Medium Access Control

3035/GZ01 Networked Systems
Kyle Jamieson
Lecture 3

Department of Computer Science
University College London
The link layer

- Link layer functionality:
  - Framing bits and packets
  - **Sharing: Medium access control**
  - Reliable delivery
  - Error detection and correction (previous lecture)
Sharing at the link layer

• Medium access control (MAC)
  – How should the shared medium be divided?
  – Who gets to talk on a shared medium, and when?

Shared wire (Ethernet)

Shared satellite uplink

Shared wireless
  (WiFi, packet radio)
Sharing: Goals

• **Efficiency**
  – High throughput (bits/second through channel)
  – High utilization (throughput/channel rate)

• **Fairness**: all hosts with data to send should get a roughly equal share of the medium over time
Physical limitation: finite speed of light

From here to:

- WiMAX Base station
- Paris
- San Francisco
- Hawaii
- Boston via Geosync Satellite

Distances and times:

- 3–30 m: 10–100 ns
- 300 m: 1 μs
- 3 km: 1.7 ms
- 8,645 km: 58 ms
- 12,000 km: 43 ms
- 73,000 km: 0.2 s
Similar MAC protocols, vastly different timescales

- WiFi: CSMA/CA
- Ethernet: CSMA/CD
- WiMAX Base station
- Packet radio: ALOHA
- Geosynchronous Satellite

- 3-30 m
- 3 km
- 73,000 km
- 0.2 s
- 10 μs
- 10-100 ns
Today

1. Tour of medium access protocols (TDMA, FDMA, CDMA)

2. Random access protocol: ALOHA
   - Slotted ALOHA
   - Unslotted ALOHA

3. Random access protocol: the Ethernet
MAC Protocols: a taxonomy

Three broad classes:

- **Channel Partitioning**
  - Divide channel into smaller “pieces”:
    - (time slots, frequency, code)
  - Allocate piece to node for exclusive use

- **“Taking turns”**
  - Nodes take turns, but nodes with more to send can take longer turns

- **Random Access**
  - Channel not divided, allow conflicts → bit errors
  - “Recover” from errors
Channel Partitioning: TDMA

TDMA: time division multiple access

- Access to channel in "rounds"
- Each station gets fixed length slot (packet time) in each round
- Unused slots go idle
- Example: six stations: only 1,3,4 have data to send

![Diagram showing a 6-slot frame with stations 1, 3, and 4 using slots, and the remaining slots being idle.](image)
Channel Partitioning: FDMA

FDMA: Frequency division multiple access
- Channel spectrum divided into frequency bands
- Each station assigned fixed frequency band
- Unused frequency bands are wasted
- Example: six station LAN: only nodes 1, 3, and 4 have data to send
Channel partitioning in WiFi

- WiFi uses **channel partitioning by frequency** to create channels on different frequencies (some overlapping)
- **Random access** within each channel
Code Division Multiple Access (CDMA)

• All users share the same frequency and time!

• Each user $i$ has own binary code $c_i$ of length $M$ to encode binary data

• Code rate = $M \times$ data rate
  – Encoded bit = data bit $\times$ code
  – Decoding: inner product of encoded signal and desired user’s code

• Allows multiple users to coexist and transmit simultaneously with minimal interference
CDMA encoding and decoding

Sender

Data bits

Code

data bits

d_1 = -1

d_0 = 1

Slot 1

Slot 0

Z_{i,m} = d_i c_m

channel output Z_{i,m}

Slot 1 channel output

Slot 0 channel output

Receiver

Received input

Code

D_i = \sum_{m=1}^{M} Z_{i,m} c_m

M

d_1 = -1

d_0 = 1

Slot 1 channel output

Slot 0 channel output
CDMA: Two-sender interaction

Sender 1

Sender 2

Receiver (listening to sender 1)

Sender 1 transmission

Sender 2 transmission

Receiver 1

\[ Z_{i,m}^1 = d_i^1 c_m^1 \]

\[ Z_{i,m}^2 = d_i^2 c_m^2 \]
CDMA: How to choose codes

- Sender S1: code $c^1$
- Sender S2: code $c^2$
- Orthogonality condition: \[ c^1 \cdot c^2 = \sum_{m=1}^{M} c^1_m c^2_m = 0 \]

- Receiving bit 0, S1:
  \[
  Z_0 = +c1 + c2 \\
  Z_0 \cdot c^1 = (c^1 + c^2) \cdot c^1 \\
  = c^1 \cdot c^1 + c^1 \cdot c^2 \\
  = 1 + 0 \\
  = +1
  \]

