CSCI 330
The UNIX System

Networking

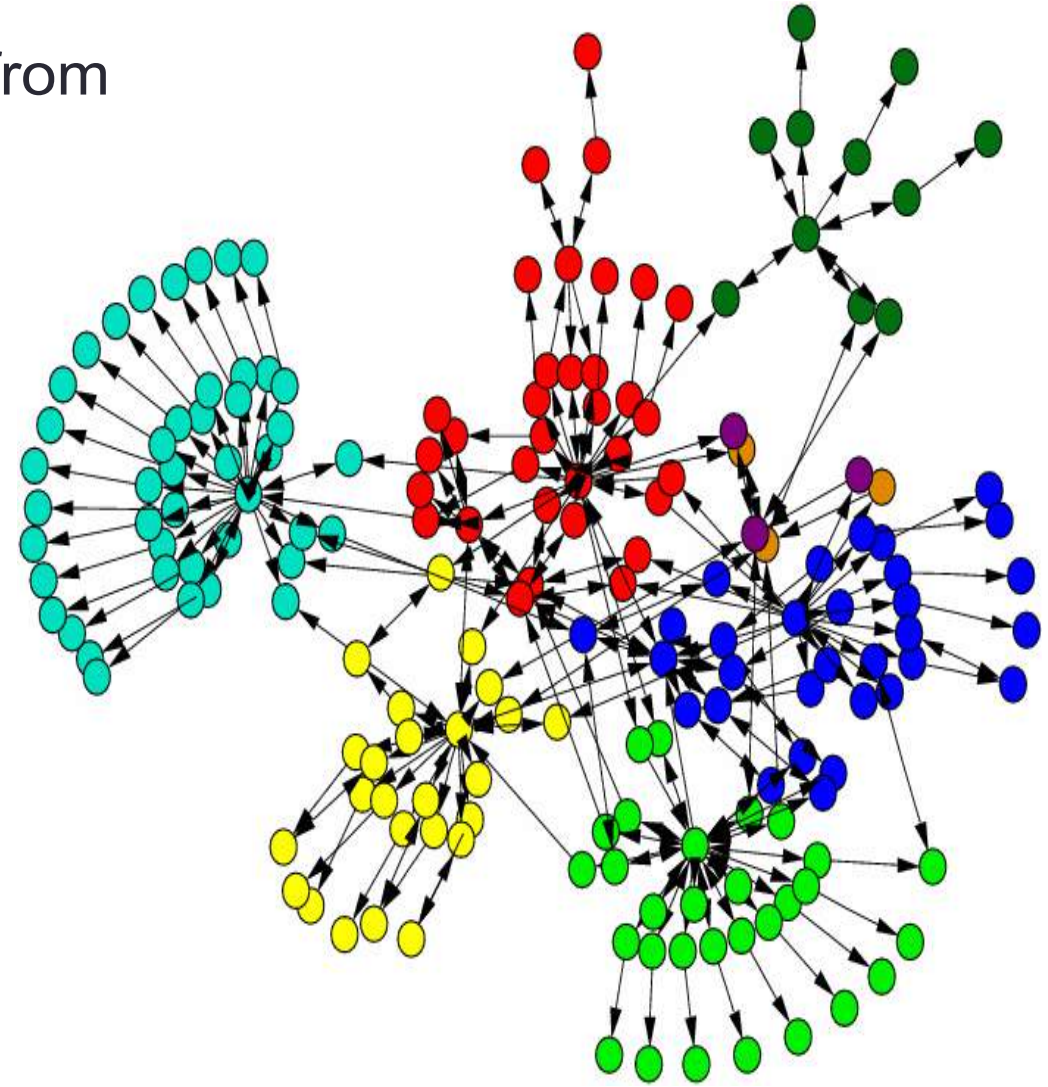
Unit Overview

- Network concepts & terminology
- OSI reference model for protocols
  - Physical layer
  - Data Link layer
  - Network layer
  - Transport layer

