Linux Network Programming

Setting the Scene

Chris Brown

In This Module …

The client/server model:
Which is the client?
Which is the server?

Connection-oriented
vs
Connectionless services

Language choice:
C
Python

Tools:
Development environment
"What's going on?" tools

Assumed Knowledge

C language

Syntax, data types, structures, pointers

Python

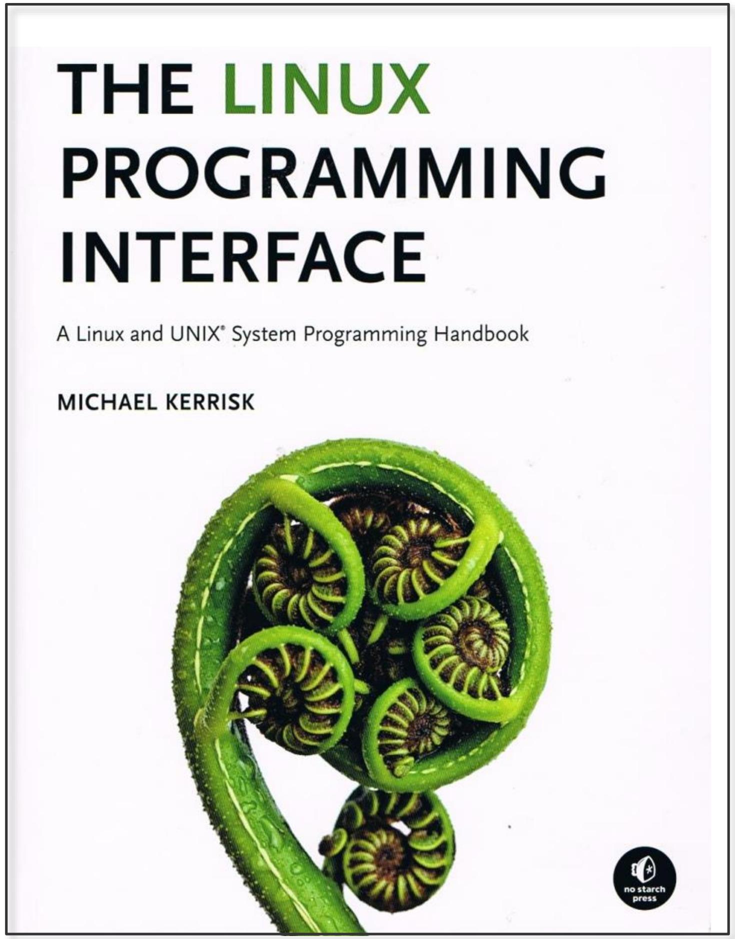
Not essential (but see "Python Fundamentals" for a good intro)

Linux

Knowledge of systems programming

This course is intended as a follow-on to the "Linux Systems Programming" course

Book Recommendation



1500 pages!

Authoritative

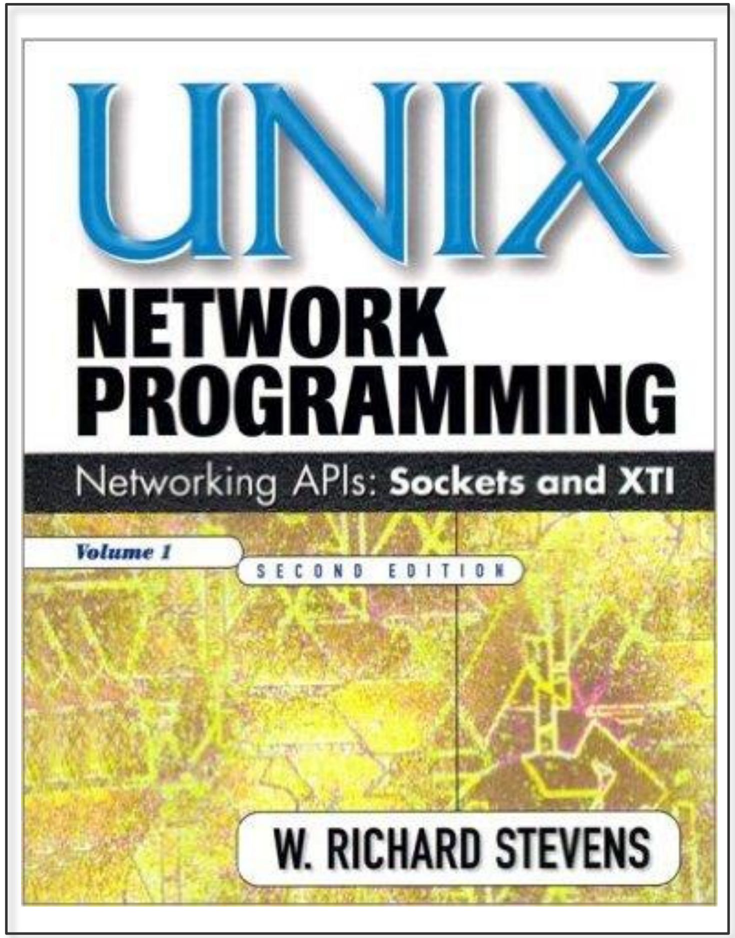
Well written

Lots of examples

Excellent!

