# Network Standards

Chapter 2

Panko and Panko
Business Data Networks and Security, 10th Edition, Global Edition

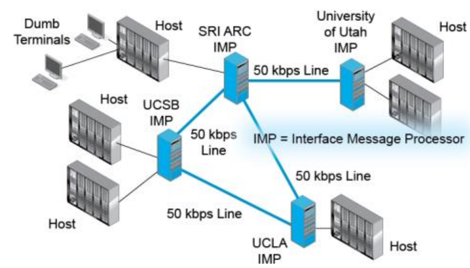





## "Standard" = "Protocol"?

- Not really, but Panko uses them nearly interchangeably.
- In this lecture we are really talking about protocols, not standards
- At low levels (L2-L4) they tend to be the same thing. And partly at level 5.
- We talked a lot about standards in previous lecture, and later we have a guest lecture on that topic.

## 2.1 The Early ARPANET (1969)






## The Network Working Group

- Bolt, Beranek, and Newman designed the IMPs
- But there were no standards for application programs or protocols for hosts to communicate with other hosts
- An informal group formed to create needed standards: the Network Working Group
- Mostly engineers and graduate students, the NWG had no formal standing



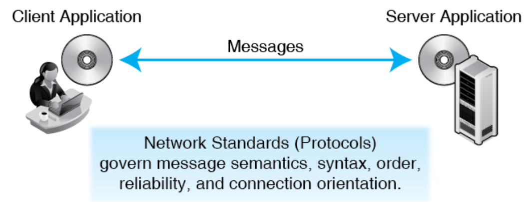
## The Network Working Group

- Called their "standards" Requests For Comments (RFCs) instead of standards
- Informal and egalitarian, working on rough consensus rather than voting
- Designed simple standards based on running code; quick and inexpensive to implement
- The Internet Engineering Task Force (IETF) emerged from the NWG, kept many of their characteristics

- How the Internet Came to Be
- Introduction
- Message Ordering
- Message Syntax
- Encoding Application Messages
- Vertical Communication on Hosts

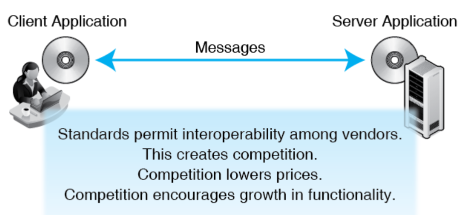


## 2.2 Network Standards



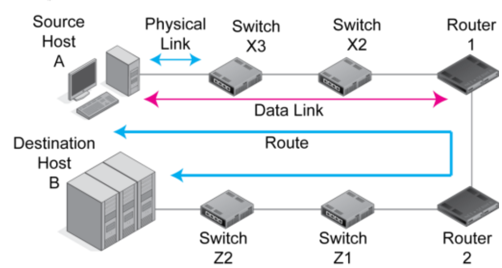



## 2.2 Network Standards



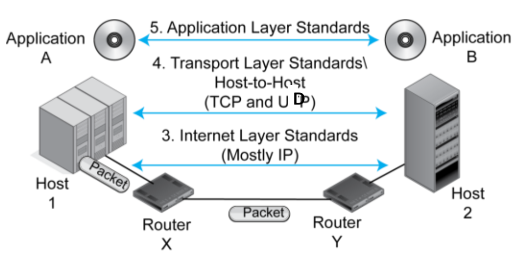



## 2.3 Physical, Data Link, and Internet Layers






## 2.4 Transport and Application Processes






## 2.5 Layers Recap

| Broad Function | Layer | Name | Specific Function |
|---|---|---|---|
| Interoperability of application programs | 5 | Application | Application layer standards govern how two applications work with each other, even if they are from different vendors. |

## 2.5 Layers Recap

| Broad Function | Layer | Name | Specific Function |
|---|---|---|---|
| Transmission across an internet | 4 | Transport | Aspects of end-to-end communication between two end hosts that are not handled by the internet layer. |
| | 3 | Internet | The transmission of packets across an internet—typically by sending them through several routers along the route. Also govern packet syntax and reliability. |



