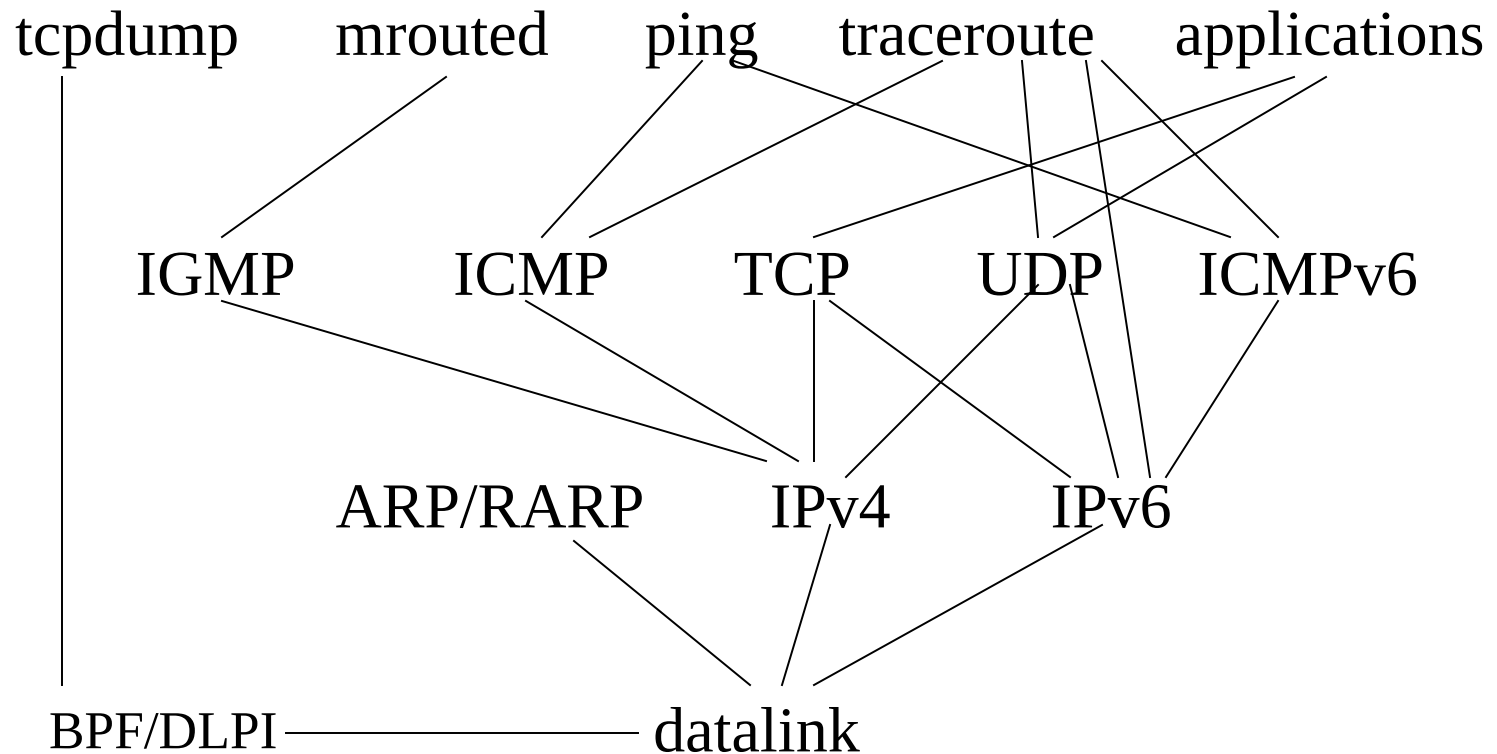


Transport Layer: TCP and UDP

- Overview of TCP/IP protocols
- Comparing TCP and UDP
- TCP connection: establishment, data transfer, and termination
- Allocation of port numbers
- Size matters: MTU, datagram, MSS, buffer
- Standard Internet services and applications
- Debugging techniques and tools