Chapters 56 - 61

Another Book Recommendation



Older (UNIX) focus
(but 99% relevant to Linux)

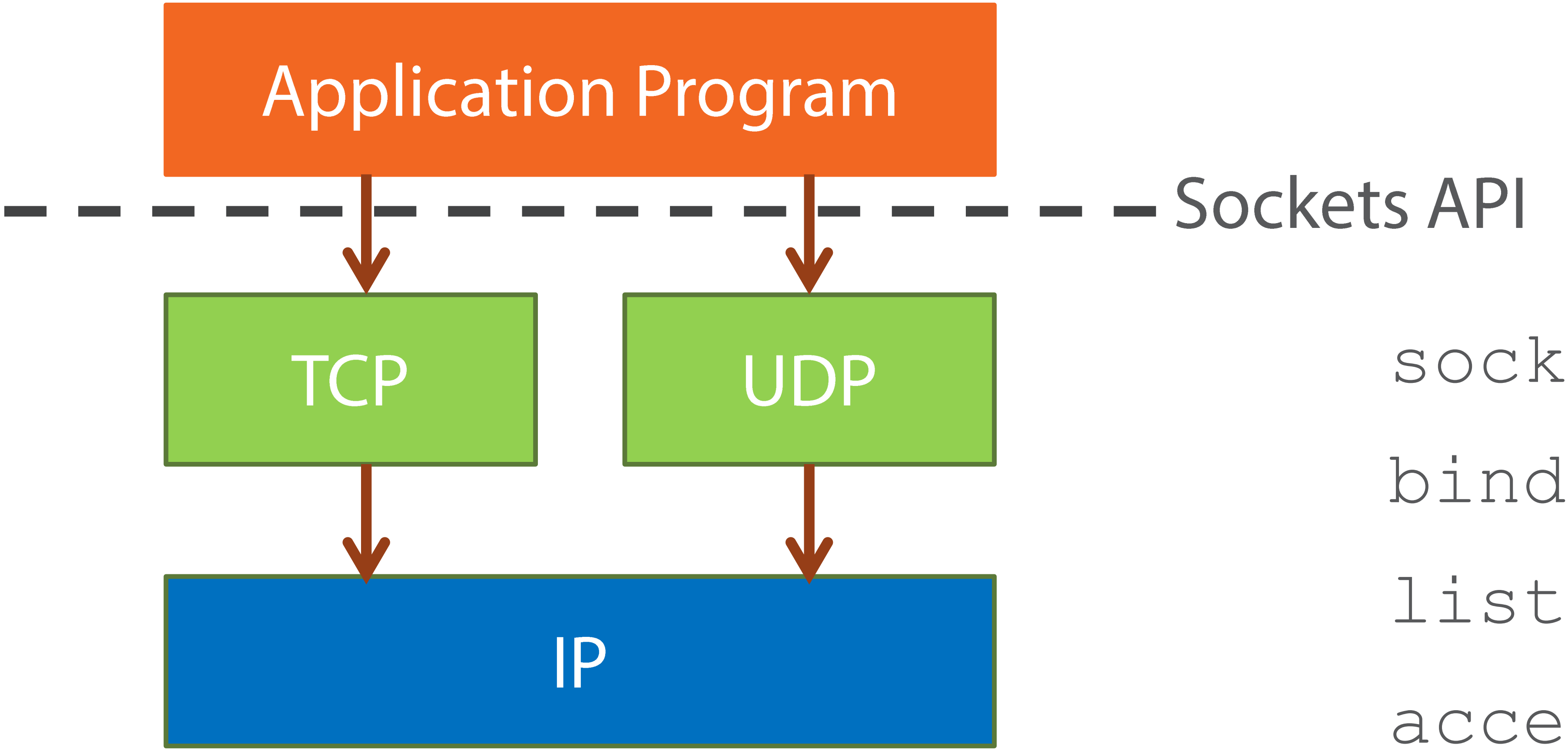
Authoritative

Well written

Lots of examples

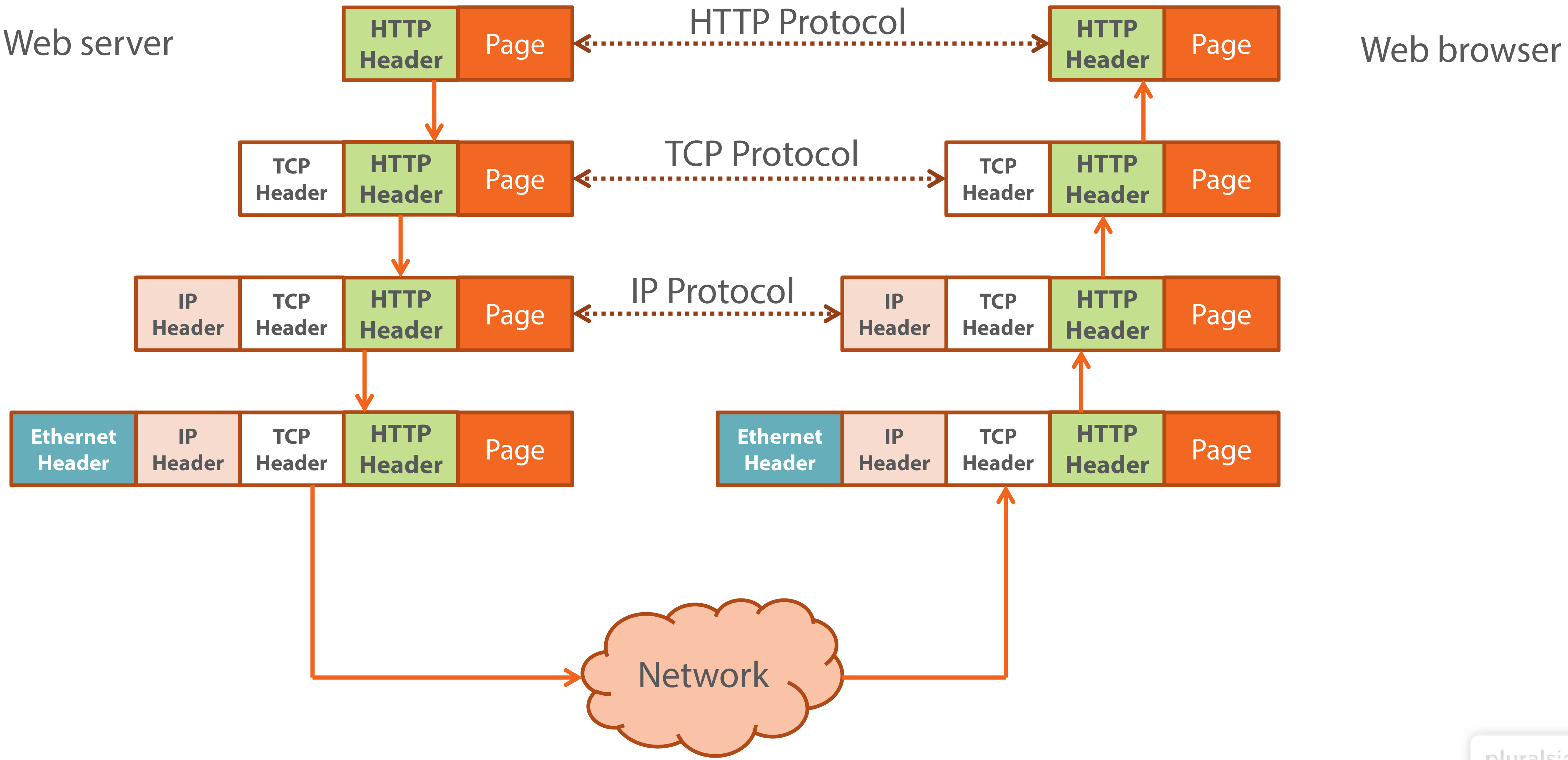
Excellent!