## 2.5 Layers Recap

| Broad Function | Layer | Name | Specific Function |
|---|---|---|---|
| Transmission across a single network | 2 | Data Link | Transmission of frames across a single network—typically by sending them through several switches along the data link. Also govern frame syntax and reliability. |
| | 1 | Physical | Transmission between adjacent devices connected by a transmission medium. |



- How the Internet Came to Be
- Introduction
- **Message Ordering**
- Message Syntax
- Encoding Application Messages
- Vertical Communication on Hosts



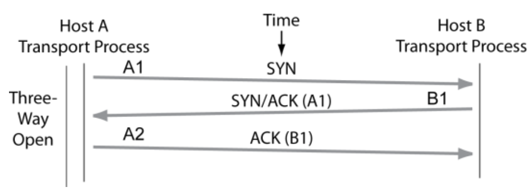
## 2.7 HTTP Request/Response Cycle



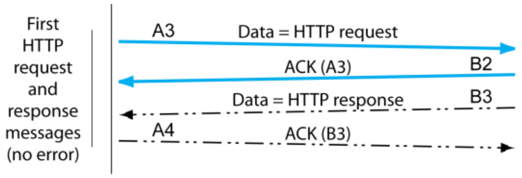

## 2.7 HTTP Request/Response Cycle




## 2.8 A TCP Session

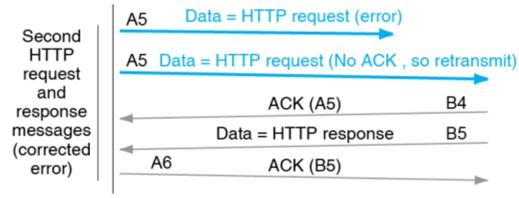


## 2.8 A TCP Session



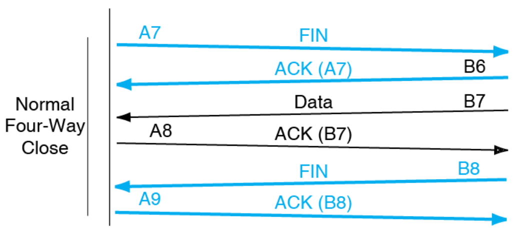



## 2.8 A TCP Session






## 2.8 A TCP Session






## Connection Orientation

**HTTP**

- Connectionless
- No openings, closings, ACKs
- Creates a light load on the network
- Unreliable
  - But TCP solves that!

**TCP**

- Connection-Oriented
- Openings, closings, ACKs place a heavy load on the network
- Reliable

