Department of Computer Science
Southern Illinois University Carbondale

CS 491/531
SECURITY IN CYBER-PHYSICAL SYSTEMS

Lecture 2: Review of Computer Networks

DR. ABDULLAH AYDEGER

LOCATION: ENGINEERING A 409F

EMAIL: AYDEGER@CS.SIU.EDU

Outline

Introduction of Computer Networks

Layers of TCP/IP Network Protocol Suite

Internet

Network of networks

- Local/regional networks – national/global networks

Billions of connected devices

An infrastructure that enables services such as;

- Games, social media, cloud
- Similarity to the postal service

But how do they manage all of these?

Computer Networks

Connects two or more computing devices

- Computers, phones, smart grid, IoT

Various ***protocols*** between different device set

- Protocol define the rules of how they interact

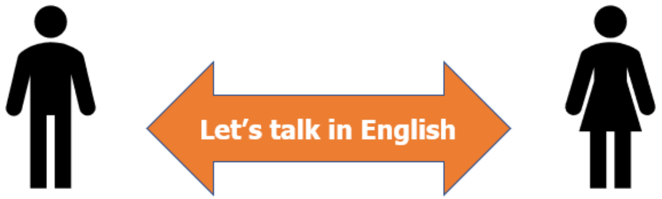
Example daily uses:

- Virtual classrooms, messaging, emails, social media, etc.

Protocols

"The official procedure or system of rules governing affairs of state or diplomatic occasions"

PROTOCOL = Set of rules to communicate.



A communication protocol:

- System of rules that allow two or more entities of a communications system to transmit information via any kind of variation of a physical quantity
- Defines the rules, syntax, semantics and synchronization of communication and possible error recovery methods

Communication/Network Protocols

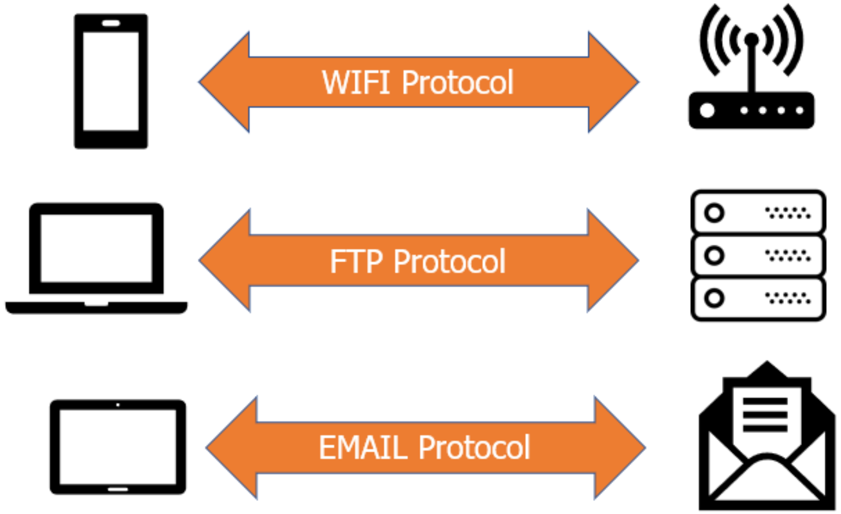
Each protocol defines different set of:

- Format of messages
- Order
- Actions
- Security?

Protocol runs on multiple nodes, and implements certain functionality of a single layer

- Works through packet header

PROTOCOL = Set of rules to communicate.



How to Define Network Protocols

Internet has billions of devices

How to manage the different devices to talk to each other

How to deal with maintenance, scalability, accountability of them?

Solution: Divide and control

Layering

A way of abstracting and organizing functionality

- without specifying implementation details

Eases maintenance, updating of system

Provides scalability

=> Leads to design protocol stack by creating different layers for different tasks