Layers



```
socket()
bind()
listen()
accept()
connect()
```

Encapsulation



The Client/server Relationship

Server offers a service

- Establishes a "communication endpoint"
- Passively waits for business

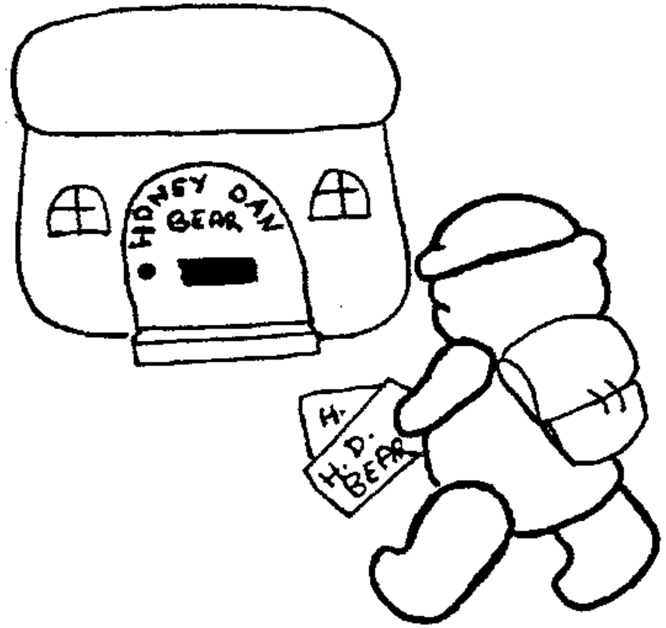
Client "consumes" a service

- Actively connects to the server

Typically:

- Client is local
- Server may be remote

Connectionless Communication



Each message is individually addressed

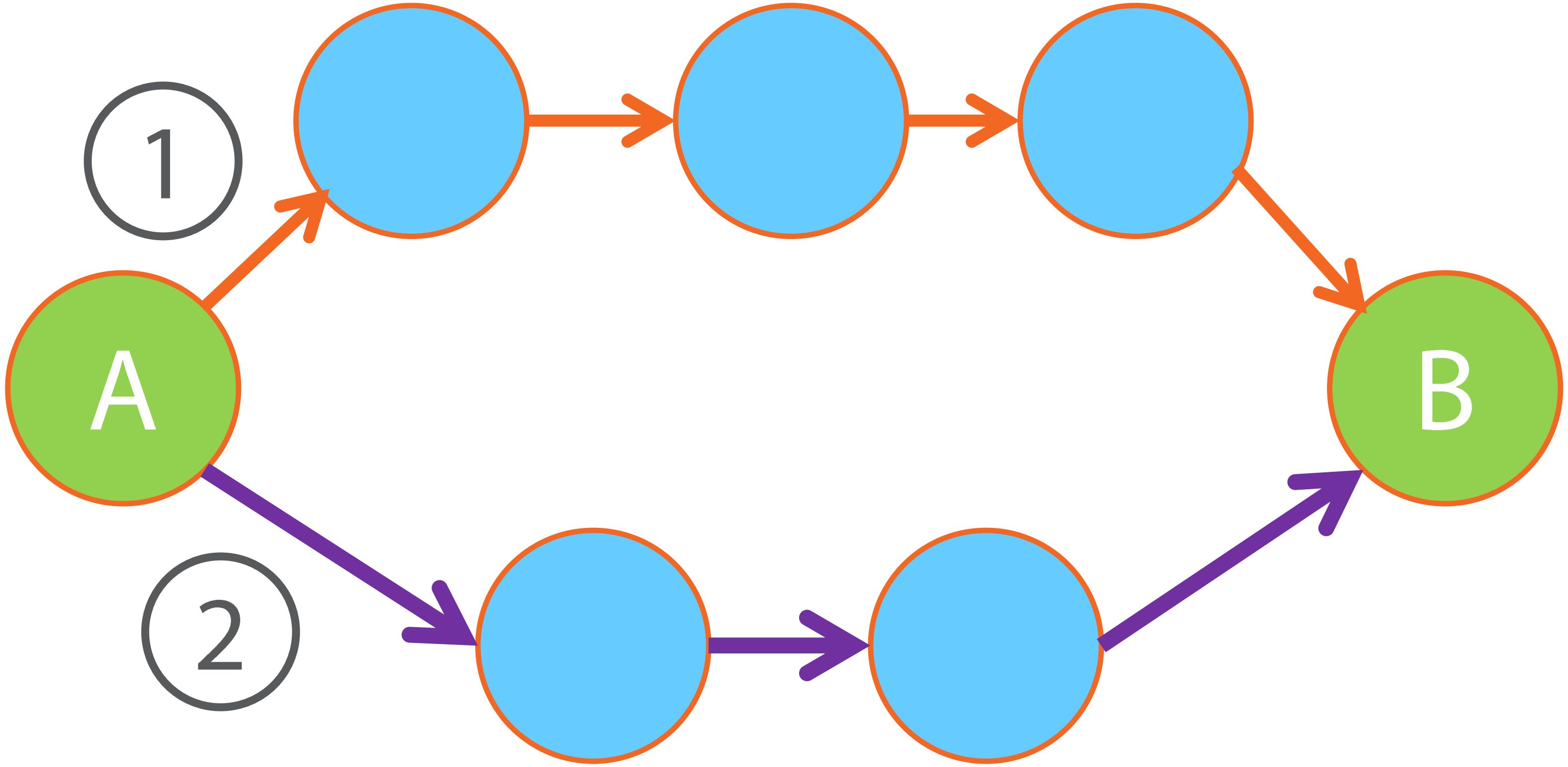
Connectionless Communication

No guarantee of delivery

Connectionless Communication

Messages may not arrive in the order they were sent

