Network Layering
Parts of a Network
Parts of a Network

- Host
- Router
- Link
- App
## Component Names

<table>
<thead>
<tr>
<th>Component</th>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong>, or app, user</td>
<td>Generates messages</td>
<td>Zoom, iTunes, Browser</td>
</tr>
<tr>
<td><strong>Host</strong>, or end-system, edge device, node, source, sink</td>
<td>Runs the app</td>
<td>Laptop, mobile, desktop</td>
</tr>
<tr>
<td><strong>Router</strong>, or switch, node, hub</td>
<td>Relays messages across links</td>
<td>Access point, cable/DSL modem</td>
</tr>
<tr>
<td><strong>Link</strong>, or channel</td>
<td>Carries messages</td>
<td>Wires, wireless</td>
</tr>
</tbody>
</table>
Parts of a Network

- Host
- Router
- Link
- App
Parts of a Network
Types of Links

- **Full-duplex**
  - Bidirectional

- **Half-duplex**
  - Bidirectional

- **Simplex**
  - Unidirectional
Wireless Links

• Message is **broadcast**
  • Received by all nodes in range
  • Not a good fit with our model
Wireless Links (2)

• Often show logical links
  • Not all possible connectivity
A Small Network

• Connect a couple of computers

• Next, a large network ...
Example Computer Networks?
Example Computer Networks

• WiFi (802.11)
• Enterprise / Ethernet
• ISP (Internet Service Provider)
• Cable / DSL
• Mobile phone / cellular (2G, 3G, 4G, 5G)
• Bluetooth
• Telephone
• Satellite ...
# Computer network names by scale

<table>
<thead>
<tr>
<th>Scale</th>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vicinity</td>
<td>PAN (Personal Area Network)</td>
<td>Bluetooth (e.g., headset)</td>
</tr>
<tr>
<td>Building</td>
<td>LAN (Local Area Network)</td>
<td>WiFi, Ethernet</td>
</tr>
<tr>
<td></td>
<td>DCN (Data Center Network)</td>
<td>Ethernet</td>
</tr>
<tr>
<td>City</td>
<td>MAN (Metropolitan Area Network)</td>
<td>Cable, DSL</td>
</tr>
<tr>
<td>Country</td>
<td>WAN (Wide Area Network)</td>
<td>Large ISP</td>
</tr>
<tr>
<td>Planet</td>
<td>The Internet (network of all networks)</td>
<td>The Internet!</td>
</tr>
</tbody>
</table>
Internetworks

• An internetwork, or internet, is what you get when you join networks together
  • Just another network

• The Internet (capital “I”) is the internet we all use
Network Boundaries

- What part is the “network”? 

Diagram:
- Host
- Router
- Link
- App
Network Boundaries (2)

- What part represents an “ISP”?
Network Boundaries (3)

• Cloud as a generic network
Key Interfaces

• Between (1) apps and network, and (2) network components
What should networks do for apps?
What should networks do for apps?

• Make and break connections
• Find a path through the network
• Transfers information reliably
• Transfers arbitrary length information
• Send as fast as the network allows
• Shares bandwidth among users
• Secures information in transit
• Lets many new hosts be added
  • ...
What should networks do for apps?

- Make and break connections
- Find a path through the network
- Transfers information reliably
- Transfers arbitrary length information
- Send as fast as the network allows
- Shares bandwidth among users
- Secures information in transit
- Lets many new hosts be added
- ...

We need modularity to help manage complexity and support reuse
Protocols and Layers

• Protocols and layering is the main structuring method used to divide up network functionality
  • Divide functionality in layers organized vertically
  • Each protocol implements the functionality of that layer
  • Each protocol instance talks virtually to its peer instances using the protocol
  • Each protocol instances uses only the services of the lower layer
Protocols and Layers (2)

- Protocols are horizontal, layers are vertical

```
Instance of protocol X

X

Protocol X

Peer instance

Lower layer instance (of protocol Y)

Y

Service provided by Protocol Y

Node 1

Node 2
```
Protocols and Layers (3)

- Set of protocols in use is called a protocol stack
Protocols and Layers (4)

• Protocols you’ve probably heard of:
  • TCP, IP, 802.11, Ethernet, HTTP, SSL, DNS, ...
    ... and many more
Protocols and Layers (5)

• Protocols you’ve probably heard of:
  • TCP, IP, 802.11, Ethernet, HTTP, SSL, DNS, ... and many more

• An example protocol stack
  • Used by a web browser on a host that is wirelessly connected to the Internet
Encapsulation

