

# ASAP: A Low-Latency Transport Layer

Wenxuan Zhou, Qingxi Li,  
Matthew Caesar, Brighten Godfrey

# Not so fast, Internet...

# Not so fast, Internet...

- Fetching a popular website

# Not so fast, Internet...

- **Fetching a popular website**

```
time wget www.phdcomics.com
```

# Not so fast, Internet...

- Fetching a popular website

```
time wget www.phdcomics.com
```

```
real 895ms
```

# Not so fast, Internet...

- Fetching a popular website

```
time wget www.phdcomics.com
```

```
real 895ms
```

```
100%[=====>] 10,560          65.2K/s   in 200ms
```

# Not so fast, Internet...

- Fetching a popular website

```
time wget www.phdcomics.com
```

```
real 895ms
```

```
100%[=====>] 10,560          65.2K/s    in 200ms
```

- Single packet latency

```
ping 69.17.116.124 (www.phdcomics.com)
```

```
time=76.3ms
```

# Not so fast, Internet...

- Fetching a popular website

```
time wget www.phdcomics.com
```

```
real 895ms
```

```
100%[=====>] 10,560          65.2K/s    in 200ms
```

- Single packet latency

```
ping 69.17.116.124 (www.phdcomics.com)
```

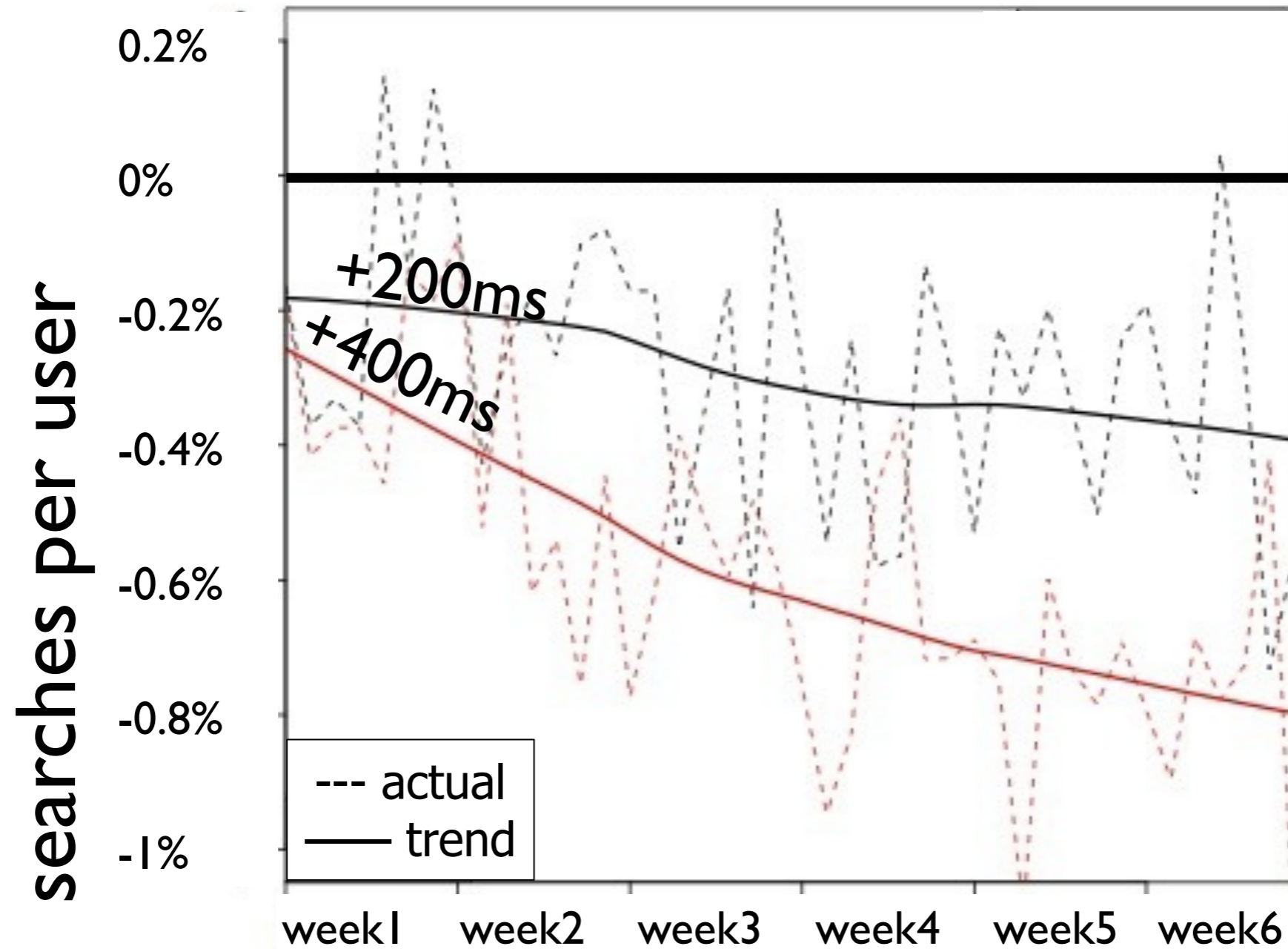
```
time=76.3ms
```

- Why is the web so slow?

# What's a few hundred milliseconds?

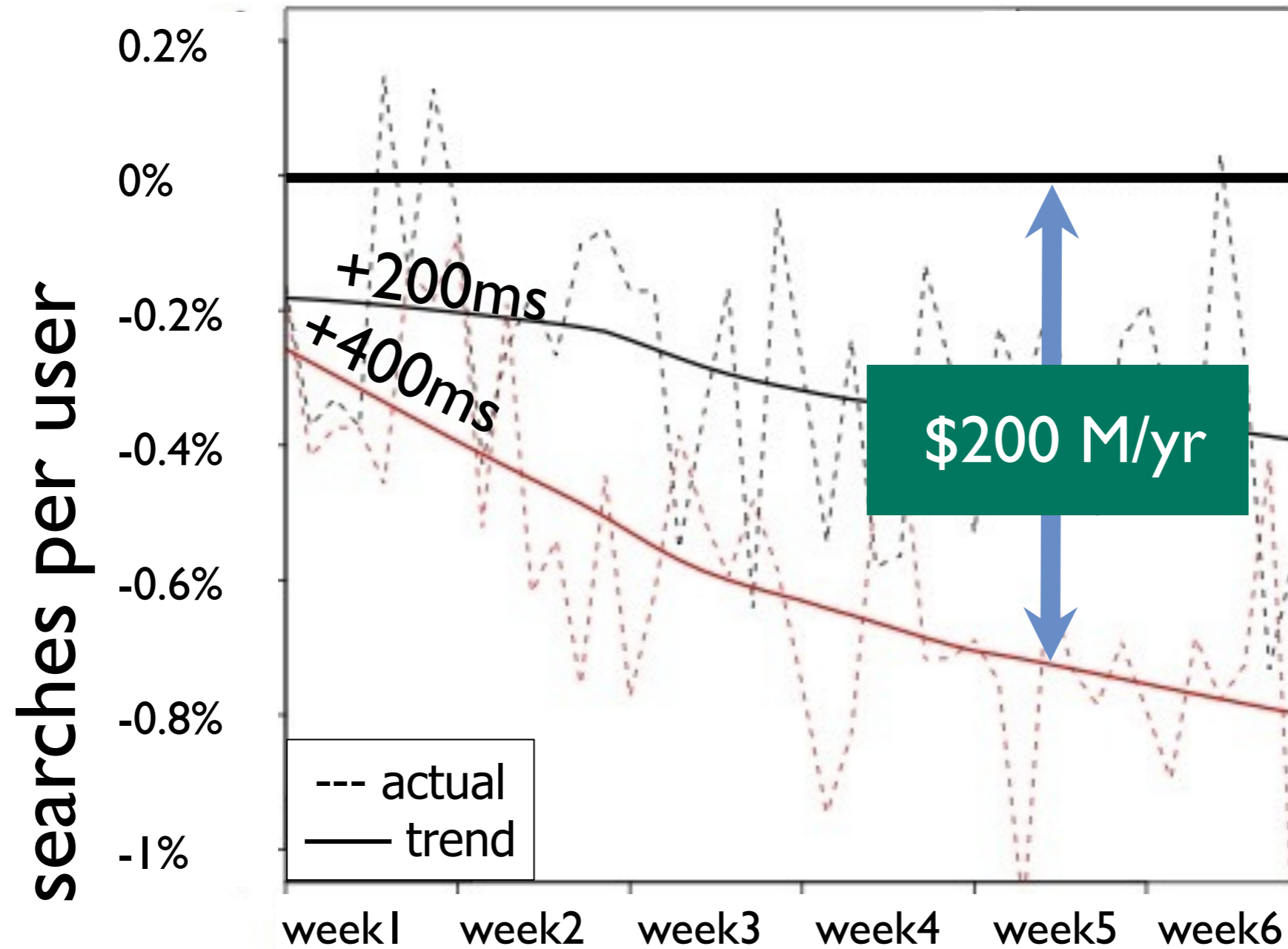


# What's a few hundred milliseconds?



[Jake Brutlag, Google, 2009]

# What's a few hundred milliseconds?



[Jake Brutlag, Google, 2009]

