

CSCI 466: Networks

Lecture 5(?): Transport Layer

Reese Pearsall
Fall 2022

Announcements

PA1 Due Monday September 26th

- Files must be pushed to a PA1 folder on your GitHub Repo
- Video demo is required
- Submit your repo link to D2L when finished

Next Friday will be a work day + help session. No Lecture

Got my new laptop today  Wireshark next week!

Announcements

PA1

Announcements

PA1

UDP Example

OSI Model

Application Layer

Presentation Layer *

Session Layer *

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Application Layer

Messages from Network Applications



Physical Layer

Bits being transmitted over a copper wire

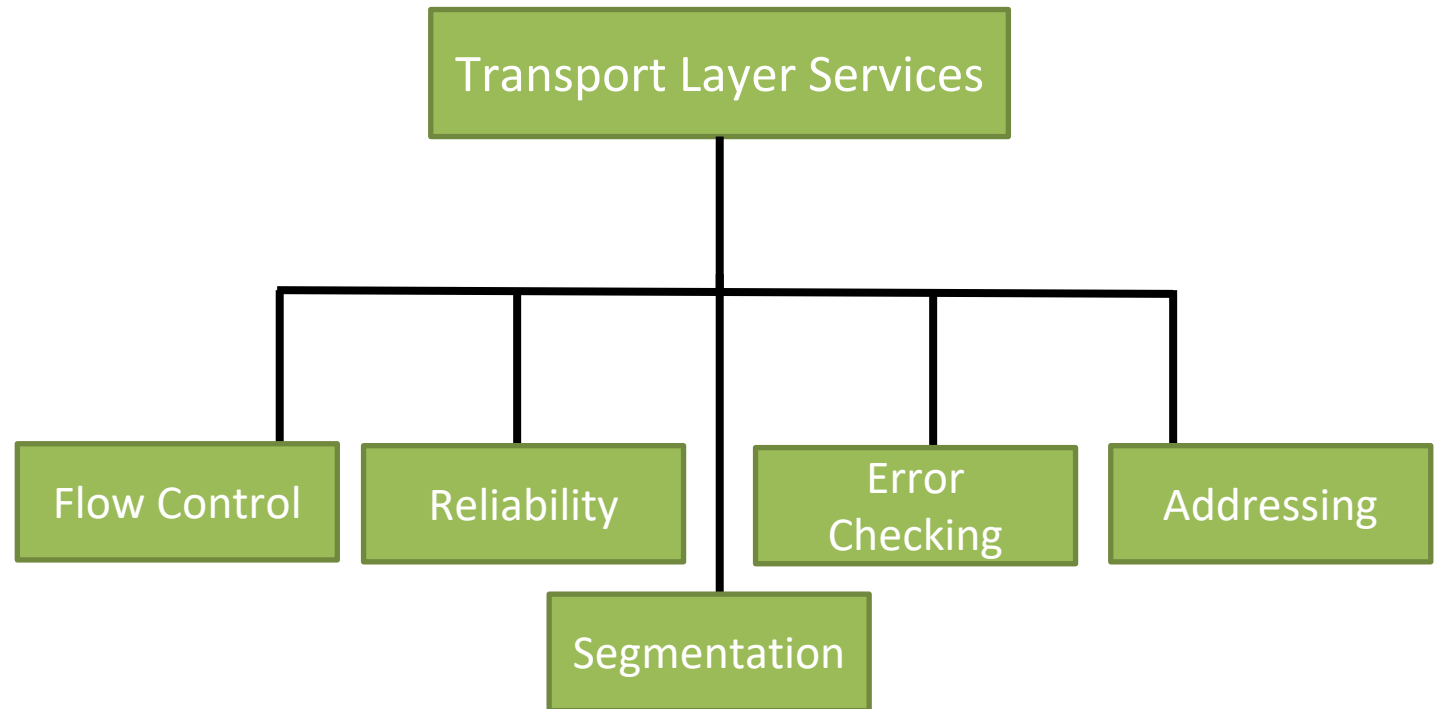
**In the textbook, they condense it to a 5-layer model, but 7 layers is what is most used*

Transport Layer

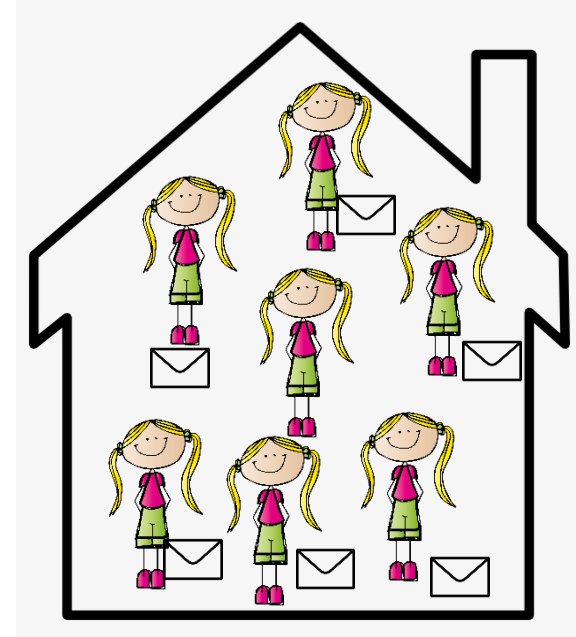
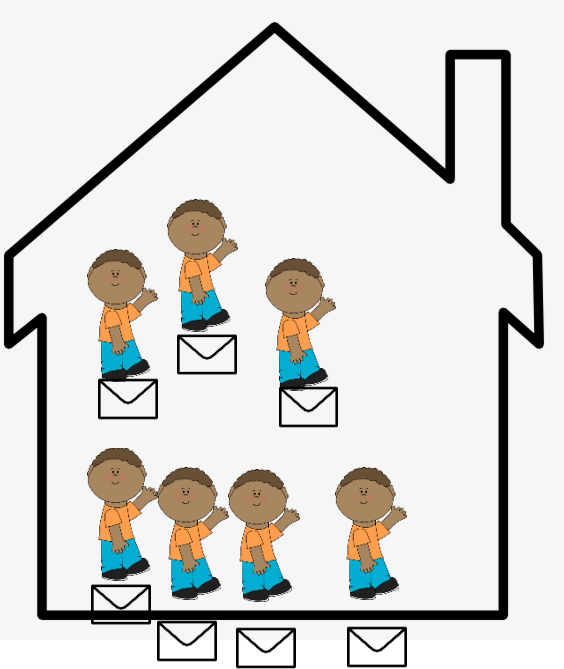
Provides *logical* end-to-end communication between application processes

Important Services

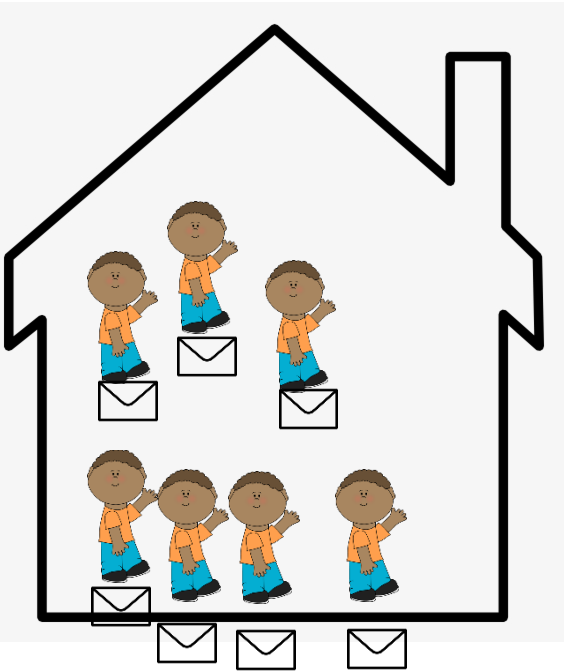
- Flow Control
- Reliability
- Segmentation
- Error Checking
- Addressing



