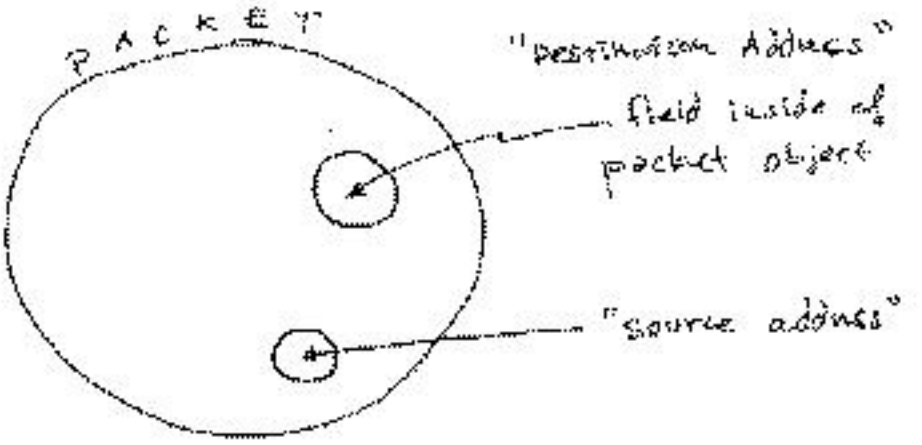
TCP/IP :
multiplexing, demultiplexing, addressing

Basic idea in Ethernet, IP, TCP, UDP:
structure application data (messages)
into addressed units
(frame, packet, datagram, etc.)



Destination address could refer to
- one host in the Internet (IP)
- one NIC (Ethernet)
- one application program (TCP or UDP)
- multiple places (broadcast, multicast)

Source address is so that recipients
can know where how to address a reply.

Process of addressing, directing traffic of many packets is multiplexing (demultiplexing)

Addresses in the TCP/IP/Ethernet hierarchy

Ethernet: 48-bit number, built into the NIC, supposedly unique for the entire world.

- move your computer, your Ethernet address goes with you
- buy a new NIC, your Ethernet address changes

MORE GENERIC TERM:
MAC ADDRESS
(ethernet, 802.11 "wifi", etc)

IP (version 4): 32-bit number, assigned by some administrator

- change NICs, change computers, your address stays the same
- Everyone in the Internet can know and use your IP address for packets.

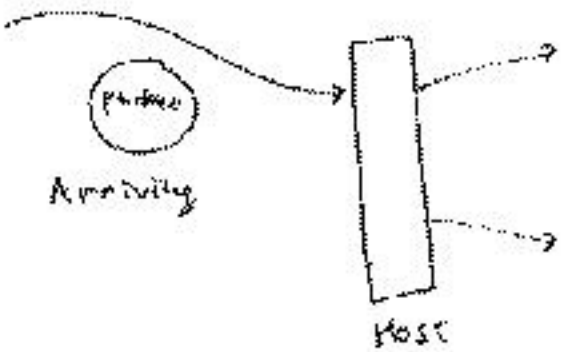
(version 6): 128-bit number

UDP :

address =

IP - address ⊕ Port Number
(32 bits) (16 bits)

Port numbers are needed to distinguish applications using UDP on the same host



which one(s) get the packet ?

⇒ need to know port #

How ?

some are "well-known"

80 — web Servers

25 — email

TCP:

address =

IP-address ⊕ Port Number ⊕ Session Number

Session numbers are to distinguish different runs (different instances) of the same application

BUT sometimes view by some people

address =

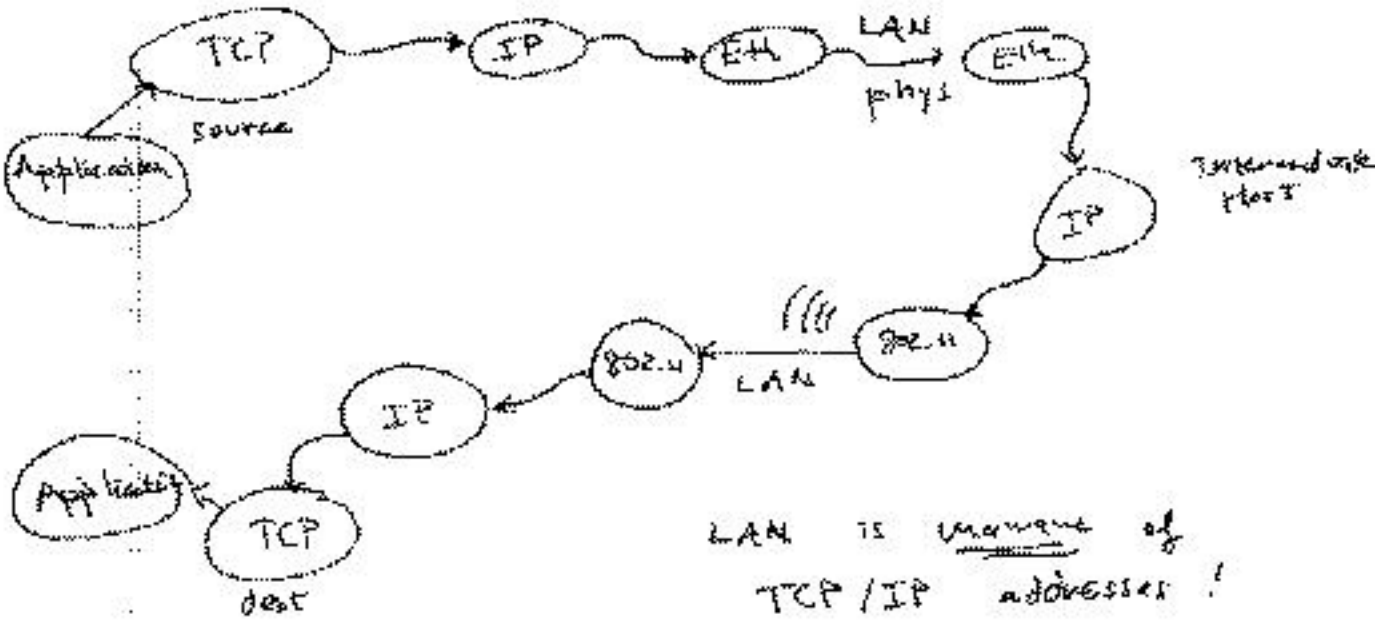
IP-address ⊕ Port number ⊕
session Number ⊕ Byte Number

Byte number indicates where data in packet should be placed in a "stream" (like position inside of a file).

Book-keeping, Translation, Forwarding

Obviously this jumble of different address types means that software has to keep track of addresses, what they are associated with, and how to convert addresses as data flows through the Internet.

Flow



LAN is unaware of TCP/IP addresses!

Programmer view