Principles of Modern Communications Network Standards

based on 2011 lecture series by Dr. S. Waharte.
Department of Computer Science and Technology,
University of Bedfordshire.

11th January 2013

Outline

Definitions and concepts

Last Lecture

- Lecture 1 introduced core concepts and principles.
- One of these was the importance of standards.
- In this lecture, we will look at standards in much more depth.

- Lecture 1 discussed standards in terms of five layers of operation.
- This will also be an important focus of Lecture 2.

Layer Name	Number
Application	5
Transport	4
Internet	3
Data link	2
Physical	1

Standards

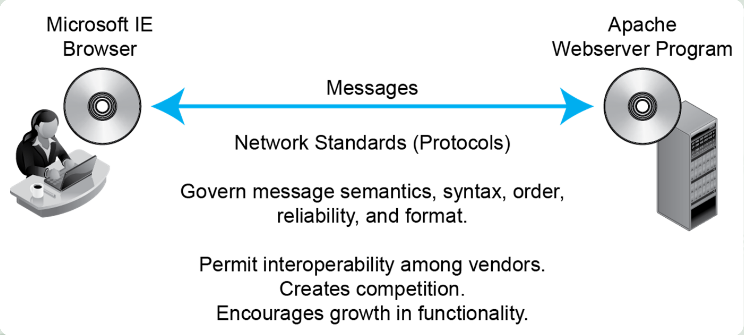
- Standards allow different systems to work together.
- The terms "standard" and "protocol" mean the same thing.

Standards

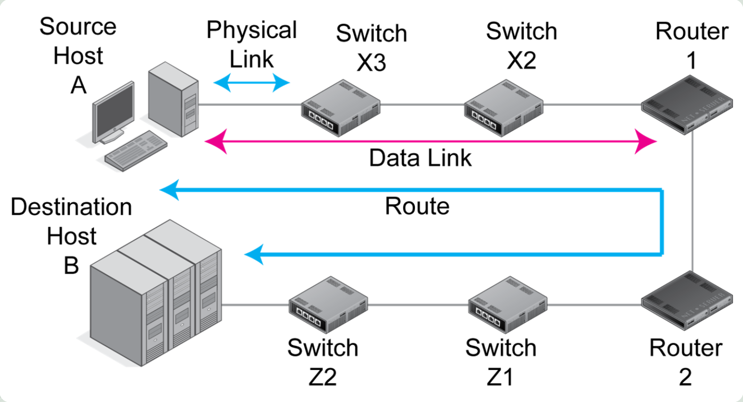
Example

- What is the standard language in this classroom?
- What would happen if you didn't have this standard?

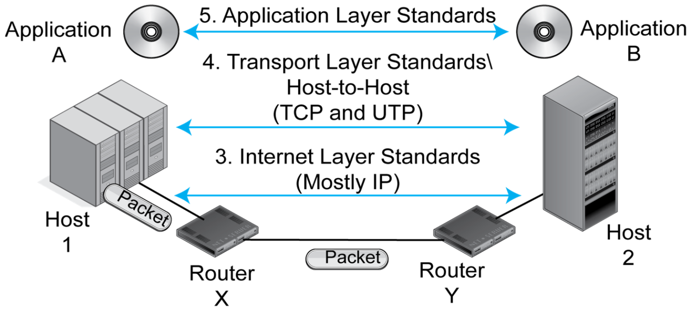
Network Standards



Physical, Data Link, and Internet Layers



Internet, Transport & Application Layers



Layers

Broad Function	Layer Number	Layer Name
Interoperability of application programs	5	Application
Transmission across an internet	4	Transport
	3	Internet
Transmission across a single switched or wireless network	2	Data Link
	1	Physical

Message ordering and reliability

Network Standards Concepts

- Network Standards
  - Network standards are rules that govern the exchange of messages between hardware or software processes on different hosts, including messages (ordering, semantics, and syntax), reliability, and connection orientation.
  - What are ordering, semantics, and syntax?

Network Standards Concepts

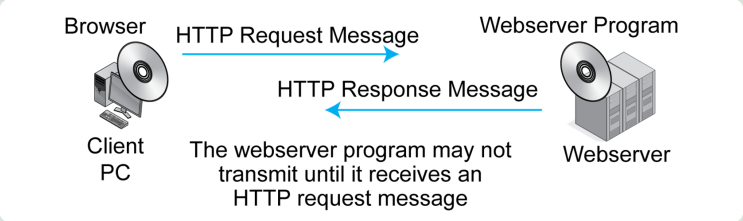
- Message Order
  - Turn taking (think of telephone conversations)
  - Order of messages in a complex transaction
  - Who must initiate communication, and so on
- Reliability
  - A reliable protocol both detects and corrects errors during transmission.
  - Error detection alone is not enough.
  - Some unreliable protocols detect errors but then only drop incorrect messages.

