Networks

- **Computer network**
  - Two or more nodes connected through a physical medium (link).

- **Physical medium**
  - Guided media
    - Signal propagates in solid media: copper, fiber
    - Examples: Twisted Pair (CAT 3, CAT 5, CAT 6), multi-mode/single mode fiber
  - Unguided media
    - Signal propagates freely: radio
    - 802.11 WLAN, 802.15.1 Bluetooth, 802.15.4 ZigBee
Circuit Switching
- Dedicated resources
- Resources reserved like for a “call”
- Circuit-like guaranteed performance
- Call setup required
- Links not used are idle

Packet Switching
- Source breaks message into smaller chunks: “packets”
- “Store-and-forward” at each switch

Circuit Switching vs. Packet Switching
- Packet Switching is good for bursty data
  - Resource sharing
  - No call setup
  - Congestion can lead to packet loss and delay
  - Protocols needed for reliable transfer
  - Congestion control

Types of Networks by Scopes
- PAN
  - Personal Area Network
  - Centered around personal devices, short range
- LAN
  - Local Area Network
  - Home, School, Lab, Office Building
- WAN
  - Wide Area Network
  - Network across different physical locations
- Internet
**ISO/OSI Reference Model**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>(7) App.</td>
<td>Application</td>
<td>Logical Ports, RPC</td>
</tr>
<tr>
<td>(6) Pre.</td>
<td>Presentation</td>
<td>TCP, UDP</td>
</tr>
<tr>
<td>(5) Sess.</td>
<td>Session</td>
<td>IP, Routers</td>
</tr>
<tr>
<td>(4) Tran.</td>
<td>Transport</td>
<td>TCP, UDP</td>
</tr>
<tr>
<td>(3) Net.</td>
<td>Network</td>
<td>IP, Routers</td>
</tr>
<tr>
<td>(2) DL</td>
<td>Data Link</td>
<td>Switches, Bridges</td>
</tr>
<tr>
<td>(1) Phys.</td>
<td>Physical</td>
<td>Cables, Hubs, etc.</td>
</tr>
</tbody>
</table>

**Multiple Access Protocol**

- Single shared communication channel, two or more nodes want to transmit
  - Potential interference, only one node can send at a time

- **Multiple Access Protocol (MAC)**
  - Distributed algorithm to determine how hosts share the channel
  - Must use the channel itself...
  - Possibilities:
    - Partitioning
    - Taking turns
    - Random access

**MAC Address**

- Physical address of the NIC
  - 48bit for most LANs, usually fixed in adapter ROM
  - Administered by IEEE
  - Manufacturers buy portions of the MAC address space

- Flat addresses
  - Have no meaning (except for the vendor prefix)
  - Portable

**Ethernet**

- Connectionless, unreliable, random access
  - Carrier sense: don’t send if somebody is already sending
  - Collision detection: abort send if detect somebody else sending
  - Wait a random time before retransmitting

- **Preamble**
  - 7 bytes with 10101010
  - 1 byte with 10101011

- **Destination Address**
  - All adapters listen, drop frame if address does not match

- **Type**
  - Indicates higher layer protocol

- **CRC**
  - Drop frame if error is detected
Thin coaxial cable in a bus topology

- **10Base2:**
  - 10 Mbps
  - 2: maximal cable length 200m
  - Repeaters are used to increase diameter of the network

- **10,100,1000BaseT, Hubs**
  - Twisted pair cable
  - Earlier forms used hubs as physical-layer repeaters
  - Star Topology
  - Hubs simply broadcast all signals to all ports

- **Bridges**
  - Break down LAN into multiple segments
  - Multiple collision domains, traffic isolation
  - Bridge filters packages and selectively forward between LAN segments
  - Bridge table filled by bridge learning sender locations

- **Switches**
  - Works like a multi-interface bridge
  - Can switch between multiple pairs of ports simultaneously
  - Frames are no longer broadcast
  - Not all at least, there is a MAC broadcast address
Internet Protocol

- Clearly, switches and bridges do not scale for large networks
- How can we build “global” networks?
  - Network of networks
- Problem is the flat address space
- What we need instead is a hierarchical address space
- Solution: Internet Protocol
- IP Addresses

