Administrivia

- Project extension to 10/3 (Wednesday); will be graded by appointment
  - Little over 1 week left
- Looking ahead; midterm 10/10
- Homework will be assigned on Wednesday
Upcoming Reading

- Several topics in switched networks I would like us to explore
  - Routing: both technical and economic (peering/settlement)
  - Routing: measured performance/reasons for it
  - Big Fast Router design
    * Router algorithmics
    * Router design
  - Next generation IP

- We’ll be starting the reading little bit early (sorry!)

- Reading for next week on class webpage
  - Peering and Settlements paper (no critique)
  - End-to-end routing behavior paper (no critique)
  - Route inflation paper (no critique)
Last Time/Today

- Last Time
  - Started Medium Access Control (MAC)
    - Static assignment: TDMA, FDMA, CDMA – not efficient in most cases
    - Reservation vs. Contention
    - Contention: ALOHA, Slotted Aloha, CSMA, CD
    - Characteristics of the link may restrict solutions (delay affects availability of carrier sense/collision detection...)
    - Ethernet as an example of contention

- Today: finish MAC; Bloom filters paper; maybe start switched networks
Ethernet Frame Format

<table>
<thead>
<tr>
<th>64</th>
<th>48</th>
<th>48</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>Dest addr</td>
<td>Src addr</td>
<td>Type</td>
<td>Body</td>
</tr>
</tbody>
</table>

- Preamble: alternating 0’s and 1’s for synchronization
  - IEEE 803.2 uses count (in bytes) instead of type field

- Addresses are globally unique 6 bytes placed in firmware by the manufacturer; Example:
  5 : 8b : 40 : 2 : cf : 4a

- Multicast address starts with 1

- Broadcast address all 1’s
  - How does a sender know the receiver’s address? (ARP; later)

- Promiscuous mode – listen to all packets
Transmission Algorithm

• If line is idle:
  – Send immediately; upper bound message is 1500 bytes, lower bound is 46 bytes (why do we need a lower bound?)
  – If a collision is detected (by one of the senders), it transmits a jamming sequence (32-bits)
  – Must wait 51\(\mu\)-sec between back to back frames; why?

• If line is busy
  – Wait until idle and transmit immediately

• This is an example of a medium access algorithm (what we called a fair distributed algorithm for sharing the medium)
• What is the maximum time that a node has to wait before it detects a collision?

• What happens on a collision?
  – Node waits up to 51.2 μ-sec, senses the line, then transmits if idle
  – If another collision, node waits up to 102.4 μ-sec, then tries again
  – Another collision? Wait up to 204.8 μ-sec
  – **Exponential back off;** gives up after 16 retries, timeout capped at 10
Contention Discussion

• Contention based protocols are good under some circumstances
  – Low probability of collisions (low load)
  – Low cost of collision (low $\tau$)

• Contention can be bad
  – Too much time wasted contending for the medium
  – Unpredictable performance

• Role of backoff
Reservation Based Protocols

- Contention based protocols are good under some circumstances
  - Low probability of collisions (low load)
  - Low cost of collision (low $\tau$)

- Contention can be bad
  - Too much time wasted contending for the medium
  - Unpredictable performance

- Reservation based protocols – collaborate on reserving the use of the medium
  - Advantage – eliminate collisions
  - Disadvantage – reservation overhead paid even when not necessary
Reservation

• How is reservation implemented?

• There is an overhead for reservation – what happens if only a few nodes have traffic?

• What do you think is the ideal case for reservation? How does it compare to contention?
IEEE 802.4 – Token Bus

• LAN standard

• physically similar to ethernet (single shared bus)

• Reservation based.
  – Station that has token can transmit
  – Forwards token to next station in schedule

• A station interested in transmitting has to register to be added to the schedule

• How will this LAN perform relative to ethernet?
Token Ring: FDDI/IEEE 802.5

- A ring version of 802.4
- Token Ring Networks – standard in wide use at one time
  - Pronet: 10Mbps and 80Mbps rings
  - Original IBM Token Ring: 4Mbps
  - IEEE 802.5 – 16Mbps
  - Fiber Distributed Data Interface (FDDI) – 100 Mbps
How does it work?

• Idea:
  – Frames flow in one direction: upstream to downstream
  – A special bit pattern (token) rotates around ring
  – Must capture token before transmitting
  – Release when done (immediate vs. delayed)
  – Remove your frame when comes back around; re-insert the token

• How does this compare to ethernet? Token bus?