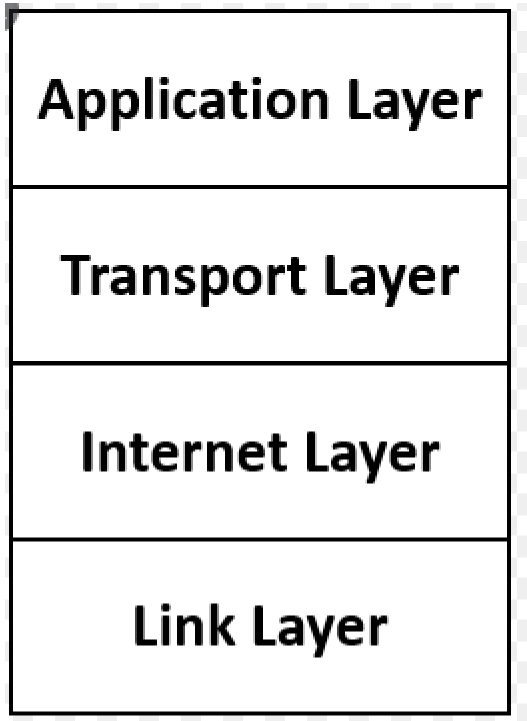
Network Protocol Stack

Application Layers: End-user applications

Transport Layer: Data transfer from end to end

Internet (Network) Layer: Routing of data from source to destination

Link (Physical) Layer: Physical media carrying the data



Application Layer

Transport Layer

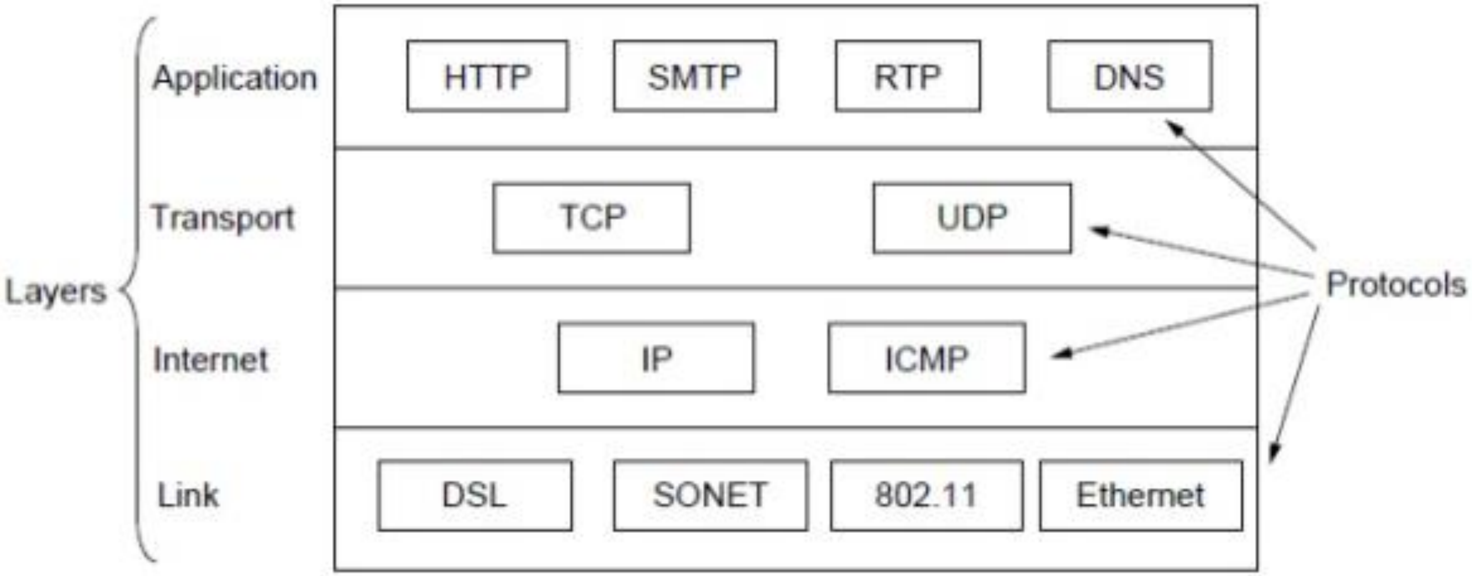
Internet Layer

Link Layer

Network Protocols in different Layers

Each layer has its own protocols

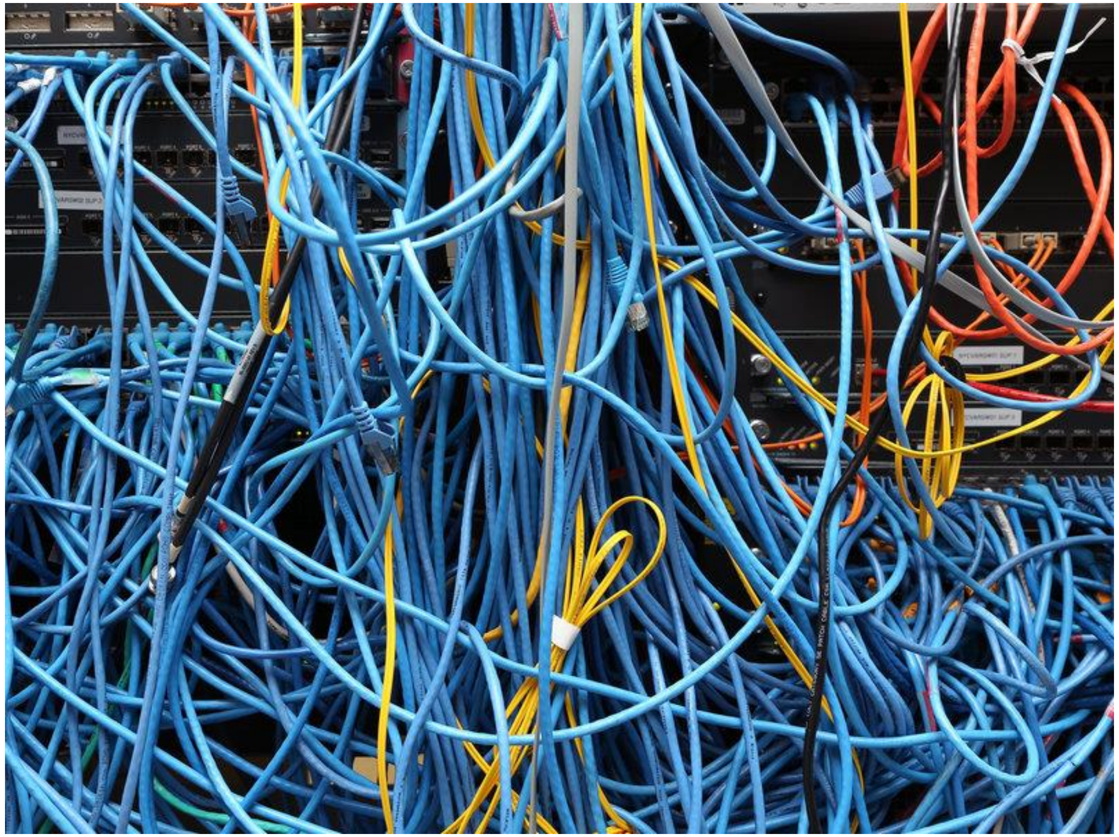
A few examples:



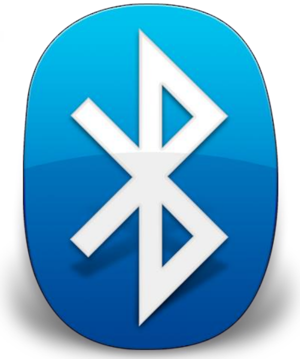
Link Layer

Link = Medium + Adapters

Enable host-to-host communication within a single local area network (LAN)



Some “Link” Examples



Some "Network Adapters"

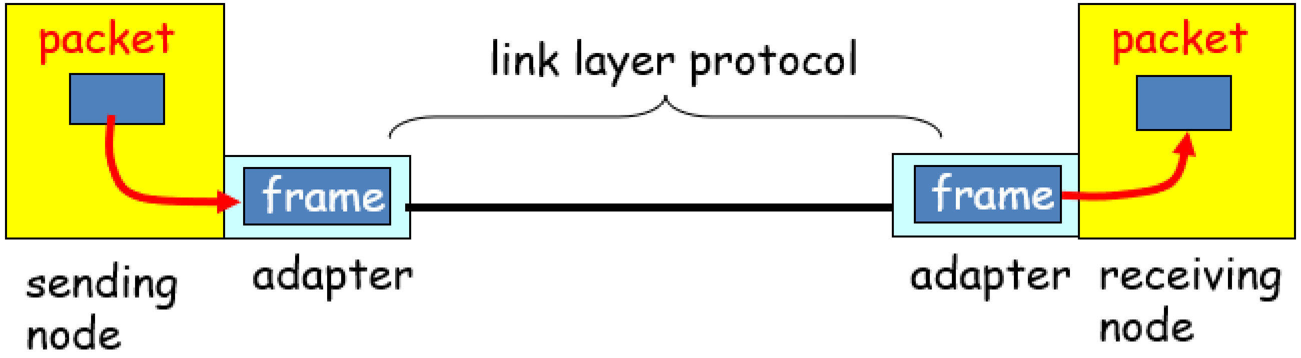
Link Layer: Functionalities

Sending side

- Encapsulates packet in a frame
- Adds error checking bits, etc.

Receiving side

- Looks for errors then decapsulates
- Extracts datagram and passes to receiving node



