Ethernet

- Overview
- Physical Layer
- MAC
- Bridged Ethernet
- VLAN
- Link Aggregation
- XON/XOFF
- 802.11
- Summary
Overview

- Typical Setup
- Names
- Operations
- Perspective
Typical Setup
Names

- Structure

  \[\text{rate}\][\text{modulation}][\text{media or distance}]\]

  - 10Base5 (10Mbps, baseband, coax, 500m)
  - 10Base-T (10Mbps, baseband, twisted pair)
  - 100Base-TX (100Mbps, baseband, 2 pair)
  - 100Base-FX (100Mbps, baseband, fiber)
  - 1000Base-CX for two pairs balanced copper cabling
  - 1000Base-LX for long wavelength optical transmission
  - 1000Base-SX for short wavelength optical transmission.

- Wireless:

  - Wi-Fi = 802.11
  - Versions: a, b, g
Operations

- Hub: Single Collision Domain

MAC Protocol: Wait until silent (carrier sense)
  - Transmit
  - If collision, wait random time & repeat

CSMA/CD
Operations

- Switch: No Collisions

Multiple transmissions are possible
Switch stores packets that wait for same output
Perspective

- Ethernet is wildly successful, partly due to low cost (compare with FDDI or Token Ring-- see text book)

- Some issues:
  - nondeterministic service
  - no priorities
  - min frame size may be large
Physical Layer

**UTP**
unshielded twisted pair
up to 110m

**Fiber**
100Mbps: 2000m
Gbps: 220m, 500m, 5000m

**Wireless**
2.4GHz DSSS: 1Mbps, 2Mbps, 5.5Mbps, and 11Mbps
25m - 200m
MAC: Media Access Control

- Frame
- Multiple Access
**Frame**

<table>
<thead>
<tr>
<th>PRE</th>
<th>DA</th>
<th>SA</th>
<th>T/L</th>
<th>USER DATA</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>46-1500</td>
<td>4</td>
</tr>
</tbody>
</table>

PREamble: alternating 1/0 combination producing 10Mhz square wave [@ 10Mbps] for 5.6 μsec; used for receiver synchronization; ends with 11 to indicate start of frame.

DA, SA: Destination and Source MAC addresses (All 1’s means broadcast)

T/L: Length if <= 1500 → followed by 802.3 frame Type if > 1500.
   Examples: 2,048: IP (length in IP header) 2,054: ARP (28 byes)
Multiple Access

- High-Level View
- Multiple Access Protocols
- Random Access Protocols
- Slotted ALOHA
- CSMA
- CSMA/CD
- Ethernet Multiple Access
High-Level View

- Goal: share a communication medium among multiple hosts connected to it
- Problem: arbitrate between connected hosts
- Solution goals:
  - High resource utilization
  - Avoid starvation
  - Simplicity (non-decentralized algorithms)
Medium Access Protocols

- Channel partitioning
  - Divide channel into smaller “pieces” (e.g., time slots, frequency)
  - Allocate a piece to node for exclusive use

- Random access
  - Allow collisions
  - “recover” from collisions

- “Taking-turns”
  - Tightly coordinate shared access to avoid collisions
Random Access protocols

- When node has packet to send
  - Transmit at full channel data rate R.
  - No *a priori* coordination among nodes
- Two or more transmitting nodes -> “collision”,
- Random access MAC protocol specifies:
  - How to detect collisions
  - How to recover from collisions
- Examples of random access MAC protocols:
  - Slotted ALOHA
  - CSMA and CSMA/CD
Slotted Aloha

- Time is divided into equal size slots (= packet transmission time)
- Node with new arriving pkt: transmit at beginning of next slot

Success (S), Collision (C), Empty (E) slots
What is the maximum fraction of successful transmissions?

Suppose N stations have packets to send

- Each transmits (independently) in slot with probability \( p \)
- Prob. successful transmission
  - by a particular node: \( p(1 - p)^{(N-1)} \)
  - by any of N nodes:
    \[
    S = \text{Prob (only one transmits)} = Np(1 - p)^{(N-1)} \approx 1/e = 0.37
    \]

Note: \((1 - a/N)^N \approx e^{-a}\) for \(N \gg 1\) if \(p = 1/N\) (optimal value)
CSMA: Carrier Sense Multiple Access

- CS (Carrier Sense) means that each node can distinguish between an idle and a busy link