# Protocols cause delay

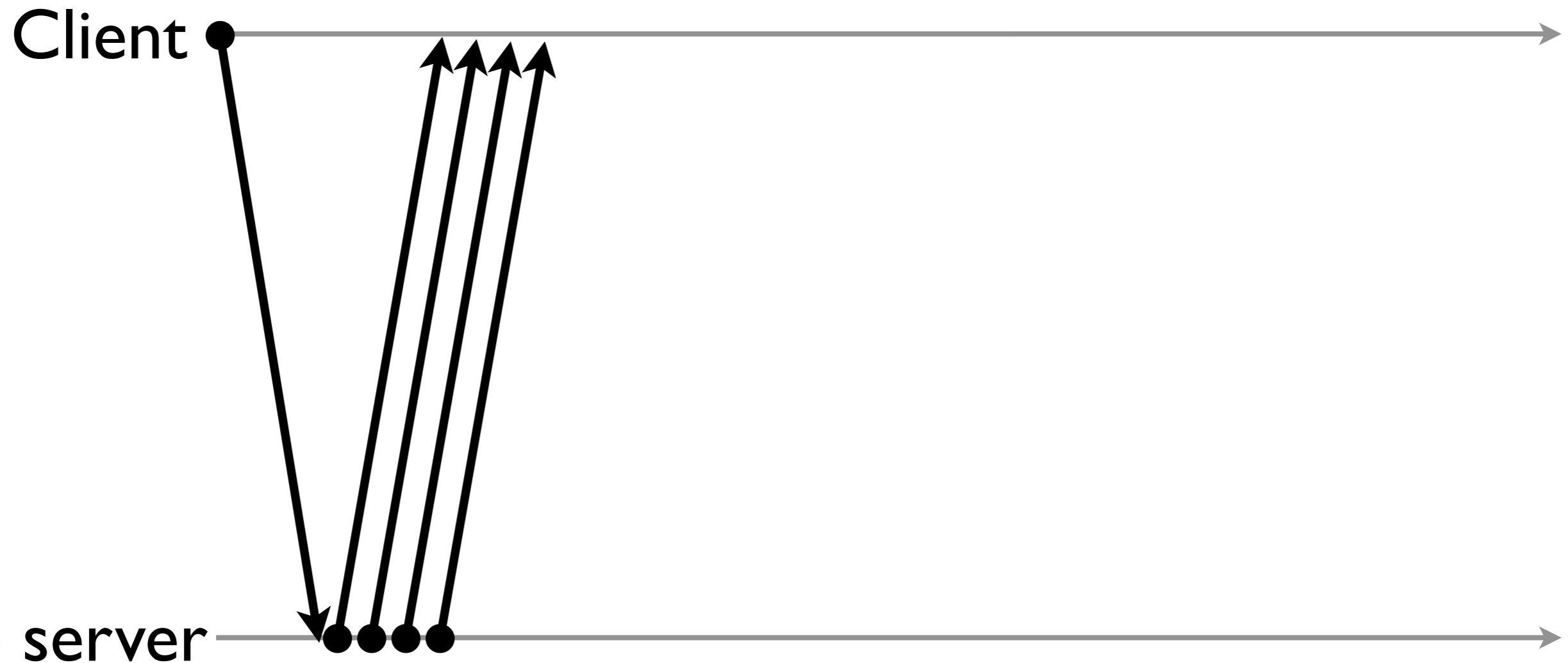
Client 

Web server 

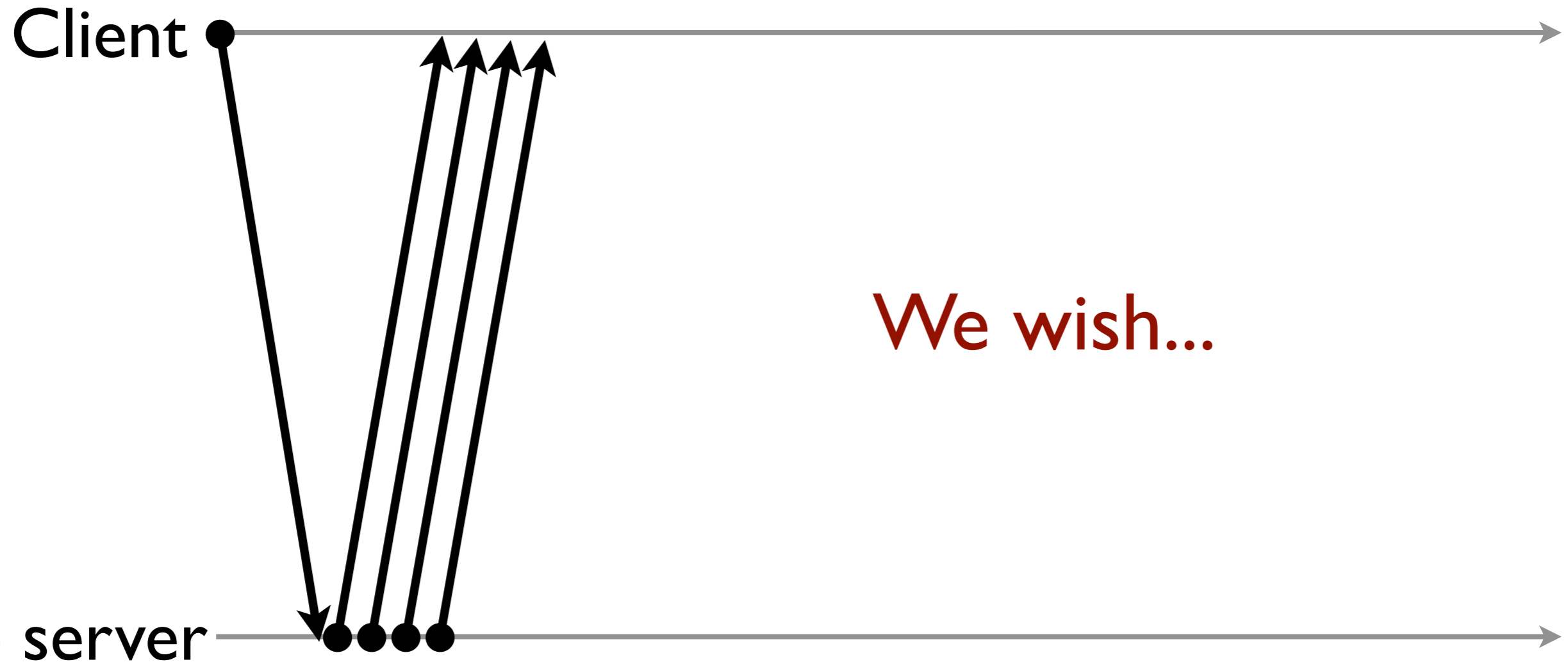
# Protocols cause delay



# Protocols cause delay



# Protocols cause delay



# Protocols cause delay

Client



Local DNS



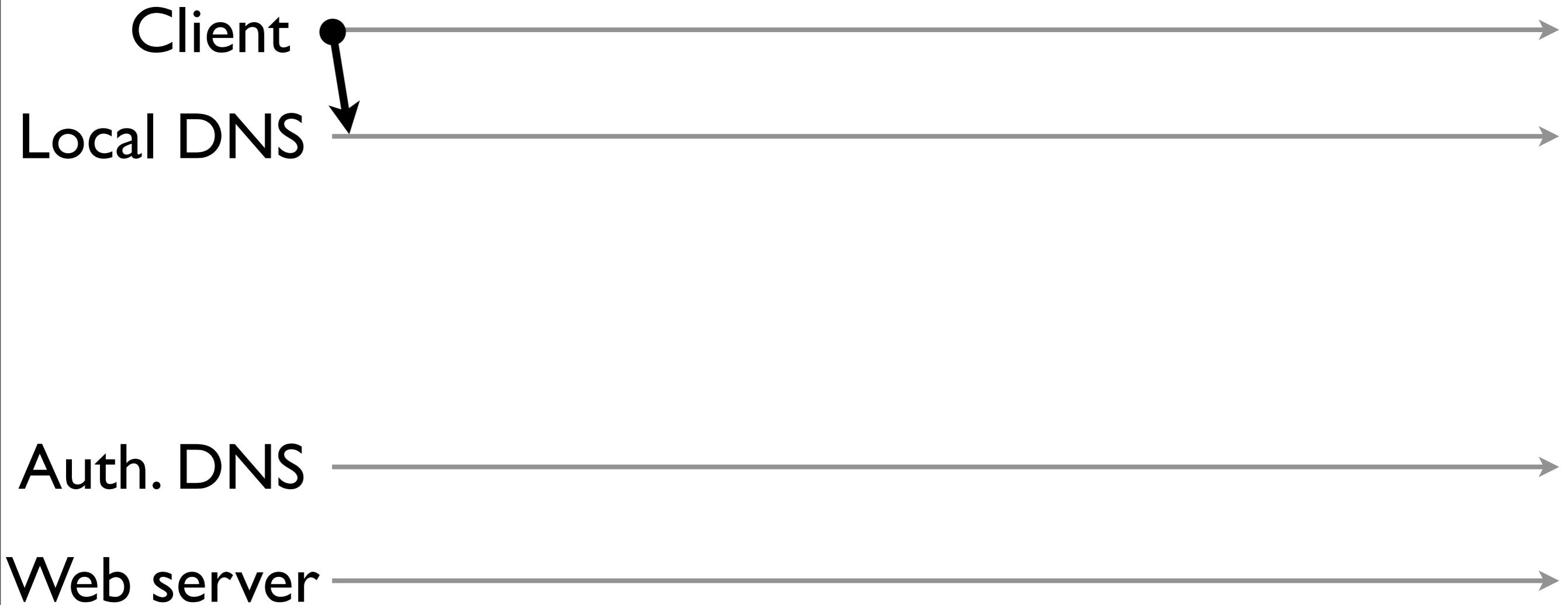
Auth. DNS



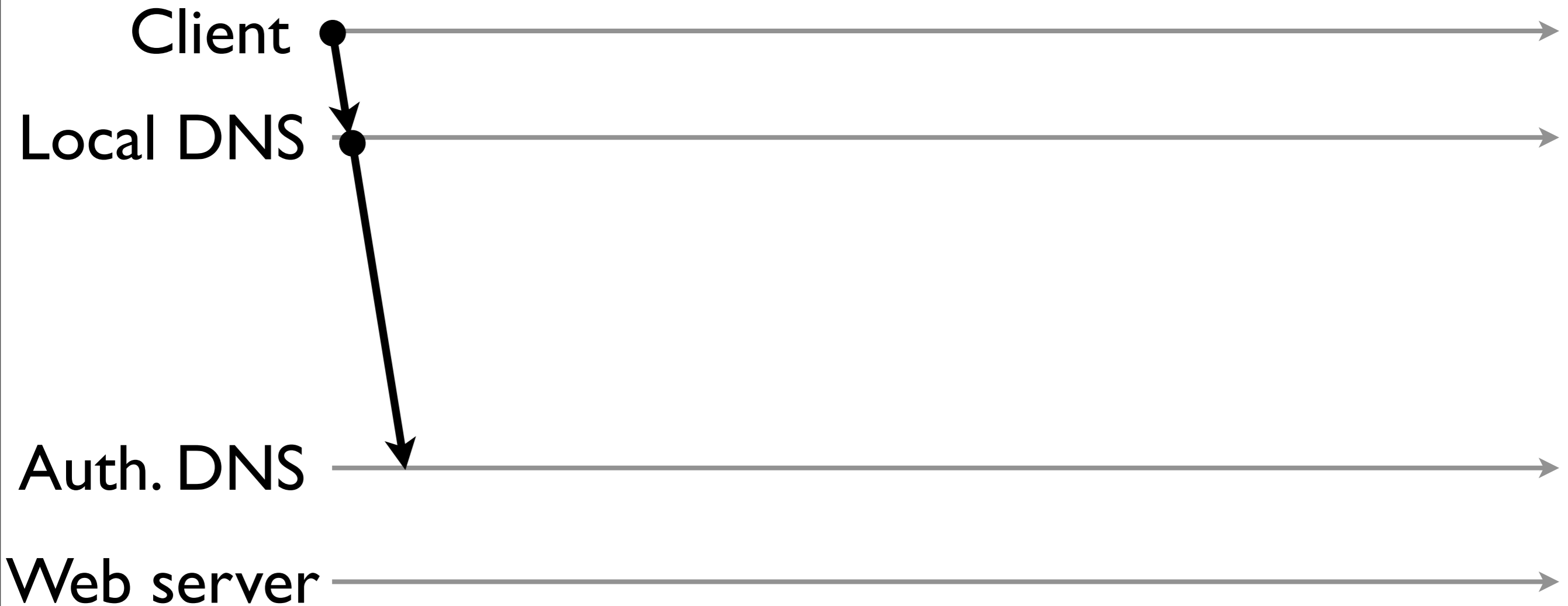
Web server



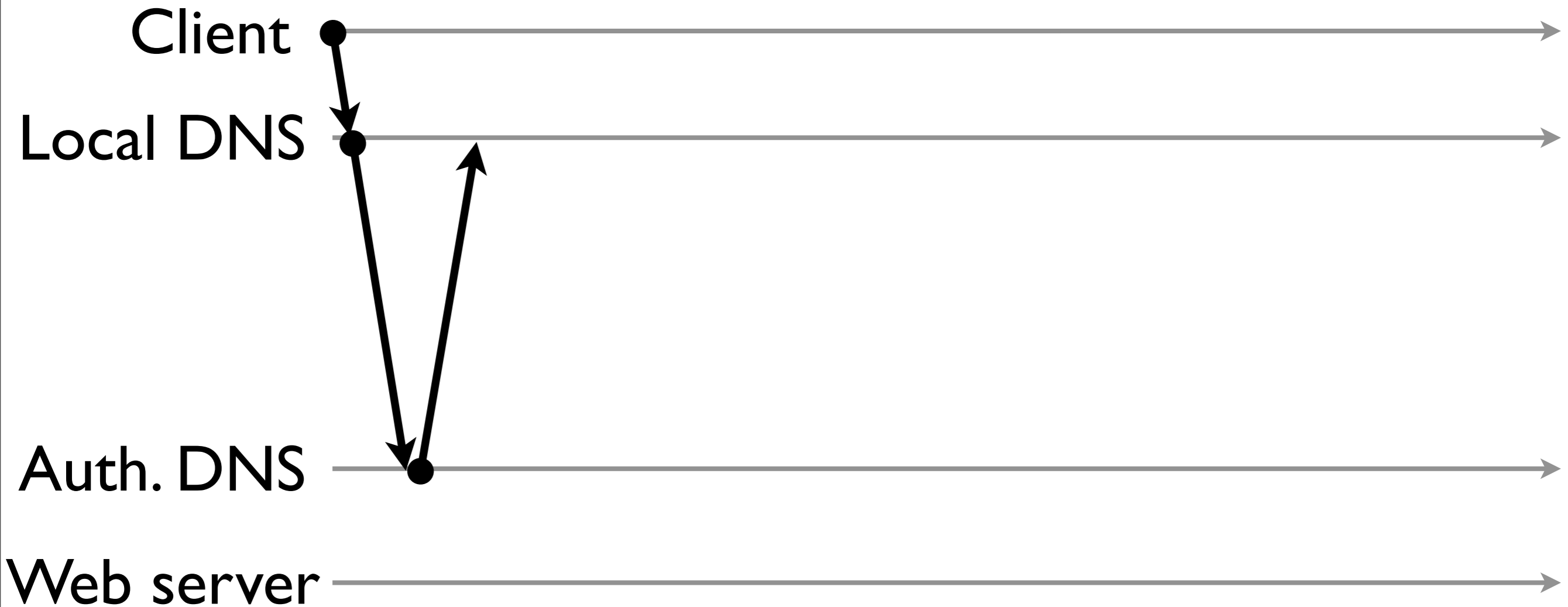
# Protocols cause delay



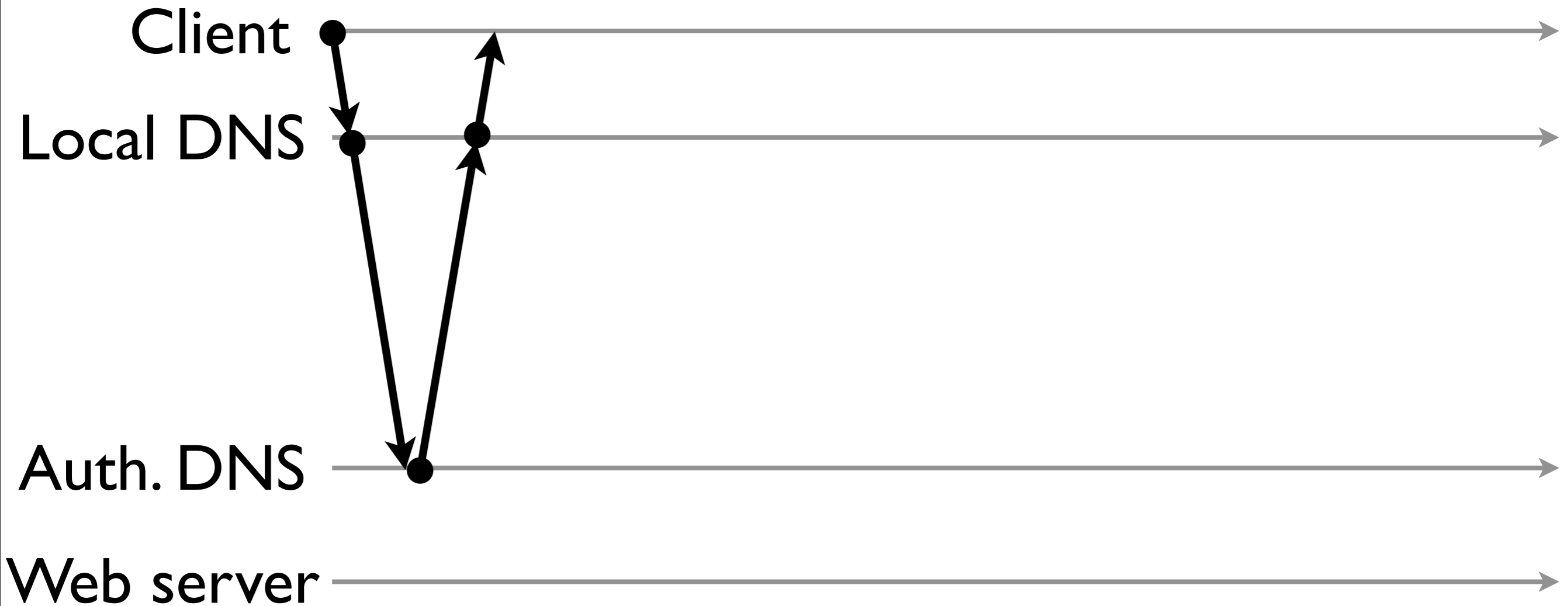
# Protocols cause delay



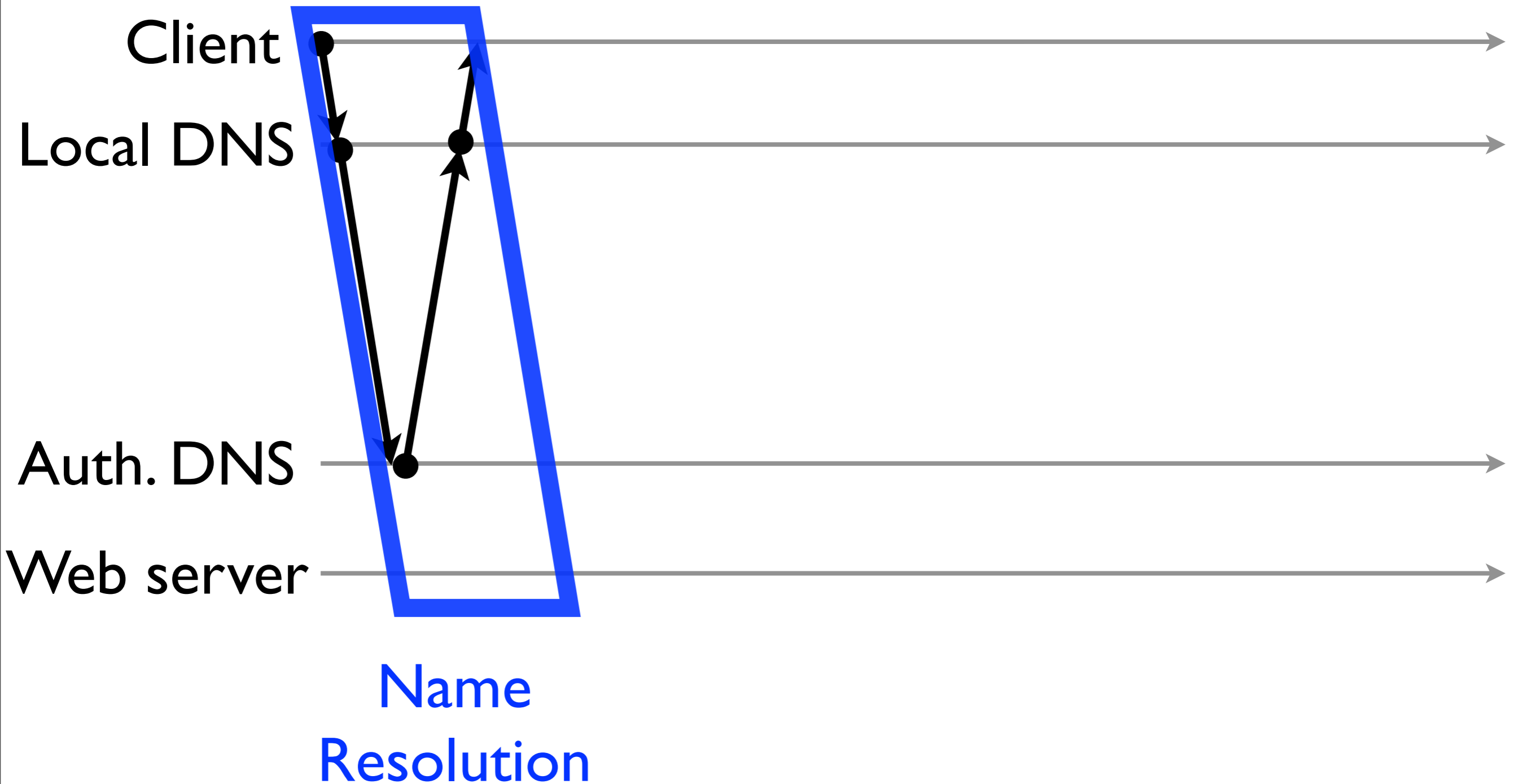
# Protocols cause delay



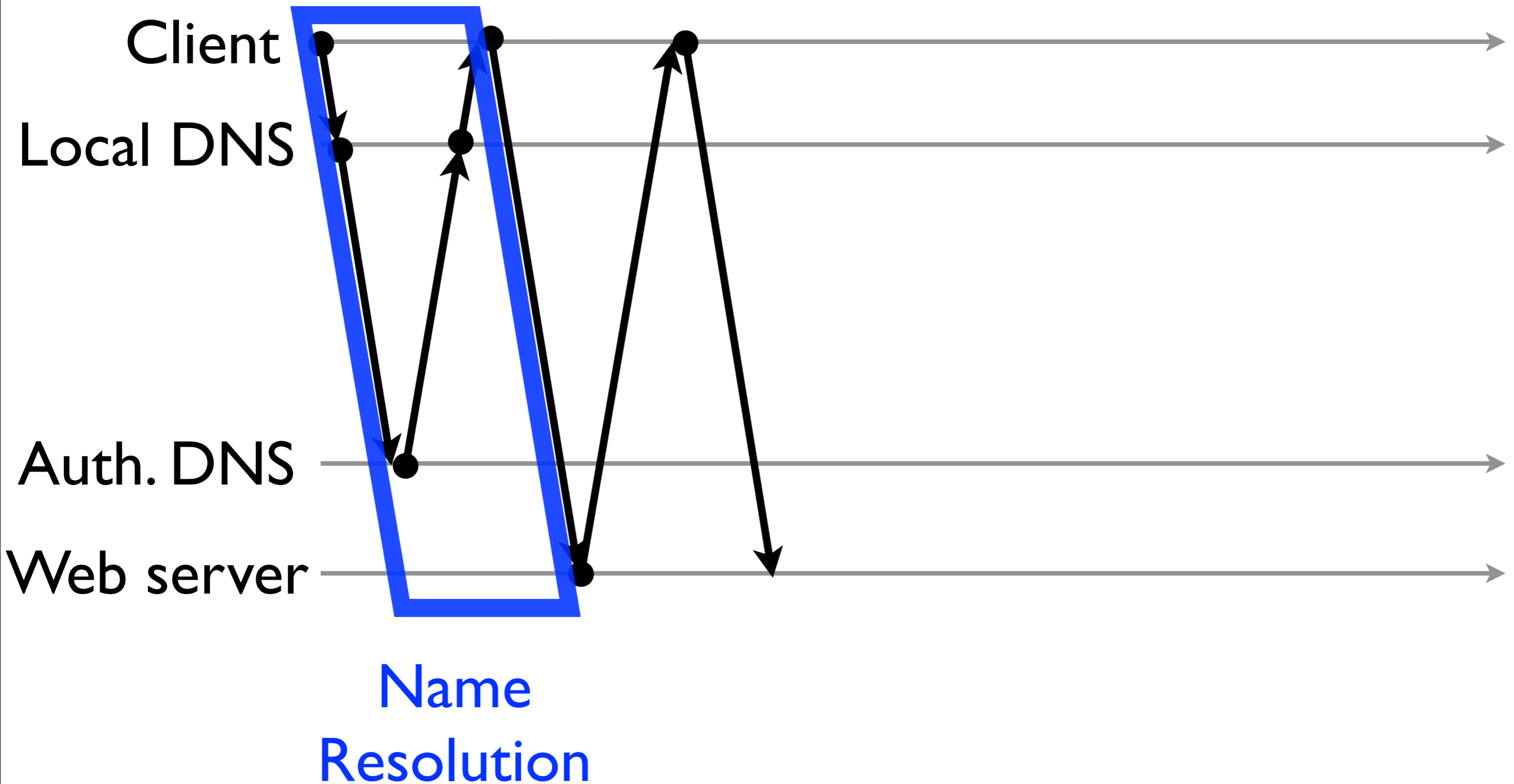