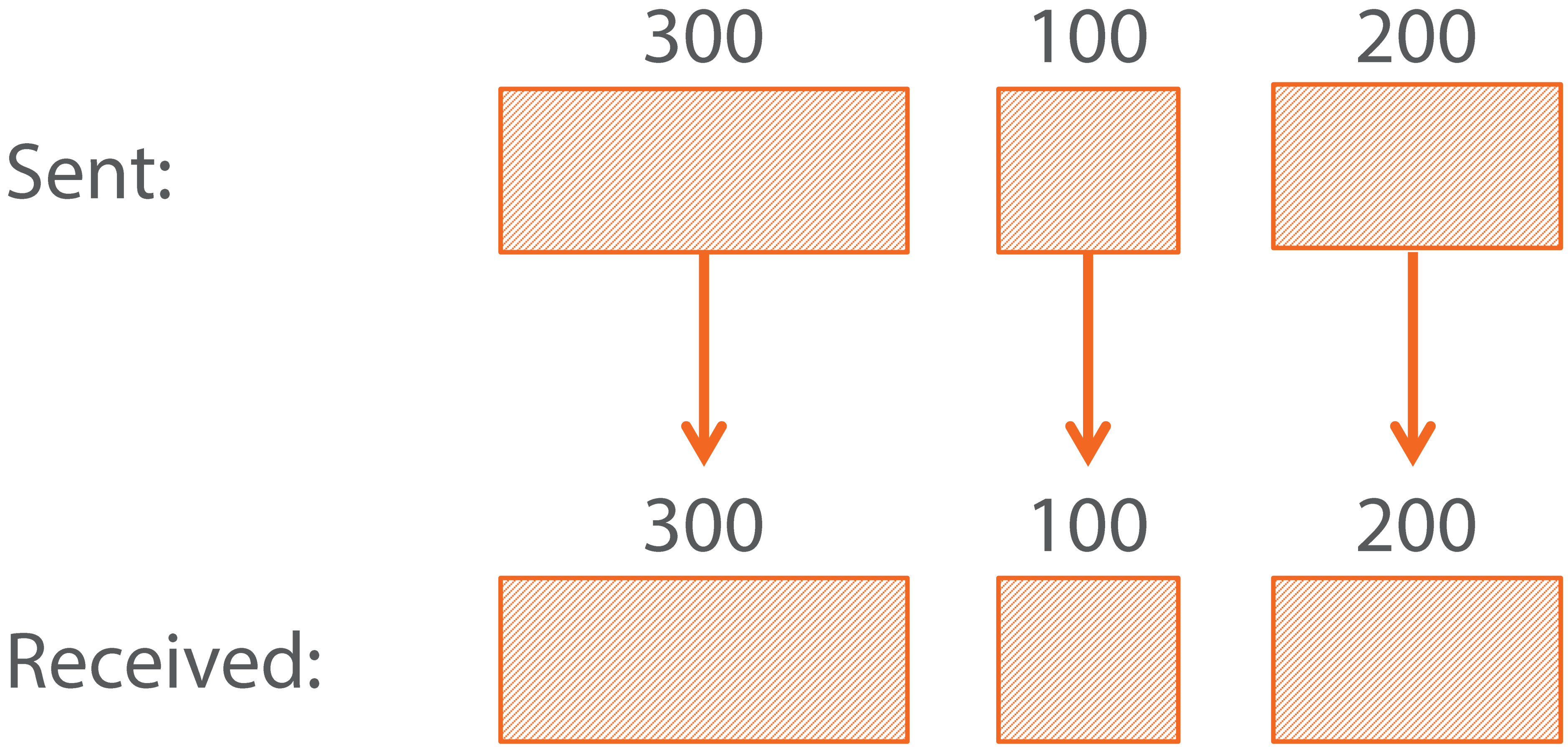
Mis-ordered Delivery



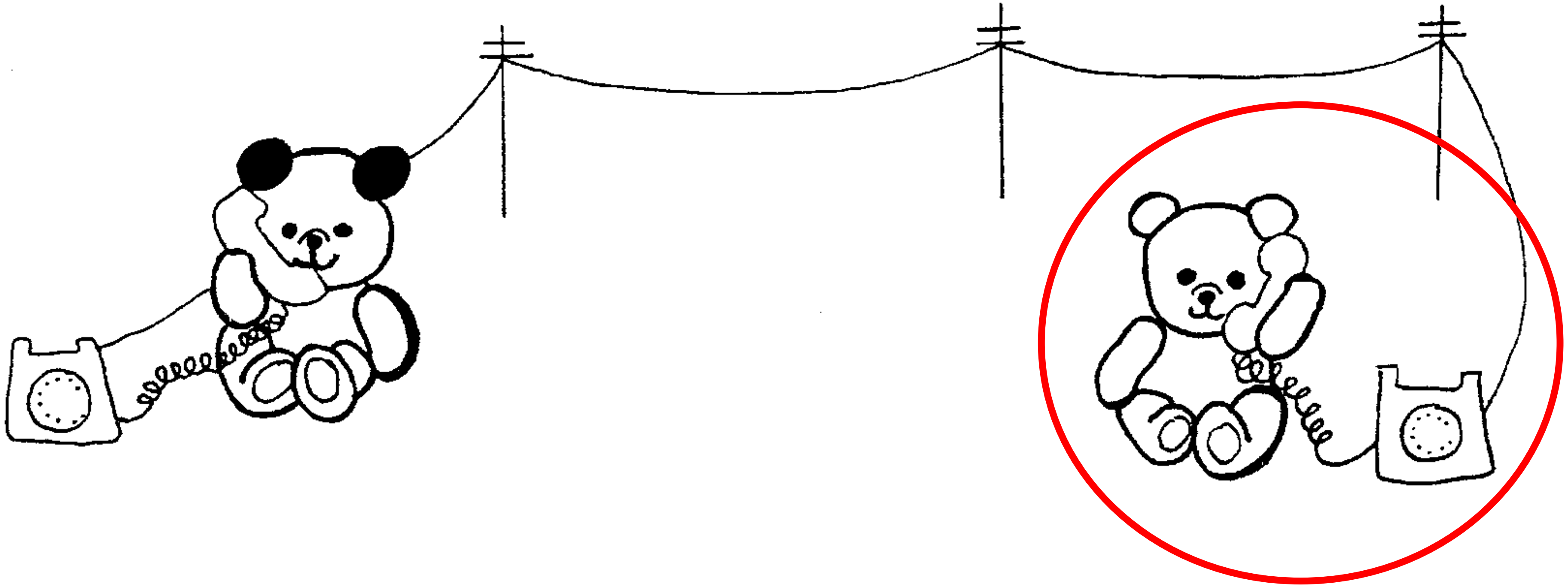
Connectionless Communication

Message boundaries are preserved

Message Boundaries Are Preserved

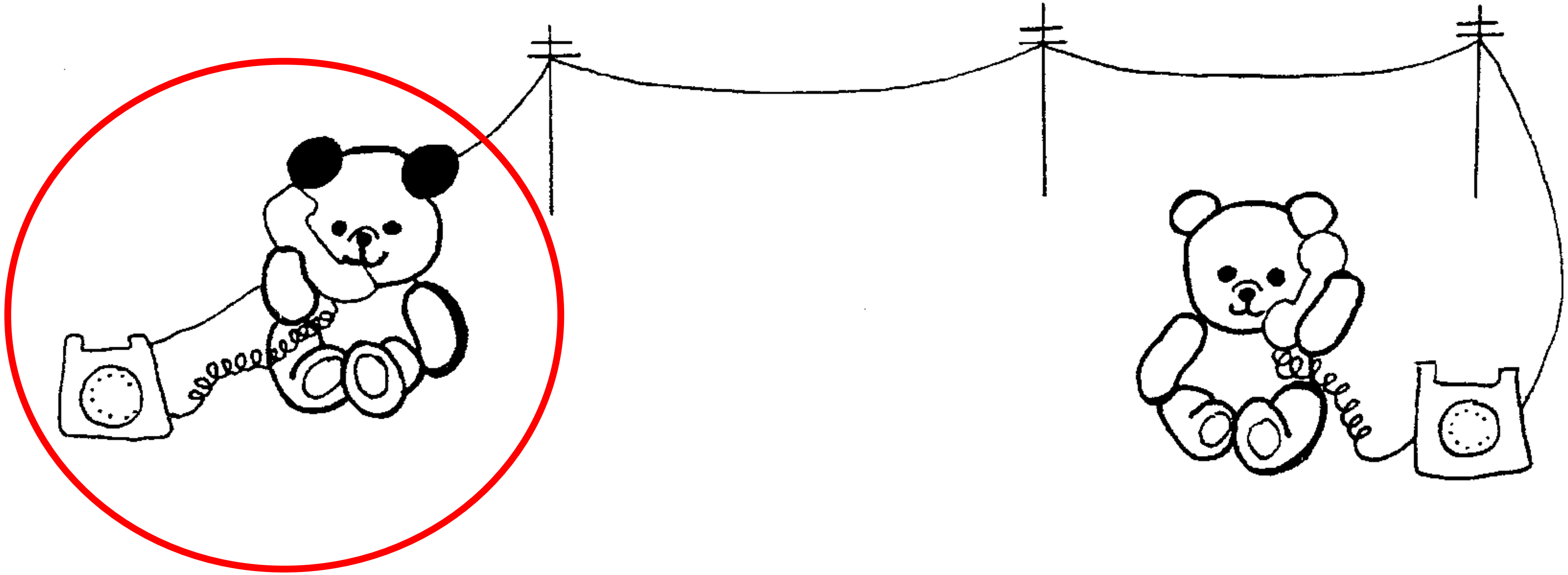


