Lecture 1

Introduction and Overview

Computer Networks

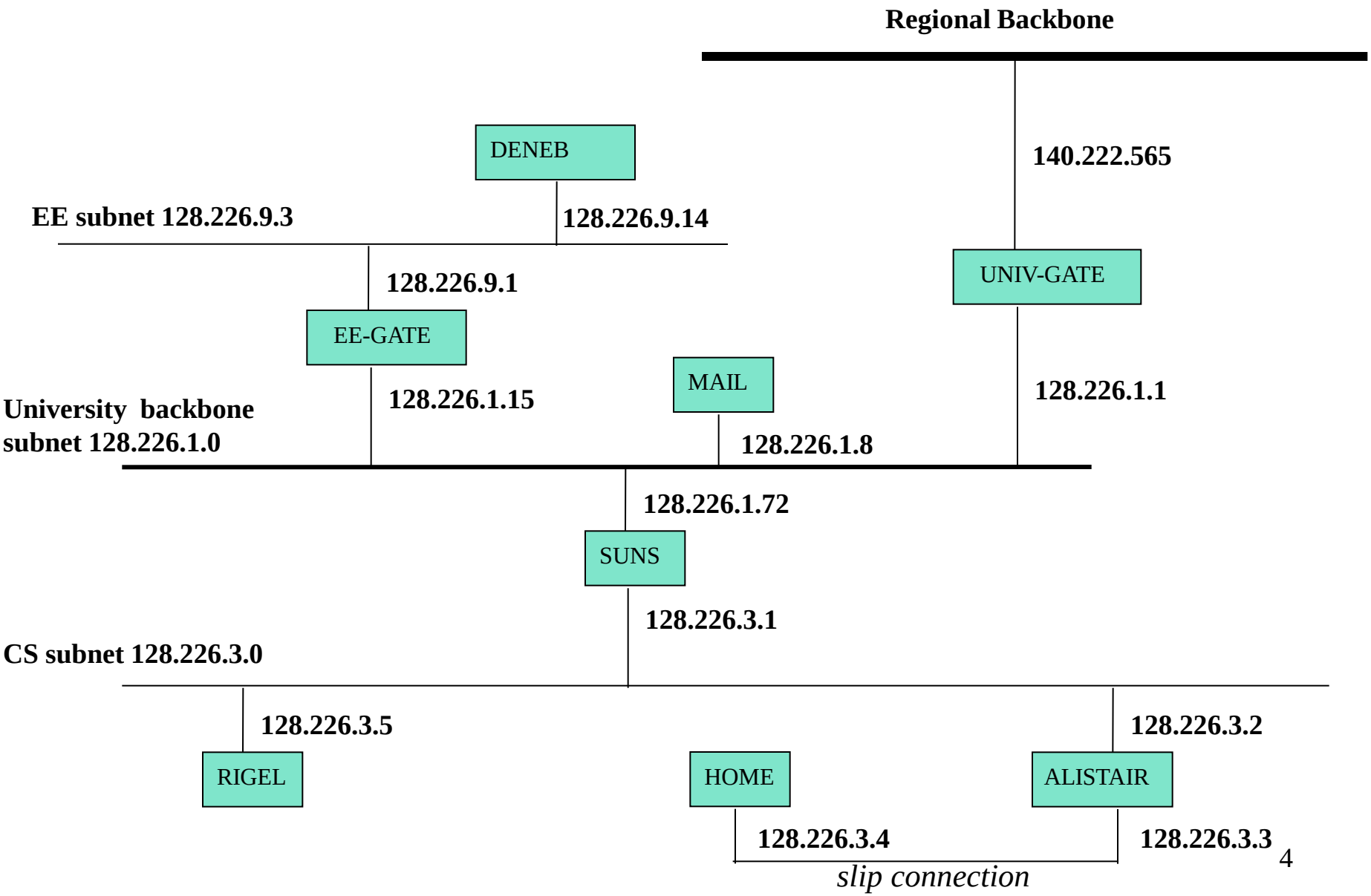
Classification of Computer Networks

- Based on physical scope
  - Personal Area Networks (PANs)
  - Local Area Networks (LANs)
    - Campus, building, room
  - Metropolitan Area Networks (MANs)
    - Cover a city – up to perhaps 50 miles in length
  - Wide Area Networks (WANs), remote networks, internets, intranets, regional
    - Cover a larger area typically beyond a city or state

Other types of Classification

- Based on Technology and Features
  - Using Wireless (Example: Wireless LAN)
  - Using Satellites
  - Using Radio
  - Public Switched Telephone Network
  - The Mobile Telephone System 1G, 2G, 3G, 4G
  - Cable Television
  - Sensor Network