# Protocols cause delay



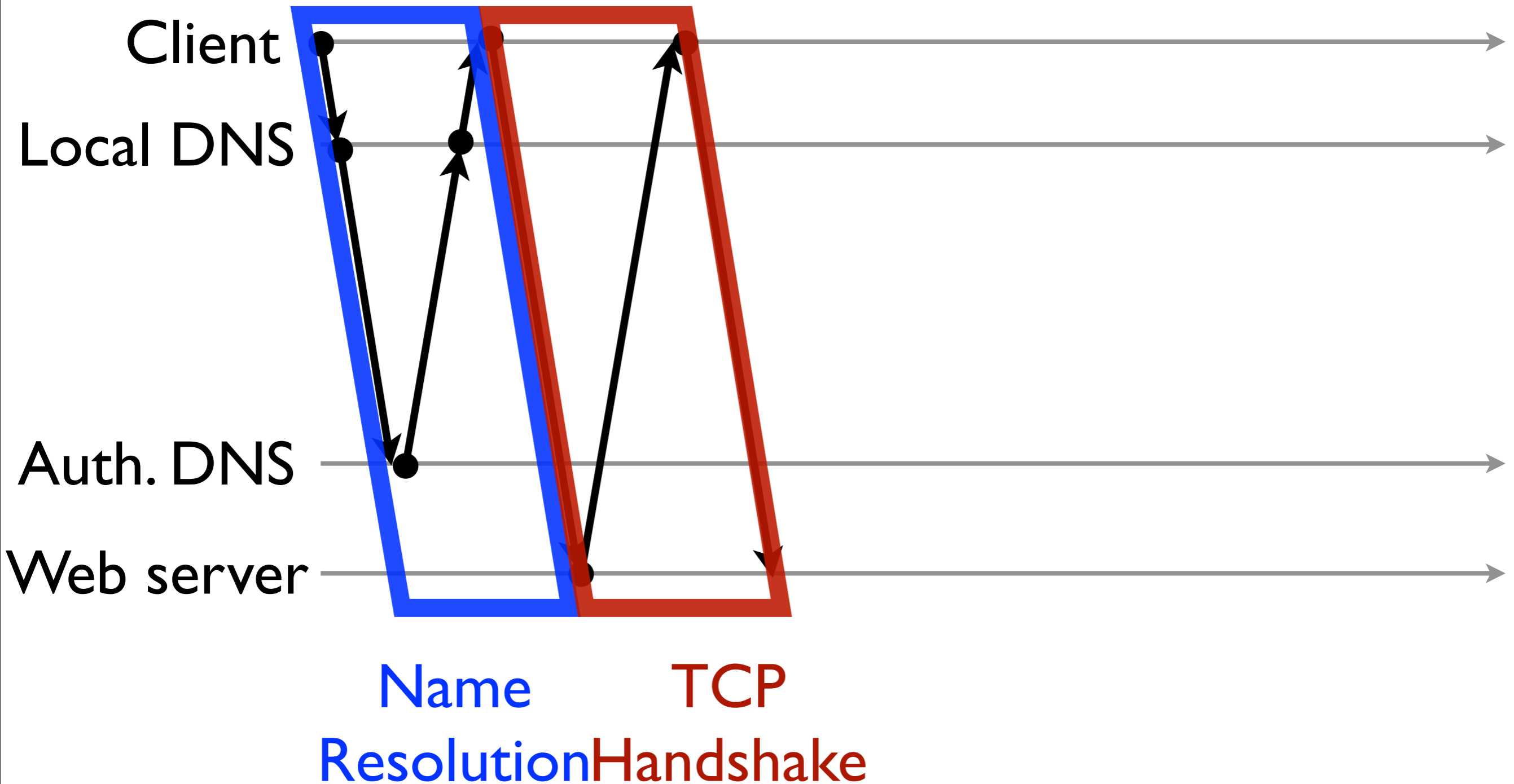
# Protocols cause delay



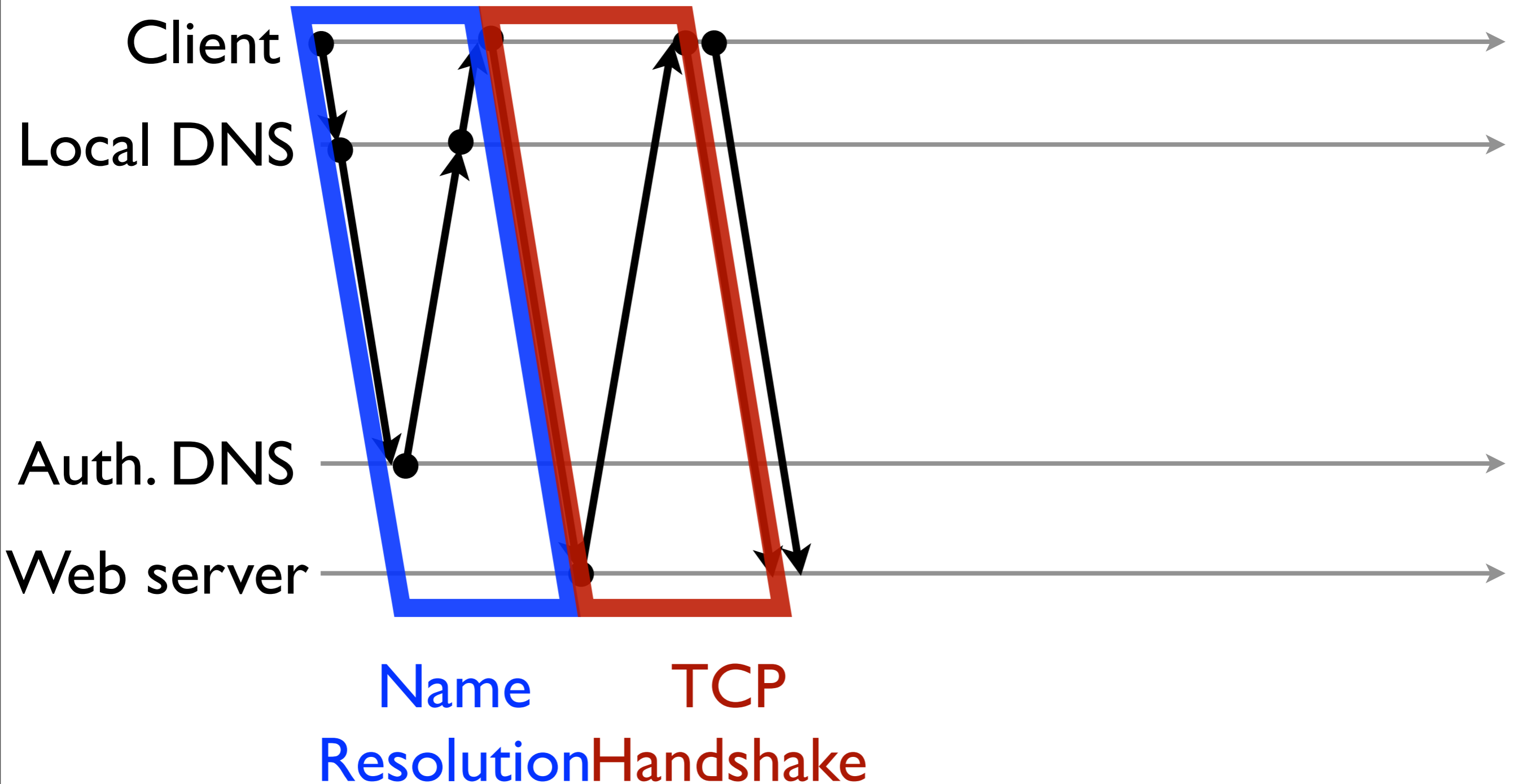
# Protocols cause delay



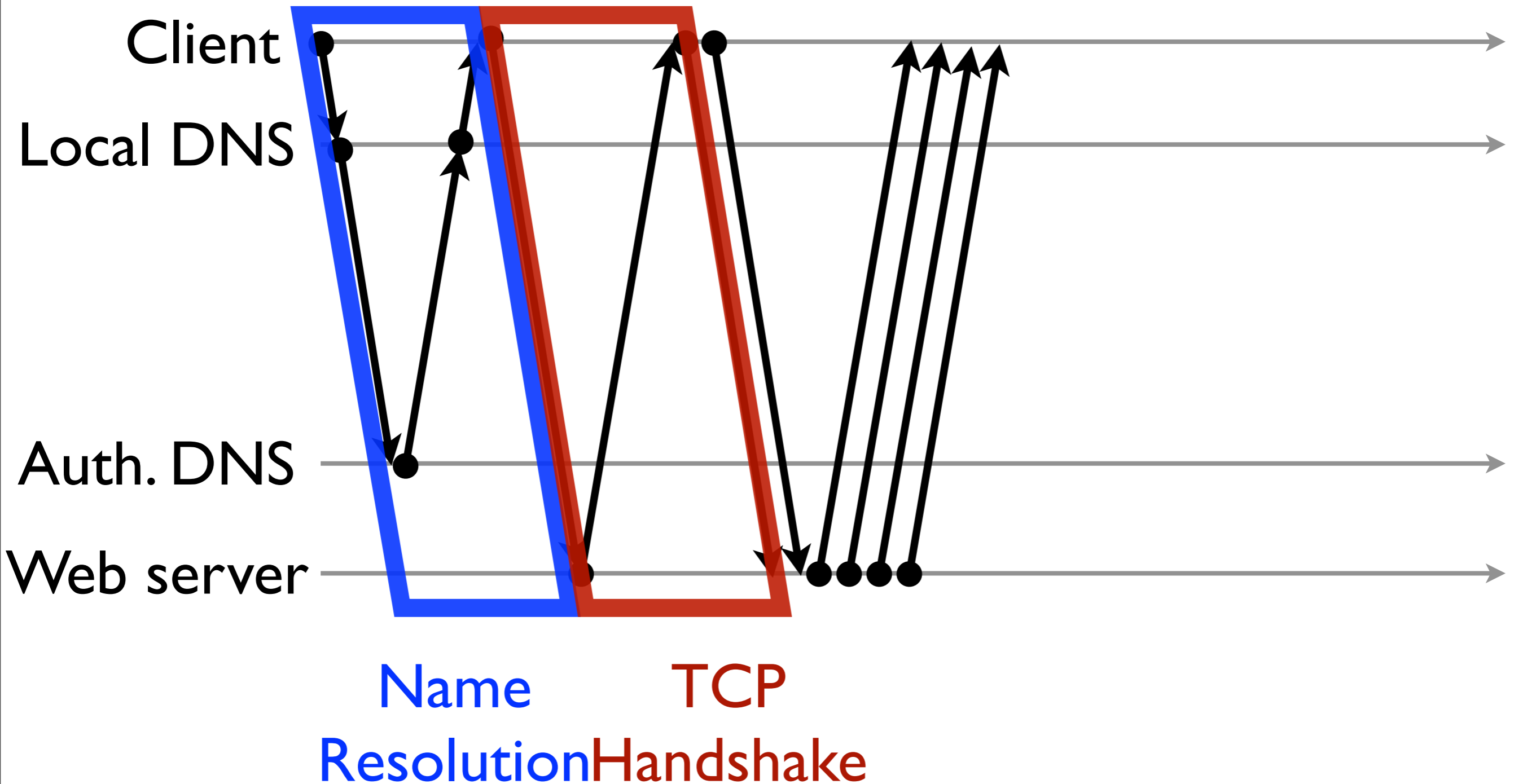
# Protocols cause delay



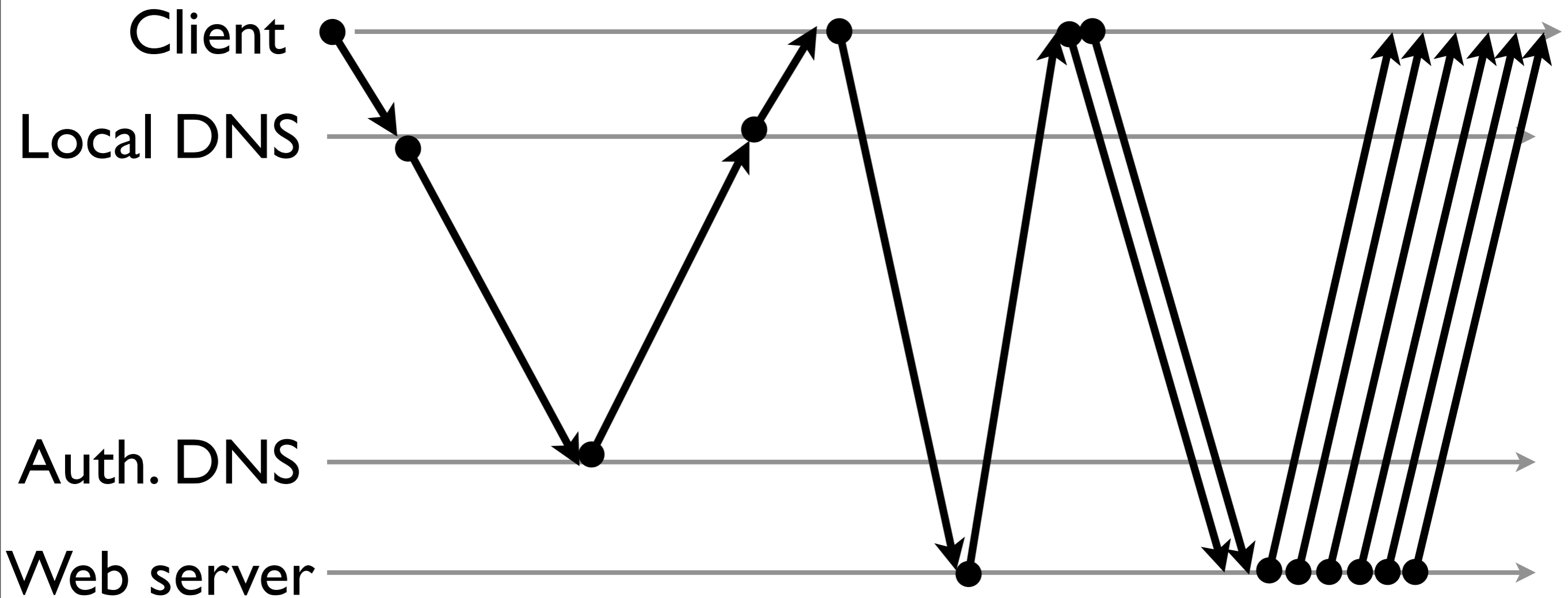
# Protocols cause delay



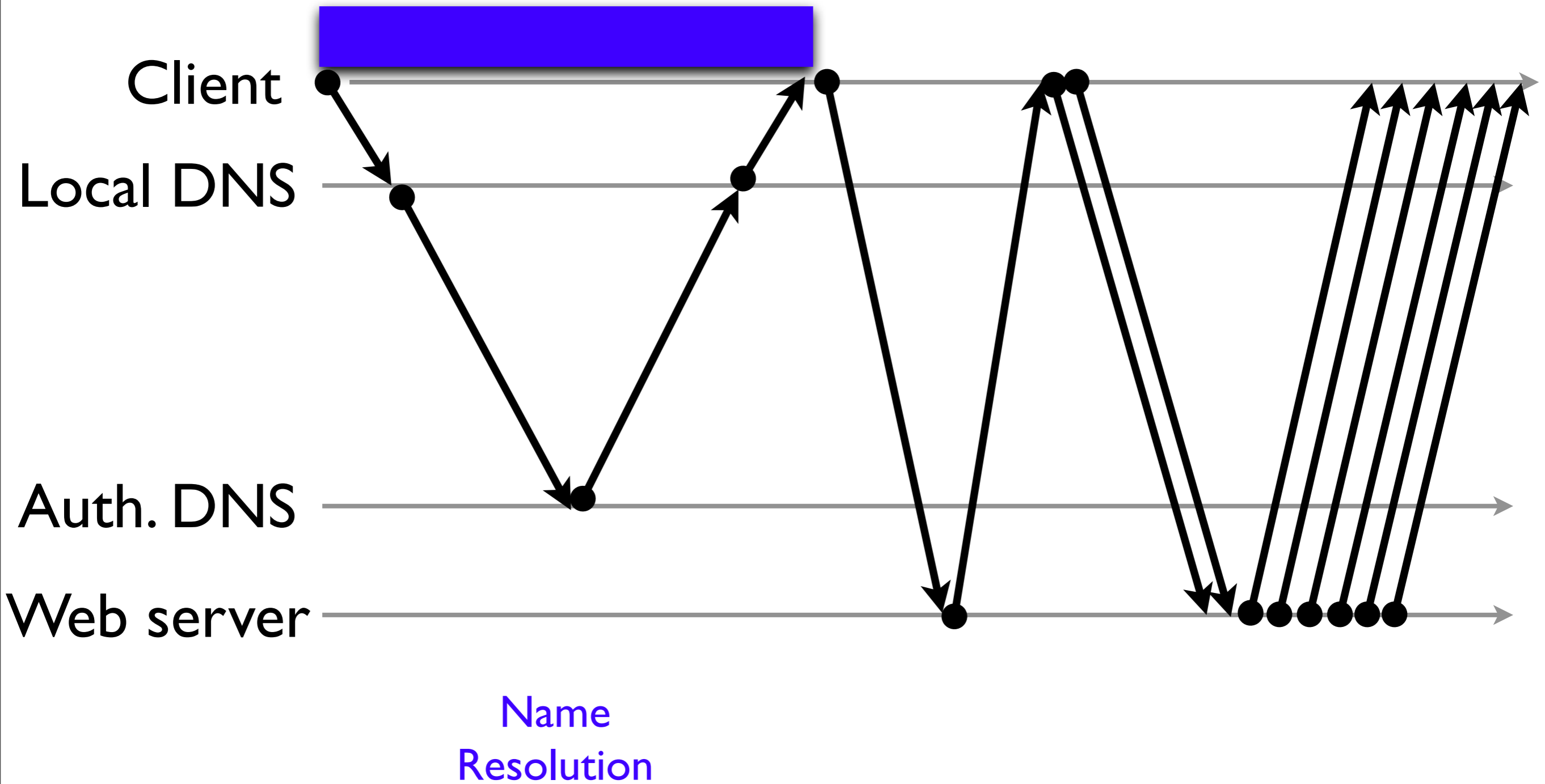
# Protocols cause delay



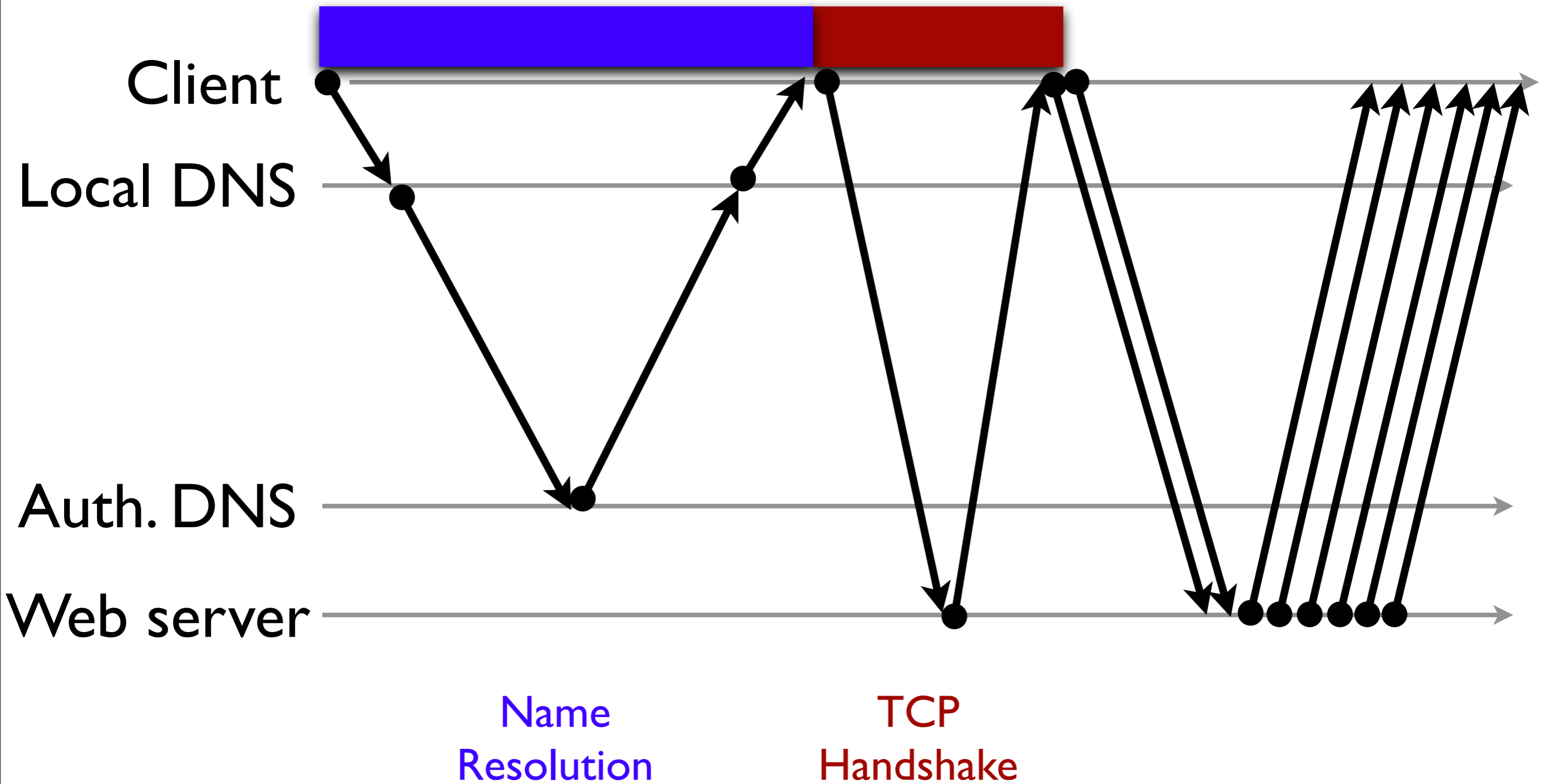
# Sources of Delay



# Sources of Delay



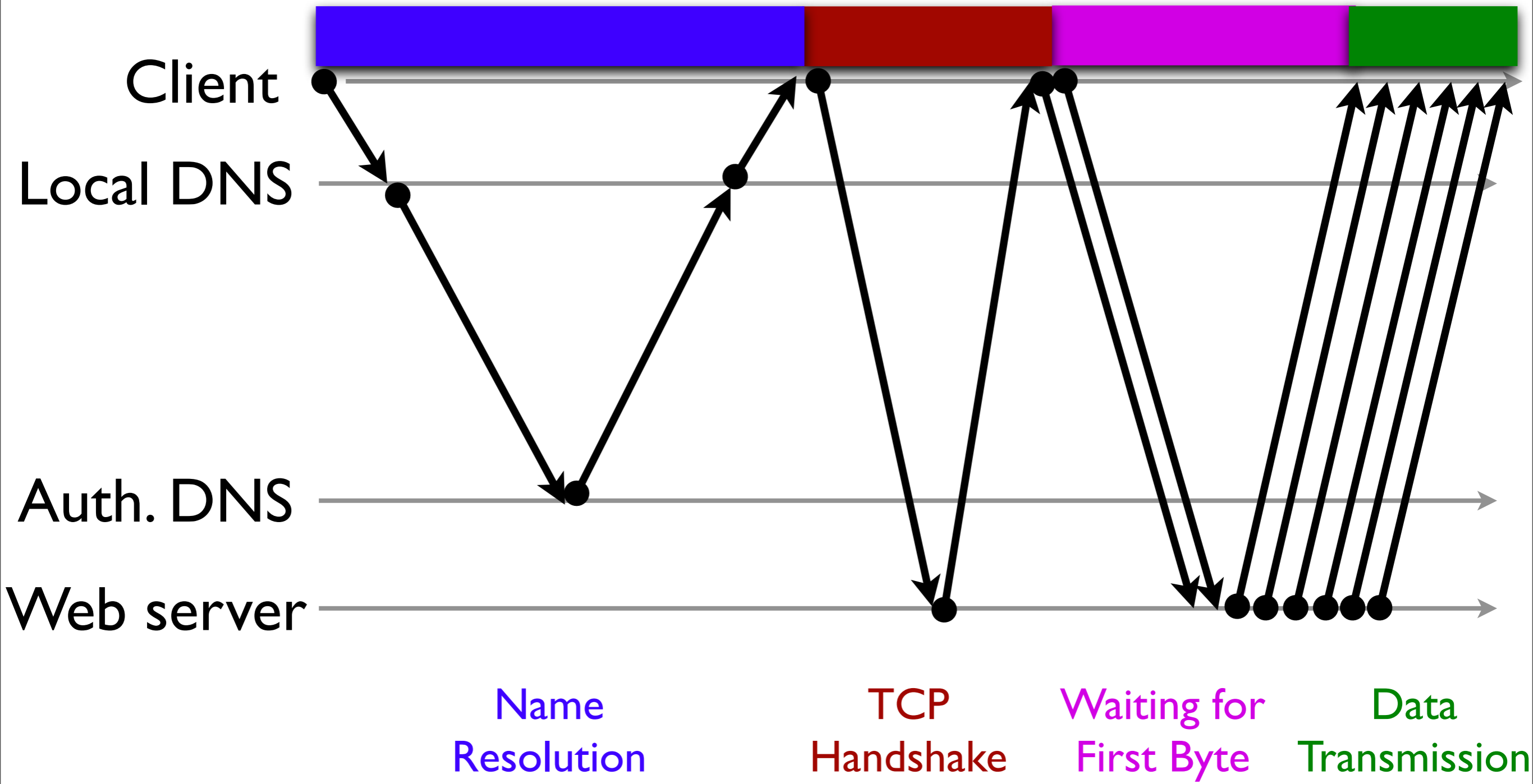
# Sources of Delay



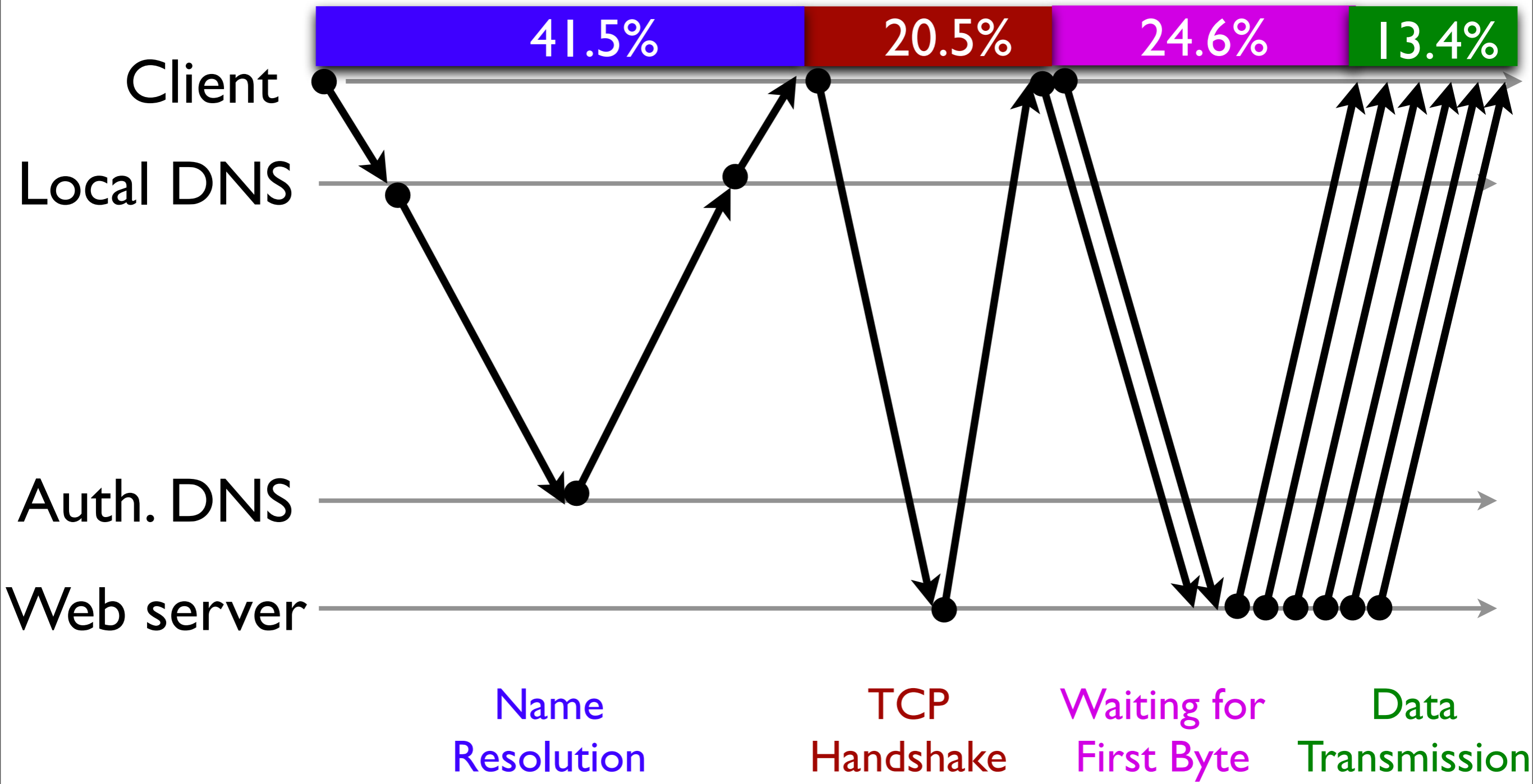
