

- File system - NFS (network file system)
- also remote file system
 - remote mounting
 - standalone file server
 - client caching
 - database workstations

Stateless Servers

- get a request
- Send a response
- next request is a completely new, unrelated request
- Server does not maintain global state

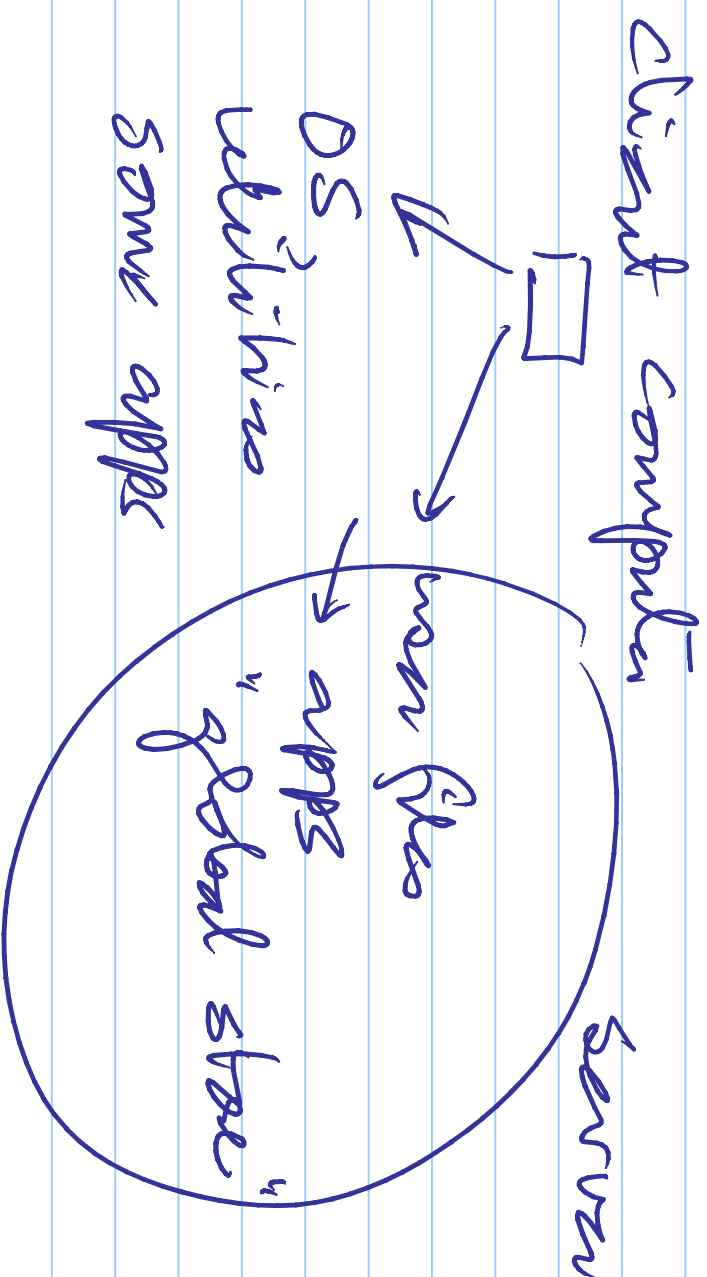
why?

- recovery is simple
- server crashes do not affect clients —
- single server code (no history)

Stalder OK?

- not always possible
- cobina can help.
- HTAL is stalled
- HTTPS " not "

Dalala workstation



Distributed File Systems

- files are stored somewhere
- no fixed mapping from file name to location
- files can be mobile

