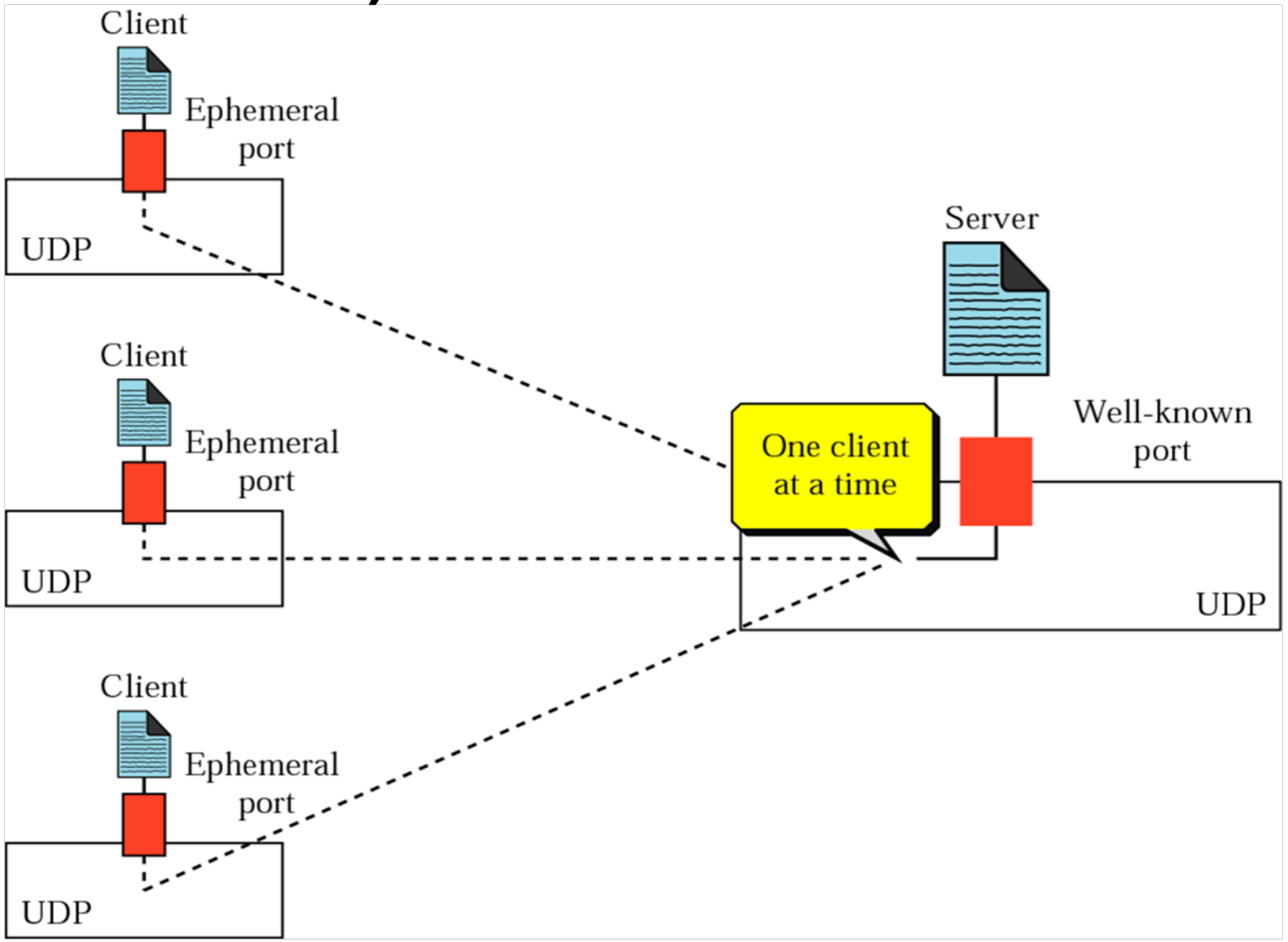
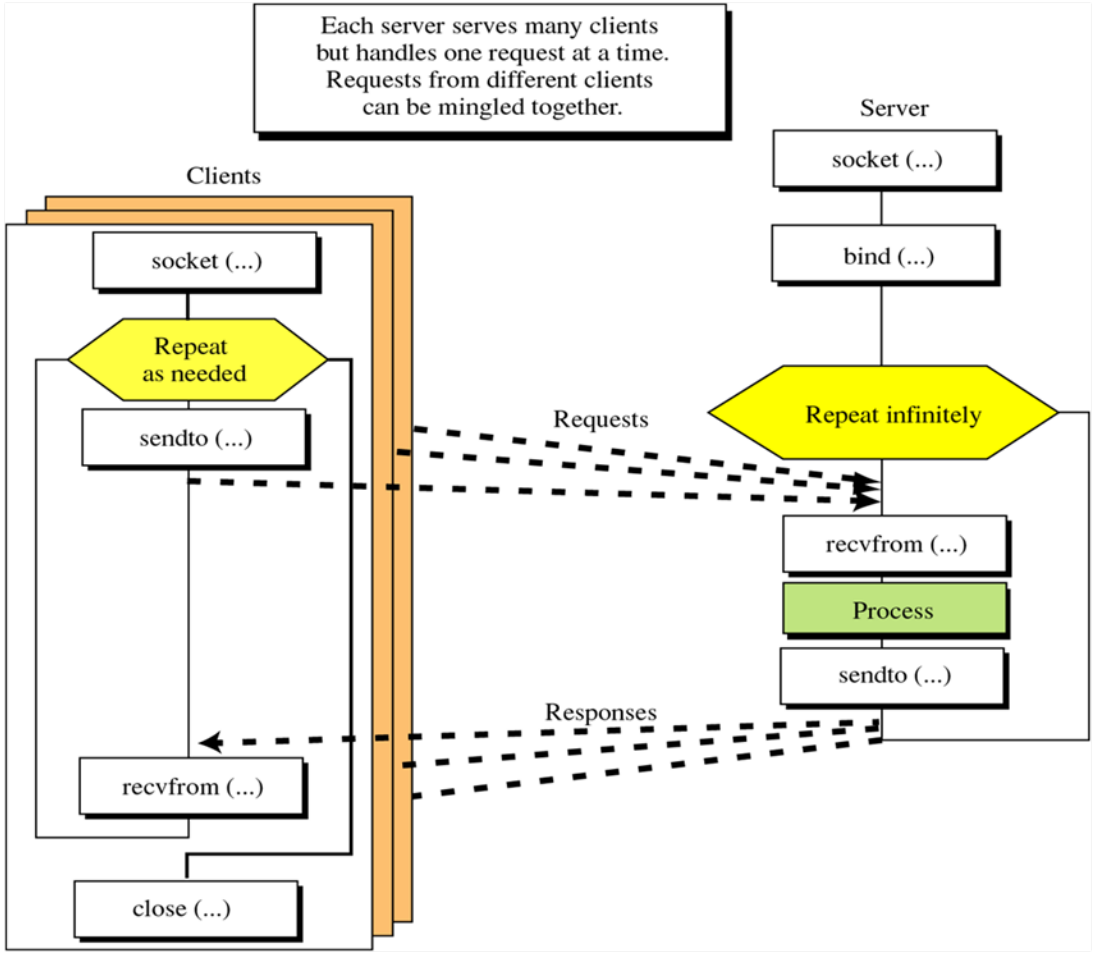
Sockets med SDL net

