Outline

• Why (or what) computer networks?
• Layered protocols
• Reference models
• Some basic definitions
• The Physical Layer (very briefly!)
• The schedule for this module (revisited...)
Why (or what) Computer Networks?
(really..?)
Examples of Computer Networks

- Examples of computer networks?
Layered Protocols

and services, interfaces, architectures, reference models...
Protocols

• What are some examples of network protocols?

• Why do we have (layered) protocols/protocol stacks?
Protocol Layering

- Example...

- The protocol at each different layers serves a different purpose

(or provides a different service)
Protocol Layering

- Protocol layering is the main structuring method used to divide up network functionality.
- Each protocol instance talks virtually to its peer.
- Each layer communicates only by using the one below.
- Lower layer services are accessed by an interface.
- At bottom, messages are carried by the medium.
- We’ll come back to what the layers are and how they’re divided later on...
Relationship of Services to Protocols

- A layer provides a service to the one above [vertical]
- A layer talks to its peer using a protocol [horizontal]
Service Primitives

- A service is provided to the layer above as primitives
- Hypothetical example of service primitives that may provide a reliable byte stream (connection-oriented) service:

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>LISTEN</td>
<td>Block waiting for an incoming connection</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Establish a connection with a waiting peer</td>
</tr>
<tr>
<td>ACCEPT</td>
<td>Accept an incoming connection from a peer</td>
</tr>
<tr>
<td>RECEIVE</td>
<td>Block waiting for an incoming message</td>
</tr>
<tr>
<td>SEND</td>
<td>Send a message to the peer</td>
</tr>
<tr>
<td>DISCONNECT</td>
<td>Terminate a connection</td>
</tr>
</tbody>
</table>
Service Primitives

- Hypothetical example of how these primitives may be used for a client-server interaction

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT (1)</td>
<td>LISTEN (0)</td>
</tr>
<tr>
<td>SEND RECEIVE (3)</td>
<td>ACCEPT RECEIVE (2)</td>
</tr>
<tr>
<td>DISCONNECT (5)</td>
<td>SEND (4)</td>
</tr>
</tbody>
</table>

- Connect request
- Accept response
- Request for data
- Reply
- Disconnect
- Disconnect
Protocol Layers

- Each lower layer adds its own header (with control information) to the message to transmit and removes it on receive.

- Layers may also split and join messages, etc.
Design Issues

- Computer networks have to address the following design issues (and more):
  - Reliability: error detection and correction
  - Packet routing
  - Addressing and naming
  - Internetworking
  - Scalability
  - Congestion and flow control
  - Throughput, latency and QoS management
  - Confidentiality and authentication

- This module will look at where and how these are addressed in the protocol layers
Connection-oriented vs Connectionless Service

• Connection-oriented service
  – A connection is negotiated and setup, and released when finished.
  – Provides a communication 'tube', and manages flow through it.
  – May provide a reliable or unreliable service:
    • Reliable: file transfers
    • Unreliable: VoIP/video conferencing

• Connectionless service
  – ‘Messages’ are sent without establishing a connection
  – Messages may be received out of order
  – May provide a reliable or unreliable service:
    • Reliable: acknowledged datagram (e.g. an SMS)
    • Unreliable: unacknowledged datagram (send and hope!)
Reference Models
Reference Models

• We will consider three:
  – The OSI Reference Model
  – The TCP/IP Reference Model
  – Model used in Tanenbaum (*which we’ll use too*)
OSI Reference Model

- Created by the International Standards Organization (ISO) in 1983
- The Open Systems Interconnection (OSI) reference model is/was a principled, international standard to connect different systems, consisting of 7 layers.

- A layer is used where a different abstraction is needed
- Each layer provides a well-defined function
- Layer boundaries are chosen to minimize information flow across interfaces
- A balance in the number of layers is made such that
  - distinct functions aren’t thrown together
  - there aren’t an unwieldy number of layers

Figure: CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011
## OSI Reference Model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Application</td>
<td>Provides functions needed by users</td>
</tr>
<tr>
<td>6 Presentation</td>
<td>Converts different representations</td>
</tr>
<tr>
<td>5 Session</td>
<td>Manages task dialogs</td>
</tr>
<tr>
<td>4 Transport</td>
<td>Provides end-to-end delivery</td>
</tr>
<tr>
<td>3 Network</td>
<td>Sends packets over multiple links</td>
</tr>
<tr>
<td>2 Data link</td>
<td>Sends frames of information</td>
</tr>
<tr>
<td>1 Physical</td>
<td>Sends bits as signals</td>
</tr>
</tbody>
</table>

*Figure: CNSE by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011*
OSI Reference Model

Image: CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011
OSI Reference Model

+ Very influential model with clear concepts
+ Excellent tool for understanding networks (and hence appears in most textbooks) but can be safely ignored, because...
  - Model created first, then protocols designed to fit (hence unforeseen shortcomings – for example, no support for internetworking: it was assumed that each country would only have one network, and it would be running the OSI protocols!)
  - Only supports connection-oriented communication in the transport layer (supports connectionless and connection-oriented in the network layer) – the user doesn’t get a choice
  - Complex models and protocols led to bad implementations, hence gained a reputation for being poor quality
  - Viewed as being pushed by the government
  - Doesn’t map well to TCP/IP

Image: CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011
TCP/IP Reference Model

- TCP/IP (Transmission Control Protocol over Internet Protocol)
- Important: forms the basis of the Internet!

- US DoD supported the ARPANET project to connect university and government sites using leased telephone lines.