Network Terminology

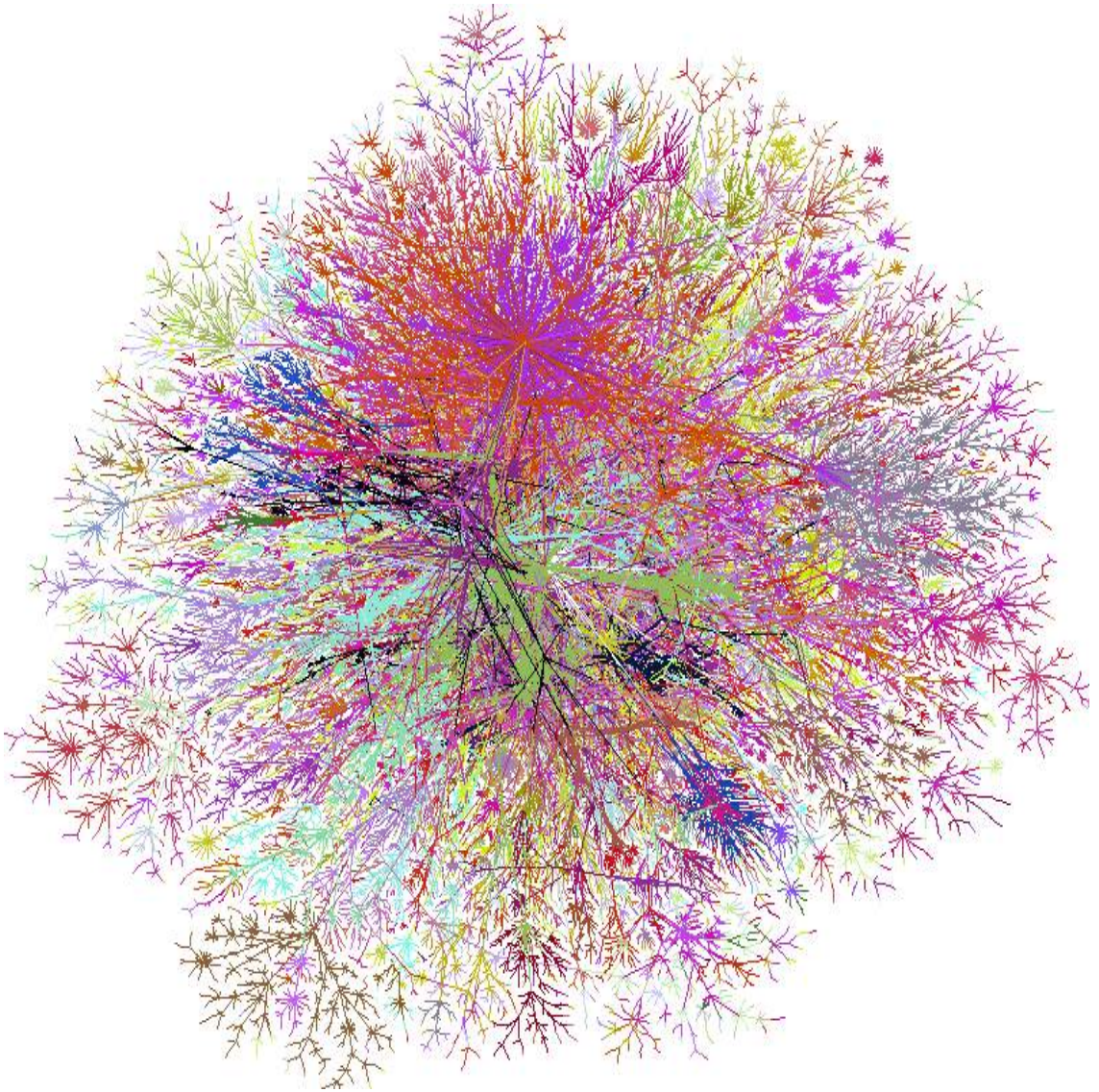
- connected graph constructed from
  - node
  - link
- nodes can reach others

  via path
  - sequence of nodes and links



Internet Terminology

- node
  - host or intermediary
- link
  - point-to-point or broadcast
- link medium
  - wired or wireless
- path
  - routed or switched



Networking Protocol

- communication in a network is governed by rules and conventions
- information is exchanged between nodes via messages
- messages use well-defined format
- each message has an exact meaning intended to provoke a defined response of the receiver