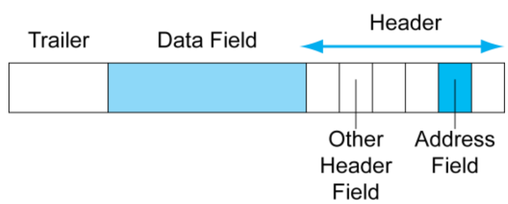




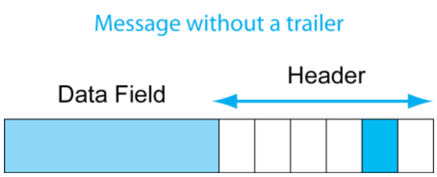


## 2.9 General Message Organization

Message with All Three Parts

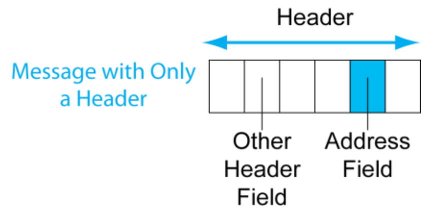


## 2.9 General Message Organization





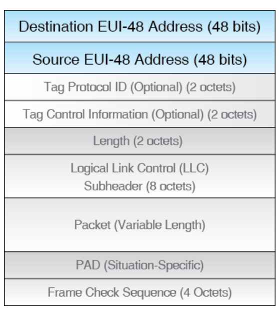

## 2.9 General Message Organization




The header may be partitioned into *fields* containing, e.g., the address



## 2.10 Ethernet Frame

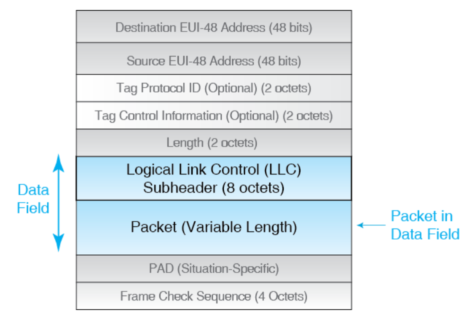



Destination and Source Data Link Layer Addresses (EUI-48). Formerly called a MAC Address

In Ethernet frame syntax, fields are listed in order, each is a certain number of octets (bytes) long

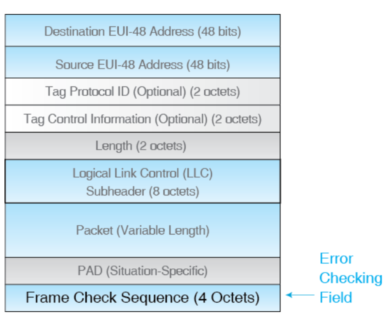


## 2.10 Ethernet Frame



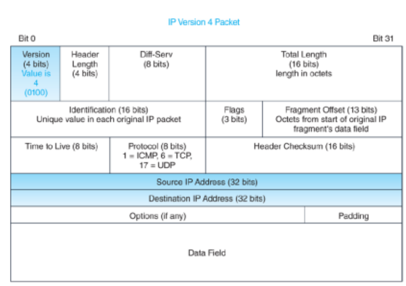



## 2.10 Ethernet Frame






## 2.11 Internet Protocol (IP) Packet in IPv4



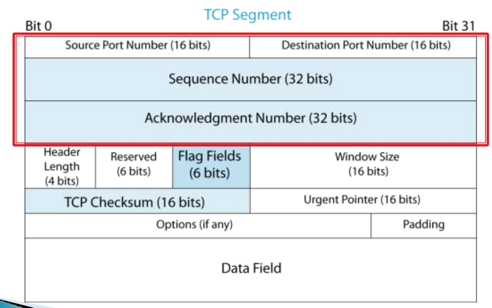

Header checksum is like an Ethernet Frame Check Sequence. If an error then discard (so an unreliable protocol)

## 1.2 IP Addresses in Dotted Decimal Notation

| Step | Action | Example | | | |
|---|---|---|---|---|---|
| 1 | Divide 32 bit IP address bits into four 8-bit segments. | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 |
| 2 | Convert each segment from binary to decimal. | 1 | 2 | 3 | 4 |
| 3 | Assemble the segments in decimal, separated by dots. | 1.2.3.4 | | | |