# Sources of Delay



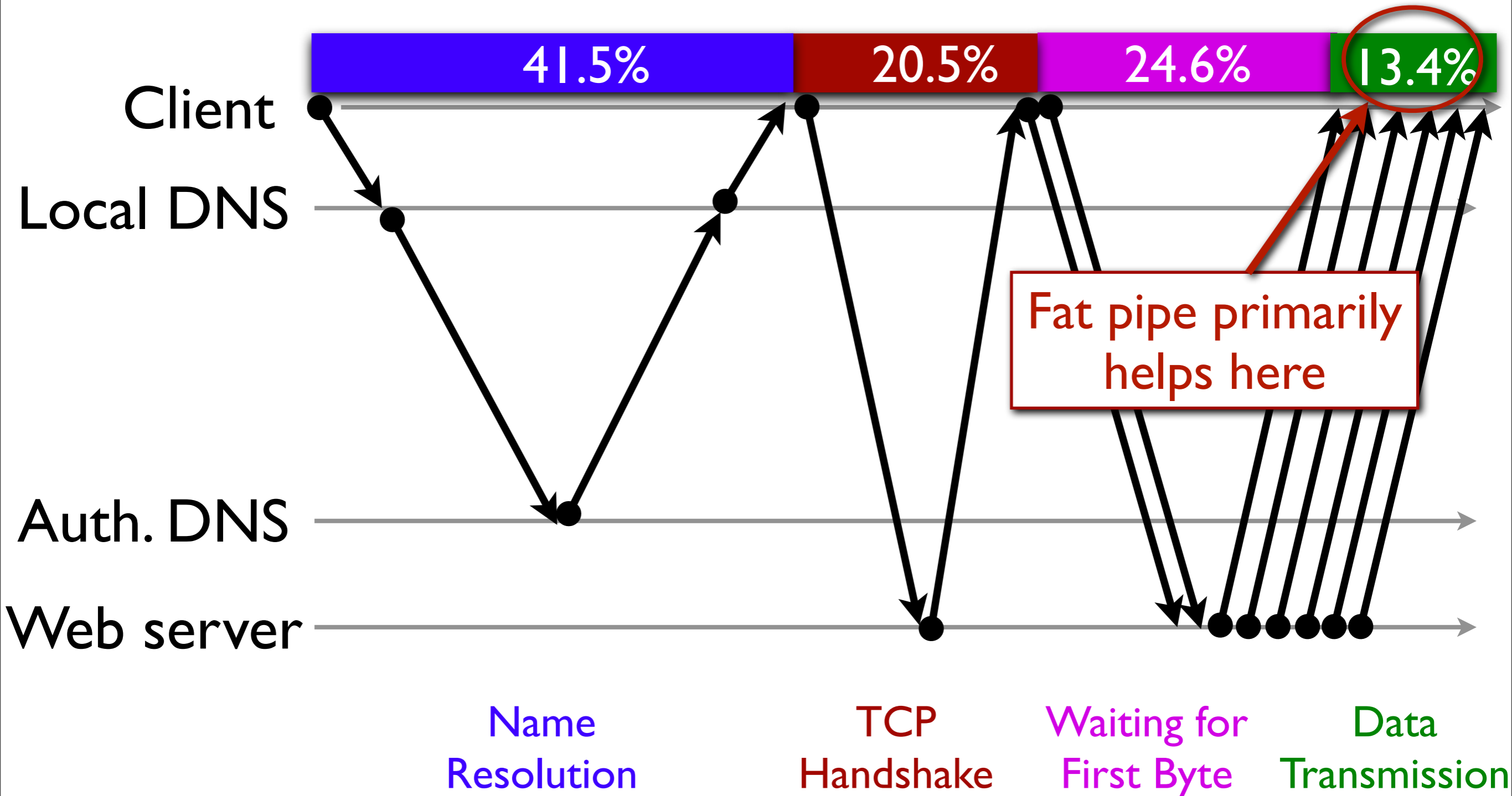
# Sources of Delay



# Sources of Delay



# Sources of Delay



# Our Design: ASAP

# Our Design: ASAP

**Goals:**

# Our Design: ASAP

## Goals:

- Reduce latency

# Our Design: ASAP

## Goals:

- Reduce latency
- Retain security and deployability

# Our Design: ASAP

## Goals:

- Reduce latency
- Retain security and deployability

Key idea 1: Piggyback connection setup on name resolution

# Our Design: ASAP

## Goals:

- Reduce latency
- Retain security and deployability

## Key idea 1: Piggyback connection setup on name resolution

- Add connection information into name lookup message.

# Our Design: ASAP

## Goals:

- Reduce latency
- Retain security and deployability

## Key idea 1: Piggyback connection setup on name resolution

- Add connection information into name lookup message.

## Key idea 2: Bypass handshaking

# Our Design: ASAP

## Goals:

- Reduce latency
- Retain security and deployability

## Key idea 1: Piggyback connection setup on name resolution

- Add connection information into name lookup message.

## Key idea 2: Bypass handshaking

- Servers respond to the request with data directly.

