CIS 551 / TCOM 401

Computer and Network Security

Spring 2009

Lecture 7

Announcements

- First project: Due: TOMORROW at 11:59 p.m.
- http://www.cis.upenn.edu/~cis551/project1.html

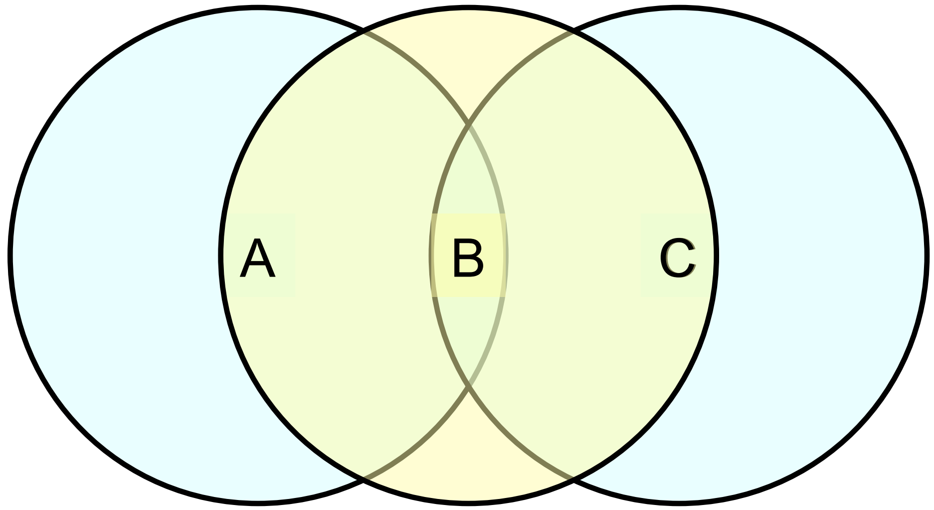
- Plan for Today:
  – Networks: 802.11 / IP / TCP

Wireless (802.11)

- Spread spectrum radio
  - 2.4GHz frequency band
- Bandwidth ranges 1, 2, 5.5, 11, 22, … Mbps

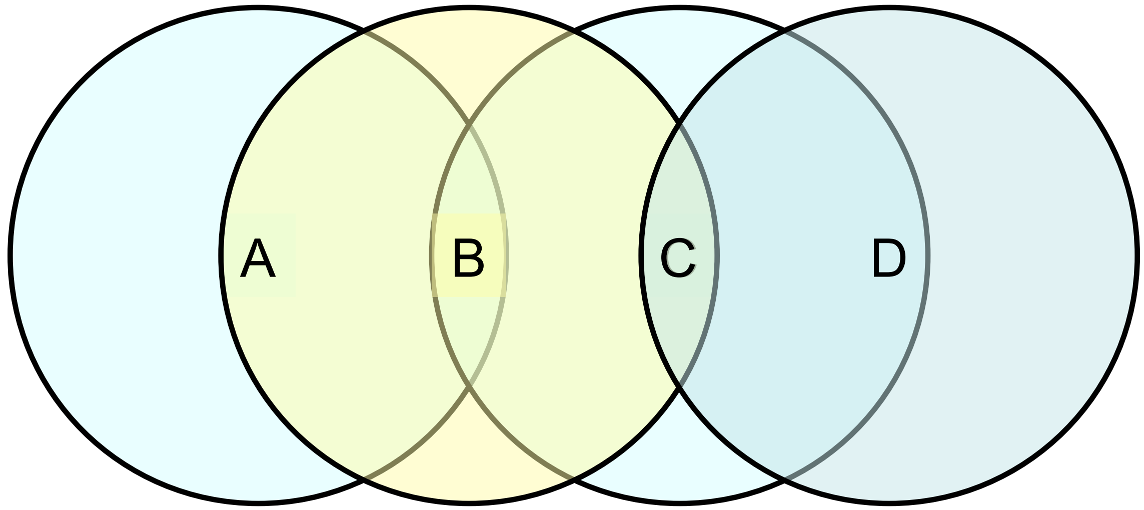
- Like Ethernet, 802.11 has shared medium
  - Need MAC (uses exponential backoff)
- Unlike Ethernet, in 802.11
  - No support for collision detection
  - Not all senders and receivers are directly connected

Hidden nodes



- A and C are *hidden* with respect to each other
  - Frames sent from A to B and C to B simultaneously may collide, but A and C can't detect the collision.

Exposed nodes

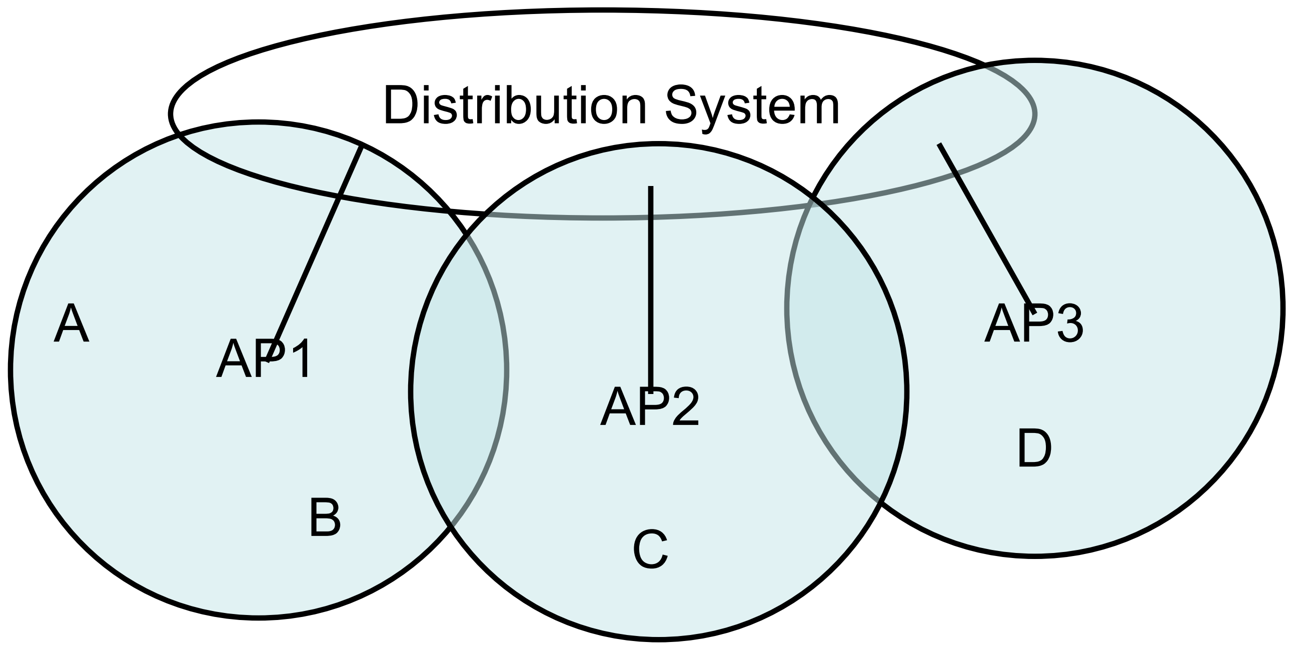


- B is exposed to C
  - Suppose B is sending to A
  - C should still be allowed to transmit to D
  - Even though C—B transmission would collide
  - (Note A to B transmission would cause collision)

