Mitigating DNS DoS Attacks

Hitesh Ballani and Paul Francis
Cornell University

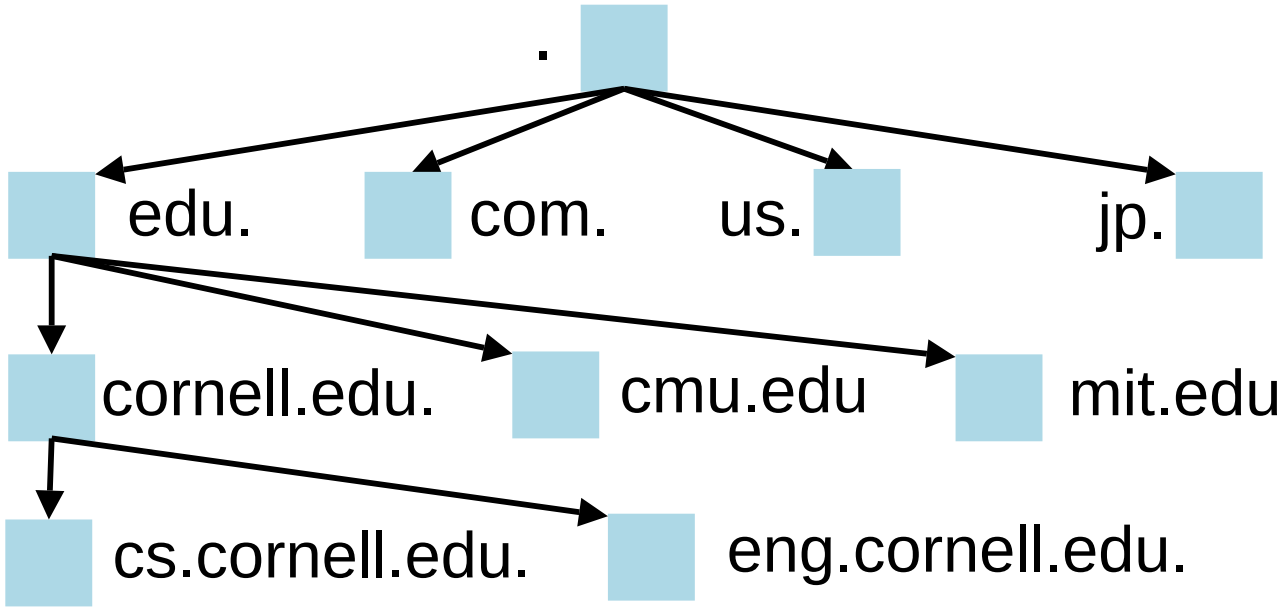
ACM CCS 2008

DoS attacks on DNS

Attack: Flood the nameservers of a DNS zone

Goal: Disrupt the resolution of

- The zone's resource records
- And the records for any of the sub-zones

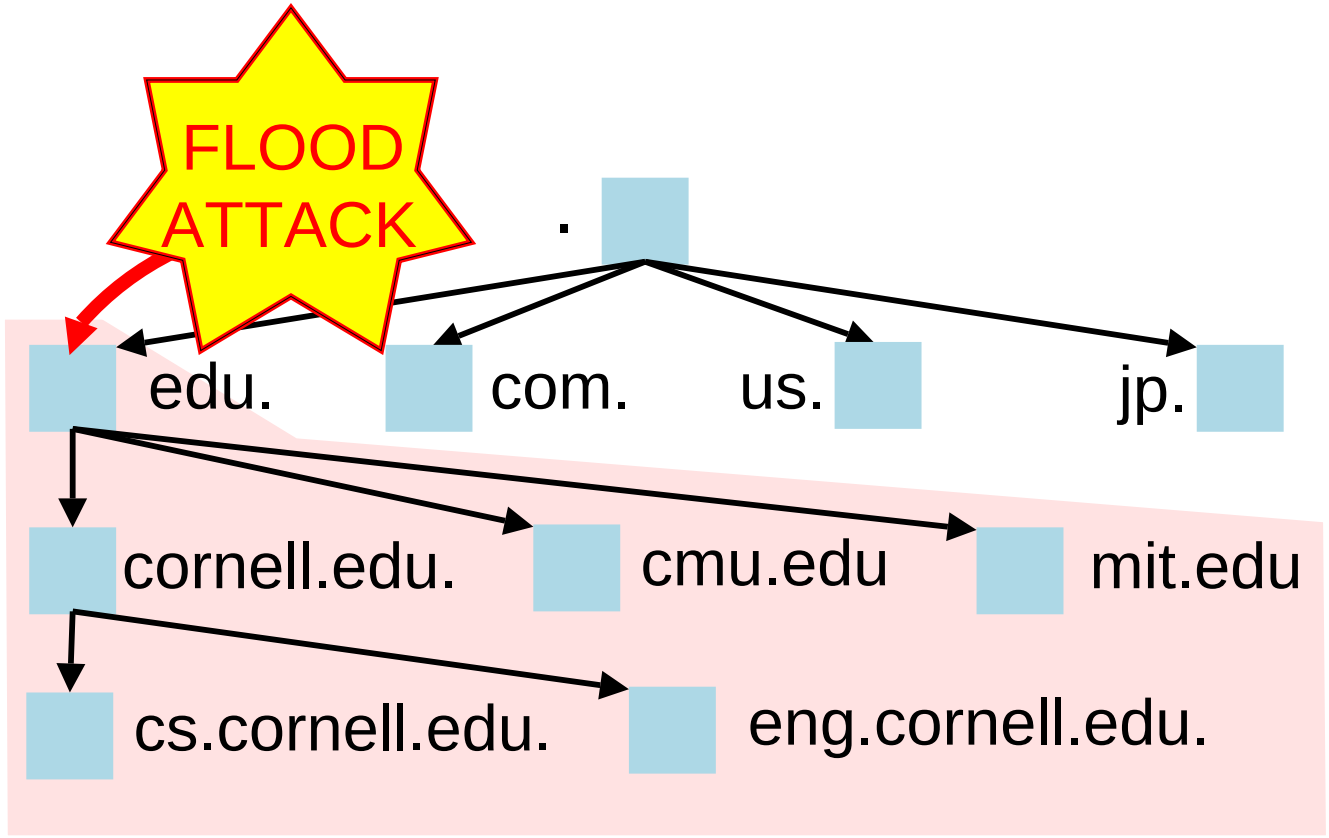


DoS attacks on DNS

Attack: Flood the nameservers of a DNS zone

Goal: Disrupt the resolution of

- The zone's resource records
- And the records for any of the sub-zones



DoS attacks on DNS

Attack: Flood the nameservers of a DNS zone

Goal: Disrupt the resolution of

- The zone's resource records
- And the records for any of the sub-zones

Attacks aplenty (some successful, other not!)

- Microsoft attacked (2001)
- DNS Root Servers attacked (2002)
- SCO attacked (2003)
- Akamai attacked (2004)
- Root Servers, TLDs and UltraDNS (2006)
- Root Servers attacked (2007)

Networking community to the rescue

- Kangasharaju et. al. [INFOCOM'00]
- Cox et. al. [IPTPS'02]
- Theimer et. al [ICDCS'02]
- Ramasubramaniam et. al. [SIGCOMM'04]
- Handley et. al. [HotNets'05]
- Deegan et. al. [SIGCOMM CCR'05]

Networking community to the rescue

- Kangasharaju et. al. [INFOCOM'00]
- Cox et. al. [IPTPS'02]
- Theimer et. al [ICDCS'02]
- Ramasubramaniam et. al. [SIGCOMM'04]
- Handley et. al. [HotNets'05]
- Deegan et. al. [SIGCOMM CCR'05]

Decouple data distribution from authority hierarchy

Ensure **availability** of data distribution mechanism

Networking community to the rescue

- Kangasharaju et. al. [INFOCOM'00]
- Cox et. al. [IPTPS'02]
- Theimer et. al [ICDCS'02]
- Ramasubramaniam et. al. [SIGCOMM'04]
- Handley et. al. [HotNets'05]
- Deegan et. al. [SIGCOMM CCR'05]

Decouple data distribution from authority hierarchy

Ensure **availability** of data distribution mechanism

- Centralized approaches

Networking community to the rescue

- Kangasharaju et. al. [INFOCOM'00]
- Cox et. al. [IPTPS'02]
- Theimer et. al [ICDCS'02]
- Ramasubramaniam et. al. [SIGCOMM'04]
- Handley et. al. [HotNets'05]
- Deegan et. al. [SIGCOMM CCR'05]

Decouple data distribution from authority hierarchy

Ensure **availability** of data distribution mechanism

- Centralized approaches
- Peer-to-peer approaches

A complementary tact to handle DoS attacks

Do away with the need for 100% availability

Clients are able to resolve a zone's records even when the zone's nameservers are not available