# Our Design: ASAP

**Client**



**Local DNS**



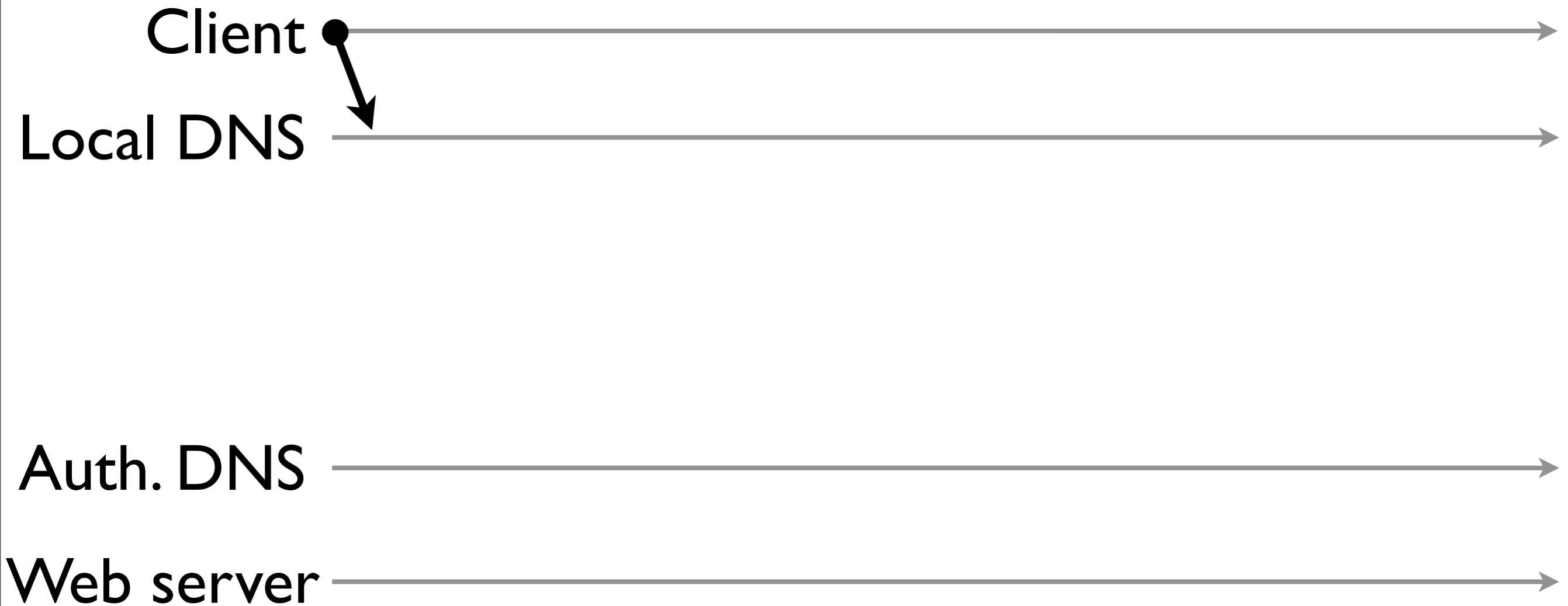
**Auth. DNS**



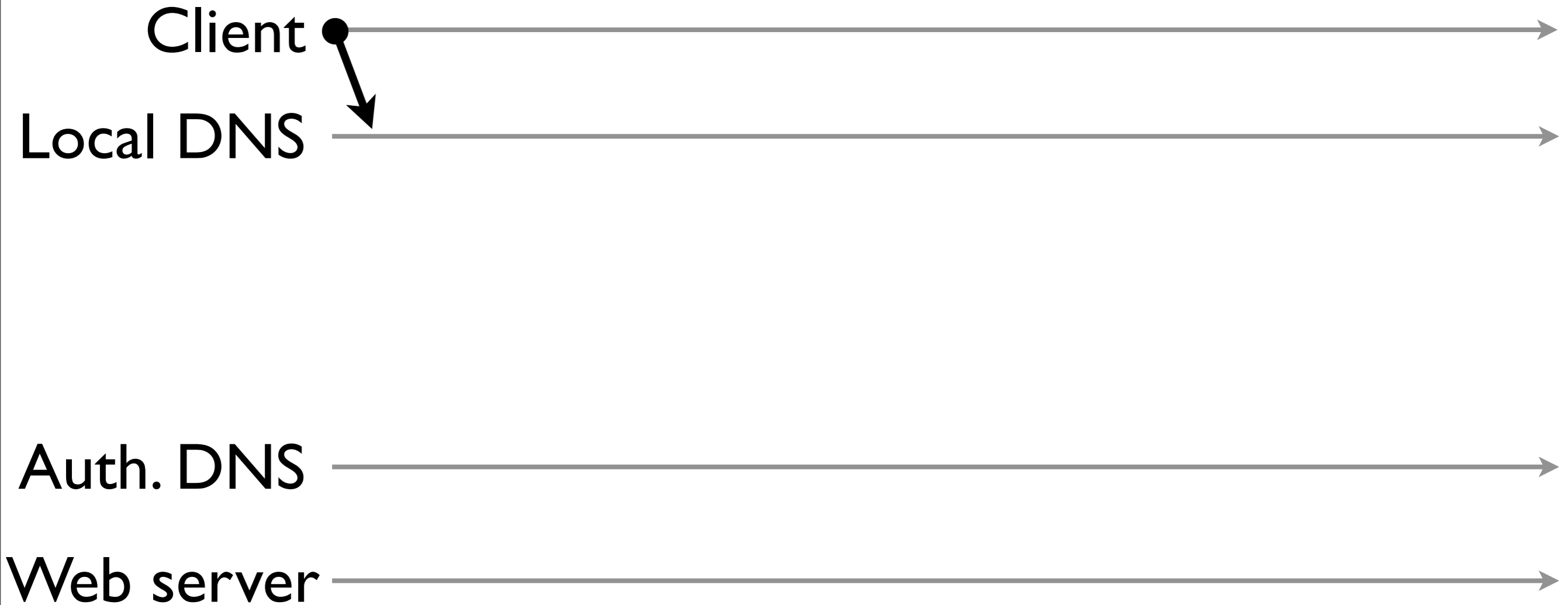
**Web server**



# Our Design: ASAP

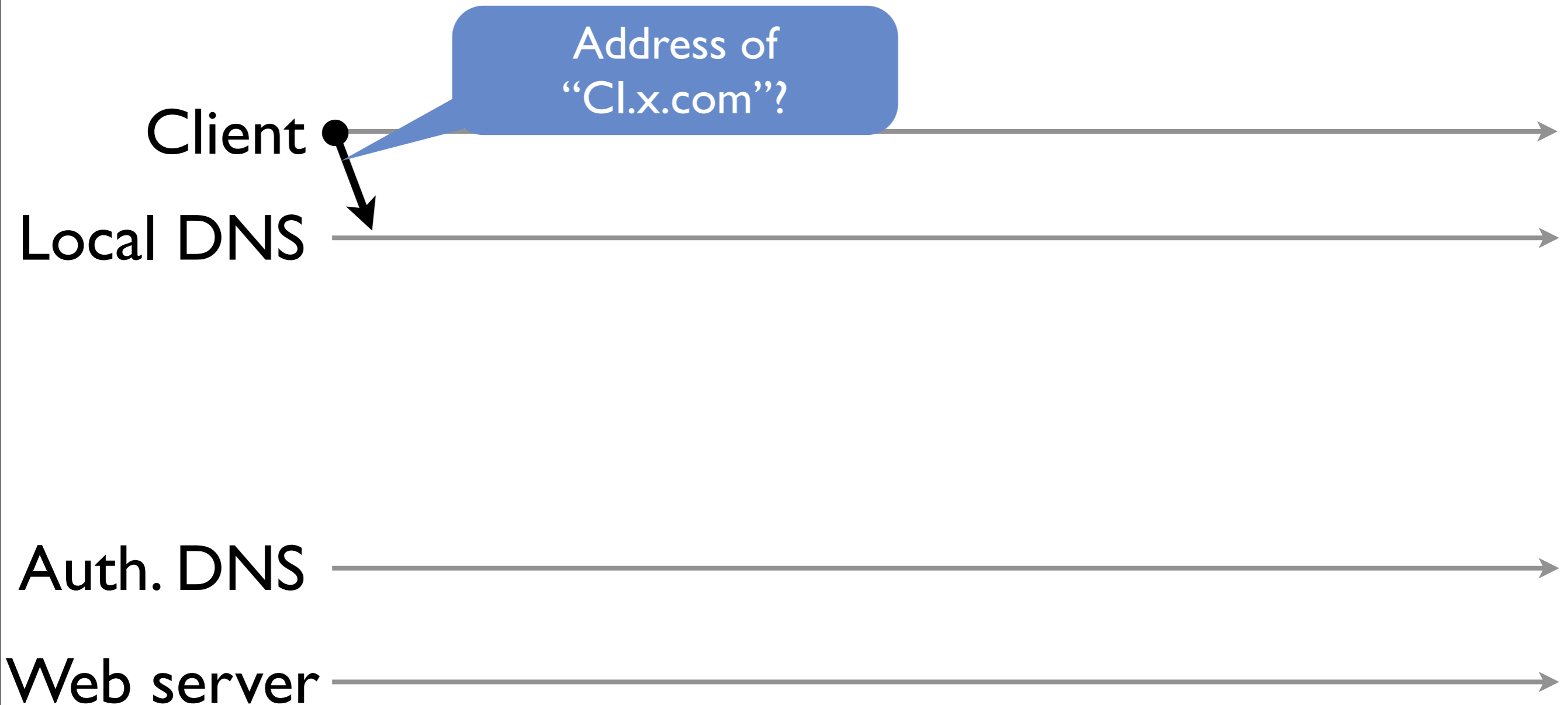


# Our Design: ASAP



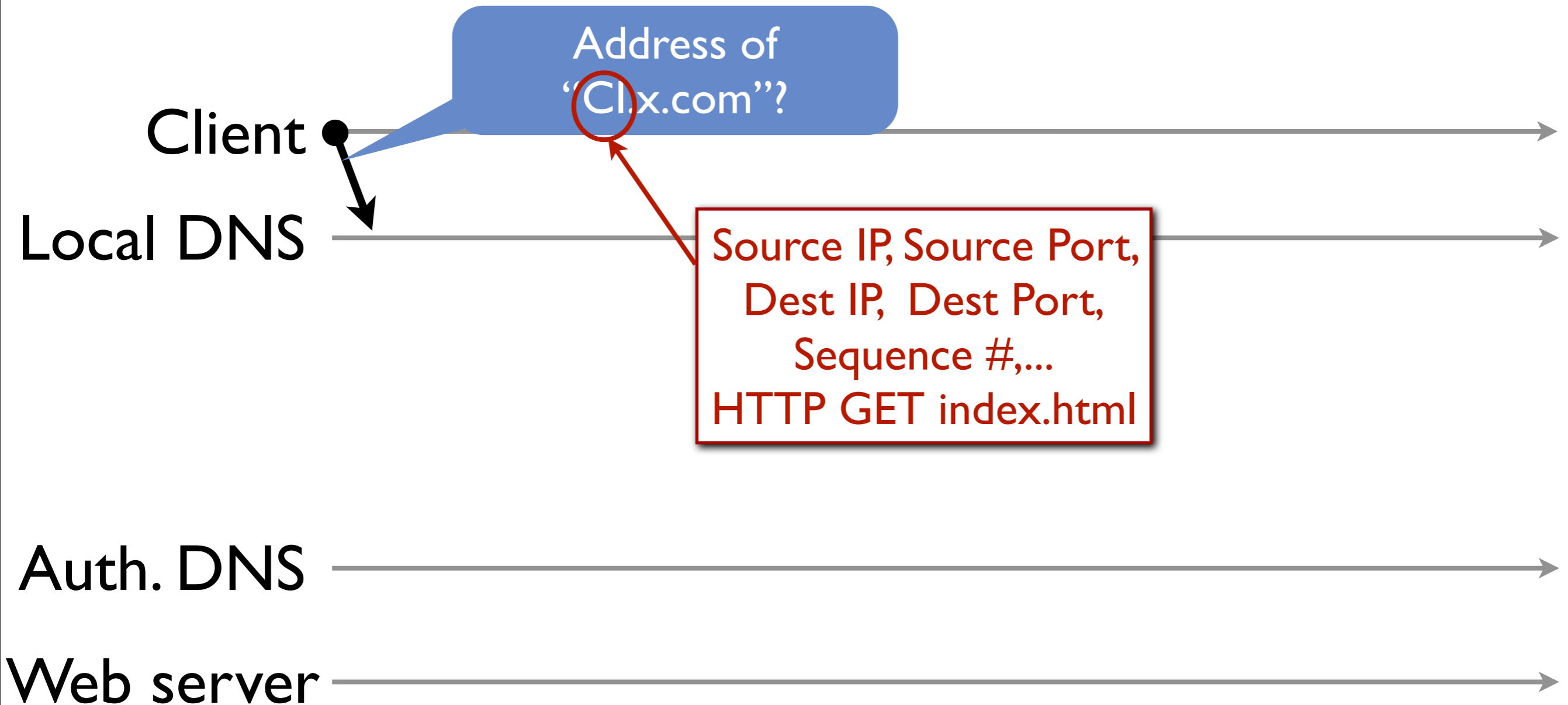
*Cl: connection information*

# Our Design: ASAP



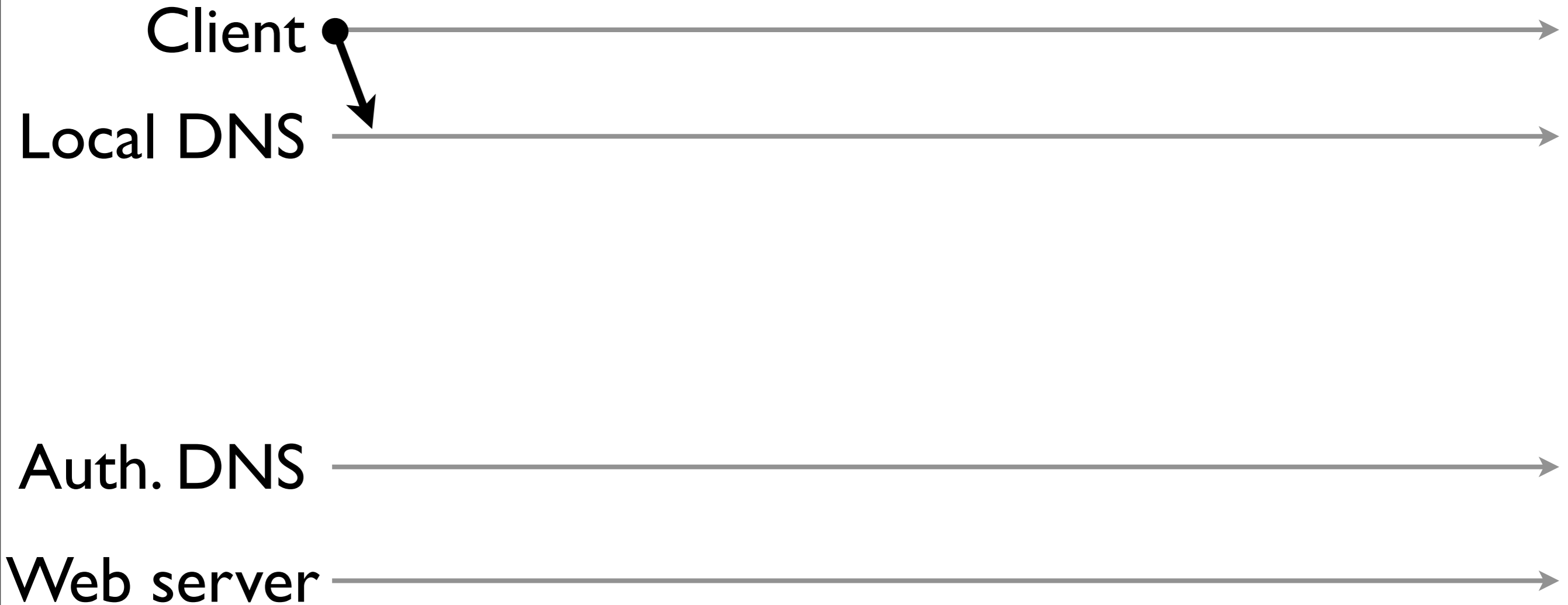