Multiple Access Collision Avoidance

- Sender transmits Request To Send (RTS)
  - Includes length of data to be transmitted
  - Timeout leads to exponential backoff (like Ethernet)
- Receiver replies with Clear To Send (CTS)
  - Echoes the length field
- Receiver sends ACK of frame to sender
- Any node that sees CTS cannot transmit for durations specified by length
- Any node that sees RTS but not CTS is not close enough to the receiver to interfere
  - It's free to transmit

Wireless Access Points



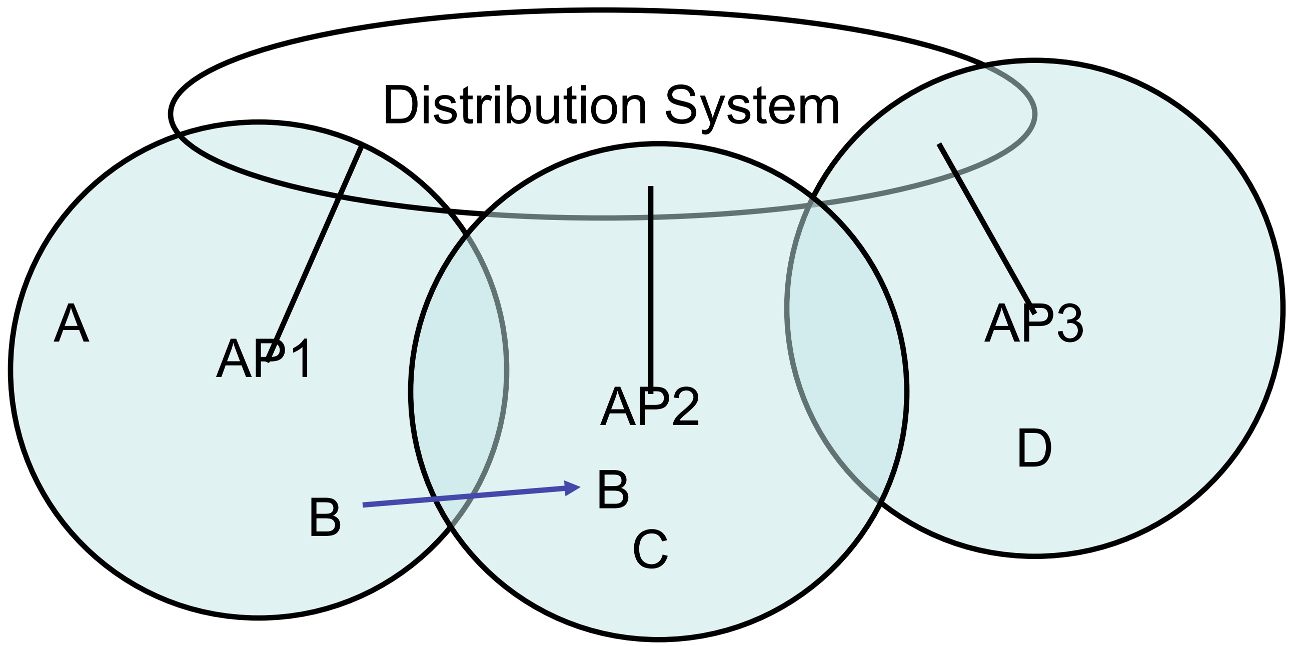
- Distribution System – wired network infrastructure
- Access points – stationary wireless device
- Roaming wireless

Selecting an Access Point

- *Active scanning*
  - Node sends a Probe frame
  - All AP’s within reach reply with a Probe Response frame
  - Node selects an AP and sends Association Request frame
  - AP replies with Association Response frame

- *Passive scanning*
  - AP periodically broadcasts Beacon frame
  - Node sends Association Request

Node Mobility



- B moves from AP1 to AP2
- B sends Probes, eventually prefers AP2 to AP1
- Sends Association Request

802.11 Security Issues

- Packet Sniffing is *worse*
  - No physical connection needed
  - Long range (6 blocks)
  - Current encryption standards (WEP, WEP2) not that good
- Denial of service
  - Association (and Disassociation) Requests are not authenticated

Wired Equivalent Privacy (WEP)

- Designed to provide same security standards as wired LANs (like Ethernet)
  - WEP uses 40 bit keys
  - WEP2 uses 128 bit keys
- Uses shared key authentication
  - Key is configured manually at the access point
  - Key is configured manually at the wireless device
- WEP frame transmission format:

$$802.11Hdr, IV, K_{S+IV}\{DATA, ICV\}$$

  - S = shared key
  - IV = 24 bit "initialization vector"
  - ICV = "integrity checksum" uses the CRC checksum algorithm
  - Encryption algorithm is RC4

Problem with WEP

- RC4 generates a keystream
  - Shared key S plus IV generates a long sequence of pseudorandom bytes RC4(IV,S)
  - Encryption is: C = P $\oplus$ RC4(IV,S) $\oplus$ = "xor"

- IV's are public -- so it's easy to detect their reuse
- Problem: if IV ever repeats, then we have
  - C1 = P1 $\oplus$ RC4(IV,S)
  - C2 = P2 $\oplus$ RC4(IV,S)
  - So C1 $\oplus$ C2 = P1 $\oplus$ P2
  - Statistical analysis or known plaintext can disentangle P1 and P2

Finding IV Collisions

- How IV is picked is not specified in the standard:
  - Standard "recommends" (but does not require) that IV be changed for every packet
  - Some vendors initialize to 0 on reset and then increment
  - Some vendors generate IV randomly per packet

- Very active links send ~1000 packets/sec
  - Exhaust 24 bit keyspace in < 1/2 day

- If IV is chosen randomly, probability is > 50% that there will be a collision after only 4823 packets

