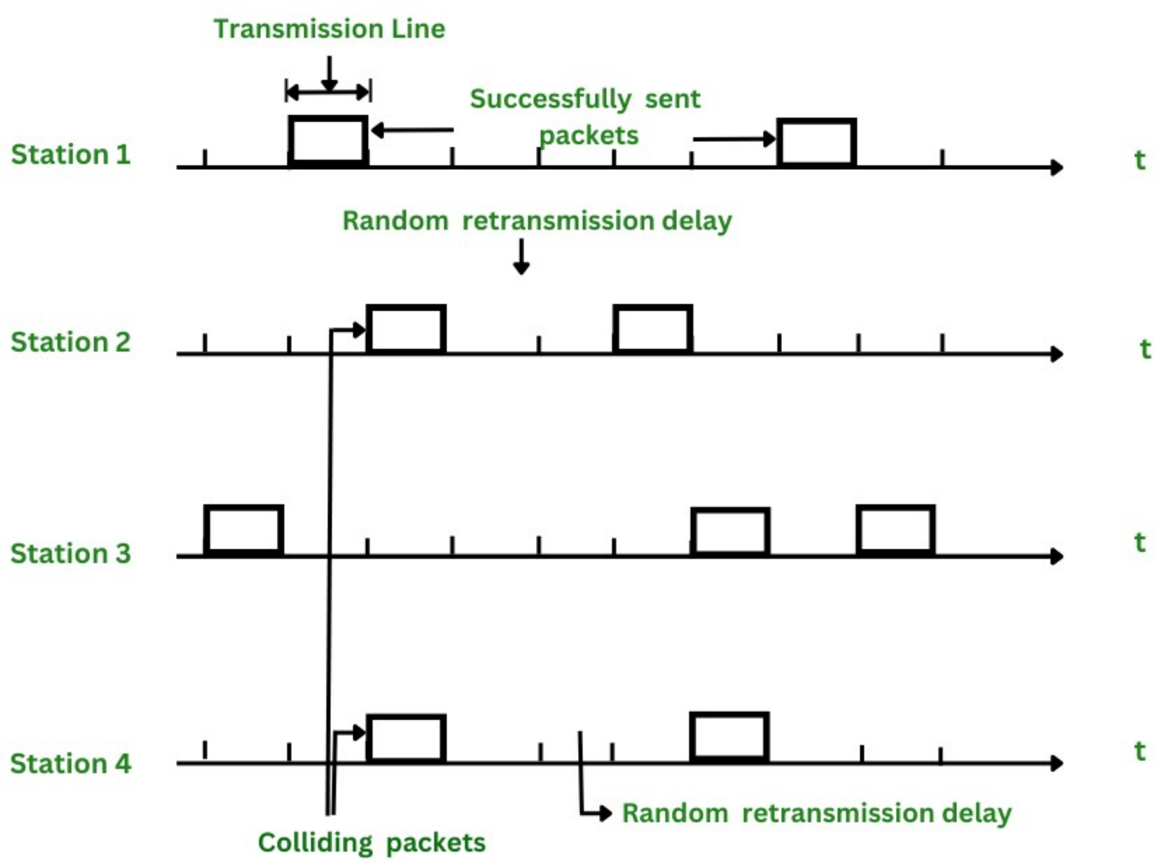

Figure: *Transmission attempts and random retransmission delay for colliding packets in slotted ALOHA*

Throughput of Slotted ALOHA:
The quantity of successful transmissions at each time slot determines the throughput of the Slotted Aloha protocol. The Slotted Aloha protocol has a maximum throughput of about 18.4%. This is because there is a significant risk of collisions when numerous nodes try to transmit at the same time, which causes missed packets and a decreased overall throughput. When less than or equal to 37% of the network's total nodes are actively transmitting data, the maximum throughput is reached.
Due to the high frequency of collisions, the throughput of Slotted Aloha is typically substantially lower than 18.4% in practice It is not a widely used protocol in today's contemporary networks because of this.

The maximum throughput of a slotted ALOHA channel is given by the formula:

Throughput (S) = G x exp(-G)
The maximum Throughput occurs at G = 1,
i.e. S = 1/e = 0.368

Where: G = the offered load (or the number of packets being transmitted per time slot). The offered load is a measure of the number of nodes attempting to transmit in a given time slot.

