CS 114: Network Security

Lecture 13 - Domain Name System

Prof. Daniel Votipka

Fall 2021

(some slides courtesy of Prof. Micah Sherr)

Tufts UNIVERSITY

Plan for today

- Administrivia
- Review worms, bots, an DoS
- Domain Name Service (DNS)
  - The protocol
  - Vulnerabilities
  - Mitigations — DNSSec

Administrivia

- Homework 1, part 2 due tonight at 11:59pm
  - If the only test case you're failing is the test_connect test case, send me a private Piazza message to grade manually

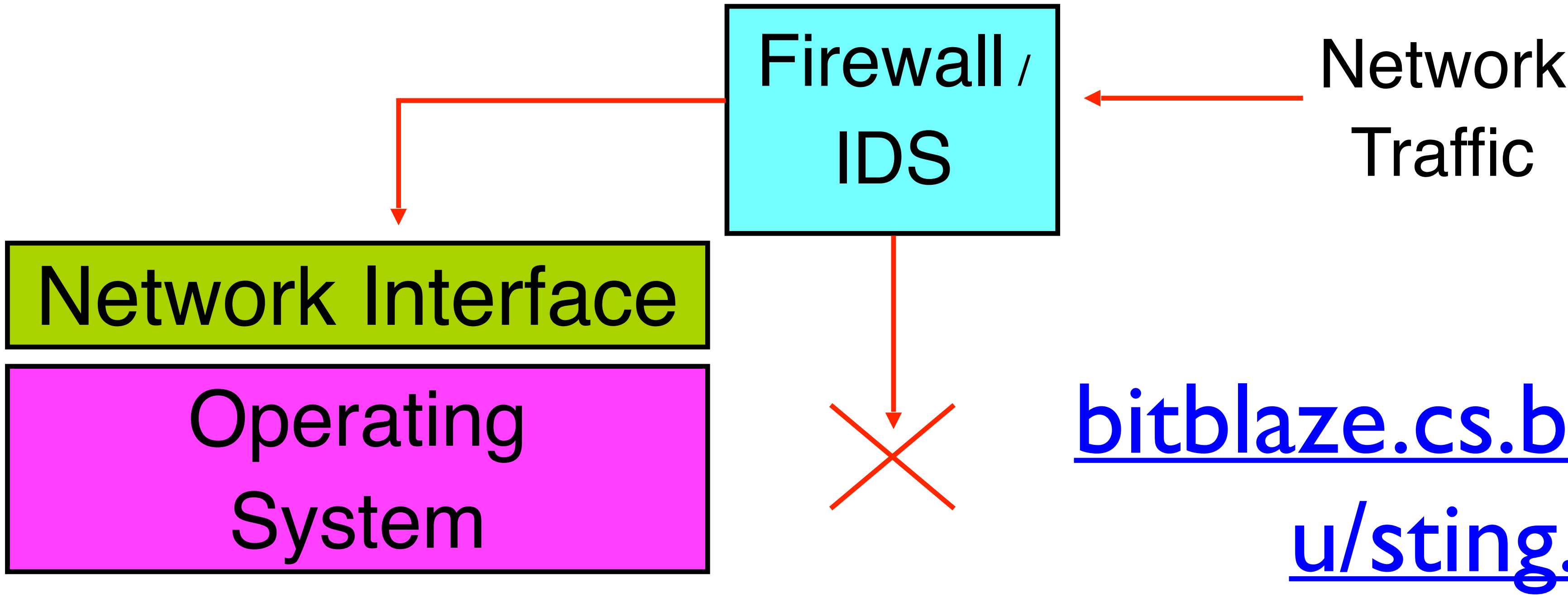
Worms, Bots, and DoS

Worms and infection

- **The effectiveness of a worm is determined by how good it is at identifying vulnerable machines**
- Multi-vector worms use lots of ways to infect: e.g., network, email, drive by downloads, etc.
- Example scanning strategies:
  - **Random IP:** select random IPs; wastes a lot of time scanning "dark" or unreachable addresses (e.g., Code Red)
  - **Signpost scanning:** use info on local host to find new targets (e.g., Morris)
  - **Local scanning:** biased randomness
  - **Permutation scanning:** "hitlist" based on shared pseudorandom sequence; when victim is already infected, infected node chooses new random position within sequence

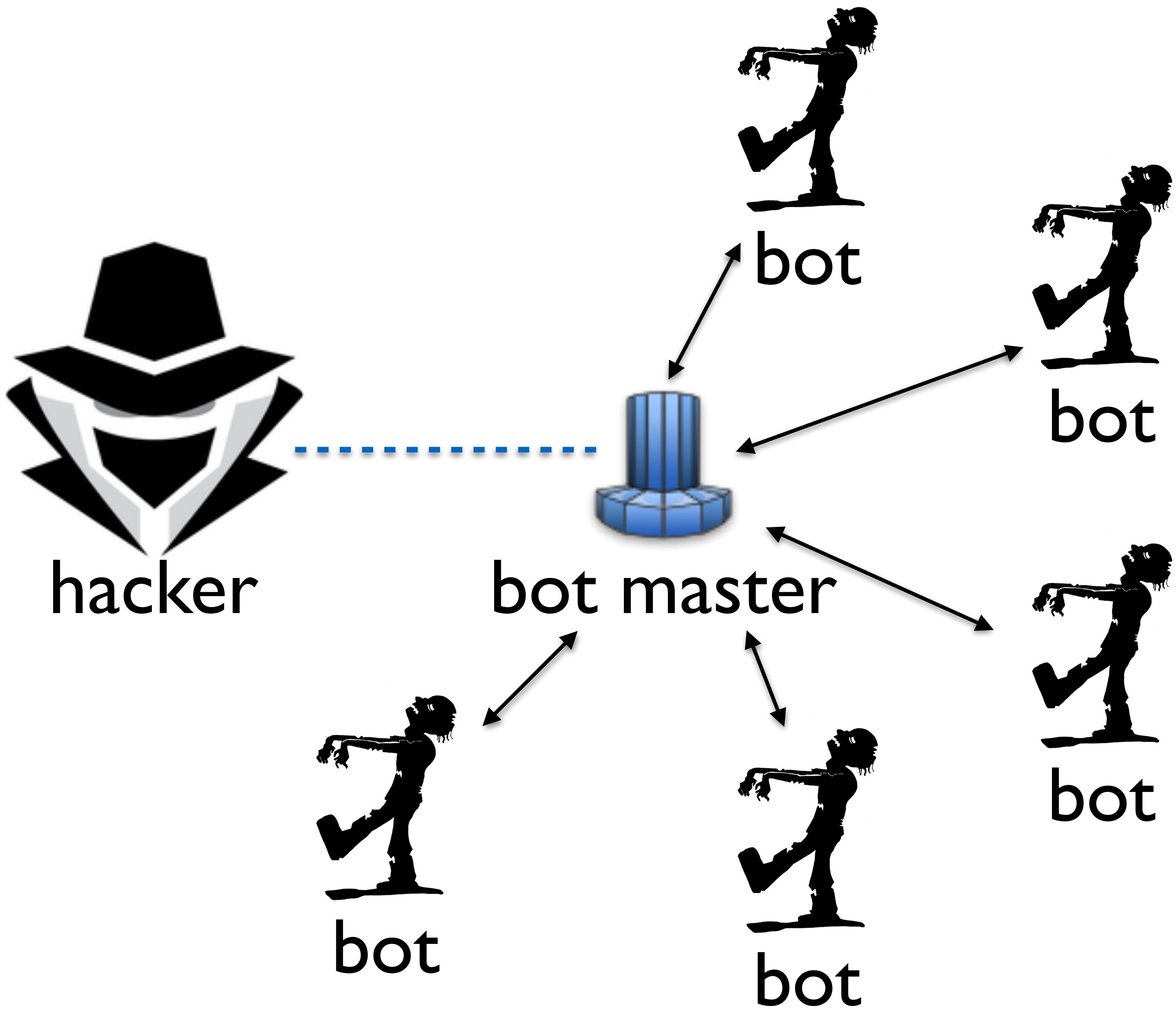
Worms: Defense Strategies

- (Auto) **patch** your systems: most large worm outbreaks have exploited known vulnerabilities (Stuxnet is an exception)
- **Heterogeneity**: use more than one vendor for your networks
- **IDS**: provides filtering for known vulnerabilities, such that they are protected immediately (analog to virus scanning)