Other WEP problems

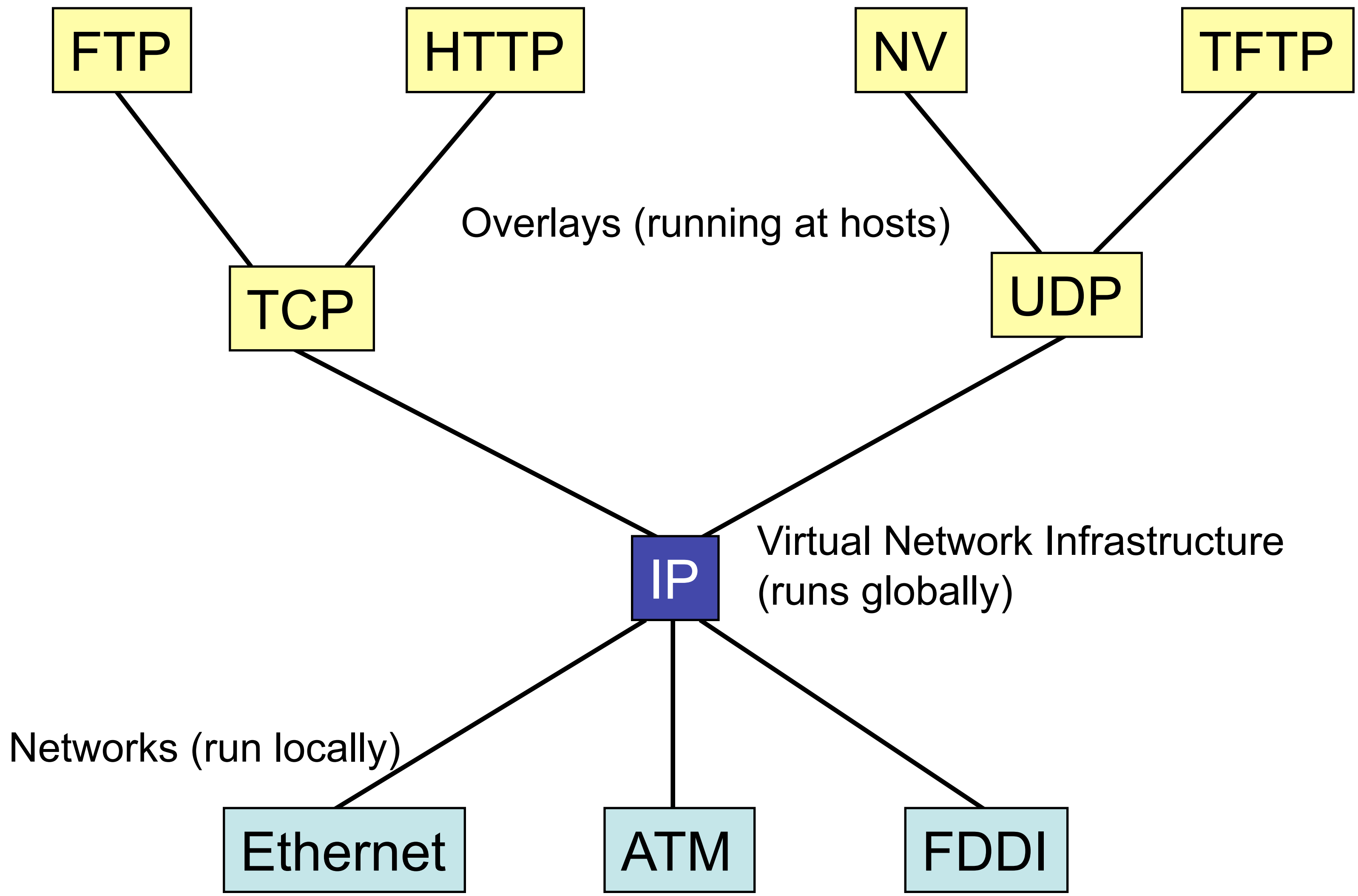
- Replay attacks
  - Standard requires the protocol to be stateless
  - Expensive to rule out replay attacks. (The sender and receiver can't keep track of expected sequence numbers)

- Integrity violations
  - Attacker can inject or corrupt WEP encrypted packets
  - CRC (Cyclic Redundancy Check) is an error detection code commonly used in internet protocols
  - CRC is good at detecting random errors (introduced by environmental noise)
  - But, CRC is not a hash function -- it is easy to find collisions
  - Attacker can arbitrarily pass off bogus WEP packets as legitimate ones

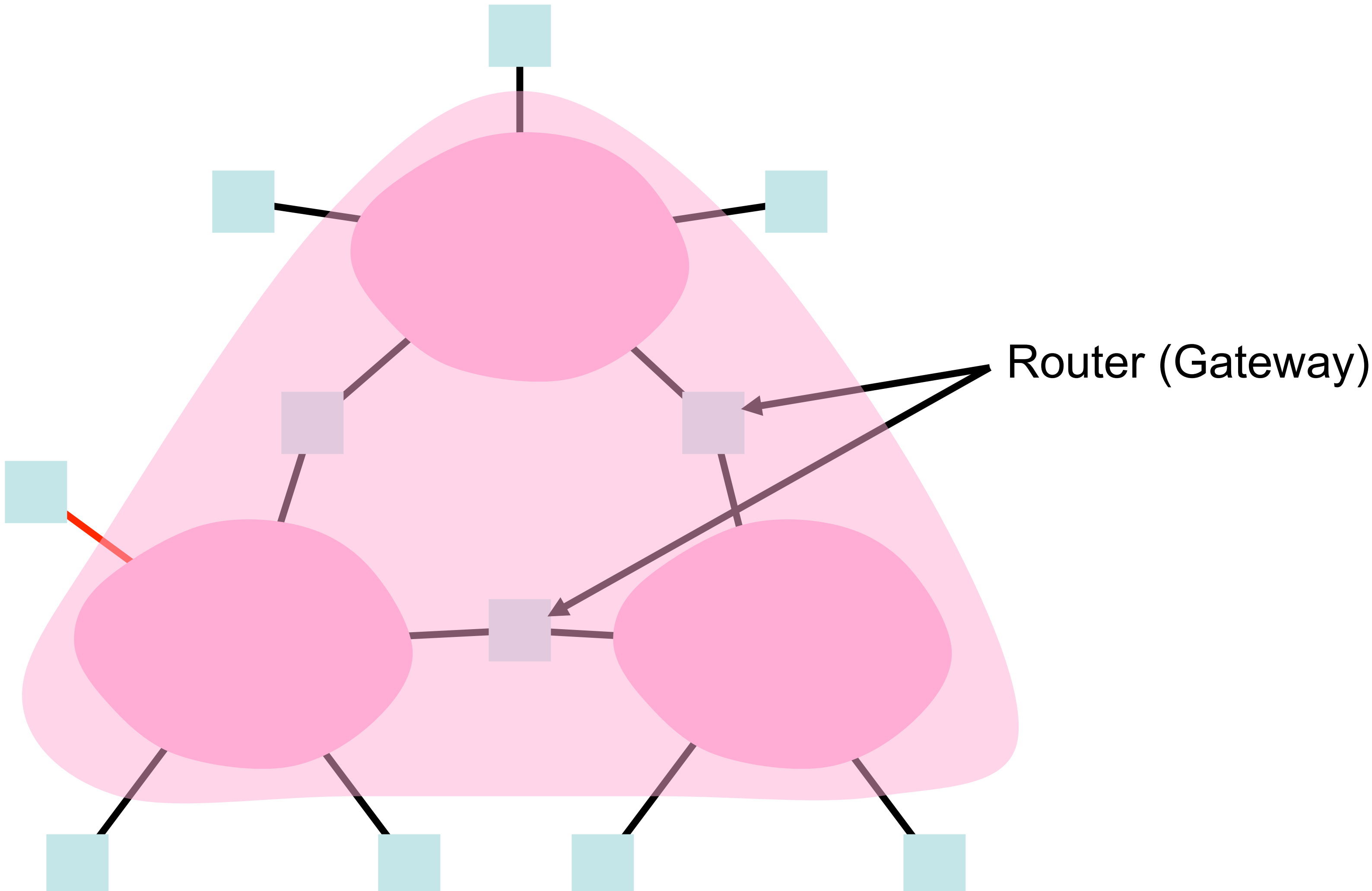
Newer 802.11 Standards

- WPA "Wi-Fi Protected Access"
  - Introduced in 801.11i
  - Uses much stronger cryptography (AES)
- EAP "Extensible Authentication Protocol"
  - Negotiates an authentication mechanism
- We will talk about such cryptographic protocols in much more detail in a few weeks.