- When satellite and radio links were later added, existing protocols weren’t sufficient, and TCP/IP was born. Required:
  - Seamless internetworking
  - Resilience
  - Flexibility

- Led to a packet-switching network based on a connectionless layer which runs across different networks (IP).
TCP/IP Reference Model

IP is the “narrow waist” of the Internet

Protocols are shown in their respective layers

Figure: CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011
TCP/IP Reference Model

+ Gained widespread use (before the OSI Reference Model was announced)
+ Had good early implementations
+ Supports connectionless and connection-oriented communication in the transport layer (but only supports connection-oriented communication in the network layer) - gives the user the choice

- Not really a model, just a description of existing protocols!
- Fits TCP/IP (perfectly), but nothing else very well!
- Doesn’t define services, interfaces and protocols – hence not useful as a guide for designing new technologies
Model Used in Tanenbaum/ELEC3222

- It is based on the TCP/IP model but we pull out the physical layer and look beyond Internet protocols.

<table>
<thead>
<tr>
<th>5</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Transport</td>
</tr>
<tr>
<td>3</td>
<td>Network</td>
</tr>
<tr>
<td>2</td>
<td>Link</td>
</tr>
<tr>
<td>1</td>
<td>Physical</td>
</tr>
</tbody>
</table>

- Programs that make use of the network. Doesn’t include the UI (e.g. web browser), but the portion that uses the network
- Strengthens the delivery guarantees of the Network layer providing increased reliability (e.g. TCP)
- How to combine multiple links into networks, and networks of networks into internetworks. Includes finding paths (routing) (e.g. IP)
- How to send finite length messages between directly connected computers with specified levels of reliability (e.g. Ethernet or 802.11)
- How to transmit bits across different kinds of media
Definitions
Definitions

• Computer Network  a collection of autonomous computers interconnected (they can exchange information) by a single technology.

• Internetwork  multiple networks joined together.

• Service  defines the functionality a layer provides, i.e. the semantics

• Interface  defines the operations and services the low layer makes to the upper one

• Protocol  an agreement between communicating parties on how communication is to proceed

• Protocol Stack  a list of the protocols used by a certain system, with one protocol per layer

• Network Architecture  a set of layers and protocols, e.g. the Internet
Metric Units

• Capitalisation of prefixes
  – For units >1 (e.g. mega, giga, peta), capitalise the symbol
    • e.g. Mb, Gb, Pb
  – For units <1 (e.g. milli, nano, pico), don’t capitalise the symbol
    • e.g. mbps, nbps, pbps (probably not used so often!)
  – kbps (kilo bits per second) is considered an exception to this rule

• Units
  – B = bytes; b = bits
    • e.g. MB = megabytes; mb = millibits

• Powers of 2 or 10 in prefixes?
  – Usually, prefix units use powers of 10
    • e.g. 1 kbps = 10^3 (1000) bps
  – However, storage (disk, memory, file, database etc) traditionally uses powers of 2
    • e.g. 1 kB = 2^10 (1024) bits
Standardisation

- Standards:
  - Increase the market for products adhering to the standards
  - Lead to mass production, economies of scale, better implementations etc

- Standards do:
  - define what a protocol should do to ensure interoperability
  - dictate the protocol ‘over the wire’

- Standards don’t:
  - define how a protocol should work so that you do a good job of implementing it
  - specify the service interface ‘inside the box’

- Some standards (‘de-facto’) come after the invention
- Other standards (‘de-jure’) come about from official formal bodies
## Standardisation

<table>
<thead>
<tr>
<th>Body</th>
<th>Area</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITU</td>
<td>Telecommunications</td>
<td>G.992, ADSL H.264, MPEG4</td>
</tr>
<tr>
<td>IEEE</td>
<td>Communications</td>
<td>802.3, Ethernet 802.11, WiFi</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet</td>
<td>RFC 2616, HTTP/1.1 RFC 1034/1035, DNS</td>
</tr>
<tr>
<td>W3C</td>
<td>Web</td>
<td>HTML5 standard CSS standard</td>
</tr>
</tbody>
</table>

*Figure: CN5E by Tanenbaum & Wetherall, © Pearson Education-Prentice Hall and D. Wetherall, 2011*
The Physical Layer
The Physical Layer

- Concerned with how to communicate raw bits over a physical communication channel
  - e.g. a wired (coaxial/optical) or wireless (radio/satellite) channel
  - Bit representation, timing, pins/wires etc

- Most of your existing modules on communications were related to this layer.

- We’re going to skip it for now, and will come back to it later on...
  - Why?
The Schedule for this Module (revised...)
List of Topics

- Network architectures and principles Layered networking models (e.g. TCP/IP)
- Physical networks and their design; wireless networks
- Data link layer (e.g. Ethernet, Wi-Fi)
- Network layer (e.g. IP networking, addressing, routing, QoS, multicast)
- Transport layer (e.g. TCP and UDP)
- Application layer
- Standardisation of communication protocols
- Principles of network security
- Emerging network technologies
- Networking for constrained embedded devices
- Development environments for constrained embedded devices and networks
ELEC3222 Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Physical</td>
</tr>
<tr>
<td>2</td>
<td>Link</td>
</tr>
<tr>
<td>3</td>
<td>Network</td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
</tr>
<tr>
<td>5</td>
<td>Application</td>
</tr>
</tbody>
</table>

Pass 1 (overview)
- Weeks 1-2
- Weeks 3-4

Pass 2 (examples, more detail etc)
- Weeks 5+
Questions?