Link Layer: Essential Components

MAC (medium access control) Addresses

- *Unique*, hard coded to adapters
- 48 bits, an example: 05-ff-13-32-a6-55

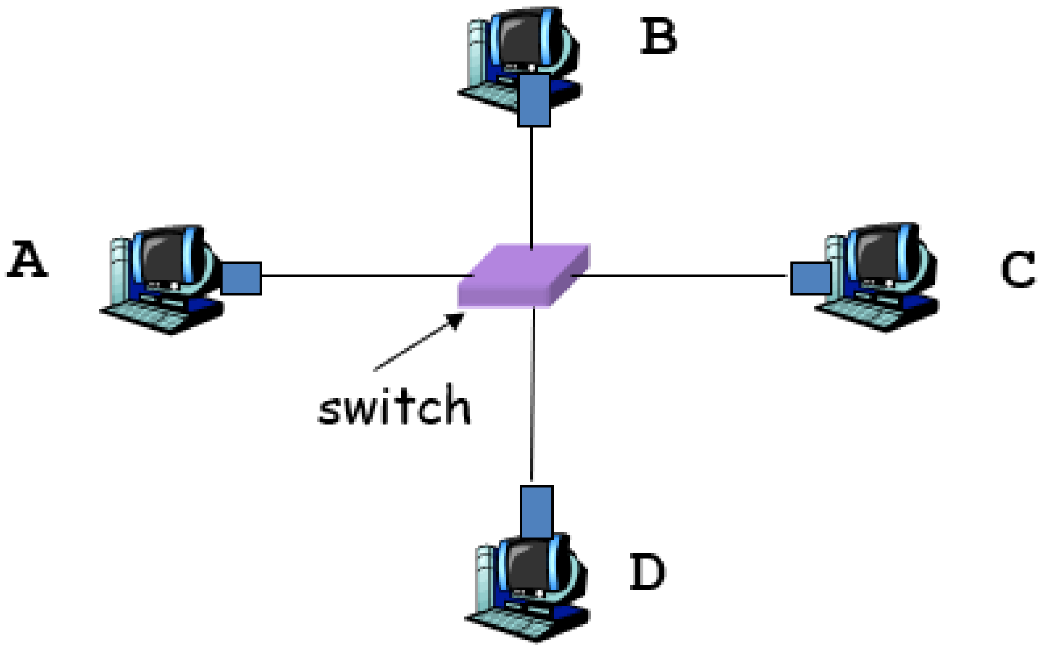
Switches

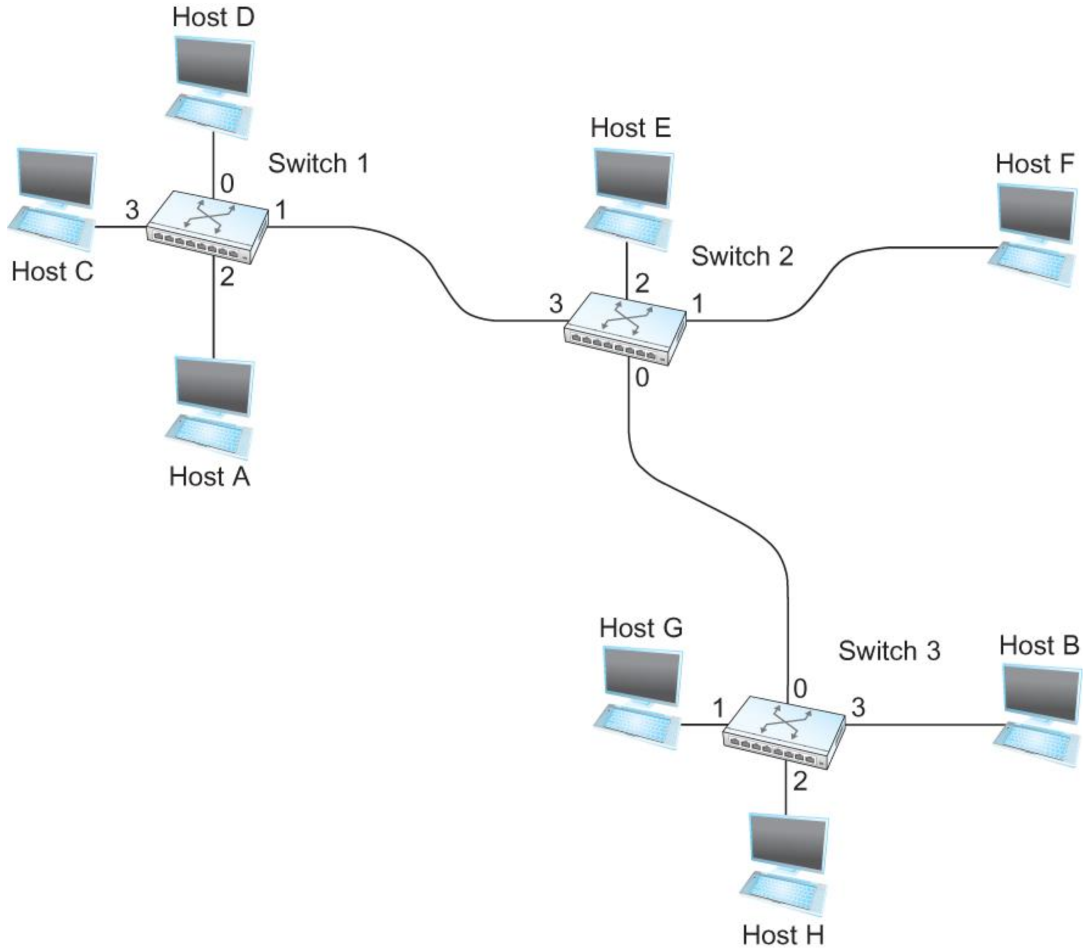
- Can connect to host or another switch
- Provides traffic isolation and concurrent communication
- Maintains forwarding table

Destination	Port
B	0
C	3
D	3
E	2
F	1
G	0
H	0

Forwarding Table for Switch

Link Layer: Topology Examples





Discussion - 1

Q: Can we build the entire Internet as one big LAN?

Answer: No;

- Inefficient
- Large tables
- Poor performance

Discussion - 2

Q: Is there any devices out there that have same MAC addresses? If so, what would happen?