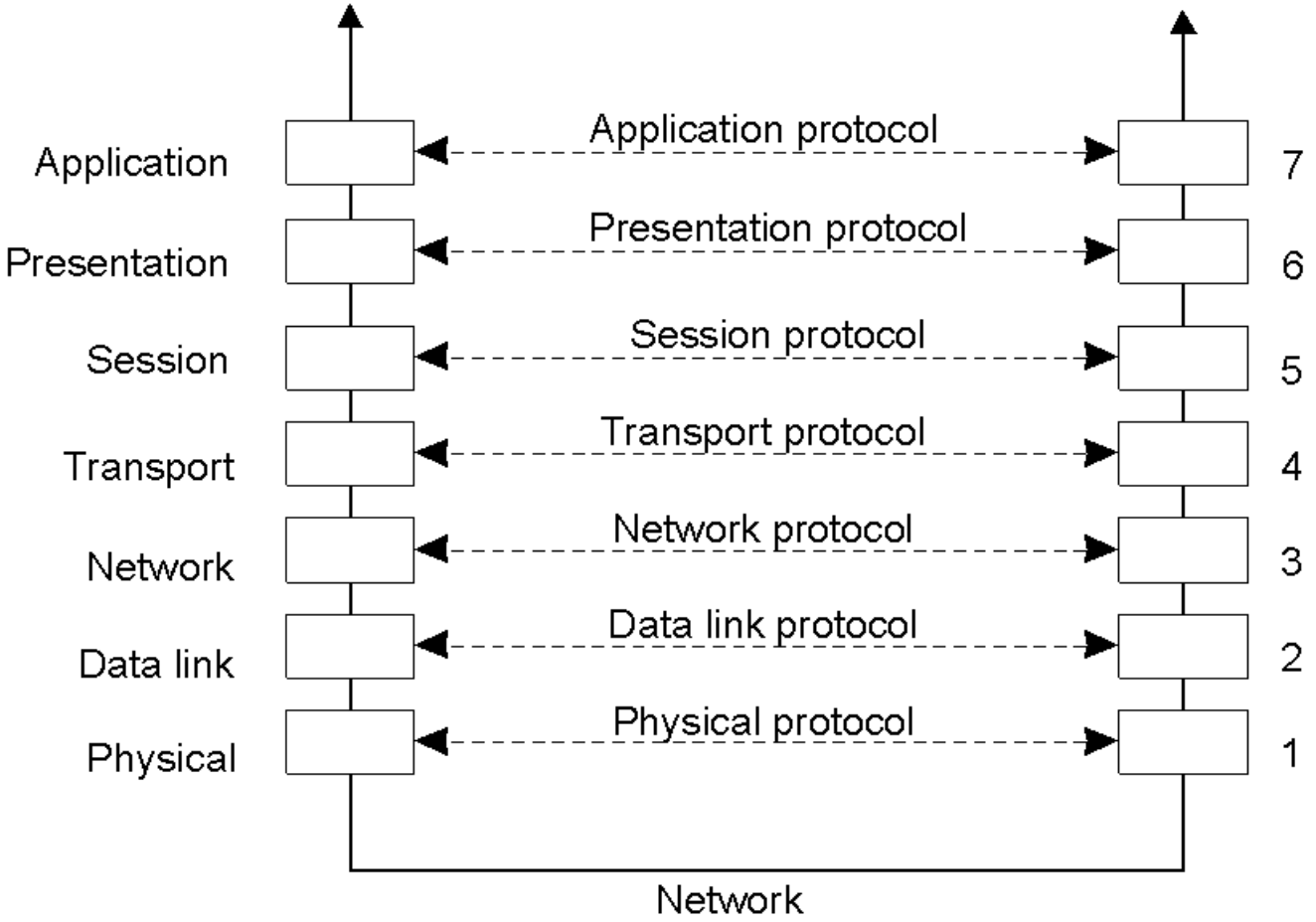
a protocol describes the syntax, semantics, and synchronization of communication

Layered protocols

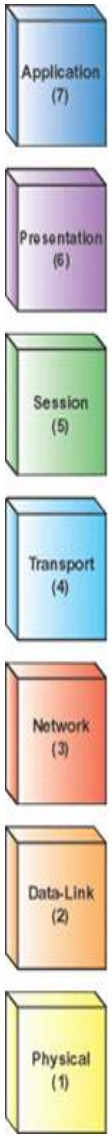
- complexities of communication organized into successive layers of protocols
  - lower-level layers: specific to medium
  - higher-level layers: specific to application
- standards achieve inter-operability

Open Systems Interconnection model (OSI reference model)