bitblaze.cs.berkeley.edu/sting.html

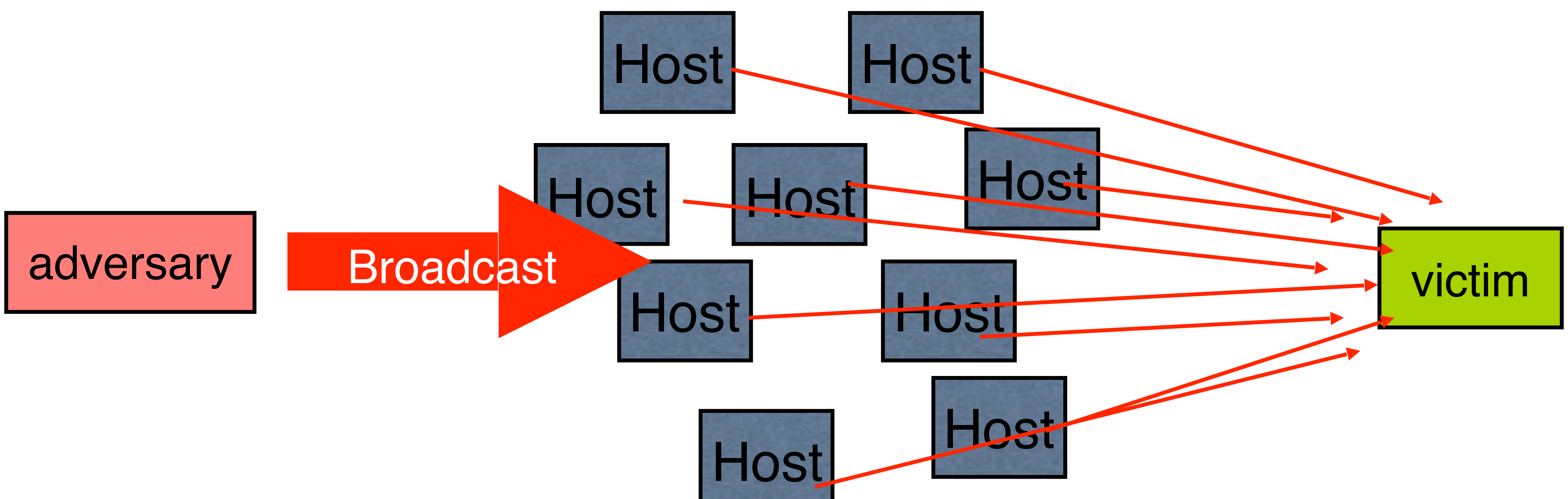
- **Filtering**: look for unnecessary or unusual communication patterns, then drop them on the floor


bot
bot
hacker
bot master
bot
bot
bot

Smurf Attacks

Example: SMURF Attacks

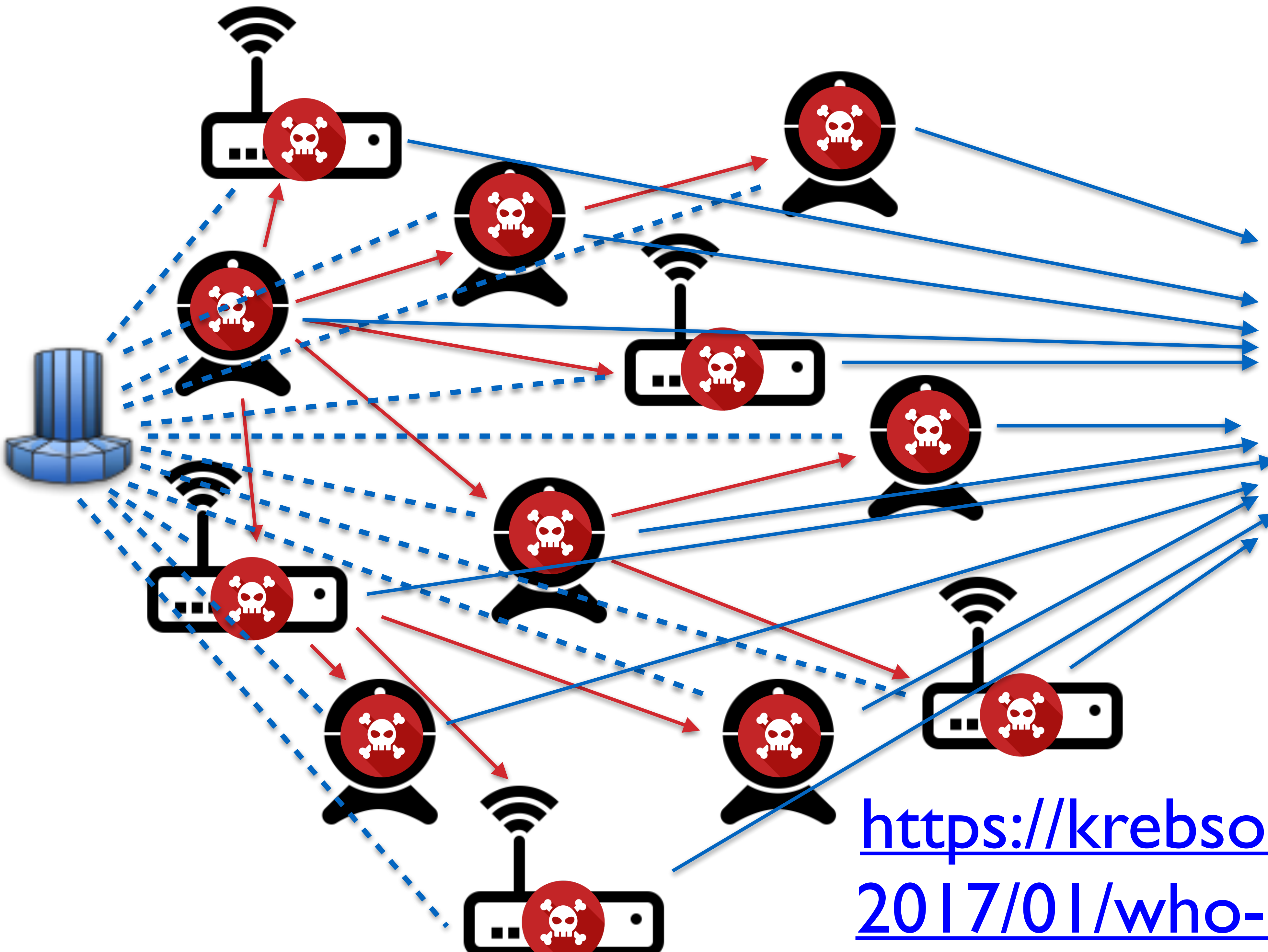
- Simple DoS attack:
  - Send a large number PING packets to a network's broadcast IP addresses (e.g., 192.168.27.254)
  - Set the source packet IP address to be your victim
  - All hosts will reflexively respond to the ping at your victim
  - … and it will be crushed under the load.
  - This is an **amplification attack** and a **reflection attack**