Transport Layer



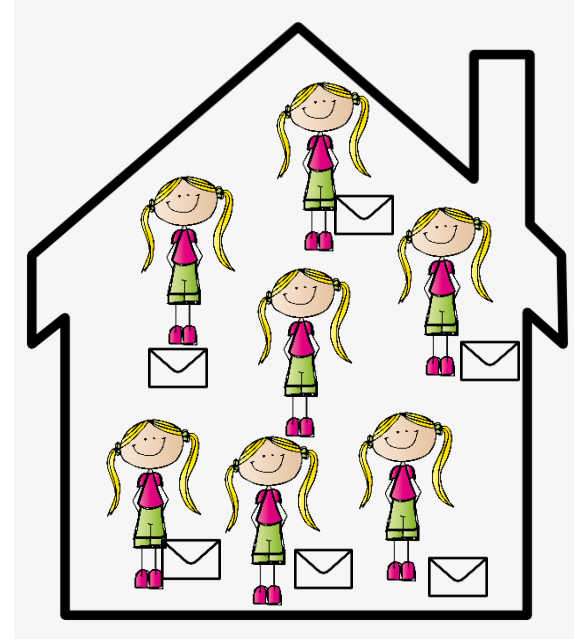
Transport Layer



Bill



Ann

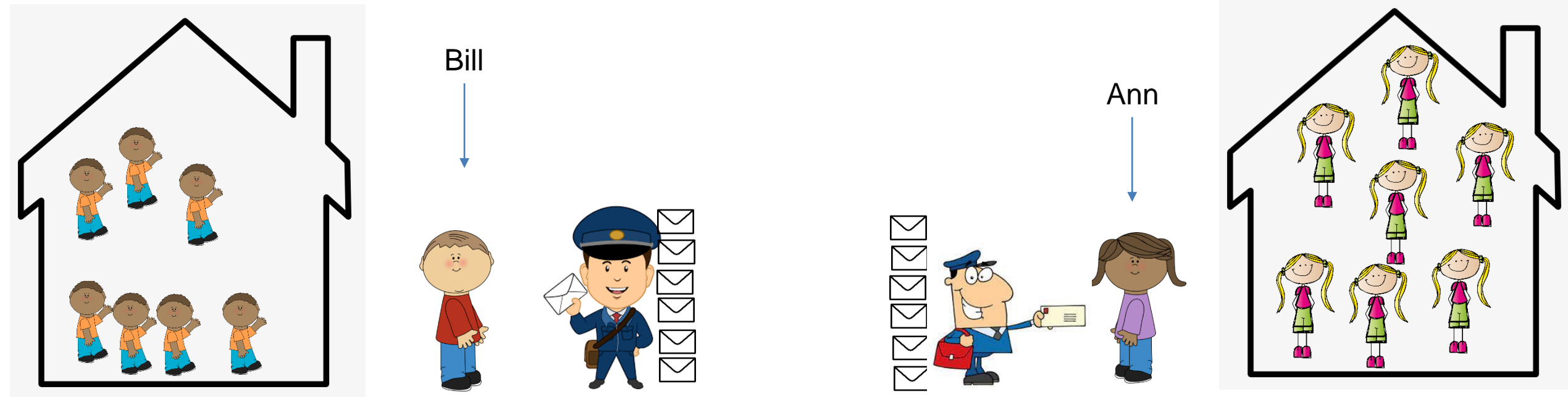


Transport Layer



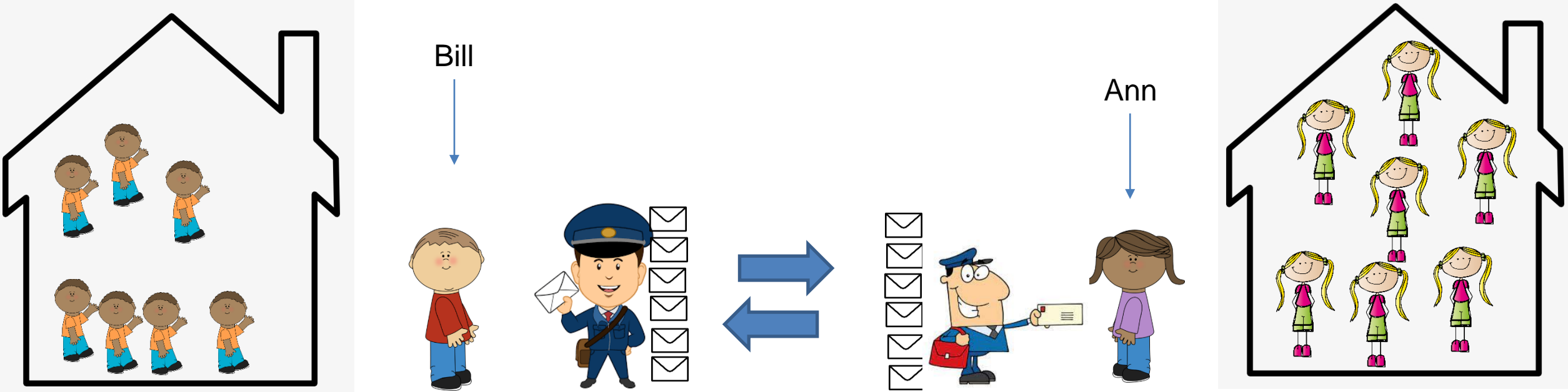
Bill and Ann are responsible for collecting their siblings mail...

Transport Layer



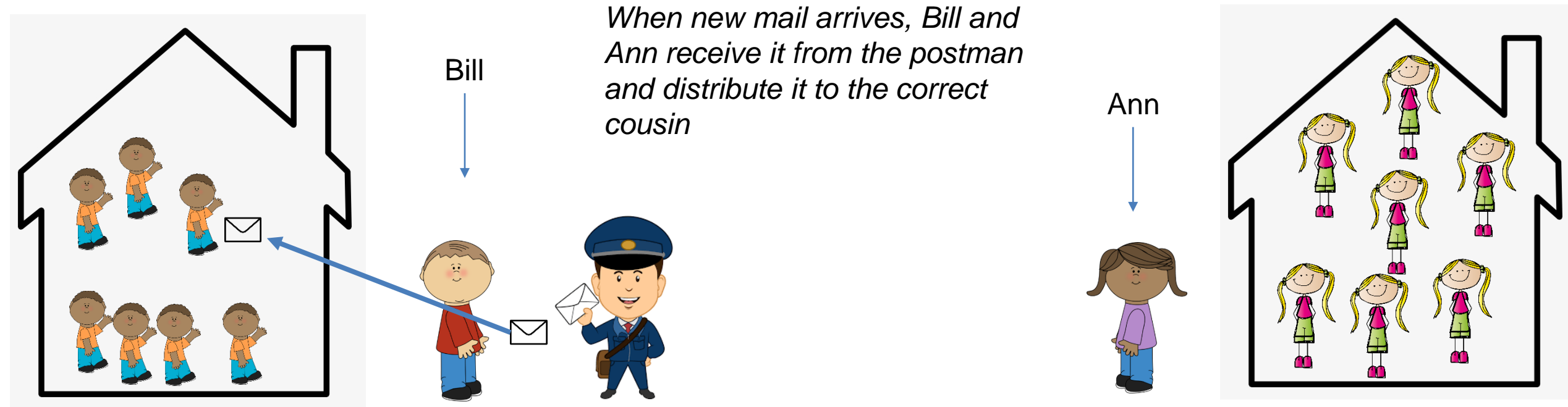
Bill and Ann are responsible for collecting their siblings mail...
And delivering it to the postal service worker

Transport Layer



Bill and Ann are responsible for collecting their siblings mail...
And delivering it to the postal service worker

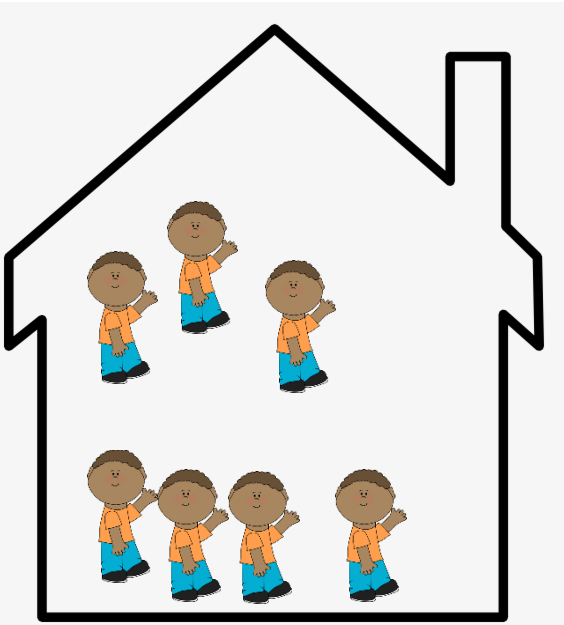
Transport Layer



Bill and Ann are responsible for collecting their siblings mail...
And delivering it to the postal service worker

Transport Layer

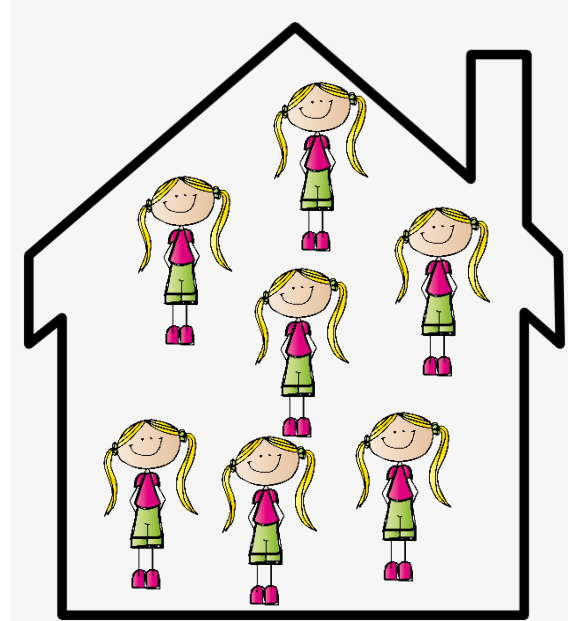
Letters in envelopes = Application messages



