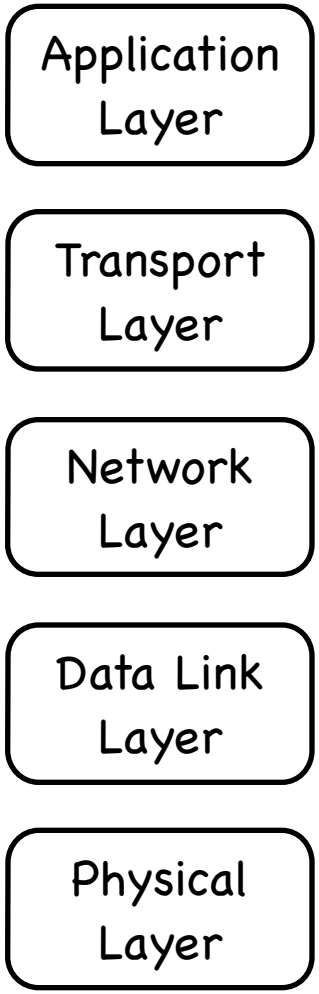
Introduction to Programming for Scientists

Lecture 6
HTML, XML and the Web
Web Scraping and How to Avoid It

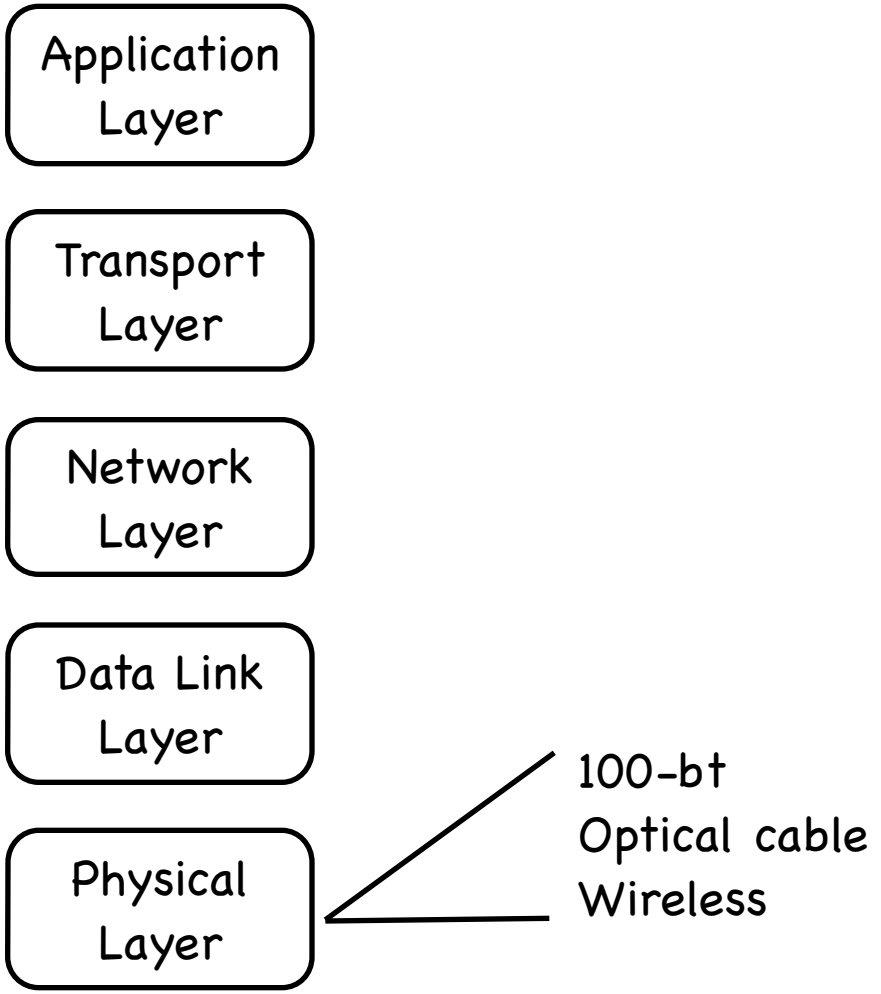
Prof. Steven Ludtke
N410.07, sludtke@bcm.edu