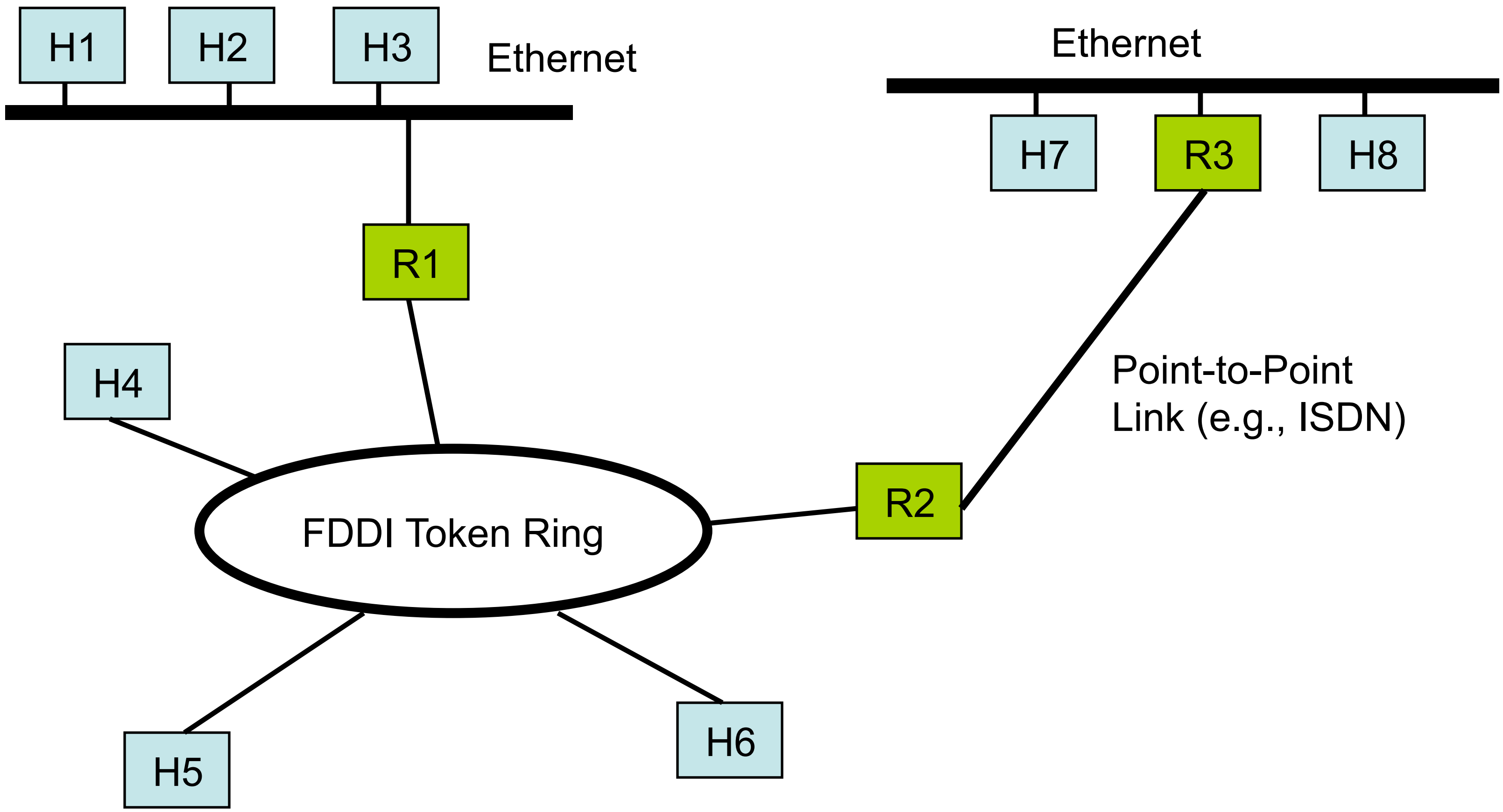
Internet Protocol Interoperability



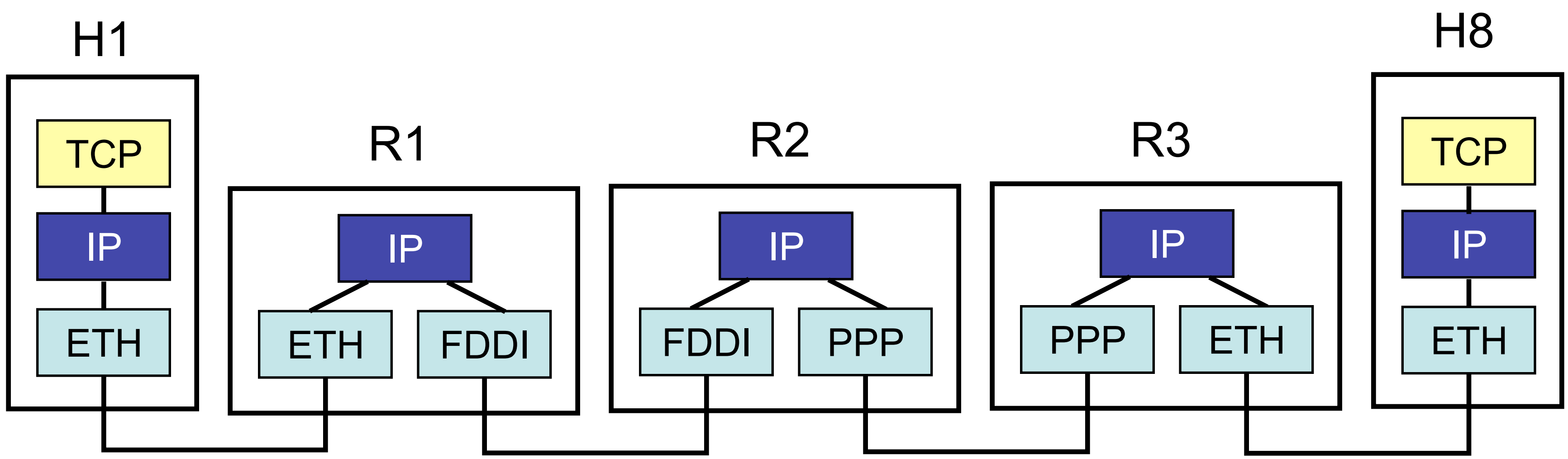
Internetworks



Internetworks



IP Encapsulation

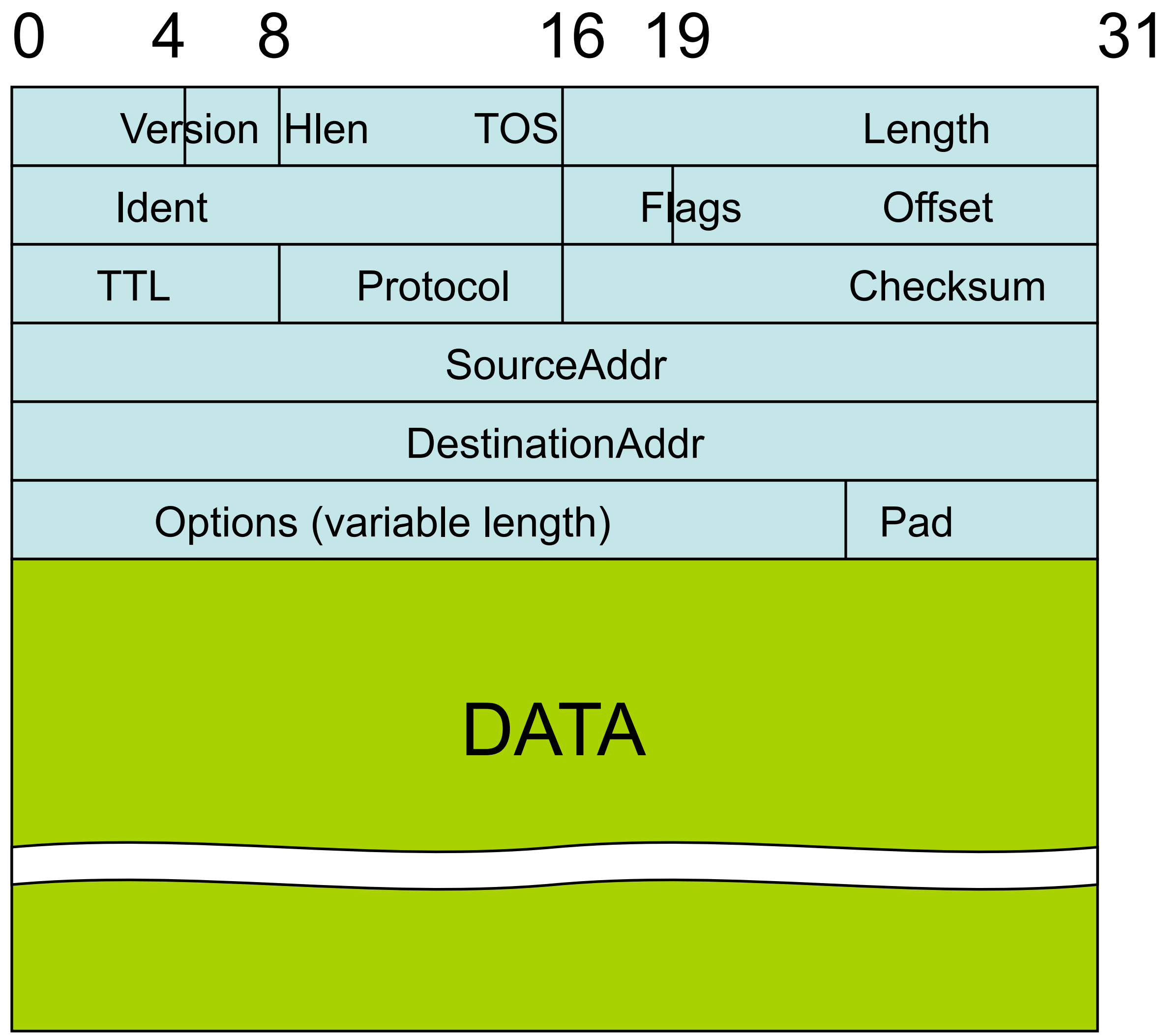


Example of protocol layers used to transmit from H1 to H8 in network shown on previous slide.

IP Service Model

- Choose minimal service model
  - All nets can implement
  - "Tin cans and a string" extremum
- Features:
  - Best-effort datagram delivery
  - Reliability, etc. as *overlays* (as in TCP/IP)
  - Packet format standardized

IPv4 Packet Format



Fields of IPv4 Header

- Version
  - Version of IP, example header is IPv4
  - First field so easy to implement case statement
- Hlen
  - Header length, in 32-bit *words*
- TOS
  - Type of Service (rarely used)
  - Priorities, delay, throughput, reliability
- Length
  - Length of datagram, in *bytes*
  - 16 bits, hence max. of 65,536 bytes
