



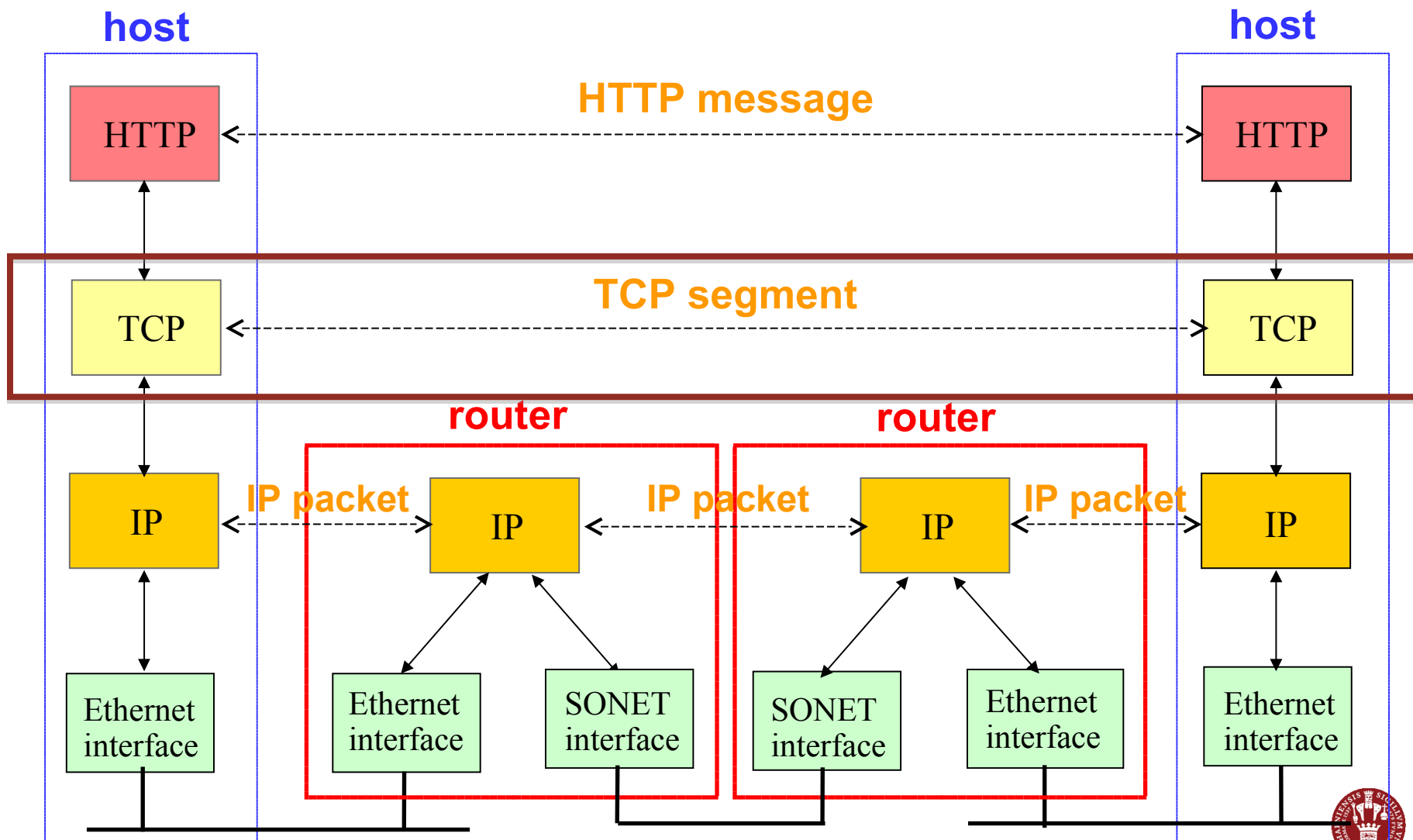
UNIVERSITY OF COPENHAGEN

Transport Layer: UDP

Vivek Shah

Based on slides compiled by Marcos Vaz Salles

Recap: Internet Layering Model

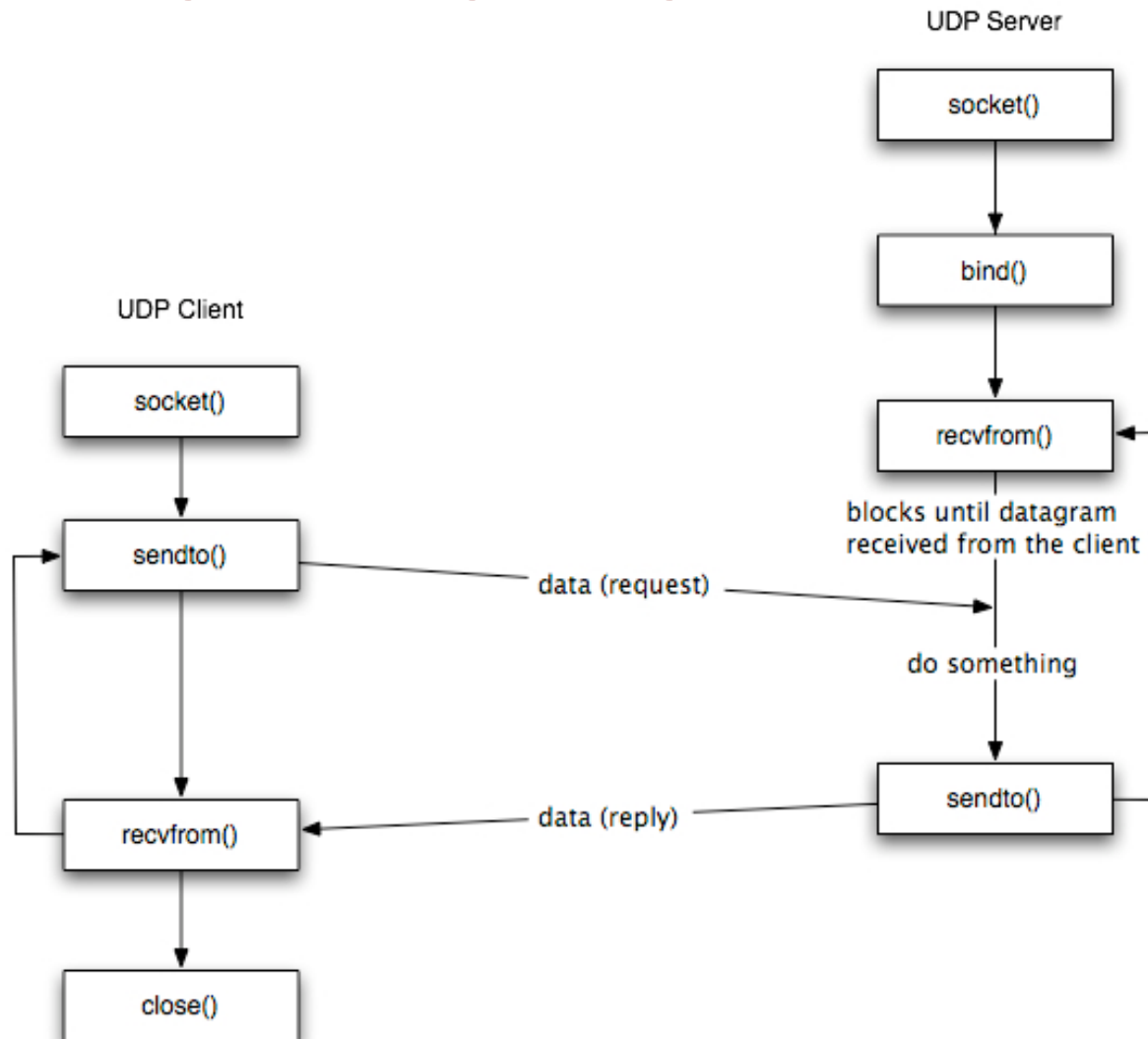


Transport Layer

- Logical Communication between processes
 - Sender divides messages into segments.
 - Receiver re-assembles messages into segments.
- Principles underlying transport-layer services
 - (De)multiplexing
 - Detecting corruption
 - Optional: Reliable delivery, Flow control, Congestion control