Overview of TCP/IP Protocols

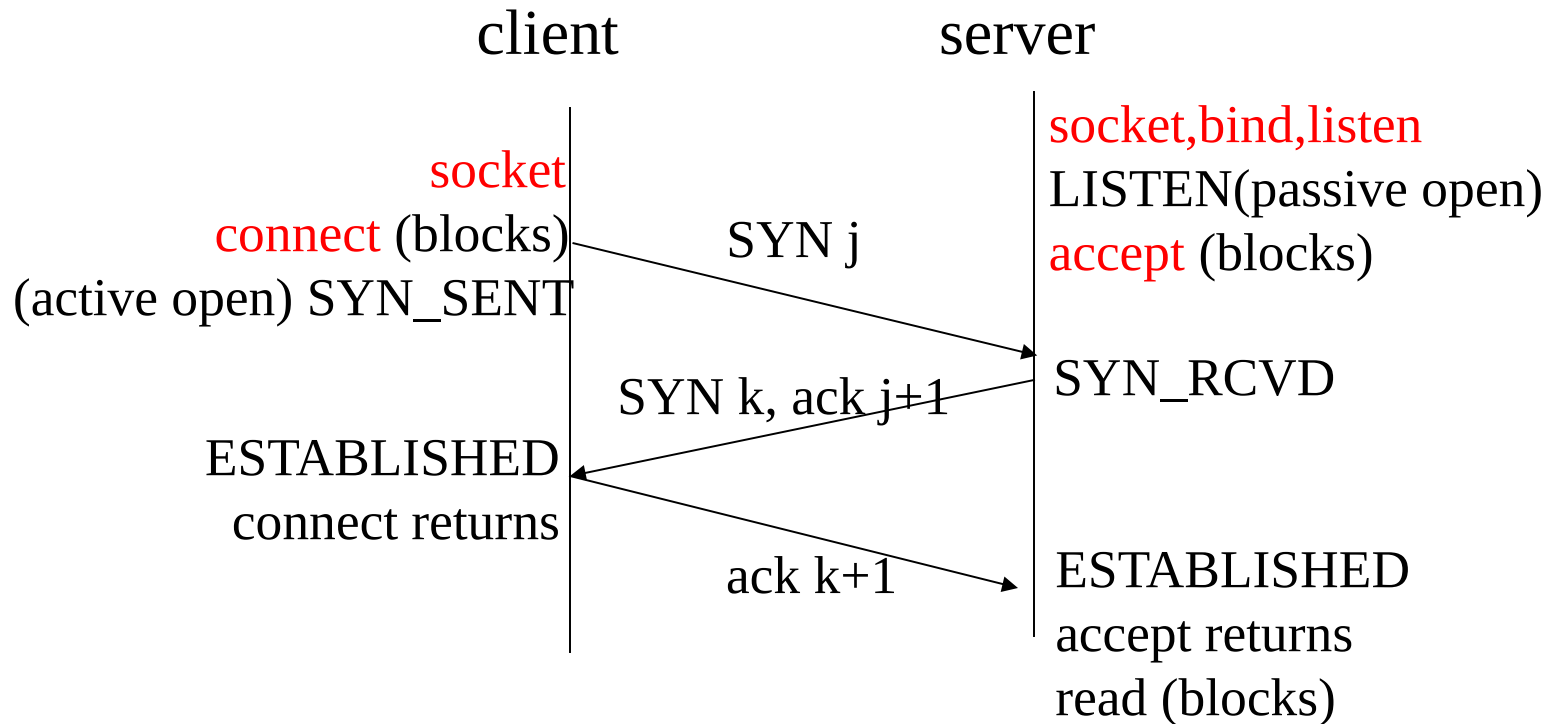


Comparing TCP and UDP

	TCP	UDP
Binding between client and server	Yes (connection-oriented)	No (connection-less)
Data	Byte-stream	Record
Reliability	Yes (ack, time-out, retx)	No
Sequencing	Yes	No
Flow control	Yes (window-based)	No
Full-duplex	Yes	Yes