Network Standards Concepts

- Message Order in HTTP
  - In HTTP, the client program initiates the communication by sending an HTTP request message

Network Standards Concepts

- Message Order in HTTP



Network Standards Concepts

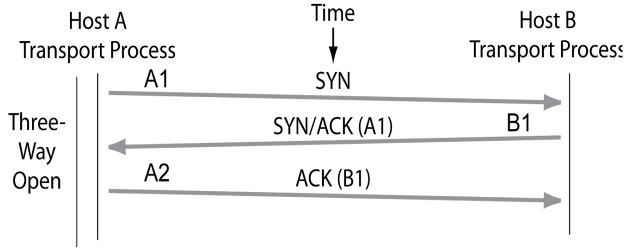
- Reflection: Message Order in HTTP
  - The client must initiate the interaction.
  - Other network standards have more complex turn taking; for instance TCP.
  - Human turn taking is loose and flexible.
  - But message order for network standards must be rigid because computers are not intelligent.
  - TCP exemplifies more complex message ordering.

A Connection

- HTTP is connectionless.
  - Every request-response cycle is independent.
- TCP is connection-oriented.
  - There is a formal opening of the connection.
  - Within the connection, messages are sequenced, acknowledged, and retransmitted if necessary.
  - There is a formal closing of the connection.

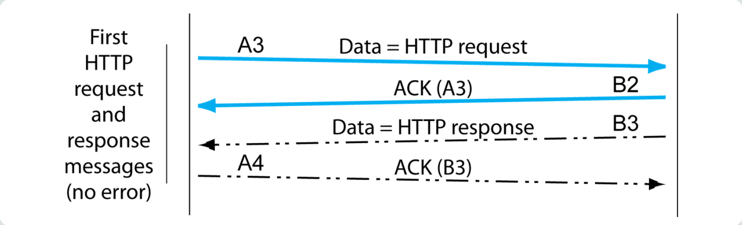
A TCP Connection

- A connection opening requires three segments.
  - SYN segments only have headers with the SYN bit set.



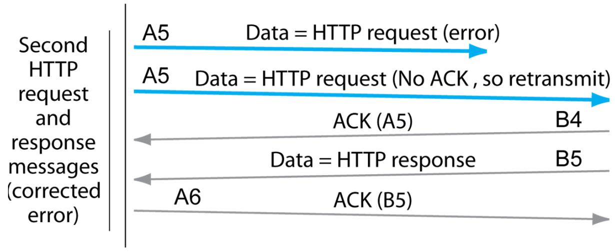
A TCP Connection

- HTTP request and response messages are acknowledged if received correctly.



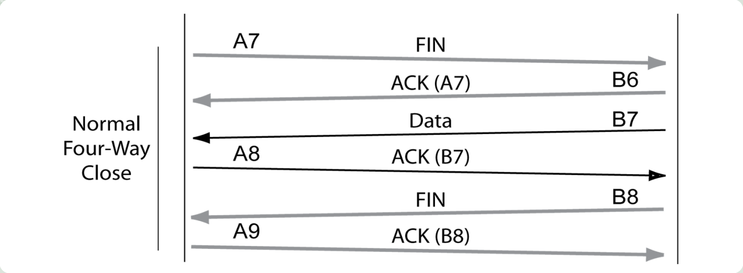
A TCP Connection

- Unacknowledged segments are retransmitted.



A TCP Connection

- Four messages close connections.
  - FIN segments are headers with the FIN bit set.



Semantics and syntax

Network Standards Concepts

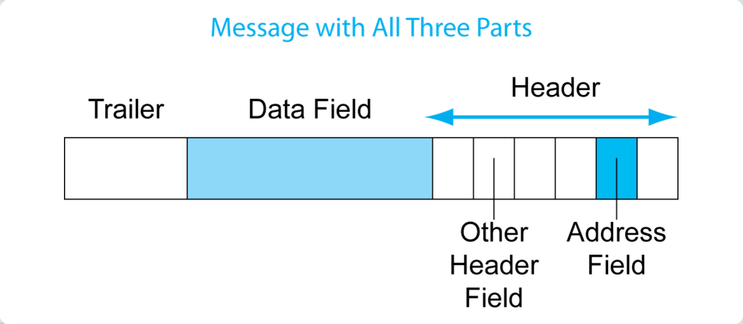
- Message Semantics
  - Semantics = the meaning of a message.
  - HTTP request message: "Please give me this file."
  - HTTP response message: Here is the file. (Or, I could not comply for the following reason...)

Network Standards Concepts

- Message Semantics
  - Network standards normally have a very limited set of possible message meanings.
  - For example, HTTP requests have only a few possible meanings.
    - GET: Please give me a file.
    - PUT: Store this file (not often used).
    - A few more.

