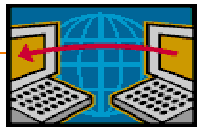


Link-Layer Contention

EE 122: Intro to Communication Networks

Fall 2007 (WF 4-5:30 in **Cory 277**)

Vern Paxson

TAs: Lisa Fowler, Daniel Killebrew & Jorge Ortiz

http://inst.eecs.berkeley.edu/~ee122/

Materials with thanks to Jennifer Rexford, Ion Stoica, and colleagues at Princeton and UC Berkeley

1

Our Story So Far

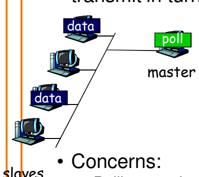
- Single shared broadcast channel
  - Avoid having multiple nodes speaking at once
  - Otherwise, collisions lead to garbled data
- Multiple access mechanism
  - Distributed algorithm for sharing the channel
  - Algorithm determines which node can transmit
- Classes of techniques
  - Channel partitioning: divide channel into pieces
    - o TDMA and FDMA (time-division & frequency-division)
  - Taking turns: scheme for trading off who gets to transmit
  - Random access: allow collisions, and then recover
    - o *Optimizes for the common case of only one sender*

2

"Taking Turns" MAC protocols

Polling

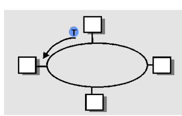
- Master node "invites" slave nodes to transmit in turn



- Concerns:
  - Polling overhead
  - Latency
  - Single point of failure (master)

Token passing

- Control token passed from one node to next sequentially
- Node must have token to send
- Concerns:
  - Token overhead
  - Latency
  - Single point of failure (token)



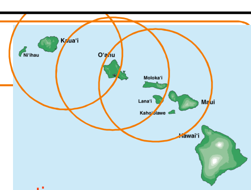
3

Random Access Protocols

- When node has packet to send
  - Transmit at full channel data rate
  - No *a priori* coordination among nodes
- Two or more transmitting nodes ⇒ collision
  - Data lost
- Random access MAC protocol specifies:
  - How to detect collisions
  - How to recover from collisions
- Examples
  - ALOHA and Slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

4

Slotted ALOHA



Assumptions

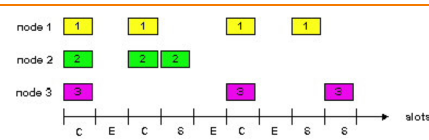
- All frames same size
- Time divided into equal slots (time to transmit a frame)
- Nodes are synchronized
- Nodes begin to transmit frames *only at start of slots*
  - No *carrier sense*
- If two or more nodes transmit, all nodes detect collision

Operation

- When node obtains fresh frame, transmits in next slot
- No collision: node can send new frame in next slot
- Collision: node retransmits frame in each subsequent slot with probability **p** until success

5

Slotted ALOHA



Pros

- Single active node can continuously transmit at full rate of channel
- Highly decentralized: only slots in nodes need to be in sync
- Simple

Cons

- Collisions, wasting slots
- Idle slots
- Nodes may be able to detect collision in less than time to transmit packet
- Clock synchronization

6

Slotted Aloha efficiency

Efficiency : long-run fraction of successful slots (many nodes, all with many frames to send)

- *Suppose:* N nodes with many frames to send, each transmits in slot with probability *p*
- Probability that given node has success in a slot = $p(1-p)^{N-1}$
- Probability that *any* node has a success = $Np(1-p)^{N-1}$

- Maximum efficiency: find *p** that maximizes $Np(1-p)^{N-1}$
- For many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

Maximum efficiency = 1/e ≈ .37

At best: under heavy load, channel wasted 63% of the time!

!

Can also show that without slots, efficiency drops to 1/(2e) ≈ .18

7