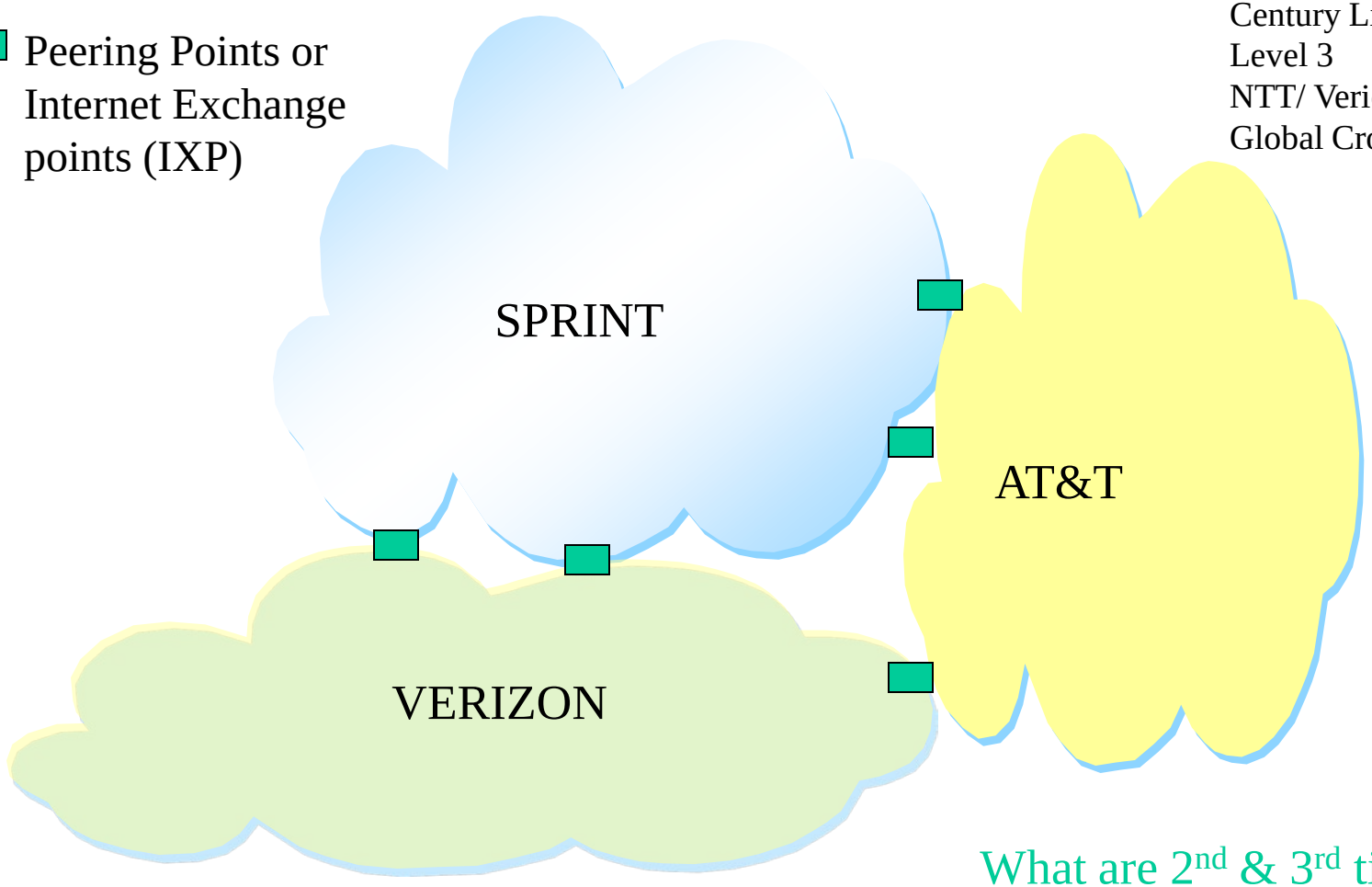
CLASSICAL LAN EXAMPLE



WAN EXAMPLE – FIRST TIER ISPs (U.S.)

(Transit without a fee or settlement-free peering)

Peering Points or Internet Exchange points (IXP)



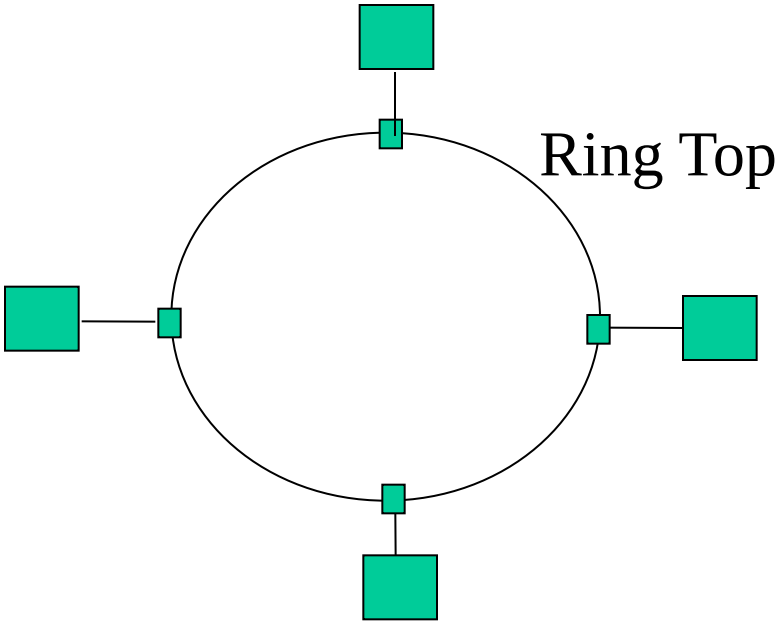
Other U.S. Tier 1
Century Link (Quest)
Level 3
NTT/ Verio
Global Crossing

What are 2nd & 3rd tier ISPs?

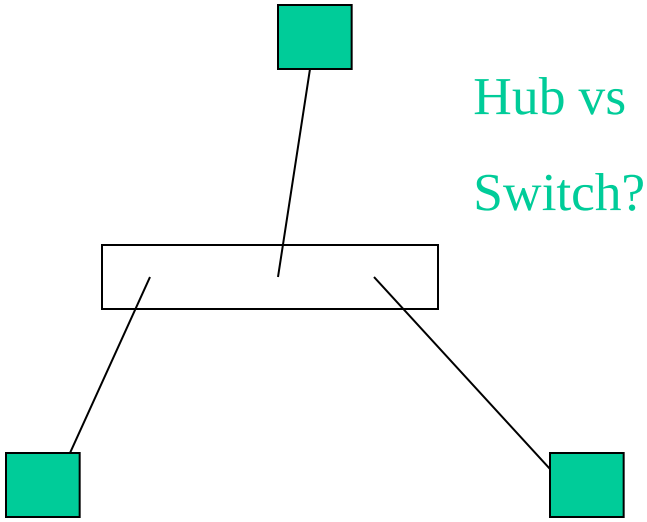
- Based on traffic type
  - –Telephone / voice, PSTN, leased lines
  - –Data networks
  - –Converged networks
- Digital vs Analog
  - –Signaling methods
  - –Capacity theorems

- Based on Topology
  - Bus based
  - Ring
  - Star
  - Mesh
  - Fully connected, point-to-point, broadcast
  - Adhoc

Bus Topology



Ring Topology



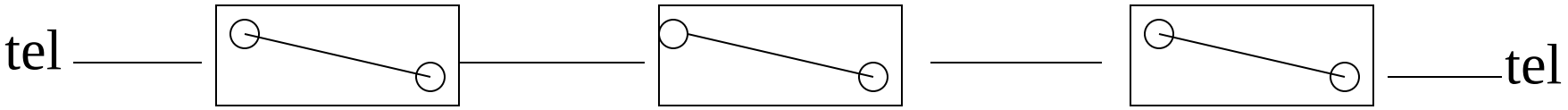
Hub vs Switch?