- non existent plus (searching)
- consistency, failures
- overhead

Primary problem with NFS

- not scalable

1 file at 1 server



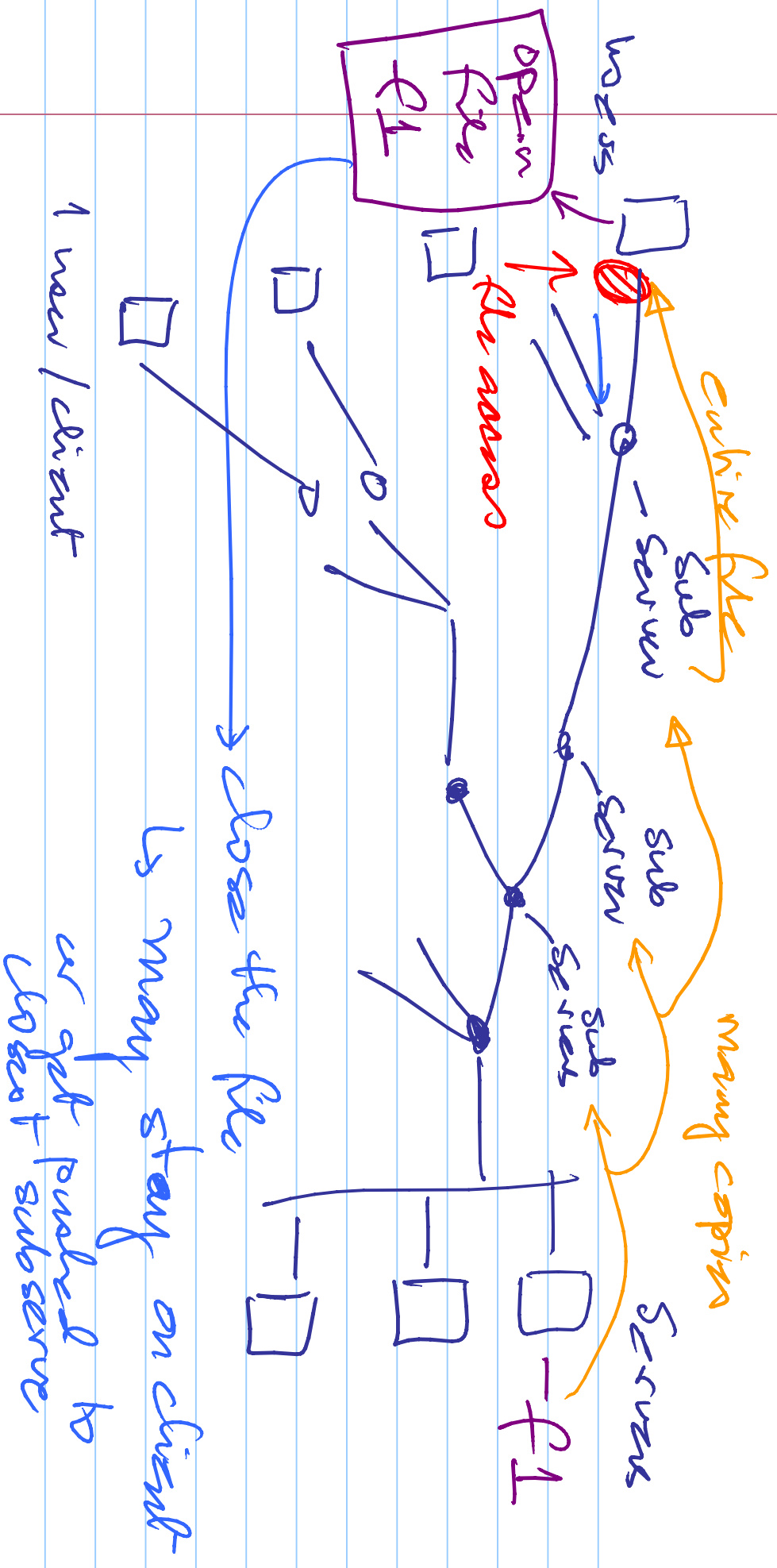
↑
bottleneck

file server that is scalable