Bill

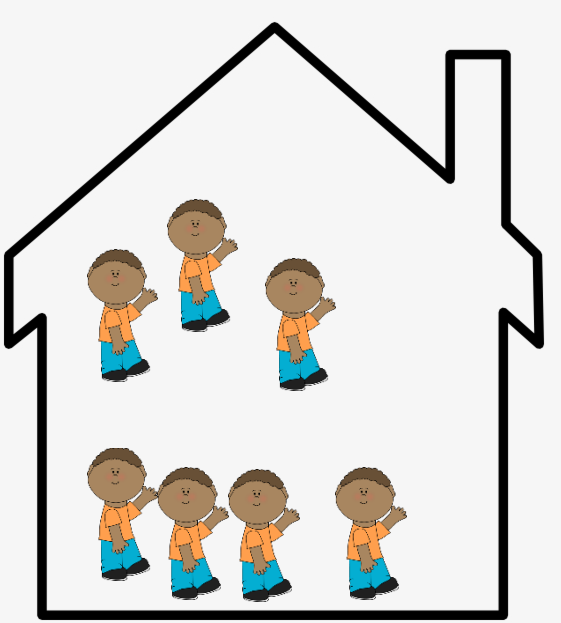


Ann



Transport Layer

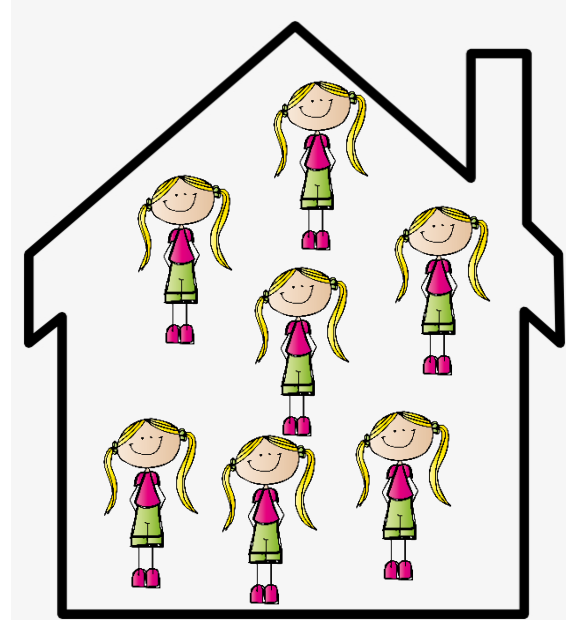
Letters in envelopes = Application messages
Cousins = Processes



Bill

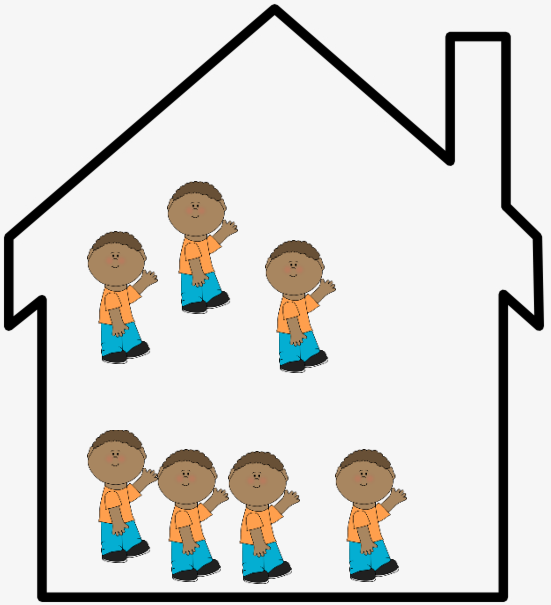


Ann



Transport Layer

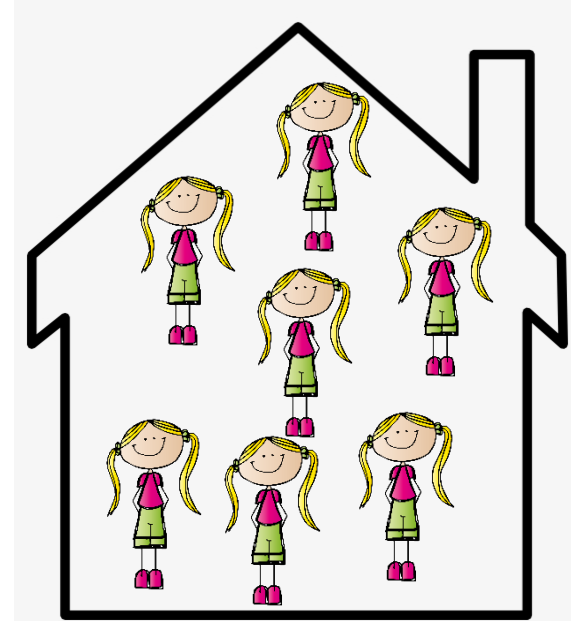
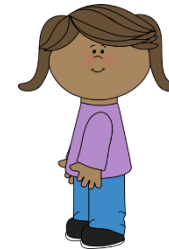
Letters in envelopes = Application messages
Cousins = Processes
Houses = Hosts/End systems