Star Topology

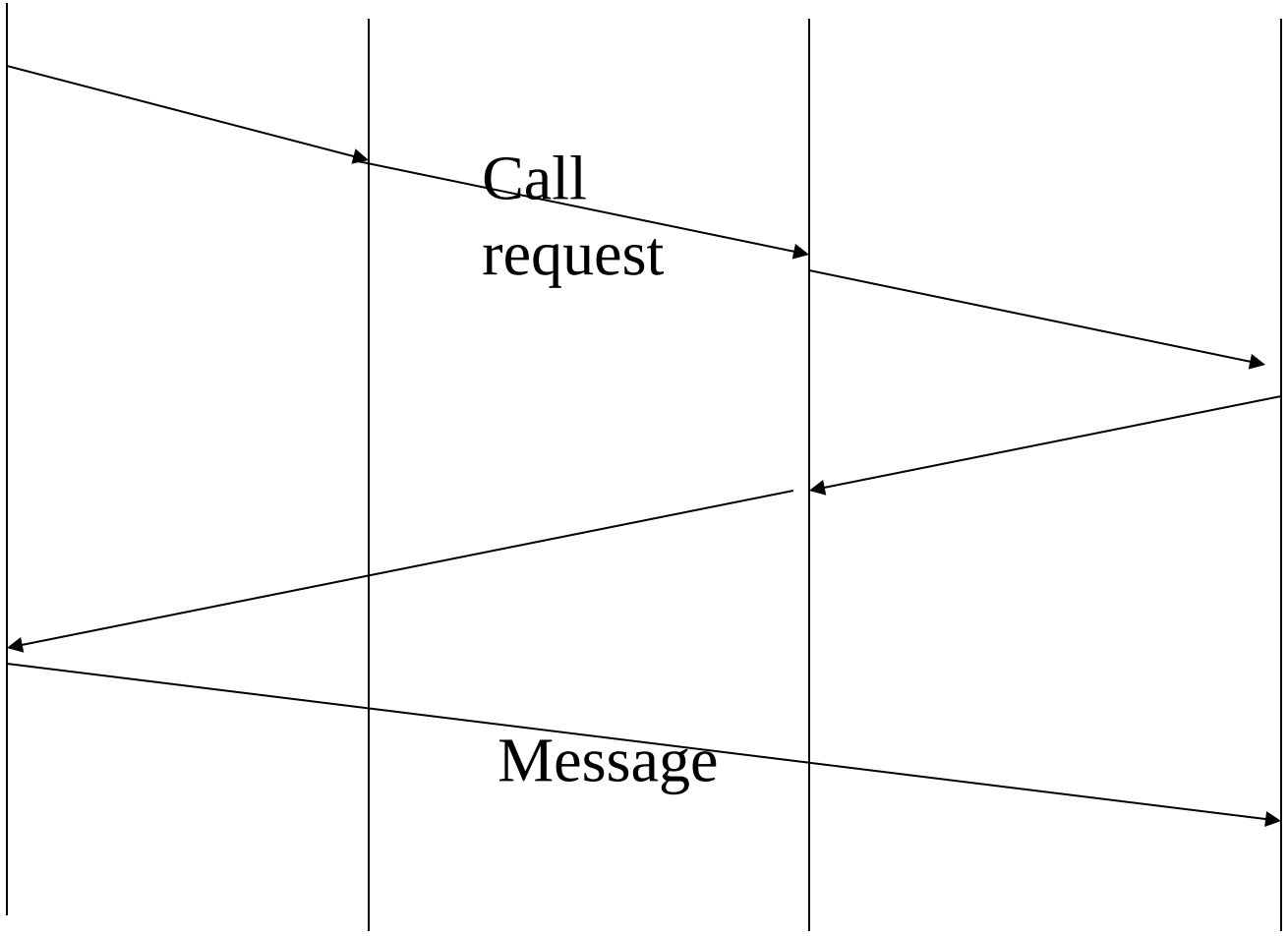
- Based on Switching
  - Circuit Switched
  - Message Switched
  - Packet Switched
  - Virtual Circuit

Circuit Switching

- Path is set up end-to-end before any data is sent. Then a dedicated path from source to destination is maintained until teardown
  - Connection phase: some *setup* delay
  - Information transfer phase: *transmission* delay
  - Disconnection phase: some delay

