
2

# CS 348 Computer Networks

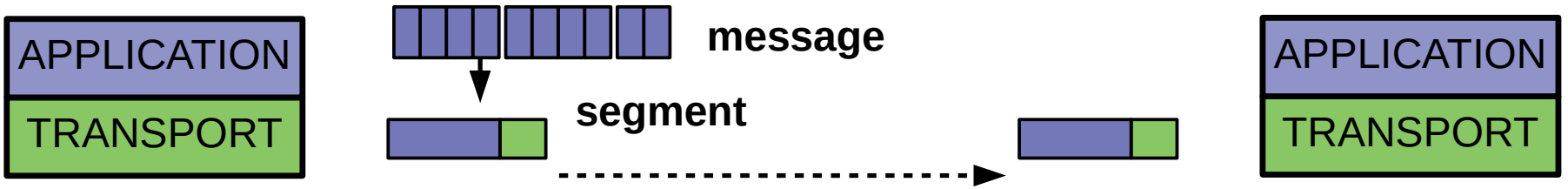
Spring 2019

Instructor: Neha Karanjkar

# Application Layer


- The layers below provide an abstraction of a **<reliable> bi-directional byte-stream** to the applications communicating over the Internet
- Interpretation of this byte-stream is application-specific
- Examples of Applications: Web-broswers, BitTorrent, Email, File Transfer, Multiplayer-games, Skype ....
- Examples of Protocols: HTTP (Web), SMTP (Email), FTP (File Transfer), SSH (Secure Shell), MQTT (used by IoT applications),....

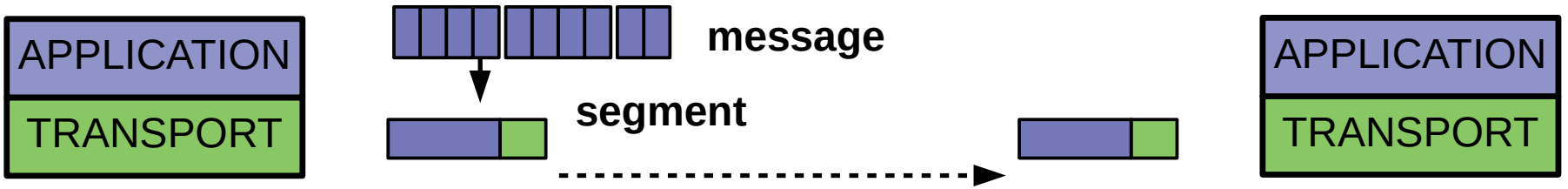
# Transport Layer




- Protocols:
  - **TCP (Transmission Control Protocol):** Connection Oriented, Reliable, requires Acknowledgements
  - **UDP** (User Datagram Protocol): Connectionless, No Acknowledgements
- **TCP : Provides the notion of a "flow" to the application layer.**
  - Reliable (receiver needs to send acks)
  - Re-ordering of segments, Detection of missing segments
  - Error-checking (using a checksum)
  - Flow control (prevent overwhelming the receiver)
  - Congestion control

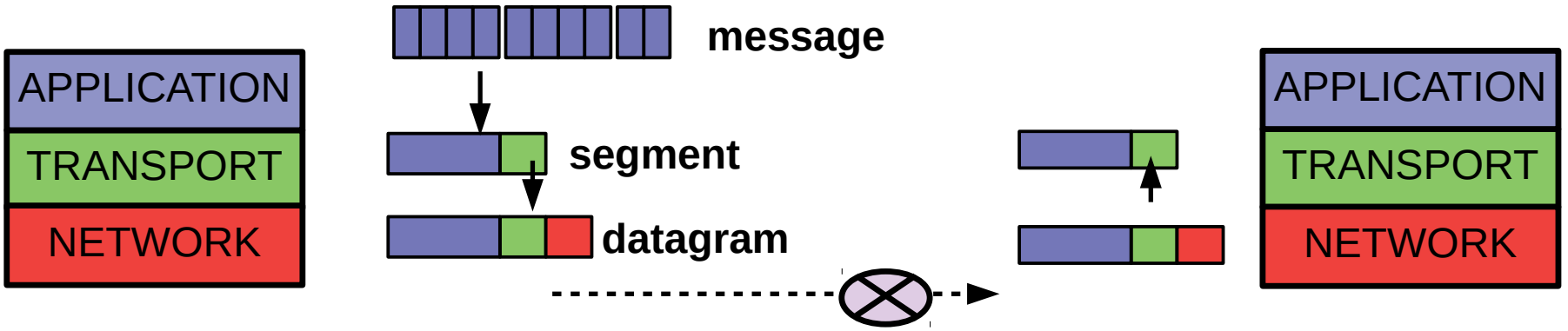
    The source/destination each maintain a **State Machine** per connection