Quick Homework Review

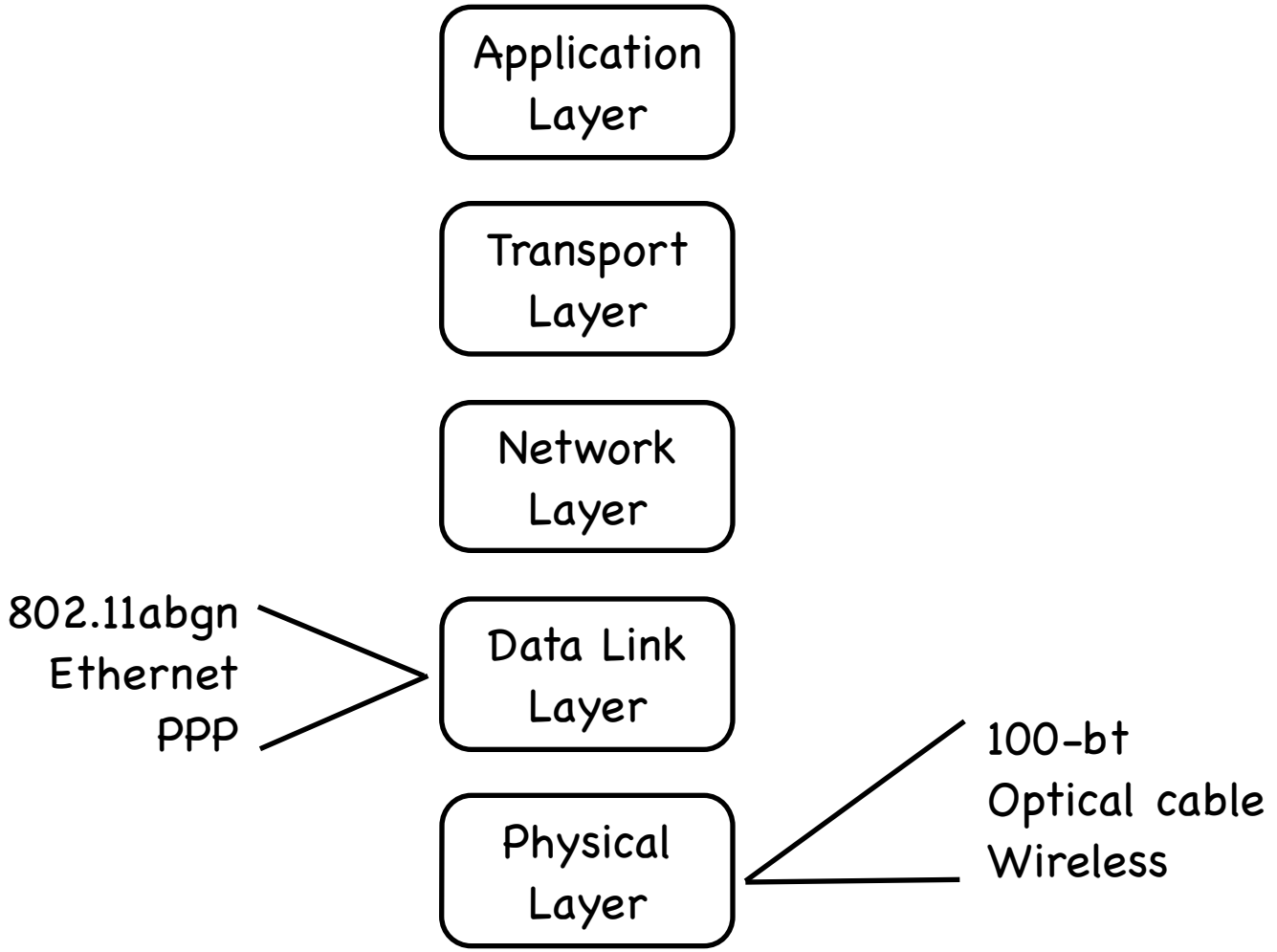
Networking


Application Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

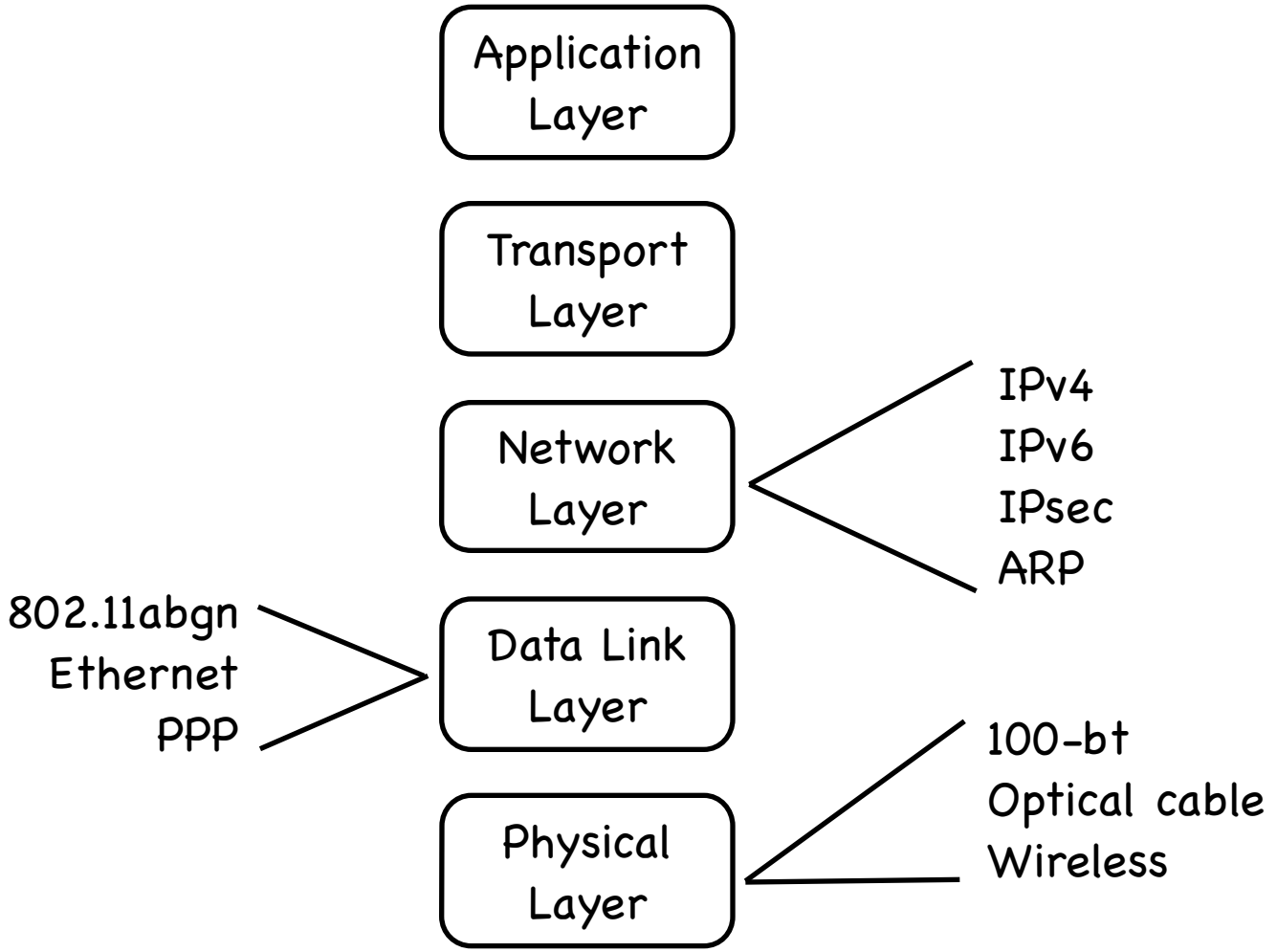
Networking


Application Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer
100-bt
Optical cable
Wireless

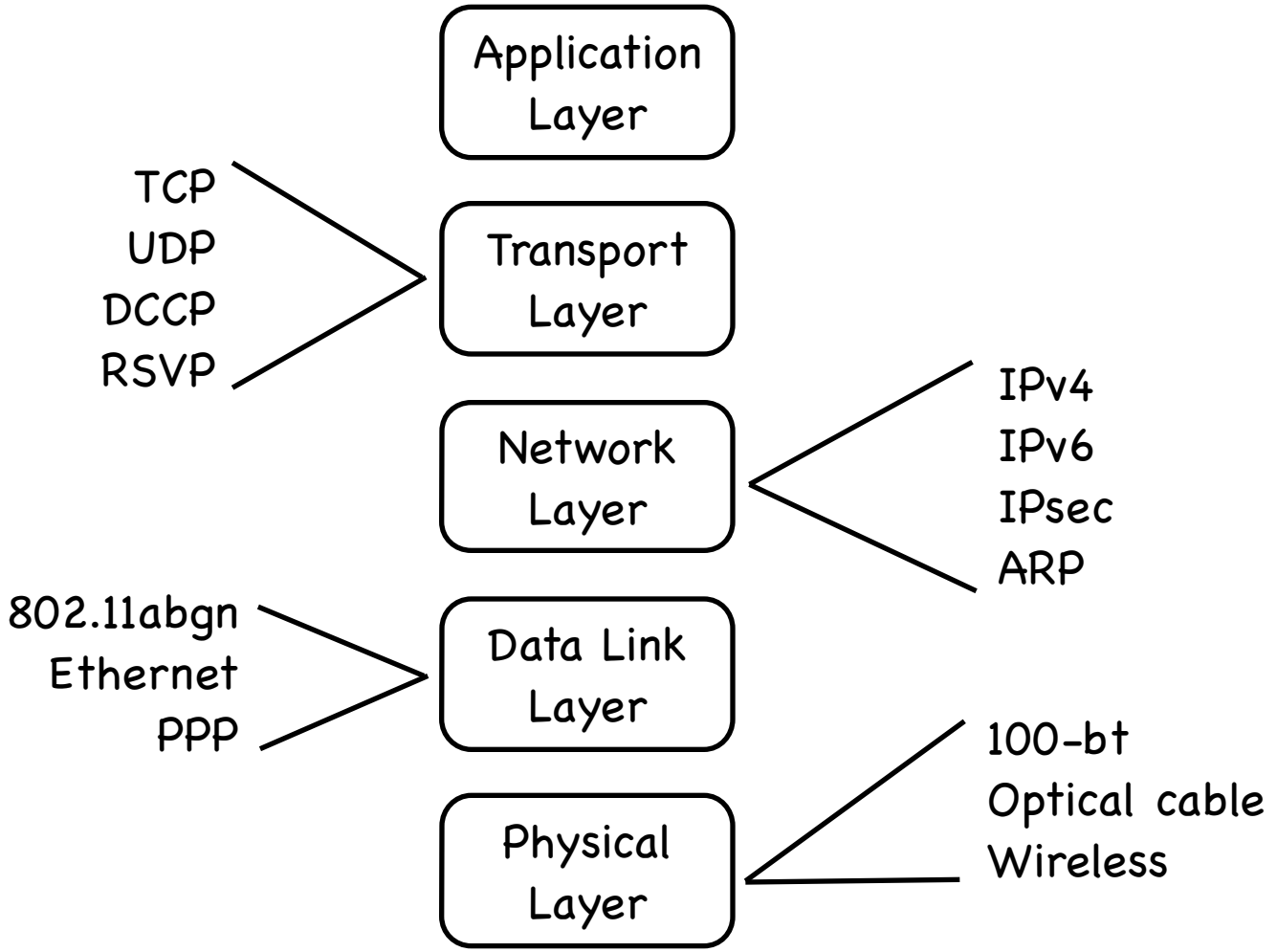
Networking


Application Layer
Transport Layer
Network Layer
802.11abgn
Ethernet
PPP
Data Link Layer
100-bt
Optical cable
Wireless
Physical Layer

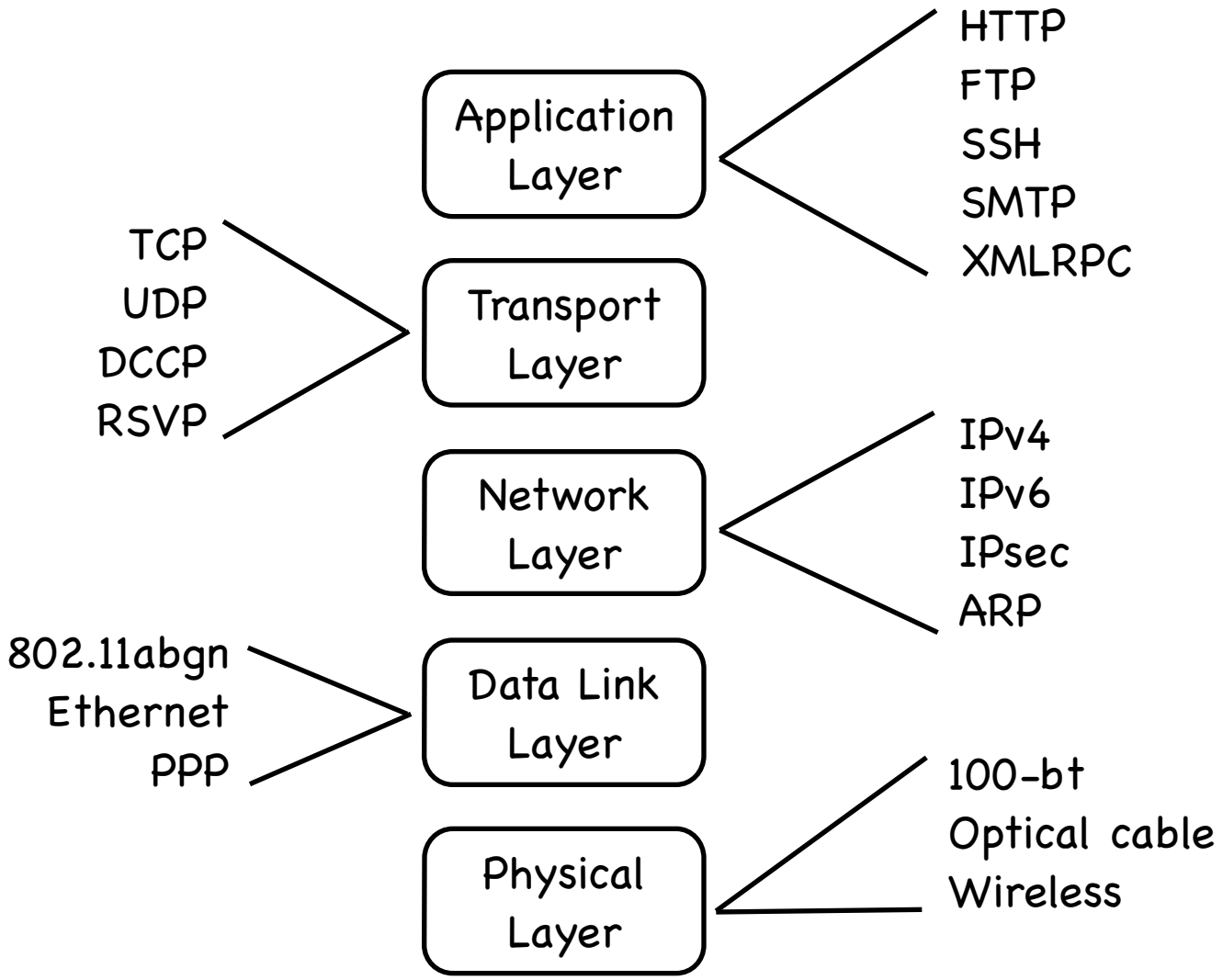
Networking


Application Layer
Transport Layer
Network Layer
IPv4
IPv6
IPsec
ARP
802.11abgn
Ethernet
PPP
Data Link Layer
100-bt
Optical cable
Wireless
Physical Layer