1000 + + ... means

Can
Andrew file system → AFS

was
AFS
lots of files
→ lots of users



1 user / client

or get pushed to
closest subserve

next time the f1 is removed

- locate latest copy

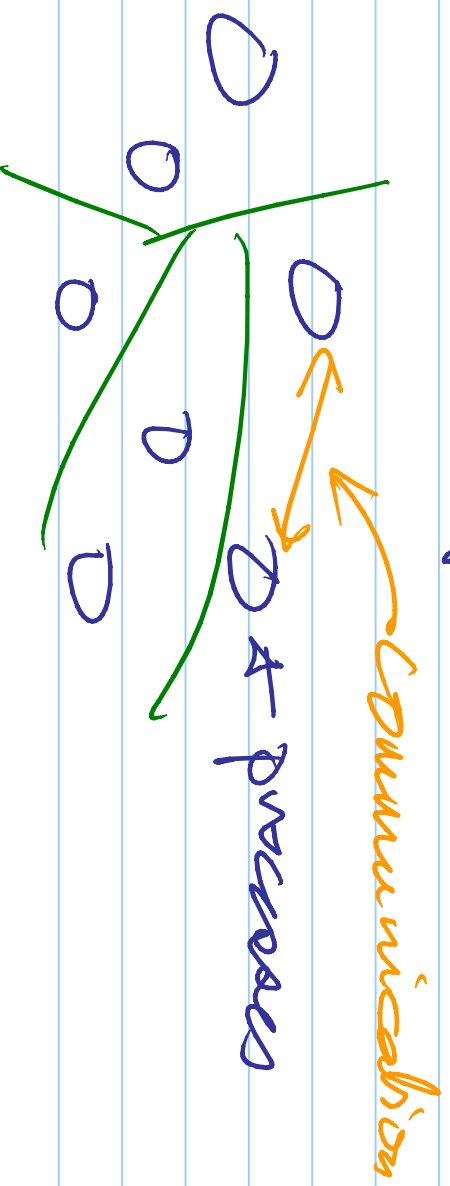
- move it close to client +
on client

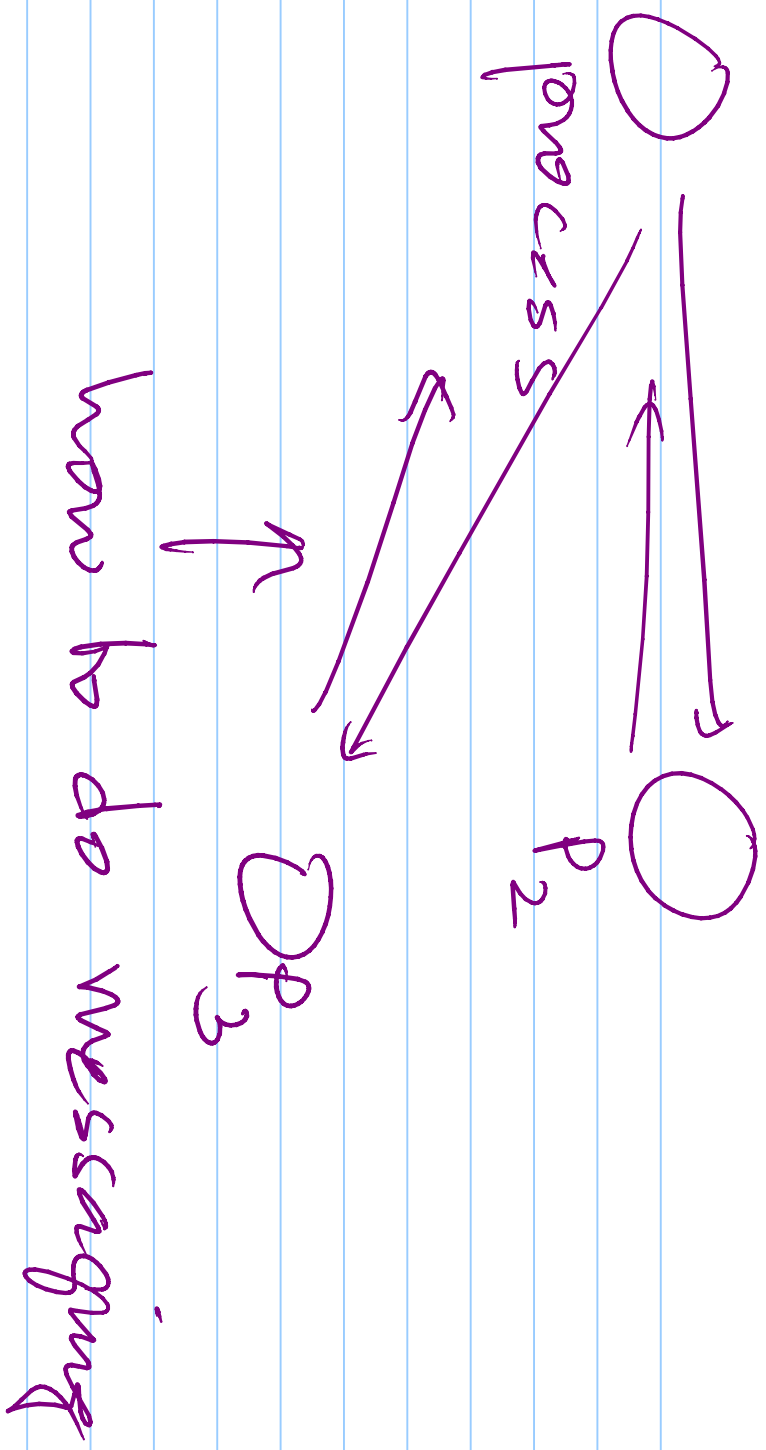
- access via local

-  recall
files on
other routes

"Distributed Operating Systems"