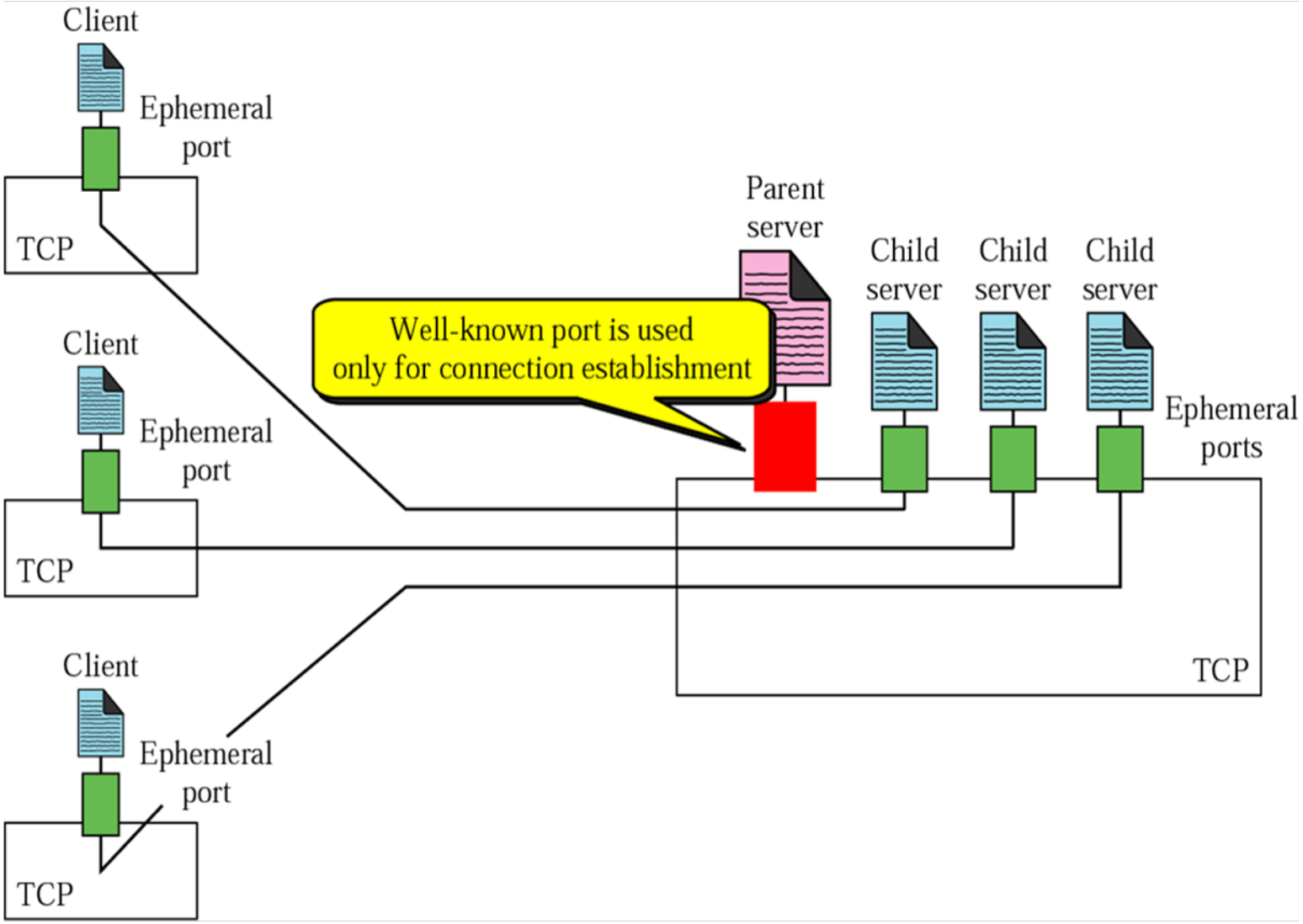
UDP, ”connection less”



