



CS 348

Computer Networks

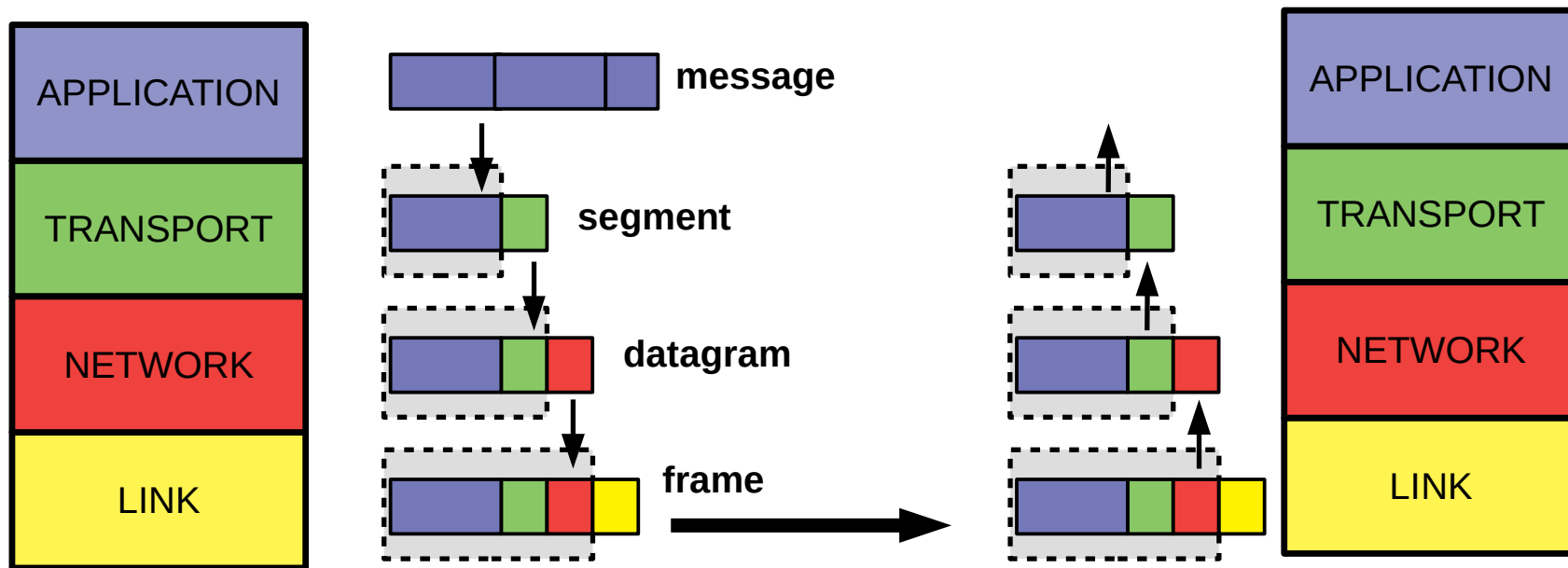
Lec 14

Spring 2020 IIT Goa

Course Instructor: Dr. Neha Karanjkar

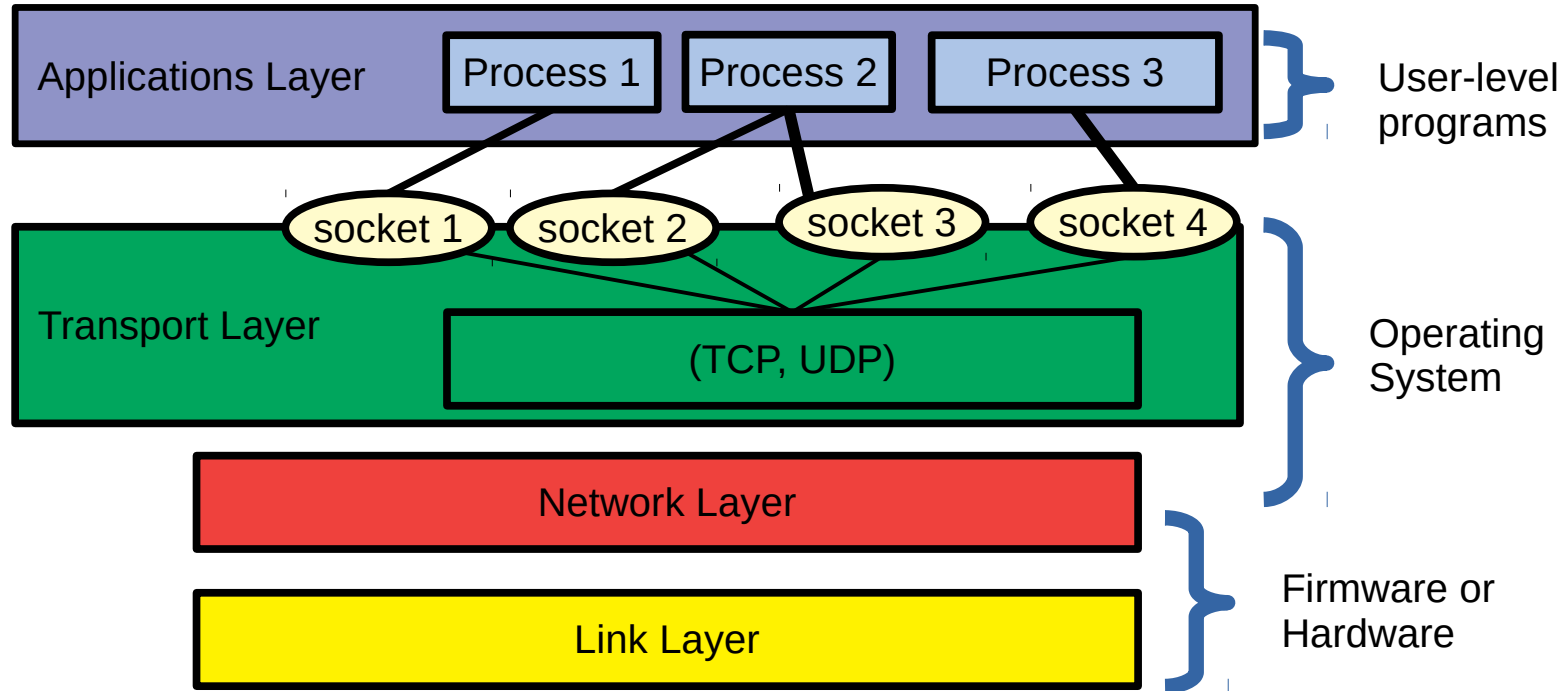
Note: These slides are adapted from “Computer Networking: A Top-down Approach” by Kurose & Ross, 7th ed

Recall: The TCP/IP Model



Recall: Sockets

- Sockets serve as an Interface between Applications and the Transport layer



Recall: Port Numbers

- A port number is a 16-bit identifier for sockets
- 0-1023 are “well-known” port numbers, reserved for well-known server-side applications. Examples:
 - 80: HTTP
 - 53: DNS
 - 443: HTTPS

PORT SCANNING

Port scanning tools (such as nmap) can be used for inferring what applications are running on a remote host
Used by admins as well as attackers.

https://en.wikipedia.org/wiki/Port_scanner

Service Models

- **Network Layer:** Best-effort **host-to-host** delivery of **a single datagram**

NW layer: “Give me a payload and a destination IP address. I will create a single datagram containing the payload and make the best effort to deliver it to the destination, but can offer no guarantee of delivery (the packet can get corrupted or lost).”

- **Transport Layer:** **Process-to-process** message delivery service for applications, which can be reliable and in-order (TCP) or best-effort (UDP)

TCP: “Give me a message (as a sequence of Bytes) and a <dest IP, dest port> address, and I will deliver them reliably and in-order, to the correct process (socket) at the destination host.”

UDP: “Give me a message and a <dest IP, dest port> address, and I will make a best-effort delivery of that message as a single packet to the correct process (socket) at the destination.