- Receiving bit 1, S1:
  \[
  Z_1 = -c1 + c2 \\
  Z_1 \cdot c^1 = (-c^1 + c^2) \cdot c^1 \\
  = -c^1 \cdot c^1 + c^1 \cdot c^2 \\
  = -1 + 0 \\
  = -1
  \]
Example of CDMA codes: Walsh codes

- $k^{th}$ user’s length-$n$ Walsh code is the $k^{th}$ row of an $n \times n$ Hadamard matrix $H(n)$
- Used in the CDMA downlink (base station to mobiles)

- **Definition:**
  
  $H(2) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$  
  
  $H(2^k) = \begin{bmatrix} H(2^{k-1}) & H(2^{k-1}) \\ H(2^{k-1}) & -H(2^{k-1}) \end{bmatrix}$

- $H(4) = \ldots$
“Taking turns”: Token ring MAC

- Special byte sequence (control “token”) passed around ring

1. Without token: relay bits as received
2. With token: discard incoming bits, send your data

- Concerns:
  - Token overhead
  - Latency
  - Single point of failure (token)
  - Complexity: token recovery protocol
Random access protocols

• When node has packet to send:
  – Transmit at full channel data rate $B$
  – No *a priori* coordination among nodes

• Two or more simultaneous transmissions: collision

• Random access MAC protocol specifies:
  – How to detect collisions
  – How to recover from collisions

• Examples of random access MAC protocols:
  – Slotted ALOHA
  – Unslotted (pure) ALOHA
  – CSMA/CD, CSMA/CA
Today

1. Tour of medium access protocols (TDMA, FDMA, CDMA)

2. Random access protocol: ALOHA
   - Slotted ALOHA
   - Unslotted ALOHA

3. Random access protocol: the Ethernet
ALOHAnet

- Context: Norm Abramson, 1970 at the University of Hawaii
  - Seven colleges on four islands
  - Keep terminals in touch with the mainframe
  - Telephone costs high: build a packet radio network
The Slotted-ALOHA MAC protocol

- Divide time into slots, transmit **only** in slot
  - Packets are all of length $L$ bits (add padding if not)
  - Transmit bit rate: $B$ bits/second
  - Each of $N$ nodes transmits with probability $p$ in each slot
  - Aggregate transmission rate therefore $\lambda = Np$ (packets/slot)

- Assumption: If one transmission in a slot, can receive; if $\geq 2$ in slot: no one can receive (collision)
Throughput of the Slotted-ALOHA MAC

Suppose $N$ nodes, each transmit with probability $p$ in each slot; what is the $p$ that maximizes their throughput?

- $\Pr[A \text{ node is successful in a slot}] = p(1-p)^{N-1}$
- $\Pr[\text{Success in each slot}] = Np(1-p)^{N-1}$
- Throughput = $\Pr[\text{Success}] \times \frac{L}{L/B} = B \times \Pr[\text{Success}]$

\[
\frac{d}{dp} \Pr(\text{success}) = N(1-p)^{N-1} - Np(N-1)(1-p)^{N-2} = 0
\]

\[
(1-p) - p(N-1) = 0
\]

\[
1 - p - pN + p = 0
\]

\[
\therefore p^* = \frac{1}{N}
\]

$\Pr(\text{success}) = \left(1 - \frac{1}{N}\right)^{N-1} \rightarrow \frac{1}{e} \approx 37\% \text{ (as } N \rightarrow \infty )$

\[
\lim_{x \to \infty} \left(1 + \frac{1}{x}\right)^x = e
\]

<table>
<thead>
<tr>
<th>$N$</th>
<th>$p^*$</th>
<th>$\Pr(\text{Success}) = \left(1 - \frac{1}{N}\right)^{N-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2} = 0.50$</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{1}{3}$</td>
<td>$\left(\frac{2}{3}\right)^2 \approx 0.44$</td>
</tr>
<tr>
<td>4</td>
<td>$\frac{1}{4}$</td>
<td>$\left(\frac{3}{4}\right)^3 \approx 0.42$</td>
</tr>
<tr>
<td>5</td>
<td>$\frac{1}{5}$</td>
<td>$\left(\frac{4}{5}\right)^4 \approx 0.41$</td>
</tr>
</tbody>
</table>

Throughput best when each node sends with probability $1/N$, decreases with increasing numbers of nodes $N$
Throughput of the Slotted-ALOHA MAC

Suppose $N$ nodes, each transmit with probability $p$ in each slot. What is the throughput as a function of aggregate rate $\lambda = Np$?

