

User Datagram Protocol (UDP)

Reading: Chapter 25

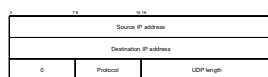
UDP

- Unreliable, connectionless datagram service
- Desirable for:
 - Short transactions, avoiding overhead of establishing/tearing down a connection
 - DNS, time, etc
 - Applications withstanding packet losses but normally not delay
 - Real-time audio/video
- Pretty simple protocol
 - Multiplexing application processes

Packet format



UDP packet format



Pseudo packet header

UDP packet header

- Source port number
- Destination port number
- UDP length
 - Including both header and payload of UDP
- Checksum
 - Covering both header and payload, and the pseudo header

Implementation

- **Data structures**
 - Passing payload
 - UDP datagram
 - Interface to application layer
 - Interface to IP
- **Functions**
 - Sending UDP datagrams
 - Receiving UDP datagrams

Fall 2004

FSU CIS 5930 Internet Protocols

5

Passing payload

*msg_name
msg_namelen
*msg_iov
msg_iovlen
msg_control
msg_controllen
msg_flags

struct msghdr

*iov_base
iov_len

struct iovec

Fall 2004

FSU CIS 5930 Internet Protocols

6

UDP datagram

source
dest
len
check

struct udphdr

uh
saddr
daddr
*iov
wcheck

struct udpfakehdr

Fall 2004

FSU CIS 5930 Internet Protocols

7

Interface to application layer

name
close
connect
disconnect
ioctl
sendmsg
recvmsg
get_port
.....

struct proto

Fall 2004

FSU CIS 5930 Internet Protocols

8

Interface to IP

handler
err_handler
next
protocol
name

struct inet_protocol

Sending UDP datagram

- **udp_sendmsg()**
 - Forming udpfakeudp data structure
 - Msg, or connection state
 - Processing control messages
 - ip_cmsg_send()
 - Routing
 - If routing cache exists, use it
 - ip_route_output()
 - Passing to IP
 - ip_build_xmit() with callback function:
 - udp_getfrag()/udp_getfrag_nosum()
 - Getting the real payload

udp_getfrag()

- **Passing the payload to IP**
- **If it is the last segment (containing UDP header)**
 - Computing checksum for leftover data, UDP header, UDP faked header
 - Inserting checksum into UDP header
- **Passing data (one segment) to IP**
 - csum_partial_copy_fromiovecend()

Receiving UDP datagram

- **udp_rcv()**
 - Computing checksum, make sure matching the one in the packet
 - If multicasting/broadcasting packet
 - udp_mcast_deliver()
 - Checking to which socket queue the packet belongs
 - udp_v4_lookup()
 - Putting packet into socket queue if found queue
 - udp_queue_rcv_skb()
 - sock_queue_rcv_skb()
 - Otherwise, sending ICMP

udp_mcast_deliver()

- Looking up queues to which the packet belongs
 - Similar to `udp_v4_lookup`
 - But it needs to find all the socket queues
- For each matched queue, passing the packet
 - `skb_clone()`
 - `udp_queue_rcv_skb()`

udp_recvmmsg()

- Used to receive packet from socket queue by users
- Removing an `skb_buff` from queue
- Returning the corresponding payload

- Some helpful functions
 - `skb_rcv_datagram()`
 - `skb_copy_datagram_iovec()`
 - `skb_free_datagram()`