*Cl: connection information*

# Our Design: ASAP



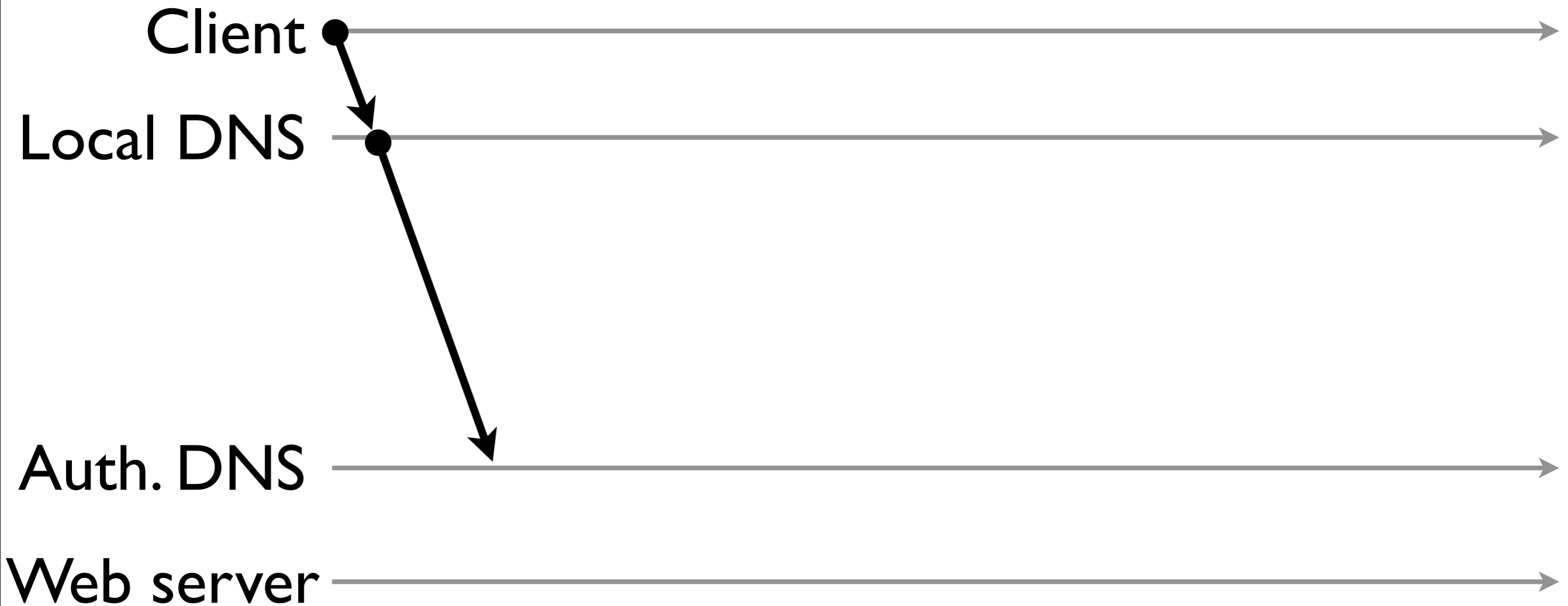
*Cl: connection information*

# Our Design: ASAP



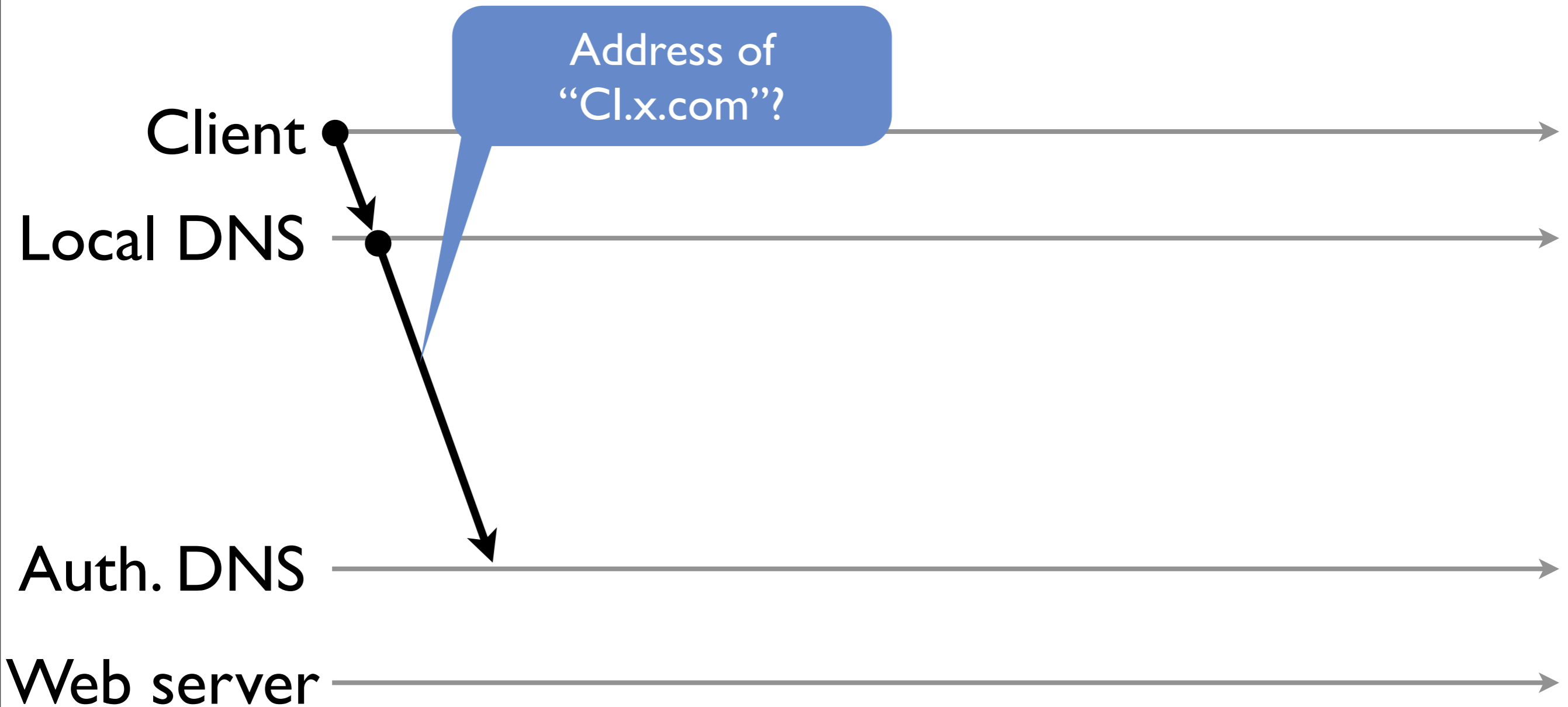
*Cl: connection information*

# Our Design: ASAP



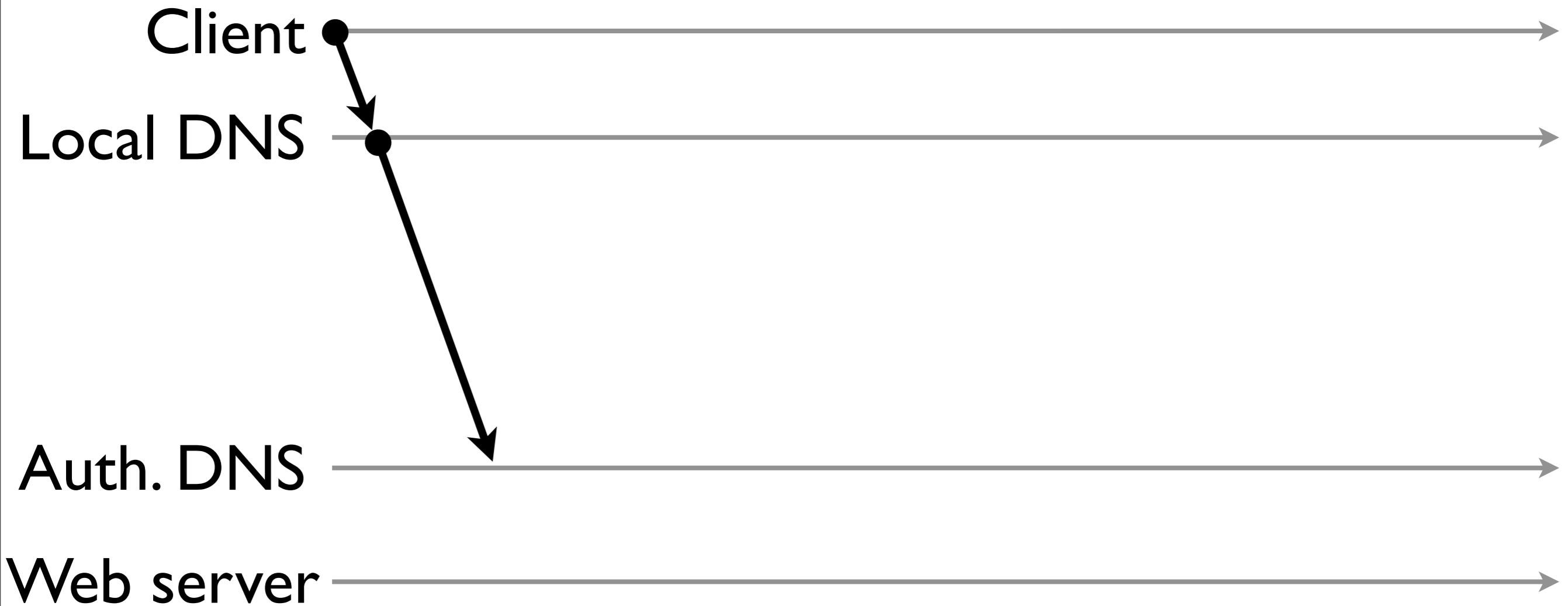
*Cl: connection information*

# Our Design: ASAP



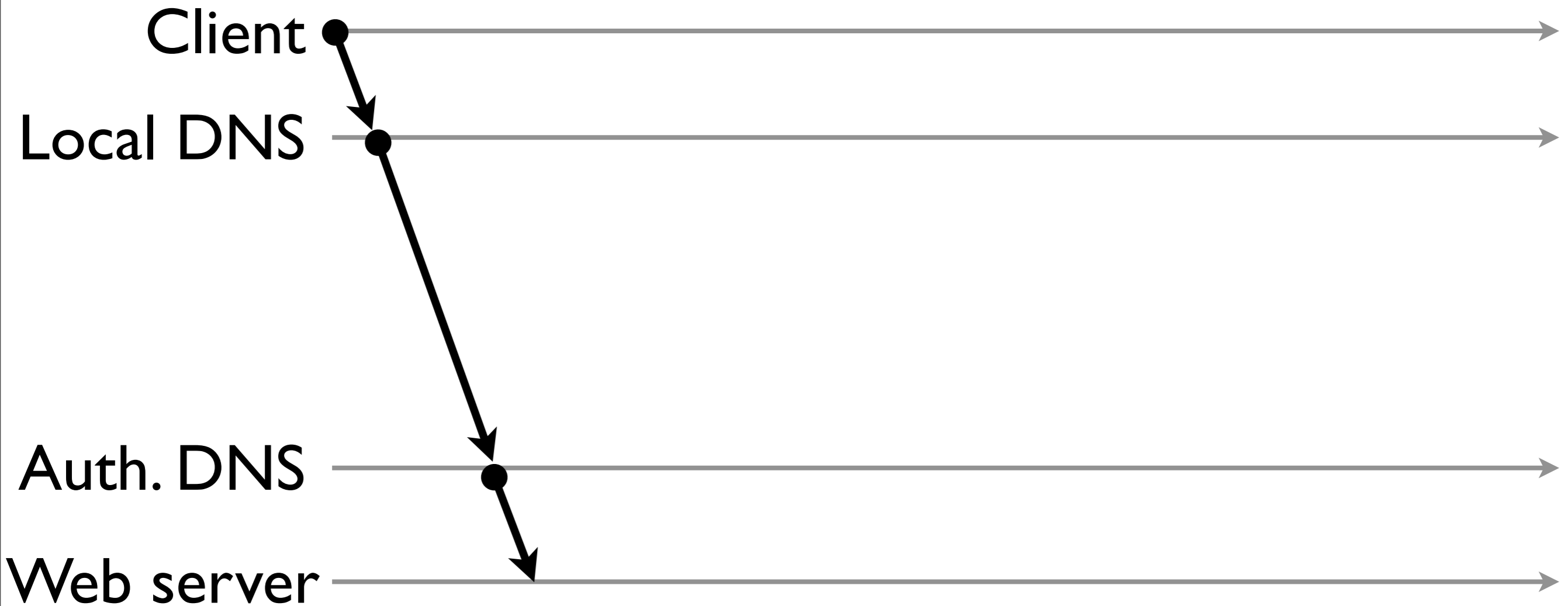