Bill



Ann



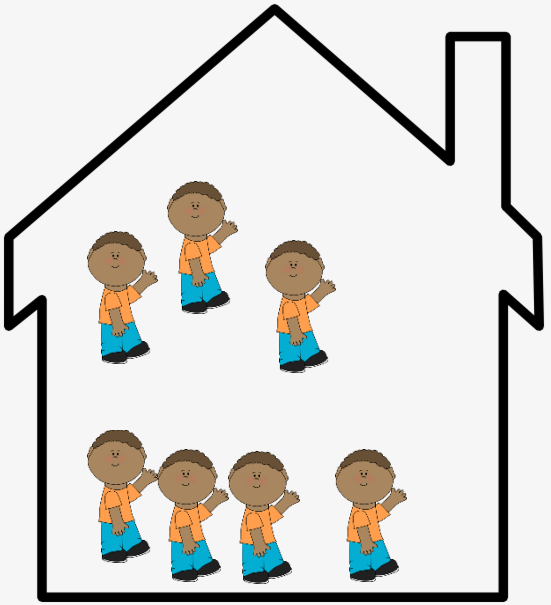
Transport Layer

Letters in envelopes = Application messages

Cousins = Processes

Houses = Hosts/End systems

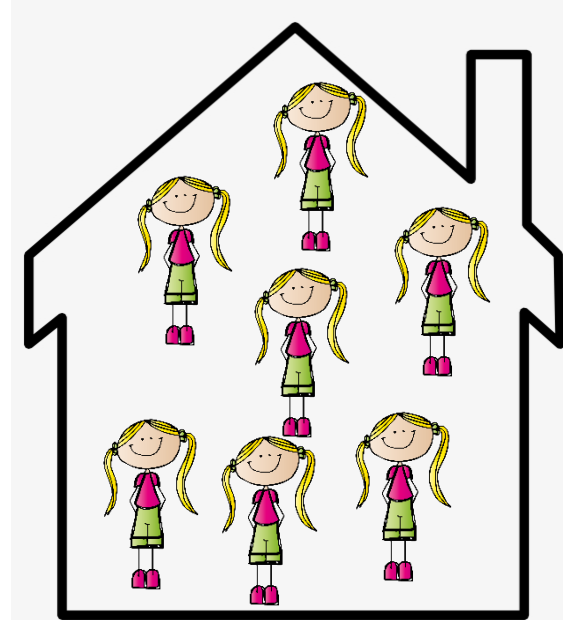
Ann and Bill = Transport Layer Protocol



Bill

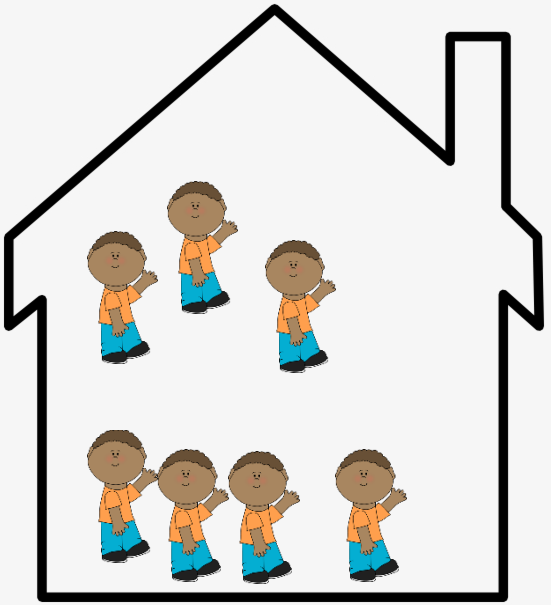


Ann



Transport Layer

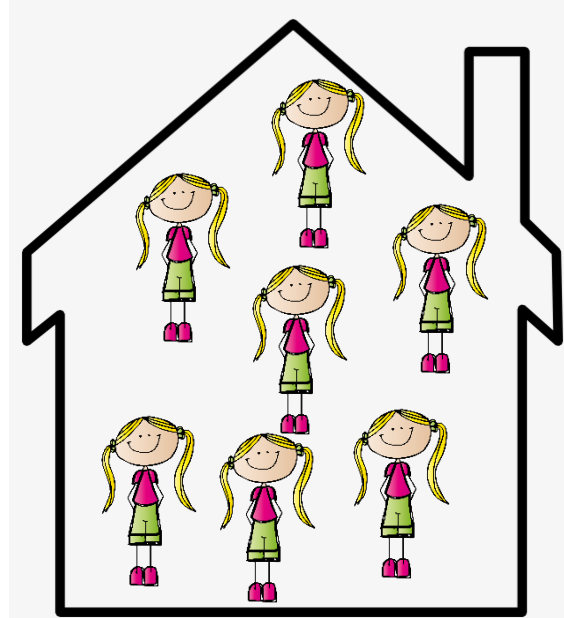
Letters in envelopes = Application messages
Cousins = Processes
Houses = Hosts/End systems
Ann and Bill = Transport Layer Protocol
Postal Service = Postal Service



Bill



Ann



Transport Layer

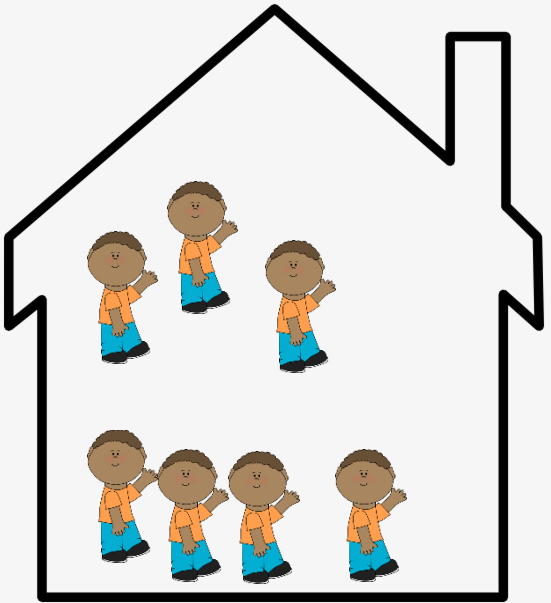
Letters in envelopes = Application messages

Cousins = Processes

Houses = Hosts/End systems

Ann and Bill = Transport Layer Protocol

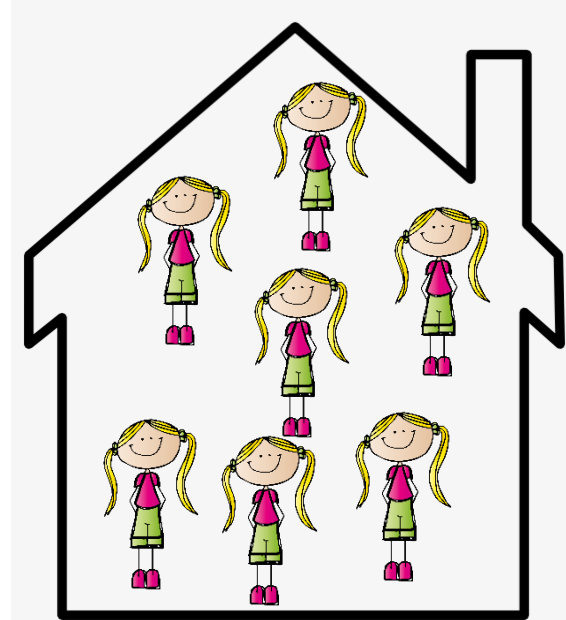
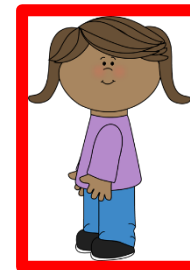
Postal Service = Postal Service



