Lecture #13/14 : Packet Multiple Access: The Aloha protocol

Eytan Modiano
Massachusetts Institute of Technology

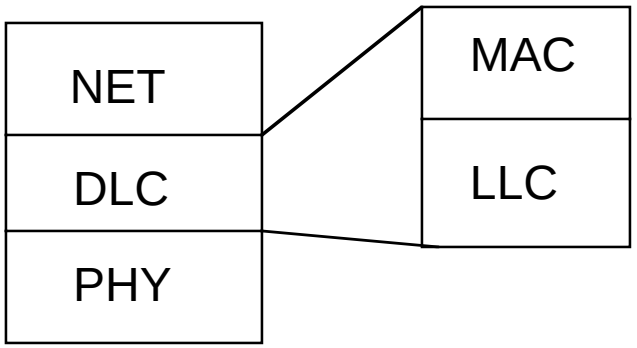
Multiple Access

- **Shared Transmission Medium**
  - **a receiver can hear multiple transmitters**
  - **a transmitter can be heard by multiple receivers**

- **the major problem with multi-access is allocating the channel between the users; the nodes do not know when the other nodes have data to send**
  - **Need to coordinate transmissions**

Examples of Multiple Access Channels

- **Local area networks (LANs)**
  - **Traditional Ethernet**
  - **Recent trend to non-multi-access LANs**
- **satellite channels**
- **Multi-drop telephone**
- **Wireless radio**



- **Medium Access Control (MAC)**
  - **Regulates access to channel**
- **Logical Link Control (LLC)**
  - **All other DLC functions**

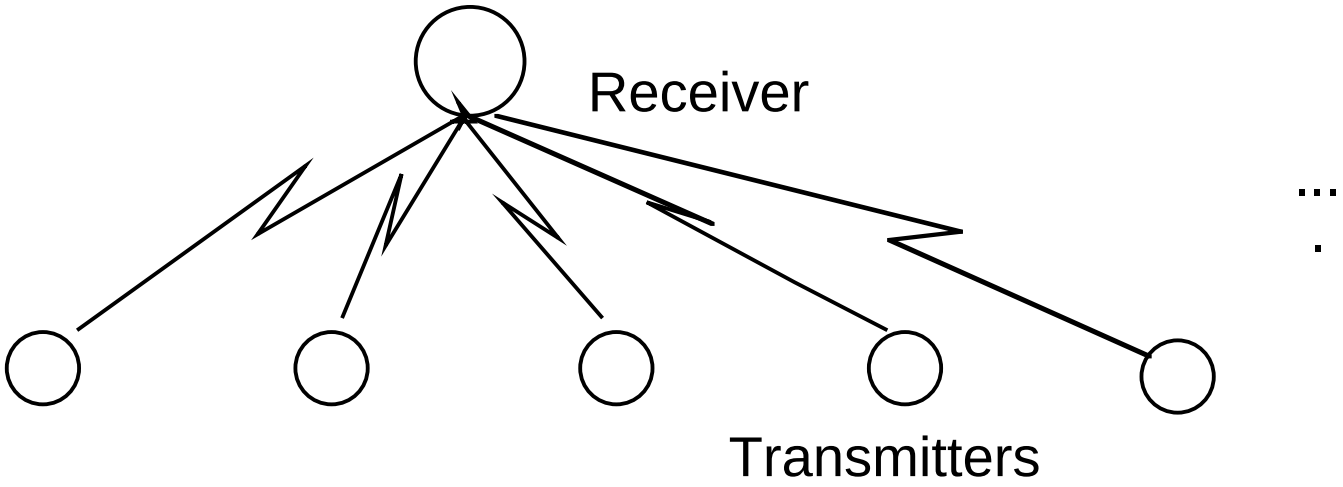
Approaches to Multiple Access

- **Fixed Assignment (TDMA, FDMA, CDMA)**
  - **each node is allocated a fixed fraction of bandwidth**
  - **Equivalent to circuit switching**
  - **very inefficient for low duty factor traffic**

- **Contention systems**
  - **Polling**
  - **Reservations and Scheduling**
  - **Random Access**