- Sender operations:
  - If channel sensed idle: transmit entire packet
  - If channel sensed busy, defer transmission
    - Persistent CSMA: retry immediately with probability $p$ when channel becomes idle
    - Non-persistent CSMA: retry after a random time interval
CSMA: collisions

Collisions can occur:
propagation delay means two nodes may not hear each other’s transmission

Collision:
entire packet transmission time wasted

Note:
role of distance and propagation delay in determining collision prob.
CSMA/CD (collision detection)

- Overview
- Timing
- Ethernet
Overview

- Collisions *detected* within short time
- Colliding transmissions aborted, reducing channel wastage
- Easy in wired LANs: measure signal strengths, compare transmitted, received signals
- Difficult in wireless LANs
Timing

[Diagram showing time and space axes, with labels for time intervals and collision detection/abandon time.]
Ethernet

- Overview
- Collision Detection
- Minimum Frame Size
- Maximum Frame Size
- Operations
- Efficiency
- Addressing
- Fast Ethernet
- Gigabit Ethernet
Overview

- Will discuss “classical” Ethernet primarily
- Single segments up to 500m; with up to 4 repeaters gives 2500m max length
- Baseband signals broadcast, Manchester encoding, 32-bit CRC for error detection
- Max 100 stations/segment, 1024 stations/Ethernet
Collision Detection

- CD circuit operates by looking for voltage exceeding a transmitted voltage
- Want to ensure that a station does not complete transmission prior to 1st bit arriving at farthest-away station
- Time to CD can thus take up to $2\times \text{max prop. delay}$
Minimum Frame Size

- Speed of light is about 4\(\mu\)s/km in copper
- So, max Ethernet signal prop time is about 10\(\mu\)sec, or 20\(\mu\)sec RTT
- With repeaters, etc. 802.3 requires 51\(\mu\)sec, corresponding to 512 bit-times
- Thus, minimum frame size is 512 bits (64 bytes); also called slot time
Maximum Frame Size

- 1500 byte limitation on maximum frame transmission size
- We will call this the *MTU*
- limits maximum buffers at receiver
- allows for other stations to send
  - also requires 96 bit Inter-Packet-Gap (IPG)
Operations

- When ready & line idle, await IPG (96 bit times) and send while listening (CD)
- If CD true, send max 48-bit jamming sequence and do exponential backoff
- Jamming sequence used to inform all stations that a collision has occurred
Operations

- For retransmission N (1<=N<=10)
  - choose k at random on U(0..2^N-1)
  - wait k * (51.2µsec) to retransmit
  - send on idle; repeat on another collision
  - for (11<=N<=15), use U(0..1023)
  - if N = 16, drop frame

- Longer wait implies lower priority (strategy is not “fair”)
Operations: Capture Effect

- Given two stations A & B, unfair strategy can cause A to continue to “win”
- Assume A & B always ready to send:
  - if busy, both wait, send and collide
  - suppose A wins, B backs off
  - next time, B’s chances of winning are halved
Efficiency

Typical Sequence of Events:

- Start Transmitting
- Wait Random Time
- Collision
- Successful Transmission

Average = 5T

T = max. prop. time between 2 nodes

Average = L/R seconds

L = average packet length

Efficiency = \frac{L}{R} = \frac{1}{1 + 5a}

a = \frac{T}{L/R} = \frac{RT}{L}

TOC – Ethernet – MAC – Multiple Access - Ethernet – Efficiency
Efficiency

- Efficiency impacts what happens during simultaneous transmission:
  - a small $\rightarrow$ early collision detection
    $\rightarrow$ Efficient
  - a large $\rightarrow$ late detection
    $\rightarrow$ Inefficient

Example 1: 10Mbps, 1000m
$\Rightarrow T = (1\text{km})(4\mu\text{s/km}) = 4\mu\text{s}$; $RT = 400$ bits
$L = 4000$ bits, say
$5a = 2000/4000 = 0.5 \Rightarrow$ efficiency $= 66$

Example 2: 1Gbps, 200m
$\Rightarrow T = (0.2\text{km})(4\mu\text{s/km}) = 0.8\mu\text{s}$; $RT = 800$ bits
$L = 4000$ bits; $5a = 4000/4000 = 1 \Rightarrow$ efficiency $= 50$

$a = RT/L$
$\text{eff} = 1/(1 + 5a)$
Efficiency - Analysis

Model:

Slot = 2T

N stations compete by transmitting with probability p, independently
If success => transmit L bits
If failure (idle or collision), try next slot