# Transport Layer




- A TCP Header contains the following fields
  - Source **port number**, Destination **port number**
  - Sequence number (id of the first Byte in the segment)
    - useful at the receiver for re-sequencing and detecting missing segments
  - Ack, and ack sequence number (id of the next Byte expected)
  - Checksum

# Network Layer (IP)




- **Connectionless:**
  - Each datagram routed individually, no per-flow state
  - Packets can be delivered out-of-order
- **Unreliable, best-effort**
  - Packets can be dropped
- Prevents packets from looping forever (using a TTL field in the header: Time-To-Live)
- Fragments packets if too long, as per link-layer constraints
- Header checksum

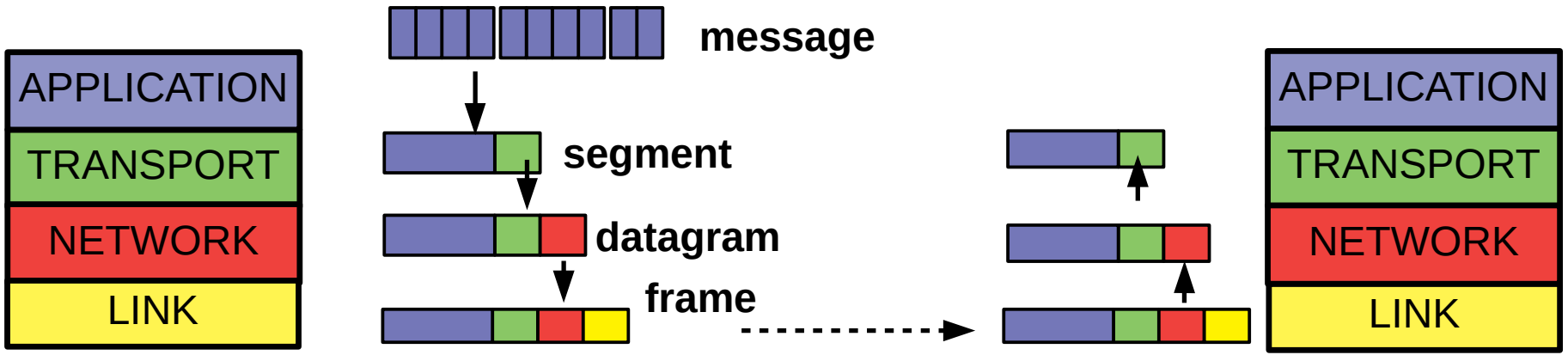
# Network Layer Header (IPv4)

**IPv4 Header Format**

| Octet | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0 | Version | | | | IHL | | | | DSCP | | | | | | ECN | | Total Length | | | | | | | | | | | | | | | |
| 32 | Identification | | | | | | | | | | | | | | | | Flags | | | Fragment Offset | | | | | | | | | | | |
| 64 | Time To Live | | | | | | | | Protocol | | | | | | | | Header Checksum | | | | | | | | | | | | | | | |
| 96 | Source IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 128 | Destination IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 160 | Options (if IHL > 5) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 192 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 224 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 256 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

- Image source: Wikipedia

# Link Layer




- **Deliver a datagram (encapsulated inside a frame) over a single link**
- Examples of link layer standards:  Ethernet (IEEE 802.3), WiFi (IEEE 802.11)
- Control/collision avoidance in a shared medium
- Frame synchronization
- Some link-layer protocols provide basic error detection and recovery (using checksums)

# Addressing

# Addressing

| | |
|---|---|
| APPLICATION | Application-specific addresses. Examples: URL (for web-browsing), Skype ID, torrent file, email ID .... |
| TRANSPORT | **Port Number**. 16b |
| NETWORK | **IP address**. 32b (for IPv4). 4 octets in dec. Example: 192.168.5.3 |
| LINK | **Physical (MAC) address**. 48b (for Ethernet, WiFi). Written as 6 octets in hex. Example: 01:AF:34:93:12:2E |