Aloha

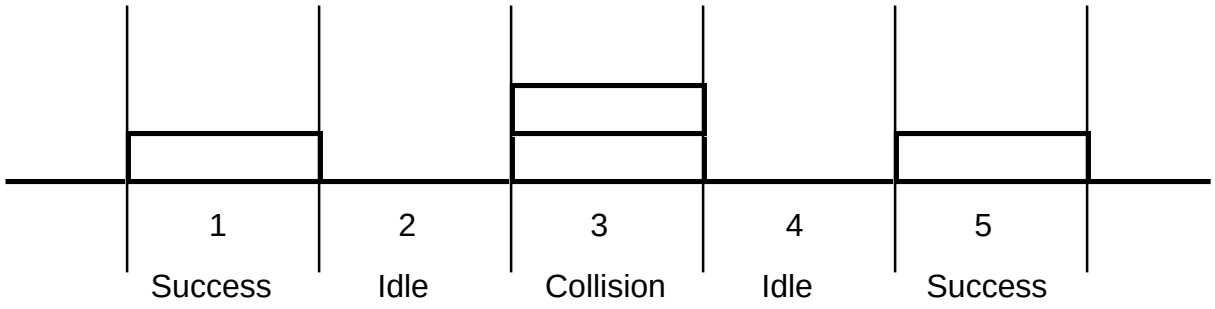
Single receiver, many transmitters



E.g., Satellite system, wireless

Slotted Aloha

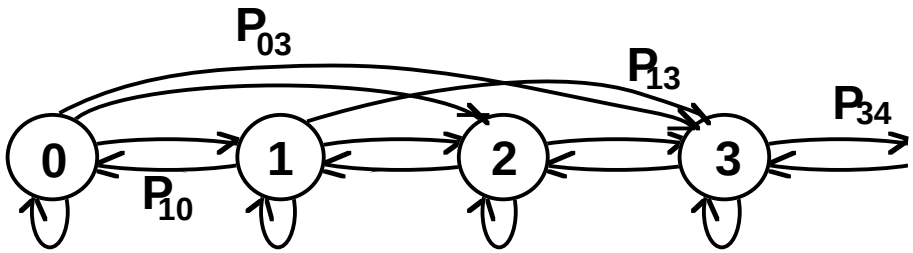
- **Time is divided into “slots” of one packet duration**
  - **E.g., fixed size packets**
- **When a node has a packet to send, it waits until the start of the next slot to send it**
  - **Requires synchronization**
- **If no other nodes attempt transmission during that slot, the transmission is successful**
  - **Otherwise “collision”**
  - **Collided packet are retransmitted after a random delay**



Slotted Aloha Assumptions

- Poisson external arrivals
- No capture
  - Packets involved in a collision are lost
  - Capture models are also possible
- Immediate feedback
  - Idle (0) , Success (1), Collision (e)
- If a new packet arrives during a slot, transmit in next slot
- If a transmission has a collision, node becomes backlogged
  - while backlogged, transmit in each slot with probability q_r until successful
- Infinite nodes where each arriving packet arrives at a new node
  - Equivalent to no buffering at a node (queue size = 1)
  - Pessimistic assumption gives a lower bound on Aloha performance

Markov chain for slotted aloha



- state (n) of system is number of backlogged nodes.

  pi,i-1 = prob. of one backlogged attempt and no new arrival

  pi,i =prob. of one new arrival and no backlogged attempts or no new arrival and no success

  pi,i+1= prob of one new arrival and one or more backlogged attempts

  pi,i+j = Prob. Of J new arrivals and one or more backlogged attempts or J+1 new arrivals and no backlogged attempts

- Steady state probabilities do not exists
  - Backlog tends to infinity => system unstable
  - More later

slotted aloha

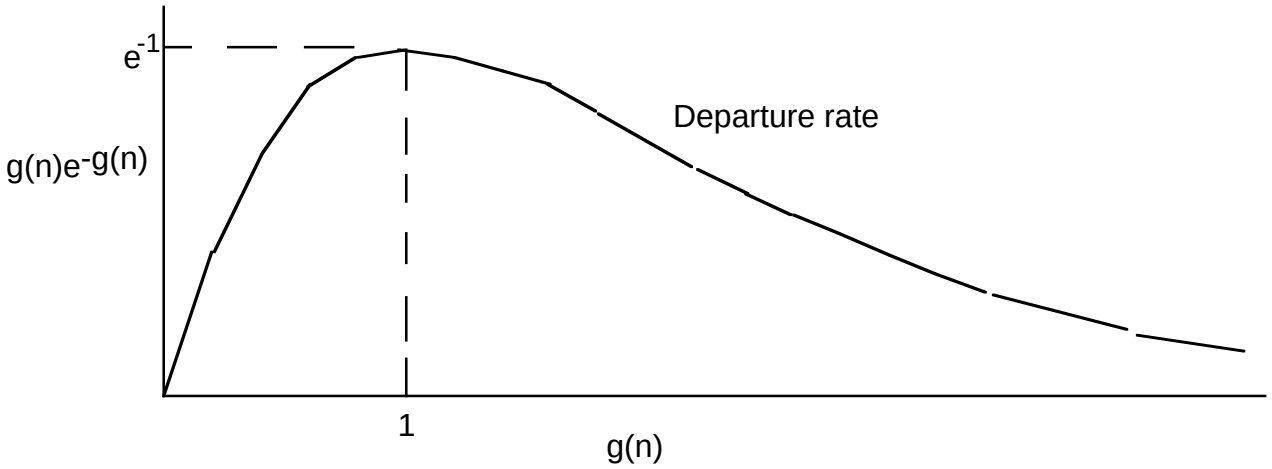
- let g(n) be the attempt rate (the expected number of packets transmitted in a slot) in state n