Answer:

- (1) Probably yes.
  - Limits of 48 bits address and the story of old MAC addresses
- (2) If they are in different networks (different domain or Local Area Network(LAN)) - there will not be any problem. If they are in same LAN, neither computer can communicate properly and collisions will occur.
  - This is similar to student ID, think about it in detail.

Network (Internet) Layer

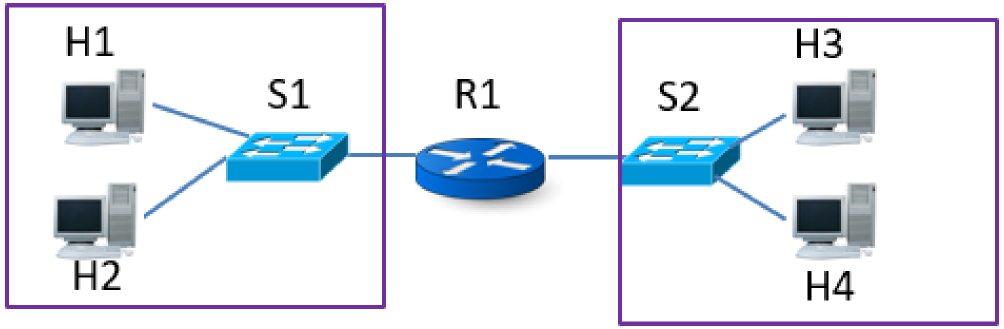
Connecting networks

- Forwarding: move from one port to another
- Routing: calculate the route it should take to arrive destination

Internet Protocol (IP): IP addresses

- IPv4, e.g. 157.23.54.201
- IPv6, e.g. 3002:0db8:85a3:ffff:0000:8a2e:0370:7114

Routers are the main devices

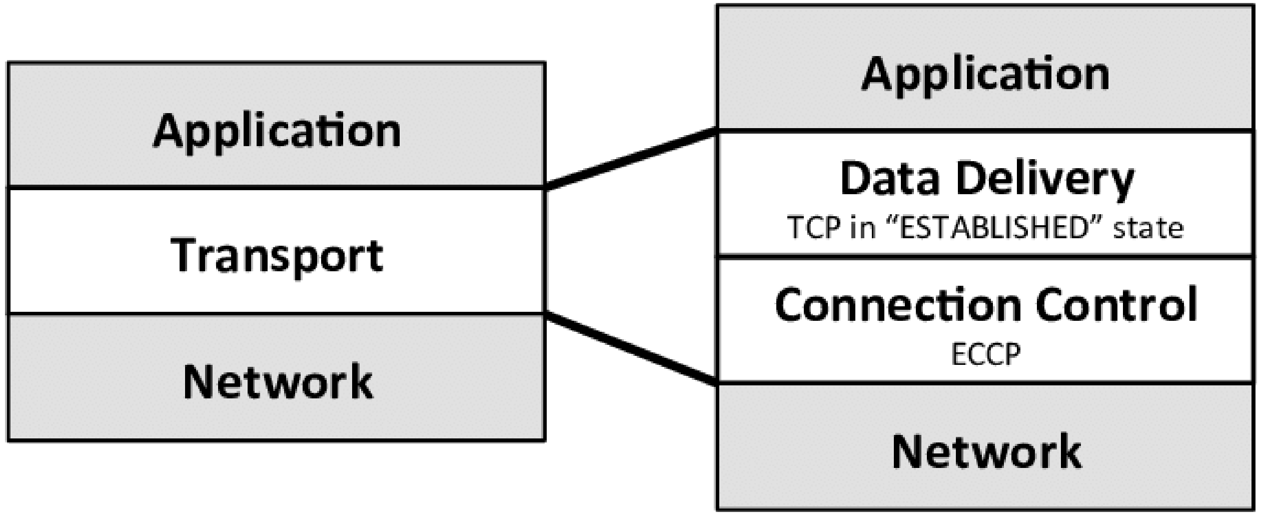


Transport Layer

Layer 1 & 2 deal with forwarding packets from one place to another.

- Thus mechanisms for finding paths, locating destination, etc.

Layer 3 provides two extra functionality on top of forwarding



Transport Layer: Functionalities

Multiplexing: Enable concurrent data streams to different application processes

- Using "port numbers" to distinguish different processes
  - Port number here is simply a logic number
  - Not get confused with switch/router port (which is a physical interface)
  - The TCP/UDP header contains source/destination process port number

Value-added services: These services include in-order delivery, end-to-end reliability, congestion control, etc.