Bill



Ann

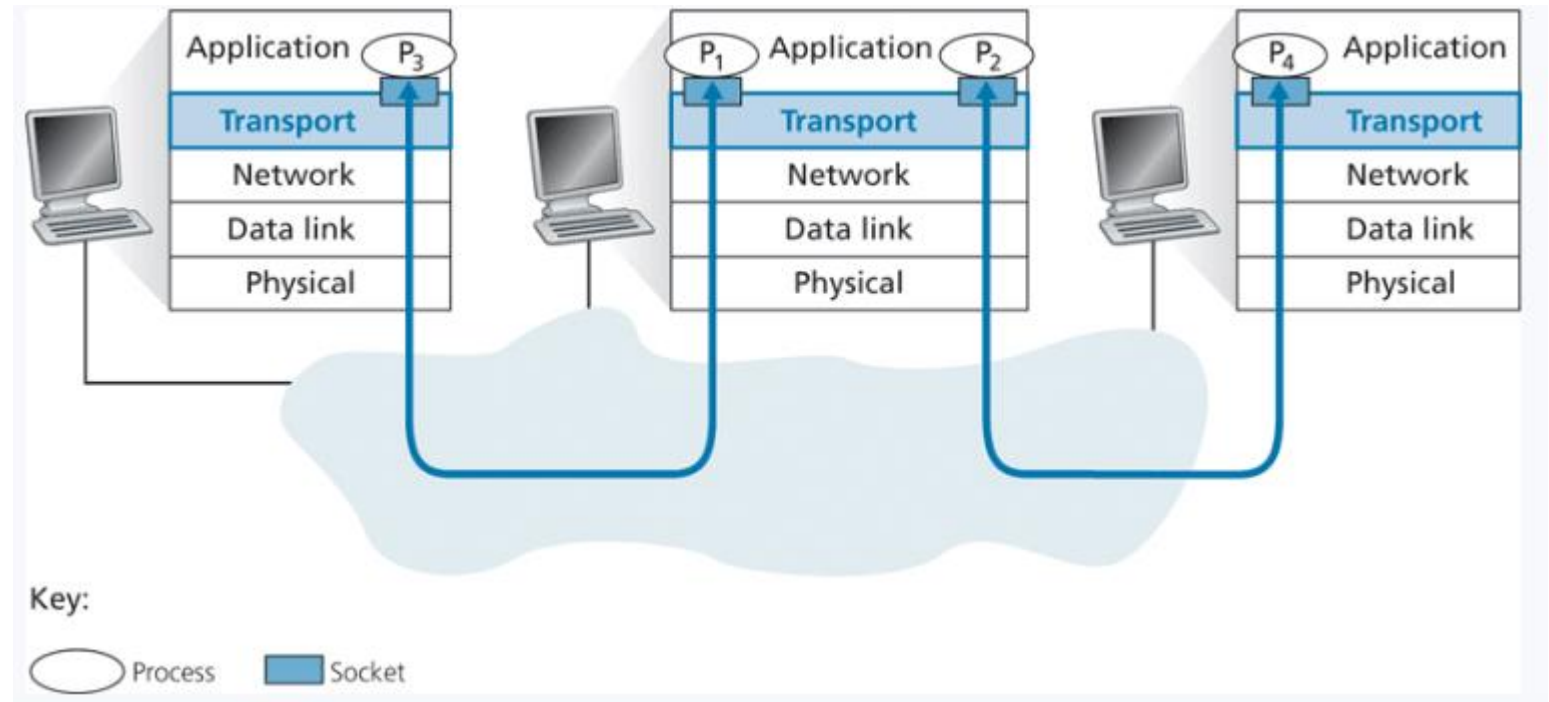


Transport Layer

TCP – Reliable, Connection-oriented transport layer protocol

UDP – Unreliable, connectionless transport layer protocol

Transport layer delivers to **sockets**

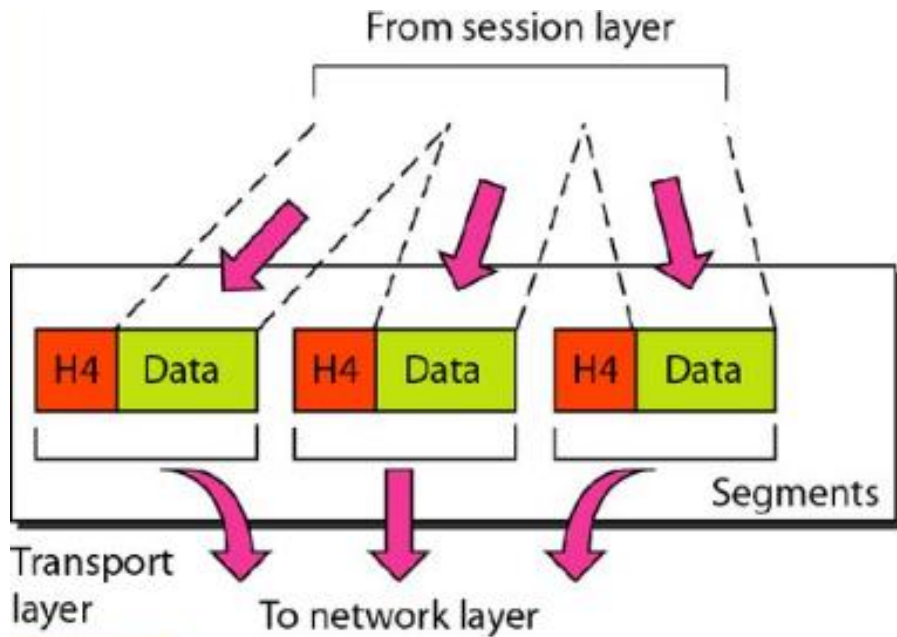


Transport Layer

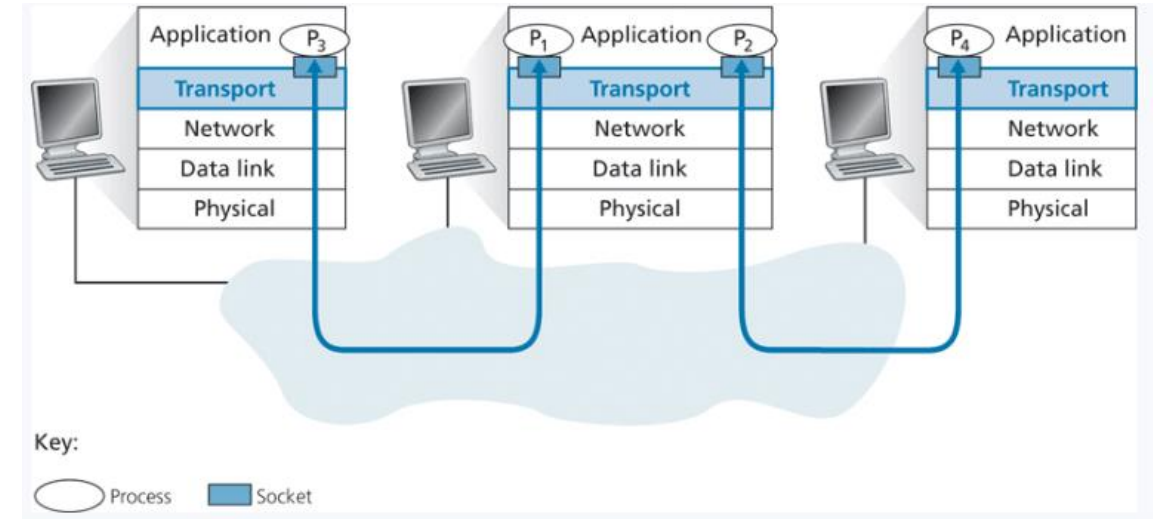
TCP – Reliable, Connection-oriented transport layer protocol

UDP – Unreliable, connectionless transport layer protocol

Transport layer delivers to **sockets**



Messages from the Application Layer/Session Layer are split into smaller chunks called **segments**, and passed into the network layer

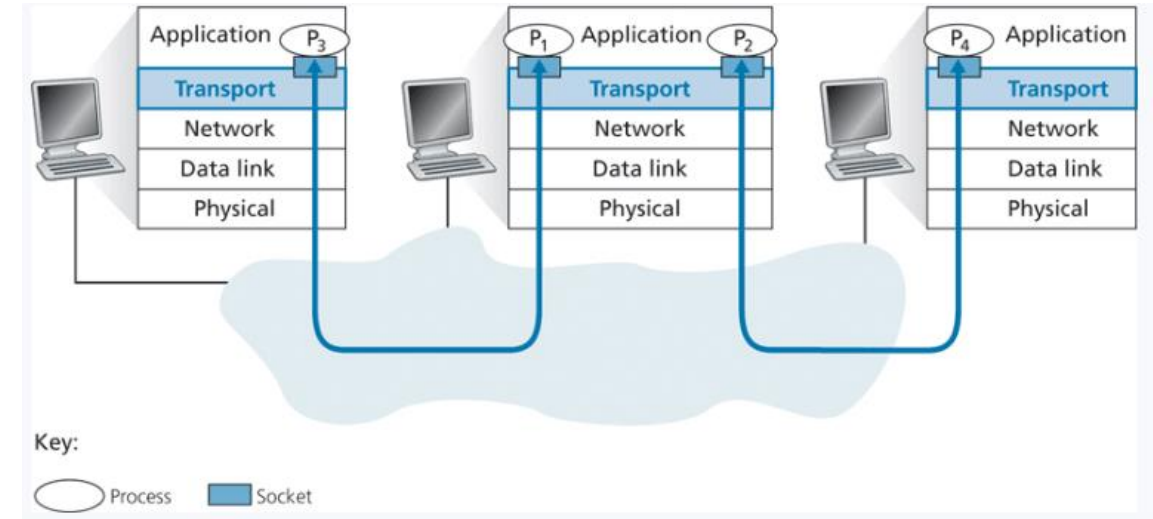
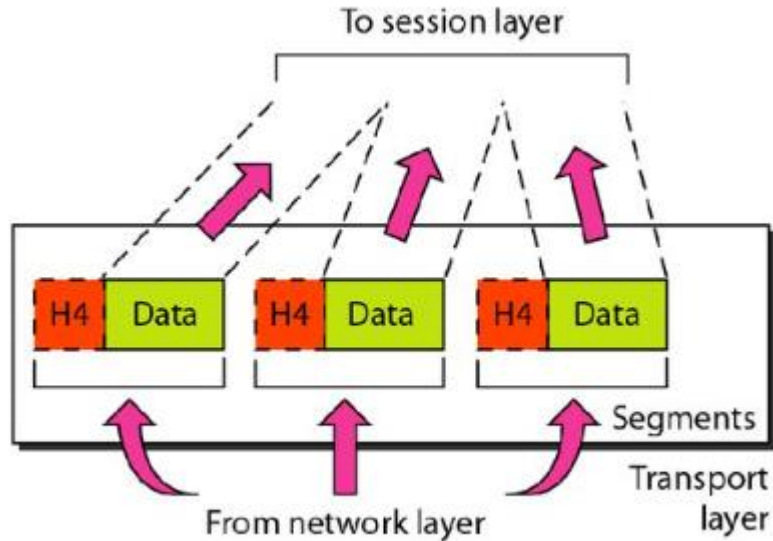