OSI reference model

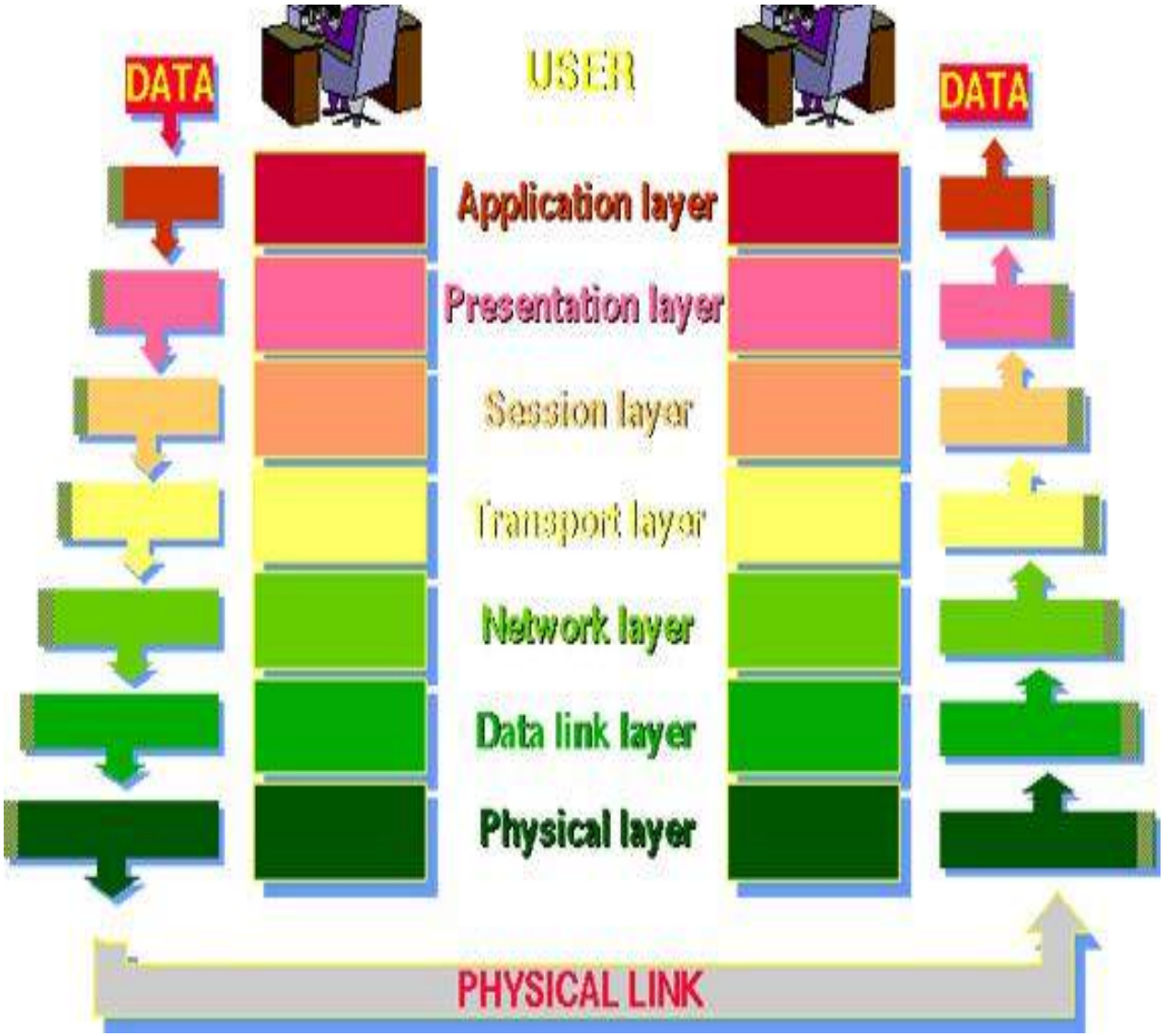


OSI reference model layers



- provides services directly to user applications
- performs data transformations to provide common interface for user applications
- establishes, manages and ends user connection
- provides functions to guarantee reliable network link
- establishes, maintains and terminates network connections
- ensures the reliability of link
- controls transmission of the raw bit stream over the medium

OSI reference model layers



Physical Layer: Wired Media

- Ethernet
  - 10BASE-T, 100BASE-TX, 1000BASE-T
  - 10GbE, 40GbE, 100GbE
- Business/backbone:
  - DS1(T1): 1.54Mbs to DS5: 400Mbs
  - OC-1: 50Mbs to OC-768: 40Gbs
- Last mile:
  - Modem
  - DSL
  - cable: DOCSIS
  - FiOS
  - up to 100Mbs ?

Physical Layer: Wireless Media

- Cellphone Data
  - 3G: EDGE, GPRS: 384Kbs
  - 3.5G HSPA+ up to 88Mbs
  - 4G LTE up to over 100Mbs
- Satellite
  - Wildblue: 12Mbs
  - HughesNet: 15Mbs
- WiFi: 802.11
  - up to 150Mbs & MIMO
  - new: “ac” up to 1Gbs
- WiMax: 802.16
  - up to 40Mbs
- WPAN
  - BlueTooth up to 2Mbs
  - NFC up to 423Kbs
  - ZigBee up to 256Kbs