↳ single, unified view of an operating system



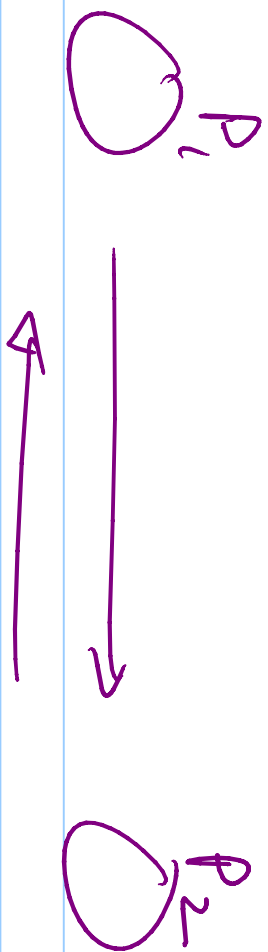


TCP-IP

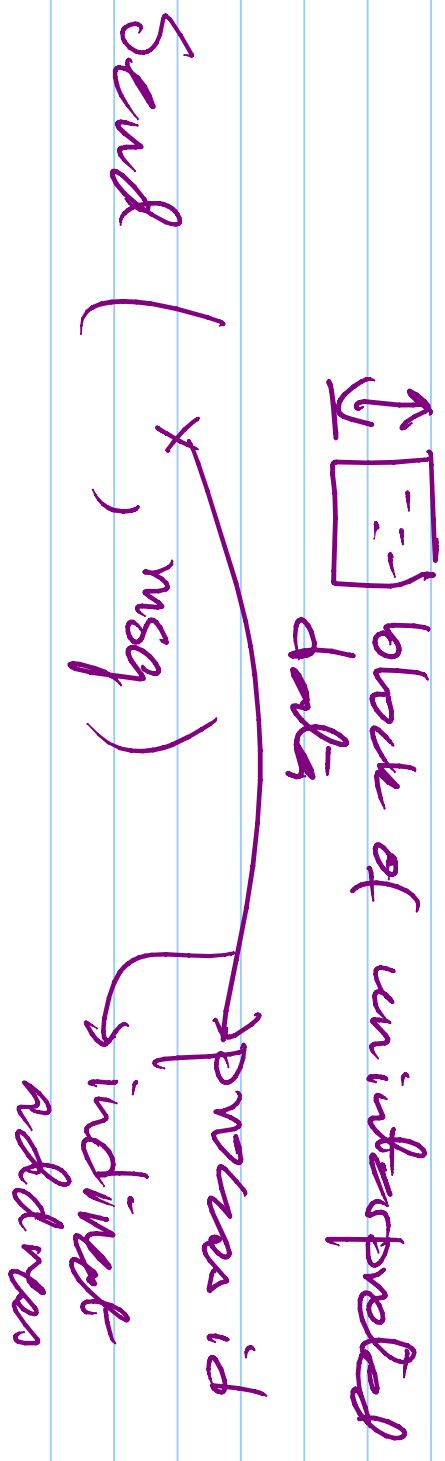
↳ based on domain names

↳ connection oriented

↳ "transport level"

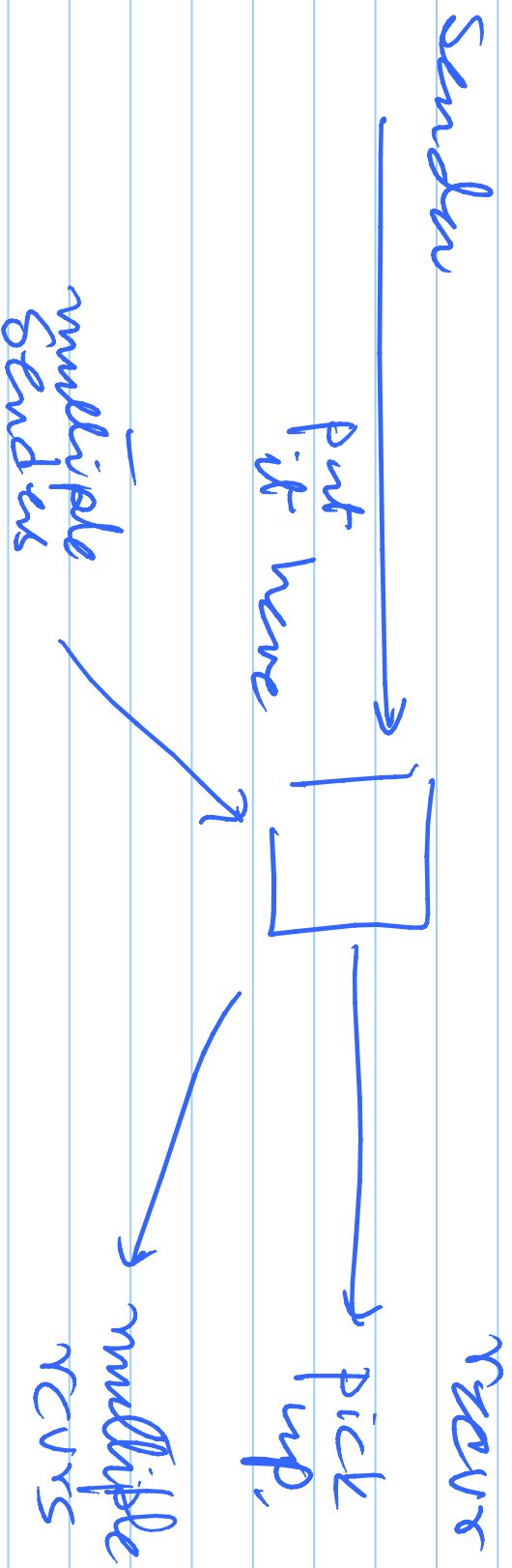


Send a msg $\rightarrow$ how?