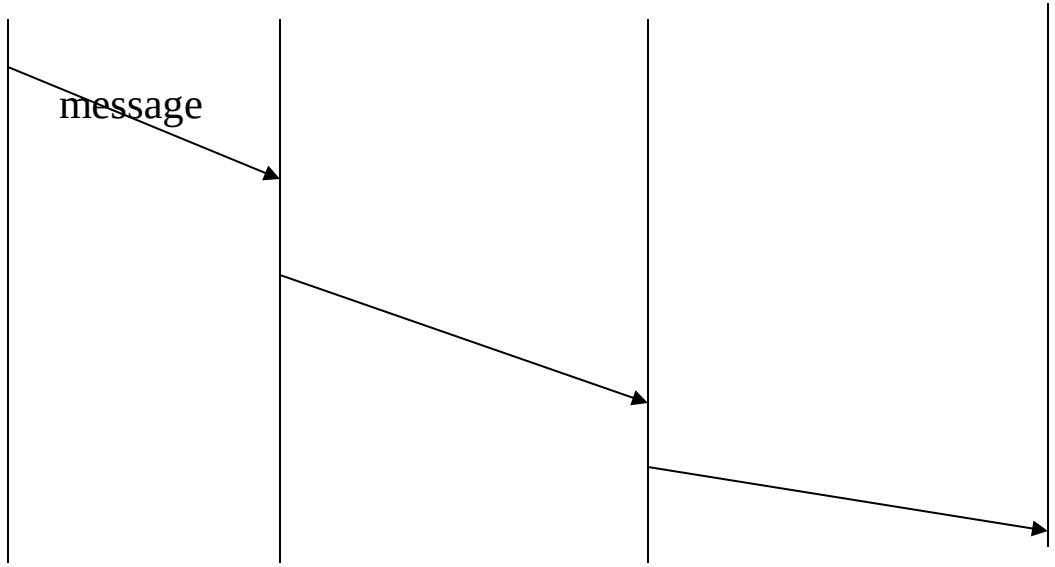


- Timing Diagram



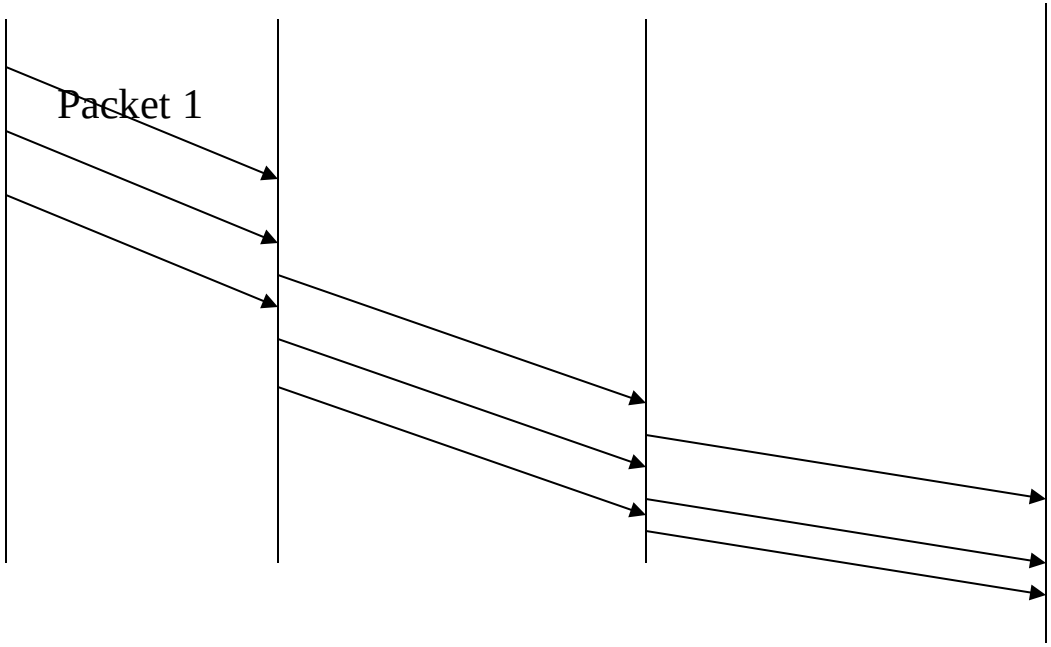
Message Switching

- Store and Forward – No setup or disconnection delay, but queuing delay, overhead in message for routing



Packet Switching

- Store and Forward
- Pipelining and multiple paths
- Retransmission of erroneous packets easier
- More overhead, fragmentation & reassembly, sequencing



Virtual Circuit

- Virtual circuit mimics a circuit switched connection (to some extent) by using packet switching technology
  - Virtual circuit is set up, typically by the initial packet(s)
  - Subsequent packets follow the virtual circuit
  - Switches do more work to maintain vc information
- ? *What is MPLS – Multiprotocol Label Switching*

Multiplexing

- Combining several “logical streams” over a physical medium or a logical medium
- Two main types of multiplexing are:
  - Frequency division multiplexing (also called wave division multiplexing)
  - Time division multiplexing
- *What is framing?*

Layered Architecture

- Why layering?
  - Network software is very complex: naming or addressing, fragmentation/reassembling of packets, multiplexing of packets, forwarding and routing, handling errors, ……
    - How to deal with complex software? Divide-and-conquer
    - The layered model simplifies the design
  - Heterogeneity in the network environment
    - Different machines, switches, links, interfaces from different vendors.
    - Solution: break up the system into different layers. Each layer provides an abstraction for its upper layer.
    - E.g. connections: modem, Ethernet, Token ring--> a *link* in the second layer. Third layer will then only be concerned with *link*s.

- Layering is a useful abstraction
- Another example of abstraction: programming languages.
  - Machine language (low level, works only on one machine)
  - Assembly language (higher level, works for several machines)
  - C++ (works on almost all machines).

- Differences between layering in the network software and layering in programming language?
  - Programming language:
    - » interface between upper layer and lower layer
  - Network software:
    - » Interface between upper layer and lower layer
    - » Flow of information up and down
    - » Interface between peers in the same layer.

– Some terminology:

- *entity*: an active element in a layer (machine, procedure, process, I/O chip).
- *Peer entities*: entities on the corresponding layers on different machines. Exchange well-defined pieces of information
- *Message*, *packet*, *frame:* structured sequence of bits that are exchanged
- *Protocol*: a set of coordination rules governing the communication between two peer entities.
  - Each layer has a protocol -- layer n protocol.
- *Service interface*: the interface between upper layer and lower layer.
  - Upper layer: *service user*. Lower layer: *service provider*.

- Service types:
  - connection-oriented service & connectionless service.
    - Connection-oriented -- like the telephone.
      - » Establish the connection, use the connection, then release the connection. Even when multiplexed, minimal header information
      - » The packets received are in the same order as the packets sent.
    - Connectionless (datagram) -- like the postal system.
      - » Each message carries its destination's address and is routed to the destination independently.
      - » The packets received may not be in the same order as the packets send.
  - reliable and unreliable
    - reliable: all packets sent are received correctly and in order
    - unreliable: packets sent may not be received, or may be received out of order. *Is a packet that is received correct?*

- Service types:
  - four choices: reliable connection-oriented, reliable connectionless, unreliable connection-oriented and unreliable connectionless.
  - Example:
    - send 1, 2, 3, 4, 5
    - receive: 1 2 3 4 5, 1 3 2 5 4, 1 2 4, 3 1 4.
  - *What is the difference between connection-oriented and streaming?*
  - Why not just reliable connection-oriented?
    - too costly to support connection-oriented services at all layers.
    - Sometimes applications may not need such service:
      - » telephone: unreliable connection-oriented.
      - » Junk mail: unreliable connectionless.
  - *How do we handle audio and video?*

- Elements of a protocol:
  - Coordination Rules for each participating protocol entity
  - syntax: what is a valid message in terms of structure?
  - semantics: what is a valid message in terms of meaning?
  - Is timing important?: relative order of messages.