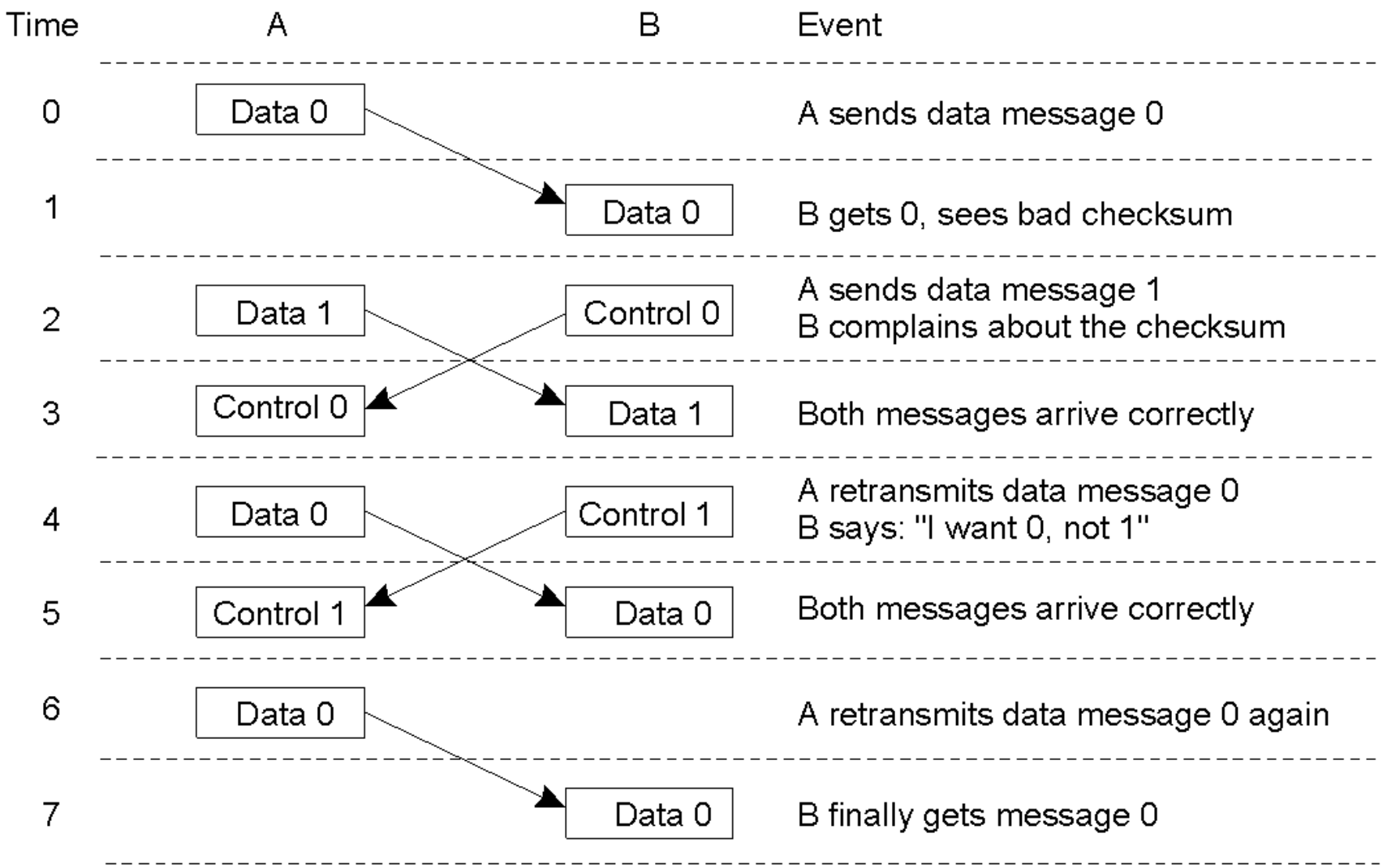
Data Link Layer: functionalities

- Medium access control
  - arbitrate who transmits
- Addressing
  - address of receiver, address of sender
- Framing
  - delimited unit of transmission for data & control
- Error control and reliability
- Flow control

Example: Ethernet frame

Preamble	Destination MAC address	Source MAC address	Type/ Length	User Data	Frame Check Sequence (FCS)
8	6	6	2	46 - 1500	4

Example: Data Link flow



Network Layer

- also called: Internet Protocol Layer
  - provides host to host transmission service,

    where hosts are not necessarily adjacent
- layer provides services:
  - addressing
  - hosts have global addresses: IPv4, IPv6
  - uses data link layer protocol to translate address: ARP
  - routing and forwarding
  - find path from host to host

IPv4 Address

- IP address
  - 32bit unique identifier, written as quad
- network
  - first n bits of IP number, written as “/n”
  - 8 - class A, 16 - class B, 24 - class C
  - more than 24 - class D
- netmask
  - 32 bit number with first n bits all 1, rest 0
- broadcast
  - network number (first n bits), rest all 1
- gateway IP
- name server IP