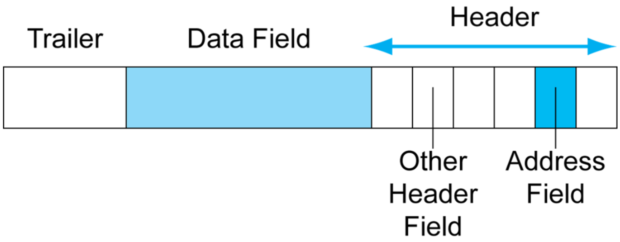
General Message Organization

- Message Syntax: Message Layout
  - Like human grammar, but more rigid.
  - In general, messages may have three parts.



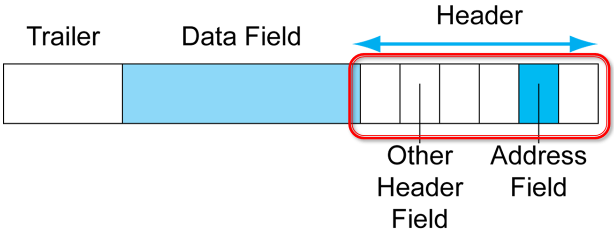
General Message Organization

- Message Syntax
  - The data field contains the content being delivered.
  - The header is everything before the data field.
  - The trailer is everything after the data field.



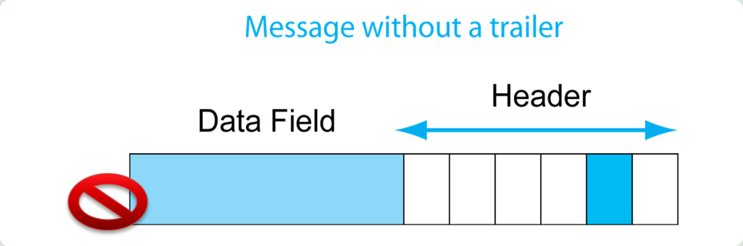
General Message Organization

- Message Syntax
  - The header is usually broken into smaller sections called header fields.
  - There often is an address field to indicate where to deliver the message.



General Message Organization

- Most Messages Do Not Have Trailers.

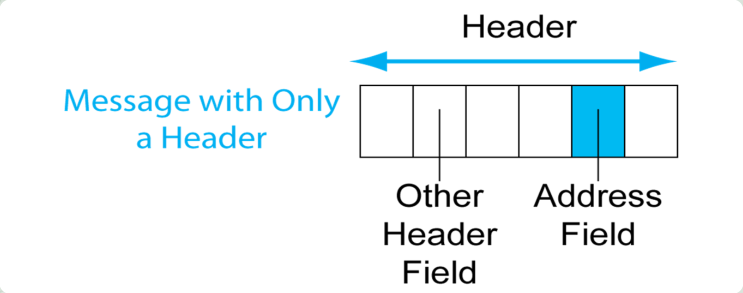


General Message Organization

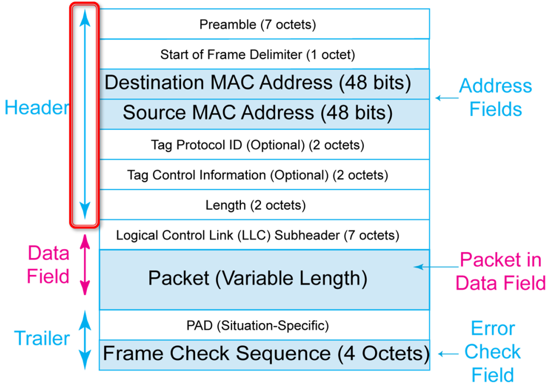
- Field lengths may be measured in bits or bytes.
- Another name for byte is “octet.”
- The term “octet” is used frequently in networking.

General Message Organization

- Some Messages Do Not Have Data Fields
- Example: TCP SYN segments only have headers.

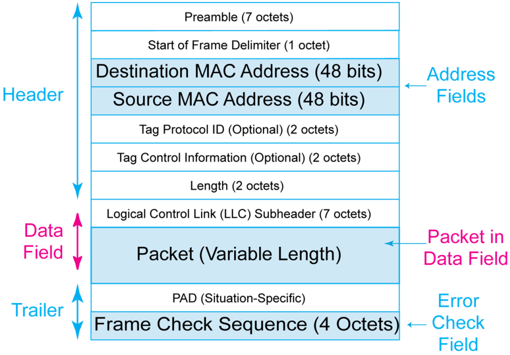


Ethernet Frame



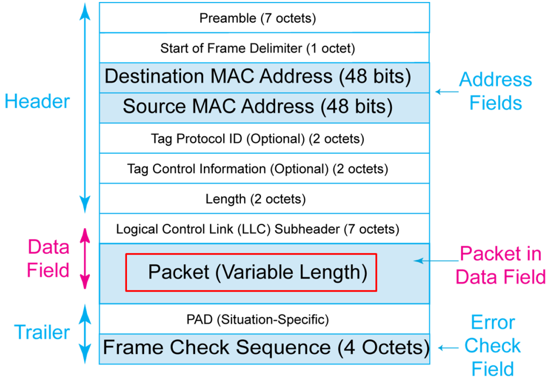
- The Ethernet frame has a header, data field, and trailer. The header has multiple fields.

