Interconnection Network: BUS

Dr. A. Sahu
CSE, IIT Guwahati

Bus

- Bus (Common Connection)
- Shared medium
- Same protocol/arbitration as CSMA/CD in MAC layer
- Analysis is bit different but similar

Bus interconnection

Taxonomy of multiple-access protocols

Pure ALOHA Protocol

While there is a new frame A to send DO
1. Send frame A and wait for ACK
2. If after "some" time ACK is not received (timer times out), wait a random amount of time and go to 1.
End
**Frames in a pure ALOHA network**

- Station 1 sends Frame 1.1.
- Station 2 sends Frame 2.1.
- Station 3 sends Frame 3.1.
- Station 4 sends Frame 4.1.

**Analysis of Pure ALOHA**

- Collides with the start of the shaded frame at $t_0$.
- Collides with the end of the shaded frame at $t_0 + 3t$.
- Vulnerable period for the shaded frame is $t_0$ to $t_0 + 3t$.

**Slotted ALOHA**

- Channel is organized into uniform slots whose size equals the frame transmission time.
- Transmission is permitted only to begin at a slot boundary.

Here is the procedure:

1. Send frame A at a slot boundary and wait for ACK.
2. If after “some” time ACK is not received, wait a random amount of time and go to 1.

End
Medium Access Control

- Medium Access Control (MAC):
  - How to share a common medium among the users?
- MAC layer is very important in LANs, nearly all of which use a multiaccess channel as the basis of their communication.

Assumptions

- Infinite population of users
- New frames are generated according to a Poisson distribution with mean \( S \) packets per packet time.
  - Probability that \( k \) packets are generated during a given packet time:
    \[
    \Pr[k] = \frac{S^k e^{-S}}{k!}
    \]

Observation on \( S \)

- If \( S > 1 \), packets are generated at a higher rate than the channel can handle.
- Therefore, we expect \( 0 < S < 1 \)
- If the channel can handle all the packets, then \( S \) is the throughput.

Packet Retransmission

- In addition to the new packets, the stations also generate retransmissions of packets that previously suffered collisions.
- Assume that the packet (new + retransmitted) generated is also Poisson with mean \( G \) per packet time.
  \[
  \Pr[k] = \frac{G^k e^{-G}}{k!}
  \]

Relation between \( G \) and \( S \)

- Clearly, \( G \geq S \)
- At low load, few collisions: \( G \approx S \)
- At high load, many collisions: \( G > S \)
- Under all loads, \( S = GP_0 \)
  where \( P_0 \) is the probability that a packet does not suffer a collision.

Vulnerable Period: Pure Aloha

- Under what conditions will the shaded packet arrive undamaged?
**Throughput**

- Vulnerable period: from \( t_0 \) to \( t_0 + 2t \)
- Probability of no other packet generated during the vulnerable period is:
  \[ P_0 = e^{-2G} \]
- Using \( S = GP_0 \), we get
  \[ S = Ge^{-2G} \]

**Relation between \( G \) and \( S \)**

Max throughput occurs at \( G = 0.5 \), with \( S = 1/(2e) = 0.184 \).

Hence, max. channel utilization is 18.4%.

**Carrier Sense Multiple Access (CSMA)**

- Additional assumption:
  - Each station is capable of sensing the medium to determine if another transmission is underway.

**Non-persistent CSMA**

While there is a new frame A to send do
1. Check the medium
2. If the medium is busy, **wait some time**, and go to 1.
3. (medium idle) Send frame A and wait for ACK
4. If after some time ACK is not received (timer times out), wait a random amount of time and go to 1.
End.

**1-persistent CSMA**

While there is a new frame A to send do
1. Check the medium
2. If the medium is busy, go to 1.
3. (medium idle) Send frame A and wait for ACK
4. If after some time ACK is not received (timer times out), wait a random amount of time and go to 1.
End.

**p-persistent CSMA**

While there is a new frame A to send do
1. Check the medium
2. If the medium is busy, go to 1.
3. (medium idle) With probability \( p \) send frame A and the go to 4, and probability (1- \( p \)) delay one time slot and go to 1.
4. If after some time ACK is not received (timer times out), wait a random amount of time and go to 1.
End.
**Comparison of throughput versus load for various random access protocols.**

- **Persistent and Non-persistent CSMA**
  - CSMA/CD
  - CSMA/CA
  - CSMA/AC

**BUS Protocol: Queue Based**

- **BUS Arbiter**
  - Grand based on Policy/Priority to one S/D of Queue

- **S (Source) and D (Destination)**
  - may be Processor/memory

**Achieved BW on a relative scale**

\[
\frac{1 - (1 - \rho)^n}{n}
\]

**Required BW**

\[ n \cdot \rho \]

**Available BW**

\[ 1 \]

**Achieve BW Per Processor**

\[
\frac{1 - (1 - \rho)^n}{n}
\]

**Effect of re-submitted requests**

- Two State: Accepted or Waiting, Request are accepted with probability \( P_A \)
- Once in accepted state
  - Remain their by another request and having it accepted \( \rho P_A \) or by not making a request \( 1 - \rho \)
  - If requested but rejected with \( \rho (1 - P_A) \)
- A Processor in W state
  - Always resubmit a request and remain in W if it is rejected \( 1 - P_A \)

\[
\begin{align*}
\text{Accept} & : \text{prob} = q_A \\
\text{Wait} & : \text{prob} = q_W
\end{align*}
\]

\[
\begin{align*}
1 - \rho + \rho P_A & = \rho (1 - P_A) \\
1 - P_A & = 1 - \rho (1 - P_A)
\end{align*}
\]
**Effect of re-submitted requests**

- Available BW is \( n\rho_A \)
- Probability of having request accepted \( P_A = \frac{\rho_A}{\rho} \)

\[
\text{BW}=n\rho_A = 1-(1-a)^n
\]

\[
P_A = \frac{\rho_A}{\rho} \Rightarrow a = \frac{\rho}{\rho + \frac{\rho_A}{\rho}(1-\rho)}
\]

- Both the above equation are interdependent
  - So solve iteratively to get Actual offered

**Waiting time**

Waiting time = \( i \times T_{\text{bus}} \)

if request is rejected \( i \) times and accepted on \( (i+1)^{th} \) attempt

probability of this = \( (1-P_A)^i \times P_A \)

Expected value of waiting time = \( T_p = \sum_{i=1}^{\infty} i \times T_{\text{bus}} \times (1-P_A)^i \times P_A \)

\[
= T_{\text{bus}} \times P_A \times \sum_{i=1}^{\infty} i \times (1-P_A)^i = T_{\text{bus}} \times P_A \times \frac{1-P_A}{(1-(1-P_A))}
\]

\[
= \frac{1-P_A}{P_A} T_{\text{bus}} = \frac{\rho - \rho_A}{\rho_A} T_{\text{bus}}
\]

**Shared BUS: BW per Processor**

**Shared BUS: Utilization**

**Verdict: Share BUS**

- Utilization saturates with number of requests
- Saturate more quickly as processor increases

- So it is not scalable with number of processor
- If number of processor > (8 or 10), Bus interconnection is Bad