Local Area Network (LAN)
LAN = communications network
- Small area (building, set of buildings)
- Same, sometimes shared, transmission medium
- High data rate: 1 Mbps – 100 Gbps
- Low latency
- Devices are peers
  • any device can initiate a data transfer with any other device

Most elements on a LAN are workstations
- endpoints on a LAN are called nodes

Connecting nodes to LANs
Adapter
- expansion slot (PCIx, USB dongle)
- usually integrated onto main board

Network adapters are referred to as
Network Interface Controllers (NICs) or adapters
(or Network Interface Component, Network Interface Card)

Multiple Access Protocols
Ways of enabling multiple devices to share one network
Three approaches

1. Channel partitioning
   - Time Division Multiplexing (TDM)
     - each node gets a time slot
   - Frequency Division Multiplexing (FDM)
     - each channel gets a frequency band
   Fails badly; Complex

2. Taking turns
   - Polling protocol – master polls nodes in sequence
   - Token passing protocol – node needs a token to transmit
   Sounds like it shouldn’t work well … but it does!

3. Random access
   - No scheduled time slots
   - Statistical multiplexing
   - Retransmit if there’s a collision
   Bandwidth allocated even if nothing to transmit!

Modes of connection
Circuit-switching (virtual circuit)
- Dedicated path (route) – established at setup
- Guaranteed (fixed) bandwidth – routers commit to resources
- Typically fixed-length packets (cells) – each cell only needs a virtual circuit ID
- Constant latency

Packet-switching (datagram)
- Shared connection; competition for use with others
- Data is broken into chunks called packets
- Each packet contains a destination address
- Available bandwidth ≤ channel capacity
- Variable latency

© 2014-2015 Paul Krzyzanowski
Client-Server Networking

What’s in the data?
For effective communication
– same language, same conventions
For computers:
– electrical encoding of data
– where is the start of the packet?
– which bits contain the length?
– is there a checksum? where is it?
  how is it computed?
– what is the format of an address?
  byte ordering
These instructions and conventions are known as protocols

Layering
To ease software development and maximize flexibility:
– Network protocols are generally organized in layers
– Replace one layer without replacing surrounding layers
– Higher-level software does not have to know how to format an Ethernet packet
  … or even know that Ethernet is being used

Layering
Most popular model of guiding (not specifying) protocol layers is

OSI reference model
Created & adopted by ISO
Defines 7 layers of protocols

OSI Reference Model: Layer 1
Transmits and receives raw data to communication medium
Does not care about contents
Media, voltage levels, speed, connectors

Deals with representing bits
Examples: USB, Bluetooth, 1000BaseT, Wi-Fi

OSI Reference Model: Layer 2
Detects and corrects errors
Organizes data into frames before passing it down. Sequences packets (if necessary)
Accepts acknowledgements from receiver

Deals with frames
Examples: Ethernet MAC, PPP
OSI Reference Model: Layer 2

An ethernet switch is an example of a device that works on layer 2.

It forwards ethernet frames from one host to another as long as the hosts are connected to the switch (switches may be cascaded).

This set of hosts and switches defines the local area network (LAN).

OSI Reference Model: Layer 3

A router is an example of a device that works on layer 3.

A router takes an incoming IP packet and determines which interface to send it out.

It enables multiple networks to be connected together.

OSI Reference Model: Layer 4

Transport

Provides an interface for end-to-end (application-to-application) communication: sends & receives segments of data. Manages flow control. May include end-to-end reliability.

Network interface is similar to a mailbox.

Examples: TCP, UDP

OSI Reference Model: Layer 5

Session

Services to coordinate dialogue and manage data exchange.

Software implemented switch.

Manage multiple logical connections.

Keep track of who is talking; establish & end communications.

Examples: HTTP 1.1, SSL

OSI Reference Model: Layer 6

Presentation

Data representation.

Concerned with the meaning of data bits.

Convert between machine representations.

Examples: XDR, ASN.1, MIME, JSON, XML
**OSI Reference Model: Layer 7**

Collection of application-specific protocols

Examples:
- web (HTTP)
- email (SMTP, POP, IMAP)
- file transfer (FTP)
- directory services (LDAP)

**Internet Protocol**

- A set of protocols designed to handle the interconnection of a large number of local and wide-area networks that comprise the Internet

- **IPv4 & IPv6: network layer**
  - Other protocols include TCP, UDP, RSVP, ICMP, etc.
  - Relies on routing from one physical network to another
  - IP is connectionless
    - No state needs to be saved at each router
  - Survivable design: support multiple paths for data
    - … but packet delivery is not guaranteed!