- $\Pr[\text{A node is successful in a slot}] = p(1-p)^{N-1}$
- $\Pr[\text{Success in a slot}] = Np(1-p)^{N-1}$

$$\Pr(\text{success}) = \lambda \left(1 - \frac{\lambda}{N}\right)^{N-1}$$

$$\lim_{N \to \infty} \lambda \left(1 - \frac{\lambda}{N}\right)^{N-1} = \lambda e^{-\lambda}$$

Throughput best when each node sends with probability $1/N$, decreases with aggregate rate $\lambda < 1$ or $\lambda > 1$.
The Unslotted-ALOHA MAC protocol

- $N$ senders in total
- Just transmit a packet without delay

Assume: $\Pr(\text{Packet send in } [t, t + \Delta t])$ is $\lambda \Delta t$
  - This is called a Poisson process
  - $\lambda$ is aggregate rate from all $N$ senders
  - Individual rate $\lambda/N$ for each sender
Throughput of the Unslotted-ALOHA MAC

• Focus on node \( i \)'s frame: vulnerable time = 2

\[
\Pr(\text{no send from a node}) = 1 - \frac{2\lambda}{N}
\]

\[
\Pr(\text{no send in time 2}) = \left(1 - \frac{2\lambda}{N}\right)^{N-1}
\]

\[
\lim_{N \to \infty} \left(1 - \frac{2\lambda}{N}\right)^{N-1} \to e^{-2\lambda}
\]

• Throughput  \( = \lambda \times \Pr(\text{no other transmission in 2}) \)
  \( = \lambda e^{-2\lambda} \)
ALOHA throughput comparison: slotted versus unslotted

Just by forcing nodes to transmit on slot boundaries, we double peak throughput!

Unslotted ALOHA: $\lambda e^{-2\lambda}$

Slotted ALOHA: $\lambda e^{-\lambda}$

$1/e \approx 36\%$

$1/2e \approx 18\%$
Summary

• ALOHA throughput
  – Slotted: maximum 36% efficiency
  – Unslotted: maximum 18% efficiency

• Later in the term, we’ll see how MACAW and 802.11 increase efficiency with by *listening* before transmitting
  – This is called *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA)
Today

1. Tour of medium access protocols (TDMA, FDMA, CDMA)
2. Random access protocol: ALOHA
   - Slotted ALOHA
   - Unslotted ALOHA
3. Random access protocol: the Ethernet
The Ethernet: Context

- Bob Metcalfe, PhD student at Harvard in early 1970s
  - Working on protocols for the ARPAnet
- Xerox Palo Alto Research Center (PARC), 1973
  - Computer on every desk:
    - (Alto Workstation)
    - Butler Lampson *ea.*
  - Needed a way to network the Altos
  - ALOHA packet radio

- Metcalfe later founds *3Com*, acquired by HP in April ’10 for $2.7 bn
The Ethernet: Physical design

- Coaxial cable, propagation delay $\tau$
  - Propagation speed: $\frac{3}{5} \times$ speed of light

- Experimental Ethernet
  - Data rate: $B = 3$ Mbits/s
  - Maximum length: 1000 m
  \[
  \tau = \frac{10^3 \text{ m}}{\frac{3}{5} \left(3 \times 10^8 \text{ m/s}\right)} \approx 5 \mu s
  \]
Building the link: Encoding bits

• **Goal:** Move bits from one place to another
  – Sender and receiver have independent clocks
  – No separate “clock signal” sent on the Ethernet

• **Problem:** Agree on clock tick period

<table>
<thead>
<tr>
<th>Time</th>
<th>Sender clock</th>
<th>Receiver clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 0 0 0 1 0 1 0 1</td>
<td>&quot;1&quot; &quot;0&quot; &quot;0&quot; &quot;0&quot; &quot;1&quot; &quot;?&quot; &quot;?&quot; &quot;?&quot; &quot;?&quot;</td>
</tr>
</tbody>
</table>

• **Problem:** Agree on clock tick alignment (phase)

<table>
<thead>
<tr>
<th>Time</th>
<th>Sender clock</th>
<th>Receiver clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 0 0 0 1 0 1 0 1</td>
<td>&quot;?&quot; &quot;?&quot; &quot;?&quot; &quot;?&quot; &quot;?&quot; &quot;?&quot; &quot;?&quot; &quot;?&quot; &quot;?&quot;</td>
</tr>
</tbody>
</table>
How to encode bits?