• **Encapsulation** is the mechanism used to effect protocol layering
  • Lower layer wraps higher layer content, adding its own information to make a new message for delivery
  • Like sending a letter in an envelope; postal service doesn’t look inside
Encapsulation (2)

- Message “on the wire” begins to look like an onion
  - Lower layers are outermost
Encapsulation (3)
Encapsulation (4)

• Normally draw message like this:
  • Each layer adds its own header

<table>
<thead>
<tr>
<th>802.11</th>
<th>IP</th>
<th>TCP</th>
<th>HTTP</th>
</tr>
</thead>
</table>

  First bits on the wire

  Last bits

• More involved in practice
  • Trailers as well as headers, encrypt/compress contents
  • Segmentation (divide long message) and reassembly
Multiple protocols in a layer
Demultiplexing

• Pass incoming message to the protocols that it uses
Demultiplexing (2)

• Done with **demultiplexing identifiers in the headers**

![Diagram showing the demultiplexing process]

- **Ethertype value**
- **IP protocol field**
- **TCP port number**
Advantage of Layering

• Information hiding and reuse
Advantage of Layering (2)

• Information hiding and reuse
Advantage of Layering (3)

• Using information hiding to connect different systems
Advantage of Layering (4)

- Information hiding to connect different systems
Advantage of Layering (5)

• Information hiding to connect different systems
Disadvantages of Layering

• ?
Disadvantage of Layering

• Adds overhead
  • More problematic with short messages

• Hides information
  • App might care about network properties (e.g., latency, bandwidth, etc)
  • Network may need to know about app priorities (e.g., QoS)
# OSI Layers

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application (7)</td>
<td>Services that are used with end user applications</td>
<td>SMTP,</td>
</tr>
<tr>
<td>Presentation (6)</td>
<td>Formats the data so that it can be viewed by the user encrypt and decrypt</td>
<td>JPG, GIF, HTTPS, SSL, TLS</td>
</tr>
<tr>
<td>Session (5)</td>
<td>Establishes/ends connections between two hosts</td>
<td>NetBIOS, PPTP</td>
</tr>
<tr>
<td>Transport (4)</td>
<td>Responsible for the transport protocol and error handling</td>
<td>TCP, UDP</td>
</tr>
<tr>
<td>Network (3)</td>
<td>Reads the IP address form the packet.</td>
<td>Routers, Layer 3 Switches</td>
</tr>
<tr>
<td>Data Link (2)</td>
<td>Reads the MAC address from the data packet</td>
<td>Switches</td>
</tr>
<tr>
<td>Physical (1)</td>
<td>Send data on to the physical wire.</td>
<td>Hubs, NICS, Cable</td>
</tr>
</tbody>
</table>
Protocols and Layering

• The real internet protocol stacks:

1 Link
   - Ethernet
   - 3G
   - Cable
   - DSL
   - 802.11

2 Internet
   - IP

3 Transport
   - TCP
   - UDP

4 Application
   - SMTP
   - HTTP
   - RTP
   - DNS
   - “Narrow waist”
Course Reference Model

• We mostly follow the Internet

<table>
<thead>
<tr>
<th>Level</th>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Application</td>
<td>Programs that use network service</td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
<td>Provides end-to-end data delivery</td>
</tr>
<tr>
<td>3</td>
<td>Network</td>
<td>Send packets over multiple networks</td>
</tr>
<tr>
<td>2</td>
<td>Link</td>
<td>Send frames over one or more links</td>
</tr>
<tr>
<td>1</td>
<td>Physical</td>
<td>Send bits using signals</td>
</tr>
</tbody>
</table>
Lecture Progression

Middle → top → bottom

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Application</td>
<td>- HTTP, DNS, CDNs</td>
</tr>
<tr>
<td>1. Transport</td>
<td>- TCP, UDP</td>
</tr>
<tr>
<td>2. Network</td>
<td>- IP, NAT, BGP</td>
</tr>
<tr>
<td>5. Link</td>
<td>- Ethernet, 802.11</td>
</tr>
<tr>
<td>4. Physical</td>
<td>- wires, fiber, wireless</td>
</tr>
</tbody>
</table>

Followed by more detail on cross-cutting elements:

- Quality of service, Security (VPN, SSL)
What should networks do for apps?

- Make and break connections
- Find a path through the network
- Transfers information reliably
- Transfers arbitrary length information
- Send as fast as the network allows
- Shares bandwidth among users
- Secures information in transit
- Lets many new hosts be added
- ...

Alternative ways of organizing all of this functionality?