IP Address

- 32bit for host interface
  - Interface: connection between host and physical link
  - Hosts can have multiple interfaces
- Typically represented in dotted notation:
  - 9.3.45.239
  - 0000 1001 0000 0011 0010 1101 1110 1111
- Hierarchical: network part and host part
  - Subnet mask: 255.255.255.0
  - Network part: 9.3.45.0
  - Host part: 239

Class-full Addressing

- Early players got large “real estate”
CIDR

- Classless Inter Domain Routing
- Network portion of address of arbitrary length

Equivalence
- Class A: /8
- Class B: /16
- Class C: /24

IP Addressing

- Route aggregation

Organization A
200.23.18.0/23

Organization B
200.23.20.0/23

Organization F
200.23.30.0/23

ISP 1
200.23.16.0/20

ISP 2
199.31.0.0/16

IP Datagrams

IP Datagrams
- Best effort, unreliable
- Packets get fragmented and reassembled
  - MTU = maximum transmission unit

Packet Format

<table>
<thead>
<tr>
<th>Version</th>
<th>Hdr length</th>
<th>Flags</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Upper Layer Protocol</th>
<th>Header Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA</th>
<th>Options (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Routing**

- Forwarding packets through the network

- Datagram remains unchanged, address fields are of interest

```
223.1.1
223.1.2
223.1.3
223.1.1.1
223.1.1.2
223.1.1.3
223.1.1.4
223.1.2.9
223.1.2.1
223.1.3.2
223.1.3.1
223.1.3.27
223.1.3.27
```

**Routing Table in A**

<table>
<thead>
<tr>
<th>Dest. Net.</th>
<th>next router</th>
<th>#hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>223.1.1</td>
<td>223.1.1.4</td>
<td>1</td>
</tr>
<tr>
<td>223.1.2</td>
<td>223.1.1.4</td>
<td>2</td>
</tr>
<tr>
<td>223.1.3</td>
<td>223.1.1.4</td>
<td>2</td>
</tr>
</tbody>
</table>

**DHCP**

- Host/router interfaces can be statically configured an IP address

- Alternatively, they can dynamically be assigned one by DHCP

  - Lease-based assignment
    - host broadcasts “DHCP discover” message
    - DHCP server responds with “DHCP offer” message
    - host requests IP address: “DHCP request” message
    - DHCP server sends address: “DHCP ack” message

**ARP**

- Think of an endpoint connected to an Ethernet that wants to send an IP datagram...
- How can a network node map IP addresses to MAC addresses?
- Address Resolution Protocol (ARP)
  - Send an Ethernet broadcast message
  - Matching host responds with its IP address
  - Requesting node typically caches this information

- Variation
  - Nodes announce themselves through a “gratuitous” ARP message"
Naming
- Do you typically retrieve web pages by their IP addresses?
- Naming: mapping from (possibly) human readable names to IP addresses
- Implemented through Domain Name System (DNS)
  - Distributed database of names
  - Nodes contact a DNS server to resolve names to IP addresses
  - Implemented as an application-level protocol
- Local name servers
  - Each ISP, company has a local default name server
  - Query first goes to this server, then possibly forwarded
- Authoritative name server

DNS
- 13 “logical” root name servers around the world

Transport
- How does the server figure out which packets are intended for the web server and which are for the SSH server?
  - Not explainable with the IP protocol
  - We need a logical communication channels
  - Transport Protocol
- Adding an additional header for the transport protocol
- Concept of a port
- Multiplexing
  - multiple applications using the same network connection
- Demultiplexing
  - deliver incoming segments to the correct application process
UDP
- User Datagram Protocol
- Connectionless
- "best effort"
  - segments may be lost
  - segments may be delivered out of order
- Used for loss tolerant and/or rate sensitive applications
  - Video streaming
  - DNS
  - SNMP

<table>
<thead>
<tr>
<th>source port #</th>
<th>dest port #</th>
<th>length</th>
<th>checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data

Reliable Transport
- Problems:
  - Message Corruption
    - Flipped bits
    - Truncated message
  - Message Loss
    - Problem: asynchronous communication
    - How to distinguish slow from lost message?
- What can we do?
  - Send ACKs
  - Use timeouts

