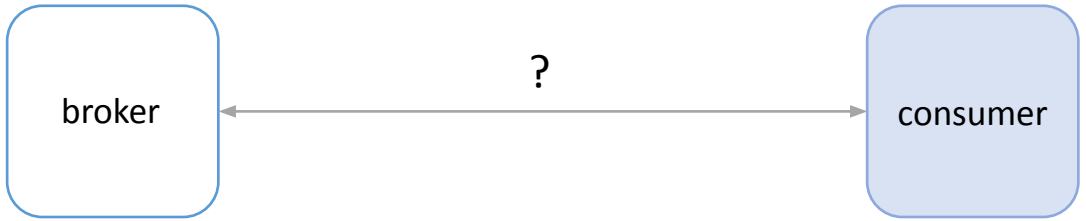
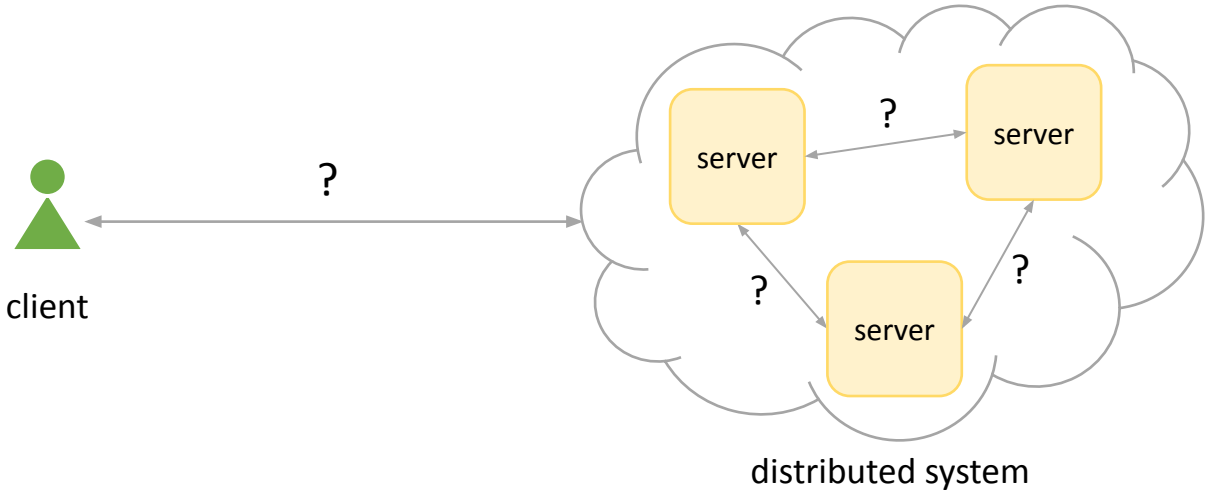
How to choose a network protocol

UDP TCP HTTP

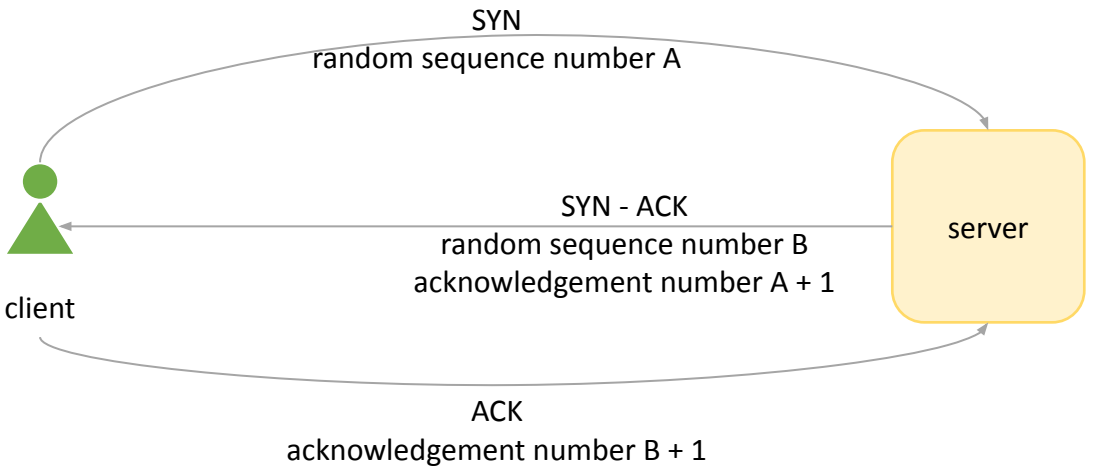




How to choose a network protocol

TCP

reliable, ordered, connection-oriented



- client may establish multiple TCP connections to the server
- TCP connection consumes resources (mainly memory)
- modern servers can handle hundreds of thousands of simultaneous TCP connections
- handling many connections ≠ handling many requests