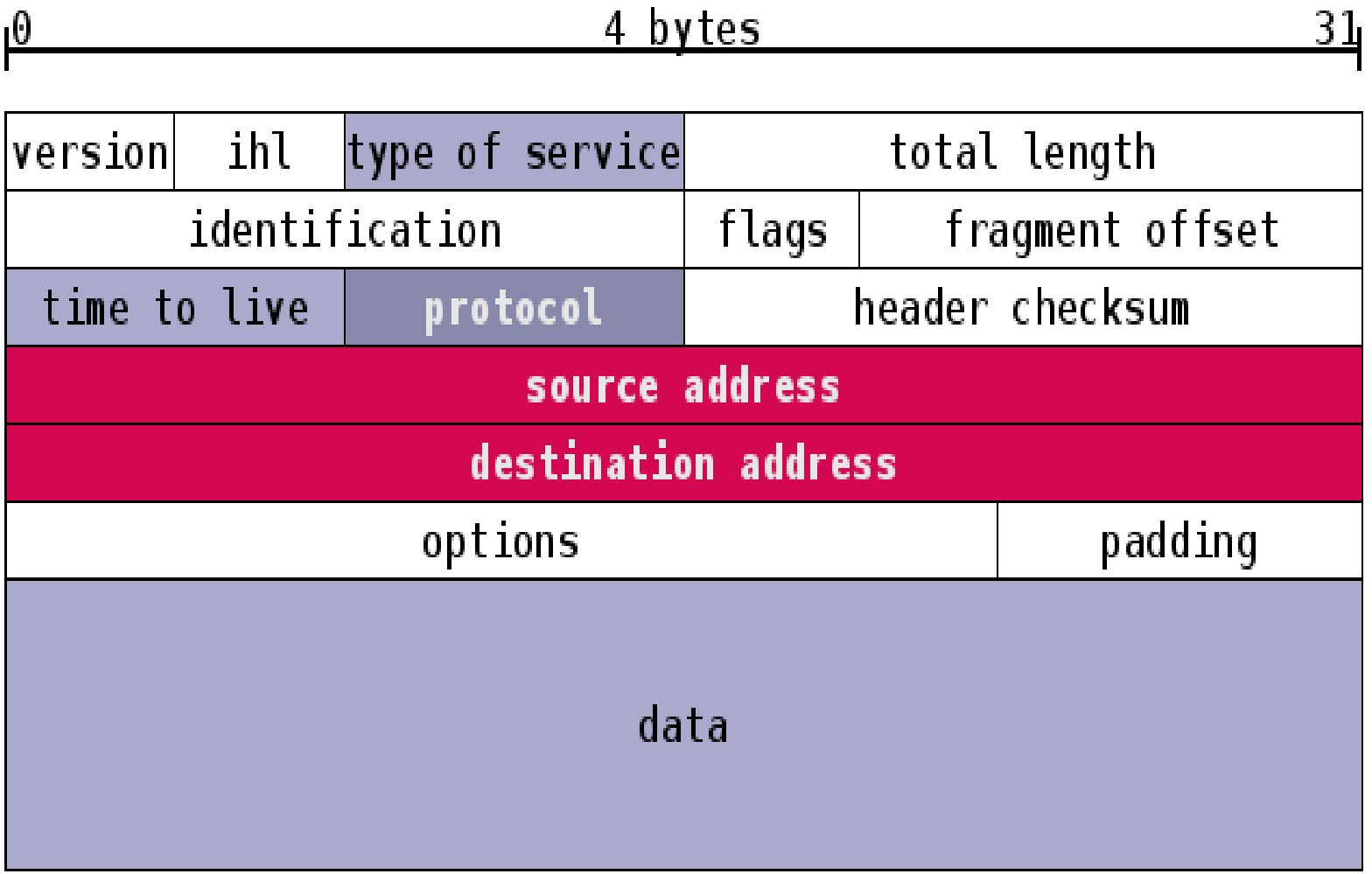
- 127.0.0.1
- 131.156.145.90
- 131.156.0.0/16
- 131.156.145.0/24
- 255.255.255.0
- 131.156.145.255
- 131.156.145.1
- 131.156.145.2

IPv6 Address

- IP address: 128-bit unique identifier
- 8 groups of 16-bit values,

  each group in 4 hex digits, separated by ":"
  - ex.: `2001:0db8:0000:0000:0000:ff00:0042:8329`
- can be abbreviated:
  - remove leading zeroes: `42` instead of `0042`
  - omit consecutive sections of zeroes:
    `2001:db8::ff00:42:8329`

Internet Protocol Packet



IP Layer: routing and forwarding

- done by hosts on path from sender to receiver
- forwarding:
  - host has 2 network interfaces
  - transfers packet from incoming to outgoing interface
- routing:
  - finds path from sender to receiver
  - simple routing: know receiver or send to gateway
  - advanced routing: determine which gateway to send to

    (typically with multiple outgoing network interfaces)