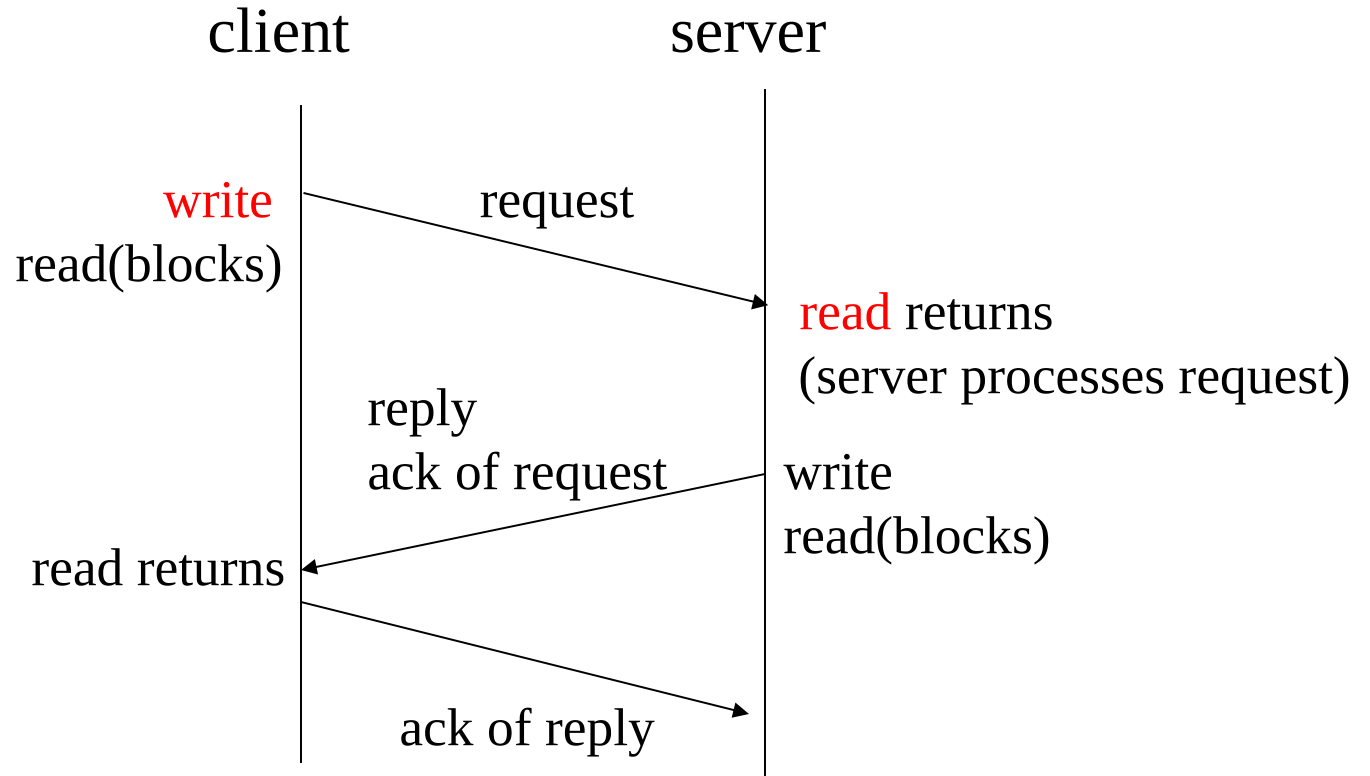
TCP Connection: Establishment

Three-way handshake



TCP options (in SYN): MSS (maximum segment size) option, window scale option (advertized window up to 65535×2^{14} , 1GB), timestamp option (the latter two: long fat pipe options)