In this paper

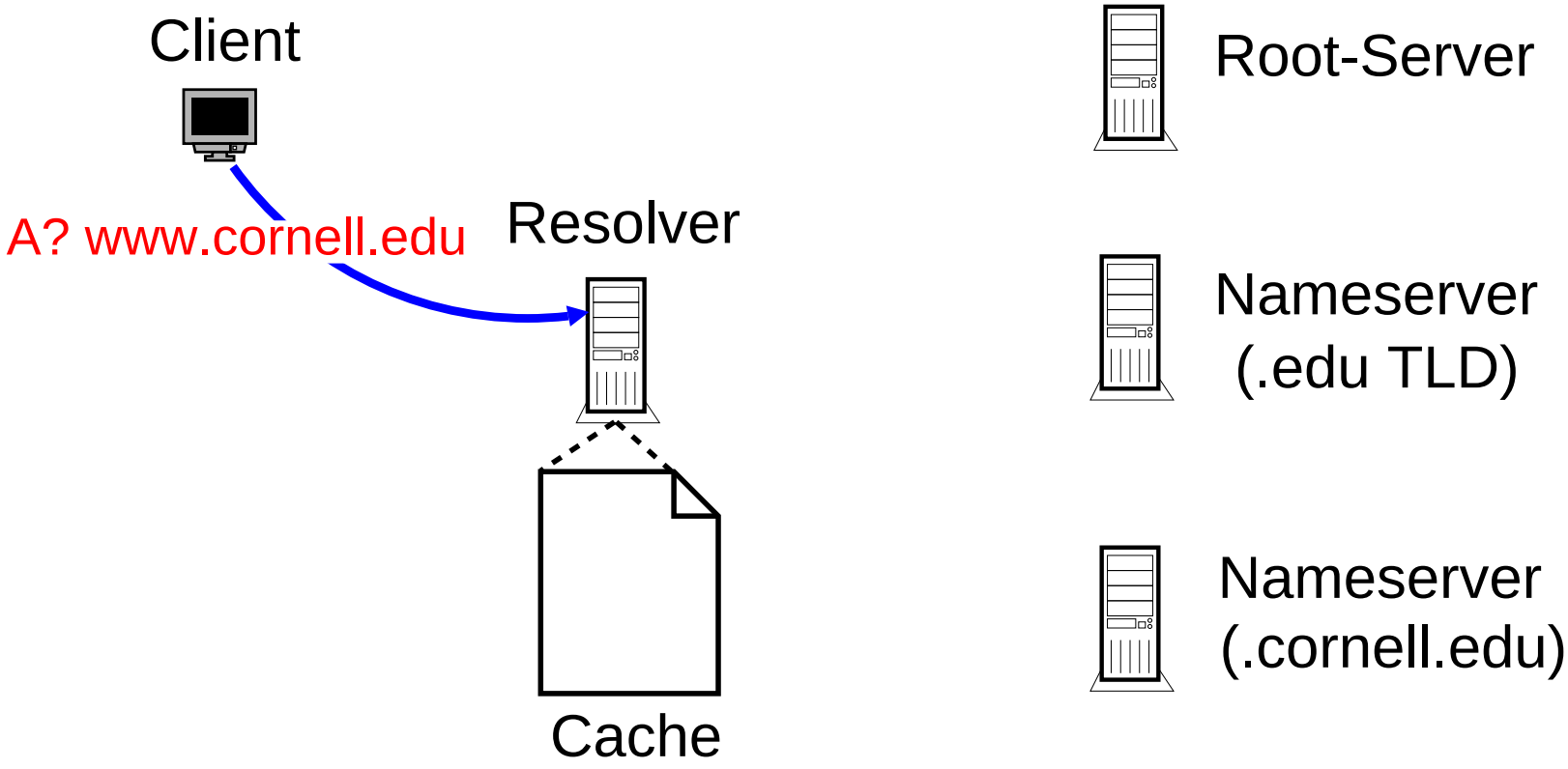
A minor modification in the caching behavior of DNS resolvers

- Reduces the need for nameserver availability in the **existing DNS framework**
- Mitigates the impact of DoS attacks on DNS

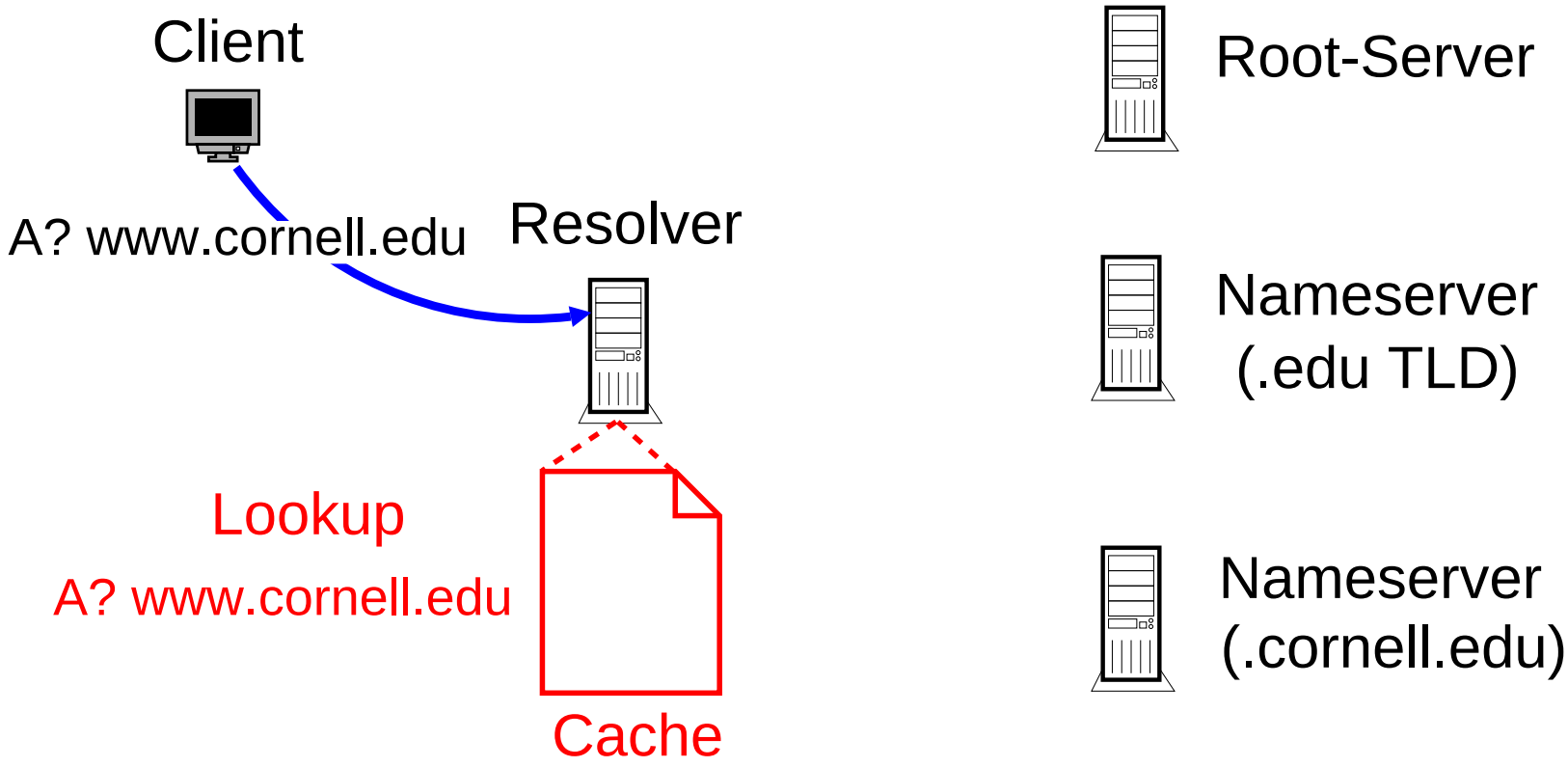
Talk Outline

- Introduction
- DNS Resolvers Today
- Proposed Modification
- Evaluation
- The Good
- The Bad and the Ugly

