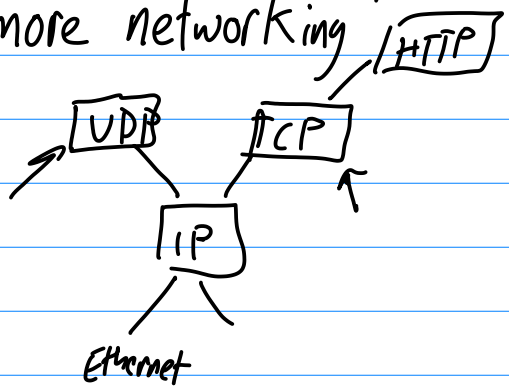
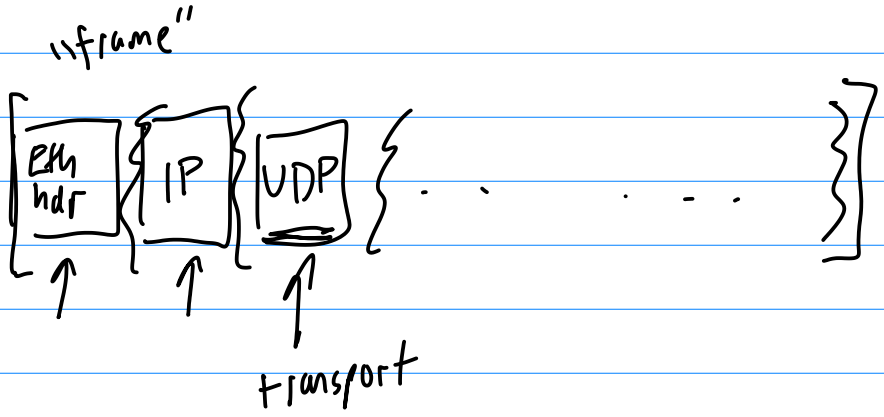
CS3214 lecture #24 "more networking"



layers

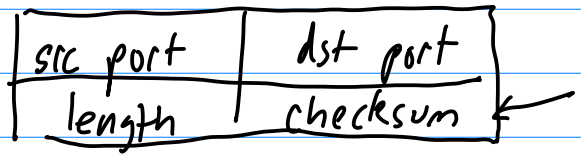
	app	
	transport	←
L3:	network	IP
L2:	link	Ethernet
	phys	

encapsulation

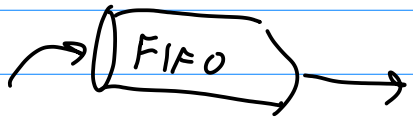


proto src ip, src port, dst ip, dst port

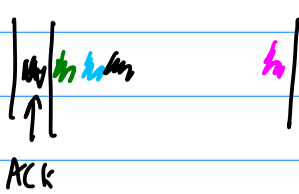
UDP
- unreliable
- connectionless



TCP
- point-to-point
- reliable "stream" guaranteed delivery: ACK (retransmits)
  in-order

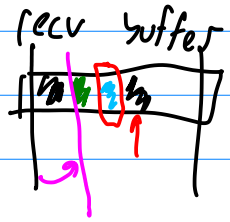


send buffer
"retransmission buffer"

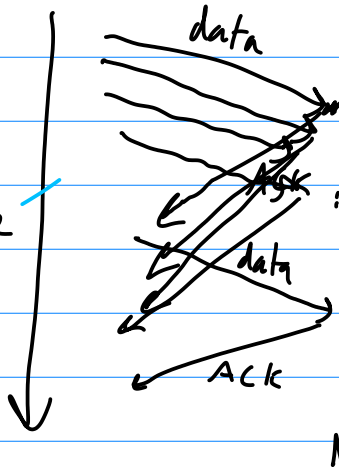


pipelined

recv buffer

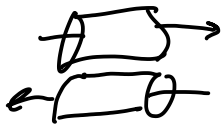


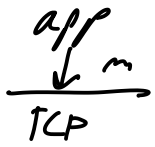
sliding window



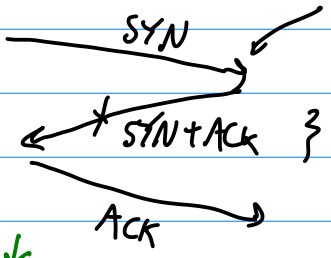
RFC

NACK

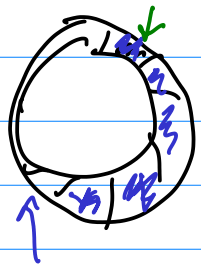




- full duplex (bi-directional in same connection)
- connection ... handshake
  3-way handshake

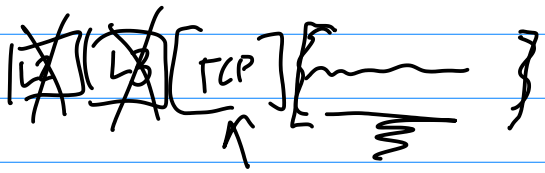


- flow controlled
  sender will not overrun recv.



- congestion controlled
  ↑
  "TCP saved the Internet"

TCP header



RFC

- src port dst port
- seq num
- ACK #
- flags (SYN, FIN, etc.)
- window size
  checksum
  (options)

The Socket API - first BSD UNIX 1981
- general IPC "unix socket"
- network: UDP/TCP between processes

Java.net.Socket
Python socket
:

sockets are fd's: read/write/close

UDP

server

fd ← socket()
bind()
recvfrom()
sendto()

client

fd = socket()
sendto(...)
recvfrom()

Socket
connect() ← not real connection
send()
recv()

all OS stuff → socket(int domain, int type, proto)

type: SOCK_DGRAM, STREAM

domain: PF_INET, PF_UNIX, PF_INET6

proto: IPPROTO_UDP, -TCP

all OS stuff → bind(int sockfd, {struct sockaddr *myaddr, int len});

myaddr: IP address, port

TCP:

server

socket()
bind()
listen()
accept()

client

socket()
connect() ← initiate SYN
write(fd)
read(fd)
close(fd) ← FIN

OS start queuing requests

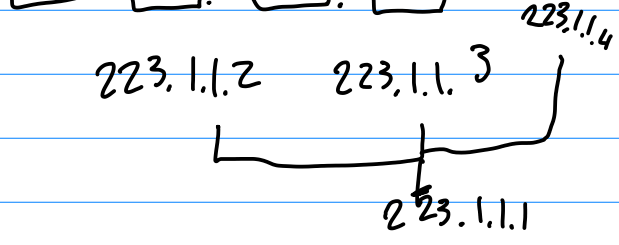
read(fd)
write(fd)

block until a client is pending
return a new socket with client connection

IPv4 vs IPv6

IP address per interface

Subnet: connected interfaces
share common prefix
no routing between subnet



CIDR prefixes

223.1.1.0/24 223.1.1.0 - 223.1.1.255

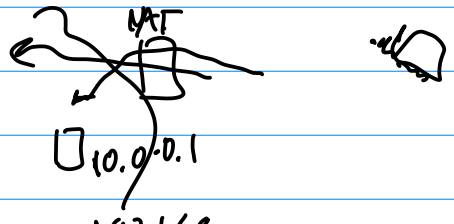
netmask 255.255.255.0

223.1.1.0/30 223.1.1.0 - .3

netmask 255.255.255.252

only 4 billion address — address exhaustion

NAT network address translation



IPv6 ~1990s 128 bits

↑ slow adoption

IPv4 only
IPv4 + IPv6
IPv6 only
} Support all

let system tell you

man pages { getaddrinfo()
getnameinfo()

transport layer + socket interface

app layer: HTTP