  - E.g. IPX packet:

  checksum | length | transp. Control | type | dst | src | data

- Network architecture: a set of layers/protocols/service interfaces that define how functionality is divided up.

- Reference models:
  - ISO: International Standards Organization
  - OSI: Open Systems Interconnection.
  - Seven layers ISO/OSI reference model:

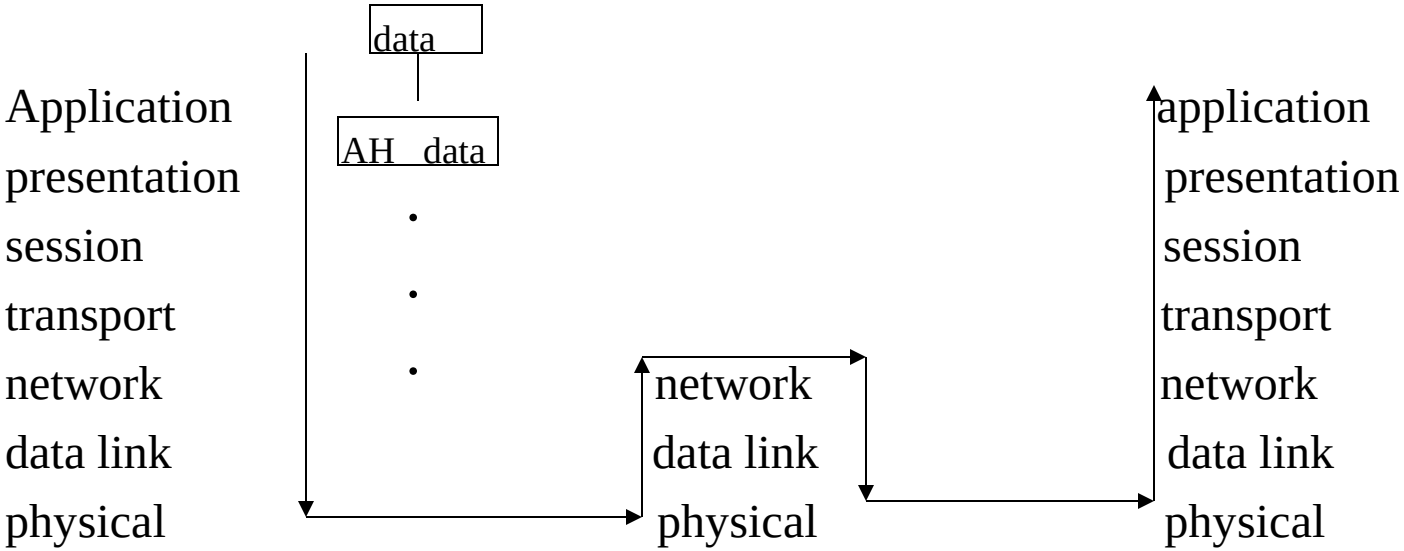
Application		Application
Presentation		Presentation
Session		Session
Transport		Transport
Network	Network	Network
Data Link	Data link	Data link
Physical	Physical	Physical

- Physical layer: how to transfer bits correctly
  - conversion of bits into signals, what is 0, 1? How long does a bit lasts? How many pins in a connection?
- Data link layer: how to transfer frames correctly
  - reliably transfer frames over a link, how to identify a frame, error control, speed mismatch between senders and receivers.
  - Divided into Media Access Control (MAC) and Logical Link Control (LLC) layers
- Network layer: how to send a packet to the destination (hop by hop)?
  - Forwarding, routing, congestion control, format conversion (internetworking), accounting
- Transport: end to end communication.
  - First layer that runs at end points but not at intermediate hops.
  - Connection establishment/management/termination, error control/flow control/multiplexing
  - Reliability

- Session layer: allows users to establish session, enhanced services.
  - Checkpointing.
- Presentation layer: provides general solutions to users.
  - Compression, syntax conversion, cryptography
- Application layer: variety of protocols that are commonly used.
  - Email, FTP, Telnet.

- Functionality of the network software are partitioned into different layers.
  - Flow control (speed mismatch between two machines): data link and transport
  - Routing: network layer
  - addressing: almost all layers
  - ……

– Data transmission using the OSI model:



DH	NH	TH	SH	PH	AH	Data

Assuming 100Mbps Ethernet is used to send 30 bytes (user) data, let AH=PH=SH=TH=NH=DH=10, what is the maximum throughput the application can observe?

- OSI model was not very successful!!
  - The TCP/IP protocols became the de facto network software standard.
  - From the TCP/IP protocols, people derived the TCP/IP reference model.
- TCP/IP reference model:

  Application layer (Telnet, FTP SMTP, DNS, NNTP, HTTP)

  Transport layer (TCP, UDP)

  Internet layer (IP)

  Host to Network layer (Ethernet, FDDI, X.25)

- TCP/IP reference model:

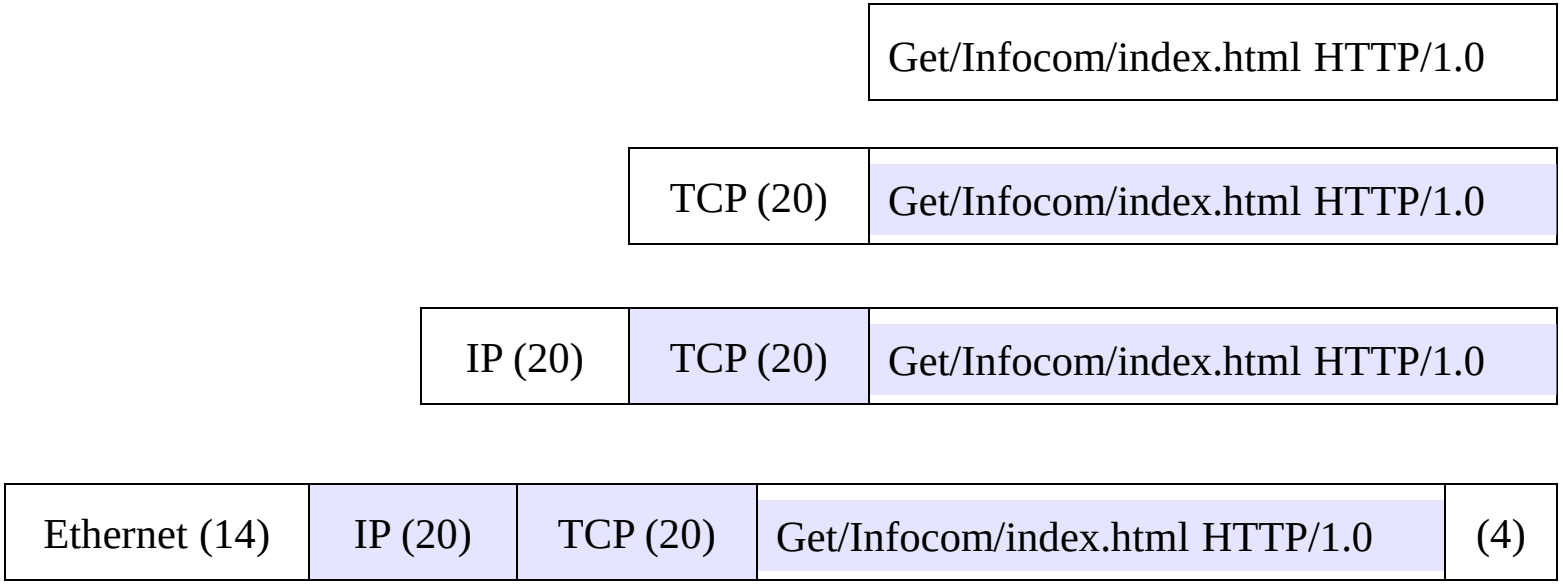
  Application layer (Telnet, FTP SMTP, DNS, NNTP, HTTP)
  - No session and presentation layers.

  Transport layer (TCP, UDP)
  - Allow entities at end hosts to communicate
  - TCP (transmission control protocol): reliable connection-oriented
  - UDP (user datagram protocol): unreliable connectionless

  Internet layer (IP)
  - A packet switching network based on connectionless communication. Hosts send packets into the network and then the packets travel independently to their destinations.
  - Format conversion: for different networks.
  - Packet format and protocol: IP

  Host to Network layer (Ethernet, FDDI, X.25)
  - Undefined, rely on the existing technology - must be able to send IP packets over this layer.

Encapsulation & Decapsulation



Ethernet header exclude preamble and start frame of 8 bytes

- TCP/IP model vs. OSI model:
  - similarity:
    - based on the concept of the stack of independent protocols.
    - Similar functionality
  - Differences:
    - The concepts of services/interfaces/protocols are clear in the OSI model, but not in the TCP/IP model.
    - The OSI model was devised before the protocols were invented, it misses some important issues. The TCP/IP model was devised after the protocols were designed, so the model may not fit other protocols.
    - Difference in the network layer:
      - » TCP/IP: only connectionless (the IP protocol)
      - » OSI: connectionless and connection-oriented.

- The five-layer hybrid model:

  Application

  Transport

  Network

  Data Link

  Physical

- *? At what layer is security addressed*

Computer Communication: An example

- What will happen when I click on

  http://www.cs.fsu.edu/~sudhir/courses/2025fcnt4504

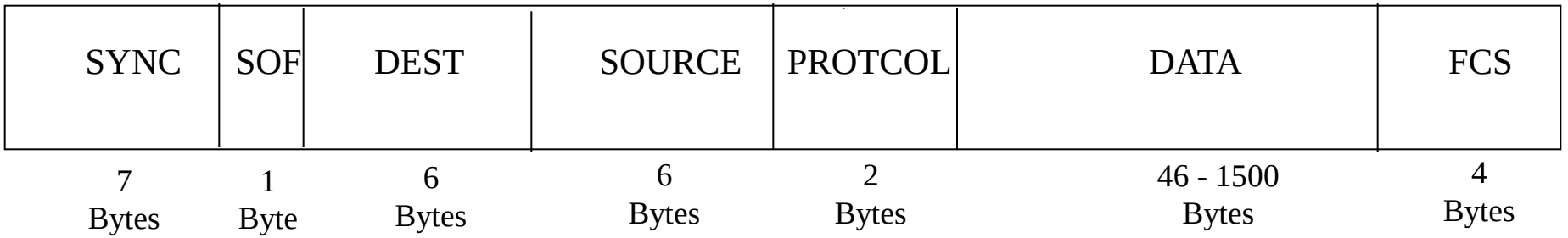
addressing, naming, routing

- Interlinked concepts
  - Addressing is often a low level identification of an object of interest
  - Naming is a high-level user friendly identification
  - Routing is determining a path to from one object to another
  - We will use address as a generic term
- What can be addressed?
  - A computer used for general computing - a host or end system
  - A communications processor or server - multi-homed host, gateway, router, intermediate system
  - A specific communications adaptor connected to a subnetwork
  - Software and associated data structures - ports, service access points, selectors

Addressing

- An object can be addressed in many ways
  - Social security number – 800-55-1212
  - Name – John Q Public
  - Designation – The Boss
  - Set of persons – NYPD
- Ultimately all addresses are bit strings representing numbers, characters, or just bits
- Properties of addresses
  - What is the context in which this address can be understood. Is there a unique object corresponding to this address
  - What happens if object is moved
  - Who what is the addressing authority
  - Multicast or broadcast designating many objects simultaneously

Logical Format of an Ethernet (DIX) Frame (original specification – not 802.3)



SYNC: synchronization bytes

SOF: synchronization byte indicating start of frame

DEST: destination link layer address (physical address)

SOURCE: source link layer address (physical address)

PROT. TYPE: encapsulated layer protocol identification

DATA: upper layer data (variable number of bytes in length)

FCS: frame check sequence for error detection

Why "logical format – what is physical format" ?