DNS Resolvers Today



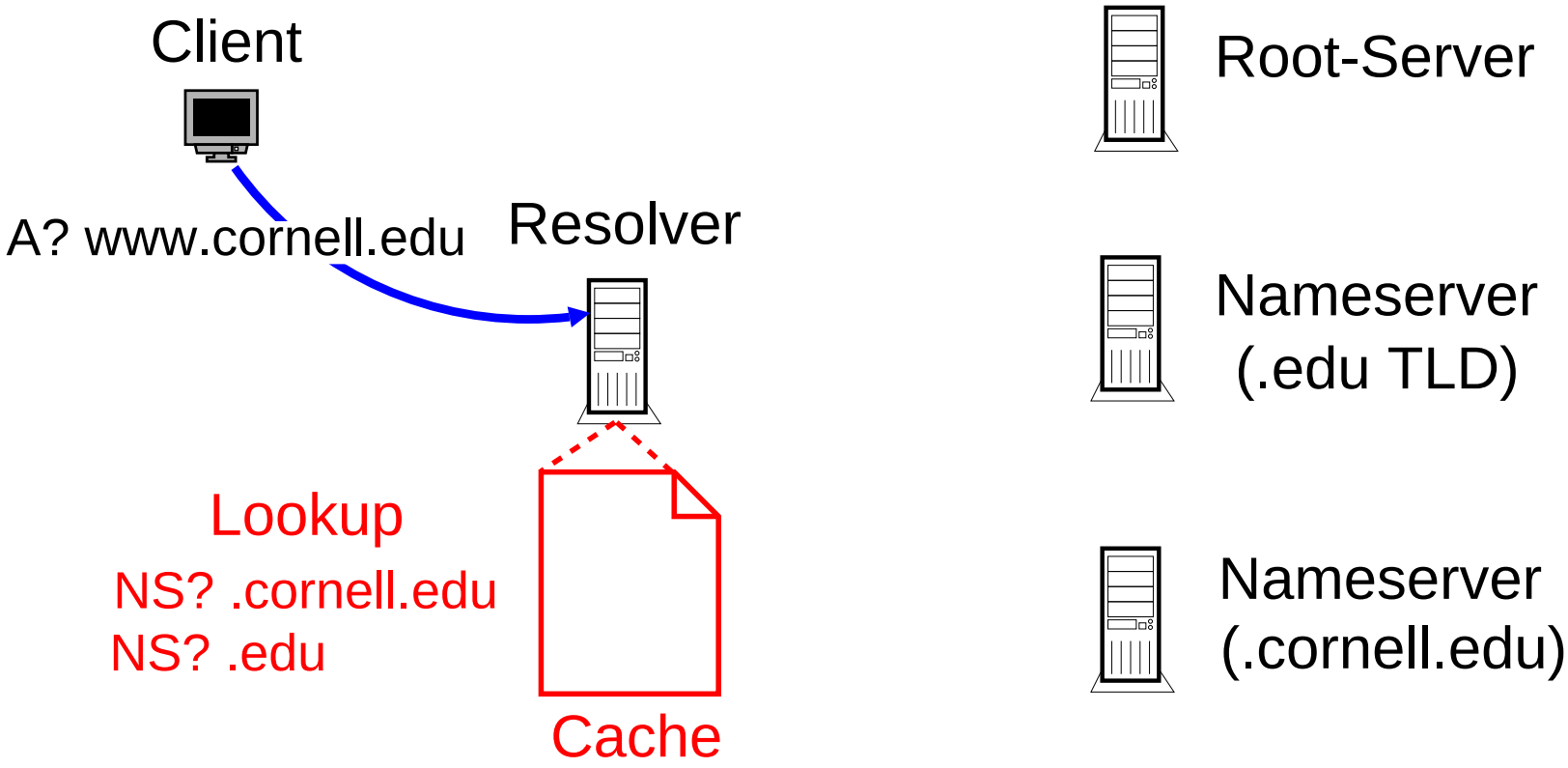
DNS Resolvers Today



Resolution Process

1. Lookup the resolver cache
2. Traverse down the DNS hierarchy
3. Traversal fails $\Rightarrow$ Resolution fails

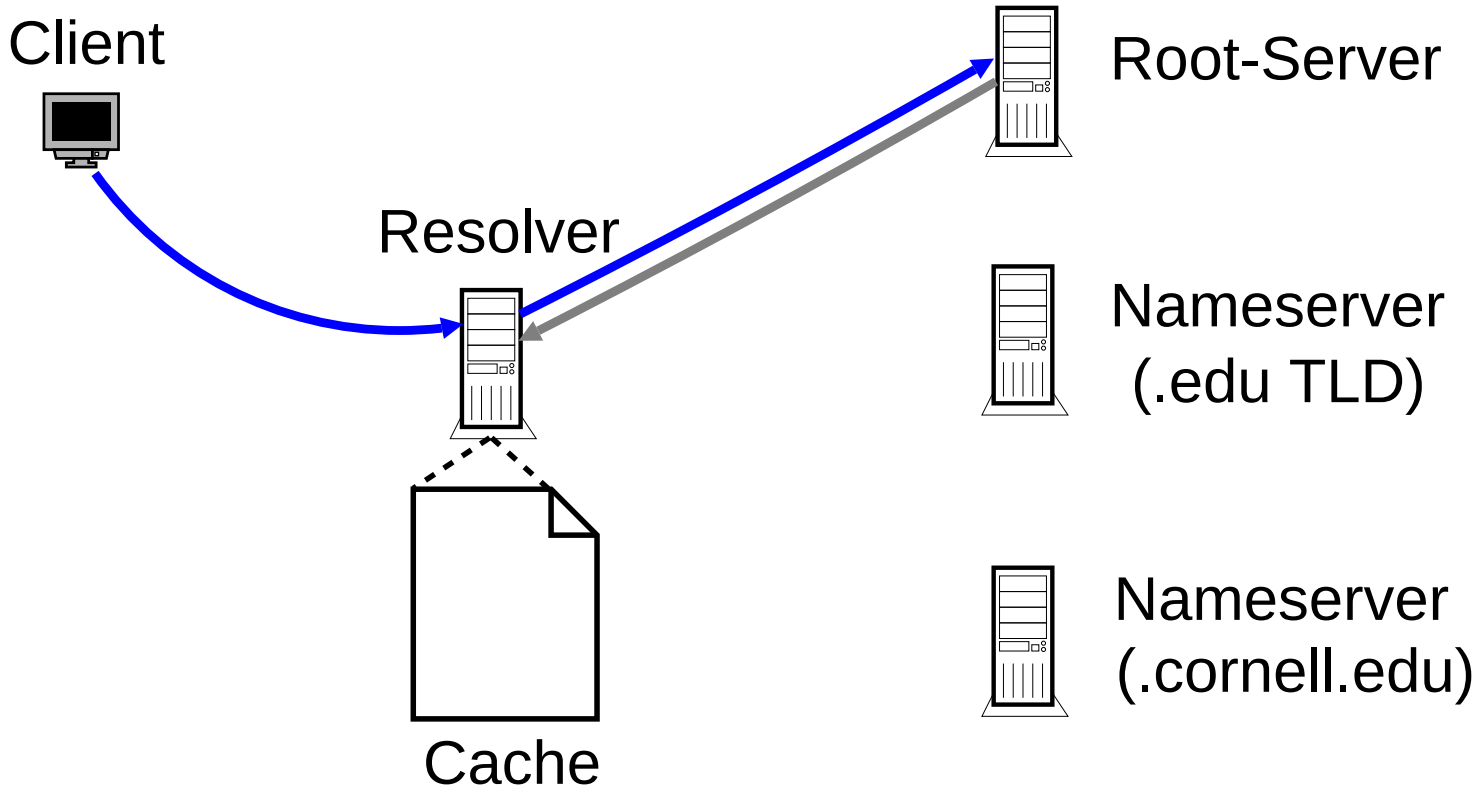
DNS Resolvers Today



Resolution Process

1. Lookup the resolver cache
2. Traverse down the DNS hierarchy
3. Traversal fails $\Rightarrow$ Resolution fails