Transport Layer

TCP – Reliable, Connection-oriented transport layer protocol

UDP – Unreliable, connectionless transport layer protocol

Transport layer delivers to **sockets**



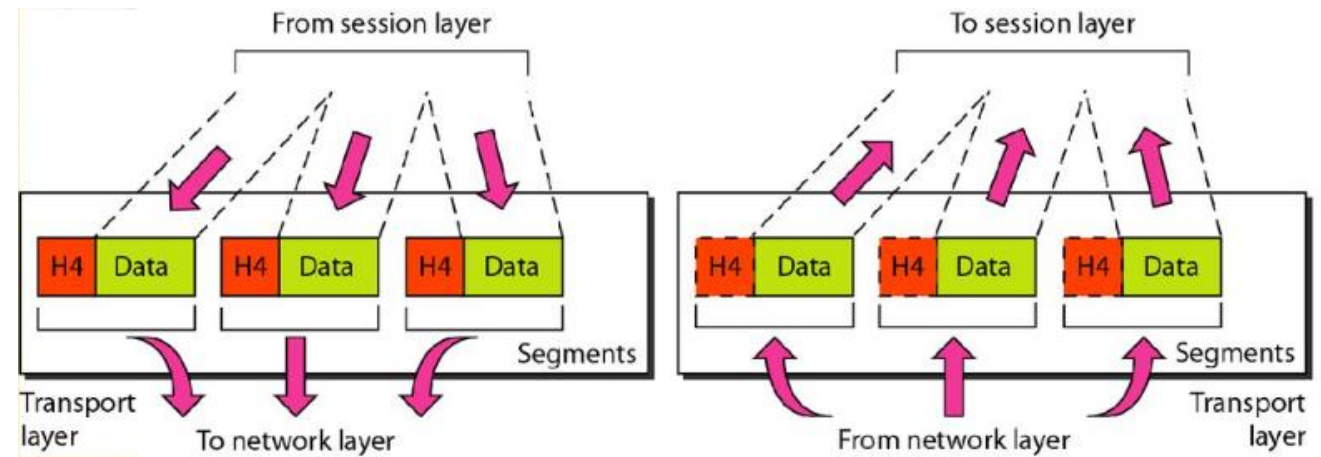
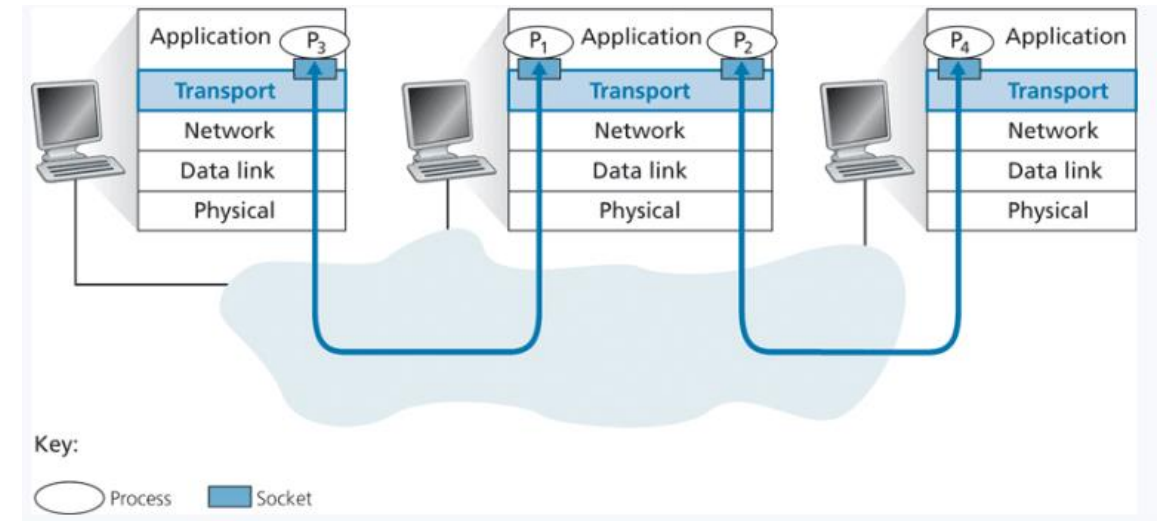
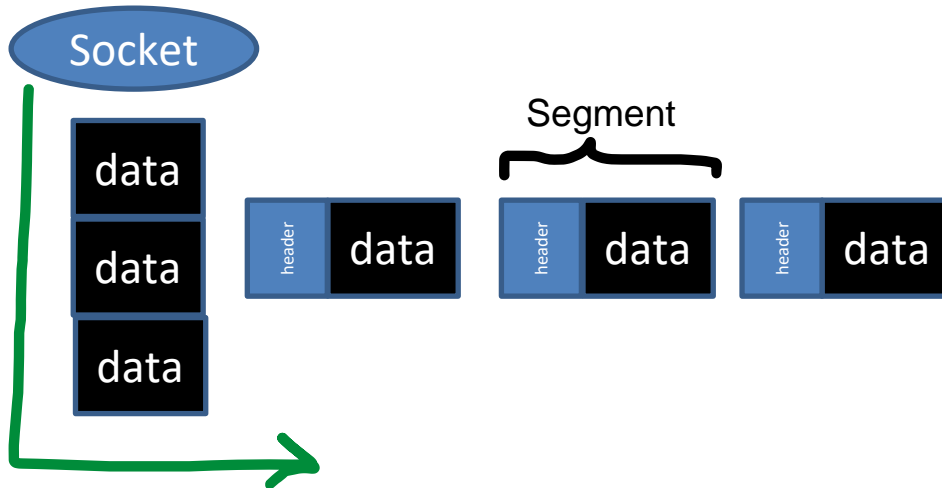
Messages from the network layer arrive as segments. Transport layer must reassemble to send it to the correct process

Transport Layer

TCP – Reliable, Connection-oriented transport layer protocol

UDP – Unreliable, connectionless transport layer protocol

Multiplexing is the process of gathering chunks from sockets, encapsulating chunks with header information, and passing the segment into the network layer

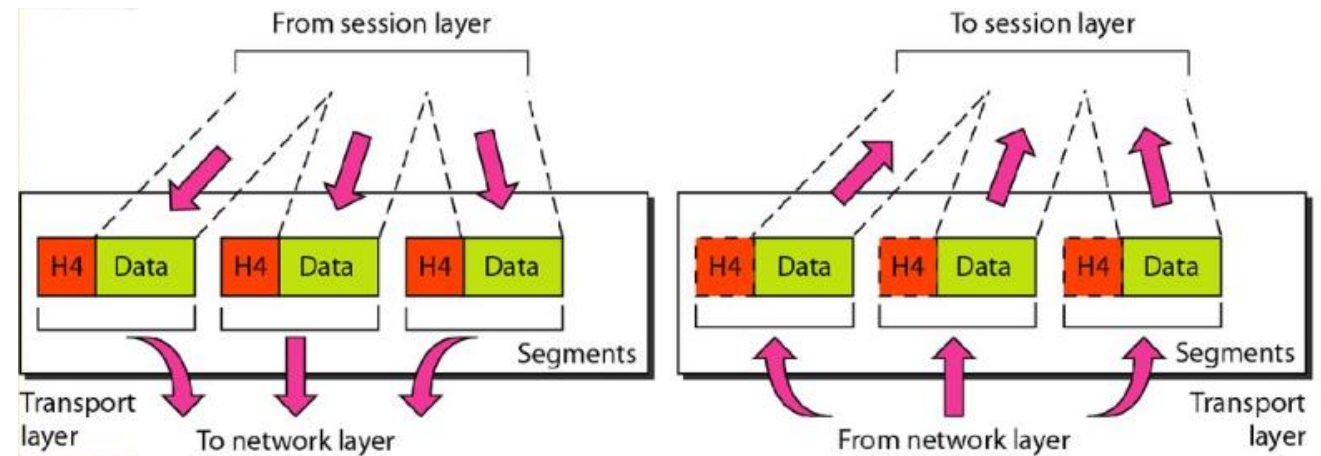
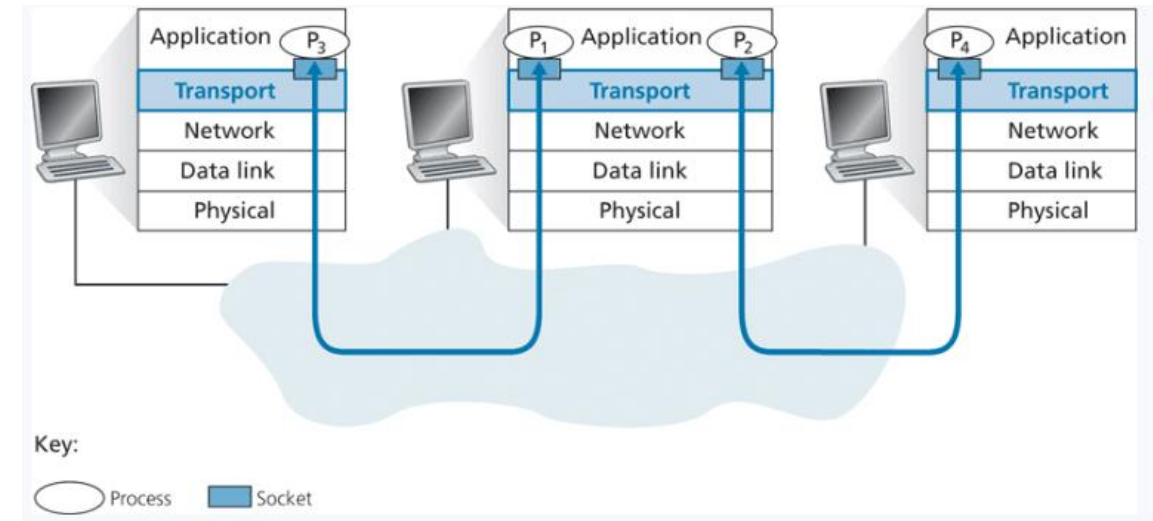
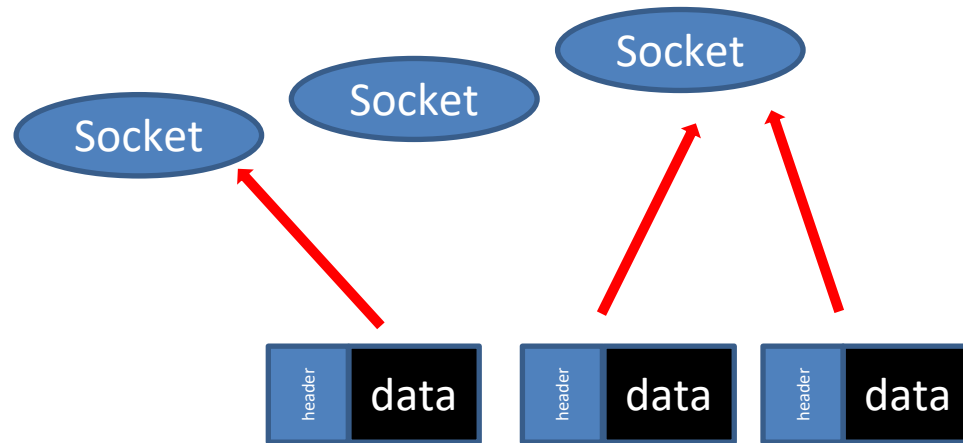


Transport Layer

TCP – Reliable, Connection-oriented transport layer protocol

UDP – Unreliable, connectionless transport layer protocol

Demultiplexing is the receiving segments from the transport layer and delivering the segment to the correct socket.