Ethernet Frame



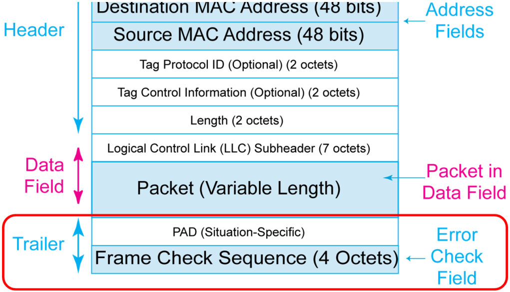
- There are source and destination address fields.
- Ethernet addresses are called MAC addresses (We will see why in Chapter 6).
- MAC addresses are 48 bits long. (In contrast, IP addresses in IP packets are 32 bits long.)

Ethernet Frame



- The Ethernet data field contains the packet that the frame is delivering.

Ethernet Frame



- The trailer contains the Frame Check Sequence field for error detection.
- The sender computes the value based on all bits. The receiver redoes the calculation.
- If the receiver gets a different value, it discards the frame. There is no error correction.
- Ethernet is not reliable.

Internet Protocol (IP) Packet

David Goodwin
University of Bedfordshire

Bit 0				Bit 31
Version Number (4 bits)	Header Length (4 bits)	Diff-Serv (8 bits)	Total Length (16 bits)	
Identification (16 bits)			Flags (3 bits)	Fragment Offset (13 bits)
Time to Live (8 bits)		Protocol (8 bits)	Header Checksum (16 bits)	
Source IP Address (32 bits)				
Destination IP Address (32 bits)				
Options (if any)				Padding (to 32-bit boundary)
Data Field (dozens, hundreds, or thousands of bits) Often contains a TCP segment				

- The IP packet is a long string of bits. It is drawn 32 bits on each line.
- The first line has the bits 0 through 31. The second line has bits 32 to 63, and so on.

Internet Protocol (IP) Packet

Bit 0 … Bit 31

<table>
<tr><td>Version Number (4 bits)</td><td>Header Length (4 bits)</td><td>Diff-Serv (8 bits)</td><td colspan="2">Total Length (16 bits)</td></tr>
<tr><td colspan="3">Identification (16 bits)</td><td>Flags (3 bits)</td><td>Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live (8 bits)</td><td>Protocol (8 bits)</td><td colspan="2">Header Checksum (16 bits)</td></tr>
<tr><td colspan="5">Source IP Address (32 bits)</td></tr>
<tr><td colspan="5">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="4">Options (if any)</td><td>Padding (to 32-bit boundary)</td></tr>
<tr><td colspan="5">Data Field (dozens, hundreds, or thousands of bits)
Often contains a TCP segment</td></tr>
</table>

- For the Header Checksum field, the sender calculates a number based on other bits and places the number in the Header Checksum field.
- The receiver redoes the calculation. If the receiver does not get what the sender computed, then there has been an error, and the packet is dropped.

Internet Protocol (IP) Packet

Bit 0 … Bit 31

<table>
<tr><td>Version Number (4 bits)</td><td>Header Length (4 bits)</td><td>Diff-Serv (8 bits)</td><td colspan="2">Total Length (16 bits)</td></tr>
<tr><td colspan="3">Identification (16 bits)</td><td>Flags (3 bits)</td><td>Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live (8 bits)</td><td>Protocol (8 bits)</td><td colspan="2">Header Checksum (16 bits)</td></tr>
<tr><td colspan="5">Source IP Address (32 bits)</td></tr>
<tr><td colspan="5">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="4">Options (if any)</td><td>Padding (to 32-bit boundary)</td></tr>
<tr><td colspan="5">Data Field (dozens, hundreds, or thousands of bits)
Often contains a TCP segment</td></tr>
</table>

- There are 32-bit fields for the source IP address and the destination IP address.