- Required by many applications
- But not provided by the network itself
  - IP provides best-effort service only

Transport Layer: Functionalities

TCP service:

- *connection-oriented:* setup required between client and server
- *reliable transport* between sender and receiver
- *flow control*: sender won't overwhelm receiver
- *congestion control*: throttle sender when network overloaded
- does not provide: timing, minimum throughput guarantee, **security**

UDP service:

- *unreliable data transfer* between sending and receiving processes
- *does not provide*: reliability, flow control, congestion control, timing, throughput guarantee, **security**, or connection setup

Application Layer

It defines:

- types of messages exchanged,
  - request, response
- message syntax:
  - what fields in messages & how fields are delineated
- message semantics
  - meaning of information in fields
- _rules_ for when and how processes send & respond to messages

Open protocols:

- defined in RFCs
- allows for interoperability
- e.g., HTTP

Proprietary protocols:

- E.g., Zoom, Skype

Data Communication through Network Layers

Application data; user input

Transport layer adds its header

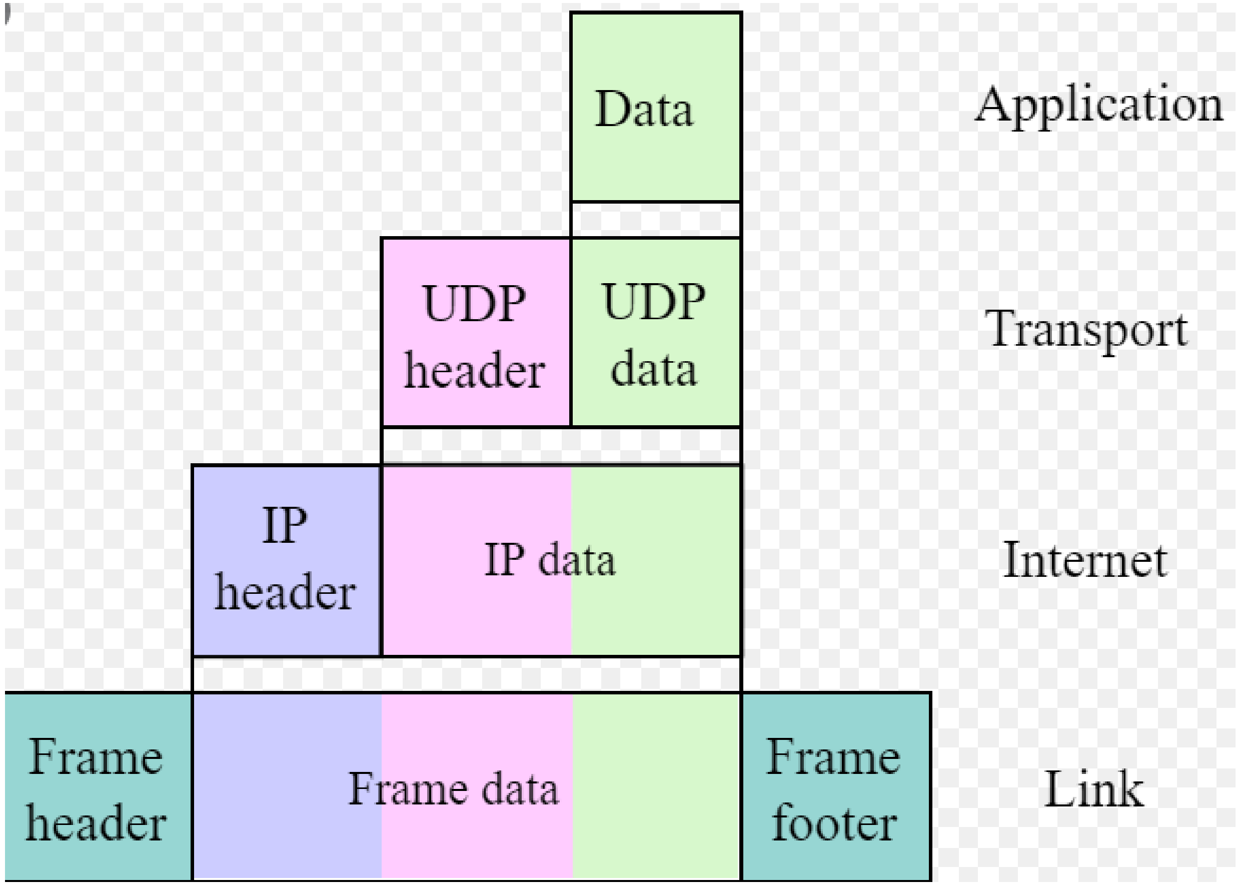
- TCP or UDP

Internet layer adds

- IP header (or ICMP)

Link Layer adds extra info

- Such as error correction, etc.