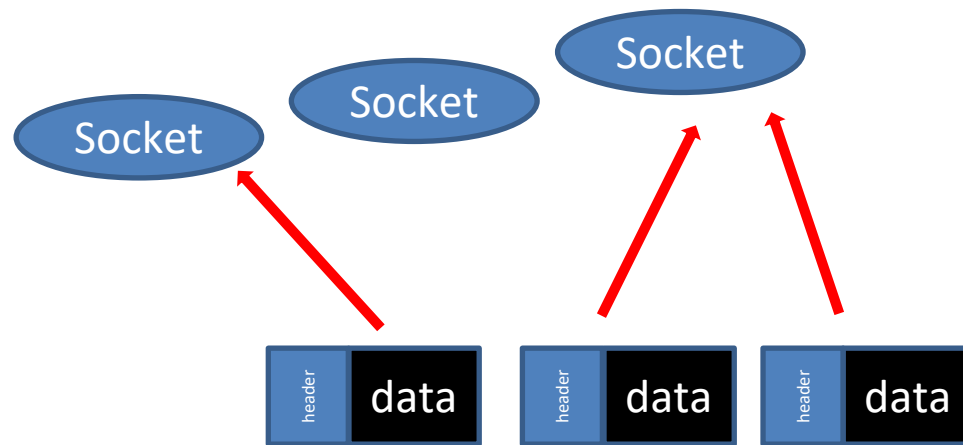


Transport Layer

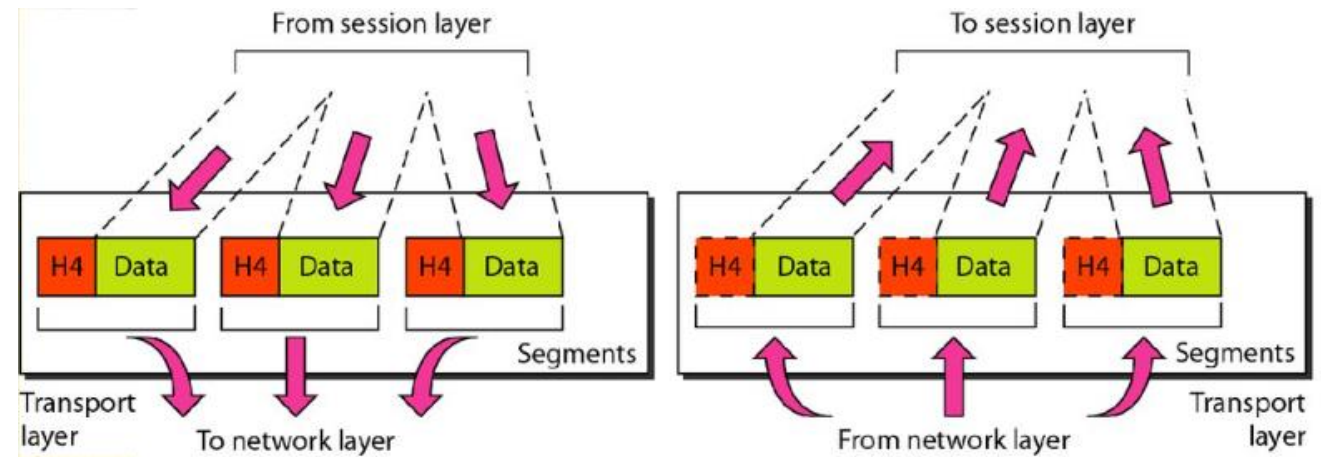
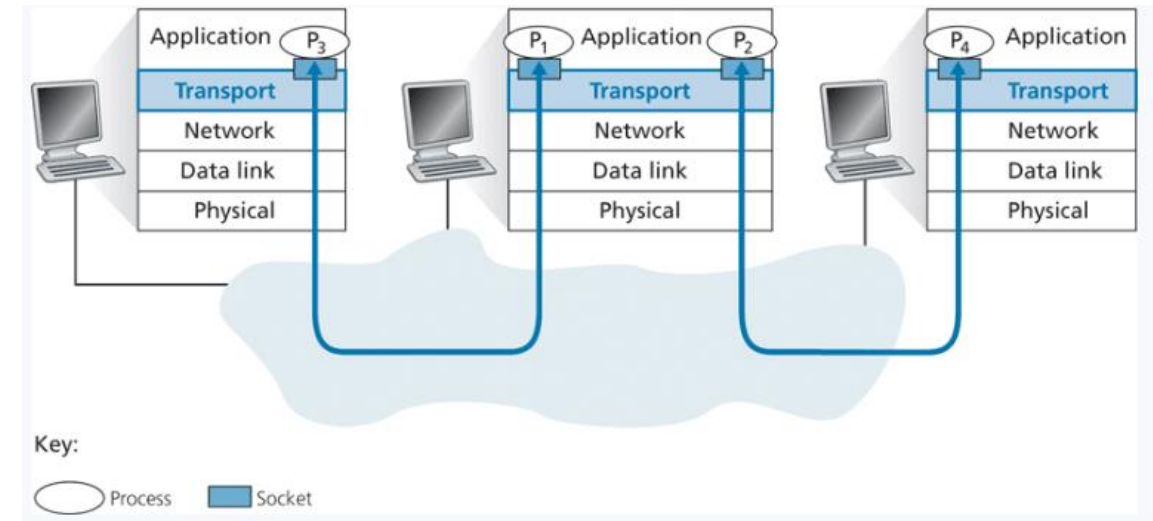
TCP – Reliable, Connection-oriented transport layer protocol

UDP – Unreliable, connectionless transport layer protocol

Demultiplexing is the receiving segments from the transport layer and delivering the segment to the correct socket.



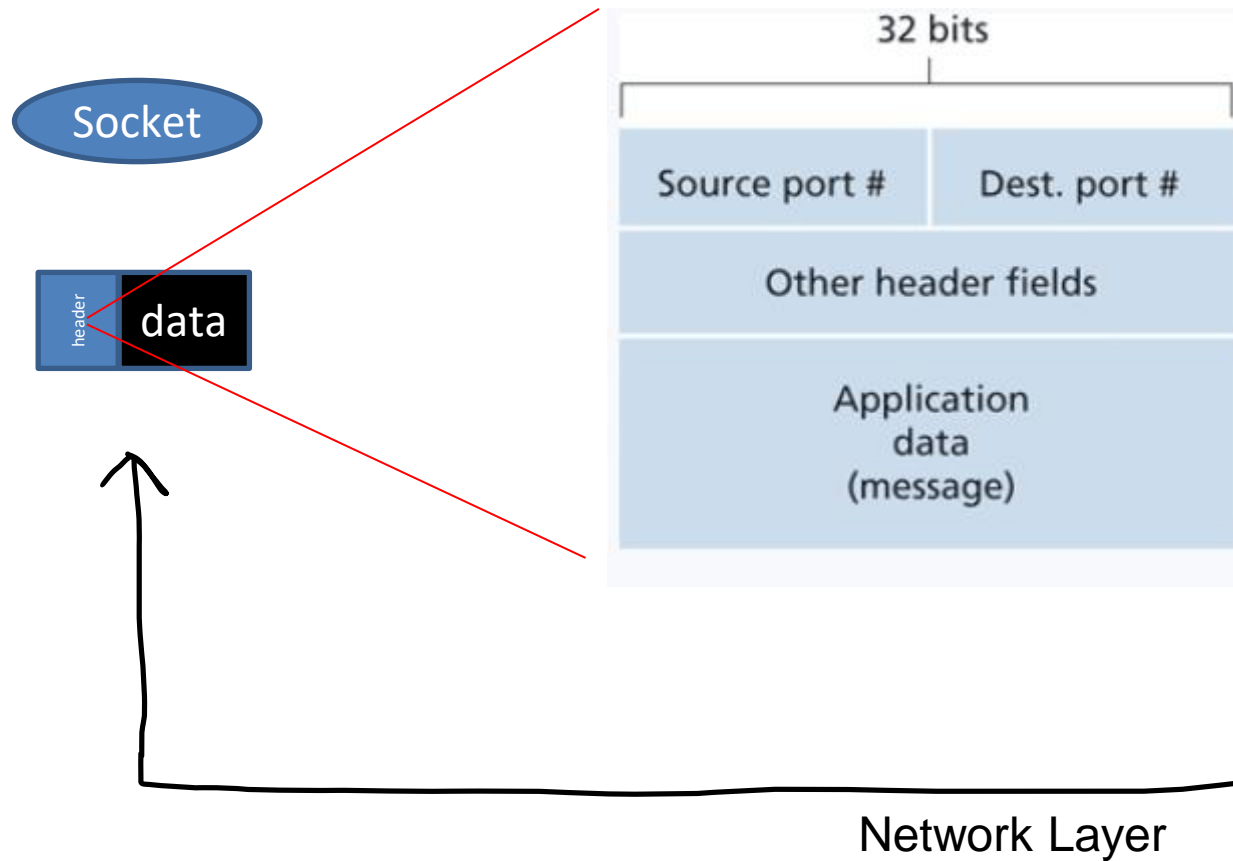
Bill and Ann?



