

CEN445 – Network Protocols and Algorithms

Chapter 6 – Transport Layer

6.4 Internet Transport Protocols: UDP

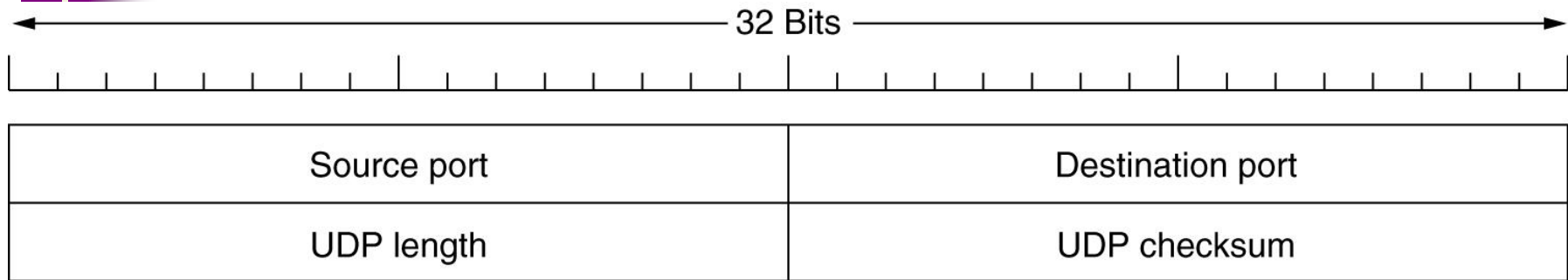
Dr. Mostafa Hassan Dahshan
Department of Computer Engineering
College of Computer and Information Sciences
King Saud University
mdahshan@ksu.edu.sa
<http://faculty.ksu.edu.sa/mdahshan>



Introduction to UDP

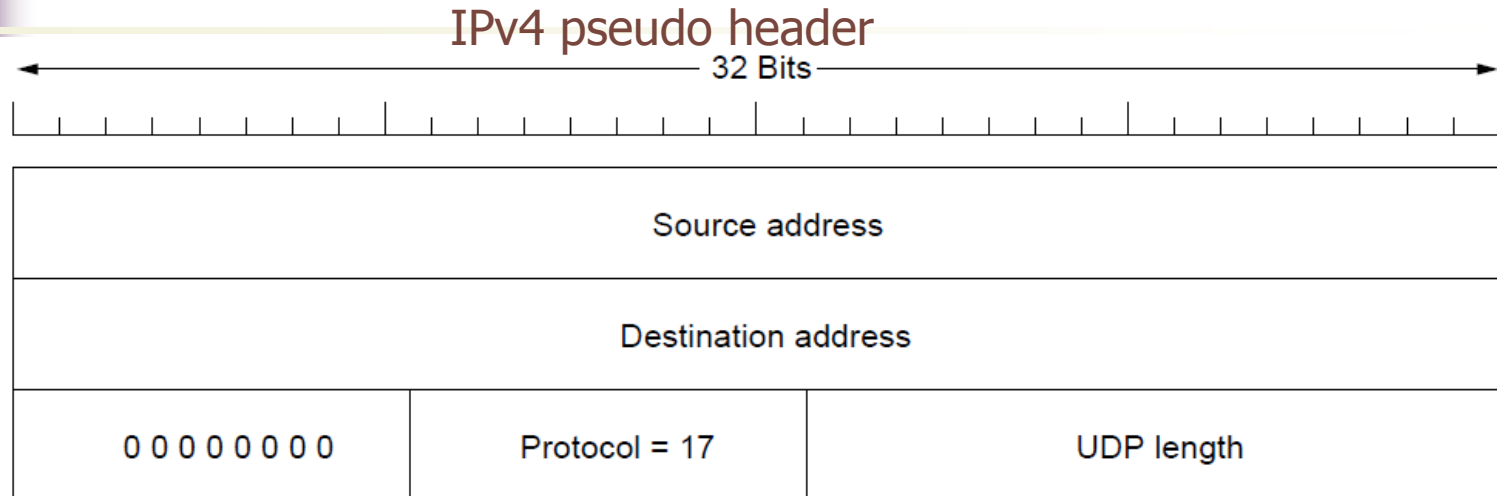
- User Datagram Protocol
- Simplest transport protocol
- Connectionless
- App can send IP datagrams w no connection
- Transmit **segments**: 8-bytes header, payload
- Ports identify connection endpoint
- Server attached to UDP port: identify app
- UDP is just raw IP with src, dest ports

Introduction to UDP



- Src port used when reply needed, swap w dst
- UDP length: header + data
- Min length 8 bytes, max 65,515 (due to IP limit)
- Checksum optional: header+data+IP pseudo hdr
- Chksm set to 0, data is padded to even # of bytes
- Add all 16-bit words using 1's complement
- Take 1's complement on result

Introduction to UDP



- IP psuedoheader to detect misbehaved packets
- Violates protocols hierarchy
- Also used in TCP