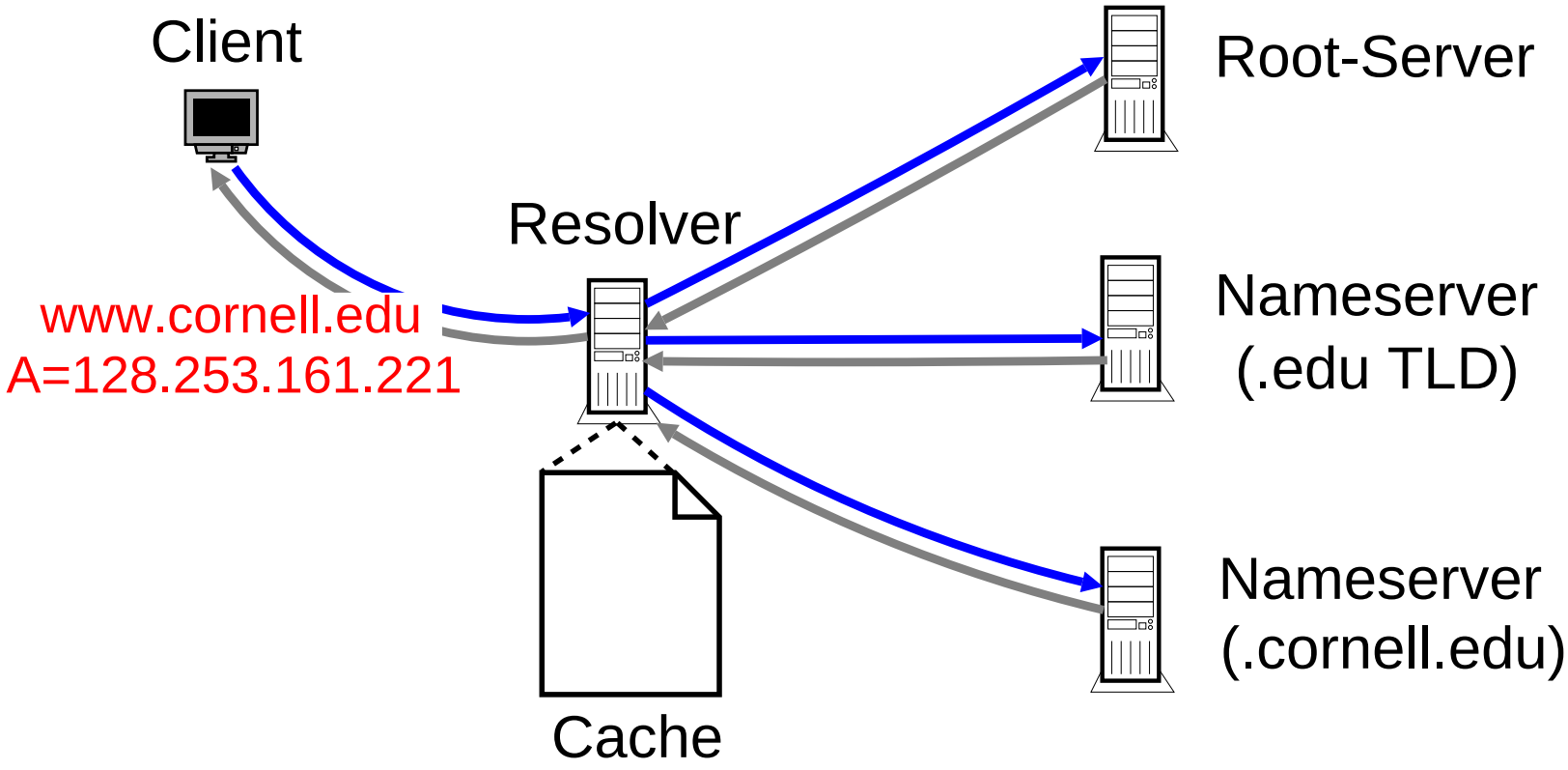
DNS Resolvers Today



Resolution Process

1. Lookup the resolver cache
2. Traverse down the DNS hierarchy
3. Traversal fails $\Rightarrow$ Resolution fails

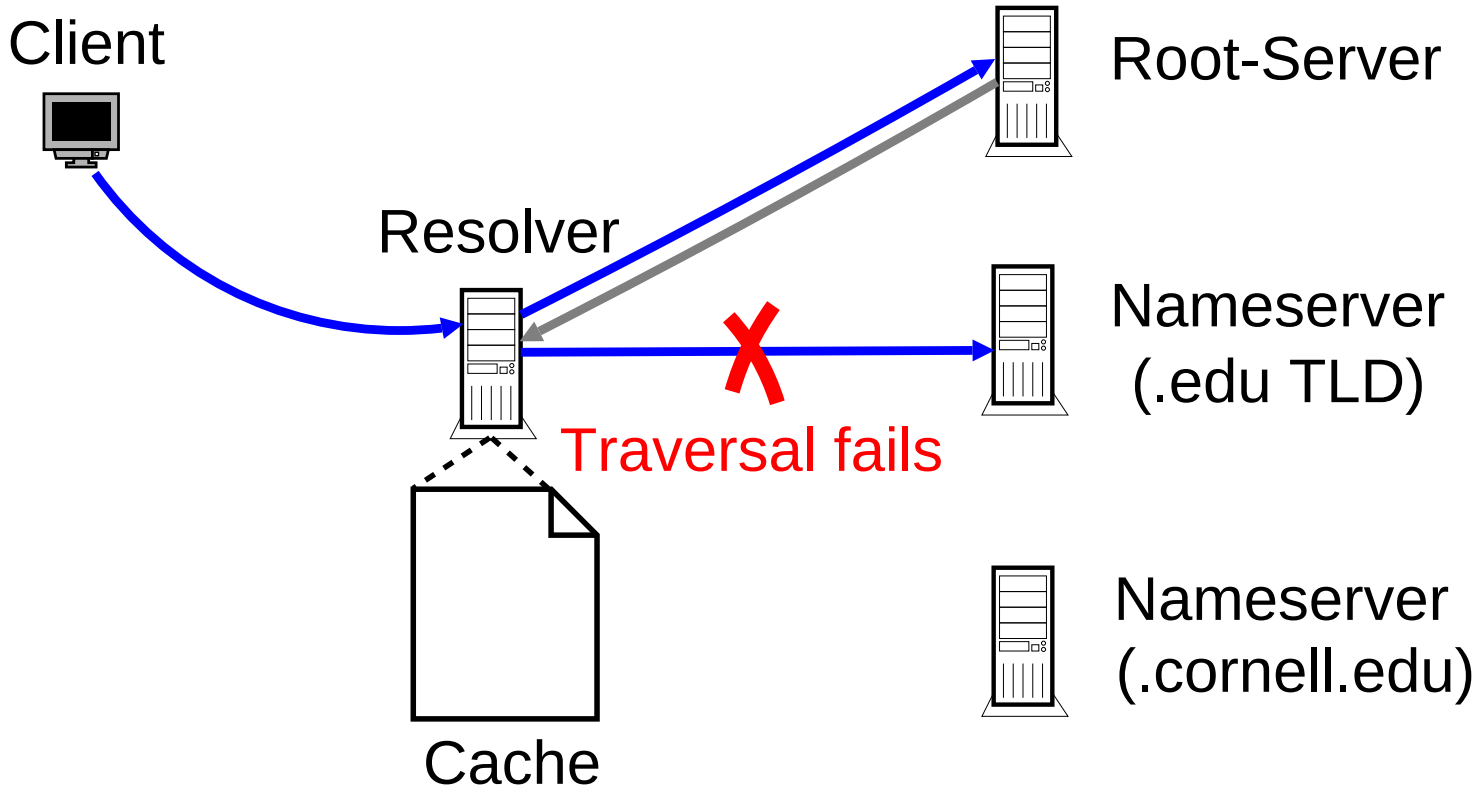
DNS Resolvers Today



Resolution Process

1. Lookup the resolver cache
2. Traverse down the DNS hierarchy
3. Traversal fails $\Rightarrow$ Resolution fails