Transport Layer

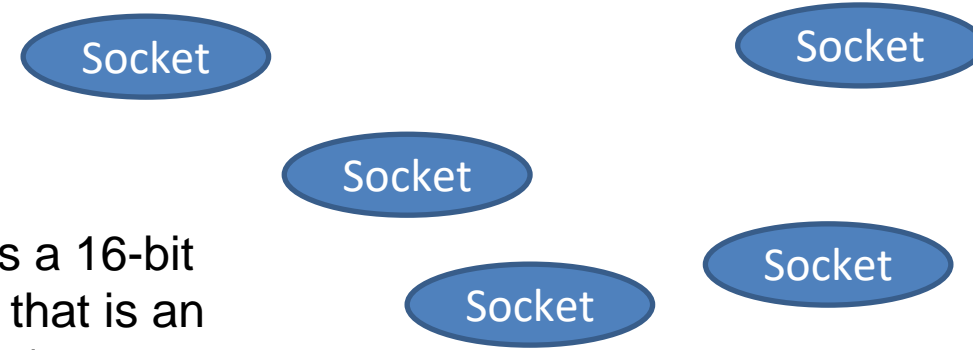
TCP – Reliable, Connection-oriented transport layer protocol

UDP – Unreliable, connectionless transport layer protocol



A **port** is a 16-bit number that is an entrance/exit point of a machine

An active port is associated with a specific process or service



Transport Layer

TCP – Reliable, Connection-oriented transport layer protocol

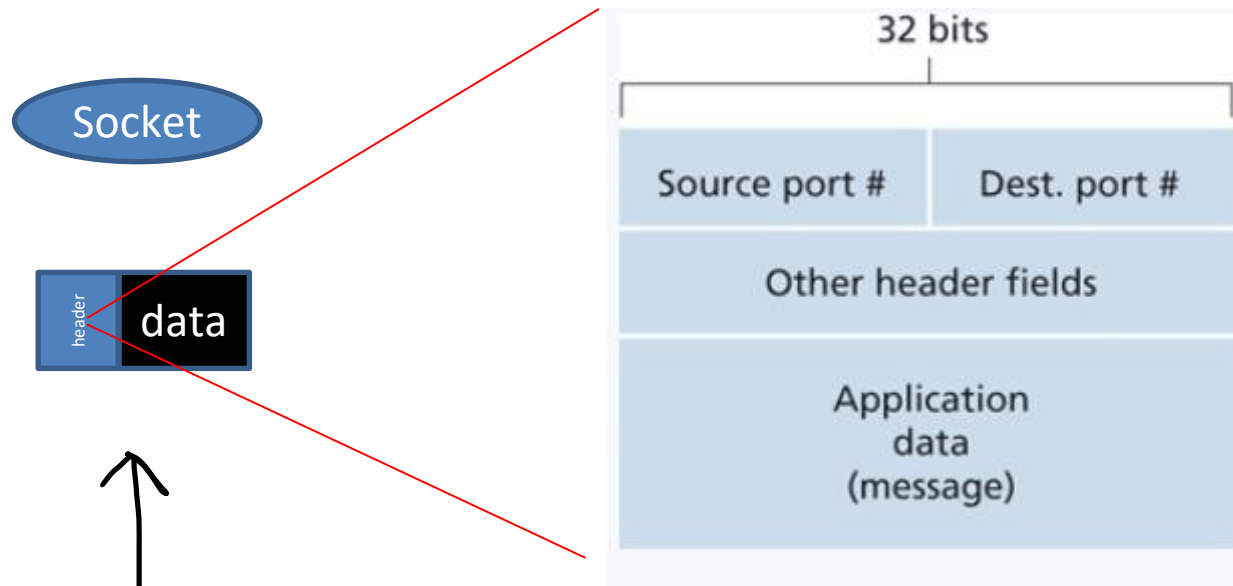
UDP – Unreliable, connectionless transport layer protocol



Transport Layer

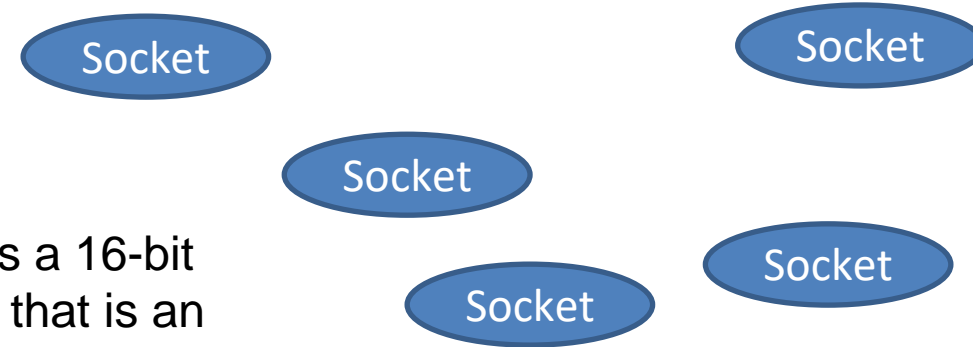
TCP – Reliable, Connection-oriented transport layer protocol

UDP – Unreliable, connectionless transport layer protocol



A **port** is a 16-bit number that is an entrance/exit point of a machine

An active port is associated with a specific process or service



When developing a new application, we need to assign the application a port number (greater than 1024)

Network Layer

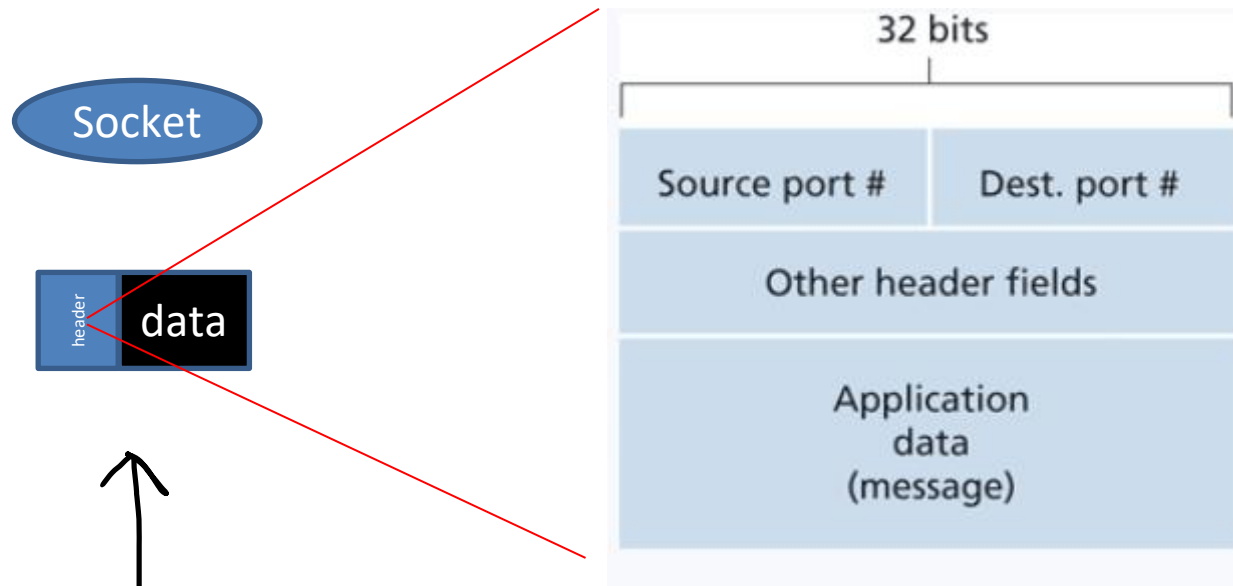
```
clientSocket.bind('', 19157))
```

```
clientSocket = socket(AF_INET, SOCK_DGRAM)
```

Transport Layer

TCP – Reliable, Connection-oriented transport layer protocol

UDP – Unreliable, connectionless transport layer protocol



A **port** is a 16-bit number that is an entrance/exit point of a machine

An active port is associated with a specific process or service

Why does a UDP packet header need a source port??

Network Layer