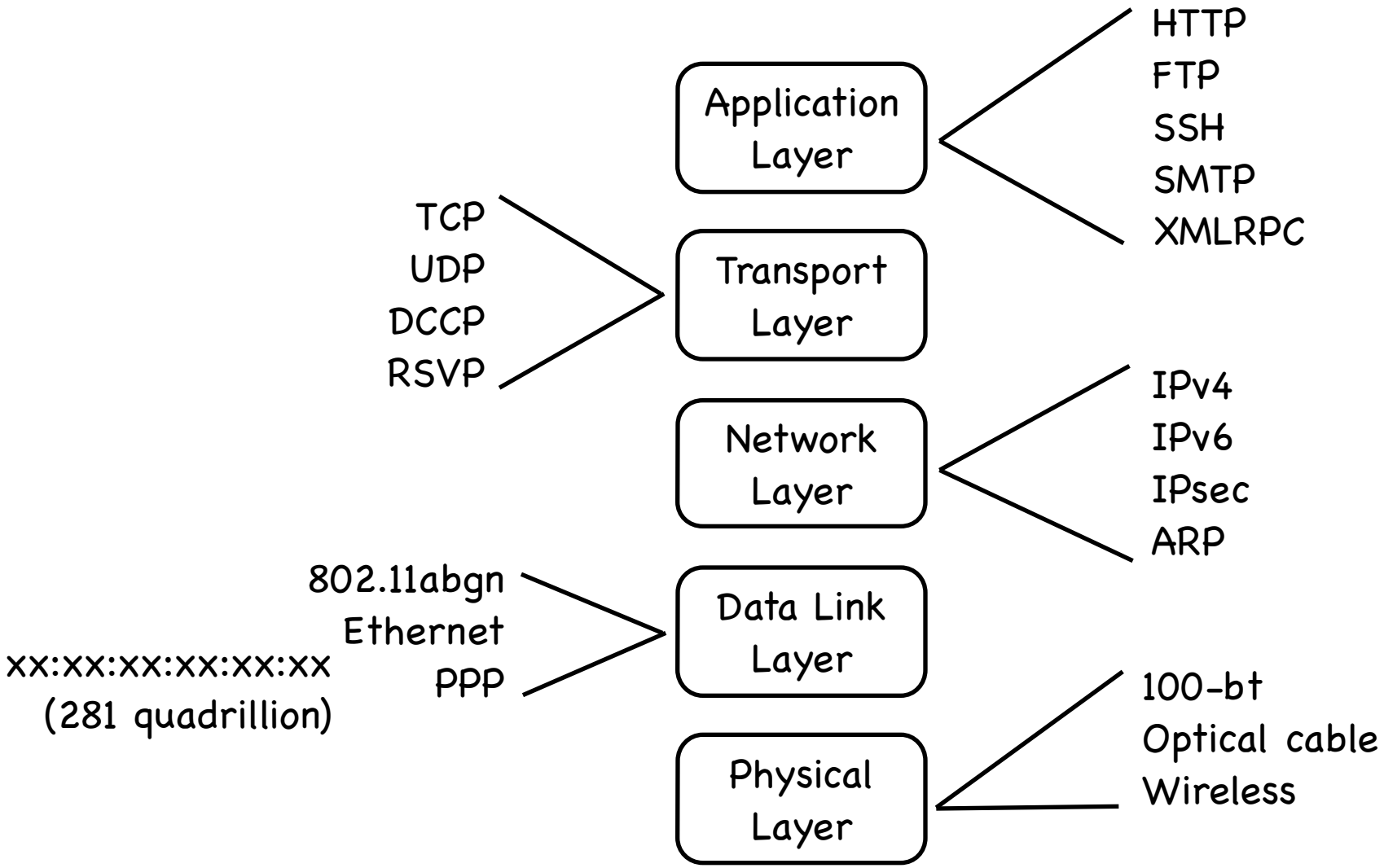
Networking


Application Layer
TCP
UDP
DCCP
RSVP
Transport Layer
Network Layer
IPv4
IPv6
IPsec
ARP
802.11abgn
Ethernet
PPP
Data Link Layer
Physical Layer
100-bt
Optical cable
Wireless

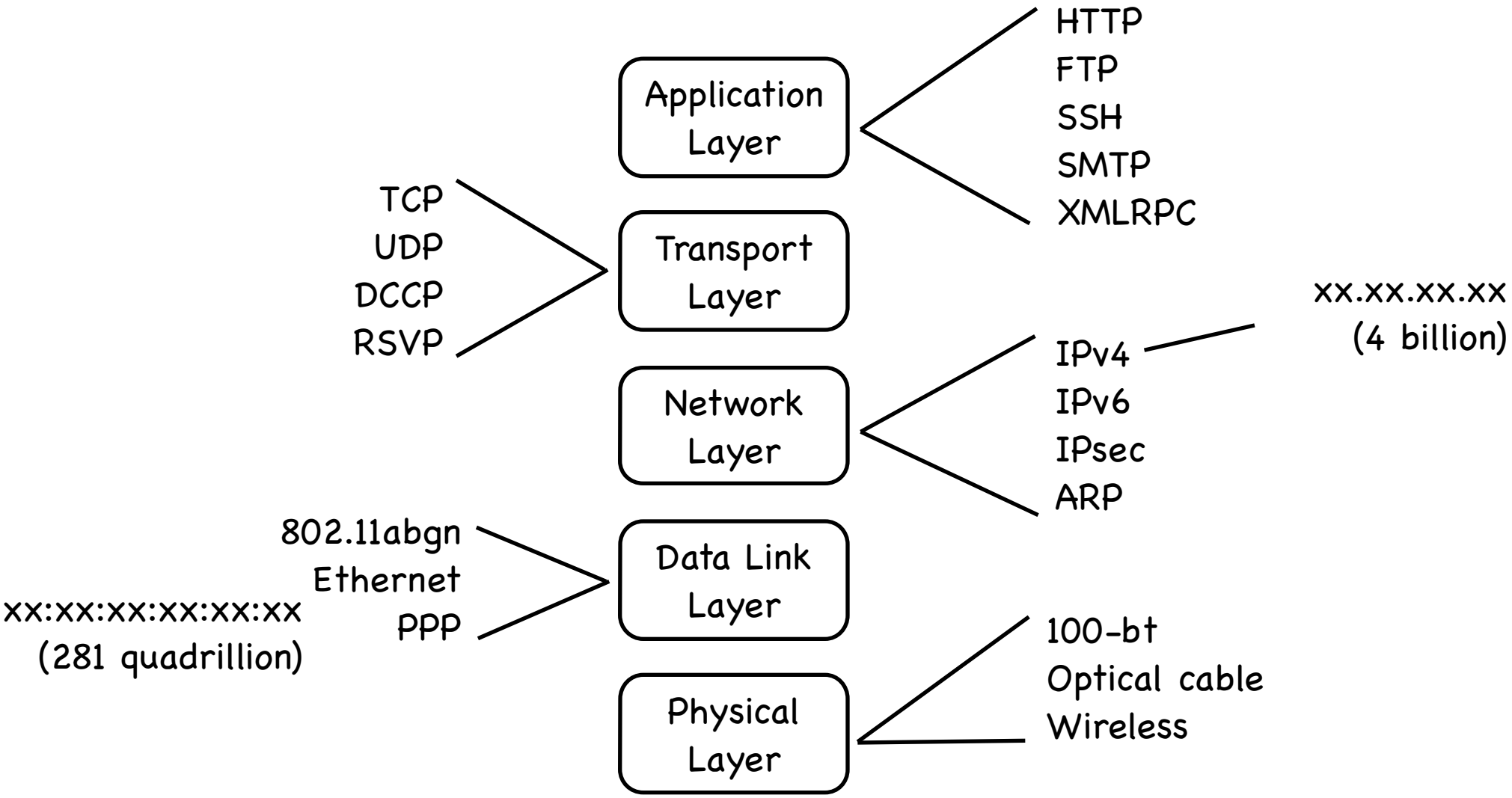
Networking


Application Layer
HTTP
FTP
SSH
SMTP
XMLRPC
Transport Layer
TCP
UDP
DCCP
RSVP
Network Layer
IPv4
IPv6
IPsec
ARP
Data Link Layer
802.11abgn
Ethernet
PPP
Physical Layer
100-bt
Optical cable
Wireless