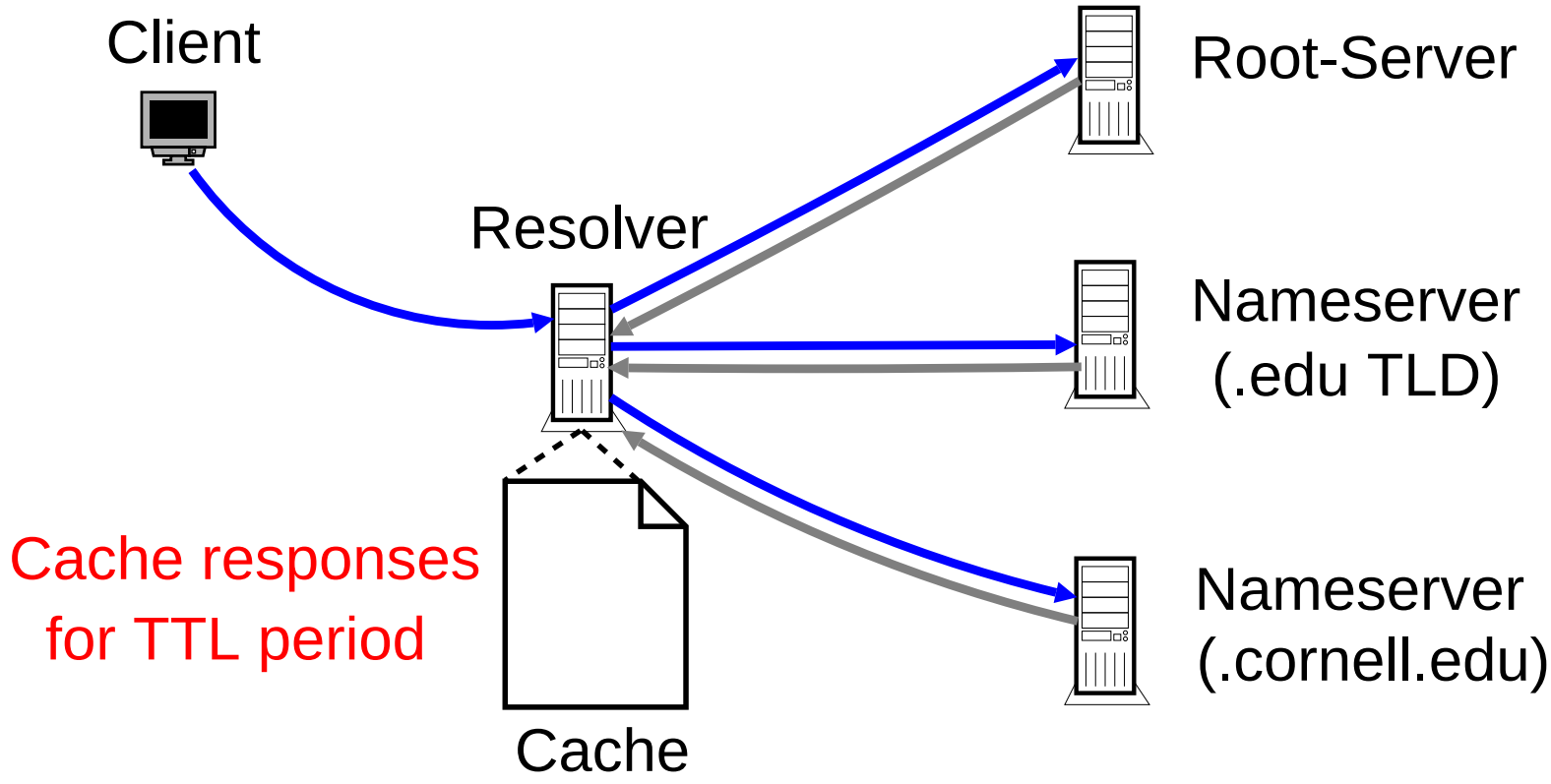
DNS Resolvers Today



Resolution Process

1. Lookup the resolver cache
2. Traverse down the DNS hierarchy
3. Traversal fails $\Rightarrow$ Resolution fails

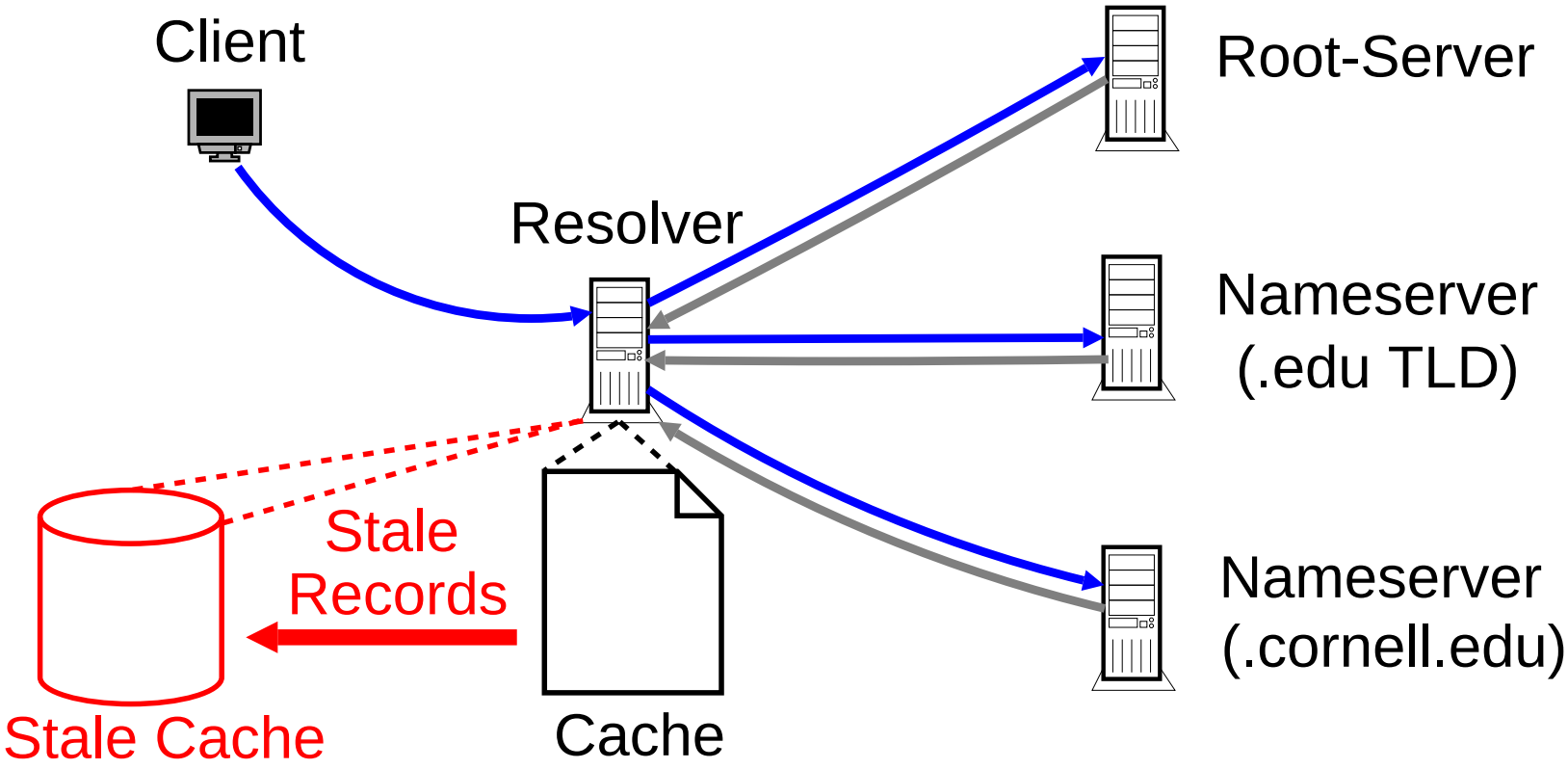
DNS Resolvers Today



Resolver caching behavior

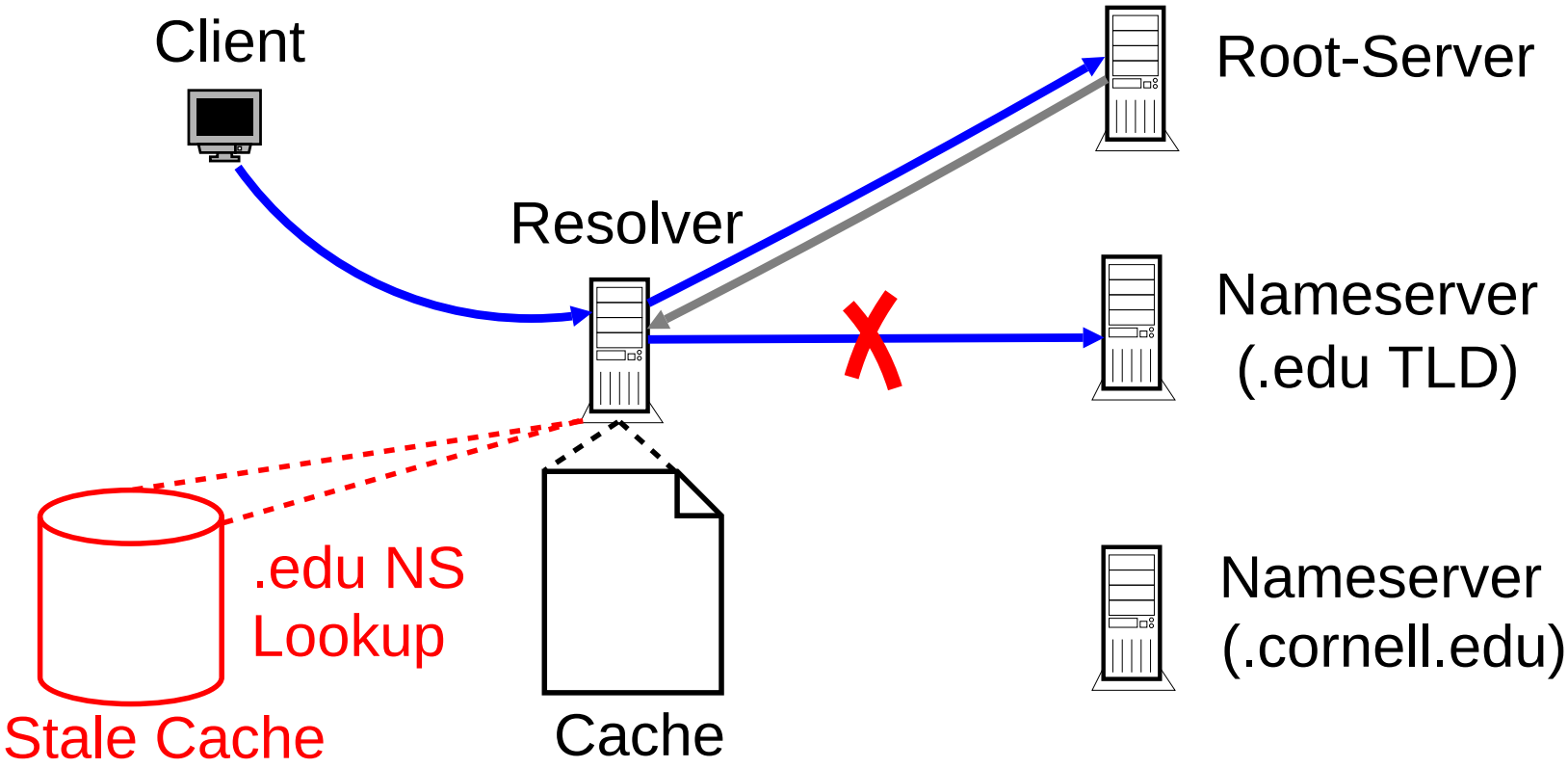
Cached records expunged after their TTL expires