Internet Protocol (IP) Packet

Bit 0 … Bit 31

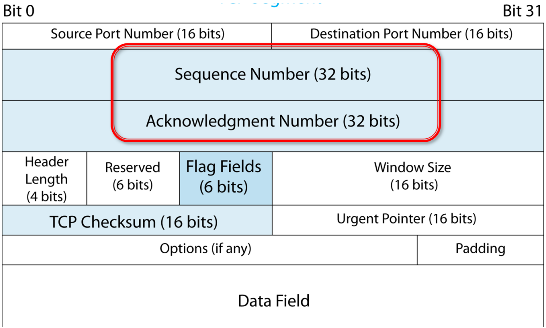
<table>
<tr><td>Version Number (4 bits)</td><td>Header Length (4 bits)</td><td>Diff-Serv (8 bits)</td><td colspan="2">Total Length (16 bits)</td></tr>
<tr><td colspan="3">Identification (16 bits)</td><td>Flags (3 bits)</td><td>Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live (8 bits)</td><td>Protocol (8 bits)</td><td colspan="2">Header Checksum (16 bits)</td></tr>
<tr><td colspan="5">Source IP Address (32 bits)</td></tr>
<tr><td colspan="5">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="4">Options (if any)</td><td>Padding (to 32-bit boundary)</td></tr>
<tr><td colspan="5">Data Field (dozens, hundreds, or thousands of bits)
Often contains a TCP segment</td></tr>
</table>

- The data field often contains a TCP segment.
- It may also contain a UDP datagram.

Internet Protocol (IP) Packet

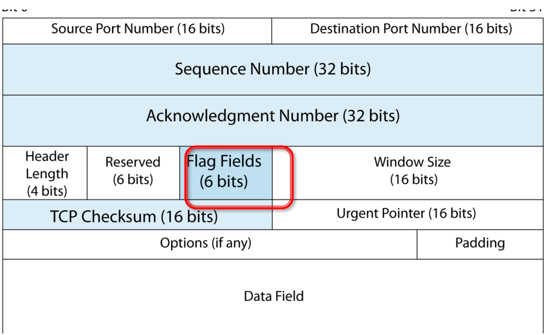
- The IP packet always has a header.
- The IP packet always has a data field.
- The IP packet NEVER has a trailer.
  - In fact, trailers are only found on Layer 2 frames.
  - In fact, many Layer 2 frames do not have trailers.

TCP Segment



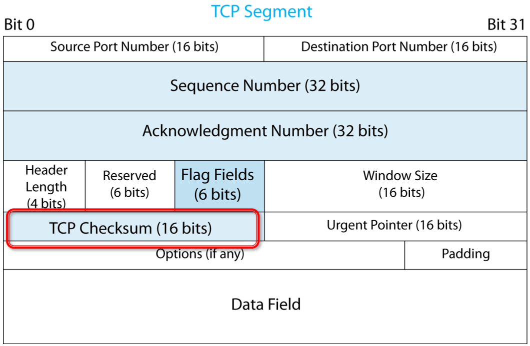
- The Sequence Number field gives the TCP segment's order in the session.
- The Acknowledgement Number field indicates the segment that this segment is acknowledging.
- These fields are 32 bits long.

TCP Segment



- Flag fields are one-bit fields.
- If the value is 1, the flag field is Set.
- If the value is 0, the flag field is Not Set.
- Flag bits are SYN, ACK, FIN, and RST.
- TCP has six flag fields.

TCP Segment



- The TCP Checksum Field is for error correction.
- The sender computes the value in the field.
- If the receiver computes the same value, it sends an ACK.
- If not, the receiver discards the segment and sends nothing. The sender will resend the segment. TCP is reliable.

TCP Segment

- TCP segments always have headers.
- SOME TCP segments do not have data fields.
  - Supervisory segments, such as SYN, do not carry data because the information to be conveyed, such as opening a connection, delivers no data.
- TCP segments NEVER have trailers.

UDP Datagram

UDP Datagram

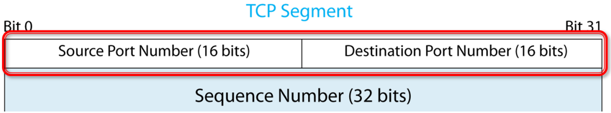
Bit 0	Bit 31
Source Port Number (16 bits)	Destination Port Number (16 bits)
UDP Length (16 bits)	UDP Checksum (16 bits)
Data Field	

- In UDP, the application message must fit into a single UDP datagram.
- So few header fields are needed.
- The UDP Checksum field is for error detection.
- If there is an error, UDP discards the datagram.
- If it does not detect an error, it accepts the datagram but does not send an acknowledgement.

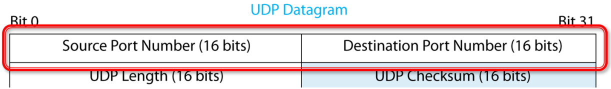
Perspective on TCP and UDP

- If the transport layer process detects an error in an arriving TCP segment, what does it do?
- If the transport layer process detects an error in an arriving UDP datagram, what does it do?
- Are these actions different?