Distributed Denial-of-service (DDoS)

- DDoS: Network oriented attacks aimed at preventing access to network, host or service
  - Saturate the target's network with traffic
  - Consume all network resources (e.g., SYN flooding)
  - Overload a service with requests
    - Use "expensive" requests (e.g., "sign this data")
  - Can be extremely costly
- Result: service/host/network is unavailable
- Criminals sometimes use DDoS for racketeering
- Note: IP addresses of perpetrators are often hidden (spoofed)

Mirai Botnet



https://krebsonsecurity.com/2017/01/who-is-anna-senpai-the-mirai-worm-author/

Simple DDoS Mitigation

- **Ingress/Egress Filtering**: Helps spoofed sources, not much else
- Better Security
  - Limit availability of zombies (not feasible)
  - Prevent compromise and viruses (maybe in wonderful magic land where it rains chocolate and doughnuts)
- Quality of Service Guarantees (QoS)
  - Pre- or dynamically allocated bandwidth (e.g., diffserv)
  - Helps where such things are available
- Content replication
  - E.g,. CDS
  - Useful for static content

DDoS Reality

- None of the "protocol oriented" solutions have really seen any adoption
  - too many untrusting, ill-informed, mutually suspicious parties must play together
- Real Solution
  - Large ISPs police their ingress/egress points very carefully
  - Watch for DDoS attacks and filter appropriately
  - Develop products that coordinate view from many vantage points in the network to identify upswings in traffic

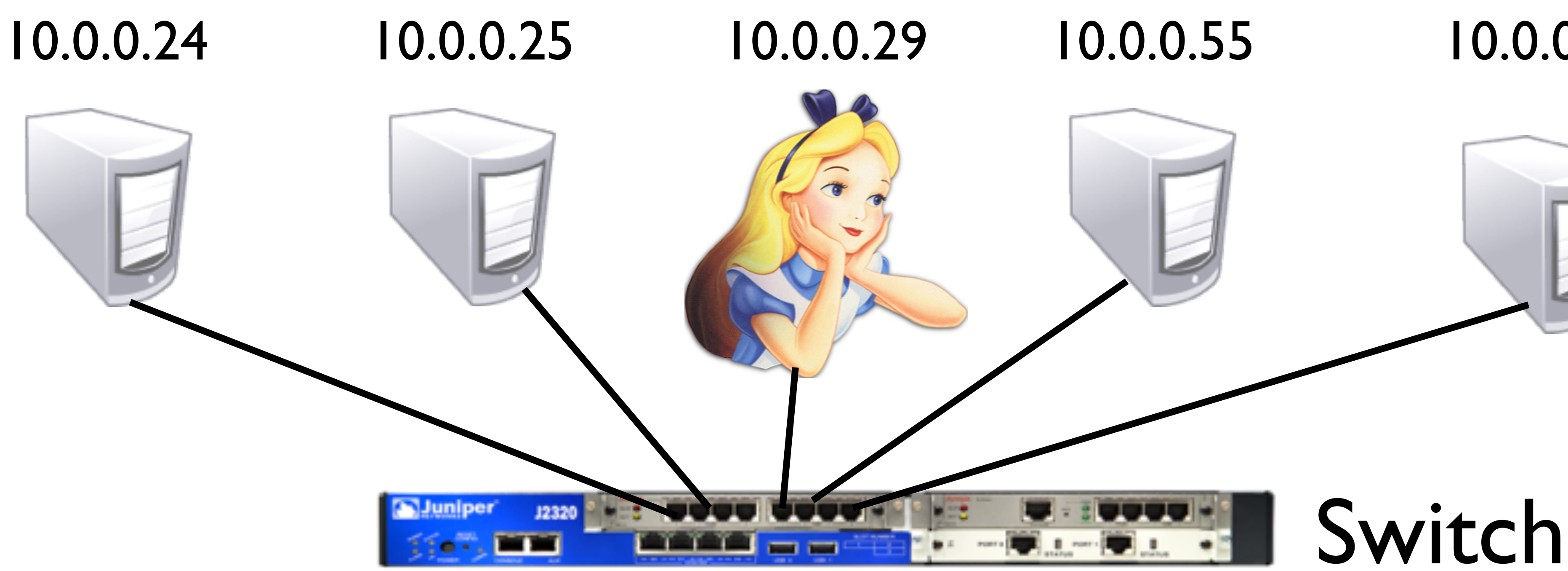
Plan for today

- Administrivia
- Review worms, bots, an DoS
- Domain Name Service (DNS)
  - The protocol
  - Vulnerabilities
  - Mitigations — DNSSec

A primer on routing

Routing Problem:
How do Alice's messages get to Bob?

