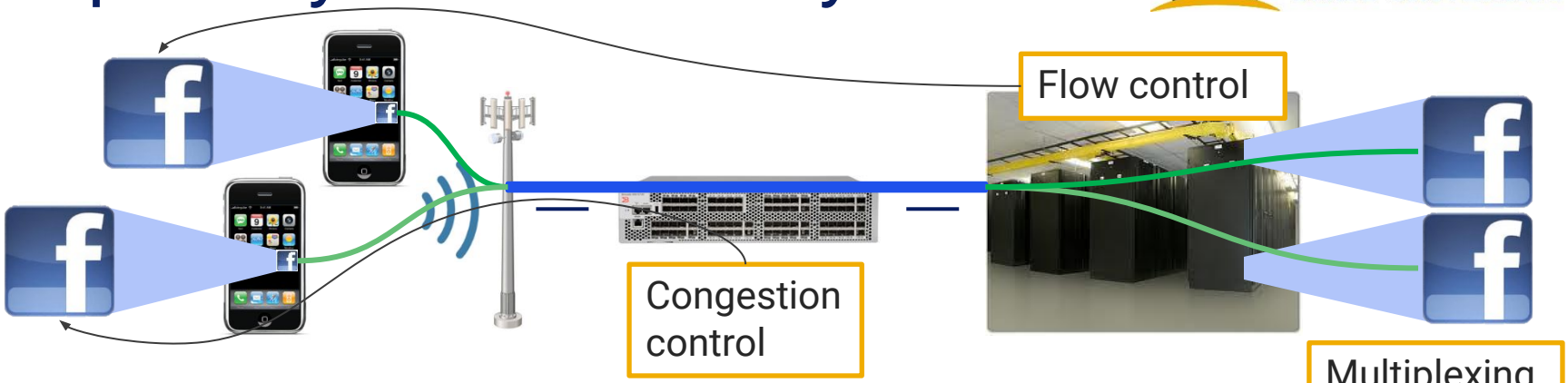
MONTANA
STATE UNIVERSITY

Chapter 3: Transport Layer

UDP

Transport layer functionality



Inter-process comm.
- Segmentation and reassembly
- Error checking

- Reliability
- In-order delivery

7: Application	⟷ Application services ⟷			7: Application
4: Transport	⟷ Logical communications ⟷			4: Transport
3: Network	⟷	3: Network	⟷	3: Network
2: Data link	⟷	2: Data link	⟷	2: Data link
1: Physical	⟷	1: Physical	⟷	1: Physical

Multiplexing via sockets

Multiplexing and Demultiplexing

MONTANA STATE UNIVERSITY

UDP

```
s = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
s.bind(('127.0.0.1', 5000))
```

http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

- Packets demuxed by:

  UDP sockets identified (`dst_IP:dst_port`) tuple

- Packets from `(153.90.118.46,3541)` and `(128.111.52.235,5502)` would go to the same socket

TCP

```
s = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
s.bind (('127.0.0.1', 80))
s.listen(1)
conn, addr = s.accept()
```

- Packet demuxed by:

  TCP sockets identified (`src_IP:src_port, dst_IP:dst_port`) four-tuple

  - Source port

- Packets from `(153.90.118.46,3541)` and `(128.111.52.235,5502)` would go different sockets


I need someone to connect.
...the world is changing. I can't help you. I saw the light and i-i it was terrible. The problem with the people is that
...when the time comes, every leaf turns the face to the sun. Everthing that converges must rise.
UDP
I need someone to connect.
TCP
Daniel Stori {turnoff.us}

UDP Packet structure

- Header information:
  - Source and destination ports for demux
  - Checksum for error detection
  - Packet length in bytes (headers and data)

Why use packet headers, as opposed to JSON?

Where are source and destination IP addresses?

Why not send fixed length packets?