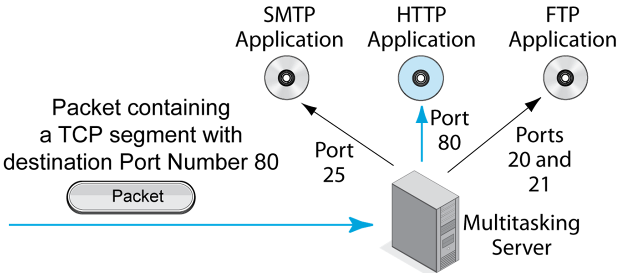
Recap: Port Number Fields



- Both TCP and UDP headers begin with
  - a source port number field and
  - a destination port number field.

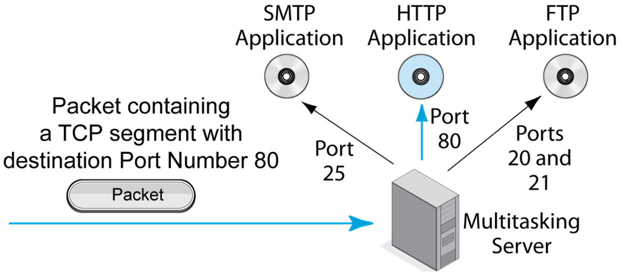


Port Numbers on Servers



- Multitasking servers run multiple applications.
- Each application is assigned a port number.

Port Numbers on Servers



- Major applications usually are given well-known port numbers from 0 to 1023.
- HTTP's well-known port number is 80.
- The File Transfer Protocol has two: 20 and 21.

Hypertext Transfer Protocol Message Syntax

- HTTP
  - The application layer is the highest layer.
  - It has more standards than any other layer.
  - HTTP is not the only application layer standard; it is one of many.
  - Many application layer protocols, such as SMTP for e-mail, are much more complex than HTTP.

Hypertext Transfer Protocol Message Syntax

- HTTP Syntax Uses [CRLF]
  - This stands for **carriage return plus line feed**.
  - On printers, carriage moves the print head back to the left of the same line, and line feed moves the print head down a line.
  - In simpler terms, [CRLF] begins a new line.
  - Question: What will two [CRLFs] in a row do?

Hypertext Transfer Protocol Message Syntax

- HTTP Request Message:
  - GET /panko/home.htm HTTP/1.1[CRLF]
  - Host: voyager.shidler.hawaii.edu

- In the first line:
  - GET says that this is a request to get a file.
  - /panko/home.htm is the location of the file.
  - HTTP/1.1 says that the browser speaks HTTP/1.1.
- The second line specifies the host to receive this HTTP request message.

Hypertext Transfer Protocol Message Syntax

- In HTTP, most lines have this syntax:
  - Keyword: Value
  - Example: Host: voyager.shidler.hawaii.edu
  - This is like e-mail, with its To:, From:, and so on, fields.
  - HTTP was based on e-mail header concepts.
  - Consequently, it feels old-fashioned.

Hypertext Transfer Protocol Message Syntax

- HTTP Response Message
  - HTTP/1.1 200 OK[CRLF]
  - Date: Tuesday, 20-MAR-2011 18:32:15 GMT[CRLF]
  - Server: name of server software[CRLF]
  - MIME-version: 1.0[CRLF]
  - Content-type: text/plain[CRLF]
  - -[CRLF]
  - File to be downloaded. A string of bytes that may be text, graphics, sound, video, or other content.

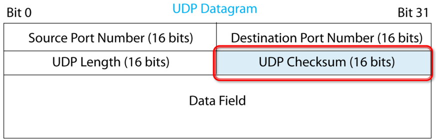
Perspective: Syntax

- We have spent much time on message syntax because it is extremely important in networking.
- Ethernet frames are drawn with fields one below the other.
- IP, TCP, and UDP syntax are drawn with fields beginning with bit position.
- HTTP header syntax is shown as a series of lines of text, most of which have the Keyword: Value format.

Encoding application messages

Encoding Application Messages

- Encoding
  - Applications must convert application message content into bits.
  - This is necessary because all lower layers have fields consisting only of ones and zeros.
  - This is called encoding.



Encoding Text

- Text often is encoded with 7-bit ASCII.
  - The eighth bit in each byte is unused.

Broad Function	Layer Number	Layer Name
Interoperability of application programs	5	Application
Transmission across an internet	4	Transport
	3	Internet
Transmission across a single switched or wireless network	2	Data Link
	1	Physical

Binary Representations of Whole Numbers

Basic Rules

				1
0	0	1	1	+1
+0	+1	+0	+1	+1
=0	=1	=1	=10	=11

Counting begins with 0, not 1. So the first three items are 0, 1, and 10 (10 is 2 in binary).