Routing *within* the local network

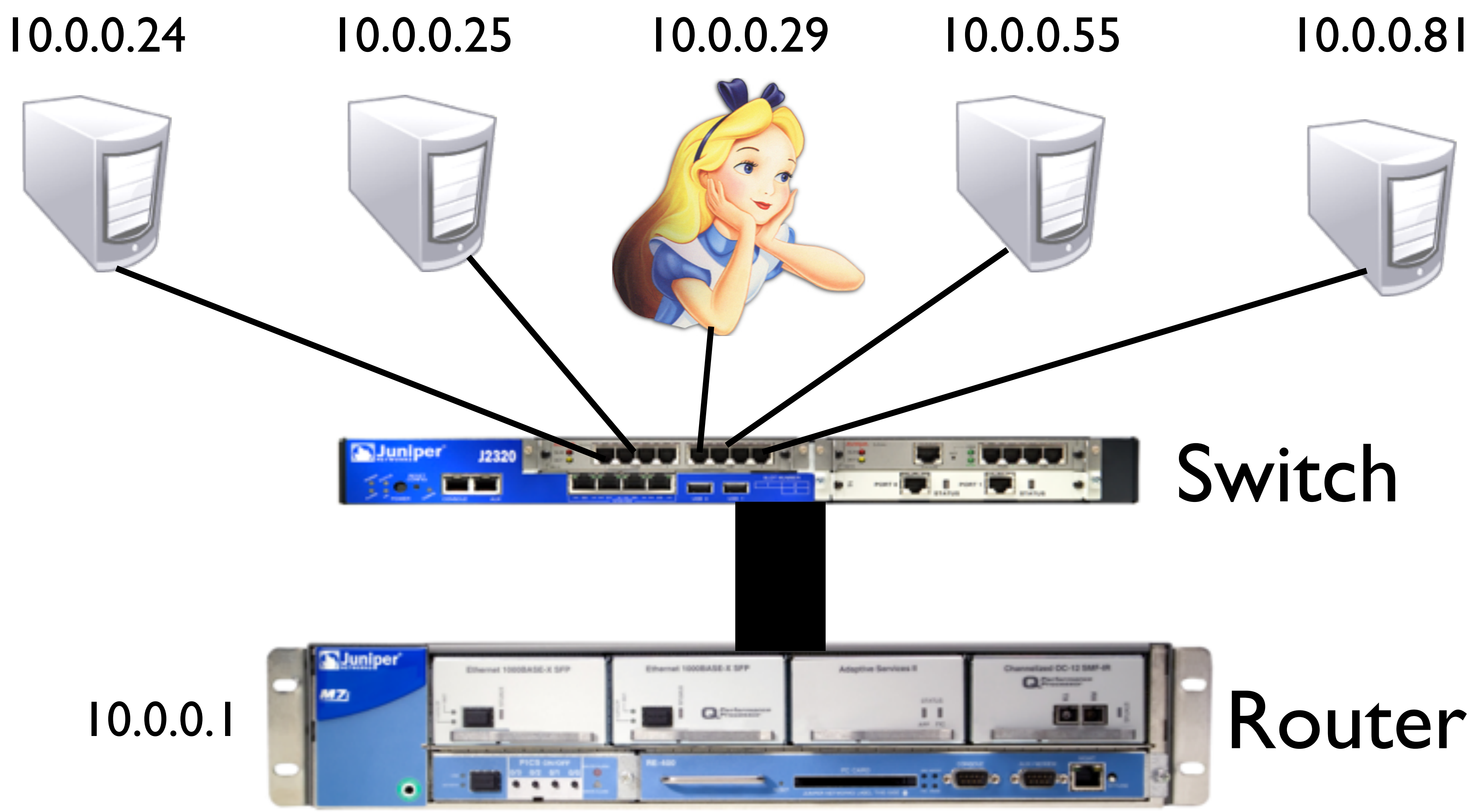


If Alice wants to communicate with node in local network, she uses ARP to discover the node's IP address and relies on the (layer 2) switch to correctly deliver the message.

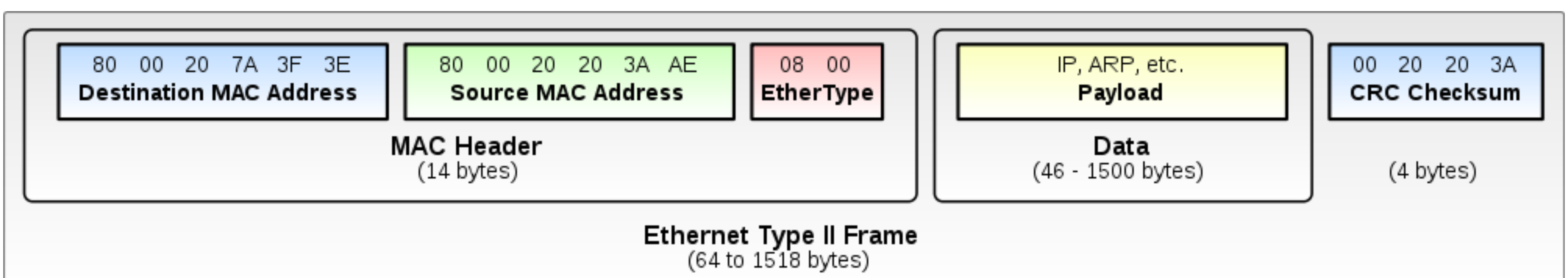
- Each host knows the network prefix of the local network
  - All nodes within the local network are reachable within 1 hop
  - **CIDR Notation**: BaseAddress/Prefix_Size
    - e.g., 10.0.0.0/24:
      - Network prefix is 10.0.0 (first 24 bits -- or 3 octets)
      - Number of possible addresses in network: 32-24 = 8 bits → 2^8 = 256 addresses

But what if Alice wants to route *outside* of her local network?

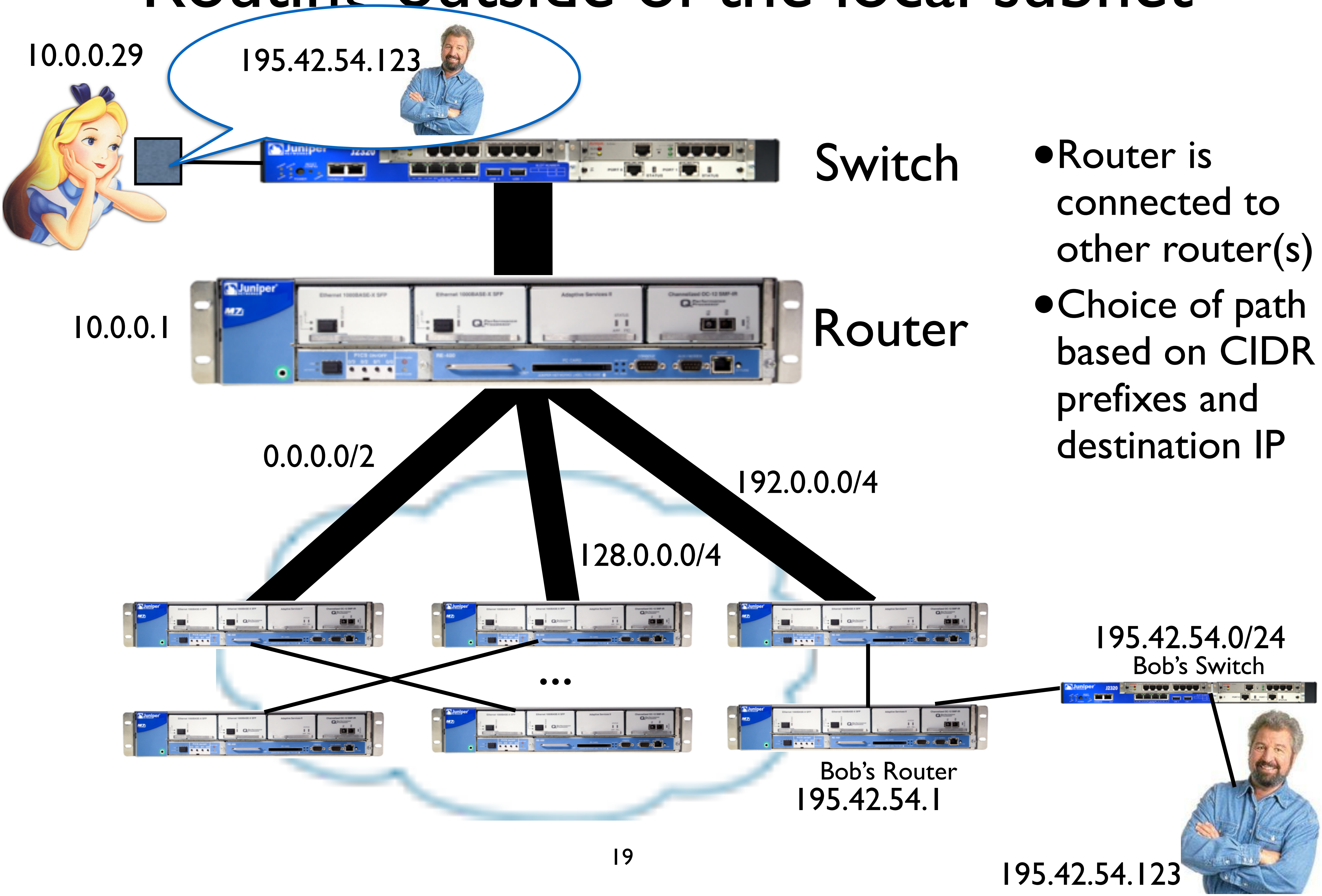
Routing outside of the local subnet



- Alice relays her message thru her subnet's *router*
  - Specifies Bob's IP address as destination IP in IP header
  - But specifies router's MAC address as destination in Ethernet frame
  - Switch relays Alice's message to router