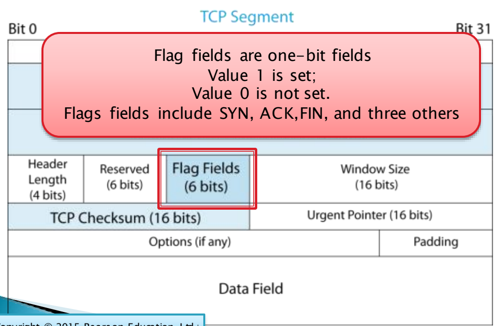


## 2.12 TCP Segment and UDP Datagram



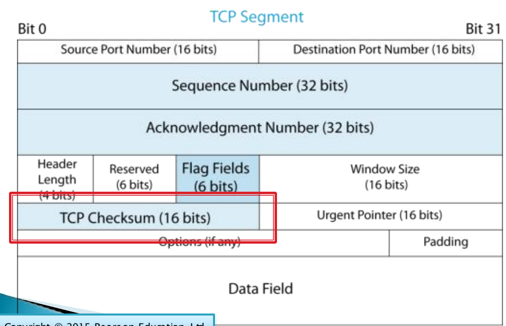



## 2.12 TCP Segment and UDP Datagram



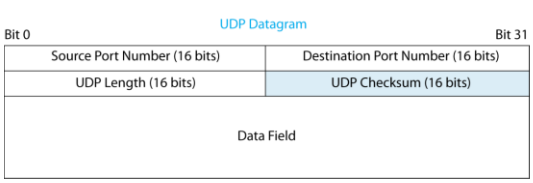



## 2.12 TCP Segment and UDP Datagram



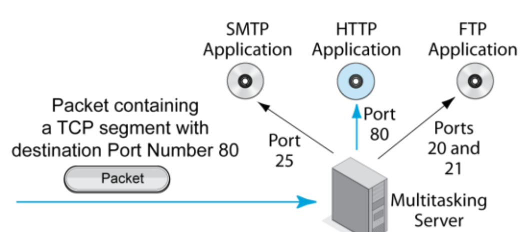



## 2.12 TCP Segment and UDP Datagram



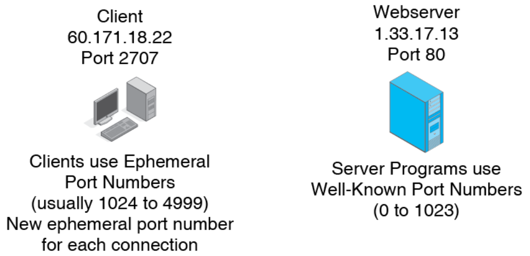



## 2.13 Server Port Numbers



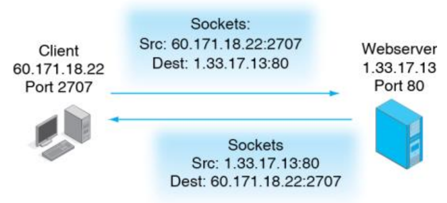

Multitasking servers run multiple applications. Each is associated with a port number.

## 2.14 Client Port Numbers and Sockets



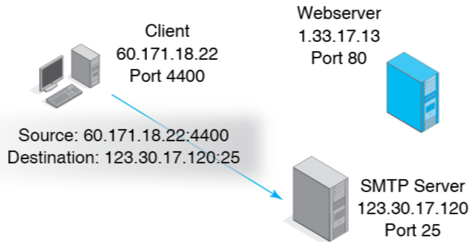



## 2.14 Client Port Numbers and Sockets



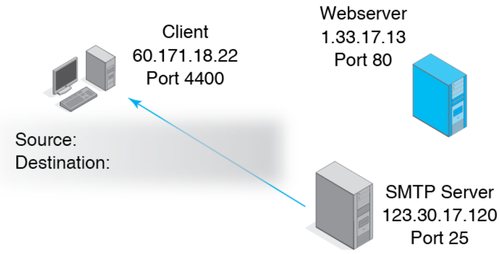



## 2.14 Client Port Numbers and Sockets






## 2.14 Client Port Numbers and Sockets






## 2.15 HTTP Message Syntax

- **HTTP**
  - The application layer is the highest layer
  - It has more standards than at 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



## 2.15 HTTP Message Syntax

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

## 2.15 HTTP Message Syntax

- HTTP Response Message
  - HTTP/1.1 200 OK[CRLF]
  - Date: Tuesday, 20-MAR-2014 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.



## 2.15 HTTP Message Syntax

- Notes
  - A relatively old feeling protocol *(really?)*
  - Fields ended by CRLF (2 characters), which starts a new line
  - Based on e-mail (an old protocol) for rapid development







## 2.16 Encoding Text as ASCII

| Category | Meaning | 7-Bit ASCII Code | 8th bit in Transmitted Byte |
|---|---|---|---|
| Upper-Case Letters | A | 1000001 | Unused |
| Lower-Case Letters | a | 1100001 | Unused |
| Digits (0 through 9) | 3 | 0110011 | Unused |
| Punctuation | Period | 0101110 | Unused |
| Punctuation | Space | 0100000 | Unused |
| Control Codes | Carriage Return | 0001101 | Unused |
| Control Codes | Line Field | 0001010 | Unused |