**Internet Protocol stack vs. OSI protocol stack**

- **IP vs. OSI stack**

<table>
<thead>
<tr>
<th>Internet protocol stack</th>
<th>OSI protocol stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>Physical</td>
</tr>
<tr>
<td>Data Link</td>
<td>Data Link</td>
</tr>
<tr>
<td>Network</td>
<td>Network</td>
</tr>
<tr>
<td>Transport</td>
<td>Transport</td>
</tr>
<tr>
<td>Session</td>
<td>Session</td>
</tr>
<tr>
<td>Presentation</td>
<td>Presentation</td>
</tr>
<tr>
<td>Application</td>
<td>Application</td>
</tr>
</tbody>
</table>

**Protocol Encapsulation**

At any layer
- The higher level protocol headers are just treated like data
- Lower level protocol headers can be ignored

An Ethernet switch or Ethernet driver sees this:

![Ethernet frame](image)

A router or IP driver sees this:

![IP frame](image)

A TCP driver sees this:

![TCP frame](image)

An application sees this:

![Application frame](image)

**Addressing machines (data link layer)**

Each interface on a host has a unique MAC address
- E.g., aramis.rutgers.edu: 48-bit ethernet address = 00:03:ba:09:1b:b0

This allows us to address ethernet frames to machines on the same LAN

**Client – Server Communication**

- MAC = Media Access Control

© 2014-2015 Paul Krzyzanowski
Addressing machines (network layer)

Each interface on a host is given a unique IP address

- IPv4 (still the most common in the U.S.): 32-bit number
  - Example, cs.rutgers.edu = 128.6.4.2 = 0x80060402

- IPv6: 128-bit number
  - Example, cs.rutgers.edu = 0:0:0:0:FFFF:128.6.4.2 = ::FFFF:8006:0402

- Routable across networks
  - We can send IP packets to machines on the Internet
  - BUT … this only gets us to the machine, not the application

Address translation

- Domain name → IP address translation
  - Domain Name System (DNS)
    - Hierarchical human-friendly names (e.g., www.cs.rutgers.edu)
    - User-level network service to look up IP domain names
    - Cache results to avoid future look-ups
    - The kernel’s network drivers do not handle domain names

- IP → Ethernet MAC address translation
  - Address Resolution Protocol (ARP)
    - How does the OS know which ethernet address to use?
    - Broadcast an ARP query and wait for a response
      “Who has 128.6.4.2?”
    - Cache results to avoid future look-ups

Ethernet & IP – Message Reliability

- Ethernet
  - LAN connectivity
  - Higher-level protocols (IP) encapsulated inside
  - Unreliable delivery
    - Frames may be lost to congestion, errors, or collision

- IP
  - Datagram delivery is also unreliable
  - Frames may be lost due to dropped ethernet frames, errors, congestion, or time-to-live expiration

IP transport layer

IP give us two transport-layer protocols

- TCP: Transmission Control Protocol
  - Connection-oriented service
  - Operating system keeps state: simulates a virtual circuit over a datagram network
  - Full-duplex connection: both sides can send messages over the same link
  - Reliable data transfer: the protocol handles retransmission
  - In-order data transfer: the protocol keeps track of sequence numbers

- UDP: User Datagram Protocol
  - Connectionless service: lightweight transport layer over IP
  - Data may be lost
  - Data may arrive out of sequence
  - Checksum for corrupt data: operating system drops bad packets

Addressing applications (transport layer)

Communication endpoint at the machine

- Port number: 16-bit value
- Port number = transport endpoint
- Allows application-application communication
- Identifies a specific data stream
- Some services use well-known port numbers (0 – 1023)
- IANA: Internet Assigned Numbers Authority (www.iana.org)
- Also see the file /etc/services
  - ftp: 21/TCP
  - ssh: 22/tcp
  - smtp: 25/tcp
  - http: 80/tcp
  - ntp: 123/udp
- Ports for proprietary apps: 1024 – 49151
- Dynamic/private ports: 49152 – 65535
- To communicate with applications, we use a transport layer protocol and an IP address and port number

Network API

- App developers need access to the network
- A Network Application Programming Interface (API) provides this
  - Core services provided by the operating system
    - Operating System controls access to resources
  - Libraries may handle the rest

- We will only look at IP-based communication
Programming: connection-oriented protocols

1. establish connection  
   [dial phone number]
2. [negotiate protocol]  
   [decide on a language]
3. exchange data  
   [speak]
4. terminate connection  
   [hang up]

**virtual circuit service**

- provides illusion of having a dedicated circuit
- messages guaranteed to arrive in-order
- application does not have to address each message

*Not to be confused with virtual circuit networks*

- Which provide constant latency & guaranteed bandwidth
- TCP simulates a virtual circuit network ... sort of

Programming: connectionless protocols

- no call setup
- send/receive data  
  (each packet addressed)
- no termination

**datagram service**

- client is not positive whether message arrived at destination
- no state has to be maintained at client or server
- cheaper but less reliable than virtual circuit service