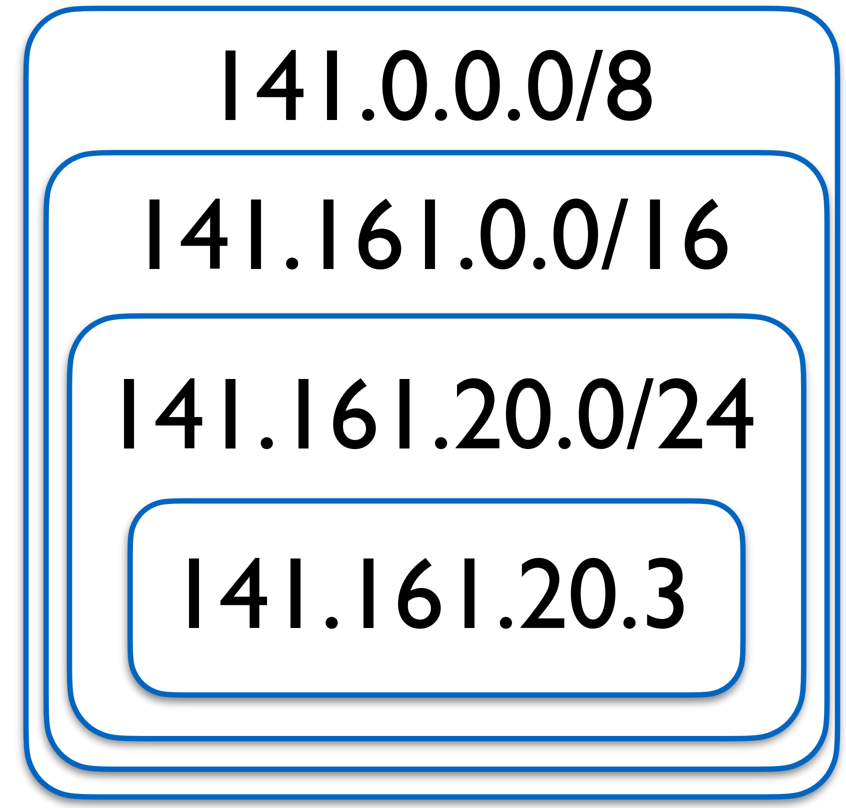
Routing outside of the local subnet



- Router is connected to other router(s)
- Choice of path based on CIDR prefixes and destination IP

But what if Alice doesn’t know Bob’s (bob.com) IP address?

Internet Hierarchies

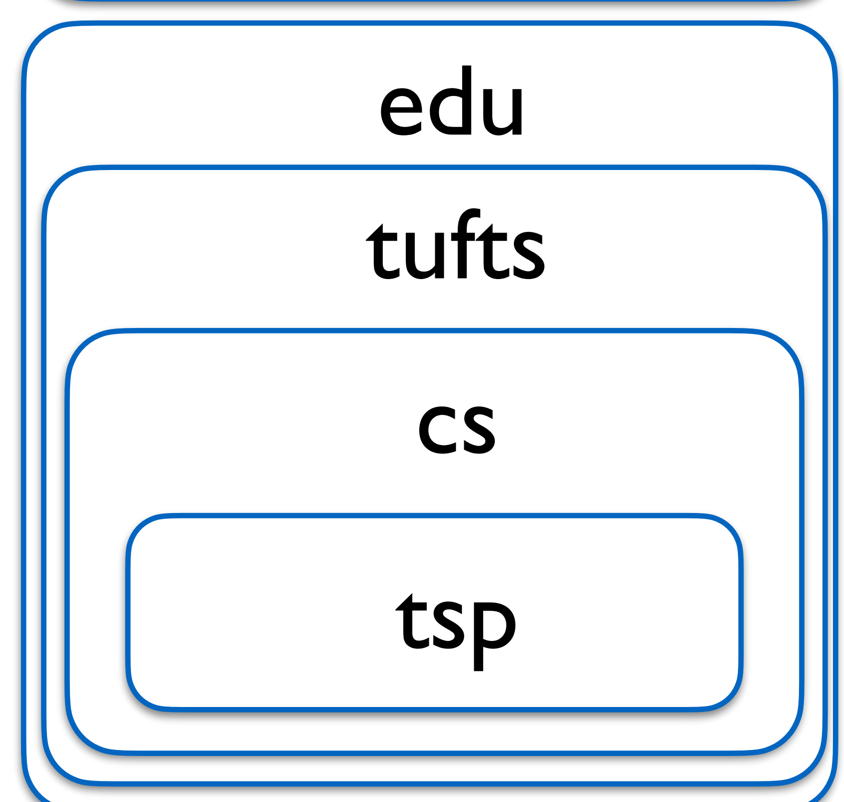


- IP Addresses:
  - e.g., 141.161.20.3

High-order ⟷ Low-order

- Hostnames:
  - e.g., tsp.cs.tufts.edu

Low-order ⟷ High-order



The Old Fashioned Way

- Each host stores mapping between hostnames and IP addresses
- Local */etc/hosts* file:

```
127.0.0.1        localhost
141.161.20.3     karma.cs.georgetown.edu karma
158.130.69.163   www.cis.upenn.edu
18.9.22.169      www.mit.edu
```