• Binary encoding is called *Nonreturn to Zero* (NRZ)
  – Drawback: *Baseline wander* when receiver uses average of signal to distinguish between high and low
  – Drawback: Clock recovery in the presence of long runs of 0s or 1s in the data

• *Nonreturn to Zero Inverted* (NRZI): Transition for a “1”, no transition for a “0”
Manchester (phase) encoding

- Manchester encoding:
  - Exclusive-OR of the NRZ signal and the clock signal
  - “0” is a low-to-high transition; “1” is a high-to-low transition
- Receiver looks for transitions to synchronize
- Transition guaranteed on every bit
- *Phase encoding* in the experimental Ethernet [Metcalf et al.]
- Drawback: Halves data rate
4B/5B encoding

- **Properties**
  - No code has more than one leading zero
  - No code has more than two trailing zeros
  - When sent back-to-back, no pair of 5-bit codes contains more than three consecutive zeros

<table>
<thead>
<tr>
<th>4-bit data</th>
<th>5-bit code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>11110</td>
</tr>
<tr>
<td>0001</td>
<td>01001</td>
</tr>
<tr>
<td>0010</td>
<td>10100</td>
</tr>
<tr>
<td>0011</td>
<td>10101</td>
</tr>
<tr>
<td>0100</td>
<td>01010</td>
</tr>
<tr>
<td>0101</td>
<td>01011</td>
</tr>
<tr>
<td>0110</td>
<td>01110</td>
</tr>
<tr>
<td>0111</td>
<td>01111</td>
</tr>
<tr>
<td>1000</td>
<td>10010</td>
</tr>
<tr>
<td>1001</td>
<td>10011</td>
</tr>
<tr>
<td>1010</td>
<td>10110</td>
</tr>
<tr>
<td>1011</td>
<td>10111</td>
</tr>
<tr>
<td>1100</td>
<td>11010</td>
</tr>
<tr>
<td>1101</td>
<td>11011</td>
</tr>
<tr>
<td>1110</td>
<td>11100</td>
</tr>
<tr>
<td>1111</td>
<td>11101</td>
</tr>
</tbody>
</table>

- **Encoding process:**
  1. Encode data using lookup table
  2. Send coded bits with NRZI

- **Used in later Ethernet standards**
Collisions

• Packet of $N$ bits: $N/B$ seconds on the wire
• From the perspective of a receiver (B):
  – Overlapping packets at B means signals sum
  – Not time-synchronized: result is bit errors at B
  – No fate-sharing among receivers: C receives okay in this example
The Ethernet: Packet format

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Destination</th>
<th>Source</th>
<th>Type</th>
<th>Data</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 bits</td>
<td>8 bits</td>
<td>16 bits</td>
<td></td>
</tr>
</tbody>
</table>

- **Type** field added to commercial Ethernet
  - Multiplex higher layer protocols

- **CRC (Cyclic Redundancy Check)** protects against errors on the Ether
  - Does not guard against errors introduced by the tap: rely on higher-layer checksums

- Destination address allows filtering at the link layer
Goals

• “Controlled chaos:” anyone can send at almost any time

• Subject to rules for minimizing collisions:
  – Don’t send when someone else is sending
  – If you start sending then discover someone else is sending, you both quickly abort
Problem: Who gets to transmit?

Solution: Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

1. Begin the transmission procedure at any time

2. **Carrier sensing**: never transmit a frame if you sense that another station is transmitting

3. **Collision detection**: while sending, immediately abort your transmission if you detect another station transmitting
Carrier sensing

- Mechanism: measure voltage on the wire
- Binary encoding: voltage depends on the data
- Manchester coding: constant average voltage
Collision detection

- Paper isn’t clear on this point (authors did have a patent in the filing process)
- Mechanism: monitor average voltage on cable
  - Manchester encoding means your transmission will have a predictable average voltage $V_0$; others will increase $V_0$
  - Abort transmission immediately if $V_{\text{measured}} > V_0$
When does a collision happen?

- Suppose Station A begins transmitting at time 0
- Assume that the packet lasts much longer than $\tau$
- All stations sense transmission and **defer** by time $\tau$
  - Don’t begin any new transmissions
- At time $\tau$, will a packet be collision-free?
  
  Only if no other transmissions began before time $\tau$
How long does a collision take to detect?

- Suppose Station A begins transmitting at time 0
- $\tau$ seconds after Z starts, A hears Z’s transmission
- When does A know whether its packet collided or not? At time $2\tau$
How big must packets be for collisions to be detectable?