Examples

1000	8
+1	+1
=1001	=9
+1	+1
=1010	=10
+1	+1
=1011	=11
+1	+1
=1100	=12

Encoding Application Messages

- Encoding Alternatives
  - Some information consists of alternatives that have no particular order:
    - Gender (male or female): 2 alternatives
    - Sales region (north, south, east, west): 4 alternatives
    - City of birth: Many alternatives
- How many bits do you need to represent alternatives?
  - Must be encoded into a field of fixed length

Encoding Application Messages

- Encoding Alternatives
  - If a field is N bits long, it can represent 2^N alternatives

Field size (bits)	Possible # of alternatives	Example
1	$2^1=2$	Gender (male or female)
2	$2^2=4$	Direction (north, south, east, west)
3	$2^3=8$	The seven seas. One alternative is not used
4	$2^4=16$	Ten sales regions (3 bits is not enough)

Encoding Application Messages

- Each additional bit doubles the number of alternatives a field can represent.

Field size (bits)	Possible # of alternatives	Example
1	2^1=2	Gender (male or female)
2	2^2=4	Direction (north, south, east, west)
3	2^3=8	The seven seas. One alternative is not used
4	2^4=16	Ten sales regions (3 bits is not enough)

Encoding Application Messages

- Each additional bit doubles the number of alternatives a field can represent.

Example

- Give the number of possible alternatives for

1. 1 bit:
2. 2 bits:
3. 3 bits:
4. 4 bits:
5. 5 bits:
6. 6 bits:
7. 7 bits:
8. 8 bits:
9. 9 bits:
10. 10 bits:

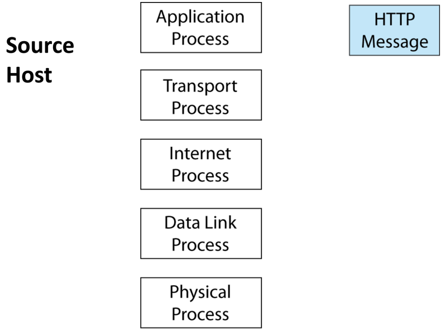
Encoding Application Messages

Example

1. What is the formula for the number of alternatives you can represent in a field?
2. A field is three bits long. How many alternatives can it represent?
3. A field is four bits long. How many alternatives can it represent?
4. If you want to encode the 12 months, how many bits will you need in the field?

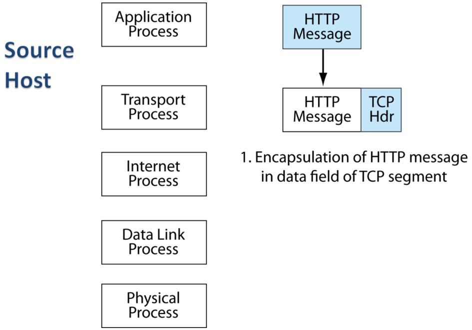
Vertical communication on the source host

Vertical communication on the source host



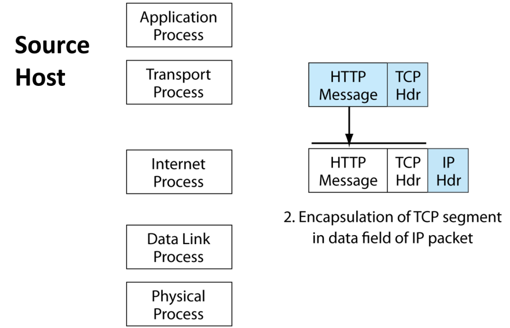
- The application process (in this case a browser) creates an application message for the application process on the destination host.
- In this case, the application process is a browser.

Vertical communication on the source host



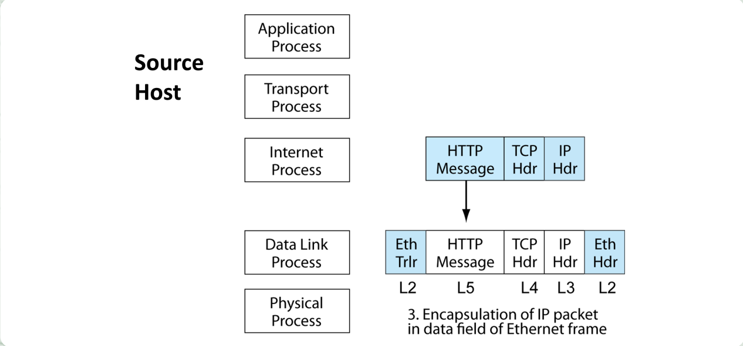
- The application process passes the HTTP message down to the transport layer process.
- The transport process encapsulates the HTTP message in the data field of TCP segment.
- This requires adding a TCP header.