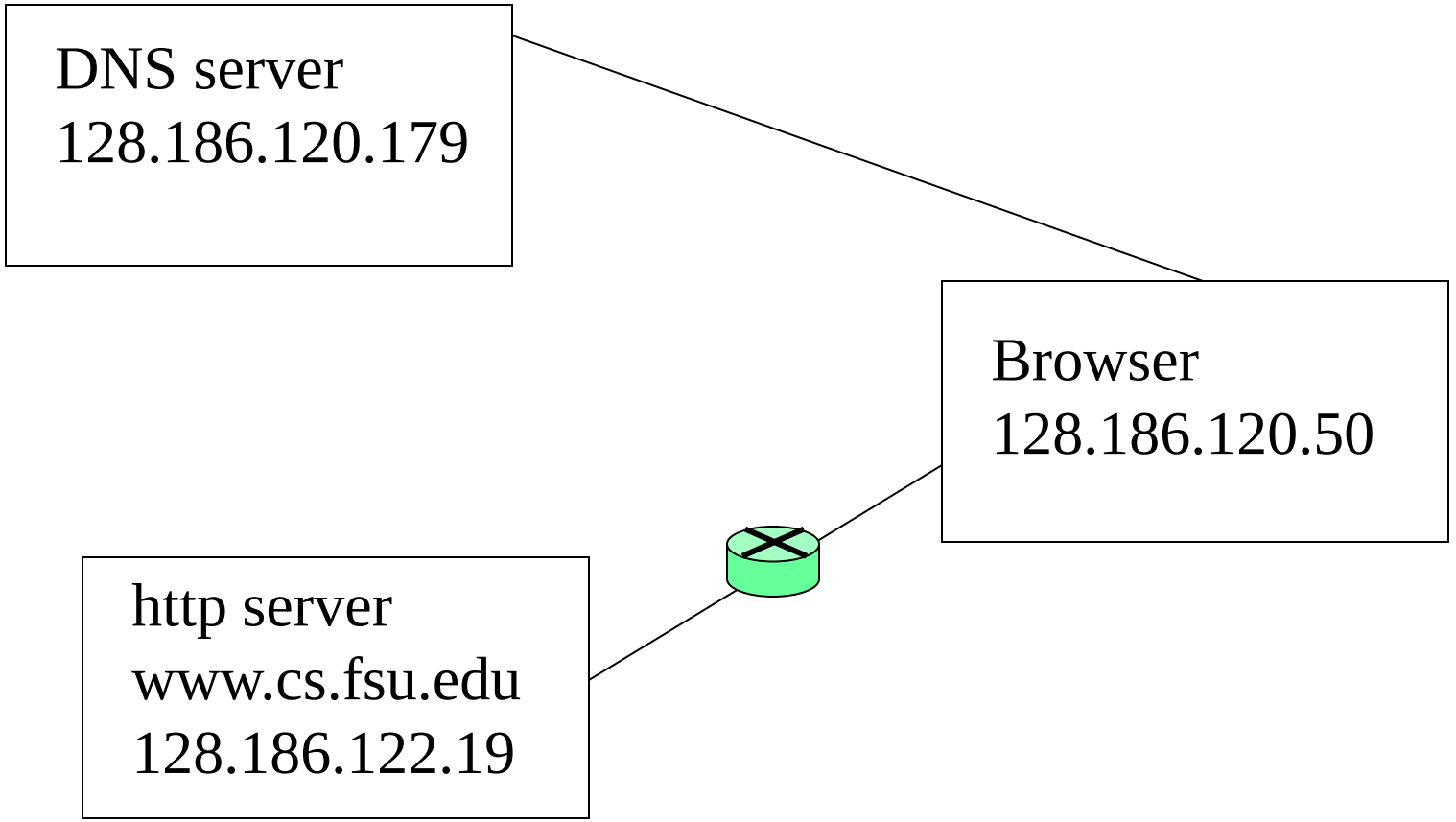
What is the physical LAN address vs logical IP address?

Browser needs to figure out what to do

– Three components in the URL:
  - Which machine? *www.cs.fsu.edu*
  - Which file? */~sudhir/courses/2025fcnt4504*
  - How do I get the file? Through the *HTTP* protocol.

– I need to talk to the http daemon (using *HTTP*) on *www.cs.fsu.edu* to get the file.

– Browser (on my machine with IP address, say *128.186.120.50*), needs to first find out what is the network address of *www.cs.fsu.edu*, the http server.
  - It does this using the DNS (**Domain Name System**) protocol to query the local DNS server.


DNS server
128.186.120.179
Browser
128.186.120.50
http server
www.cs.fsu.edu
128.186.122.19

Finding the network address of *www.cs.fsu.edu*

- Browser first needs to go to a domain name server (DNS) to find out the IP address of *www.cs.fsu.edu*

  host = gethostbyname (*www.cs.fsu.edu*)

- DNS server in our department:

  nu.cs.fsu.edu (*128.186.120.179*)

- How does the browser know the address of the DNS server?
  - One possibility: Hard-coded on machine at system (network) configuration. E.g. /etc/resolv.conf in a Linux system.
  - Or: Dynamically get the address upon request - DHCP
  - Or: Find it from the browser cache

Getting the address from the DNS server

- The Browser, using the DNS protocol sets up a dialogue with the DNS Server at *128.186.120.719/53* to get the desired address (note that 53 is the port number)
- The transport protocol UDP over IP is used to send datagram packets between the Browser and the DNS Server.
- Turns out that *128.186.120.179* (DNS server) and *128.186.120.50* (Browser) are on the same Ethernet LAN, so can send directly via Ethernet.
- *How to use Ethernet to communicate with DNS Server? Next slide!*

Using Ethernet to Send IP Packet

- Need to find out the Ethernet address (physical or hardware address) corresponding to the logical IP address *128.186.120.179*.
- Using the ARP protocol, send an ARP packet over the LAN:

  What is the address of *128.186.120.179*? Assume the result gives 8:00:20:7d:4f:49.
- The IP module now asks Ethernet to send an Ethernet frame with the IP packet to 08:00:20:7d:4f:49.
- Ethernet on *nu.cs.fsu.edu* (DNS server) receives a frame; turns out to have an IP packet destined for it, passes it to the IP module.