indexed address

→ mailbox



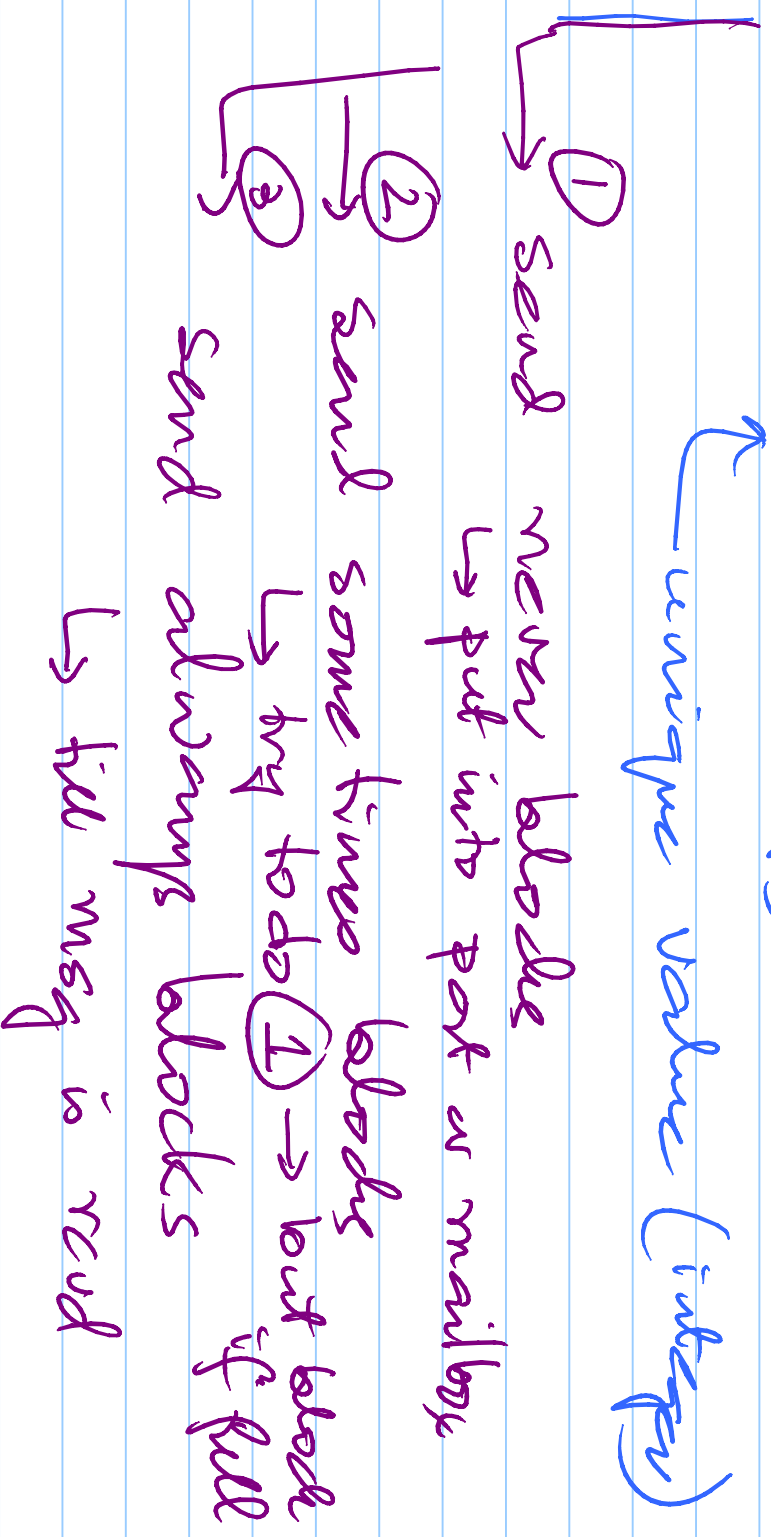
Send (port-id, msg)

↓
must be known to sender
↳ number

TCP-IP

↳ port #, IP address

Send (port #, msg)



① - Never block

→ not good

→ msg loss → dropped

→ return with error

↳ retry?

② - block on full → programmer like this

③ block till recv. + best

↓
Synchronous

messaging

asynchronous messaging

for the receiver →

data gets copied

- recv (port #, msg) here
explicit
recv to
get
a msg

↳ non blocking

↳ returns if no

msg (with error code)

↳ blocking recv → block till

msg arrives

↳ " " + timeout

rcv (port #, msg)

↳ blocks till msg recd.

↳ if send is synchronous
wake up sender