Proposed Modification



Cached records expunged to a **Stale Cache**

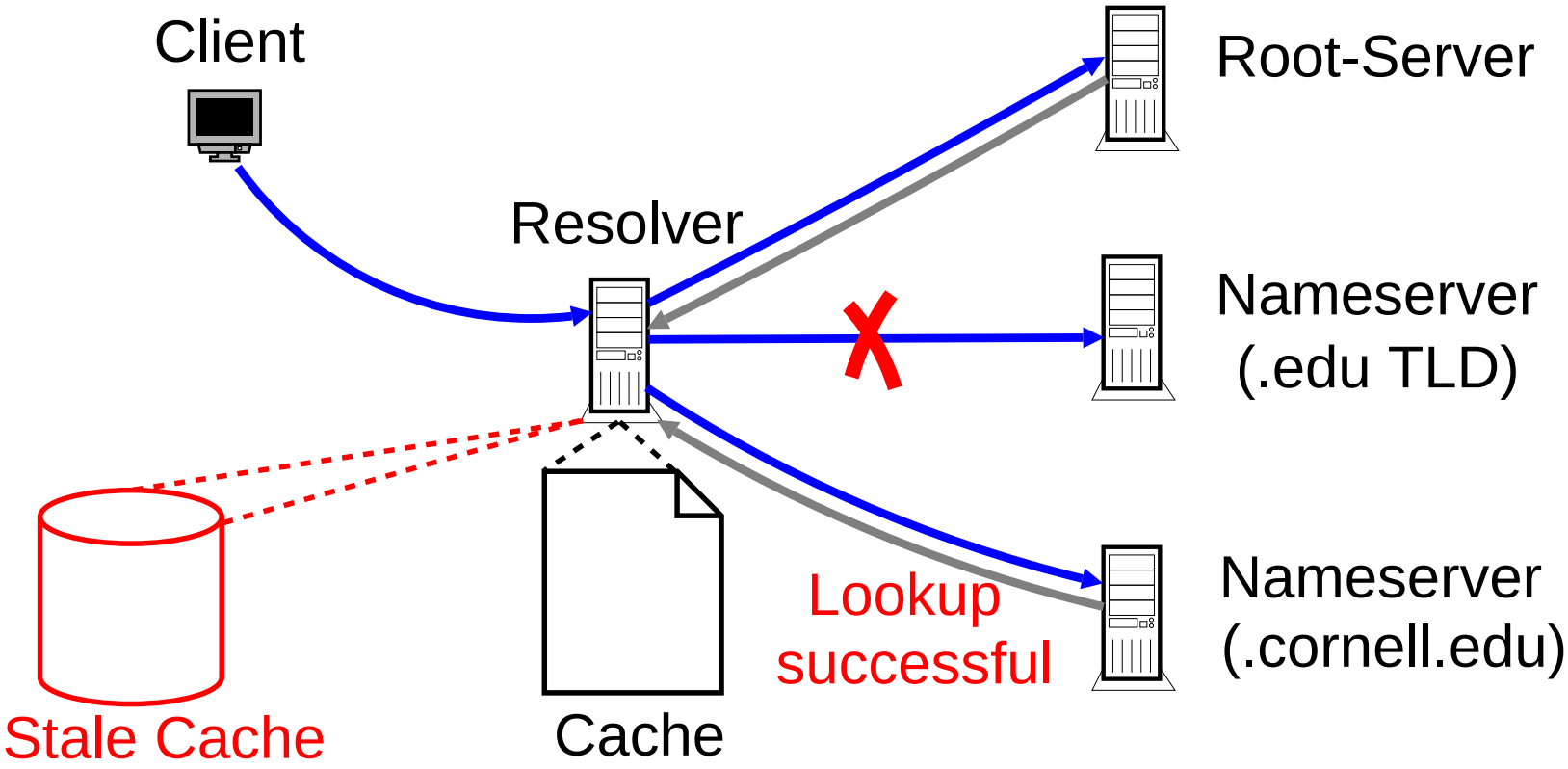
Proposed Modification



Modified Resolution Process

1. Lookup the resolver cache
2. Traverse down the DNS hierarchy
3. Traversal fails $\Rightarrow$ Resolution can continue

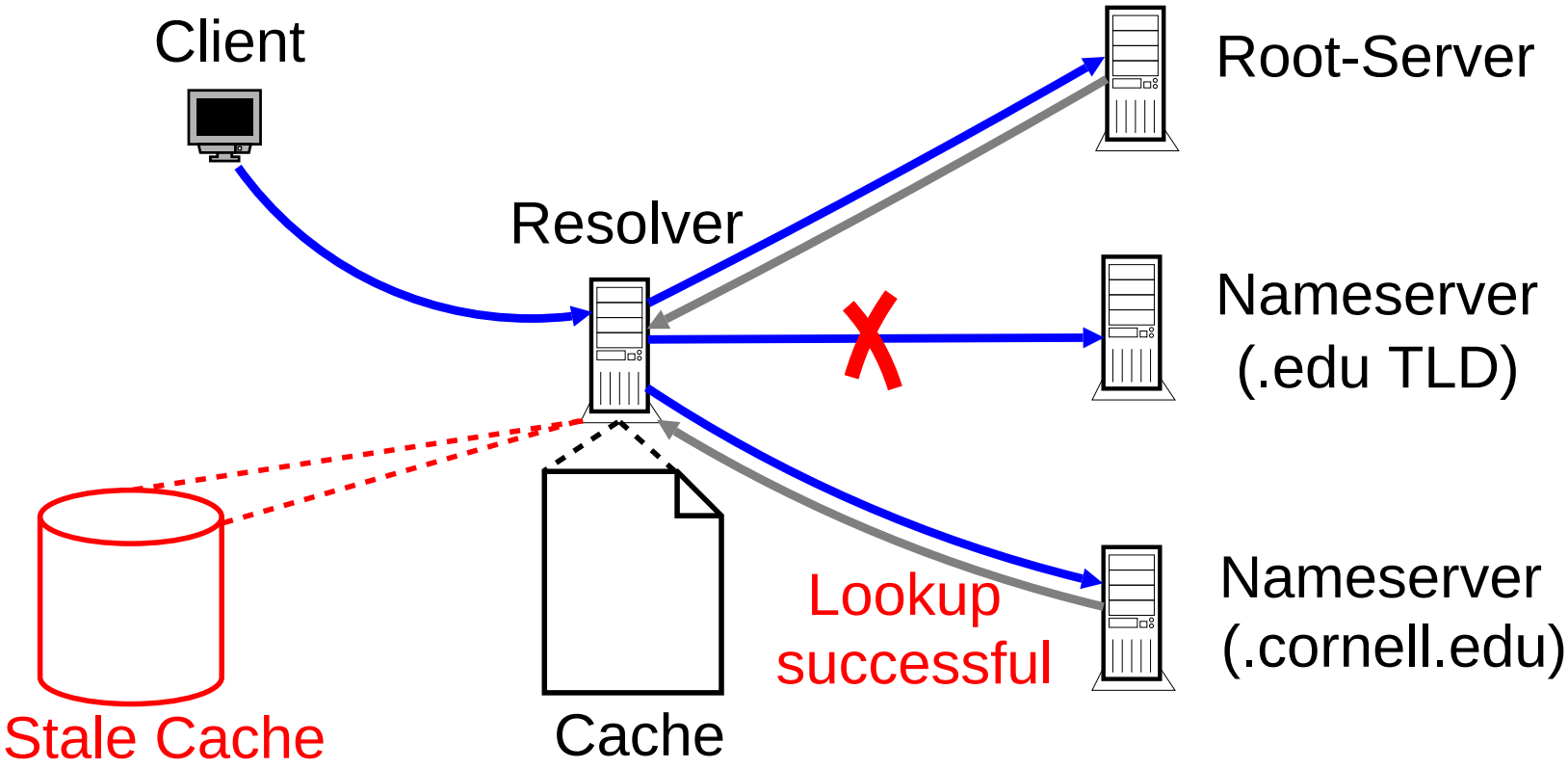
Proposed Modification



Modified Resolution Process

1. Lookup the resolver cache
2. Traverse down the DNS hierarchy
3. Traversal fails $\Rightarrow$ Resolution can continue

Proposed Modification



Stale records for a zone used **only** when the nameservers for the zone are unavailable