- Transport-layer protocols in the Internet
 - User Datagram Protocol (UDP)
 - Simple (unreliable) message delivery
 - Transmission Control Protocol (TCP)
 - Reliable bidirectional stream of bytes



Socket Programming Using UDP



Socket Programming Using UDP

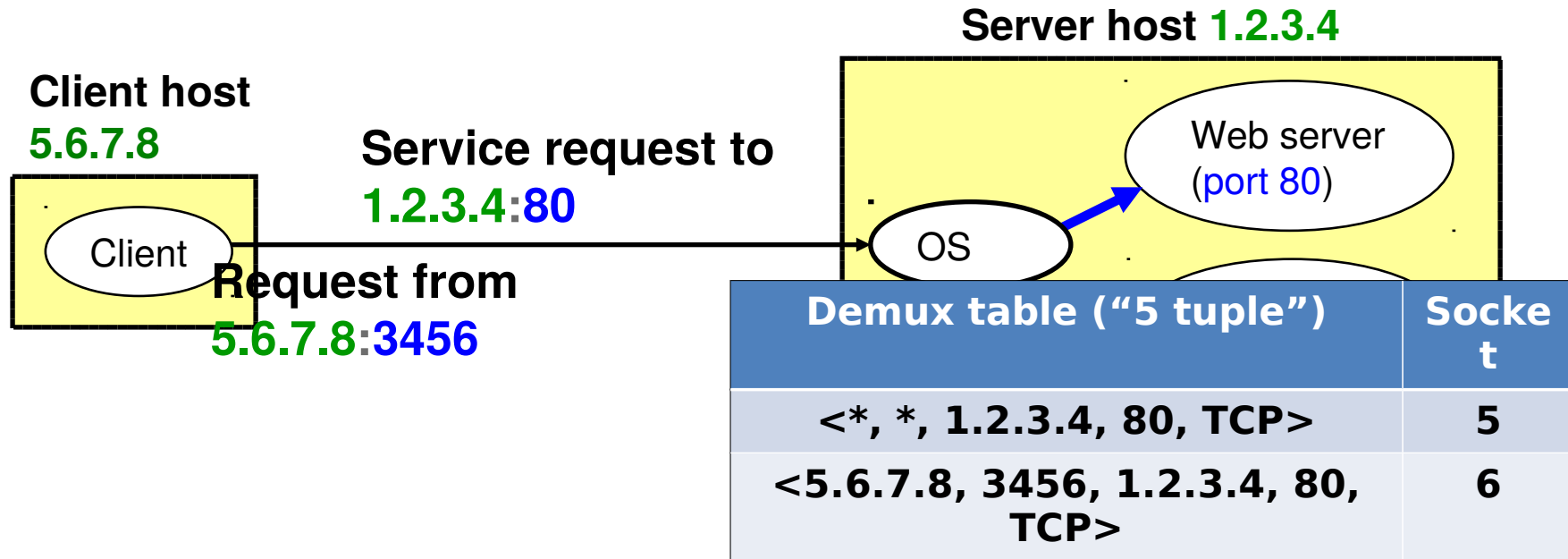
```
ssize_t recvfrom(int sockfd, void* buff,  
    size_t nbytes, int flags, struct sockaddr* from,  
    socklen_t *addrlen);
```

```
ssize_t sendto(int sockfd, const void *buff,  
    size_t nbytes, int flags,  
    const struct sockaddr *to, socklen_t addrlen);
```