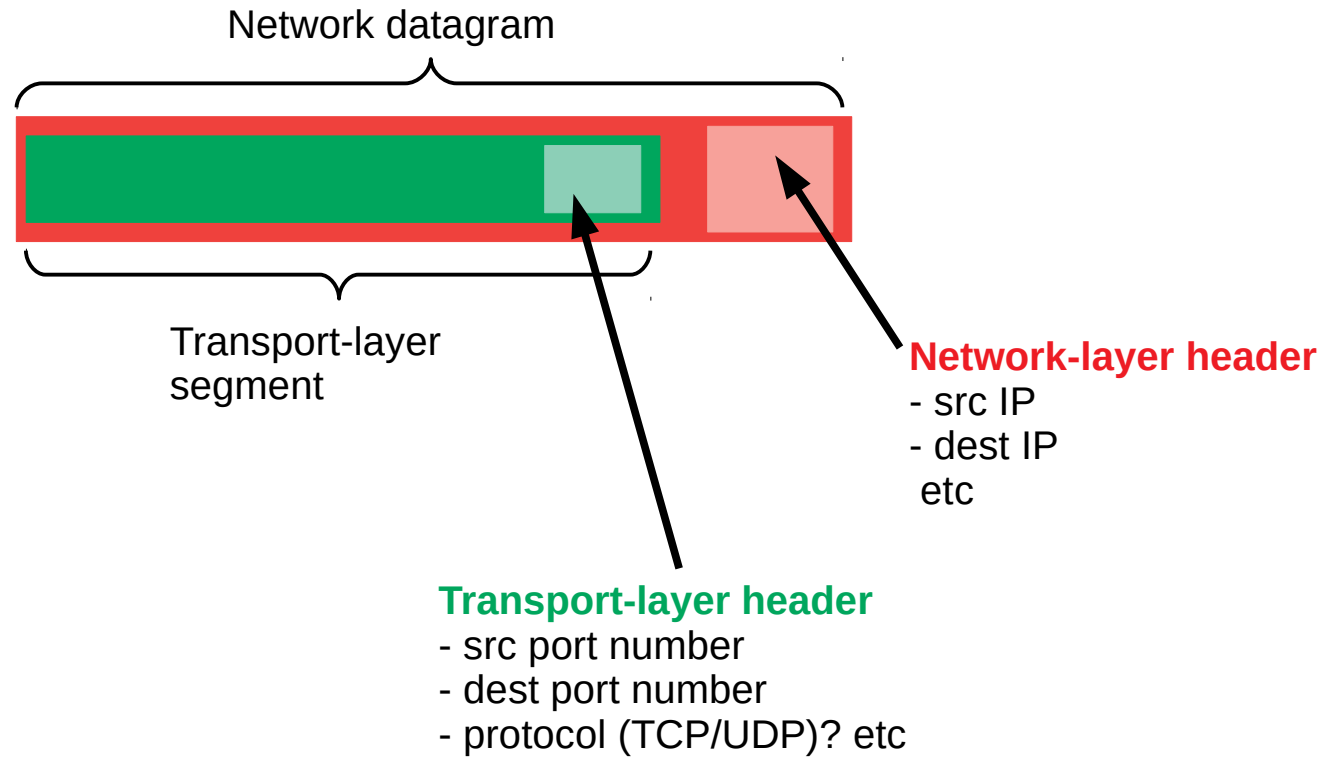
Questions

- What other services are offered by the Transport layer?
- UDP doesn't seem to do much on top of the service provided by the Network layer. Why do we need UDP at all?
- How does TCP achieve reliable data transfer using the unreliable delivery service provided by the Network layer?

The Transport Layer

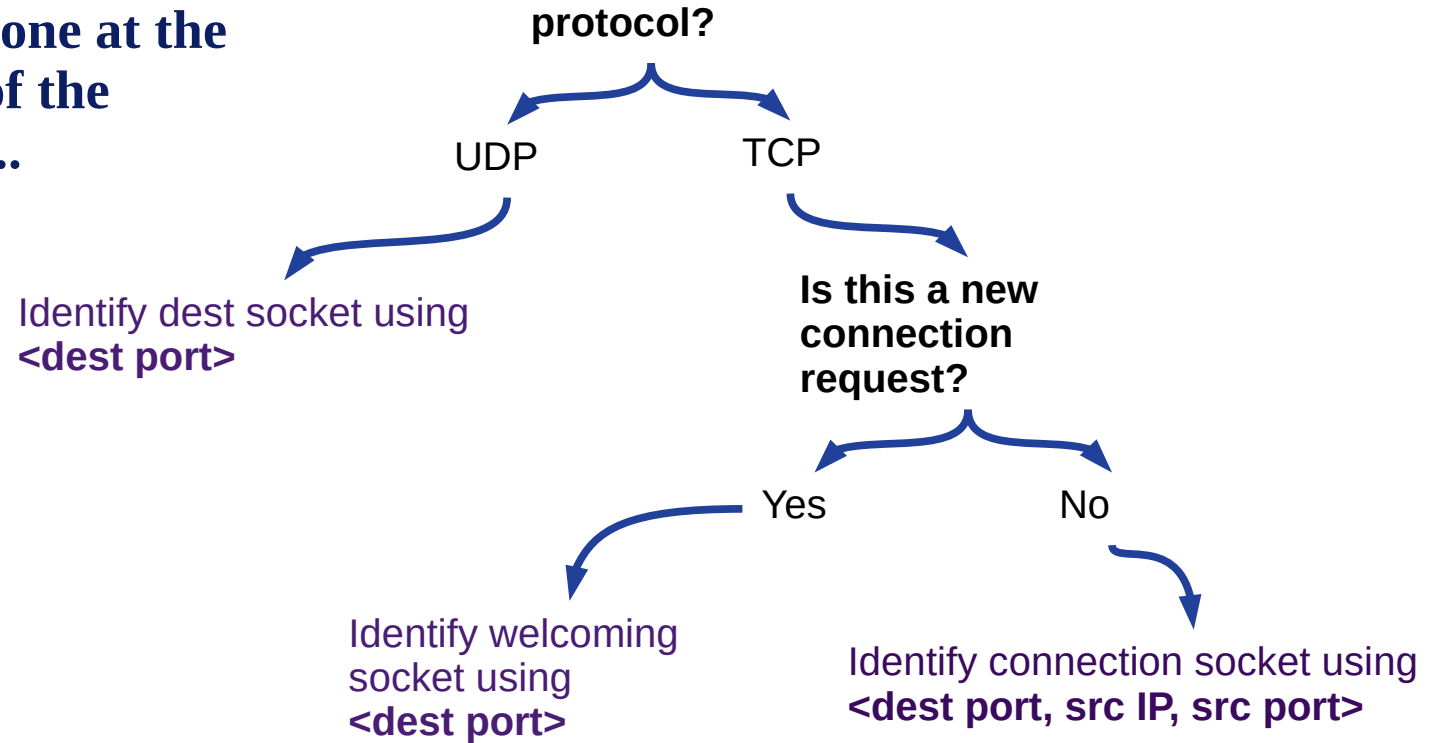
- **“Basic” services (offered by UDP as well as TCP):**
 - Process-to-process delivery (multiplexing and demultiplexing)
 - Basic error detection (using checksums)
- **Services offered by TCP (in addition to the above):**
 - Reliable Data Transfer (using seq numbers, acks etc)
 - Congestion Control
 - Flow control