Stale Cache Details

Expunging records from the Stale Cache

Responses from nameservers used to clean up the stale cache

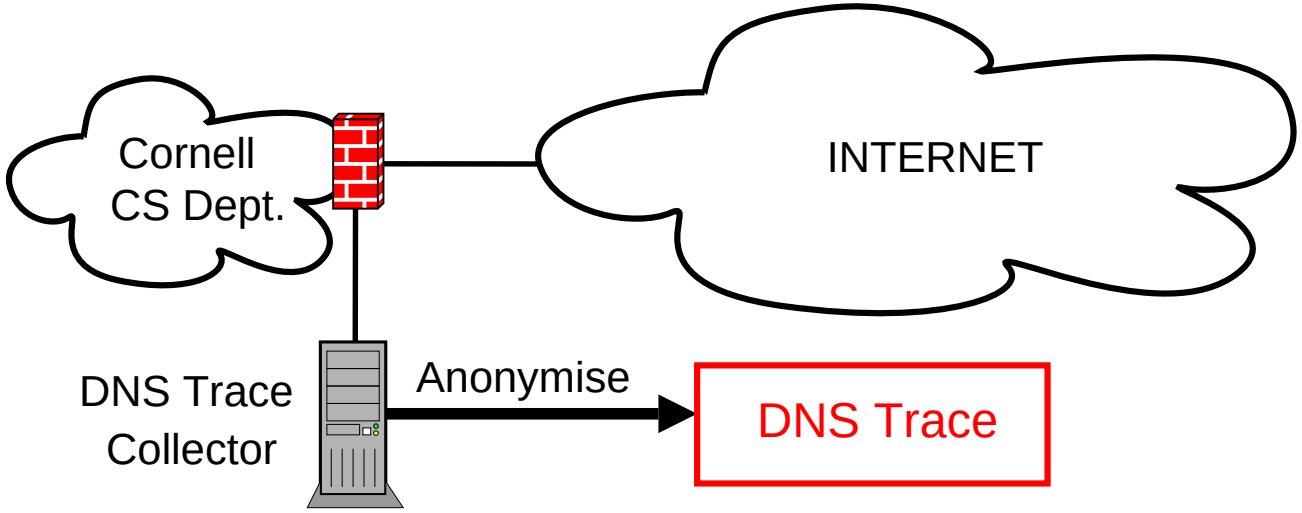
Stale Cache lookup time not critical

Lookups can be done while querying the nameservers

Talk Outline

- Introduction
- DNS Resolvers Today
- Proposed Modification
- Evaluation
- The Good
- The Bad and the Ugly

DNS Trace Collection



65-Day DNS Trace

- 138 million DNS packets
- 4.5 million unique DNS names

Trace collected at Border Router

- Network's resolvers could not answer the collected queries

Stale Cache Simulation

Simulation Variables

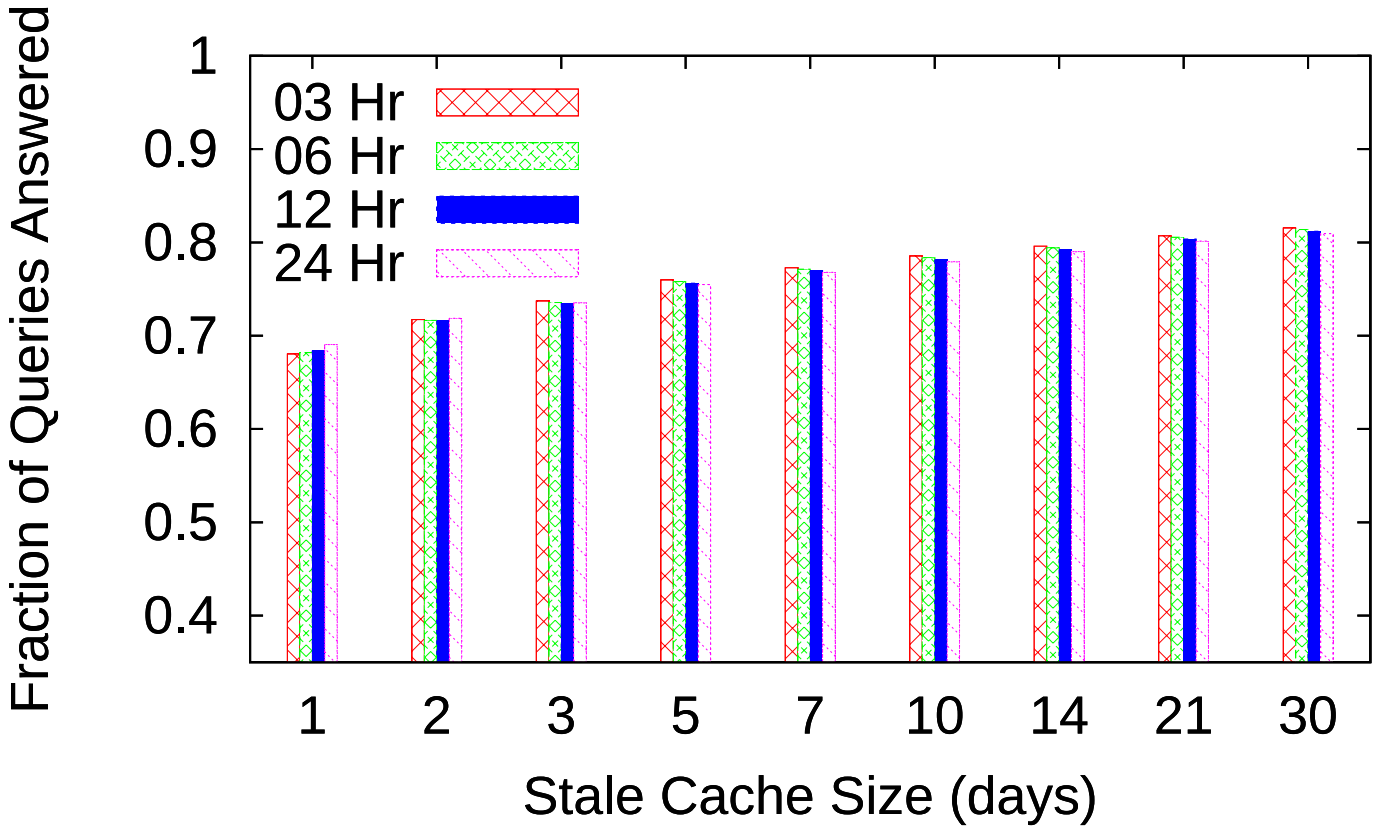
- Stale Cache Size
  - Number of days for which stale records are kept
  - Varied from 1 to 30 days
- Attack Duration
  - Varied from 3 to 24 hours

Is History Useful?

Attack: Entire DNS Fails

Is History Useful?

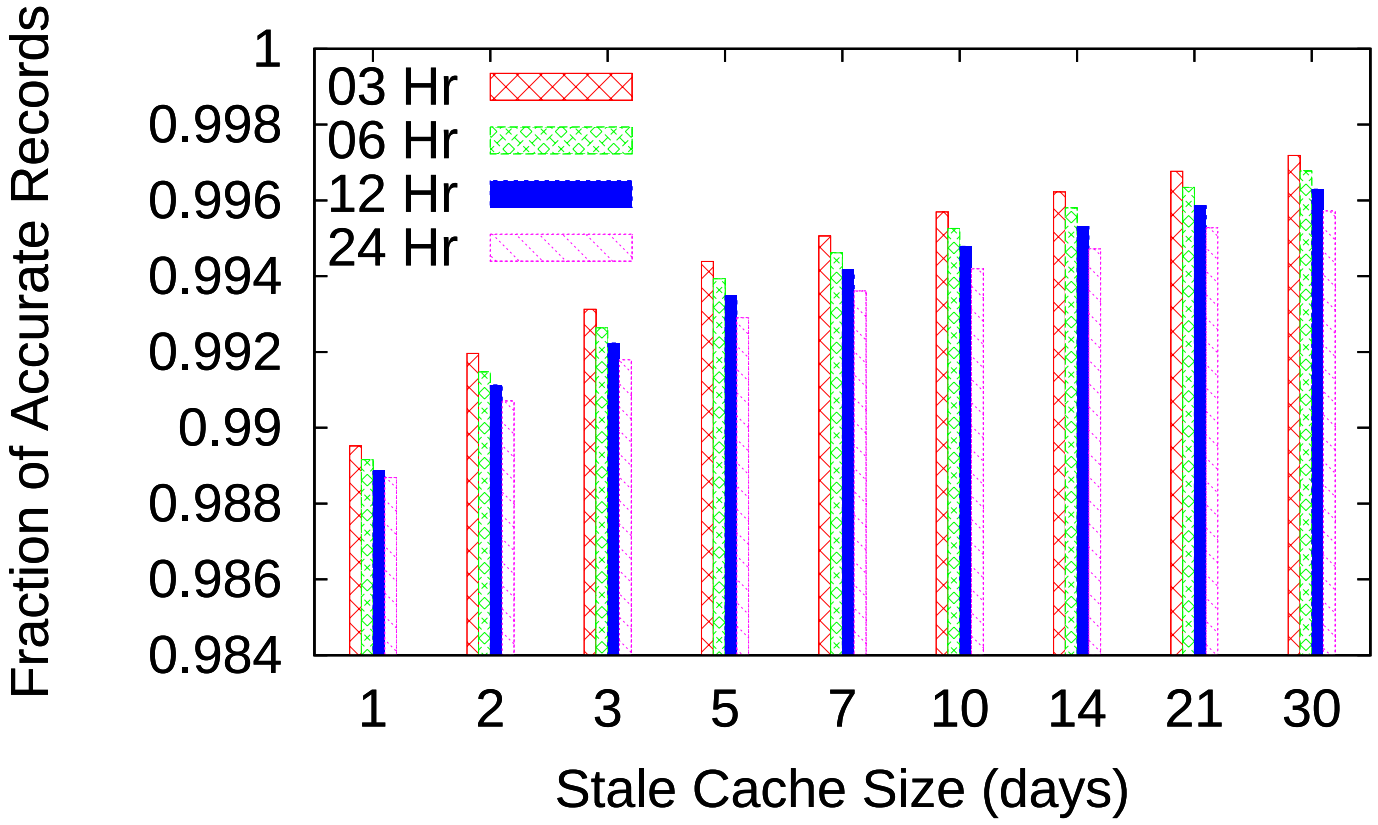
Attack: Entire DNS Fails



14-Day Stale Cache can answer ∼80% queries

Is History Useful?

