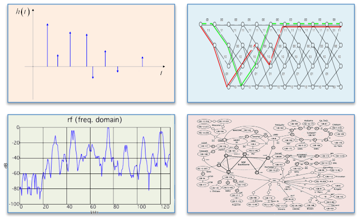


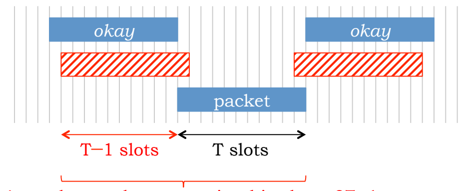
INTRODUCTION TO EECS II
DIGITAL COMMUNICATION SYSTEMS

6.02 Fall 2011
Lecture #18

- Unslotted Aloha
- Carrier sense multiple access, contention windows
- err, ummm… oops!

Unslotted Aloha

- Packets take T time slots to transmit
  - As slots get smaller and T grows, approximates transmission at arbitrary times.
- Collisions are no longer "perfect"
  - Any overlap between multi-slot packets is a collision
  - Larger window of vulnerability to other transmissions



Utilization in Unslotted Aloha

Probability of no transmission for 2T−1 slots:

$$(1-p)^{2T-1}$$

Probability of a sender experiencing no collisions:

$$\geq \left[p(1-p)^{2T-2}\right]\left[(1-p)^{2T-1}\right]^{N-1} = p(1-p)^{(2T-1)N-1}$$

= for nodes that try to send new packet while busy with last one!

Utilization = throughput/maximum rate:

$$U_{\text{unslotted Aloha}} \geq \frac{Np(1-p)^{(2T-1)N-1}}{1/T} = TNp(1-p)^{(2T-1)N-1}$$

U_{max} for Unslotted Aloha

Maximization with respect to p:

$$\log(\ldots) = \text{const} + \log(p) + \left[(2T-1)N-1\right]\log(1-p)$$

Derivative:

$$\frac{1}{p} + \frac{(2T-1)N-1}{1-p}, \text{ which equals 0 at } p = \frac{1}{(2T-1)N}$$

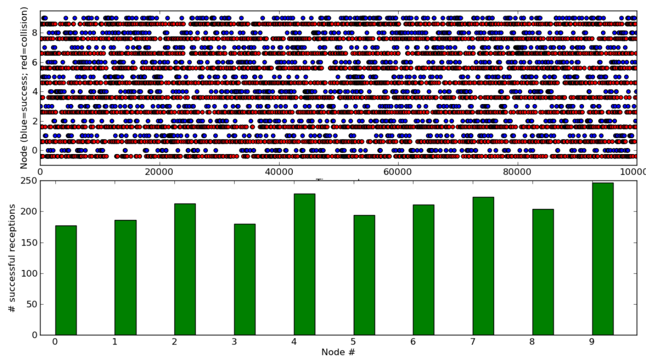
Plugging back into U:

$$U_{\max} = \frac{T}{2T-1}\left(1-\frac{1}{(2T-1)N}\right)^{(2T-1)N-1}$$

For large N: $U_{\max} \approx \left(\frac{T}{2T-1}\right)\frac{1}{e}$ For large N, T: $U_{\max} \approx \frac{1}{2e}$

Half the utilization of slotted Aloha;
makes sense: twice the window of vulnerability

Simulation of Unslotted Aloha



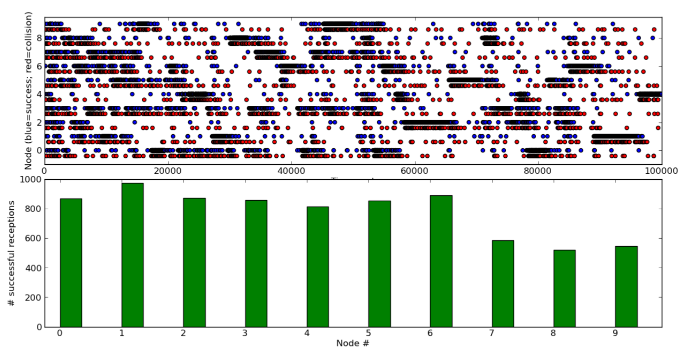
Utilization = .21, Fairness = .99

```
python PS6_3.py -r -n 10 -t 100000 --pmin=.005 --pmax=0.8 -s 10
```