TCP Connection: Data Transfer



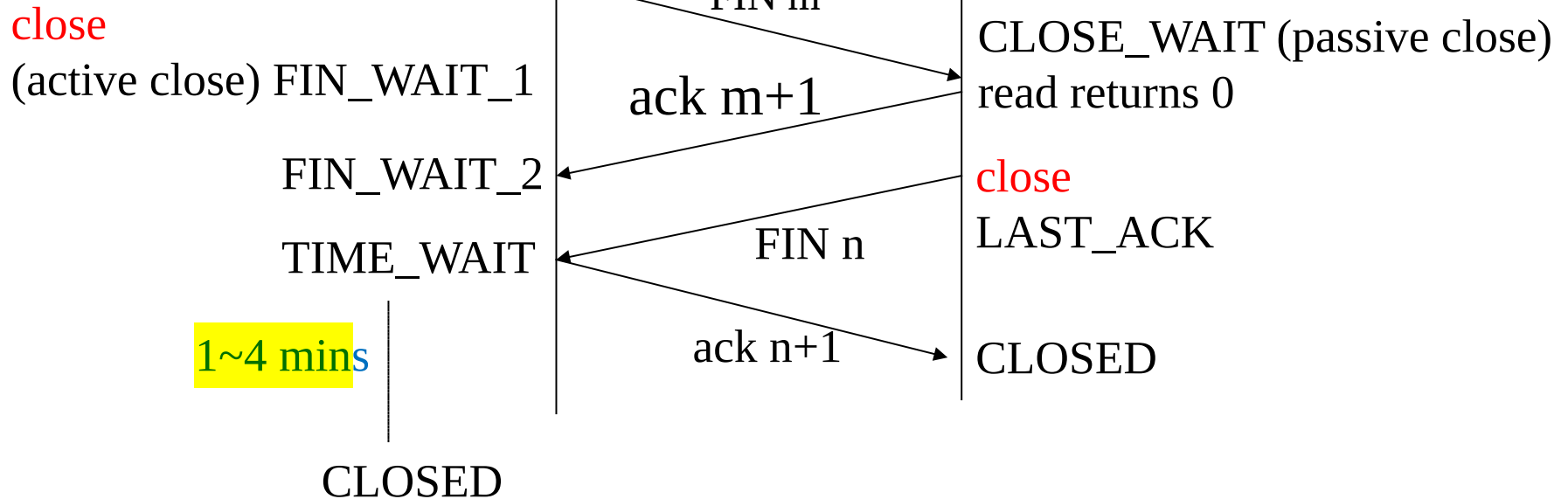
TCP Connection: Termination

Four-way handshake



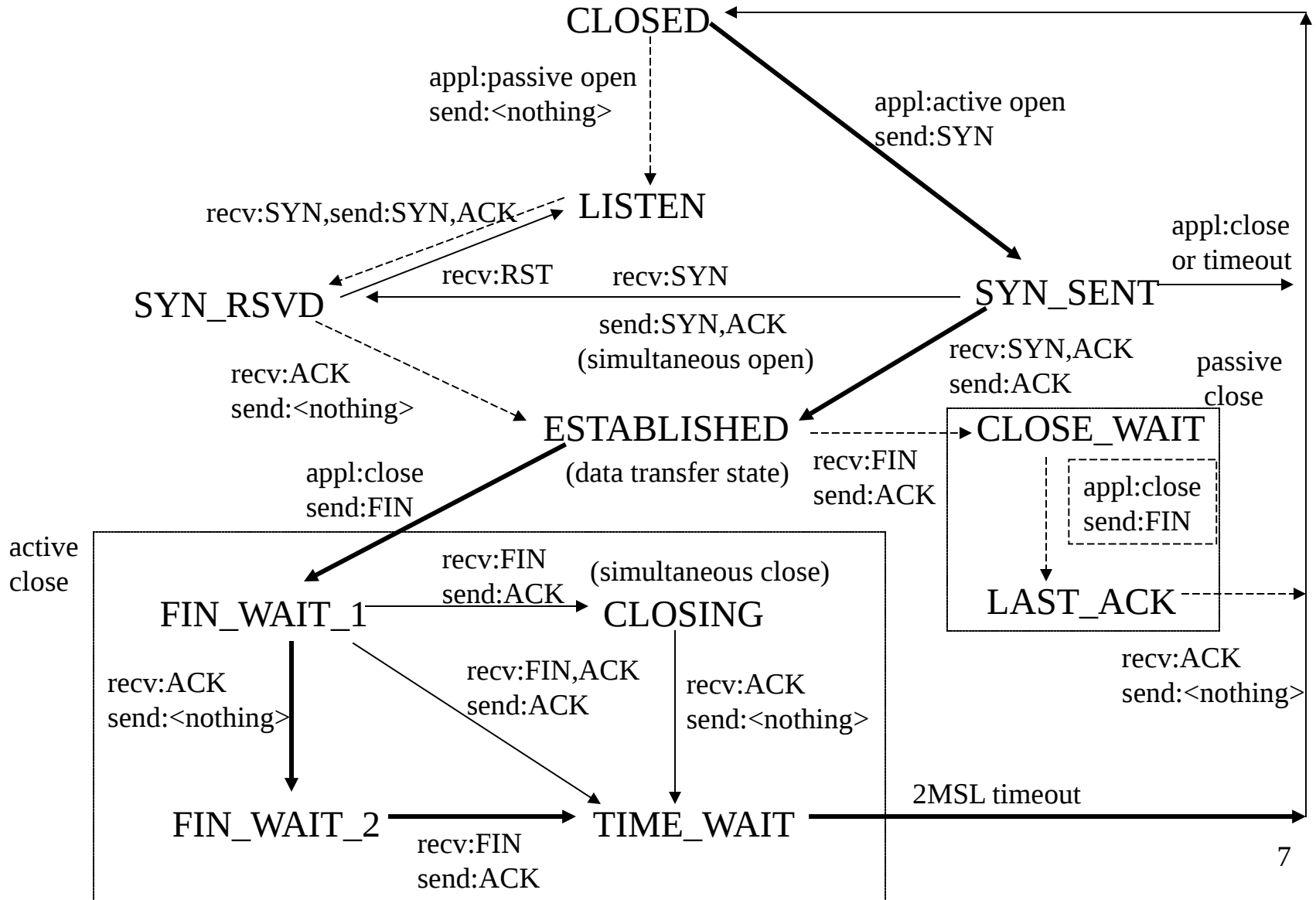
Client/Server

Server/Client