Sending Acknowledgements
- Confirm that the receiver successfully received the packet
- New problem: Lost ACKs
- Retransmission, but...
  - ...need a way to identify packets
  - Sequence numbers

Use a binary counter as sequence number
- Packet loss: retransmit packet
  - OK
Reliable Transport

Use a binary counter as sequence number

- Packet loss: retransmit packet
  - OK
- ACK loss: retransmit packet
  - OK

Reliable Transport

Use a binary counter as sequence number

- Packet loss: retransmit packet
  - OK
- ACK loss: retransmit packet
  - OK
- Slow ACK
  - Premature timeout
  - OK

Reliable Transport Performance

- Latency or Delay:
  - Transmission delay
    - $T_{\text{transmit}} = \frac{L}{R}$
  - Propagation delay
    - $T_{\text{propagate}} = \frac{d}{s}$

1 Gbps link, 15 ms propagation delay, 1kB packet

$T_{\text{transmit}} = \frac{8kb/\text{pkt}}{10^9\text{b/s}} = 8\mu\text{s}$

With the propagation delay the ACK arrives 30.008ms later.

Utilization

$U = \frac{8kb / 30.008ms}{1Gb/s} \approx 0.027\%$
Sliding Windows

- Problem: ability to send/receive new segments is a function of how large the remaining space in the buffer is
- Does not make sense to send more when the receiver is unable to buffer more

Flow Control Protocol:
- Receiver
  - Informs sender of the currently available space in its receive buffer
    - RcvWindow field in the header
- Sender
  - Keeps the amount of unACKed data less than most recently received RcvWindow

TCP
- Reliable, stream-oriented protocol
  - Provides abstraction of byte streams over actual message boundaries
- Connection-oriented, point to point
- Full duplex

TCP Handshake
- Setting up a connection
  - Three-way handshake
  - Exchange initial sequence numbers
    - Initial sequence numbers are random
    - ACKs are always sequence number + 1
- Tearing down a connection
  - Client sends FIN
  - Server replies with ACK, then closes connection and sends FIN
Network Congestion

- Assume two pairs of hosts, router with infinite buffer space
  - What could cause network congestion?

- Expected behavior:

TCP Congestion Control

- Two possibilities:
  - Network-assisted congestion control
    - Routers give feedback
  - End-to-end congestion control
    - Endpoints infer congestion by observing the traffic

TCP Congestion Control

- If host observes long delays and/or packet loss, assume congestion
  - Probe for usable bandwidth
  - Adjust sending rate ("congestion window")

TCP Congestion Control

- Conservative: additive increase, multiplicative decrease
  - Increase linearly when last congestion window successfully sent
  - Back off quickly, halve the window when loss is detected
  - Important for stability
TCP Congestion Control

- TCP segments have a maximum size (MSS)
  - determined by lower layer protocols
  - increase window by MSS bytes
  - never decrease to < 1 MSS.
- TCP “Slow Start”, then congestion avoidance
  
  \[
  \text{throughput} = \frac{w \cdot \text{MSS}}{\text{RTT}} \quad \text{Bytes/sec}
  \]

- w segments, each with MSS bytes sent in one RTT:

```
send Base

nextWindow

already ack'd
sent, not yet ack'd
usable, not yet sent
not usable

window size
```

NAT

- Network Address Translation
  - Address translation table
    - Outgoing traffic: replace <source IP, port> with <NAT IP, new port #> as source address
    - Incoming traffic: do the reverse using the table
  - 16 bit port numbers
    - 60k possible NAT translations
  - NAT is “hacky”
    - But used in practice

IPv6

- IP (v4) uses 32 bit addresses
  - \(2^{32}\) addresses \(\approx 4.3\) billion
- IPv6 uses 128 bit addresses
  - Longer addresses \(\rightarrow\) easier to allocate blocks in a hierarchical manner
- Some changes to the header
  - Redesigned from “lessons learned”
  - E.g., allow for better QOS
  - Simplify header structure to ease the task of routing
- No fragmentation
- As a result: IPv4 and IPv6 are incompatible
  - Tunneling is possible