• What if a node is turned off/has a failure?
FDDI continued

- Each station imposes a delay
- Maximum of 500 stations
- Upper limit of 100km
- Uses 4B/5B encoding (802.5 uses Manchester)
- Can be implemented over copper (CDDI)
Problem

- What if only one station is active?
  - Bad: token has to go all the way around between every transmission of the packet

- FDDI has an interesting solution (in a second)

- What if token is lost?
Timed Token Algorithm

- Token Holding Time (THT): upper limit on how long a station can hold the token
- Token Rotation Time (TRT): time it takes for the token to traverse the ring
- \( TRT \leq ActiveNodes \times THT + RingLatency \)
- Target Token Rotation Time (TTRT): agreed upon upper bound on TRT
Timed Token – Algorithm

- Each node measures TRT between successive arrivals of the token
- If measured TRT is bigger than TTRT then the token is late; do not send data
- If TRT is less than TTRT, the token is early, so OK to send data
- Two classes of traffic defined:
  - Synchronous data: can always send; exception:
    * Limit of TTRT synchronous data per rotation
  - Asynchronous data: can send only if token is early
- Worst case: 2 TTRT between seeing tokens (why?)
- Not possible to have back to back rotations that take 2 TTRT since asynchronous is not sent if token is already late
Satellite Networks

- Ground station transmits via uplink to satellite; satellite broadcasts down (downlink) to all stations

- Different types of satellite “constellations”
  - Geostationary Earth Orbit (GEOs) are very high (> 22,000 miles). Fixed orbit relative to earth
  - Low Earth Orbit (LEOs) are low orbit (few hundred miles), but move relative to earth. Latency much lower
  - MEO’s in the middle

- “Long fat pipes”: delay is a few hundred millisecs (270 msec typically used); bandwidth can be high
Satellite Networks

- Carrier sense not feasible:
  - Can’t sense uplink, it is a point to point connection
  - Sensing downlink results in sensing what was transmitted 270 msec ago
  - For same reason collision detection not useful (what is the equivalent of ethernets 46bytes minimum packet length?)

- What are our options? Given that CSMA and CD dont work:
  - Either Use ALOHA
  - Or, need some reservation based protocol
  - Claim: Static allocation (FDMA, TDMA, CDMA) methods are ultimate reservation protocol
Another Example, Wireless Networks

- High Bit Error Rate (BER) – frequently lost frames
  - unshielded medium – interference
  - fading, multi-path effect
- Low data rate
  - Limited available bandwidth
  - low signal to noise ratio
  - strict FCC regulations
- Carrier sense possible? Collision detection?
- Complex connectivity pattern
  - shared medium – sharing nonuniform
  - connectivity depends on transmission power, distance, and surrounding environment
  - User mobility complicates things
- Two types of wireless networks (Last hop and Ad hoc)
Last Hop Networks

- Only last hop is wireless
- Every node communicates with a base station (or access point)
- Base station connected to a wired network
- Base station schedules access to the medium
  - In IEEE 802.11, PCF, the base station polls every node asking it if it has data to transmit – what is this MAC?
Last Hop Networks

- Mobility is handled by hand-offs to another base station
  - Soft or hard handoff

- Problem: collision among different “cells”

- Problem: Bandwidth is limited; how can we maximize throughput?

- How do cellular phone networks solve this problem?
Ad Hoc Networks

• All nodes are mobile, no base station, no infrastructure

• Recall, Carrier sense is imprecise
  – You are sensing the medium at the sender, but not at the receiver

• Collision detection not possible
  – Contention is expensive

• Reservation also difficult...no base station. What to do?
  – Aloha?
Ad Hoc MAC

- Idea: Use Busy Tone
  - Sense the medium; if it is idle, transmit a busy tone to ensure no one else transmits
  - Two problems (Klienrock, 1976): Hidden Terminal and Exposed Terminal
CSMA/CA

- Sender Broadcasts Request to Send (RTS)
- Receiver responds with Clear To Send (CTS), including the length of transmission
- Sender sends message if it receives CTS
- Any nodes receiving the CTS do not transmit for the transmission length
- Nodes receiving RTS, but not CTS know they are far enough and can transmit
Limited Contention Protocols

- CSMA great at low loads, but very bad at high loads
- Reservation great at high loads, but bad at low/bursty loads
- Limited Contention Protocols attempt to combine best of both worlds
- Adaptive Tree Walk
What about using multiple channels?

- Everything so far uses a single channel

- What about having multiple channels?
  - How do communicating nodes find each other?
  - Would you be able to communicate to multiple nodes on different channels concurrently?