Carrier Sense

- Reduce collisions with on-going transmissions by transmitting only if channel appears not to be busy.
- For large T (slots/packet) if channel is busy this cycle, the same sender will probably be transmitting more of their packet next cycle
- When the channel is idle, there's no chance of interrupting an on-going transmission.
- That leaves the possibility of colliding with another transmission that starts at the same time – a one slot window of vulnerability, not 2T-1 slots.
- Expect collisions to drop dramatically, utilization to be quite a bit better, although a "wasted" slot is now necessary
- Busy = detect energy on channel. On wireless channels, transmitters turn on carrier to transmit (we'll learn more about this after break), hence the term "carrier sense".

Simulation of Carrier Sense



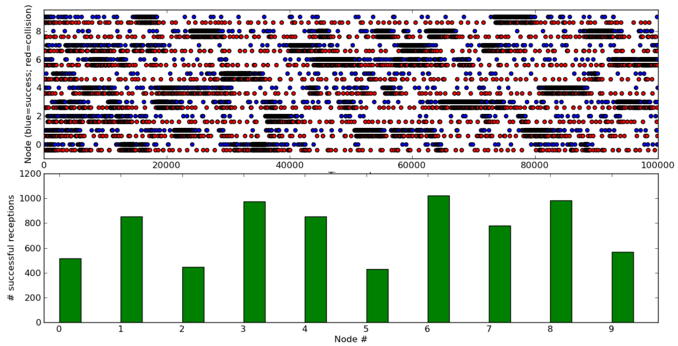
Utilization = .78, Fairness = .96

```
python PS6_4.py -r -n 10 -t 100000 --pmin=.005 --pmax=0.8 -s 10
```


Contention Windows

- Contention Window: parameter is some integer *CW*
- When node wants to transmit, it picks a random number *r* uniformly in [1,*CW*] and sends after the r^{th} <u>idle</u> slot from the current time.
- If transmission succeeds: $CW \leftarrow max(CW_{min}, CW/2)$
  If transmission collides: $CW \leftarrow min(CW_{max}, CW*2)$
- Node is guaranteed to attempt a transmission within CW slots. With the earlier scheme, there was always the chance (though exponentially decreasing) that a node may not transmit within some fixed number of time slots.

Simulation of Contention Windows



Utilization = .74, Fairness = .92

```
python PS6_5.py -r -n 10 -t 100000 -W 256 -s 10
```


Summary of MAC Protocols

- Goal of MAC protocols is to maximize utilization and fairness
- TDMA is a good choice when nodes are all (or mostly) backlogged
  - Round-robin sharing provides known communication capacity and bounded wait
  - It's precisely fair, 100% utilization if all nodes have packets
  - Poor choice when traffic is bursty or if some nodes have a higher offered load than others
  - Hard to implement in a fully distributed way (easier with "master", like a base station or access point)
- Contention protocols dynamically adapt to changing traffic
  - Distributed protocol (each node makes its own decisions based on transmission experience) avoids cost of centralized controller having to know which nodes have packets to send
  - Parameter (p or CW) that controls when packets are sent is adjusted so that Prob(sending packet) is lowered when collisions are detected and raised when transmissions are successful.

Summary (cont'd.)

- Slotted Aloha – very simple rule: transmit with probability p.
  - Dynamic adjustment of p "stabilizes" the protocol.
    - Use binary exponential backoff to adjust p downward
  - Utilization maximized when p = 1/(number of backlogged nodes)
  - For large numbers of backlogged nodes U ≈ 1/e ≈ 37%
  - For fairness: $p_{min} \leq p \leq p_{max}$
- Unslotted Aloha – packets take multiple time slots to send, models transmissions at arbitrary times
  - Gets half of the max utilization of slotted Aloha (at large N) due to doubled window of vulnerability to collisions
  - Carrier sense avoids collisions from packets once transmission has started → much better utilization
  - Fairness still requires bounds on p