Introduction to UDP

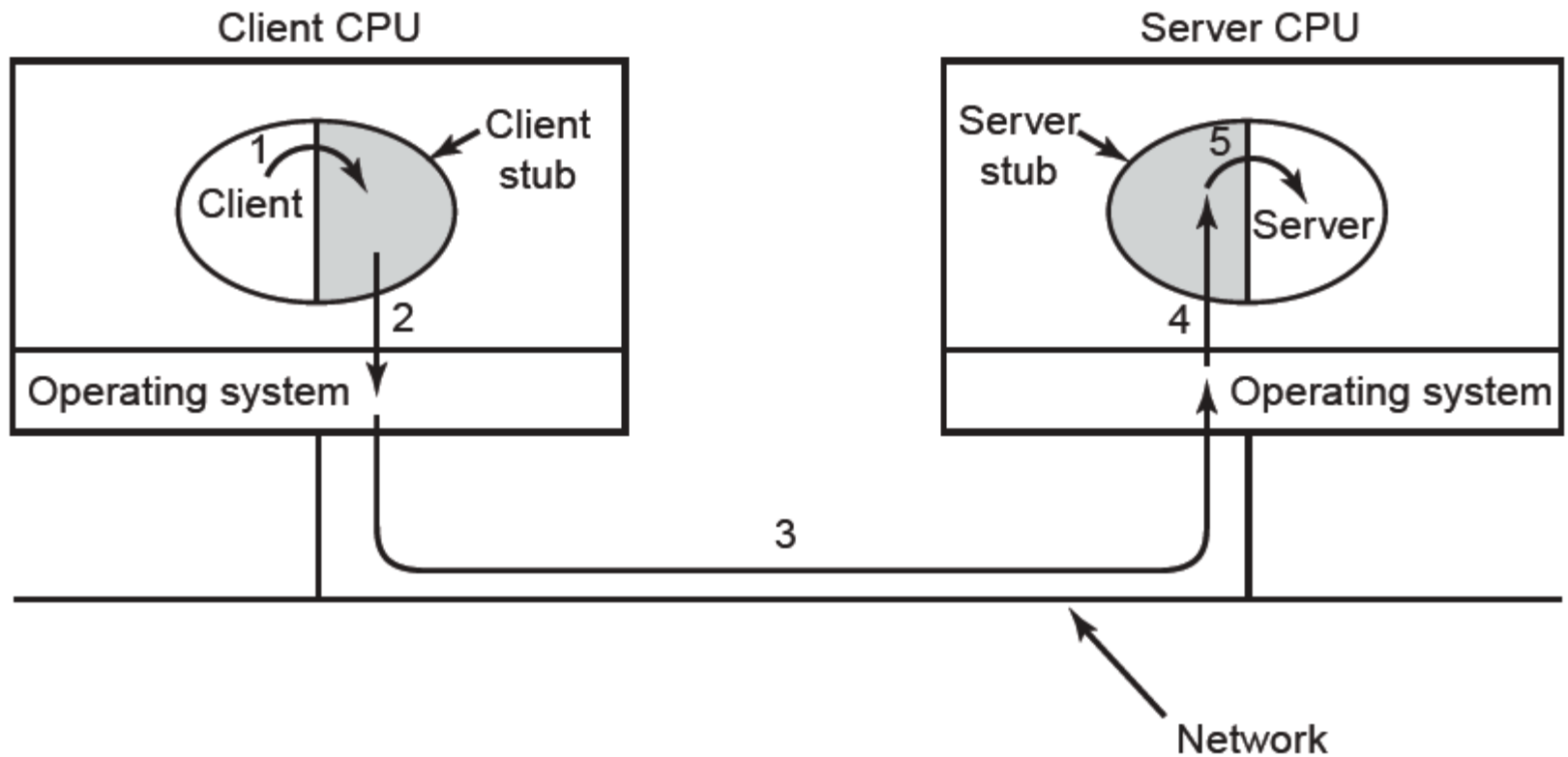
- UDP does not do
 - flow control, congestion control
 - retransmission of lost, bad segment
- UDP does
 - interface to IP, de-multiplexing, error detect
- UDP is good for
 - client-server situations
 - short messages, short replies
 - example: DNS



Remote Procedure Call (RPC)

- Sending message, getting reply, like proc call
- Allow progs call procedures on remote hosts
- Calling proc: **Client**. Called proc: **Server**
- Client prog bound w library proc represent server in client address space: **client stub**
- Server bound w procedure: **server stub**
- Message passing not visible to programmer

Remote Procedure Call (RPC)





Remote Procedure Call (RPC)

1. Client calls client stub
 2. C stub packing parameters (marshaling), make system call to send message
 3. Client OS send message to server
 4. Server OS pass packet to server stub
 5. Server stub calls server procedure with unmarshaled parameters
- Reply traces back in the other direction



Remote Procedure Call (RPC)

- UDP is good TL protocol to implement RPC
- Request, reply can be sent in 1 UDP packet
- Must keep timer in case packet lost
- May need multiple packets if $>$ max size
- Sequence # to match request, reply
- May not be idempotent (safe to repeat)
- DNS is idempotent, counter increment is not