P(success) = P(exactly 1 out of N transmits) = Np(1 – p)^{N-1}

Indeed: N possibilities of station that transmits
P(one given station transmits, others do not) = p(1 – p)^{N-1}
Efficiency - Analysis

\[ P(\text{success}) = Np(1 - p)^{N-1} \]

Assume backoff algorithm results in best \( p = 1/N \)

\[ \Rightarrow P(\text{success}) \approx \frac{1}{e} = 0.36 \]

Average time until success:

\[ A = 0.36 \times 0 + 0.64 \times (2T + A) \]

\[ \Rightarrow A = \frac{1.28T}{0.36} = 3.5T \]

In practice, backoff not quite optimal \( \Rightarrow 5T \)
Addressing

- 48 bit Ethernet/MAC/Hardware Addresses
- Prefix assigned per-vendor by IEEE
- Unique per-adapter, burned in ID PROM
- Multicast & Broadcast (all 1’s) addresses
- Many adapters support *promiscuous* mode
Addressing - Multicast

- Each vendor assignment supports $2^{24}$ individual and group (multicast) addresses.
- Each adapter supports multiple group "subscriptions".
  - Usually implemented as hash table.
  - Thus, software may have to filter at higher layer.
Fast Ethernet

“Fast Ethernet” (1995) adds:
- 10x speed increase (100m max cable length retains min 64 byte frames)
- replace Manchester with 4B/5B (from FDDI)
- full-duplex operation using switches
- speed & duplex auto-negotiation
Gigabit Ethernet
IEEE 802.3\{z,ab\} 1000 Mb/s

- "Gigabit Ethernet" (1998,9) adds:
  - 100x speed increase
  - carrier extension (invisible padding...)
  - packet bursting
Bridged Ethernets

- Objectives
- No-Frills Bridge
- Learning Bridge
- Spanning Tree Algorithm
- Bridge Limitations
Objectives

- Bridges interconnect network segments at the *link* layer (layer 2)
- Handle any layer 3 protocol (incl. non-routable ones); some can interconnect different media
- Mostly for LANs, also used in WANs (2 “half bridges” on ends of pt-to-pt links)
Objectives – Extending LANs

- Extending (interconnecting) multiple LANs. Appears as single LAN to layer 3.
- Essentially accepts and forwards all frames
- Benefits:
  - extend number of stations
  - extend size
  - limit interfering traffic
“No-Frills” Bridge

- Interconnect 2 or more LAN segments
- Listens in promiscuous mode, buffers packets and transmits them on other interfaces when able
- On average, still cannot exceed link bandwidth
  - bridge copies all traffic
  - small bursts accommodated in buffers
Learning Bridge

- Bridges “learn” which interfaces reach which end stations
  - could do this “by hand”, but a hassle
  - best if this happens transparently
- Learn by watching source addresses in frames
  - senders usually use their own addresses
  - (note that bridges don’t!)
Listen promiscuously for all traffic

Store (src addr, port) tuple in “station cache” for each new sender observed

For each received frame:
  - try to match frame dest to cache src entry
  - not there->send on all interfaces except rcv
  - is there->send on indicated, or filter if same as rcv interface

Age cache entries
Learning Bridge – Example 1

TOC – Ethernet – Bridged - Learning
Learning Bridge – Example 2

Diagram showing network connections with labels A, B, C, D, E, F, and numbers 1, 2, 3.
Spanning Tree Algorithm

- Loops hurt!
- Spanning Tree
- Algorithm
- Example
- Topology Change
Loops Hurt

- With redundant paths, bridges can loop traffic
  - can happen forever (example)
  - with more than 2, can cascade

- Cascade
  - each bridge with N interfaces may produce up to N - 2 new copies!
Spanning Tree

- Consider LAN a graph $G = (E, V)$, with LANs as vertices, and bridges as edges
- Spanning Tree:
  - A spanning tree of an undirected, connected graph $G$ is a subgraph which is both a tree and contains all vertices in $G$
  - Thus, the ST will throw out some edges and be cycle-free
- Purpose will be to provide a single path to reach each network
- Generally, graphs have many STs
- Must be a distributed algorithm
- Can result in some bridges not forwarding at all!
Algorithm

- Each bridge will decide over which ports it will forward frames
  - bridges have unique addresses *per port*
  - ports are also numbered by each bridge
  - bridges have a single unique identifier (e.g. the lowest address)