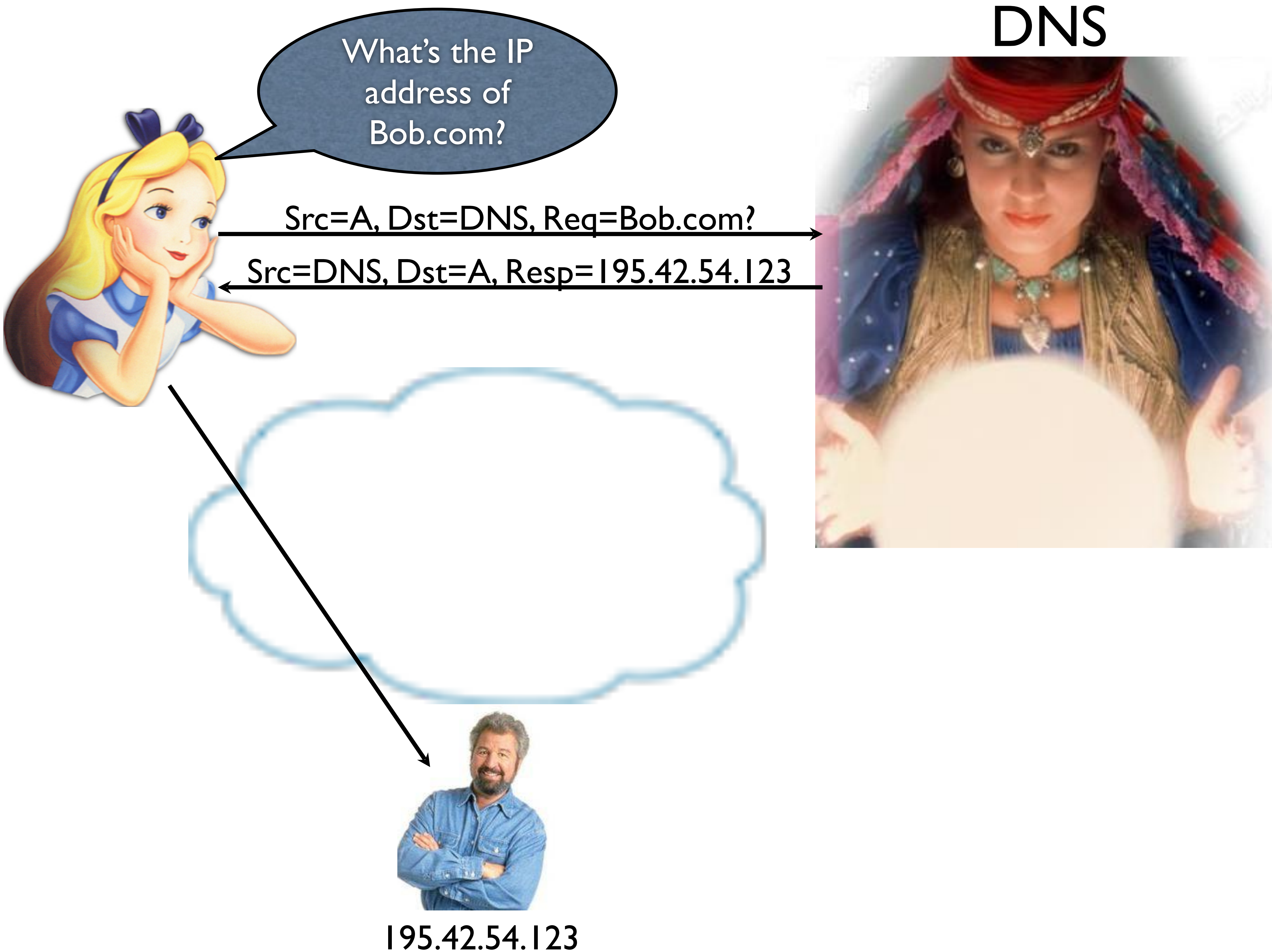
- **Q: Does this scale?**

Plan for today

- Administrivia
- Review worms, bots, an DoS
- Domain Name Service (DNS)
    - The protocol
    - Vulnerabilities
    - Mitigations — DNSSec

Domain Name System (DNS)

- **Distributed** translation service between hostnames and IP addresses
- http://tsp.cs.tufts.edu → 130.64.23.35


DNS
What's the IP address of Bob.com?
Src=A, Dst=DNS, Req=Bob.com?
Src=DNS, Dst=A, Resp=195.42.54.123
195.42.54.123

DNS

- DNS is distributed
  - Organized as a tree, with the **root nameservers** at the top
  - Each **top-level domain (TLD)** (e.g., .com, .edu, .gov, .uk) served by a separate root nameserver
  - Authoritative Name Servers responsible for their domains
  - Domain information stored as a **zone record**

Name servers

- **Authoritative Name Server**: gives authoritative results for hostnames that have been configured
- Domains are registered with a **domain name registrar** (e.g., GoDaddy)
  - Each domain must have one primary and at least one secondary name servers
  - For reliability in case of failure

TLDs

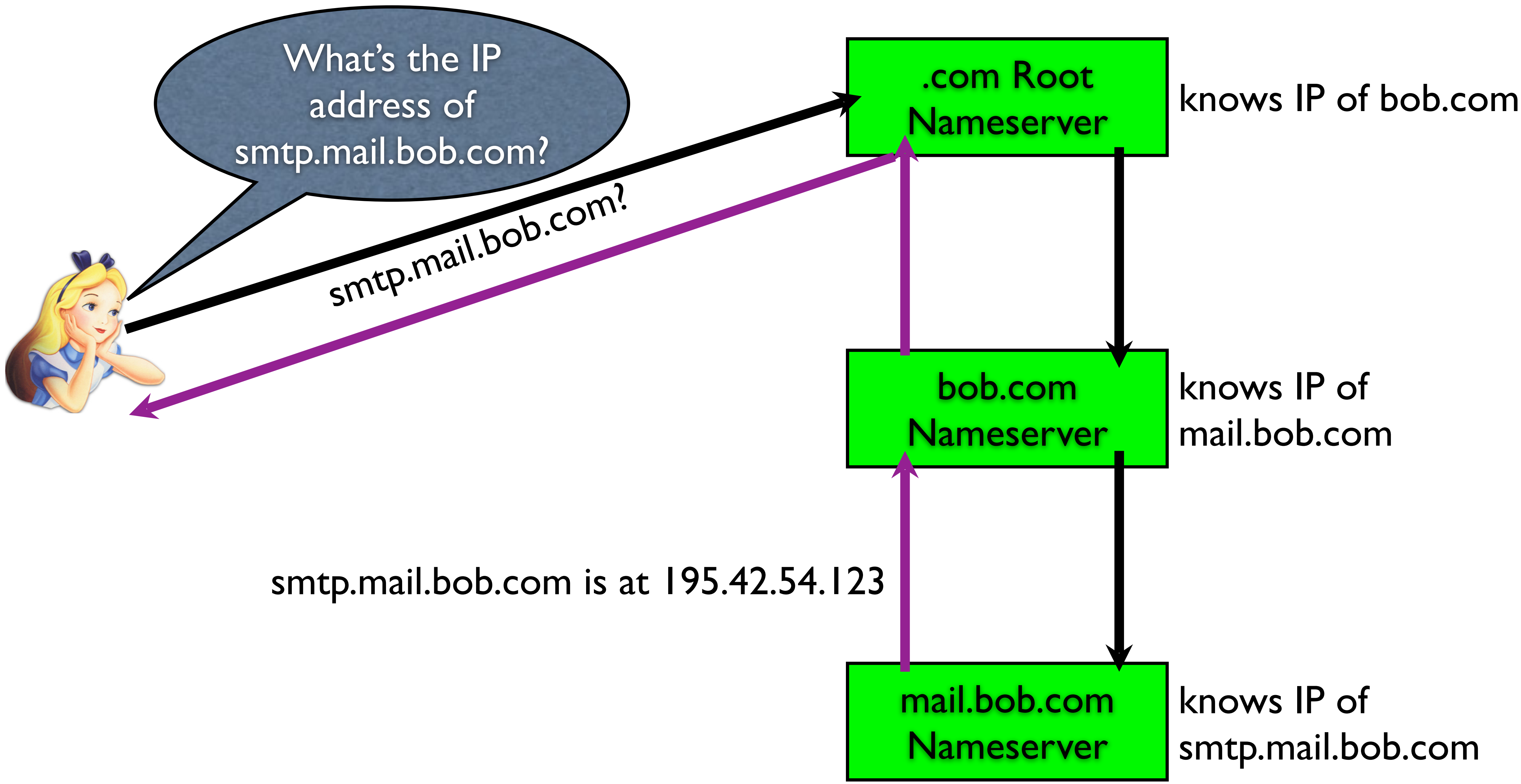
- Name servers pre-loaded with IP addresses of TLD name servers

```
A.ROOT-SERVERS.NET.  IN  A  198.41.0.4
B.ROOT-SERVERS.NET.  IN  A  192.228.79.201
C.ROOT-SERVERS.NET.  IN  A  192.33.4.12
...
M.ROOT-SERVERS.NET.  IN  A  202.12.27.33
```