## 2.17 Converting Binary to Decimal

**Representing Decimal (Base 10) Numbers**

| Position Exponent | $10^4$ | $10^3$ | $10^2$ | $10^1$ | $10^0$ |
|---|---|---|---|---|---|
| Position Value in decimal | 10,000 | 1,000 | 100 | 10 | 1 |
| Decimal Number | | | 5 | 0 | 3 |
| Decimal Representation | 503 = 5*100 + 0*10+ 3*1 | | | | |



## 2.17 Converting Binary to Decimal

**Representing Binary(Base 2) Numbers**

| Position Exponent | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|
| Position Value in decimal | 16 | 8 | 4 | 2 | 1 |
| Binary Number | 0 | 1 | 0 | 1 | 0 |
| Decimal Equivalents | 0 | 8 | 0 | 2 | 0 |
| Decimal Representation | 1010 = 1*8 + 1*2 = 10 | | | | |

## 2.18: Encoding a Decimal Number in Binary

| Position / Value / Step | | | | | Remainder |
|---|---|---|---|---|---|
| Binary Position | 4 | 3 | 2 | 1 | |
| Binary Value | 8 | 4 | 2 | 1 | |
| Step 0: Decimal num = 11 | -- | -- | -- | -- | 11 |
| Step 1: 8 is largest digit that will fit into 11 | 1 | 0 | 0 | 0 | 11-8 = 3 |
| Step 2: 2 is the largest digit that will fit into 3 | 1 | 0 | 1 | 0 | 8-2 = 1 |
| Step 3: 1 is the largest digit that will fit into 1 | 1 | 0 | 1 | 1 | 1-1 = 0 (finished) |
| Final binary number | 1 | 0 | 1 | 1 | |



## 2.19 Binary Encoding for Number of Alternatives

| Bits in Field | Number of Alternatives that can be Encoded | Possible Bit Sequences | Examples |
|---|---|---|---|
| 1 | $2^1 = 2$ | 0, 1 | Yes or No, Male or Female, etc. |
| 2 | $2^2 = 4$ | 00, 01,10, 11 | North, South, East, West; Red, Green, Blue, Black |
| 4 | $2^4 = 16$ | 0000, 0001, 0010, … | Top 10 security threats. Three bits would only give 8 alternatives. (With 4 bits, 6 values go unused) |
| 8 | $2^8 = 256$ | 00000000, 00000001, … | One byte per color gives 256 possible colors levels. |

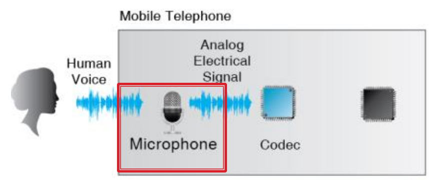


## 2.19 Binary Encoding for Number of Alternatives

| Bits in Field | Number of Alternatives that can be Encoded | Possible Bit Sequences | Examples |
|---|---|---|---|
| 16 | $2^{16} = 65,536$ | 0000000000000000, 0000000000000001, '… | Two bytes per color gives 65,536 color levels. |
| 32 | $2^{32} = 4,294,967,296$ | 000000000000000 0000000000000000, etc. | Number of Internet Protocol Version 4 addresses |



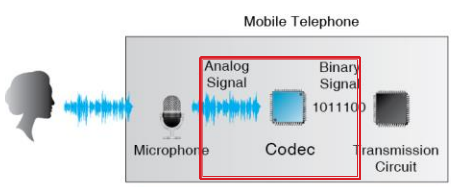
## 2.20 Encoding Voice




A microphone converts the human voice into analogous (analog) electrical signals.



## 2.20 Encoding Voice

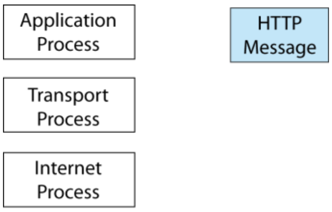



The codec converts between analog electrical signals and digital transmission signals.