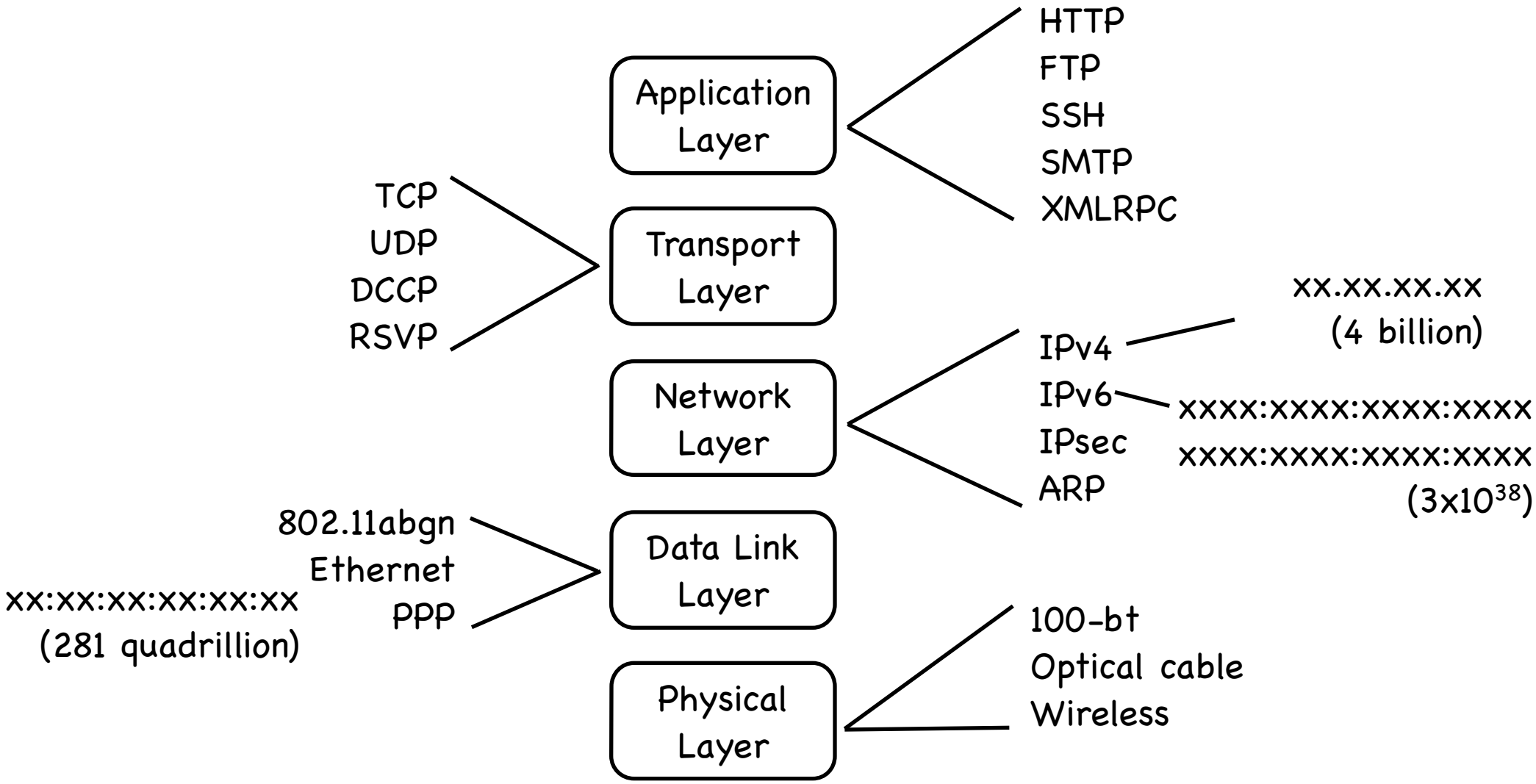
Networking


Application Layer
HTTP
FTP
SSH
SMTP
XMLRPC
Transport Layer
TCP
UDP
DCCP
RSVP
Network Layer
IPv4
IPv6
IPsec
ARP
Data Link Layer
802.11abgn
Ethernet
PPP
xx:xx:xx:xx:xx:xx
(281 quadrillion)
Physical Layer
100-bt
Optical cable
Wireless

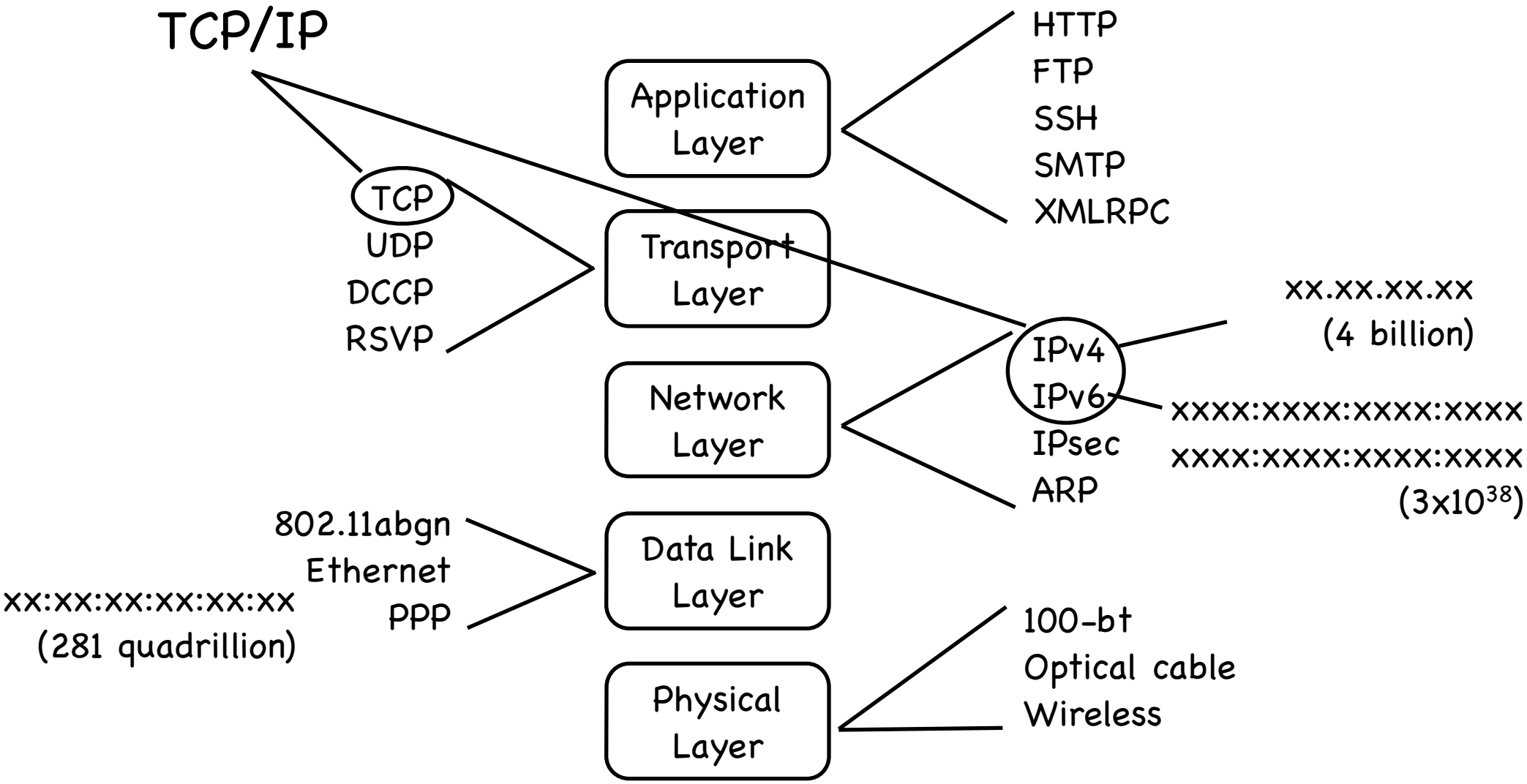
Networking


Application Layer
HTTP
FTP
SSH
SMTP
XMLRPC
Transport Layer
TCP
UDP
DCCP
RSVP
Network Layer
IPv4
xx.xx.xx.xx
(4 billion)
IPv6
IPsec
ARP
Data Link Layer
802.11abgn
Ethernet
PPP
xx:xx:xx:xx:xx:xx
(281 quadrillion)
Physical Layer
100-bt
Optical cable
Wireless

Networking


Application Layer
HTTP
FTP
SSH
SMTP
XMLRPC
Transport Layer
TCP
UDP
DCCP
RSVP
Network Layer
IPv4
xx.xx.xx.xx
(4 billion)
IPv6
xxxx:xxxx:xxxx:xxxx
xxxx:xxxx:xxxx:xxxx
(3x10^38)
IPsec
ARP
Data Link Layer
802.11abgn
Ethernet
PPP
xx:xx:xx:xx:xx:xx
(281 quadrillion)
Physical Layer
100-bt
Optical cable
Wireless

Networking


TCP/IP
Application Layer
HTTP
FTP
SSH
SMTP
XMLRPC
TCP
UDP
DCCP
RSVP
Transport Layer
xx.xx.xx.xx
(4 billion)
IPv4
IPv6
xxxx:xxxx:xxxx:xxxx
xxxx:xxxx:xxxx:xxxx
(3x10^38)
Network Layer
IPsec
ARP
802.11abgn
Ethernet
PPP
xx:xx:xx:xx:xx:xx
(281 quadrillion)
Data Link Layer
Physical Layer
100-bt
Optical cable
Wireless

IPv4 Network

- IP Address - Computer's unique* address (x.x.x.x)
- Netmask - defines local 'subnet', machines the computer can speak to 'directly'
- Router - Address used to contact machines outside subnet
- DNS Server - Address where names can be mapped to addresses
- Port - For a specific connection 0-65535, 0-1023 reserved for system services

* - Some addresses are for 'private networks'. These are 10.*.*.*, 172.16.*.*-172.31.*.*, and 192.168.*.*

Common Services

port	service
21	ftp
22	ssh
23	telnet
25	smtp (mail)
79	finger
80	http (web)
110	pop3 (email retrieval)
123	ntp (time)
137-139	Windows file sharing
143	imap (email retrieval)
443	https (secure http)

Sockets (TCP/UDP)