- Fields for *fragmentation* and *reassembly*
  - Identifier
  - Flags
  - Offset

Header fields, continued

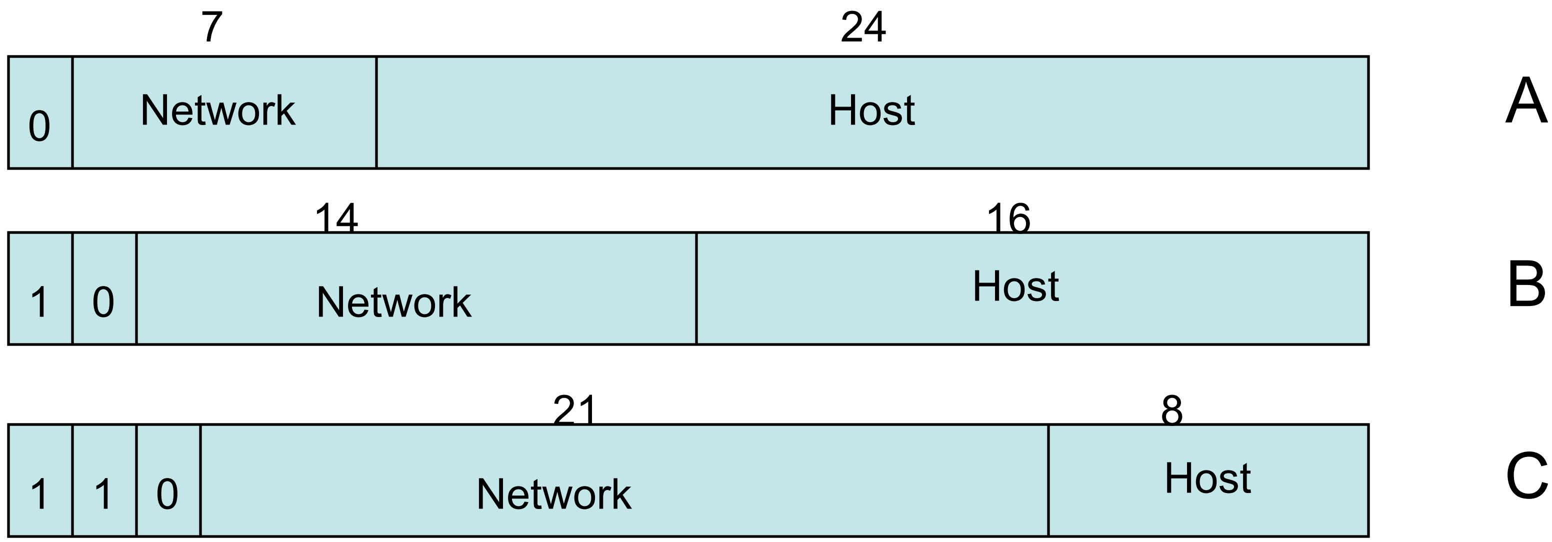
- TTL
  - Time to live (in reality, hop count)
  - 64 is the current default (128 also used)
- Protocol
  - e.g., TCP (6), UDP(17), etc.
- Checksum
  - Checksum of header (not CRC)
  - If header fails checksum, discard the whole packet
- SourceAddr, DestinationAddr
  - 32 bit IP addresses - global, IP-defined
- Options
  - length can be computed using Hlen

IP Datagram Delivery

- Every IP packet (datagram) contains the destination IP address
- The network part of the address uniquely identifies a single network that is part of the larger Internet.
- All hosts and routers that share the same network part of their address are connected to the same physical network.
- Routers can exchange packets on any network they're attached to.