Data in 16-bit words

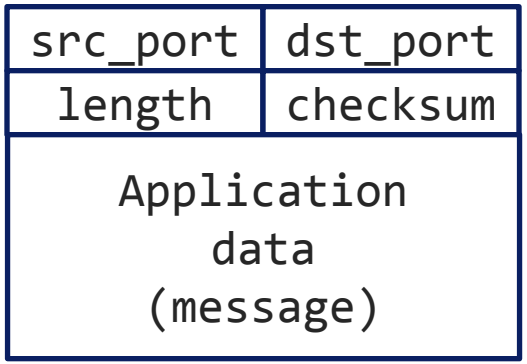
```
    1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
+   1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
=  1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
+                                   1
=   1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0
```

Wrap around: add overflow

Form one's complement

```
    0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1
```

UDP Packet Format



Error checking

1. Recompute at receiver
2. Add result to passed checksum
3. Error if not all zeros

UDP Checksum Exercise

- Consider the following bits

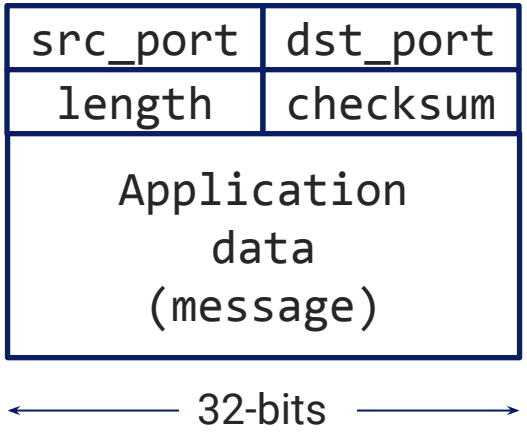
```
0001001110001000
0001011101110000
0000000000010000
1010010001111101
0001010110010101
0001001011000110
0001000011010100
1000100101100101
0000010010101100
0110100100111010
```

Identify the fields of a UDP packet

Is the UDP packet correct?

In practice UDP checksum includes fields from IP header

UDP Packet Format




```
0001001110001000 0001011101110000
0000000000010000 1010010001111101
0001010110010101 0001001011000110
0001000011010100 1000100101100101
0000010010101100 0110100100111010
```

UDP header and pseudo IP header

MONTANA STATE UNIVERSITY

IPv4 pseudo header format

Offsets	Octet	0								1								2								3							
Octet	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	0	Source IPv4 Address																															
4	32	Destination IPv4 Address																															
8	64	Zeroes								Protocol								UDP Length															
12	96	Source Port																Destination Port															
16	128	Length																Checksum															
20	160+	Data																															

- In practice UDP checksum includes routing information
- RFC 768 defines UDP checksum calculation
- Includes fields in the pseudo IPv4 (or IPv6) header available to in the UDP socket
- RFC 768: **“This information gives protection against misrouted datagrams.”**

Which transport protocol where?

MONTANA STATE UNIVERSITY

Application	Application-Layer Protocol	Underlying Transport Protocol	
Electronic mail	SMTP	TCP	Email size > UDP packet size
Remote terminal access	Telnet	TCP	In order delivery
Web	HTTP	TCP	Large pages
File transfer	FTP	TCP	Large file sizes
Remote file server	NFS	Typically UDP	High rate of transfer
Streaming multimedia	typically proprietary	UDP or TCP	Fixed data rate
Internet telephony	typically proprietary	UDP or TCP	Fixed data rate
Network management	SNMP	Typically UDP	No connection delay
Name translation	DNS	Typically UDP	No connection delay

User Datagram Prot. (UDP)

- "No frills," "bare bones" Internet transport protocol
- Based on "best effort" network model
  - UDP segments can be lost
  - Or delivered out-of-order
- Connection-less
- No support for:
  - Flow control
  - Congestion control
  - In-order delivery
  - Reliability

Advantages of UDP:

- Immediate transmission
  - No connection establishment
  - No delayed transmission
- Fixed/immediate sending rate
  - No ramp up
- Lower memory requirements
  - No connection state
- Small packet overhead
  - 8B for UDP vs 20B for TCP

Disadvantages:

- Controversial in streaming applications
- Blocked by many firewalls


MONTANA
STATE UNIVERSITY