UDP

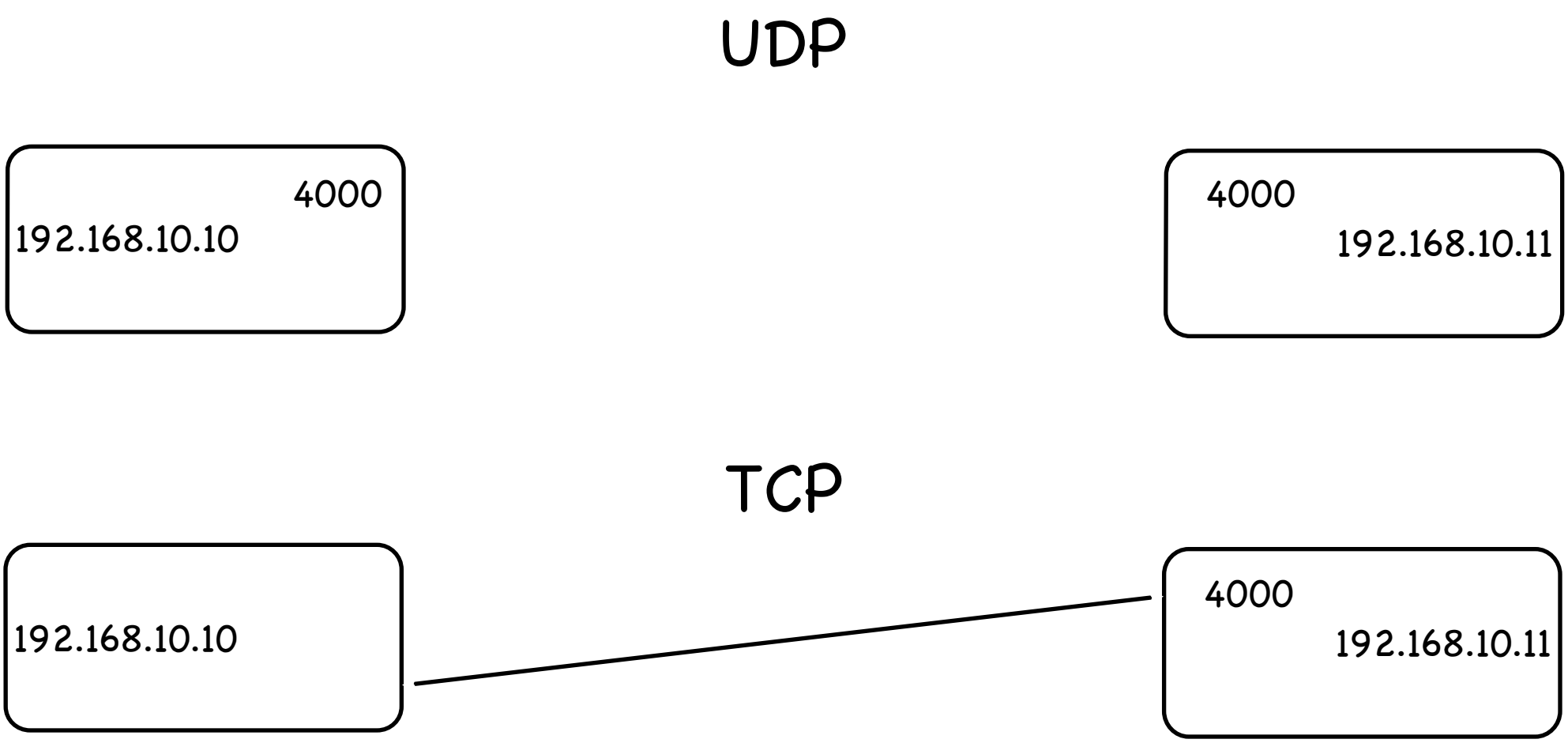
TCP

Sockets (TCP/UDP)

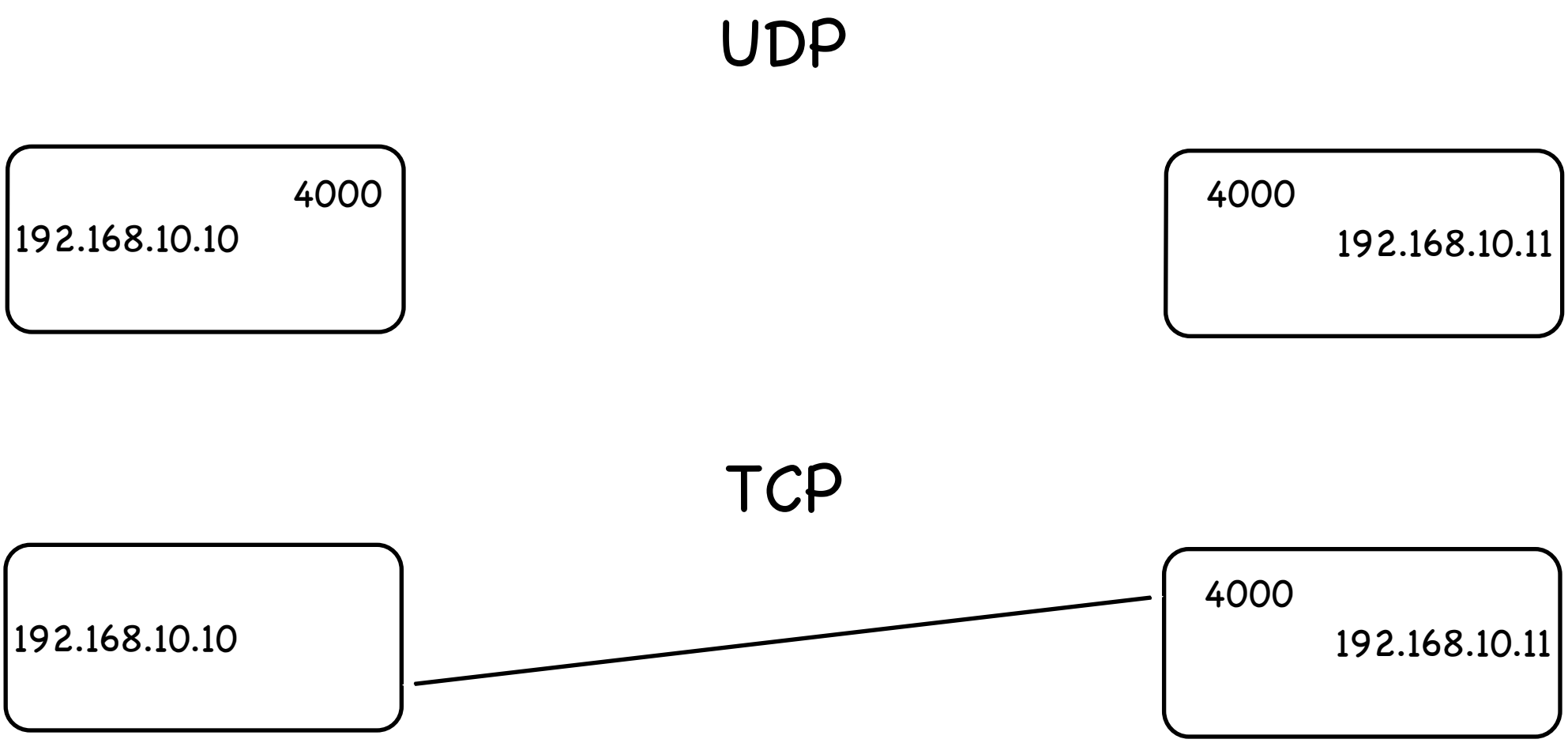
UDP

TCP

Sockets (TCP/UDP)



Sockets (TCP/UDP)

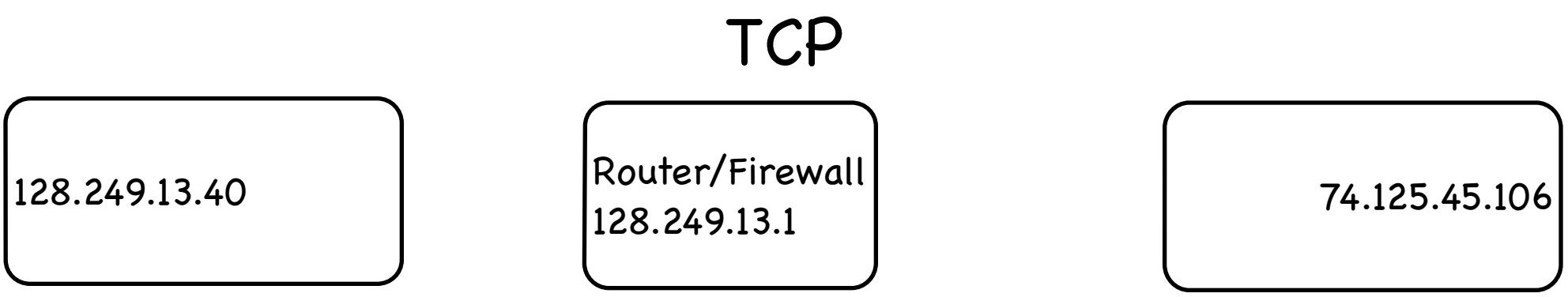


IPv4 Network Parameters

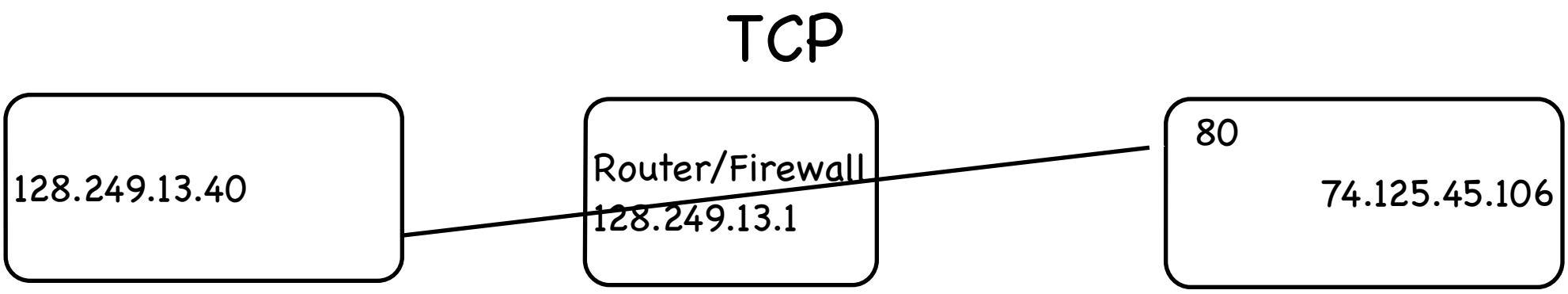
- IP Address - Computer's unique* address (x.x.x.x)
- Netmask - defines local 'subnet', machines the computer can speak to 'directly'
- Router - Address used to contact machines outside subnet
- DNS Server - Address where names can be mapped to addresses
- Port - For a specific connection 0-65535, 0-1023 reserved for system services