IP addresses

- Hierarchical, not flat as in Ethernet



- Written as four decimal numbers separated by dots: 158.130.14.2

Network Classes

Class	*# of nets*	*# of hosts per net*
A	126	~16 million
B	8192	65534
C	~2 million	254

IP Forwarding algorithm

- If (Network # dest == Network # interface) then deliver to destination over interface
- else if (Network # dest in forwarding table) deliver packet to NextHop router
- else deliver packet to default router

- Forwarding tables
  - Contain (Network #, NextHop) pairs
  - Additional information
  - Built by routing protocol that learns the network topology, adapts to changes

Subnetting

- Problem: IP addressing scheme leads to fragmentation
  - A class B network with only 300 machines on it wastes > 65,000 addresses
  - Need a way to divide up a single network address space into multiple smaller subnetworks.

- Idea: One IP network number allocated to several physical networks.
  - The multiple physical networks are called *subnets*
  - Should be close together (why?)
  - Useful when a large company (or university!) has many physical networks.

Subnet Numbers

- Solution: *Subnetting*
  - All nodes are configured with *subnet mask*
  - Allows definition of a *subnet number*
    - All hosts on a physical subnetwork share the same *subnet number*

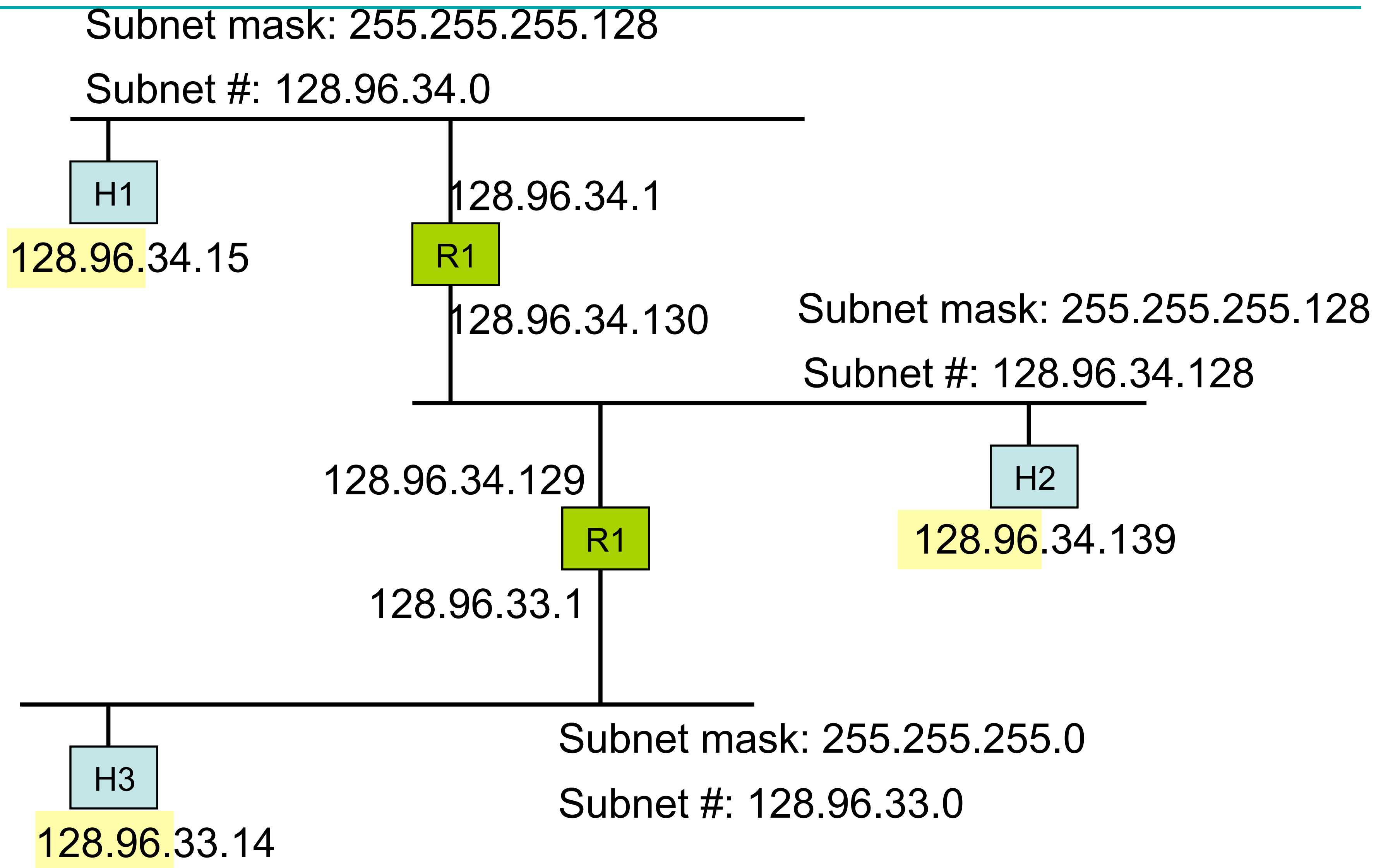
Subnet Mask (255.255.255.0)

111111111111111111111111	00000000

Subnetted Address:

Network number	Subnet ID	Host ID

Example of Subnetting



Subnets, continued

- Mask is bitwise-ANDed with address
- This is done at routers
- Router tables in this model:
  - <Subnet #, Subnet Mask, NextHop>
- Subnetting allows a set of physical networks to look like a single logical network from elsewhere

Forwarding Algorithm

```
D = destination IP address
for each forwarding table entry
(SubnetNumber, SubnetMask, NextHop)
    D1 = SubnetMask & D
    if  D1 = SubnetNumber
        if NextHop is an interface
            deliver datagram directly to destination
        else
            deliver datagram to NextHop (router)

Deliver datagram to default router (if above fails)
```

ARP - Address Resolution Protocol

- Problem:
  - Need mapping between IP and link layer addresses.