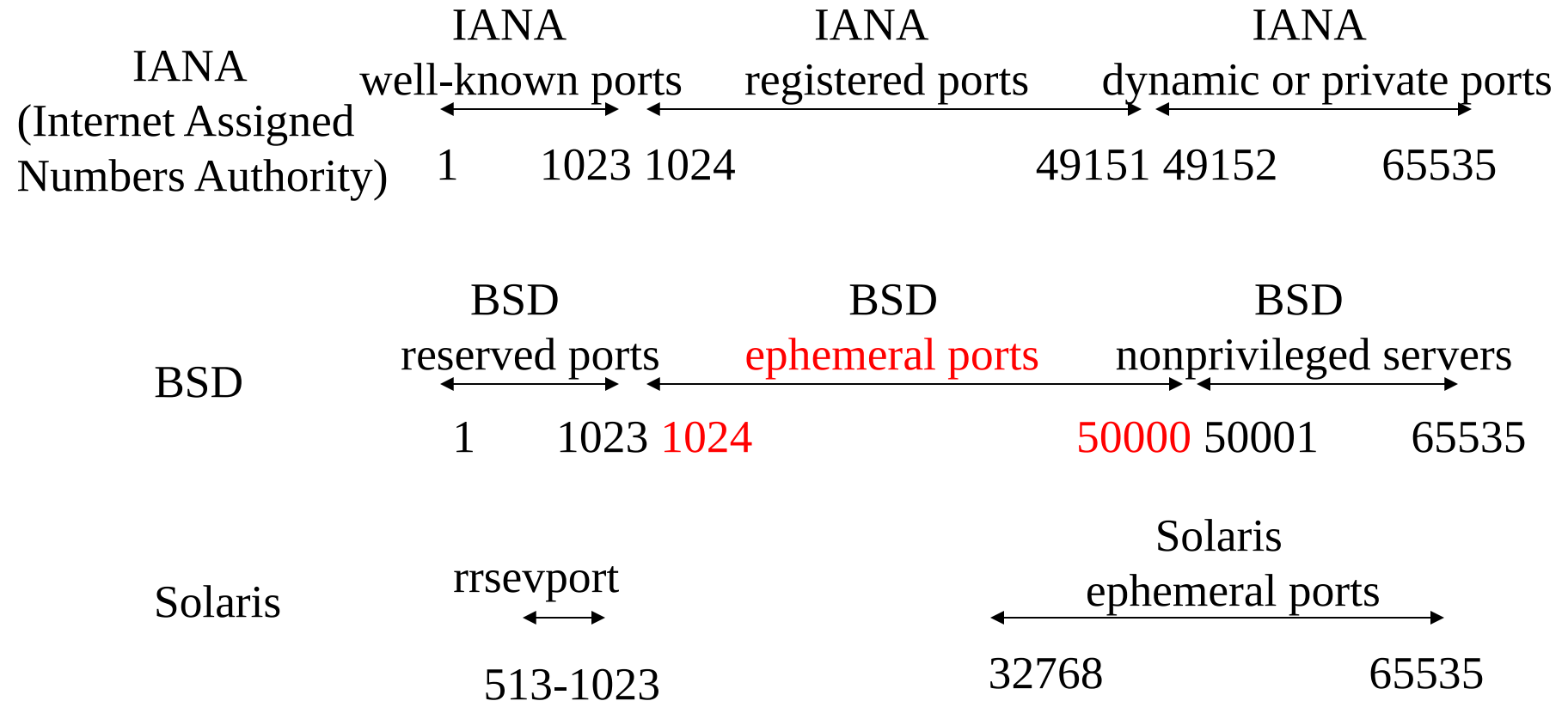


TIME_WAIT to allow old duplicate segment to expire for reliable termination
(the end performing active close might have to retx the final ACK)