* - Some addresses are for 'private networks'. These are 10.*.*.*, 172.16.*.*-172.31.*.*, and 192.168.*.*

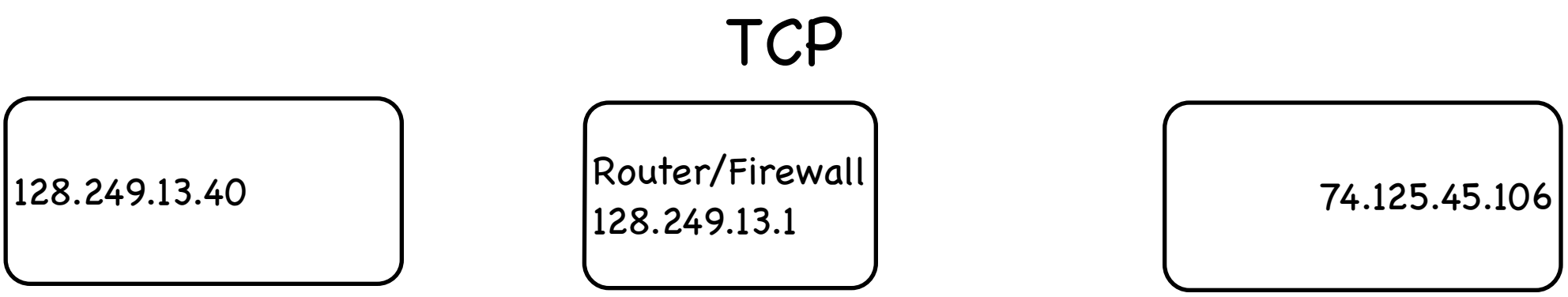
NAT



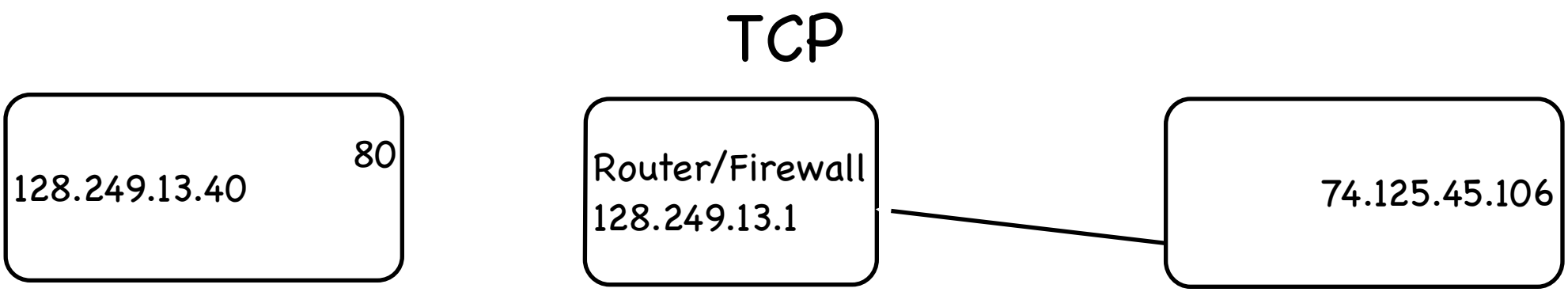
NAT



NAT



NAT

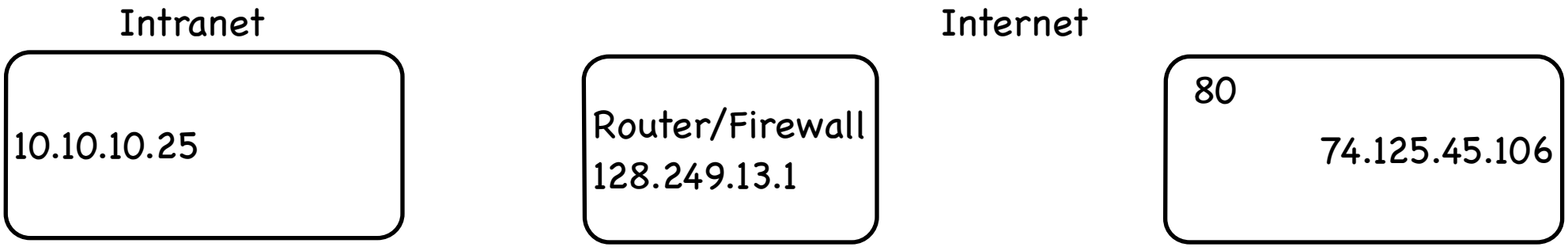


NAT

TCP

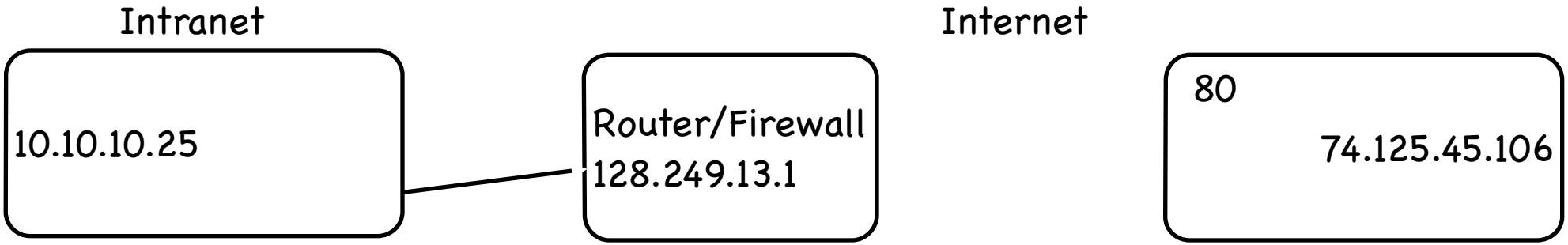
NAT

TCP


Intranet
10.10.10.25
Router/Firewall
128.249.13.1
Internet
80
74.125.45.106

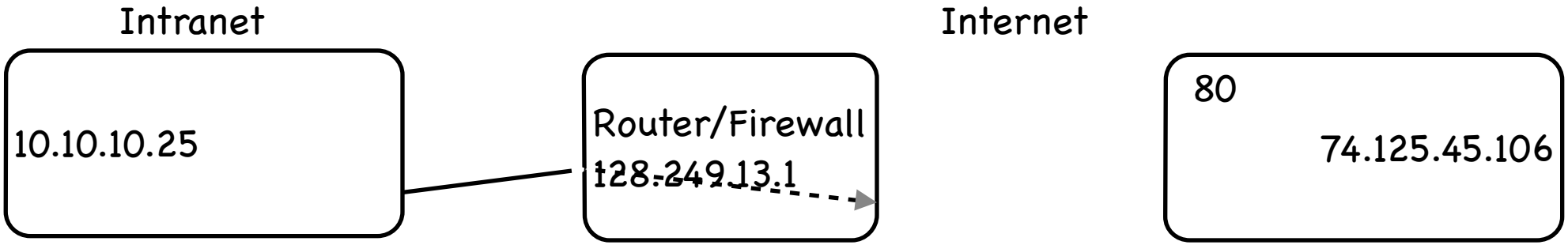
NAT

TCP


Intranet
Internet
10.10.10.25
Router/Firewall
128.249.13.1
80
74.125.45.106

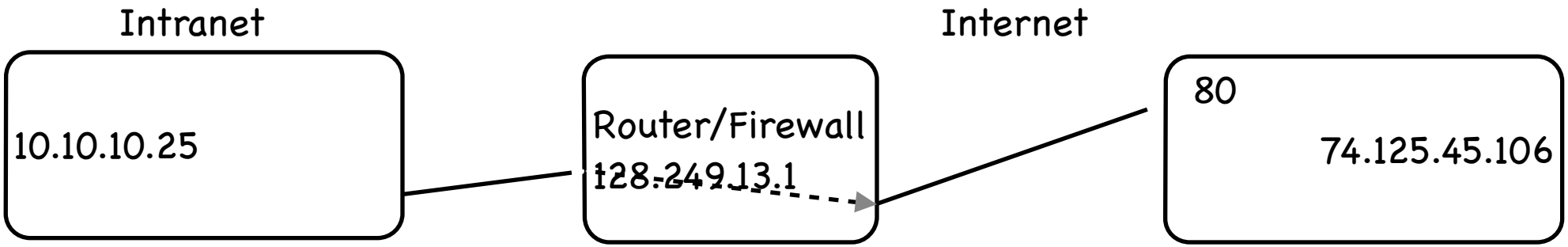
NAT

TCP


Intranet
Internet
10.10.10.25
Router/Firewall
128.249.13.1
80
74.125.45.106

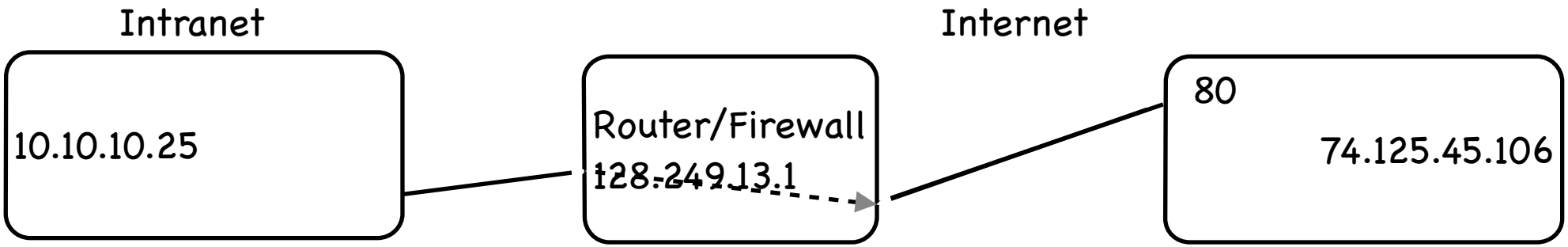
NAT

TCP


Intranet
Internet
10.10.10.25
Router/Firewall
128.249.13.1
80
74.125.45.106