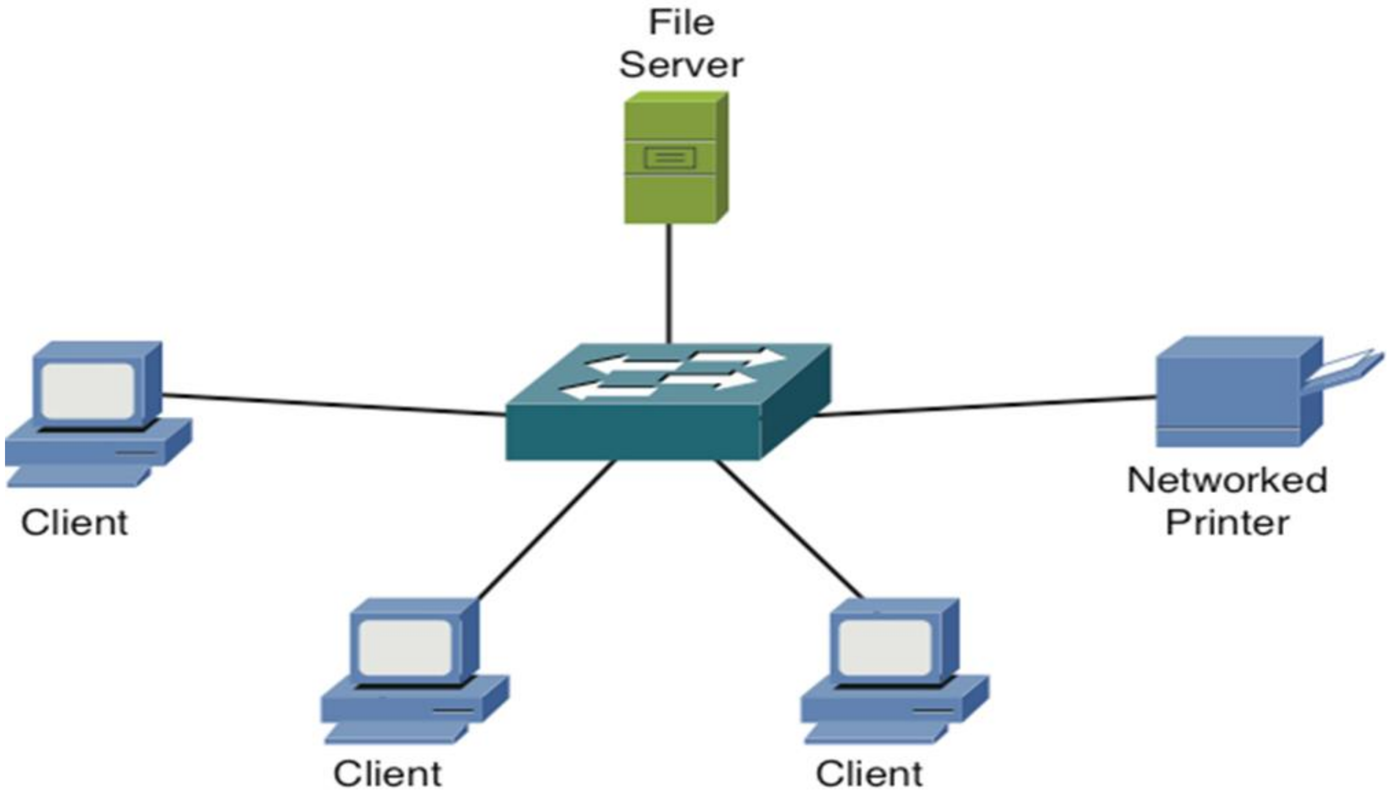


Networks by Resource Location

Client-Server

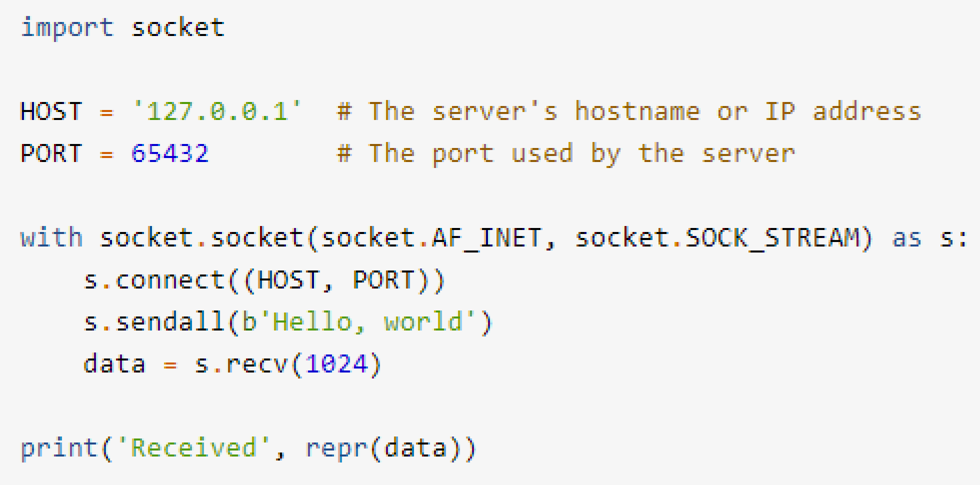
- Dedicated file server
- Shared access to files
- Networked printers
- Easier administration