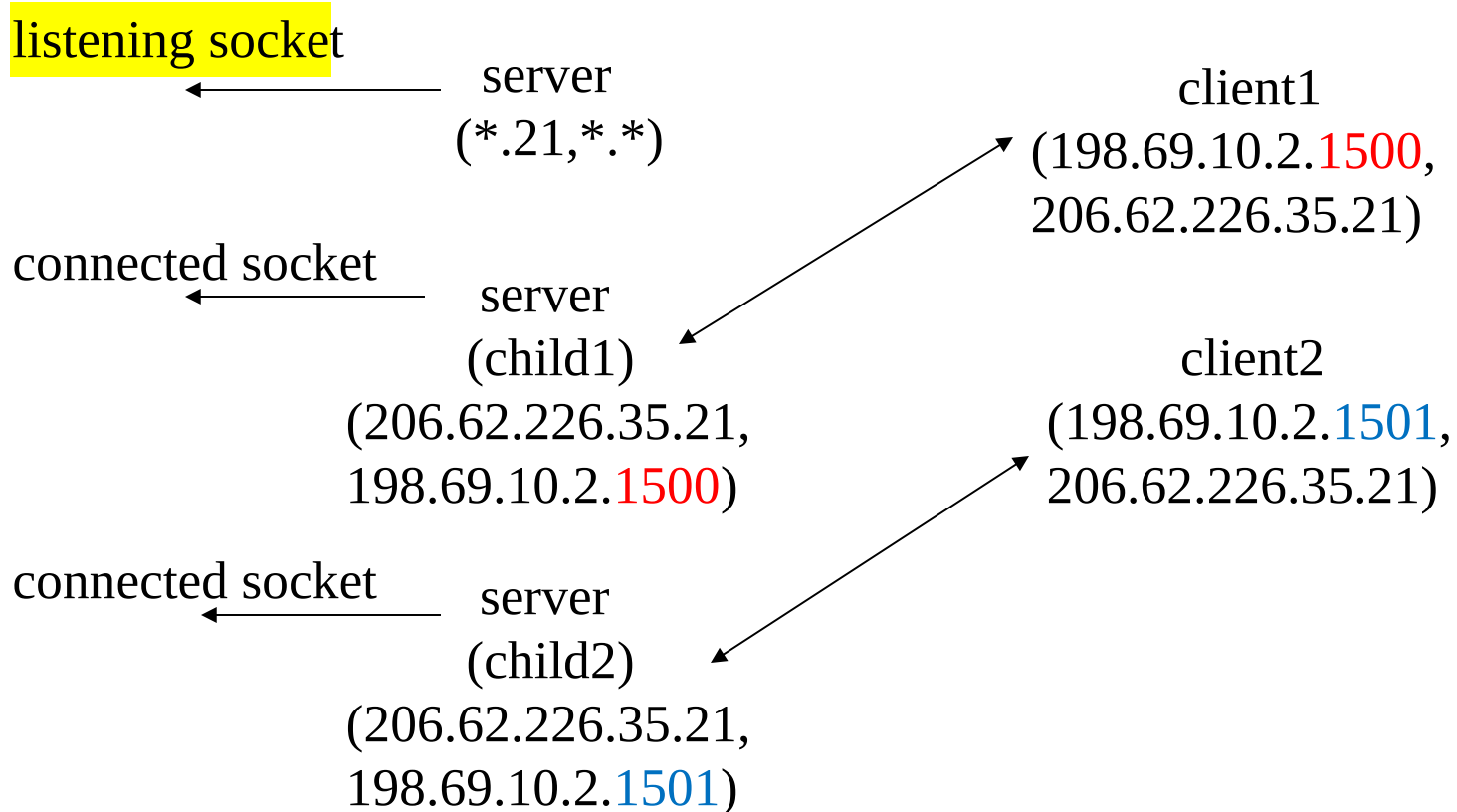
TCP State Transition Diagram



Allocation of Port Numbers




Multiple Sockets with the Same Port (in Concurrent Server)