Connection-oriented Communication



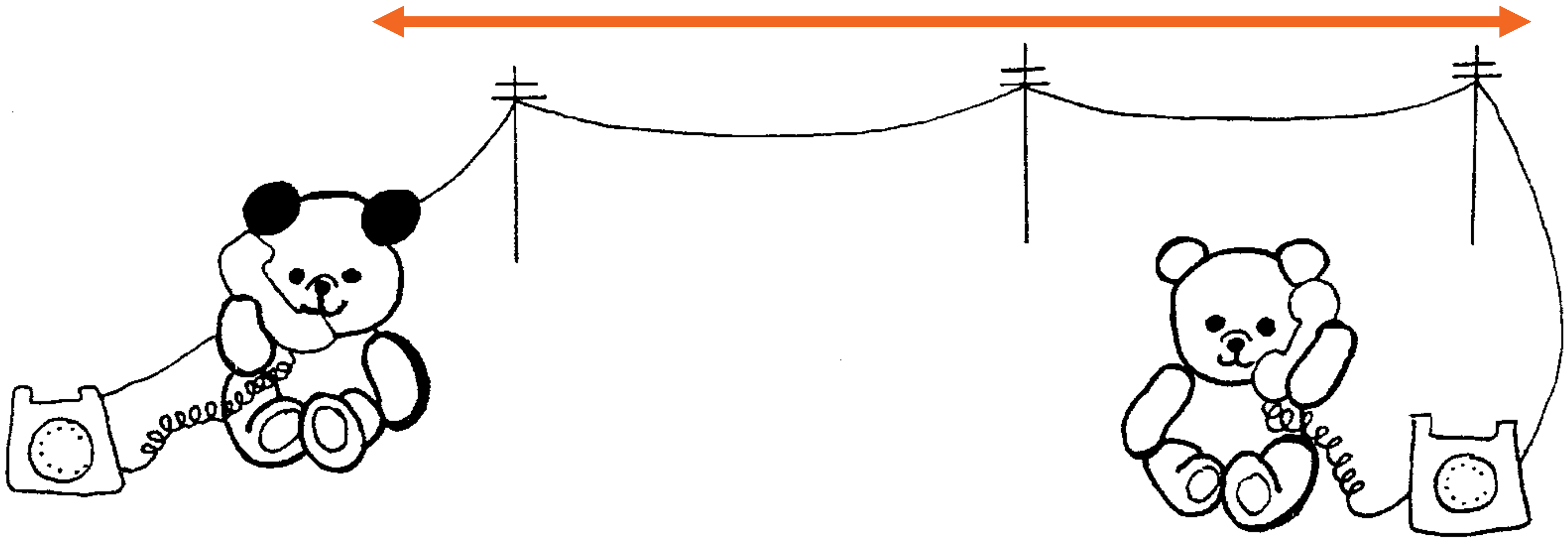
Recipient address supplied "up front" when connection established

Connection-oriented Communication



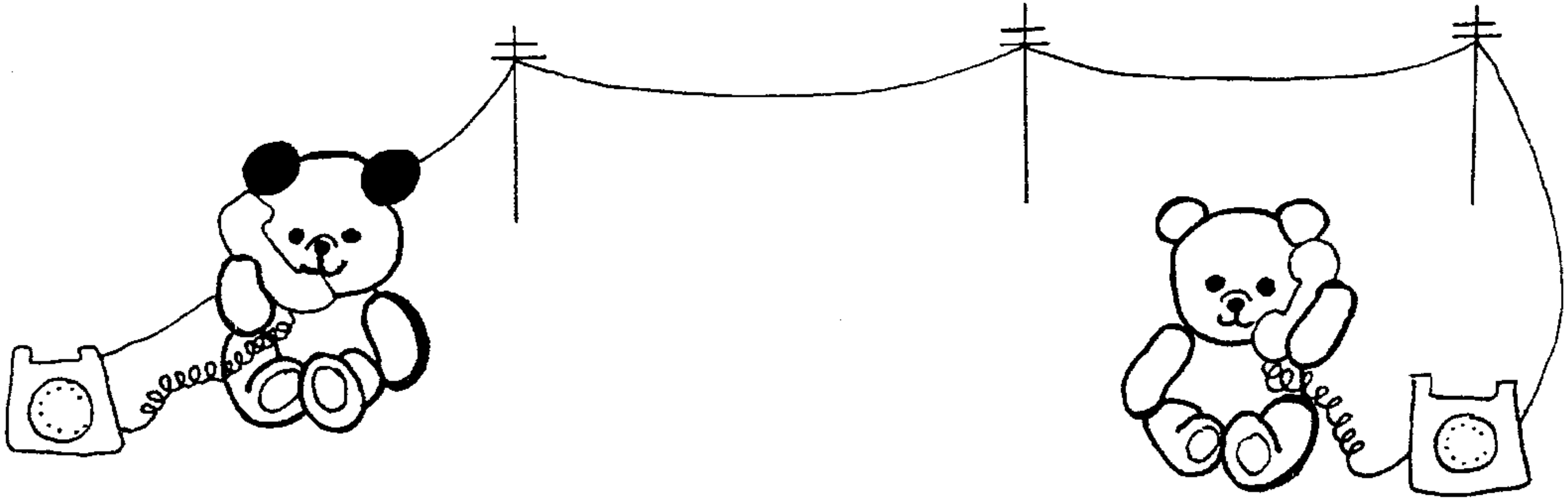
Recipient must answer the call ("accept the connection")