Transport Layer

- provides end-to-end communication services for applications
- byte format as abstraction on underlying system format
- raises reliability
- enables multiplexing:
  - provides multiple endpoints on a single node: port
  - refines connection address via port number

Transport layer ports

- 0 to 1023: well-known ports
  - 20 & 21: File Transfer Protocol (FTP)
  - 22: Secure Shell (SSH)
  - 23: Telnet remote login service
  - 25: Simple Mail Transfer Protocol (SMTP)
  - 53: Domain Name System (DNS) service
  - 80: Hypertext Transfer Protocol (HTTP) used in the World Wide Web
  - 110: Post Office Protocol (POP3)
  - 119: Network News Transfer Protocol (NNTP)
  - 143: Internet Message Access Protocol (IMAP)
  - 161: Simple Network Management Protocol (SNMP)
  - 443: HTTP Secure (HTTPS)
- 1024 to 49151: IANA registered ports
- 49152 to 65535: dynamic or private port

Transport layer programming

- common abstraction: socket
- first introduced in BSD Unix in 1981

- socket is end-point of communication link
  - identified as IP address + port number
- operates as client and server

Transport layer protocols

- TCP: transmission control protocol
  - connection oriented, guaranteed delivery
  - stream oriented: basis for: http, ftp, smtp, ssh
- UDP: user datagram protocol
  - best effort
  - datagram oriented: basis for: dns, rtp
- DCCP: datagram congestion control protocol
- SCTP: stream control transmission protocol

Domain Names

- hierarchical distributed naming system
- uses FQDN: fully qualified domain name
  - ex: faculty.cs.niu.edu
- DNS: domain name service
  - resolves query for FQDN into IP address
  - ex.: 131.156.145.186

Endianness

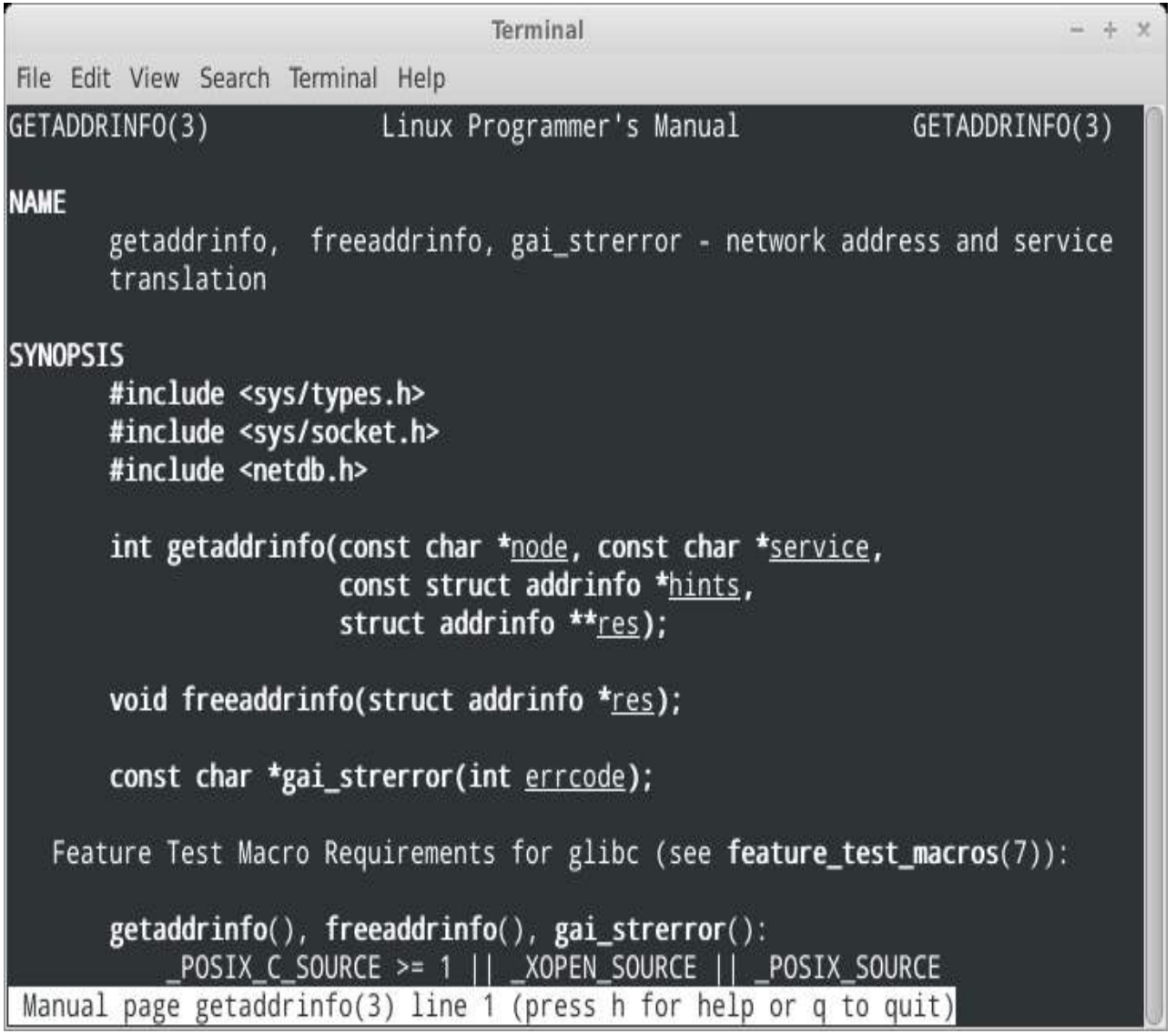
- How do you deal with data types that are larger than 1 byte?
- Big-endian – high order bytes come first
  - Motorola, IBM
- Little-endian – low order bytes come first
  - Intel architecture