  $g(n) = \lambda + nqr$

- The number of attempted packets per slot in state n is approximately a Poisson random variable of mean g(n)
  - P (m attempts) = $g(n)^m e^{-g(n)}/m!$
  - P (idle) = probability of no attempts in a slot = $e^{-g(n)}$
  - p (success) = probability of one attempt in a slot = $g(n)e^{-g(n)}$
  - P (collision) = P (two or more attempts) = 1 - P(idle) - P(success)

Throughput of Slotted Aloha

- **The throughput is the fraction of slots that contain a successful transmission = P(success) = $g(n)e^{-g(n)}$**
  - **When system is stable throughput must also equal the external arrival rate (λ)**



  - **What value of g(n) maximizes throughput?**

$$\frac{d}{dg(n)} g(n)e^{-g(n)} = e^{-g(n)} - g(n)e^{-g(n)} = 0$$

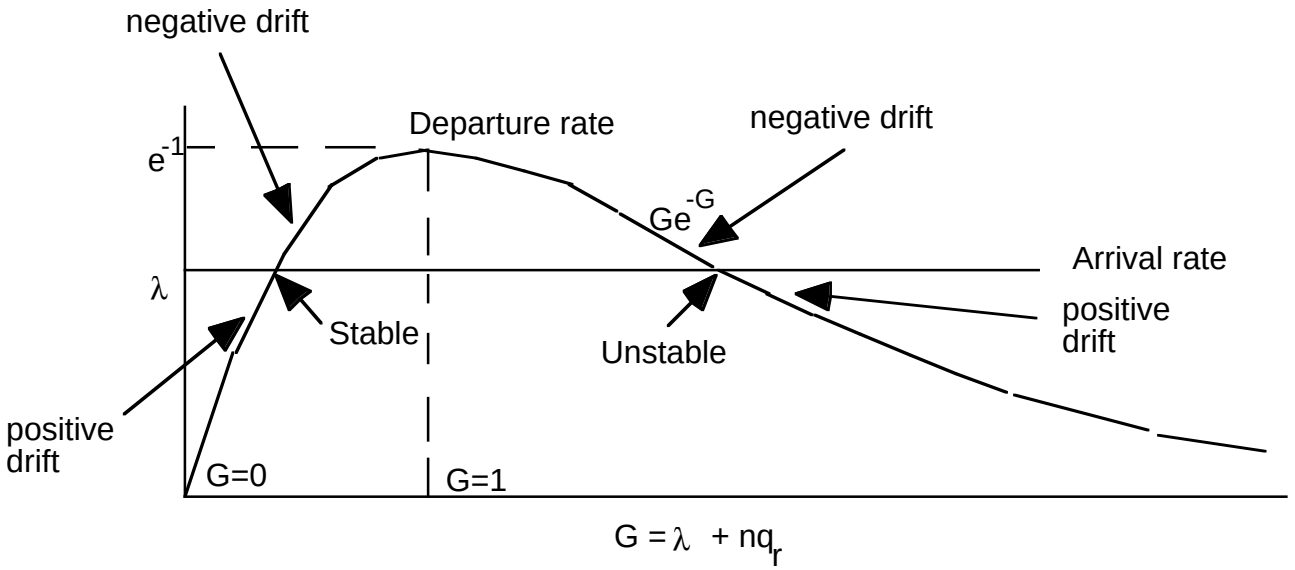
$$\Rightarrow g(n) = 1$$

$$\Rightarrow P(success) = g(n)e^{-g(n)} = 1/e \approx 0.36$$

  - **g(n) < 1 => too many idle slots**
  - **g(n) > 1 => too many collisions**
  - **If g(n) can be kept close to 1, an external arrival rate of 1/e packets per slot can be sustained**

Instability of slotted aloha

- **if backlog increases beyond unstable point (bad luck) then it tends to increase without limit and the departure rate drops to 0**
- **Drift in state n, D(n) is the expected change in backlog over one time slot**
  - **$D(n) = \lambda - P(\text{success}) = \lambda - g(n)e^{-g(n)}$**