Vertical communication on the source host



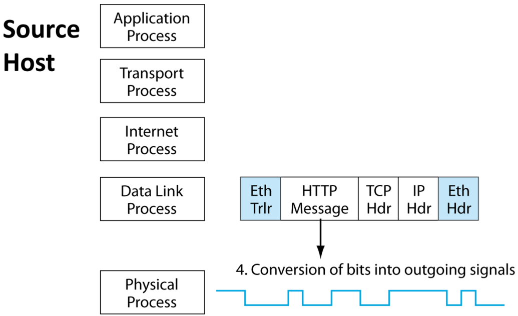
- The transport process passes the TCP segment down to the internet process, which encapsulates the segment in an IP packet.

Vertical communication on the source host



- The internet process passes the IP packet to the data link process, which is Ethernet.
- Ethernet adds a header and trailer.

Vertical communication on the source host



- The data link process passes the frame down to the physical layer.
- The physical layer does NOT do encapsulation. It merely converts the bits of the frame into signals.

Vertical communication on the source host

- The application, transport, internet, and data link layer process do the same thing.
  - Create a message in all but the originating layer (in this case, application), by encapsulation.
  - Then pass the message down to the next lower layer.
- The physical layer process is different.
  - It merely converts the bits of the frame into signals.

Standards architectures

Major Standards Architectures

- To build a house, you do not build one room, then decide what to do next.
- You create an architecture identifying the rooms that will be needed and how the rooms will relate to one another in terms of flow.
- Then you begin the design of individual rooms.

Major Standards Architectures

- In networking, standards agencies first create standards architectures that define the categories of standards to be developed.
- They define categories in terms of layers.
- They then design standards for the individual categories.

Major Standards Architectures

- There are two major standards architectures.
- TCP/IP
    - This is the standards architecture for the Internet and many corporate internets.
    - Its standards agency is the Internet Engineering Task Force (IETF).
    - As we saw in Chapter 1, most of its documents are called requests for comment (RFCs).
    - Some (but not all) RFCs are official Internet standards.

Major Standards Architectures

- OSI
  - Its two standards agency are ISO and the ITU-T.
  - ISO (the International Organization for Standardization) is a computer standards agency.
  - The ITU-T is the International Telecommunications Union—Transmission Standards Agency.
  - Don't confuse ISO the organization with OSI the architecture.

Major Standards Architectures

- OSI divides standards into seven layers:

Broad Purpose	OSI Layers
Application communication	Application (Layer 7)
	Presentation (Layer 6)
	Session (Layer 5)
Internetworking	Transport (Layer 4)
	Network (Layer 3)
Communications within a single LAN or WAN	Data Link (Layer 2)
	Physical (Layer 1)

Major Standards Architectures

- OSI divided application communication into three layers.
  - Session (5) creates a managed session between applications. If there is a break in communication, the communication only has to go back to the last roll-back point.
  - Presentation (6) was designed to translate between data formats on different computers. In practice, it is used as a category for file format standards, such as jpg and mp3.
  - Application (7) handles other details of application communication.

Major Standards Architectures

- TCP/IP divides standards into four layers.

Broad Purpose	TCP/IP
Applications	Application
Internetworking	Transport
	Internet
Communication within a single LAN or WAN	The IETF assumes that OSI standards will be used at the Physical and Data Link Layers

Hybrid TCP/IP-OSI Standards Architecture

- Companies actually use the Hybrid TCP/IP-OSI Standards Architecture that takes standards at different layers from the two architectures.

Layer	Name	Source
5	Application	TCP, OSI, and others
4	Transport	TCP/IP (TCP and UDP)
3	Internet	TCP/IP (IP)
2	Data Link	OSI (Ethernet)
1	Physical	OSI (Ethernet)

Major Standards Architectures

- Dominance
  - OSI is almost 100% dominant at the physical and data link layers.
  - It is so dominant that the IETF assumes that network users will use OSI standards at the physical and data link layers.
  - This means single switched or wireless LANs and WANs.

Major Standards Architectures

- Dominance
  - TCP/IP is about 90% dominant at the internet and transport layers, but other standards architectures are sometimes used at these layers.
    - IPX/SPX is used on older Novell Netware file servers.
    - SNA is used by many IBM mainframes.
    - AppleTalk is used by some Apple servers.

Major Standards Architectures

- Dominance
  - At the application layer, many standards come from TCP/IP, but many also come from OSI, the World Wide Web Consortium, and other standards agencies.

Recap: Characteristics of Protocols in this Chapter

Layer	Protocol	Connection-Oriented or Connectionless?	Reliable or Unreliable
5 (Application)	HTTP	Connectionless	Unreliable
4 (Transport)	TCP	Connection-Oriented	Reliable
4 (Transport)	UDP	Connectionless	Unreliable
3 (Internet)	IP	Connectionless	Unreliable
2 (Data Link)	Ethernet	Connectionless	Unreliable

Next Lecture

- Lecture 1: Core concepts and principles
- Lecture 2: Standards concepts
- Lecture 3: Physical propagation layer

84