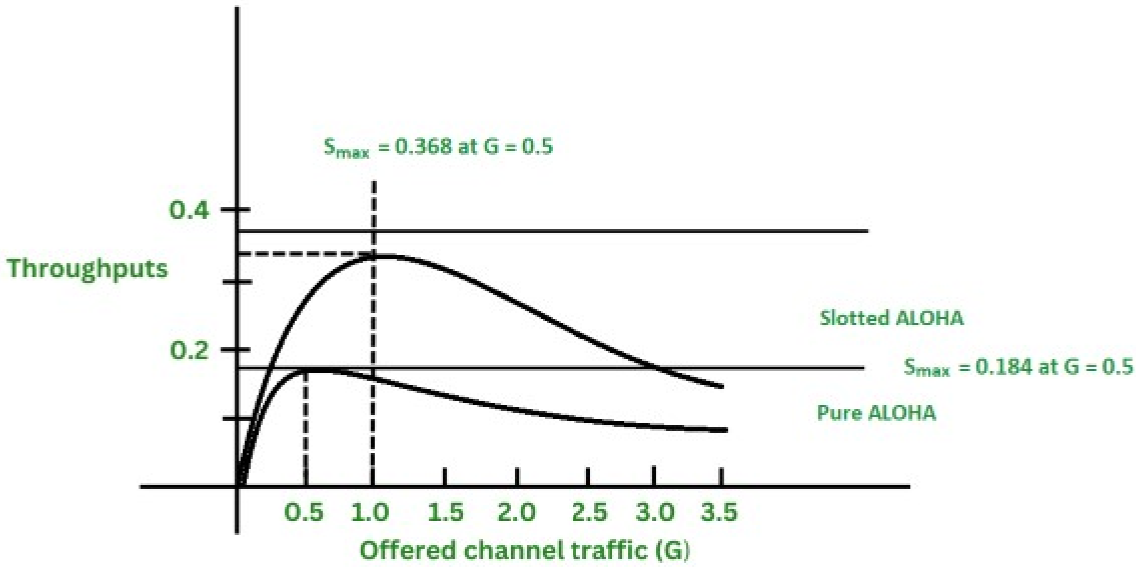

Figure: Comparison of the throughput as a function of offered load for Pure and Slotted ALOHA

The throughput is a function of the offered load and it ranges from 0 to 1. As the offered load increases, the throughput decreases as more collisions occur, resulting in less successful transmissions. The maximum throughput is achieved when the offered load is equal to 0.37 and it is approximately 0.184. It is important to note that the above equation assumes that all the packets are of the same length and that the channel is error-free. In practice, the throughput is usually much lower than this due to a number of factors such as packet errors, channel noise, and the overhead of retransmissions.

Assumption of Slotted ALOHA:

- All frames are of the same size.
- Time is divided into equal-sized slots, a slot equals the time to transmit one frame
- Nodes start to transmit frames only at beginning of slots.
- Nodes are synchronized.
- If two or more nodes transmit in a slot, all nodes detect collision before the slot ends.

Advantages of Slotted ALOHA:

- **Simplicity:** The Slotted Aloha protocol is relatively simple to implement and understand, making it an easy option for low-complexity networks.
- **Flexibility:** Slotted Aloha can be used in a wide range of network environments, including those with varying numbers of nodes and varying traffic loads
- **Low overhead:** Slotted Aloha does not require complex management or control mechanisms, which can help to reduce the overhead and complexity of the network.

Disadvantages of Slotted ALOHA:

- **Low throughput:** The maximum throughput of the Slotted Aloha protocol is relatively low at around 18.4%, which can be limiting for high-bandwidth applications.

- **High collision rate:** The high collision rate in slotted ALOHA can result in a high packet loss rate, which can negatively impact the overall performance of the network.
- **Inefficiency:** The protocol is inefficient at high loads, as the efficiency decreases as the number of nodes attempting to transmit increases.

Experimental Setup:

1. **Pure ALOHA Configuration:**
    - Set up a simple network topology with at least two nodes capable of data communication.
    - Implement the Pure ALOHA protocol on the nodes, specifying parameters such as frame size, transmission time, and acknowledgment time.
2. **Frame Transmission:**
    - Configure the nodes to generate data frames for transmission at random intervals.
    - Implement a collision detection mechanism to identify and handle frame collisions.
3. **Performance Metrics:**
    - Measure the round-trip time (RTT) for successful frame transmissions.
    - Record the number of collisions and identify their impact on the network efficiency.
    - Calculate the throughput of the Pure ALOHA protocol under different data loads.
4. **Varying Network Conditions:**
    - Introduce variations in network conditions such as increased traffic load and frame size.
    - Evaluate the effect of these variations on the protocol's performance.
5. **Data Collection and Analysis:**
    - Use a protocol analyzer tool (e.g., Wireshark) to capture and analyze the network traffic during the experiments.
    - Record the time stamps for frame transmissions, collisions, and acknowledgments.

Data Analysis:

1. Analyze the collected data to calculate the throughput of the Pure ALOHA protocol.
2. Evaluate the efficiency of the protocol by considering the impact of collisions on the overall performance.
3. Compare the observed performance metrics with the theoretical predictions for Pure ALOHA.
4. Identify any limitations or challenges encountered during the experiments.

Conclusion: Summarize the findings of the experiment, discussing the efficiency and limitations of the Pure ALOHA protocol. Provide insights into the impact of varying network conditions on the protocol's performance and draw conclusions regarding its suitability for different scenarios. Discuss potential improvements or modifications to enhance the efficiency of contention-based protocols in shared communication channels.