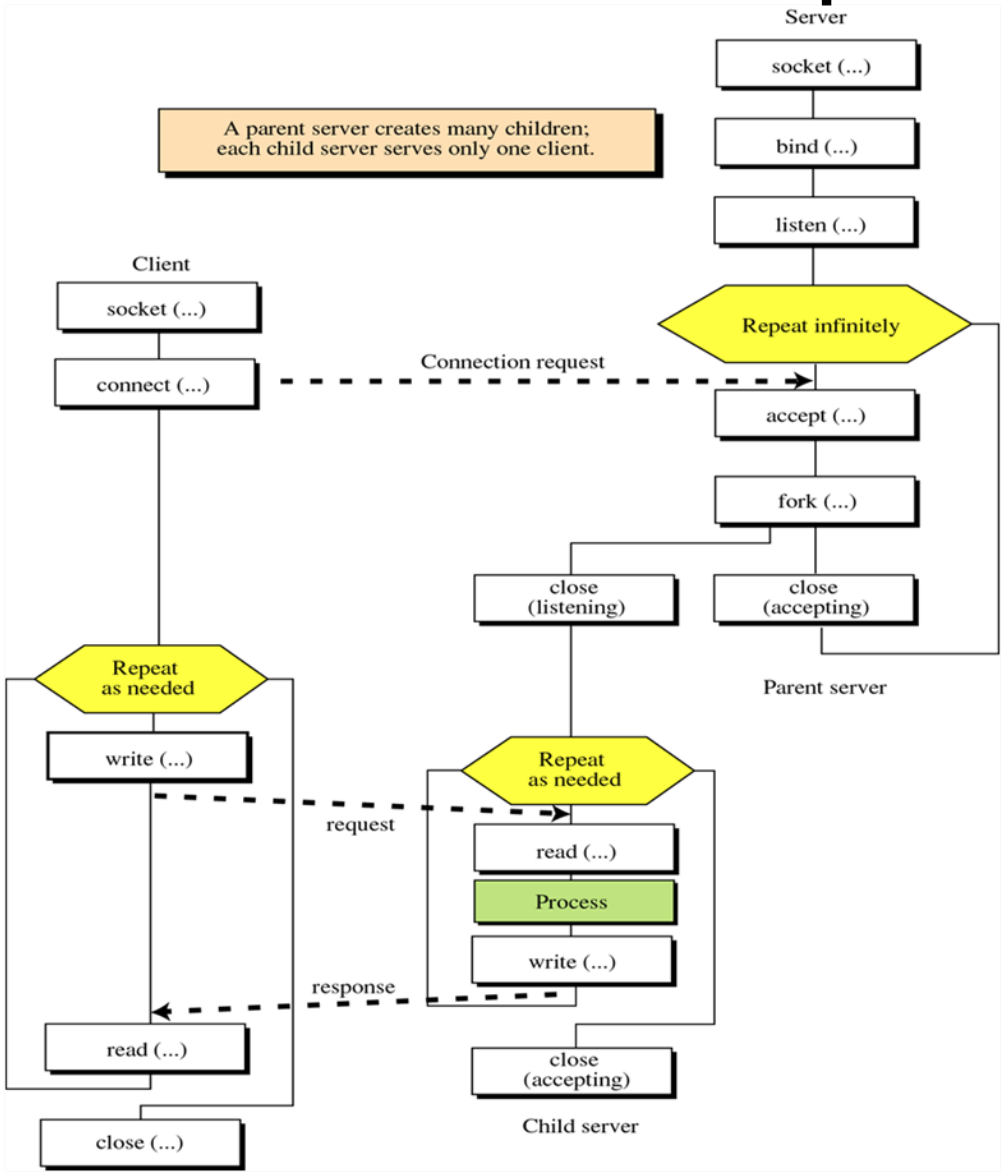
”One at a time”