- Ultimately
  - Root of tree = bridge with smallest ID
  - Tree = shortest paths to root
  - Resolve ties in favor of link to bridge with smallest ID

- Steps
  - Send [ my ID | current root ID | my distance to current root ]
  - Update when receive smaller root ID
Example

Format: [my ID | current root ID | distance to current root]

- 4 → [2|1|1]
- 3 → [1|1|0]
- 1 → [3|3|0]
- 5 → [3|1|2]
- 2 → [5|3|1]
- 6 → [6|1|1]
Topology Change

- Want to inform all bridges, but without having traffic scale as # of bridges

- Operation
  - bridges noticing change send message on root port toward root
  - root config messages subsequently contain “topology changed” flag
Limitations

- Scale: not very realistic to interconnect more than 10’s of LANs
- Heterogeneity: really works best for homogeneous systems
- All broadcasts and multicasts are flooded
Each port belongs to a set of VLANs and transmits only packets that belong to these VLANs
Link Aggregation

Links are combined. Hello protocol determines which links are working and transmits across working links.
XON/XOFF is a backpressure scheme designed to prevent nodes from saturating links. Unfortunately, in its present version, the scheme is not very effective – it stops too many flows.
Wireless (802.11)

- Overview
- Physical
- MAC
Overview

- Designed for use in limited geographical area (i.e., couple of hundreds of meters)
- Designed for three physical media (run at either 1Mbps or 2 Mbps)
  - Two based on spread spectrum radio
  - One based on diffused infrared
Physical

- **Frequency hoping (OFDM)**
  - Transmit the signal over multiple frequencies
  - The sequence of frequencies is pseudo-random, i.e., both sender and receiver use the same algorithm to generate their sequences

- **Direct sequence (DSSS)**
  - Represent each bit by multiple (e.g., $n$) bits in a frame; XOR signal with a pseudo-random generated sequence with a frequency $n$ times higher

- **Infrared signal**
  - Sender and receiver do not need a clear line of sight
  - Limited range; order of meters
Physical

- **802.11b**
  - DSSS: 1, 2, 5.5, or 11 Mbps (depending on SNR)
  - 2.4GHz
  - Each channel occupies 22MHz
  - There are 11 channels, but only 3 non-overlapping ones (1, 6, 11)
  - Max. power = 1W

- **802.11a**
  - OFDM: 6, 9, 12, 18, 24, 36, 48, 54 Mbps (last 3 are optional)
  - 5GHz
  - 12 channels
  - Max. power = 40mW, 200mW, 800mW (depending on freq.)
Physical

- **802.11g**
  - OFDM: 6, 12, 24, 18, 36, 48, 54 Mbps (last 4 are optional)
  - 2.4GHz
  - Backward compatible with 802.11b
  - 3 non-overlapping channels
MAC: The Problems

- Reachability is not transitive: if A can reach C, and C can reach D, it doesn’t necessary mean that A can reach D

- Hidden nodes: A and C send a packet to B; neither A nor C will detect the collision!

- Exposed node: B sends a packet to A; C hears this and decides not to send a packet to D (despite the fact that this will not cause interference)!
MAC: RTS/CTS

- Before every data transmission
  - Sender sends a Request to Send (RTS) frame containing the length of the transmission
  - Receiver responds with a Clear to Send (CTS) frame
  - Sender sends data
  - Receiver sends an ACK; now another sender can send data

- When the sender doesn’t get a CTS back, it assumes collision
MAC: CSMA/CA

- CA instead of CD: cannot listen while transmitting
- NAV: Network Allocation Vector: time until channel will become available – transmitted in every frame
- Distributed Coordinated Function: Before transmitting
  - Check if channel is busy, or
  - In an interval between frames [as determined from NAV], or
  - In an interval reserved for retransmission of an erroneous frame [as determined from previous transmission].
  - If one of the above, exponential backoff; else transmit.
- Centralized Controlled Access Mechanism: A point coordinator (in access point) coordinates transmissions.
  - Stations request that the PC puts them in polling list
  - Periodically, PC polls stations for traffic.
  - Without hidden terminal; with hidden terminal
Summary

- Arbitrate between multiple hosts sharing a common communication media
- Wired solution: Ethernet (use CSMA/CD protocol)
  - Detect collisions
  - Backoff exponentially on collision
- Wireless solution: 802.11
  - Use MACA protocol
  - Cannot detect collisions; try to avoid them