- Network order – big-endian

Network conversion

- Converting values between network and host format
  - **`ntohl()`**, **`htonl()`** – convert 32 bit values
  - **`ntohs()`**, **`htons()`** - convert 16 bit values

Library Function: getaddrinfo



Library Function: getaddrinfo

```
int getaddrinfo(const char *node,
      const char *service,
            const struct addrinfo *hints,
                  struct addrinfo **res)
```

- translates FQDN **`node`** into IP address
- **`res`** is pointer to list of address info structures
- **`service`** and **`hints`** can be NULL

Address info structure

```
struct addrinfo {
  int              ai_flags;
  int              ai_family;
  int              ai_socktype;
  int              ai_protocol;
  size_t           ai_addrlen;
  struct sockaddr *ai_addr;      // socket address
  char            *ai_canonname;
  struct addrinfo *ai_next;
};
```

Socket address info structure

```
struct sockaddr_in {
    short           sin_family;   // e.g. AF_INET
    unsigned short  sin_port;     // port
    struct in_addr  sin_addr;
    char            sin_zero[8];  // padding
};

struct in_addr {
    unsigned long s_addr;         // for inet_pton()
};
```

IP Address / printing conversion

```
const char * inet_ntop(int af, const void * src,
                       char * dst, socklen_t size);
```

- converts source **in_addr** to a printed from in **dst** buffer of given size. Returns same string.

```
int inet_pton(int af, const char * src,
                      void * dst);
```

- converts source string into **in_addr** in **dst**.

af in both functions is **AF_INET** for IPv4 or **AF_INET6** for IPv6.

Example: getaddrinfo

```
int main(int argc, char * argv[]) {
  struct addrinfo *res;
  int error;
  const char * hostname = "hopper.cs.niu.edu";
  char hostIP[32];

  if(argc > 1) hostname = argv[1];

  error = getaddrinfo(hostname, NULL, NULL, &res);
  if (error) {
       cerr << hostname << ": " << gai_strerror(error) << endl;
       exit(EXIT_FAILURE);
  }

  // convert generic sockaddr to Internet sockaddr_in
  struct sockaddr_in * addr = (struct sockaddr_in *) res->ai_addr;
  // convert network representation into printable representation
  cout << hostname << " is " <<
       inet_ntop(AF_INET, &(addr->sin_addr), hostIP, 32) << "\n";

  // Cleanup
  freeaddrinfo(res);
}
```

Summary: Address translations

- MAC address

ARP RARP address resolution protocol

- IP address

DNS domain name service

- domain name

Summary: OSI model protocols

4. Transport Layer

TCP · UDP · SCTP · DCCP

3. Network Layer

IP · ICMP · IPsec · IGMP · IPX · AppleTalk

2. Data Link Layer

ARP · CSLIP · SLIP · Ethernet · Frame relay · ITU-T G.hn DLL · L2TP · PPP · PPTP

1. Physical Layer

RS-232 · RS-449 · V.35 · V.34 · I.430 · I.431 · T1 · E1 · POTS · SONET/SDH · OTN · DSL · 802.11a/b/g/n PHY · ITU-T G.hn PHY · Ethernet · USB · Bluetooth

7. Application Layer

NNTP · SIP · SSI · DNS · FTP · Gopher · HTTP · NFS · NTP · SMPP · SMTP · DHCP · SNMP · Telnet · (more)

6. Presentation Layer

MIME · XDR · TLS · SSL

5. Session Layer

Named Pipes · NetBIOS · SAP · SIP

Summary

- Network concepts & terminology
- OSI reference model for protocols
  - Physical layer
  - Data Link layer
  - Network layer
  - Transport layer