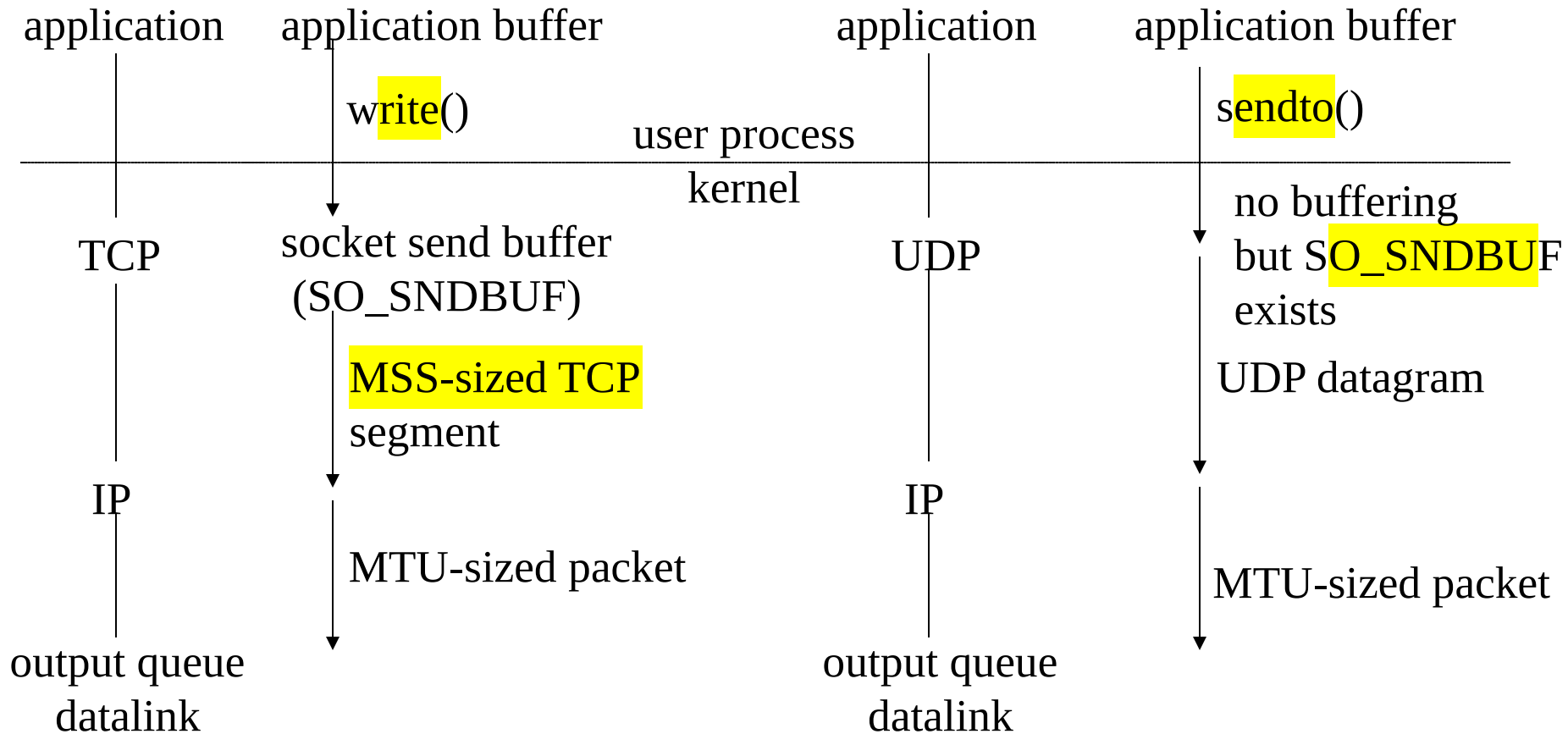
All TCP segments destined for port 21, with *socket pairs* different from `(206.62.226.35.21,198.69.10.2.1500)` and `(206.62.226.35.21,198.69.10.2.1501)`, are delivered to the original server with the listening socket.

Size Matters:

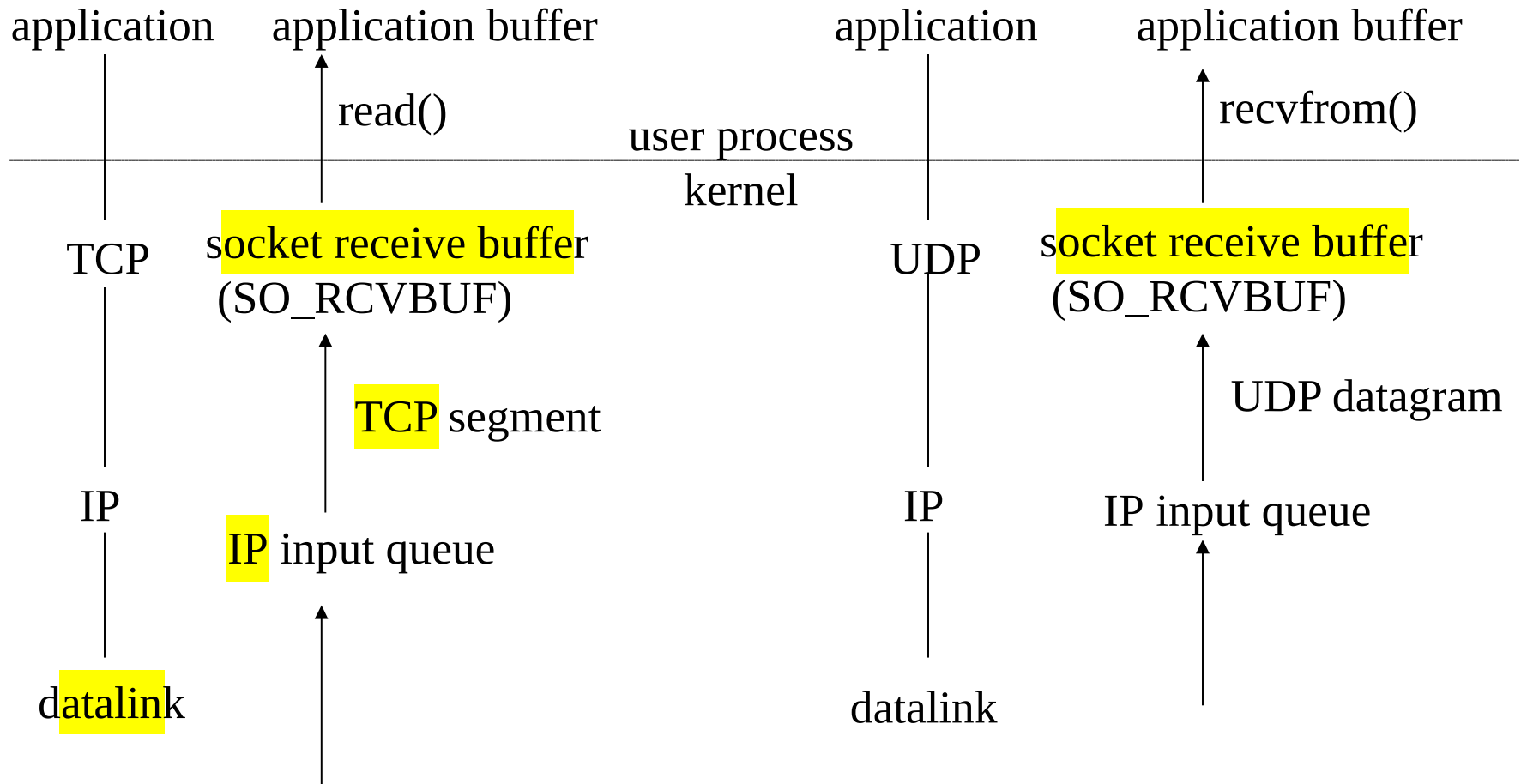
MTU, datagram, TCP MSS, buffer

- *Link* MTU (maximum transmission unit): Ethernet MTU: 1500 bytes, PPP MTU: configurable
- *Path* MTU: the smallest link MTU in the path, can be discovered by IP DF (don't fragment) bit
- Maximum IP datagram: 65535 (IPv4), 65575 (IPv6) (IPv6 has 32-bit jumbo payload option), minimum IP reassembly buffer size (576) 
- TCP MSS (maximum segment size): actual value of reassembly buffer size, often the link MTU minus IP and TCP headers, to avoid fragmentation

TCP Output and UDP Output



TCP Input and UDP Input



Standard Internet Services and Applications

- Standard services provided by *inetd* daemon:
echo/port7/RFC862, discard/port9/RFC863,
daytime/port13/RFC867, chargen/port19/RFC864,
time/port37/RFC868
- tested by “telnet machine service”, service mapped
by /etc/services
- Common application types: diagnostic, routing
protocol, datagram, virtual circuit, etc.

Protocol Usage of Various Common Applications



Application	IP	ICMP	UDP	TCP	
Ping	X	X	X	X	
Traceroute		X			
OSPF			X		
RIP					
BGP					
BOOTP			X		
DHCP			X		
NTP			X		
TFTP			X		
SNMP			X		
SMTP				X	
Telnet				X	
FTP				X	
HTTP				X	
NNTP				X	
DNS			X	X	
NFS			X	X	
RPC			X	X	

Debugging Techniques and Tools

- System call tracing: *truss* (in SVR4), *ktrace* & *kdump* (in BSD) (Note that *socket* is a system call in BSD, while *putmsg* and *getmsg* are the actual system calls in SVR4)
- *sock* developed by W.R. Stevens: used to generate special case conditions, as stdin/stdout client, stdin/stdout server, source client, sink server
- *tcpdump*: dump packets matching some criteria
- *netstat*: status of interfaces, multicast groups, per-protocol statistics, routing table, etc.
- *lsof* (list open files): which process has a socket open on a specified IP address or port