- Example: Cellular phone networks

- Space Division Multiple Access (SDMA)
Bloom Filters Survey Paper

- Why read this paper?
  - Bloom filters is a flexible/powerful technique with many networking applications

- What are Bloom filters?
Properties of Bloom Filters

- Determine set membership...
- ...efficiently
- small number of false positives tolerated
- Based on hashing ideas
- Extensible and simple to program
Bloom Filter Principle

Whenever a list or set is used, and space is a consideration, a Bloom filter should be considered. When using a Bloom Filter, consider the potential effects of false positives.
How do they Work

- m bit vector

- k truely random hash functions

- Hash a set member $x$ using each of the k hashes and set the corresponding bits in m

- To determine if $x$ is a member of the set, check all $k$ bits; if they are 1 then $x$ is likely a member

- False positives may happen
Analysis

- Probability that a bit position is zero
  \[ p' = (1 - \frac{1}{m})^{kn} \approx e^{-\frac{kn}{m}} = p \]

- Probability of false positive (approx)
  \[ (1 - e^{-\frac{kn}{m}})^k = (1 - p)^k \]
Picking the right $k$

- By taking the derivative and making it $= 0$, the false positives are minimized if $p = 0.5$

- $k = \frac{m}{n} \ln(2)$

- False positive rate is $0.5^k$

- Example: $m = 8n$, $k=6$, false positive rate is $1.5\%$ (all approximate)
Comparison with Hashing

• Hashing to represent a set
  – Hash elements into $\Theta(\log n)$ bits
  – Set is a sorted list of hash values
  – Binary search, $O(\log n)$

• Going with 2 (log n) bits

• False positives = probability of collision
  – $\Pr$ (two elements colliding) = $\frac{1}{n^2}$
  – $\Pr$ (false positive) = $\frac{1}{n}$
Hashing

- Hashing
  - Memory requirement \( \Theta(n \log n) \)
  - Search time is \( O(\log n) \) (\( O(1) \) for Bloom filters)
  - Asymptotically vanishing error probability
    * Error probability constant with constant number of bits per element

- Practical considerations make Bloom filters attractive
Tricks and Extensions

- Union of two filters is their OR (same length and functions)

- Half the size by folding the two halves and OR’ing them (mask MSB)

- Compressed Bloom filters
  - Bloom filters not optimal in space for the error rate they provide (analysis in paper)
  - 0.5 p makes compression not useful
  - Idea: use bigger sparser filters that can then be compressed to reduce overall size
Counting Filters

- Problem: can add (OR), but cannot delete (why?)

- Use count instead of single bit to allow delete

- Small count number (e.g., 16) provides almost zero probability of overflow

- Other extensions (not in this paper) include
  - spectral bloom filters (SIGMOD paper): provide element counts, not just memberships for multisets (Sets with duplicates)
  - bloomier filters (SODA 2004 paper): generalizes bloom to any function, not just set membership
Historical Applications

• Dictionaries
  – Store a bloom filter of valid words
  – Much smaller than the full dictionary
  – Faster search
  – E.g., spell checkers; or list of unsuitable passwords
  – But what is the price of false positives? Acceptable?

• Databases
  – Faster semi-join in distributed databases
  – Tracking differential files (used for batch updates)
Networking Applications – Distributed Caching

- Summary cache in distributed web caches

- Situation: Cache miss at proxy; need to determine if other proxies have the page
  - But how to know the contents of the other proxies?
    - Is this a good fit for bloom filters?
    - Set membership with space constraints problem?
    - What about false positives?
    - Basic vs. counting bloom filters
P2P/Overlay Networks

- Locating Objects
  - Distributed hashes suitable for extremely dynamic networks large scale P2P networks (we will discuss them later)
  - Can use bloom filters for smaller size more stable networks
  - Keep a summary of the objects at each node; use 8 bits per object instead of a full object pointer
  - Passes the bloom principle test?
  - PlanetP uses this idea
Approximate Set Reconciliation

- We want to reconcile the items at node A with those in B (we want to send B the items that A has but it doesn’t have)

- B sends a bloom filter to A

- A checks its items against the bloom filter, and sends the ones that are not members only

- Set intersection targets identifying the opposite (objects both in A and B)
Resource Routing

• Arrange the nodes in the network in a tree

• Nodes have
  – BF for its objects
  – BF from children (and OR of them)

• When a query happens
  – Checks own BF
  – If not found; check OR of children
  – If found check individual BFs for children
  – If not found (how come?) send to parent node

• Similar approach used for Geographic Routing to locate mobile devices
Multicast

- Usually for every group (sometimes for every source for every group) need to keep the list of interfaces to forward a packet on

- Alternatively; just use a Bloom filter of the groups that forward for every interface

- Bloom principle test?

- How to remove groups from interface?
Other Apps/Discussion

- Stochastic fair blue: identify flows that are bandwidth hogs
- IP Traceback: track down a packet’s origin even if it was spoofed
- Discussion
  - Really dynamic applications (e.g., traceback and flow monitoring); how do you forget?