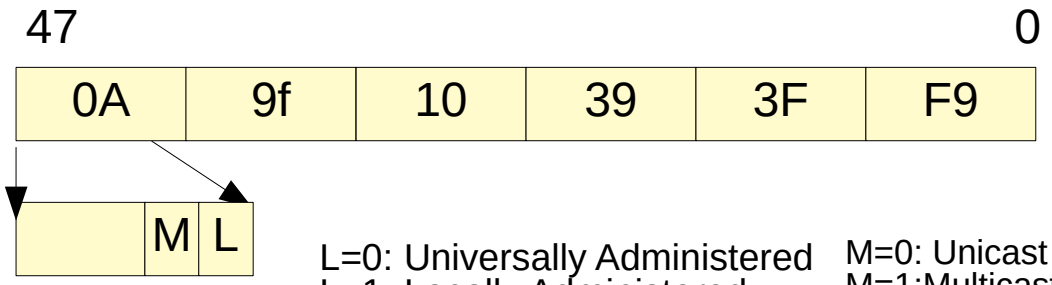
# Physical (MAC) Address

| | |
|---|---|
| LINK | **Physical (MAC) address**. 48b (for Ethernet, WiFi).<br>Written as 6 octets in hex. Example: 01:AF:34:93:12:2E |

- **A unique identifier assigned to a NIC (Network Interface Card)**
- Universally Administered Address: Assigned by manufacturer of NIC
- Locally Administered Address: Set by sysad by overriding the universal address
- Can be spoofed!

# Physical (MAC) Address

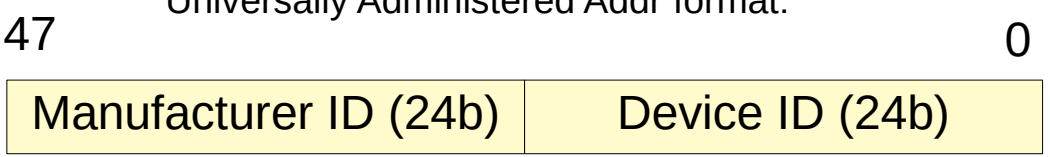
| LINK | **Physical (MAC) address**. 48b (for Ethernet, WiFi).<br>Written as 6 octets in hex. Example: 01:AF:34:93:12:2E |
|---|---|




L=0: Universally Administered
L=1: Locally Administered

M=0: Unicast
M=1:Multicast
(special case:
FF:FF:FF:FF:FF:FF is broadcast)

Universally Administered Addr format:



# IP Address

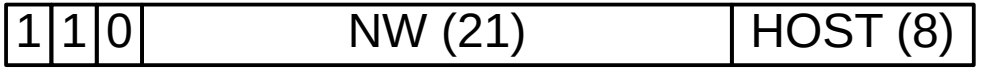
| NETWORK | **IP address**. 32b (for IPv4). 4 octets in dec. Example: 192.168.5.3 |
|---|---|

- **Herarchical: contain a network ID and a host ID**
- **Historically,** addresses were organized into 3 **classes:**

**Class A:**


**Class B:**


**Class C:**




- **Issues:** Coarse granularity, wastful use of address space

# IP Address

| NETWORK | **IP address**. 32b (for IPv4). 4 octets in dec. Example: 192.168.5.3 |
|---|---|

- **Current: Classless Inter-Domain Routing (CIDR)**
- Network-size can be any power of 2
- Notation example:
  - Original notation:  IP: 10.107.1.4,  Netmask:255.255.128.0
  - CIDR notation: 10.107.1.4**/17<-----prefix length**
- Who assigns IP addresses?
  - IANA (Internet Assigned Numbers Authority) specifies /8 addresses for  regions such as Asia-Pacific, Europe, etc
  - Regional Internet Registeries---for each region

# IP Address

| NETWORK | **IP address**. 32b (for IPv4). 4 octets in dec. Example: 192.168.5.3 |
|---|---|

- **Routers, Switches and Local Networks (Intro)**
- Address Resolution Protocol (ARP) (Intro)

# IP Address

| NETWORK | **IP address**. 32b (for IPv4). 4 octets in dec. Example: 192.168.5.3 |
|---|---|

- **Routers, Switches and Local Networks (Intro)**
- Address Resolution Protocol (ARP) (Intro)
- **Q:** If MAC addresses are unique, why do we need IP addresses?
- **Q:** If IP addresses are unique, why do we need MAC addresses?