Transmit rate $B$ bits/second

If packets take time $2\tau$, A will still be transmitting when Z’s packet arrives at A, so A will detect collision

So minimum packet size > $2\tau B$ bits

Experimental Ethernet:

\[ \tau = 5 \, \mu s, \quad B = 3 \, \text{Mbits/s} \rightarrow 2\tau B = 30 \, \text{bits} \]

Why doesn’t Metcalfe & Boggs paper mention this?
Commercial Ethernet

- **Commercial Ethernet**
  - Data rate $B = 10 \text{ Mbits/s}$
  - Maximum length: 500 m per segment with up to two *repeaters* (*hubs*)

- Repeater receives bits, relays them onto wire
  - $\tau = 20 \mu s$ worst case $\rightarrow 2\tau B = 400 \text{ bits} = 50 \text{ bytes}$
  - Could send complete packet; not see collision
Resolving collisions

• Upon abort (carrier detect), enter **backoff state**

• **Key idea:** two or more stations all wait a random time before transmitting again
  – Applicable in many other kinds of networked systems
  – How to pick random time?
  – How many stations are colliding?

• Backoff time is **slotted** and **random**
Slotted Ethernet backoff

- Each station waits $K$ slots, choosing $K$ uniformly at random from $CW = [0, 2^m-1]$
  - Reset $m \leftarrow 1$ for each new packet; cap $m$ at 10
  - First retransmit ($m = 1$): pick from $[0, 1]$
  - Second retransmit ($m = 2$): pick from $[0, 1, 2, 3]$

- Binary exponential backoff (BEB): double $CW$ on failure
- Goal: Engineer such that nodes picking different slots carrier sense (CS) and defer, thus don’t collide
SloYed Ethernet backoff: picking slot time

• **Slotted backoff:** pick multiples of a slot time
  – Slots are timed off the tail-end of the last packet
  – Stations’ clocks differ by at most $\tau$

  ![Diagram showing backoff timers](image)

  – Retransmissions scheduled to begin $> \tau$ apart won’t collide.

  **Therefore, pick slot time = $2\tau$**
Enforcing consensus on collisions

- Ethernet as described so far: collision at B and C but not at A or Z
- Lack of consensus results in differing backoff windows
- Metcalfe & Boggs: When a station detects collision, it “momentarily jams the Ether to [ensure] that all other participants in the collision will detect interference and, because of deference, will be forced to abort.”
- Result: All stations agree there was a collision, double backoff windows, backoff, and retransmit
- Jamming signal of length $2\tau$ suffices to enforce consensus
Ethernet performance

Divide time into variable-sized contention intervals,
Fixed size transmission intervals (duration $t_{packet}$)

Efficiency: $\frac{t_{packet}}{t_{packet} + (2\tau)W}$

Number of slots to acquire the Ether
Ethernet performance: Acquisition

• Suppose there are $Q$ stations waiting to send
• **Assume**: stations know $Q$ and transmit with probability $1/Q$ (BEB approximates this)

• Calculate: probability one station acquires without collision

$$p_{\text{acquire}} = Q \left( \frac{1}{Q} \right) \left( 1 - \frac{1}{Q} \right)^{Q-1}$$

$$= \left( 1 - \frac{1}{Q} \right)^{Q-1}$$

$$\approx \frac{1}{e}$$

$$\approx 37\%$$
**Ethernet performance: Waiting time**

$W$: number of slots in a contention window before acquisition of the Ether

$$\Pr(W = i) = p_a (1 - p_a)^i$$

$$E[W] = \sum_{i=0}^{\infty} i \cdot p_a (1 - p_a)^i$$

- Probability of no wait: $p_{\text{acquire}}$
- Probability wait one slot: $(1 - p_{\text{acquire}}) p_{\text{acquire}}$
- Probability wait two slots: $(1 - p_{\text{acquire}})^2 p_{\text{acquire}}$
- $E[\text{slots to wait}] = W = (1 - p_{\text{acquire}}) / p_{\text{acquire}}$

$$= p_a \sum_{i=0}^{\infty} i \cdot (1 - p_a)^i$$

$$= p_a \sum_{i=0}^{\infty} \frac{d}{dp_a} \left[ -(1 - p_a)^i \right] (1 - p_a)$$

$$= p_a \frac{d}{dp_a} \left[ \frac{-1}{p_a} \right] (1 - p_a)$$

$$= \frac{1 - p_a}{p_a}$$

$$= e - 1$$
Forwarding and Addressing in the Link Layer
Pre-Reading: P & D Sections 3.1-3.4; excerpt from Interconnections, Perlman

NEXT TIME