- Solution: ARP
  - Every host maintains IP–Link layer mapping table (cache)
  - Timeout associated with cached info (15 min.)
- Sender
  - Broadcasts "Who is IP addr X?"
  - Broadcast message includes sender's IP & Link Layer address
- Receivers
  - Any host with sender in cache "refreshes" time-out
  - Host with IP address X replies "IP X is Link Layer Y"
  - Target host adds sender (if not already in cache)

ICMP: Internet Control Message Protocol

- Collection of error & control messages
- Sent back to the source when Router or Host cannot process packet correctly
- Error Examples:
  - Destination host unreachable
  - Reassembly process failed
  - TTL reached 0
  - IP Header Checksum failed
- Control Example:
  - Redirect – tells source about a better route

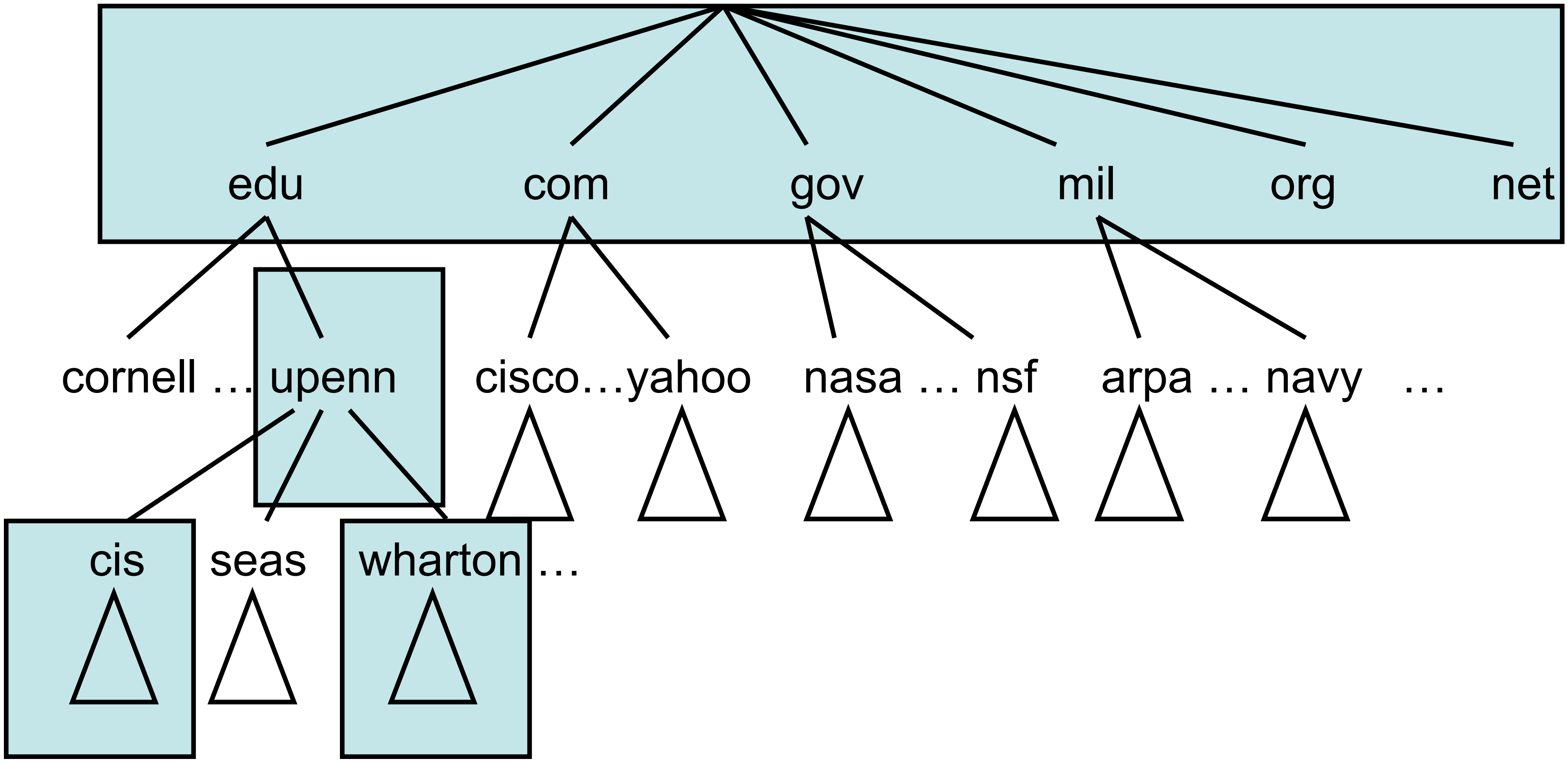
Domain Name System

- System for mapping mnemonic names for computers into IP addresses.