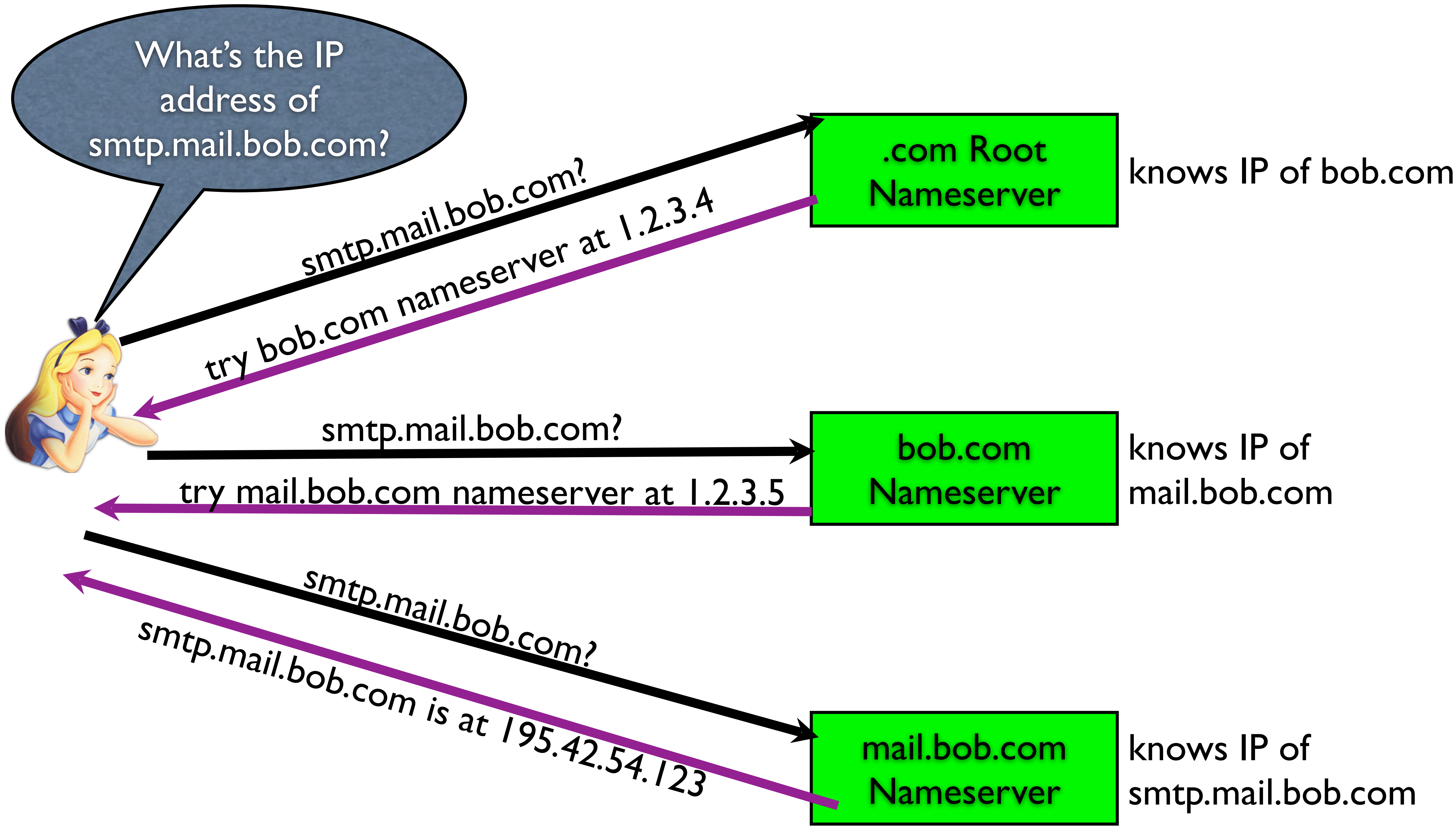
DNS

- Many record types:
  - **A** Records: Maps hostname to IPv4 address
  - **AAAA** Records: Maps hostname to IPv6 address
  - **CNAME** Records: Specifies alias for hostname
  - **MX** Records: Maps hostname to list of Mail Transfer Agents (MTAs)
  - **SOA** Records: Specifies authoritative info about zone

Naive Recursive Query



Naive Iterative Query



Naive Iterative Query

What's the IP

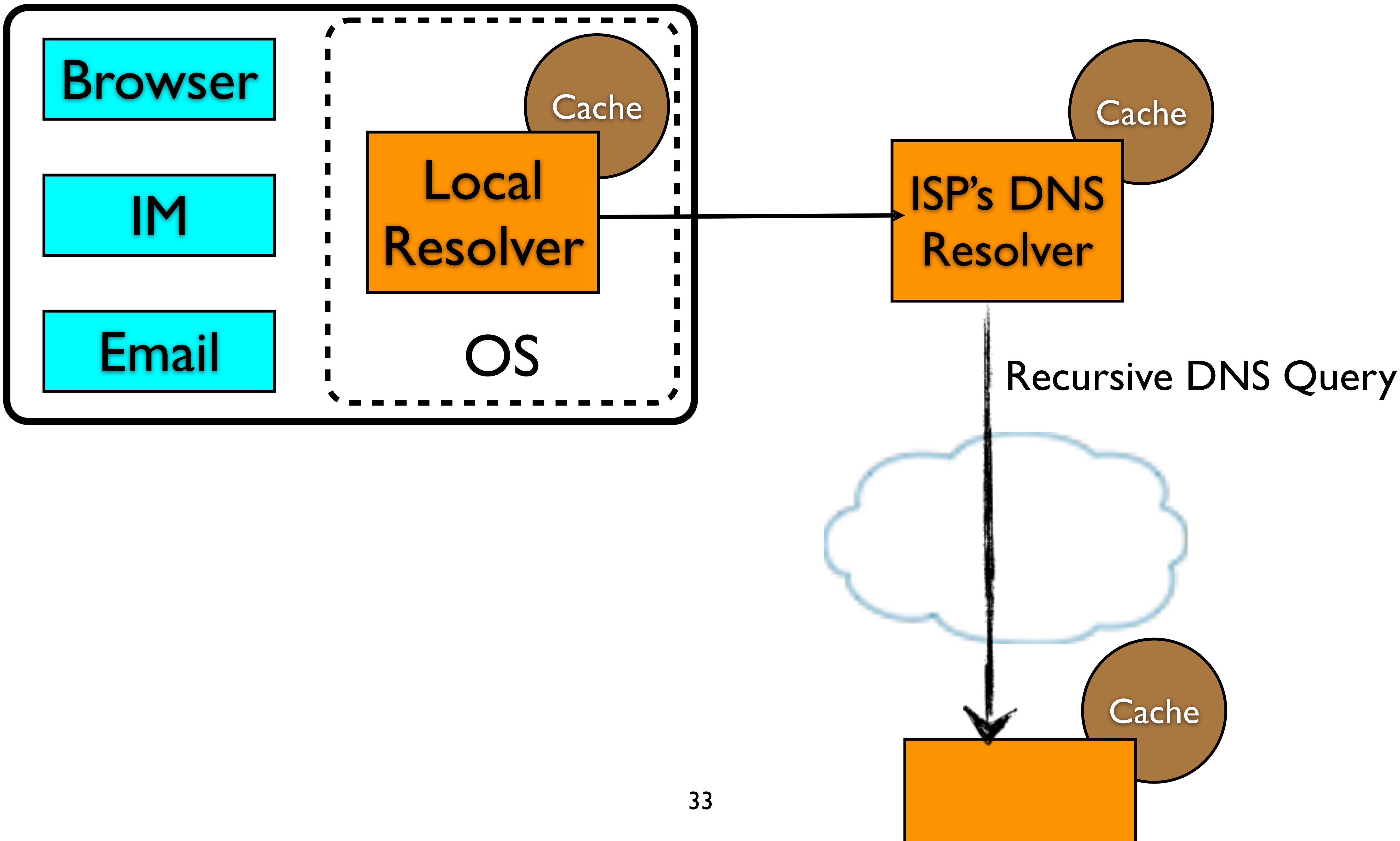
bob.com

Why are these two approaches (recursive and iterative) unscalable?