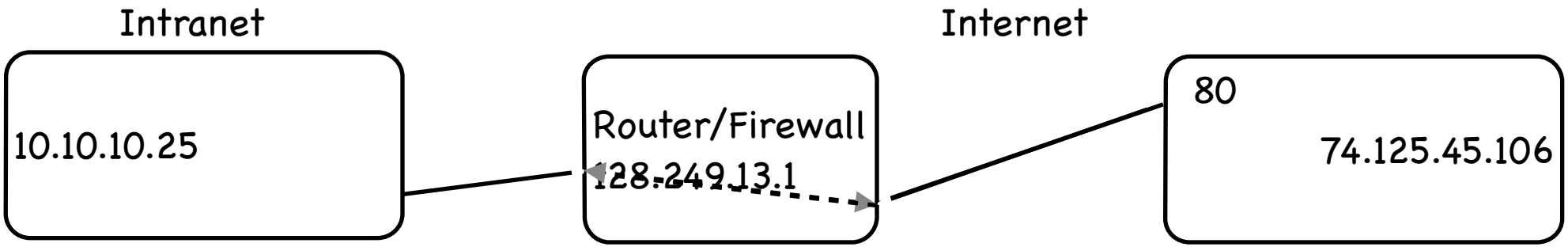
NAT

TCP


Intranet
Internet
10.10.10.25
Router/Firewall
128.249.13.1
80
74.125.45.106

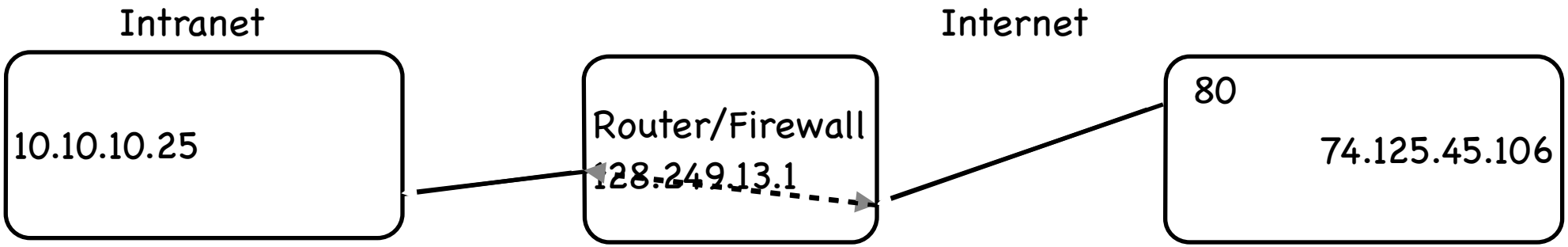
NAT

TCP


Intranet
Internet
10.10.10.25
Router/Firewall
128.249.13.1
80
74.125.45.106

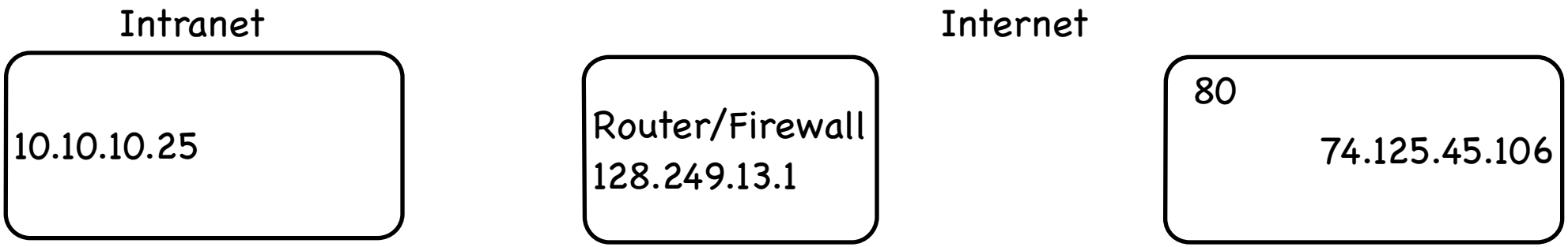
NAT

TCP


Intranet
Internet
10.10.10.25
Router/Firewall
128.249.13.1
80
74.125.45.106

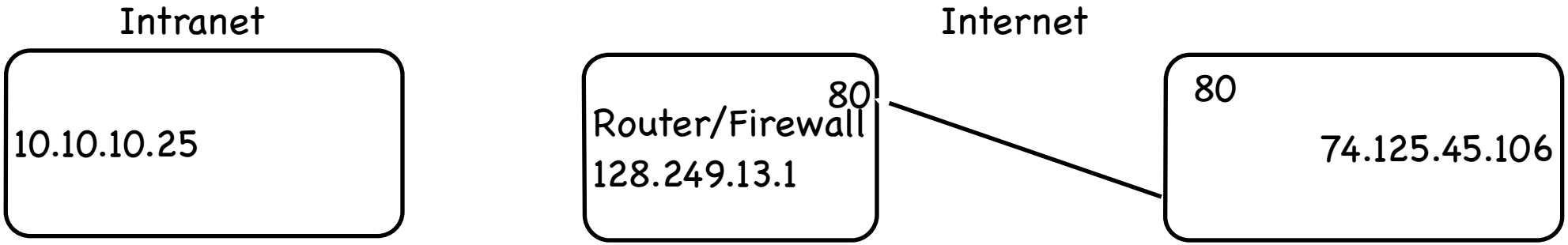
NAT

TCP


Intranet
10.10.10.25
Router/Firewall
128.249.13.1
Internet
80
74.125.45.106

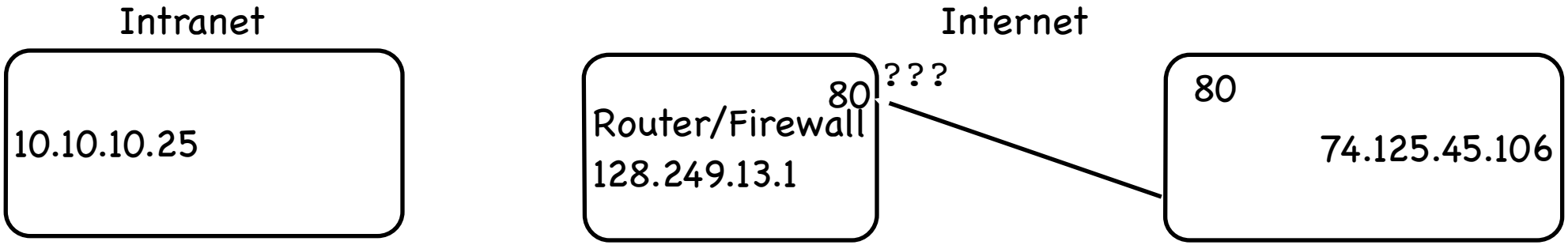
NAT

TCP


Intranet
10.10.10.25
Router/Firewall
128.249.13.1
80
Internet
80
74.125.45.106

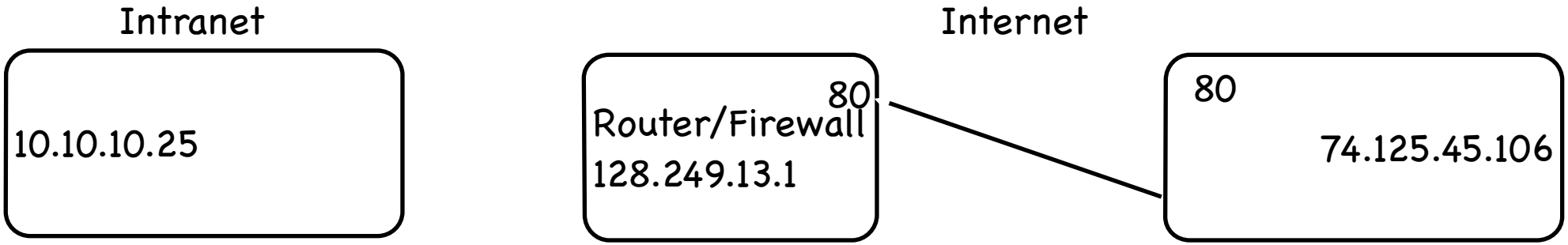
NAT

TCP


Intranet
10.10.10.25
Internet
???
80
Router/Firewall
128.249.13.1
80
74.125.45.106