Stabilizing slotted aloha

- choosing q_r small increases the backlog at which instability occurs (since $g(n) = \lambda + nq_r$), but also increases delay (since mean retry time is 1/qr)
- solution: estimate the backlog (n) from past feedback
  - Given the backlog estimate, choose q_r to keep g(n) = 1

    Assume all arrivals are immediately backlogged

    $g(n) = nq_r$, $P(\text{success}) = nq_r (1-q_r)^{n-1}$

    To maximize P(success) choose $q_r = \min\{1,1/n\}$
  - When the estimate of n is perfect:

    idles occur with probability 1/e,
    successes with 1/e, and
    collisions with 1-2/e.
  - When the estimate is too large, too many idle slots occur
  - When the estimate is too small, too many collisions occur
- Nodes can use feedback information (0,1,e) to make estimates
  - A good rule is increase the estimate of n on each collision, and to decrease it on each idle slot or successful slot

    note that the increase on a collision should be $(e-2)^{-1}$ times as large as the decrease on an idle slot

stabilized slotted aloha

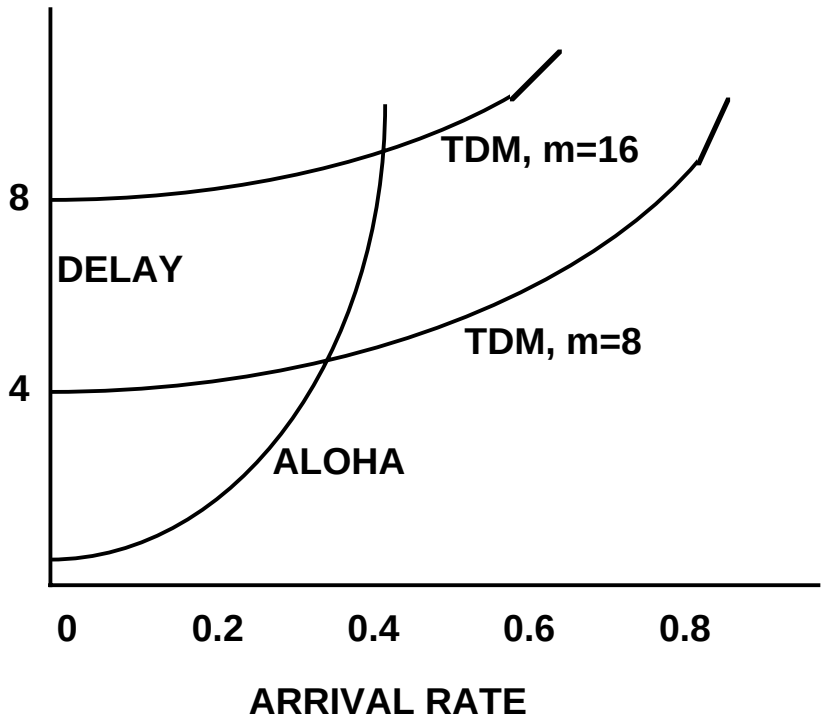
- assume all arrivals are immediately backlogged
  - $g(n) = nq_r$ = attempt rate
  - $p(\text{success}) = nq_r (1-q_r)^{n-1}$

    for max throughput set $g(n) = 1 \Rightarrow q_r = \min\{1, 1/n'\}$
    where n' is the estimate of n
  - Let n_k = estimate of backlog after k^{th} slot

$$n_{k+1} = \begin{cases} \max\{\lambda, n_k+\lambda-1\} & \text{idle or success} \\ n_k+\lambda+(e-2)^{-1} & \text{collision} \end{cases}$$

  - Can be shown to be stable for $\lambda < 1/e$

TDM vs. slotted aloha



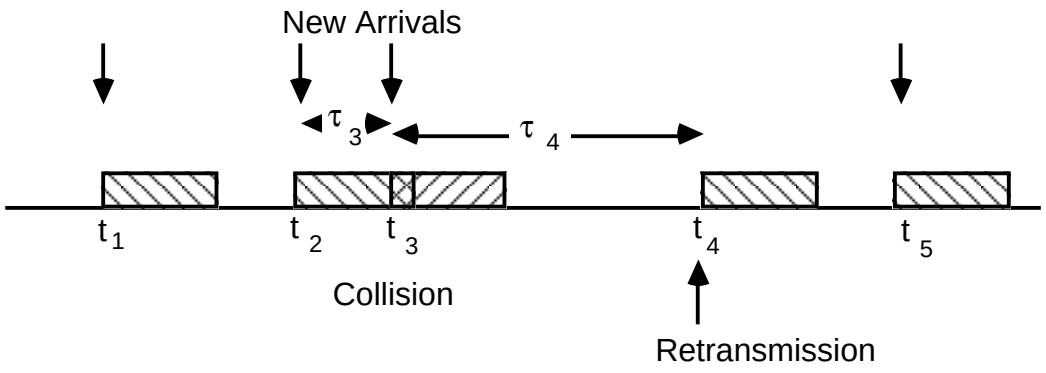
- Aloha achieves lower delays when arrival rates are low
- TDM results in very large delays with large number of users, while Aloha is independent of the number of users