HTTP Server



Client – Server Application

Python Program for Basic Data Communication

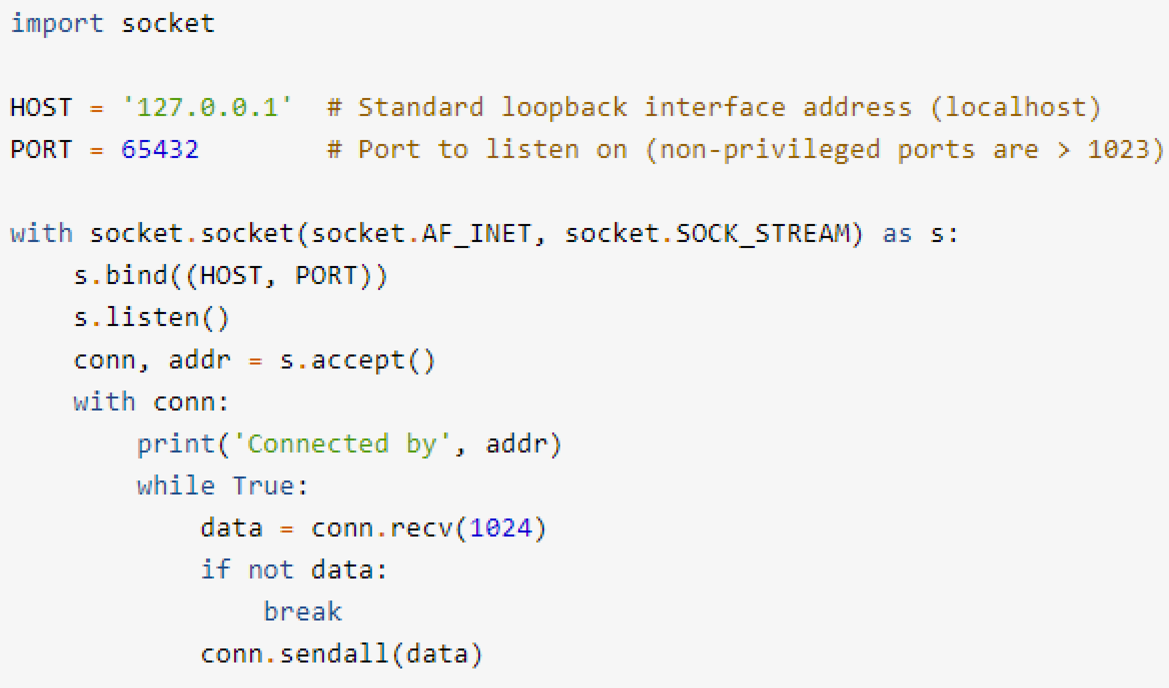


```
import socket

HOST = '127.0.0.1'  # The server's hostname or IP address
PORT = 65432        # The port used by the server

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(b'Hello, world')
    data = s.recv(1024)

print('Received', repr(data))
```



```
import socket

HOST = '127.0.0.1'  # Standard loopback interface address (localhost)
PORT = 65432        # Port to listen on (non-privileged ports are > 1023)

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
    with conn:
        print('Connected by', addr)
        while True:
            data = conn.recv(1024)
            if not data:
                break
            conn.sendall(data)
```

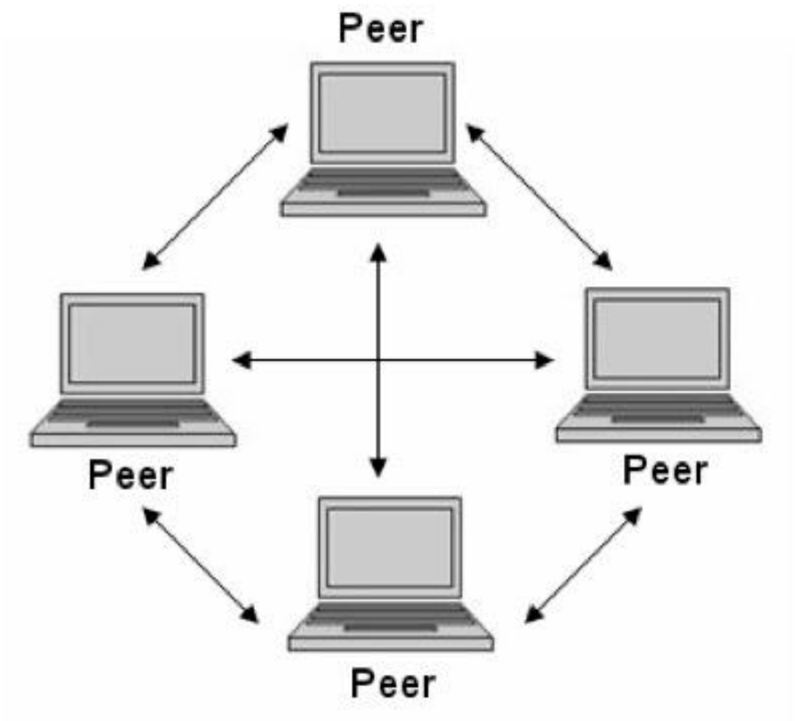
Networks by Resource Location

Peer-to-Peer

- Directly attached printers (shared)
- Scalability issues
- E.g., files shared between PCs

Torrent?

Bitcoin?


Peer
Peer
Peer
Peer