Nameserver

smtp.mail.bob.com

DNS in the Real World



DEMO

Plan for today

- Administrivia
- Review worms, bots, an DoS
- Domain Name Service (DNS)
  - The protocol
  - Vulnerabilities
  - Mitigations — DNSSec

DNS Vulnerabilities

- DNS requests and responses are not authenticated
  - Yet many applications trust DNS resolutions
  - ... or, more accurately, they don't consider the threat at all
  - Spoofing of DNS is very dangerous **-- WHY?**
- Caching doesn't help:
  - DNS relies heavily on caching for efficiency, enabling **cache pollution** attacks
  - Once something is wrong, it can remain that way in caches for a long time
  - Data may be corrupted before it gets to authoritative server

A Cache Poisoning Attack

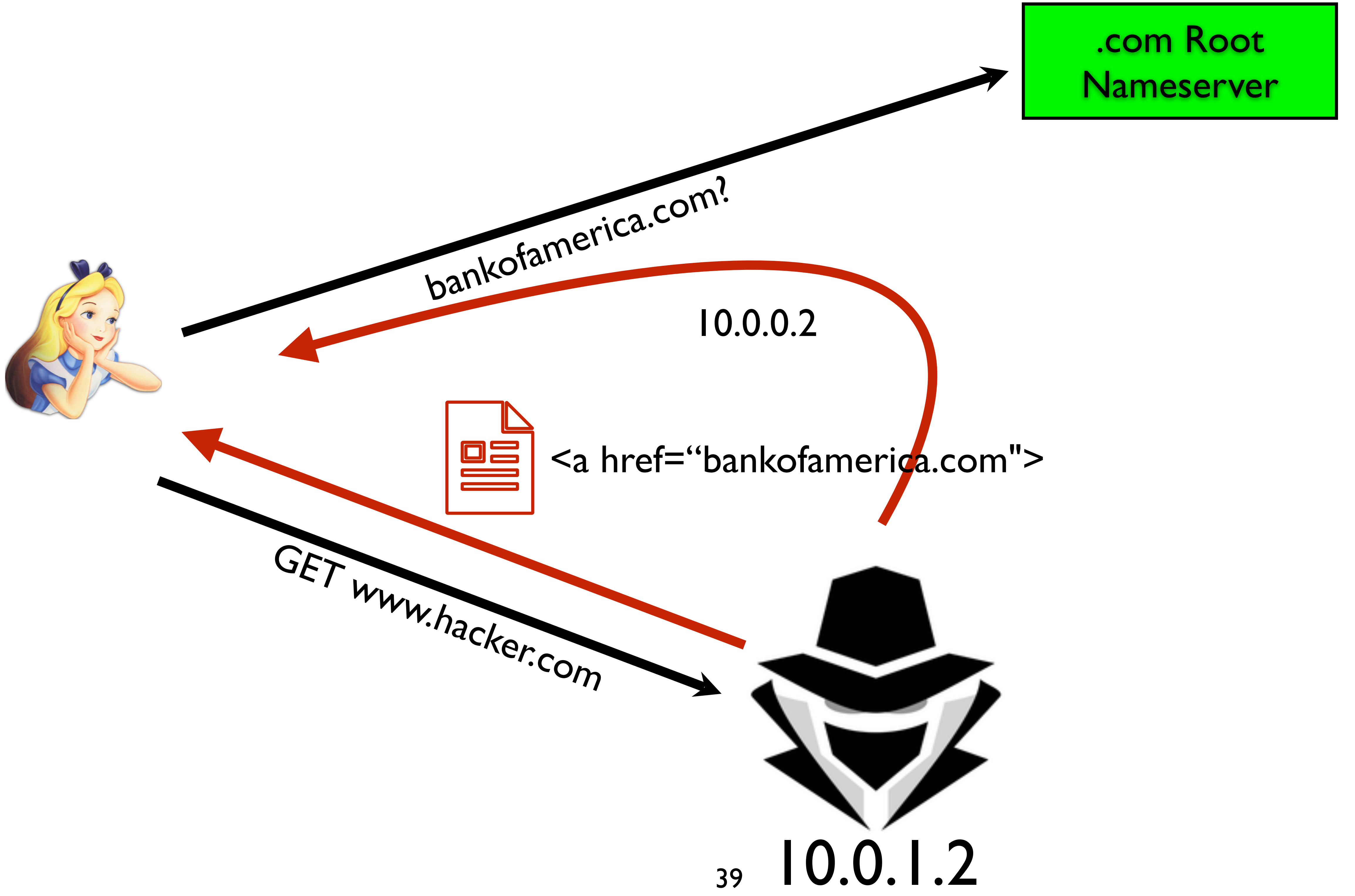
- All DNS requests have a unique query ID
- The nameserver/resolver uses this information to match up requests and responses -- this is useful since DNS uses UDP
- If an adversary can guess the query ID, then it can forge the responses and pollute the DNS cache
  - 16-bit query IDs (only 2^{16}=65536 possible query IDs)
  - Some servers increment IDs (or use some other predictable algo)
  - gethostbyname returns as soon as it gets a response, so first one in wins!!!
- Note: If you can observe the traffic going to a name server, you can pretty much arbitrarily 0wn the Internet for the clients it serves

A Cache Poisoning Attack

- A simple (and extremely effective) attack:
  1. Wait for Alice to send DNS request to nameserver
  2. Intercept request
  3. Quickly insert a fake response
- If attacker is faster and/or closer to Alice than the DNS server, then the attack is successful
  - Advantage attacker: unlike the name server, the attacker doesn't have to do any actual resolving

How can an attacker do better?

Cache Poisoning



Plan for today

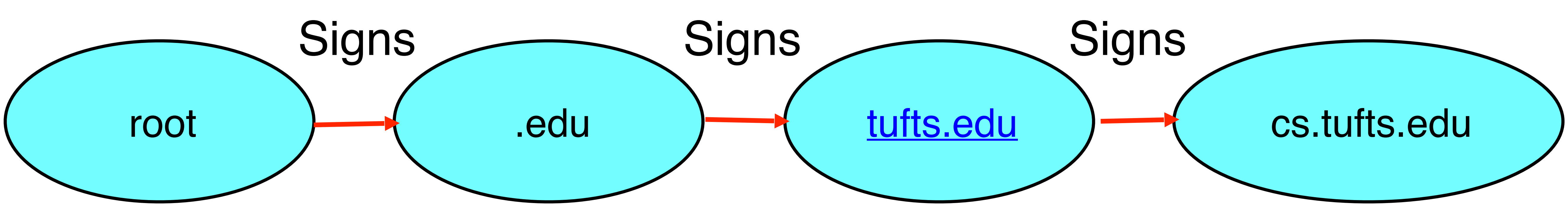
- Administrivia
- Review worms, bots, an DoS
- Domain Name Service (DNS)
  - The protocol
  - Vulnerabilities
  - Mitigations — DNSSec

DNSSEC

- A standards-based (IETF) solution to security in DNS
  - Prevents data spoofing and corruption
  - Authentication (verifiable DNS) using public key infrastructure
  - Authenticates:
    - Communication between servers
    - DNS data
      - content
      - existence
      - non-existence
    - Public keys

DNSSEC Mechanisms

- Each domain signs their "zone" with a private key
- Public keys published via DNS
- Zones signed by parent zones
- Ideally, you only need a self-signed root, and follow keys down the hierarchy



DNSSEC challenges

- Incremental deployability
  - Everyone has DNS, can't assume a flag day
- Resource imbalances
  - Some devices can't afford real authentication
- Cultural
  - Who gets to control the root keys? (US, China, EFF, Tufts?)
  - Most people don't have any strong reason to have secure DNS ($$$ not justified in most environments)
  - Lots of transitive trust assumptions
  - Take away: DNSSEC will be deployed, but it is unclear whether it will be used appropriately/widely

Currently ~25-30% of DNS queries are validated

What we did today

- Review worms, bots, an DoS
- Domain Name Service (DNS)
  - The protocol
  - Vulnerabilities
  - Mitigations — DNSSec