Connection-oriented Communication



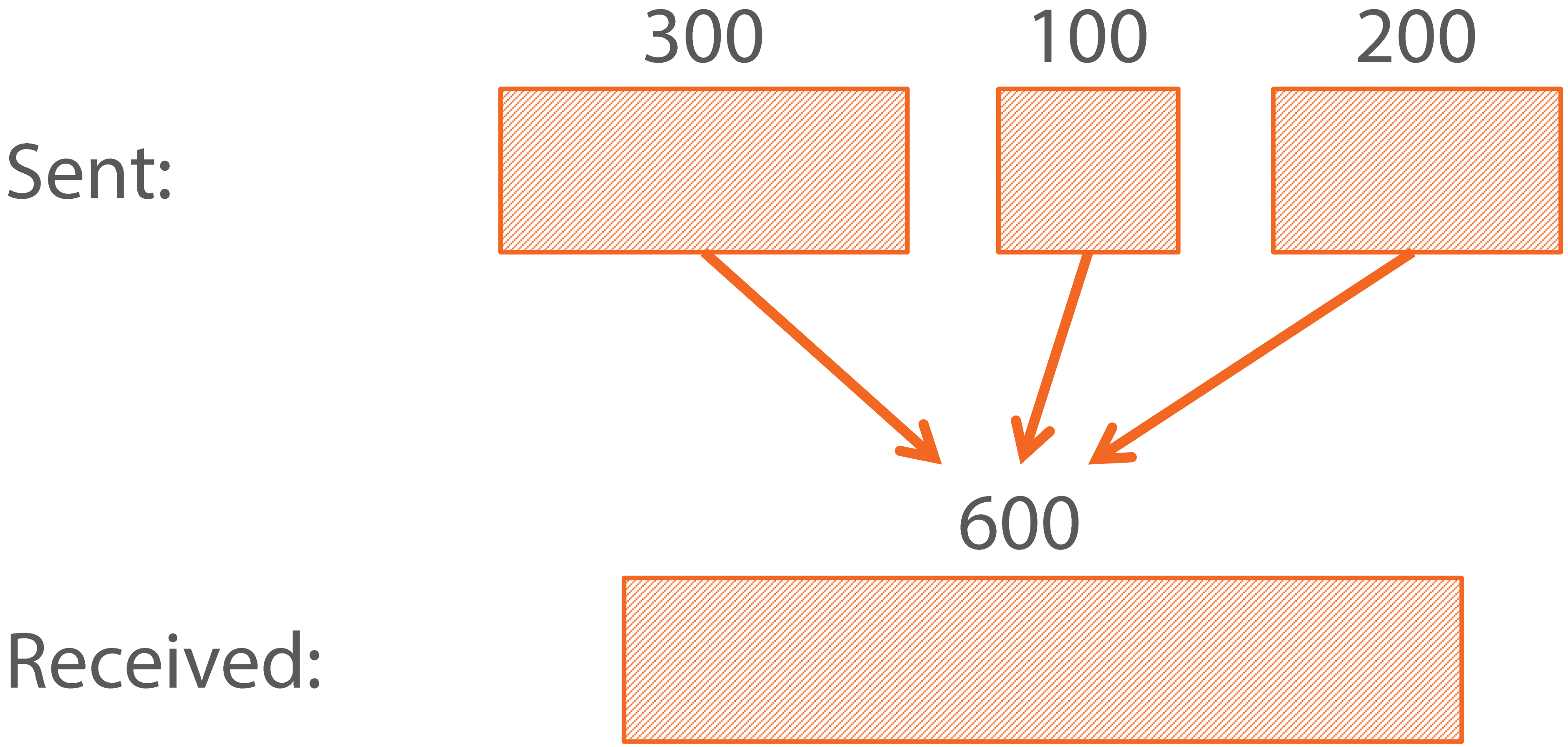
Illusion of a copper wire connecting the two ends

Connection-oriented Communication



Message boundaries are not preserved

Message Boundaries Are Not Preserved



Sent:

300

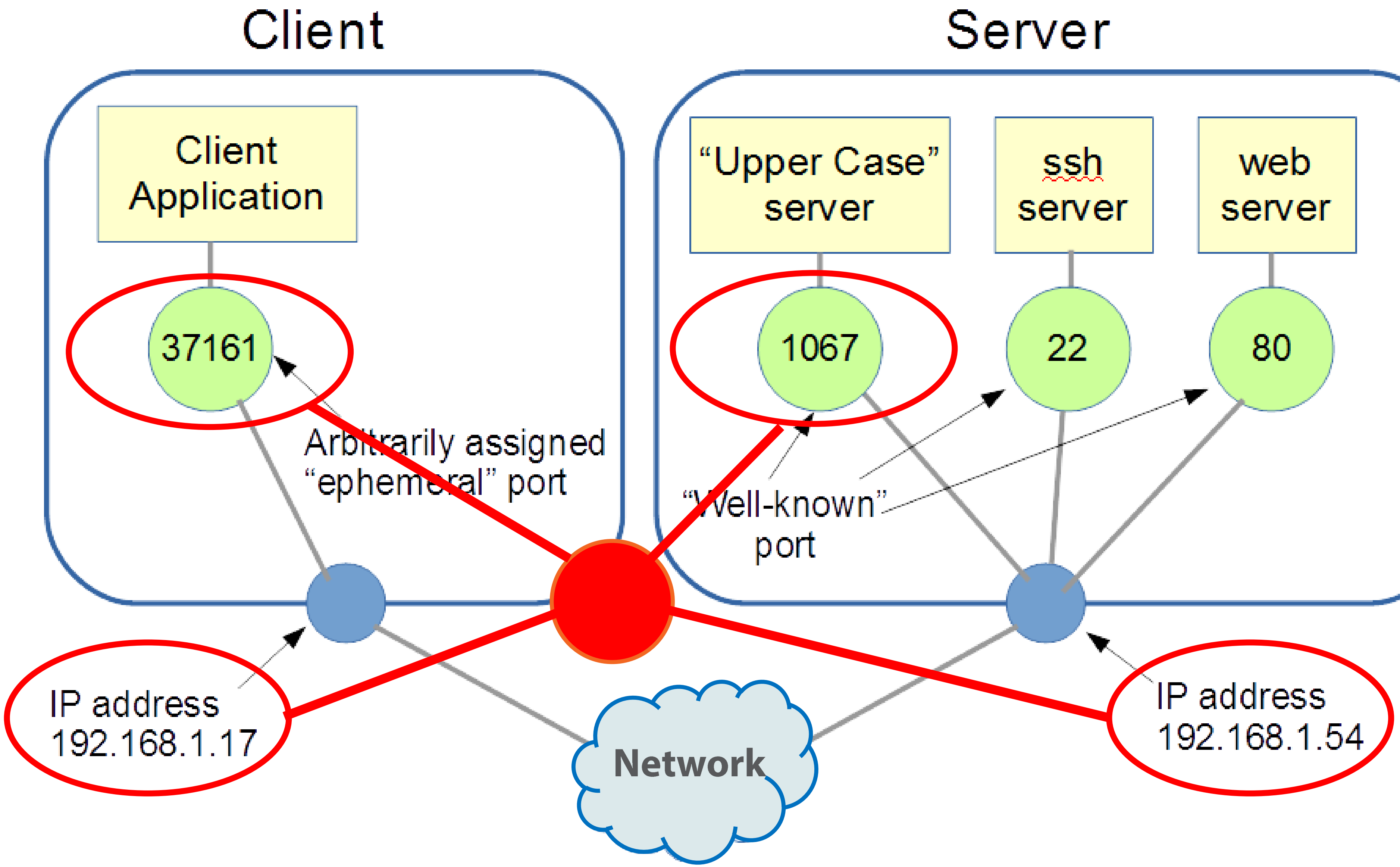
100

200

600

Received:

Client/server Association



C and Python

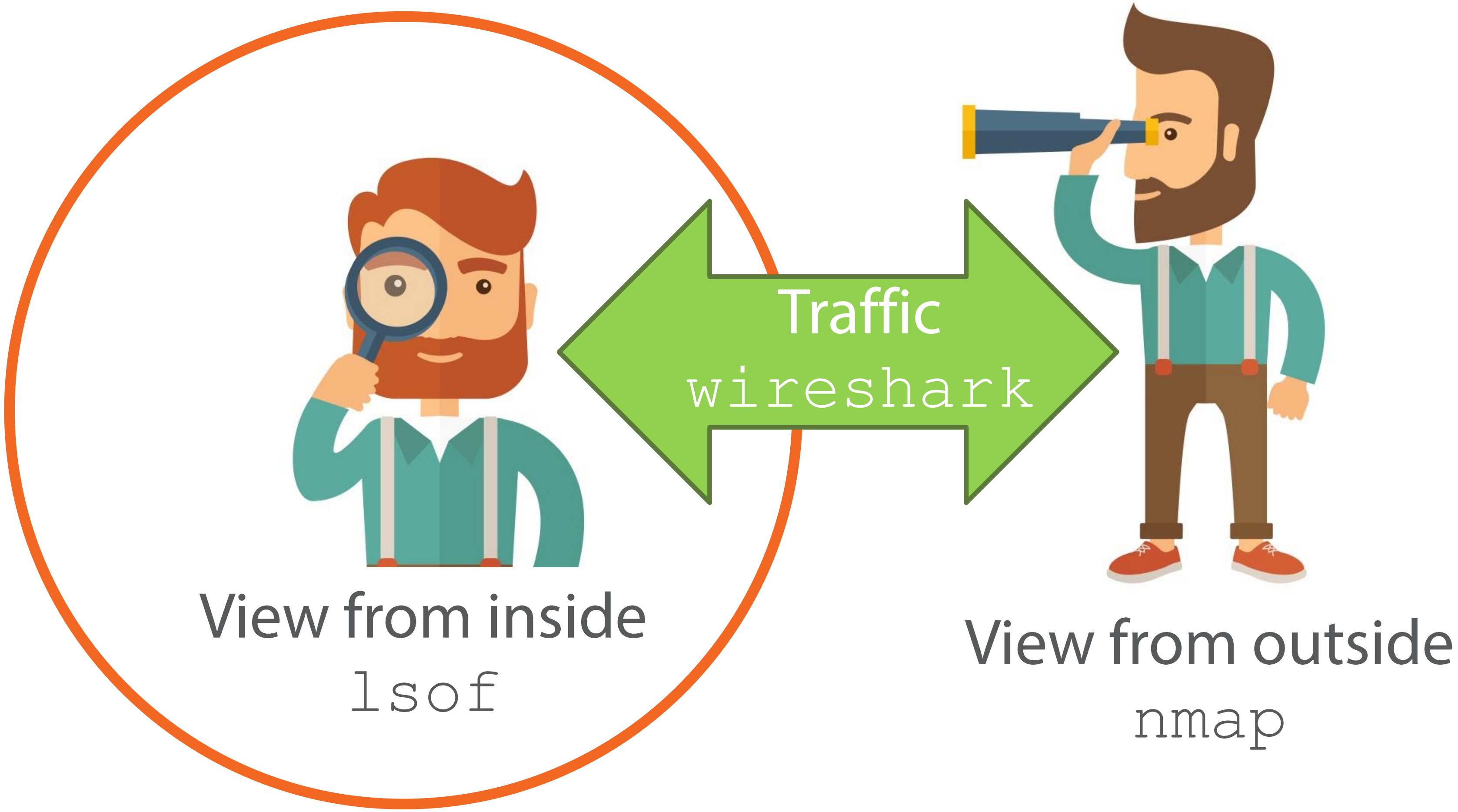
THE C PROGRAMMING LANGUAGE

- Low-level systems programming
- Dennis Ritchie, 1972
- Statically typed
- Procedural
- Fully compiled

python

- High-level, multi-purpose
- Guido van Rossum, 1991
- Dynamically typed
- Multi-paradigm including O-O
- Interpreted

Demonstration



Moving Forward…

In this module:

- Clients and servers
- Connectionless service
- Connection-oriented service
- Tools: lsof, nmap, wireshark

Coming up in the next module:

- Writing TCP-based servers