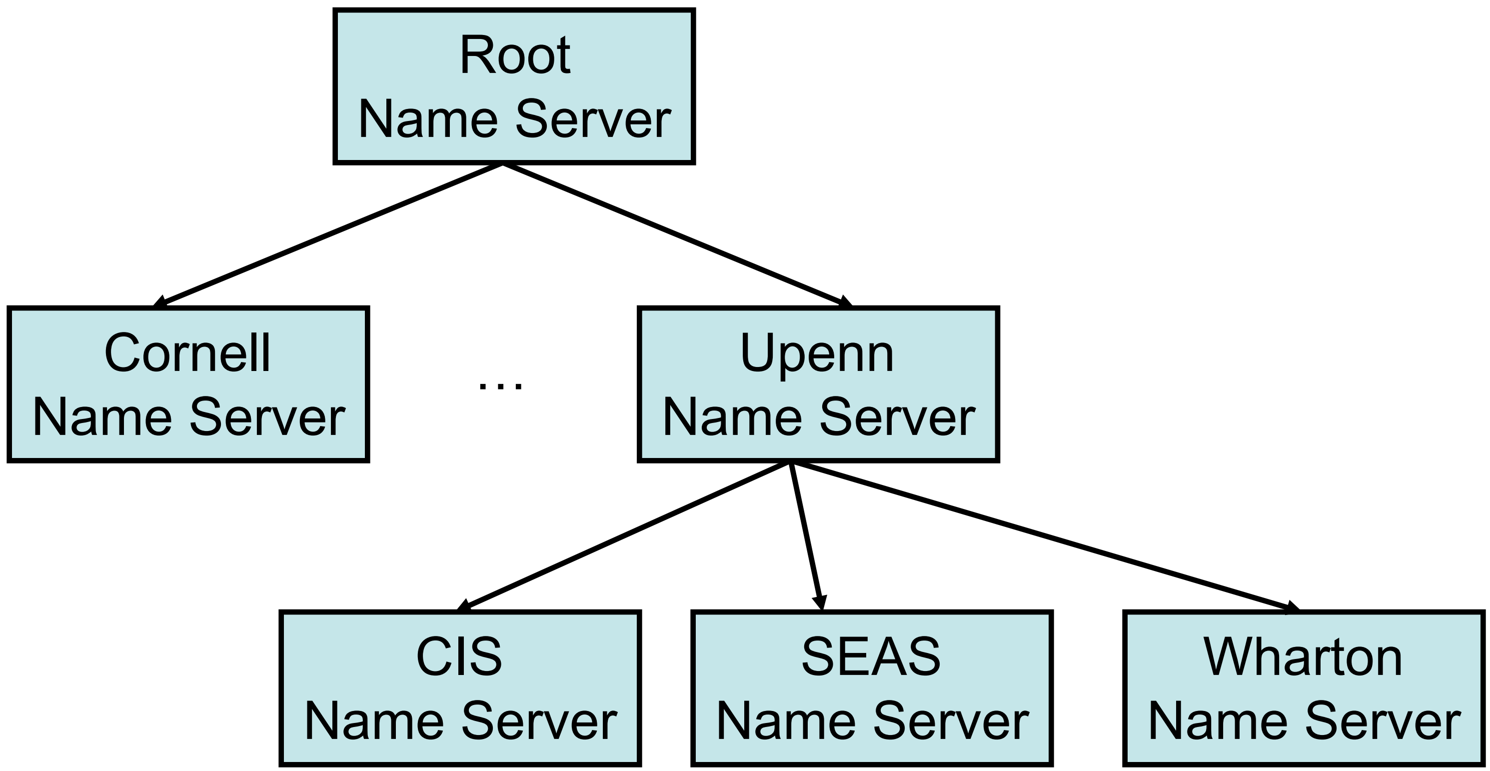
  zeta.cis.upenn.edu ⟶ 158.130.12.244

- Domain Hierarchy
- Name Servers
  – 13 Root servers map top-level domains such as ".com" or ".net"
  – (Why 13? Early UDP protocol supported only 512 bytes…)
- Name Resolution
  – Protocol for looking up hierarchical domain names to determine the IP address
  – Protocol runs on UDP port 53

Domain Name Hierarchy



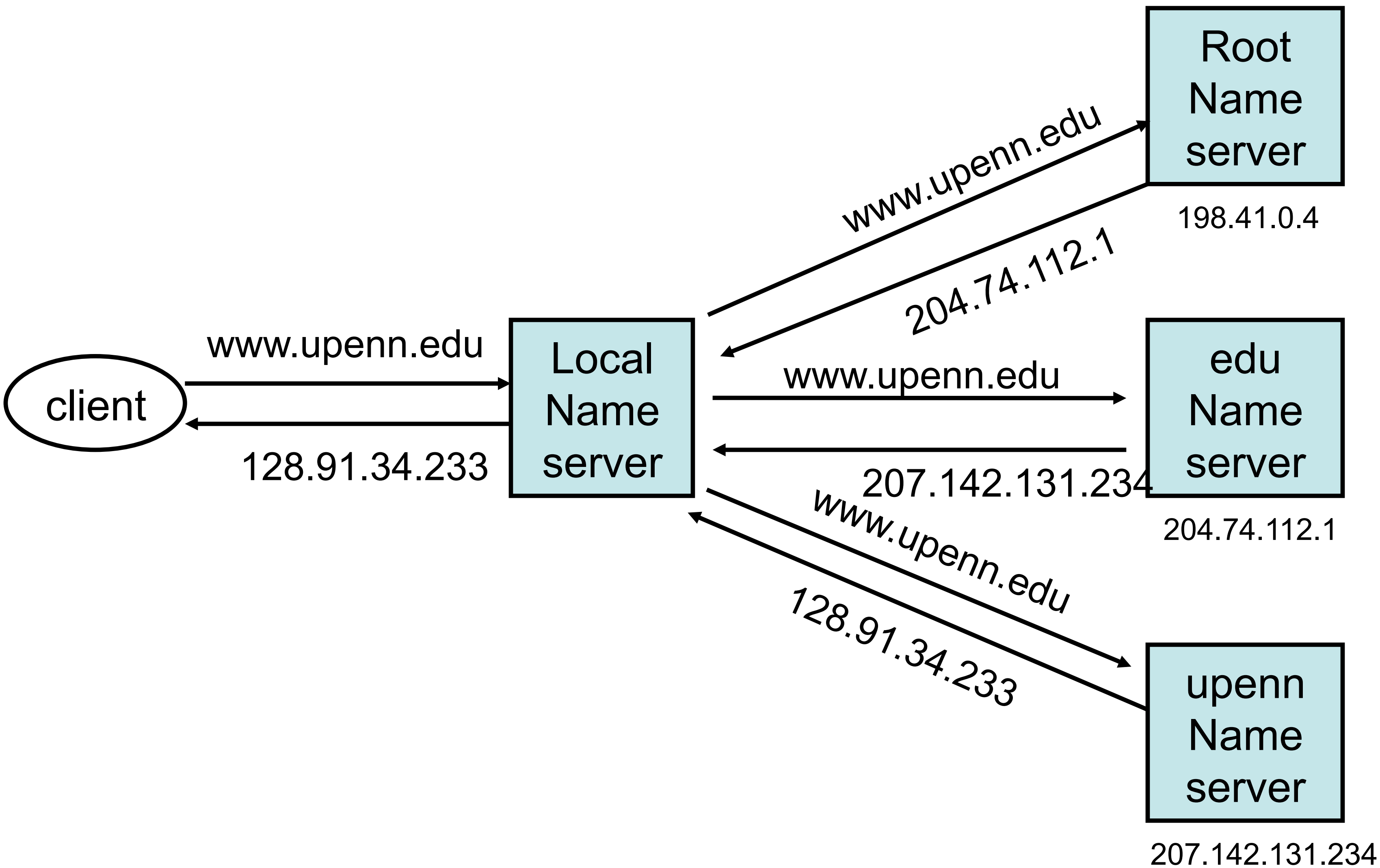
Hierarchy of Name Servers



Records on Name Servers

- < Name, Type, Class, TTL, RDLength, RDATA >
- Name of the node
- Types:
  - A Host to address mappings
  - NS Name server address mappings
  - CNAME Aliases
  - MX Mail exchange server mappings
  - … others
- Class IN for IP addresses

Name resolution



DNS Vulnerabilities

- See *"Corrupted DNS Resolution Paths: The rise of a malicious resolution authority"* by Dagon et al.
- Rogue DNS Servers
  - Compromised DNS servers that answer incorrectly
- DNS Cache Poisoning
  - Request: `subdomain.example.com IN A`
  - Reply:

```
Answer:
(no response)

Authority section:
example.com. 3600 IN ns.wikipedia.org.

Additional section:
ns.wikipedia.org IN A w.x.y.z
```

Reflected denial of service

- ICMP message with an "echo request" is called 'ping'

- Broadcast a ping request
  – For sender's address put target's address
  – All hosts reply to ping, flooding the target with responses

- Hard to trace
- Hard to prevent
  – Turn off ping? (Makes legitimate use impossible)
  – Limit with network configuration by restricting scope of broadcast messages

- Sometimes called a "smurf attack"

(Distributed) Denial of Service

- Coordinate multiple subverted machines to attack
- Flood a server with bogus requests
  - TCP SYN packet flood
  - > 600,000 packets per second
- Detection & Assessment?
  - 12,800 attacks at 5000 hosts! (in 3 week period during 2001)
  - IP Spoofing (forged source IP address)
  - http://www.cs.ucsd.edu/users/savage/papers/UsenixSec01.pdf

- Feb. 6 2007: 6 of 13 root servers suffered DDoS attack
- Oct. 21 2002: 9 of 13 root servers were swamped
  - Prompted changes in the architecture

- Prevention?
  - Filtering?
  - Decentralized file storage?