Two Basic Transport Features

- **Demultiplexing:** port numbers



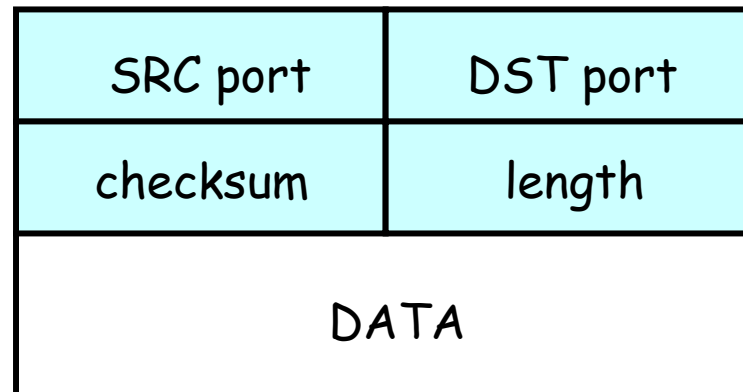
- **Error**



Source: Freedman

User Datagram Protocol (UDP)

- Datagram messaging service
 - Demultiplexing of messages: port numbers
 - Detecting corrupted messages: checksum
- Lightweight communication between processes
 - Send messages to and receive them from a socket
 - Avoid overhead and delays of ordered, reliable delivery



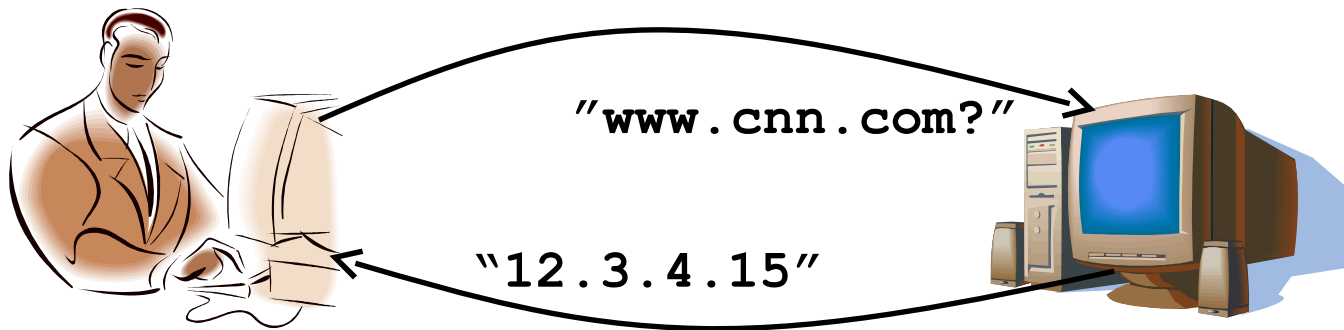
Why Would Anyone Use UDP?

- Fine control over what data is sent and when
 - As soon as app process writes into socket
 - ... UDP will package data and send packet
- No delay for connection establishment
 - UDP blasts away without any formal preliminaries
 - ... avoids introducing unnecessary delays
- No connection state (no buffers, sequence #'s, etc.)
 - Can scale to more active clients at once
- Small packet header overhead (header only 8B long)



Popular Applications That Use

- Simple query protocols like DNS
 - Overhead of connection establishment is overkill
 - Easier to have the application retransmit if needed



- Multimedia streaming (VoIP, video conferencing, ...)
 - Retransmitting lost/corrupted packets is not worthwhile
 - By time packet is retransmitted, it's too late

END TO END PRINCIPLE!

Summary

- UDP
 - basic multiplexing, checksums



What's next ? Reliable Data Transfer & TCP