Pure (unslotted) Aloha

- New arrivals are transmitted immediately (no slots)
  - No need for synchronization
  - No need for fixed length packets

- A backlogged packet is retried after an exponentially distributed random delay with some mean 1/x
- The total arrival process is a time varying Poisson process of rate $g(n) = \lambda + nx$ (n = backlog, 1/x = ave. time between retransmissions)

- Note that an attempt suffers a collision if the previous attempt is not yet finished ($t_i - t_{i-1} < 1$) or the next attempt starts too soon ($t_{i+1} - t_i < 1$)



Throughput of Unslotted Aloha

- **An attempt is successful if the inter-attempt intervals on both sides exceed 1 (for unit duration packets)**
  - **P(success) = $e^{-g(n)}$ $e^{-g(n)}$ = $e^{-2g(n)}$**
  - **Throughput (success rate) = g(n) $e^{-2g(n)}$**
  - **For max throughput at g(n) = 1/2, Throughput = 1/2e ~ 0.18**
  - **stabilization issues are similar to slotted aloha**
  - **advantages of unslotted aloha are simplicity and possibility of unequal length packets**

Splitting Algorithms

- **More efficient approach to resolving collisions**
  - **Simple feedback (0,1,e)**
  - **Basic idea: assume only two packets are involved in a collision**

    Suppose all other nodes remain quiet until collision is resolved, and nodes in the collision each transmit with probability 1/2 until one is successful

    On the next slot after this success, the other node transmits

    The expected number of slots for the first success is 2, so the expected number of slots to transmit 2 packets is 3 slots

    Throughput over the 3 slots = 2/3

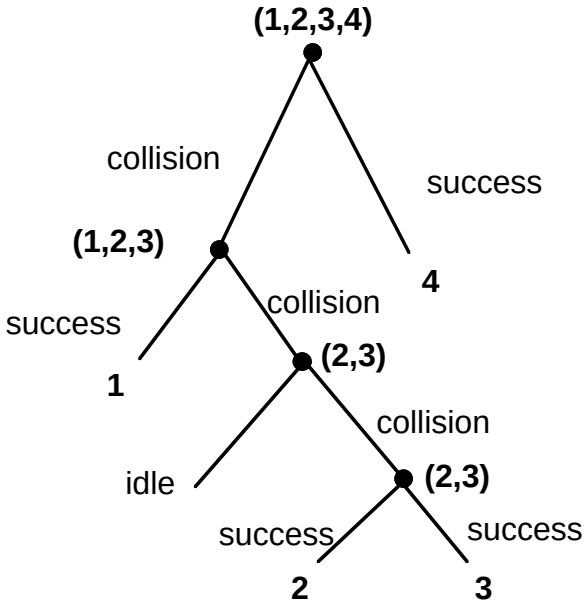
  - **In practice above algorithm cannot really work**

    Cannot assume only two users involved in collision

    Practical algorithm must allow for collisions involving unknown number of users

Tree algorithms

- **After a collision, all new arrivals and all backlogged packets not in the collision wait**
- **Each colliding packet randomly joins either one of two groups (Left and Right groups)**
  - **Toss of a fair coin**
  - **Left group transmits during next slot while Right group waits**
    - **If collision occurs Left group splits again (stack algorithm)**
    - **Right group waits until Left collision is resolved**
  - **When Left group is done, right group transmits**



Notice that after the idle slot, collision between (2,3) was sure to happen and could have been avoided

Many variations and improvements on the original tree splitting algorithm

Throughput comparison

- stabilized pure aloha T = 0.184 = (1/(2e))
- stabilized slotted aloha T = 0.368 = (1/e)
- Basic tree algorithm T = 0.434
- Best known variation on tree algorithm T = 0.4878
- Upper bound on any collision resolution algorithm with (0,1,e) feedback T <= 0.568
- TDM achieves throughputs up to 1 packet per slot, but the delay increases linearly with the number of nodes