

---

# UDP - Transport Layer

TCP/IP class

# intro

---

- ◆ UDP - user datagram protocol
- ◆ RFC 768
- ◆ UDP == “ip with ports”
- ◆ client/server both “bind” to a port and you send messages/recv them from a port
- ◆ port is 0..64k-1, unsigned short
- ◆ well-known ports associated with servers

# intro, cont

---

- ◆ UDP provides unreliable connectionless delivery
- ◆ there is a checksum, but it is configured on/off per host - has been typically off in the past (that may be changing)
- ◆ checksum is over ip pseudo header, udp header, and data
- ◆ 0 put in checksum field in header to compute, if 0 is sent, means csum off

# udp encapsulation

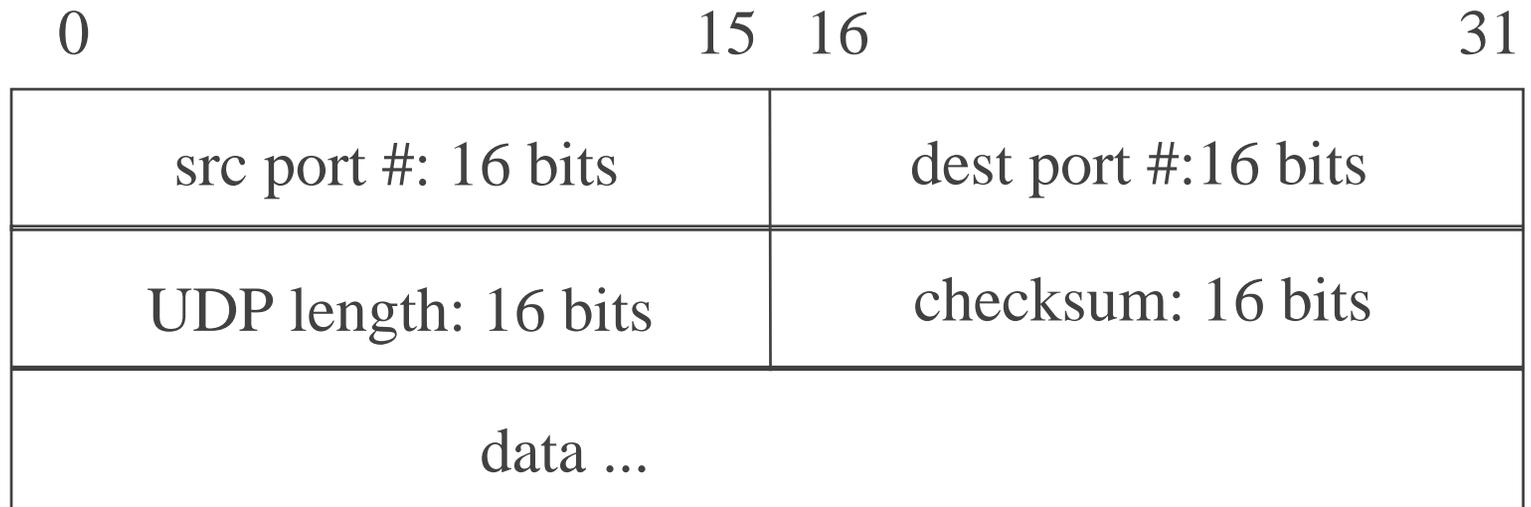
---



8 bytes (no options)

# udp header

---



length includes header, minimum is 8

# psuedo-header

---

- ◆ udp code must tack on ip pseudo-header and optionally perform checksum over it
  - 32 bit source ip address
  - 32 bit dest. ip address
  - 1 byte zero
  - 1 byte proto = 17 (UDP)
  - 2 bytes UDP length == 12 bytes in all
- ◆ idea is to include IP addresses (etc) into checksum in order to prove that dest is correct

# apps that use UDP

---

- ◆ statistically, broadcast oriented
  - routing daemons (rip and routed)
  - rwho
- ◆ multicast apps
  - need multicast, tcp can't do that
  - need sequencing, timestamp, udp doesn't do that
- ◆ message-oriented
  - snmp
  - dns

# apps, cont

---

- ◆ NFS - distributed file system - why?
  - one reason is that TCP needs too much kernel state (memory) for connections (virtual circuit problem #1)

# study questions

---

- ◆ so if UDP over localhost OR on a local network works fine, why not just use it for a file transfer?
- ◆ how can you find out if UDP checksums are on, off on your host? Can you find a way to turn them on/off?