Multiplexing and Demultiplexing

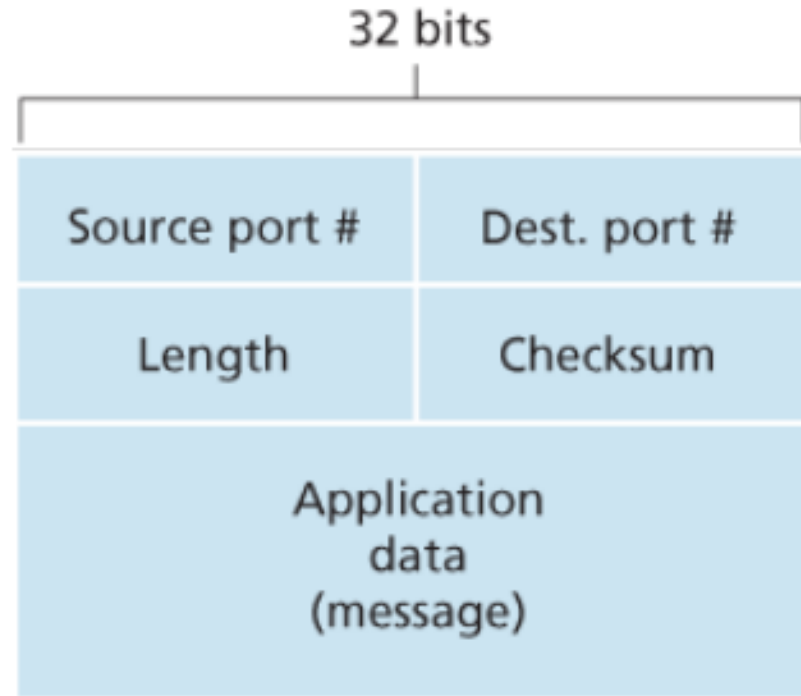


Multiplexing and Demultiplexing

**Demultiplexing done at the
Transport layer of the
destination host ...**



UDP segment (header is 8 Bytes long)



Some advantages of UDP over TCP

- Fine control over when data is sent
 - TCP's congestion control etc prevents this
- No connection establishment overhead (delays)
- No need to maintain per-connection state. Server design can be simple
- Small packet header overhead (only 8 Bytes)

Reliable Data Transfer

- How can reliable data transfer be possible using the unreliable delivery service provided by the Network layer?

Exercise

- Explore the "netstat", "ss" and "nmap" commands for listing information about open sockets on your system or a remote machine. Find out what option can be used to list only TCP or only UDP sockets, and find out what the option "-n" does for the netstat/ss commands.

References and Reading Assignment

- **Kurose and Ross 6th ed:** Sections 3.1, 3.2, 3.3

So far...

- Structure and Physical components of the Internet
- Design of the Internet: Layering and Encapsulation
- The Applications Layer:
 - Sockets Interface
 - The Web and HTTP
 - DNS
- **The Transport Layer: how it works**
 - **Basic services, UDP**
- ➡ **Principles of Reliable Data Transfer**
 - **Pipelined data transfer (Sliding window protocols)**
 - **TCP details**
 - **Congestion and Flow control**