Attack: Entire DNS Fails



99.6% accurate answers with 14-Day Stale Cache

Other Attack Scenarios

Root-Servers Fail

- Queries for Top-level Domains (TLDs) not resolved
- Stale Cache can answer all such queries

TLD Nameservers Fail

- Queries for two-level names (eg, *a.com*) not resolved
- Stale Cache can answer $>$75% such queries
- Accuracy of answers is 99.4%

Memory Footprint

Depends on network

- Number of hosts
- Query-rate

For the collected trace

- 1300 hosts and $\approx$25 DNS pkts/sec
- Month-long Stale Cache = 313MB

Proposed Modification: Pros

Increased DNS Robustness

- Nameserver availability less crucial
- Mitigates the impact of DoS attacks

Simplicity

- Does not change the basic protocol operation
- Does not impose any load on DNS
- Does not impact the query resolution latency

Incremental Deployment

- Motivation for deployment

Talk Outline

- Introduction
- DNS Resolvers Today
- Proposed Modification
- Evaluation
- The Good
- The Bad and the Ugly

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani					
Vixie					

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani					
Vixie					

Possibility of obsolete information being used

Obselete zone records used by a resolver only if

- Zone's records have been updated since the last access by the resolver
- Zone's nameservers are inaccessible

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani					
Vixie					

Possibility of obsolete information being used

Obselete zone records used by a resolver only if

- Zone's records have been updated since the last access by the resolver
- Zone's nameservers are inaccessible

Trade-off between the possibility of obsolete information being used and the inability to resolve queries

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓				
Vixie	✗				

Possibility of obsolete information being used

Obselete zone records used by a resolver only if

- Zone's records have been updated since the last access by the resolver
- Zone's nameservers are inaccessible

Trade-off between the possibility of obsolete information being used and the inability to resolve queries

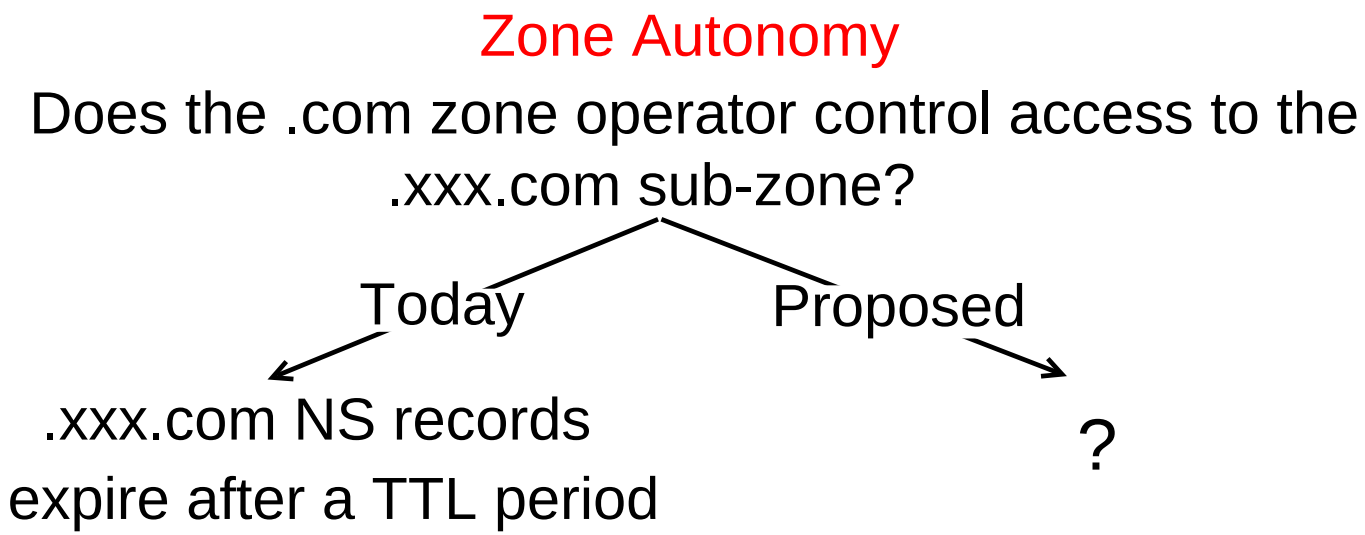
Restrict the Stale Cache size

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓				
Vixie	✗				

Greg Minshall's former CEO: "... he would sign (almost) any contract, as long as he could get out of it in a finite period of time"

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓				
Vixie	✗				

Greg Minshall's former CEO: "... he would sign (almost) any contract, as long as he could get out of it in a finite period of time"

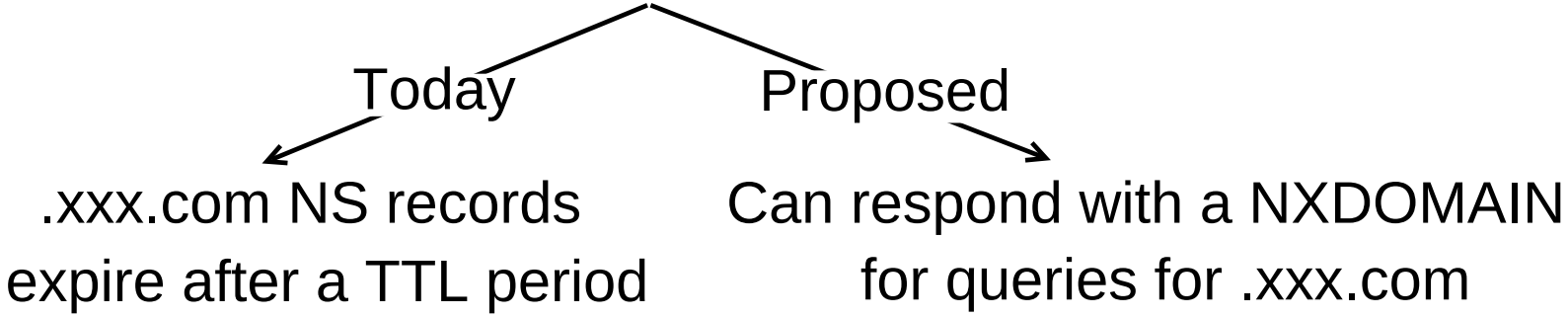


	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓	✓			
Vixie	✗	✗			

Greg Minshall's former CEO: "... he would sign (almost) any contract, as long as he could get out of it in a finite period of time"

Zone Autonomy

Does the .com zone operator control access to the .xxx.com sub-zone?



Zone operators still control access to their sub-zones

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓	✓			
Vixie	✗	✗			

Attackers forcing the use of obsolete records for a zone by

- Waiting for the zone's records to be updated
- And then flooding the zone's nameservers

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓	✓	✗		
Vixie	✗	✗	✓		

Attackers forcing the use of obsolete records for a zone by

- Waiting for the zone's records to be updated
- And then flooding the zone's nameservers

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓	✓	✗		
Vixie	✗	✗	✓		

Resolution latency in the face of attacks

- Resolver must query each nameserver of a zone before using the zone's records from the stale cache
- Given default resolver timeout configurations, this can lead to high resolution latencies

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓	✓	✗	✓	
Vixie	✗	✗	✓	✗	

Resolution latency in the face of attacks

- Resolver must query each nameserver of a zone before using the zone's records from the stale cache
- Given default resolver timeout configurations, this can lead to high resolution latencies

Alleviative: Resolvers configured with aggressive retry and timeout values

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓	✓	✗	✓	
Vixie	✗	✗	✓	✗	

DNS servers can still be overwhelmed

- Unable to update the zone's records

Application servers can still be DoS'ed

	Obsolete	Autonomy	Attack	Latency	Too specific
Ballani	✓	✓	✗	✓	–
Vixie	✗	✗	✓	✗	–

DNS servers can still be overwhelmed

- Unable to update the zone's records

Application servers can still be DoS'ed

- Do DNS servers and Application servers share the network bottleneck?

Summary

A minor modification in the caching behavior of DNS resolvers

- Reduces the need for nameserver availability in the **existing DNS framework**
- Mitigates the impact of DoS attacks on DNS

Stale Cache

- Modifies the DNS caching semantics
- **Does not impact** fundamental DNS characteristics

Thank You!