Using Ethernet to Send IP Packet (continued)

- IP module finds out that it has a UDP packet in the IP packet and passes it to the UDP module.
- UDP realizes that a process is listening to port 53, notifies the process.
- Process handles the message and replies to Browser host with a DNS reply message.
- DNS client dialogue involves several message exchanges
- The DNS server eventually sends a message back: *128.186.122.19* is the network address of *www.cs.fsu.edu*

Connecting to the http Server

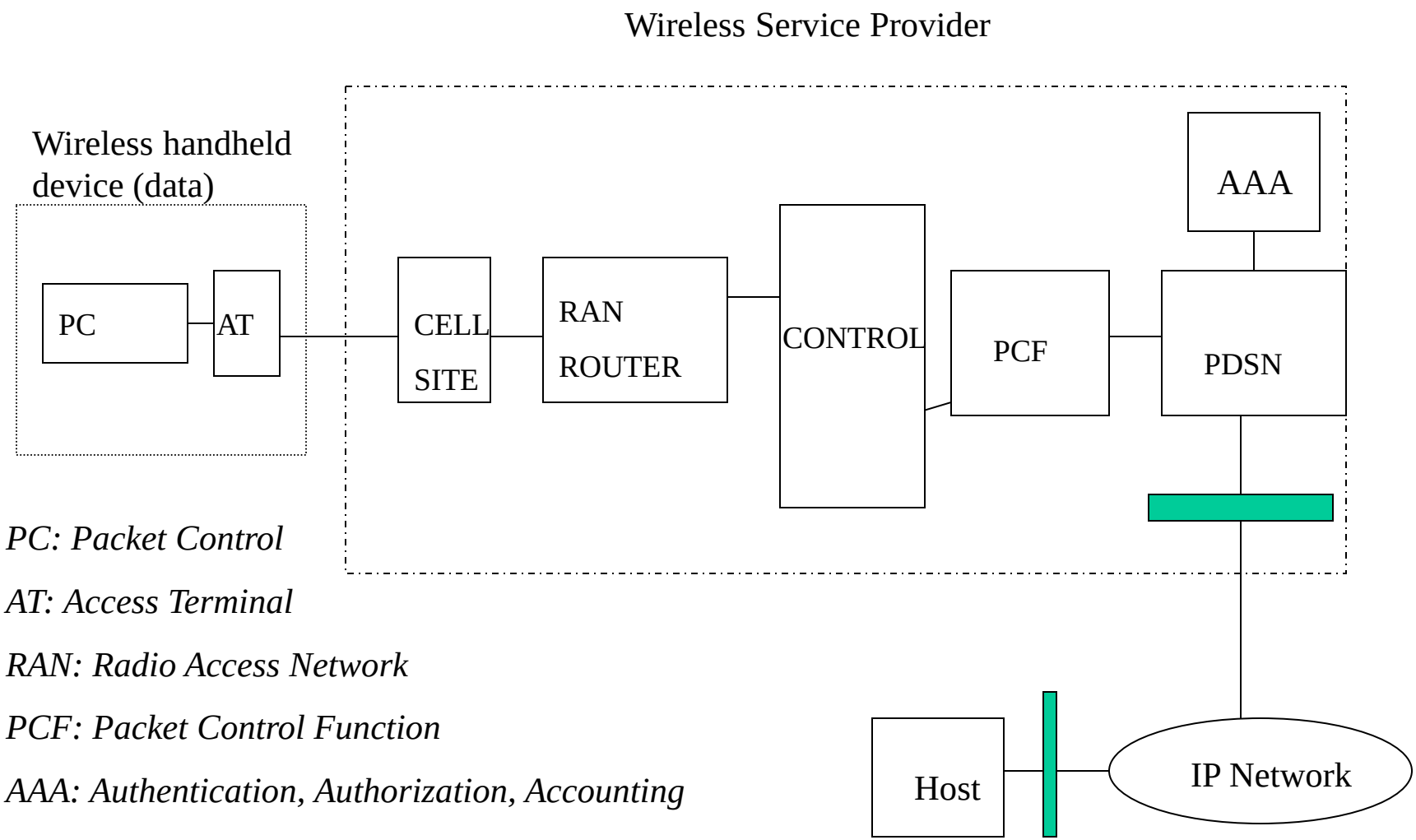
- Set up a TCP connection to the remote (*128.186.122.19) http* daemon.
  - Call TCP module to set up a connection from the Browser to 128.186.122.19/80
    - TCP module in turn calls IP module to send a datagram (IP packet) to 128.186.122.19
      - turns out that 128.186.122.19 (www.cs.fsu.edu) and the machine of the Browser are not directly connected.
      - First send to appropriate router (default gateway: 128.186.120.1) on the local network
      - Router forwards until on the same LAN as destination
      - ….
      - www.cs.fsu.edu receives a packet.
    - *? What is a daemon process*

- Talking to the *http* daemon on *www.cs.fsu.edu* using the *http* protocol.
  - Use TCP to send string:

    ` *get /~sudhir/courses/2020fcnt5505*
    - TCP entity calls IP to send a packet
    - …..
  - *www.cs.fsu.edu* responds with the content:

    /~sudhir/courses/2020fcnt5505
- Browser shows the file in the window.

Protocol Stack Wireless Example



PC: Packet Control

AT: Access Terminal

RAN: Radio Access Network

PCF: Packet Control Function

AAA: Authentication, Authorization, Accounting

PDSN: Packet Data Switching Node

Protocol Stacks