Each server serves many clients
but handles one request at a time.
Requests from different clients
can be mingled together.
Server
socket (...)
bind (...)
Repeat infinitely
recvfrom (...)
Process
sendto (...)
Clients
socket (...)
Repeat
as needed
sendto (...)
recvfrom (...)
close (...)
Requests
Responses

TCP, ”connection oriented”



Hantera flera klienter parallellt



Hantera flera klienter parallellt

- När SDLNet_TCP_Recv()/Send() anropas blockeras exekverande tråd tills data anlänt/sänts!
- Lösning - kontrollera om det finns något att läsa innan Recv anropas:
  SDL_SocketReady(socket)
  Exempel: tcpmultiserver.c I “SDL_Net_demos” (en singeltrådad chatserver)
- Alternativ:
  Hantera varje klient i en separat tråd

IPaddress

- typedef struct
  {
  Uint32 host; /* 32-bit IPv4 host address */
  Uint16 port; /* 16-bit protocol port */
  } IPaddress;

- SDLNet_ResolveHost(&ip, host_name, host_port)
- ResolveHost gör en DNS-förfrågan

TCP-klient med SDL

- Initiera SDLNet, SDLNet_Init()
- …
- SDLNet_ResolveHost(…)
- Koppla upp:
  socket= SDLNet_TCP_Open(&server_ip)
- Läs/skriv data, SDLNet_TCP_Recv()/Send()
- Stäng socketen, SDLNet_TCP_Close(…)
- …
- Avsluta SDLNet, SDLNet_Quit()

TCP-server med SDL

1. …
2. SDLNet_ResolveHost(…) – lokalt på server (NULL)
3. Öppna en socket för inkommande uppkopplingar: server_sock= SDLNet_TCP_Open(…)
4. Vänta på inkommande uppkoppling: client_sock=SDLNet_TCP_Accept(…)
5. Läs/skriv data på client_sock
6. Stäng client_sock
7. Ev: Gå tillbka till 4
8. Stäng server_sock
9. …

SMTP protokollet

- { The client connects to the mail server, TCP port 25}
- S: 220 smtp.kth.se ESMTP
- C: HELO smtp.kth.se
- S: 250
- C: MAIL FROM:<anderslm@kth.se>
- S: 250 Ok
- C: RCPT TO:<j.anders.lindstrom@gmail.com>
- S: 250 Ok
- C: DATA
- S: 354 End data with <CR><LF>.<CR><LF>
- ...

SMTP protokollet

- ...
- S: 354 End data with <CR><LF>.<CR><LF>
- C: Subject: Test message
- C: Hello,
- C: Do you get the message?
- C: /Anders
- C: .
- S: 250 Ok: queued as 12345
- C: QUIT
- S: 221 Bye
- {Connection closed by the server}