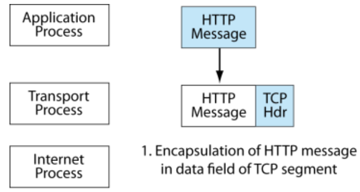
- How the Internet Came to Be
- Introduction
- Message Ordering
- Message Syntax
- Encoding Application Messages
- **Vertical Communication on Hosts**

## 2.21 Layered Communication on the Source Host



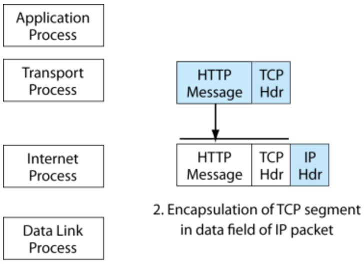



## 2.21 Layered Communication on the Source Host



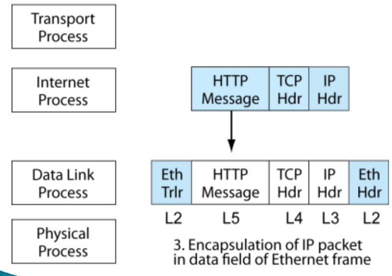



## 2.21 Layered Communication on the Source Host



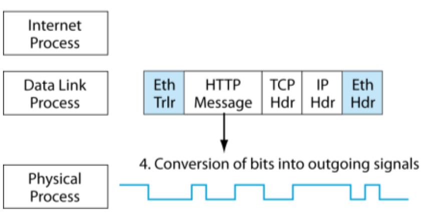



## 2.21 Layered Communication on the Source Host






## 2.21 Layered Communication on the Source Host

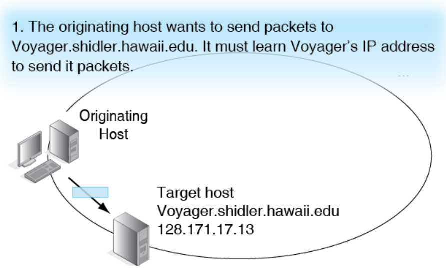





## DNS from Chapter 1 of Panko

- DNS is an application layer protocol that we need to cover now
- You use it all the time
- It relies on UDP (not TCP) at the transport layer

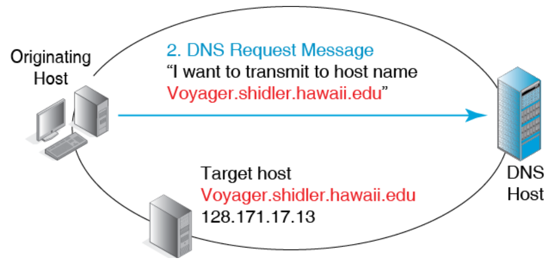
## 1.31 Domain Name System (DNS)






1-61

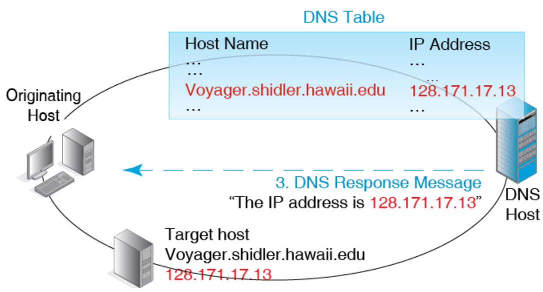
## 1.31 Domain Name System (DNS)






1-62

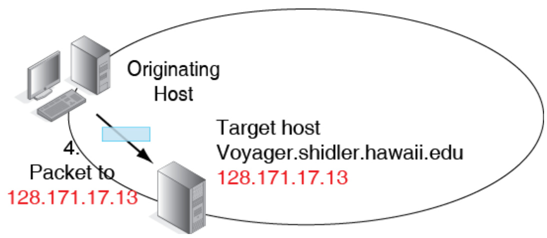
## 1.31 Domain Name System (DNS)






1-63

## 1.31 Domain Name System (DNS)






1-64

## 2.22 Protocols in This Slide Pack

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