How to choose a network protocol

no connection, no handshaking

UDP

very fast

not reliable

when to use

data must be transferred as soon as possible and occasional loss of data is tolerable

the most recent data, although may not be complete, has priority over older but more complete data

connection-oriented

TCP

fast

reliable

when to use

speed matters and we need to avoid losing data

HTTP/1.0: connection per request

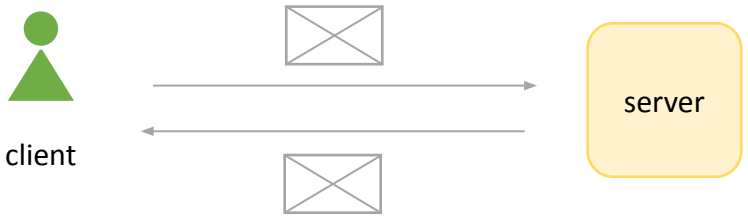
HTTP/1.1: persistent connections

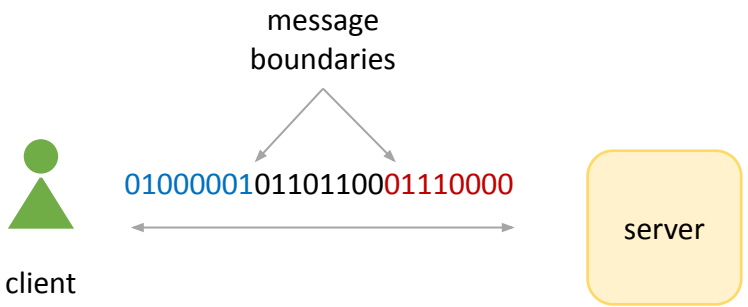
HTTP

How to choose a network protocol

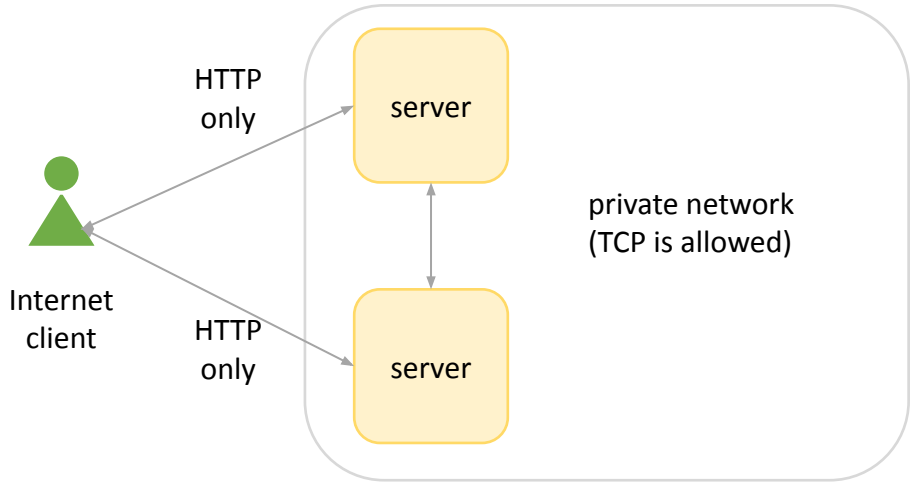
Why TCP is not enough when we need
fast and reliable communication?

reason 1
messages instead of byte streams





reason 2
security



How to choose a network protocol

no connection, no handshaking

UDP

very fast

not reliable

when to use

data must be transferred as soon as possible and occasional loss of data is tolerable

the most recent data, although may not be complete, has priority over older but more complete data

connection-oriented

TCP

fast

reliable

when to use

speed matters and we need to avoid losing data

HTTP/1.0: connection per request

HTTP/1.1: persistent connections

HTTP

widely adopted

developer-friendly

a bit slower

reliable

when to use

systems need to provide a public API