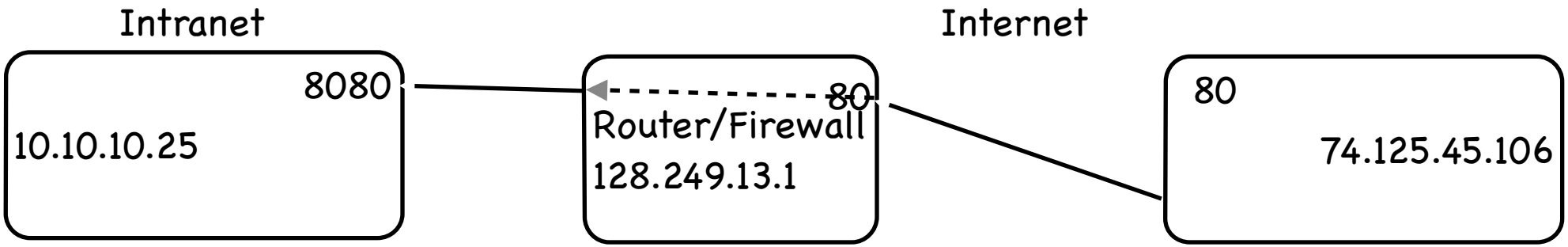
NAT

TCP


Intranet
10.10.10.25
Router/Firewall
128.249.13.1
80
Internet
80
74.125.45.106

NAT

TCP


Intranet
8080
10.10.10.25
Router/Firewall
128.249.13.1
80
Internet
80
74.125.45.106

HISTORY

port	service
ARPANET (1969)	
23	telnet (1969)
21	ftp (1971)
79	finger (1977)
25	smtp (email, 1982)
NSFNET (1986)	
70	gopher (1991)
80	http & HTML (web, 1991)
NCSA Mosaic (1993)	
443	https (secure http, 1994)
NSFNET decommissioned, Internet Commercialized (1995)	
HTML 2.0 (1995)	
22	ssh (1995)
HTML 4 - commercial infighting (1999)	
HTML5/XHTML5 (2014-2016?)	

Glossary

- HTTP - Hypertext Transport Protocol
- HTML - Hypertext Markup Lanugage
- XML - Extensible Markup Language
  - SGML - Parent of XML & HTML
- JavaScript - Python-like language on webpages (NOT Java)
- JSON - JavaScript Object Notation
- AJAX - Asynchronous JavaScript and XML (2005 Google Maps)
- AJAJ - Asynchronous JavaScript and JSON
- CSS - Cascading Style Sheet
- RSS - Really Simple Syndication

urllib2

```
import urllib2

f=urllib2.urlopen("http://blake.bcm.edu/dl/test.html")

for i in f: print i
```

HTML

- http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html
- Declarative language
- HTML is a type of XML, XHTML obeys XML rules more completely
- Python HTMLParser module
- 'commands' in HTML are denoted by
<command option=value option=value>text</command>
- For example:

<HTML>

<HEAD><TITLE>My Page</TITLE></HEAD>

<BODY>

<H3>Hi Everyone</H3>

<P>This is really just some test text to demonstrate how HTML works.
I can do interesting things like italicize or make text bold,
or even both together. ta da

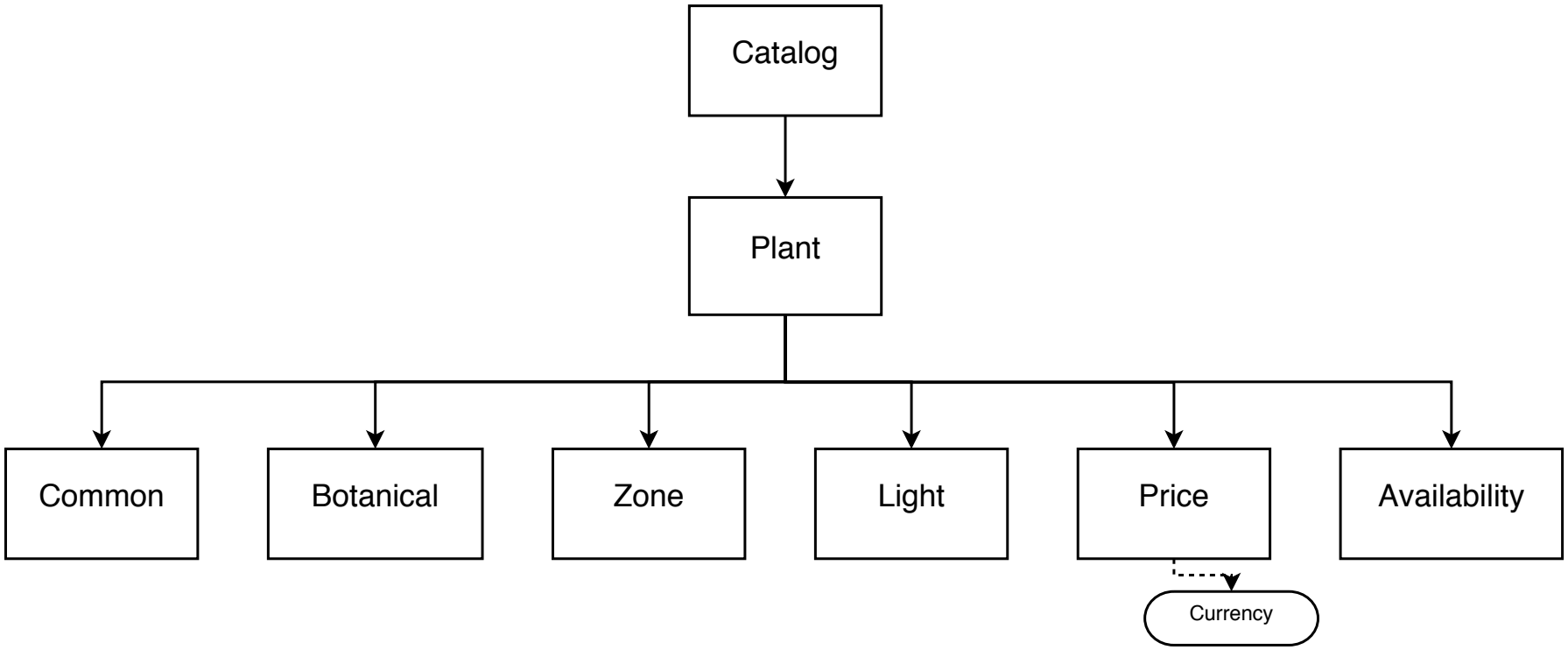
</BODY>

CSS

- Cascading Style Sheet
  - Used to present websites in a uniform way
  - Define how a page looks, then use the definition on many pages
  - <link rel="stylesheet" type="text/css" charset="utf-8" media="all" href="/moin_static185/modern/css/common.css">
  -
  -

XML Basics

- <?xml version="1.0" encoding="UTF-8" ?>
- Tags
  - <tag> content </tag> or <tag />
- Attributes
  - <tag attr1=”value” attr2=”value2”> </tag>
- Nesting
  - <tag 1>content <tag2>nested</tag2></tag1>
- Case sensitive (unlike HTML)


Catalog
Plant
Common
Botanical
Zone
Light
Price
Availability
Currency

XML Example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CATALOG>
   <PLANT>
      <COMMON>Bloodroot</COMMON>
      <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
      <ZONE>4</ZONE>
      <LIGHT>Mostly Shady</LIGHT>
      <PRICE CURRENCY="dollar">2.44</PRICE>
      <AVAILABILITY>031599</AVAILABILITY>
   </PLANT>
   <PLANT>
      <COMMON>Columbine</COMMON>
      <BOTANICAL>Aquilegia canadensis</BOTANICAL>
      <ZONE>3</ZONE>
      <LIGHT>Mostly Shady</LIGHT>
      <PRICE CURRENCY="dollar" >9.37</PRICE>
      <AVAILABILITY>030699</AVAILABILITY>
   </PLANT>
</CATALOG>
```

XML in Python

- xml.sax
  - Simple API for XML (W3C)
  - Parse XML files sequentially, callbacks
- xml.dom
  - Document Object Model (W3C)
  - View XML as a single hierarchical document
- xml.etree
  - Python specific, similar to DOM
  - Easier to use !

Using ElementTree

- import xml.etree.cElementTree (or xml.etree.ElementTree)
- et=xml.etree.cElementTree.parse("xml_example.xml")
- et=xml.etree.cElementTree.fromstring("XML CODE")
- e=et.getroot() - The root object in the XML file
- e[n] - Children of this element
- e.items() or e.attrib - An element's attributes
- e.text - Unused text between the start and end tags
- e.tag - The element's tag as a string

XML Schemas

- Schemas Specifications
  - DTD
  - XML Schema
  - RELAX NG
- Specific Schemas/Ontologies
  - http://www.bioontology.org
  - http://en.wikipedia.org/wiki/List_of_XML_markup_languages

RESTful Servers

- Generally return XML
- Client-Server - Servers store data, clients display data
- Stateless - URL uniquely identifies display
- Cacheable - Pages must declare whether their data is

Real World Example

- www.pdb.org/pdb/software/rest.do
  - Several interfaces provided

RSS

- Small XML files containing frequently updated information. Link to webpage.
- 2 variants, also Atom.
- Required elements (2.0): title, link, description
- Optional elements: language, copyright, managingeditor, webmaster, pubdate, lastbuilddate, category, generator, docs, cloud, ttl, image, rating, textinput, skiphours, skipdays

Homework 7

- Write a program that retrieves something from the web and processes it in some useful fashion (easier if you find an XML or RSS site). Turn in the program and a brief description of what it retrieves and what it does with it.

BRING YOUR LAPTOPS MONDAY !