*Cl: connection information*

# Our Design: ASAP



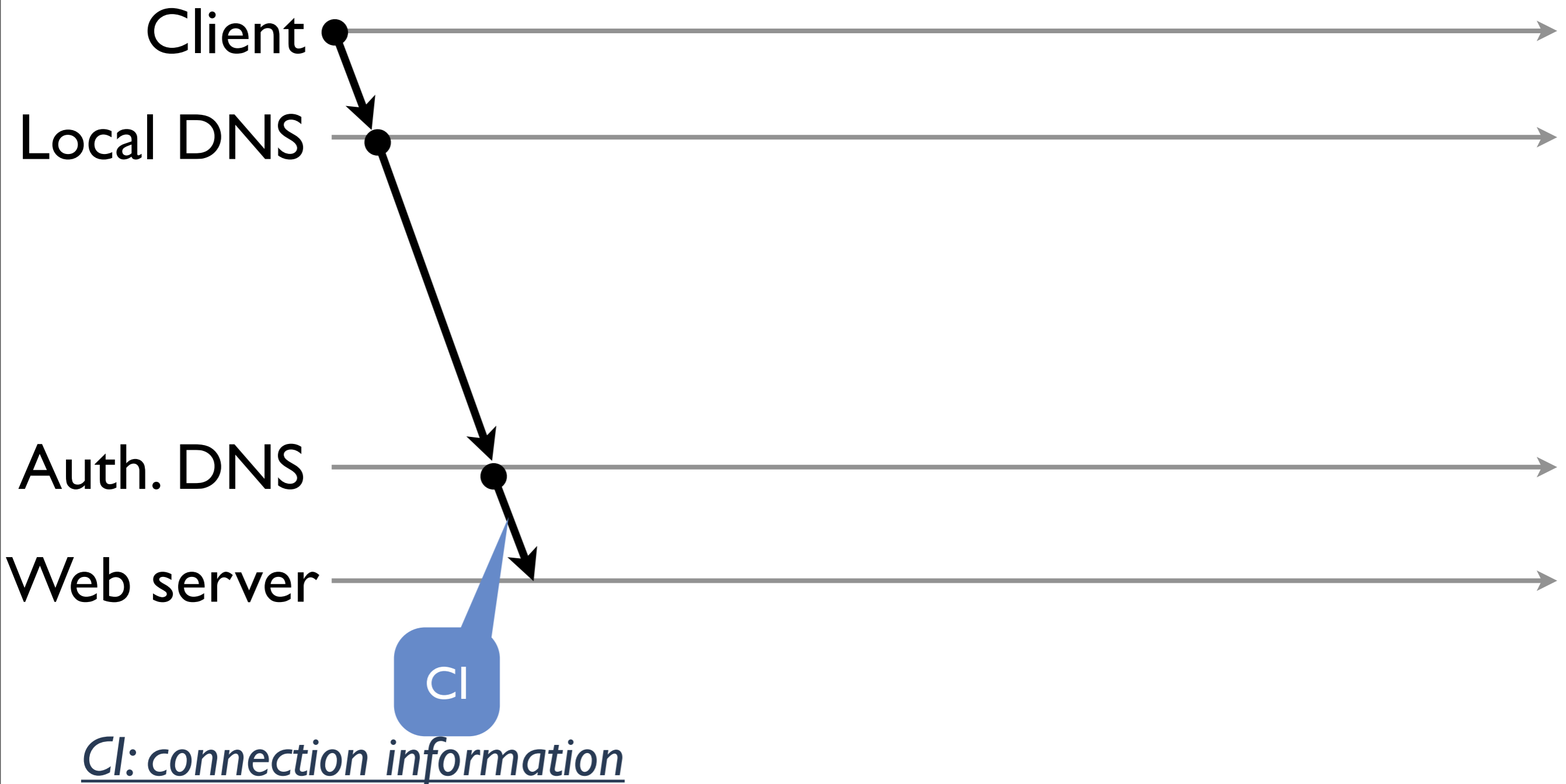
*Cl: connection information*

# Our Design: ASAP

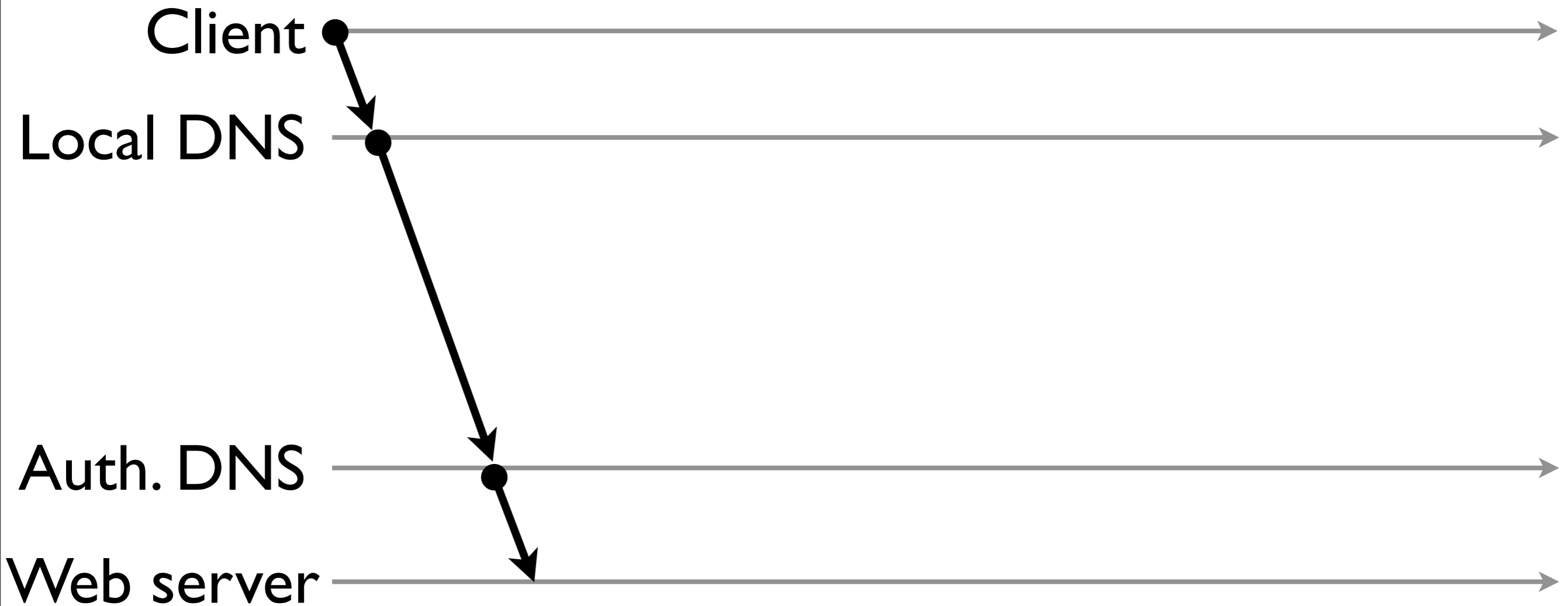


*Cl: connection information*

# Our Design: ASAP

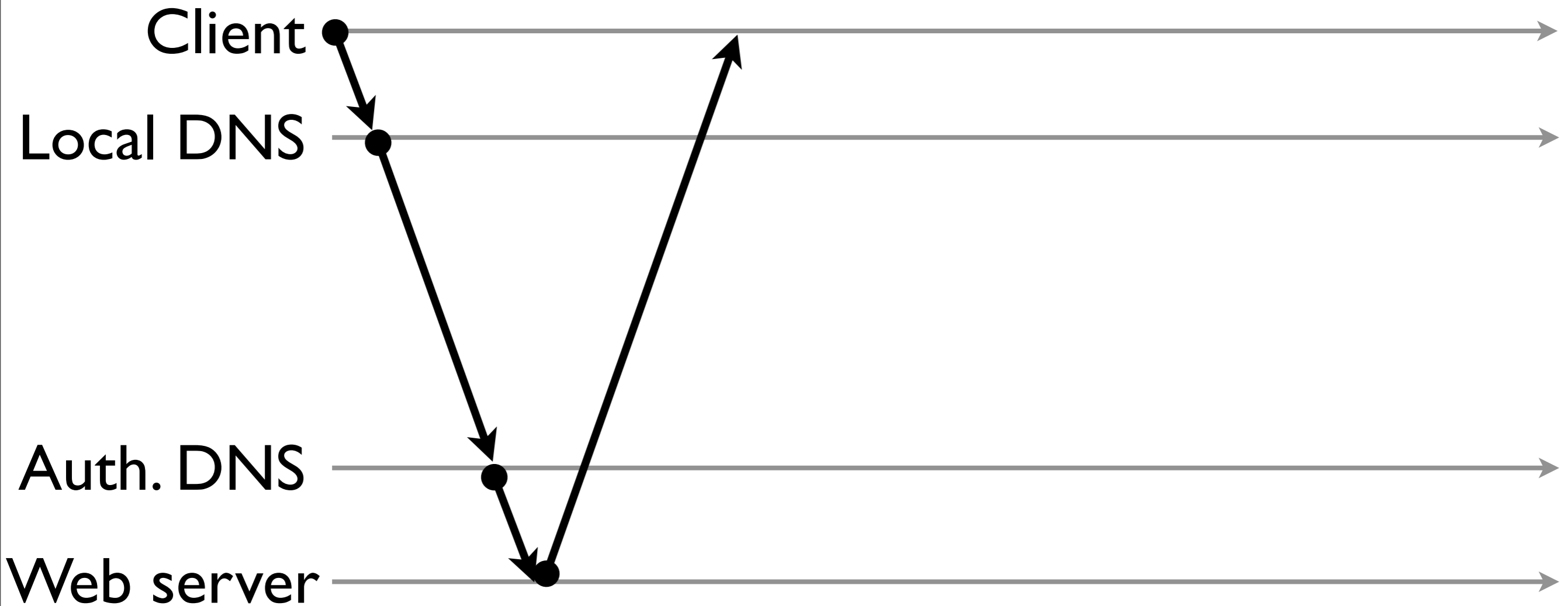


# Our Design: ASAP



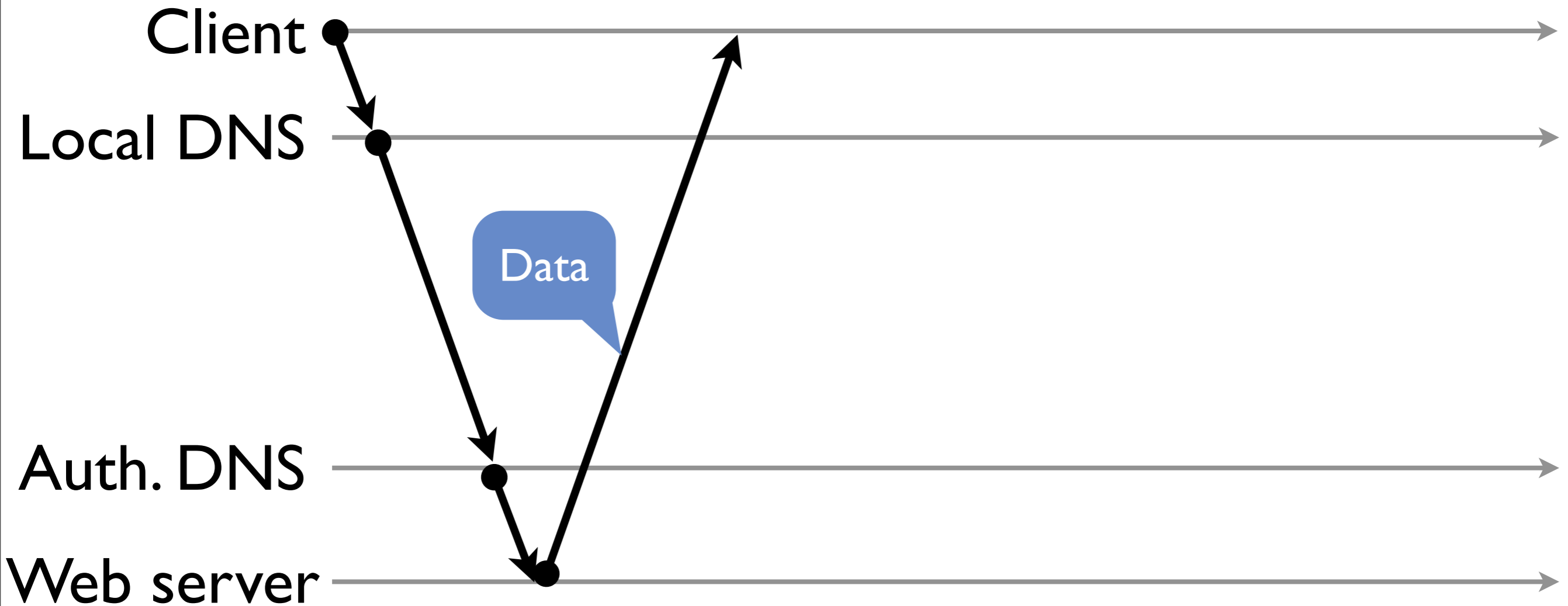
*Cl: connection information*

# Our Design: ASAP



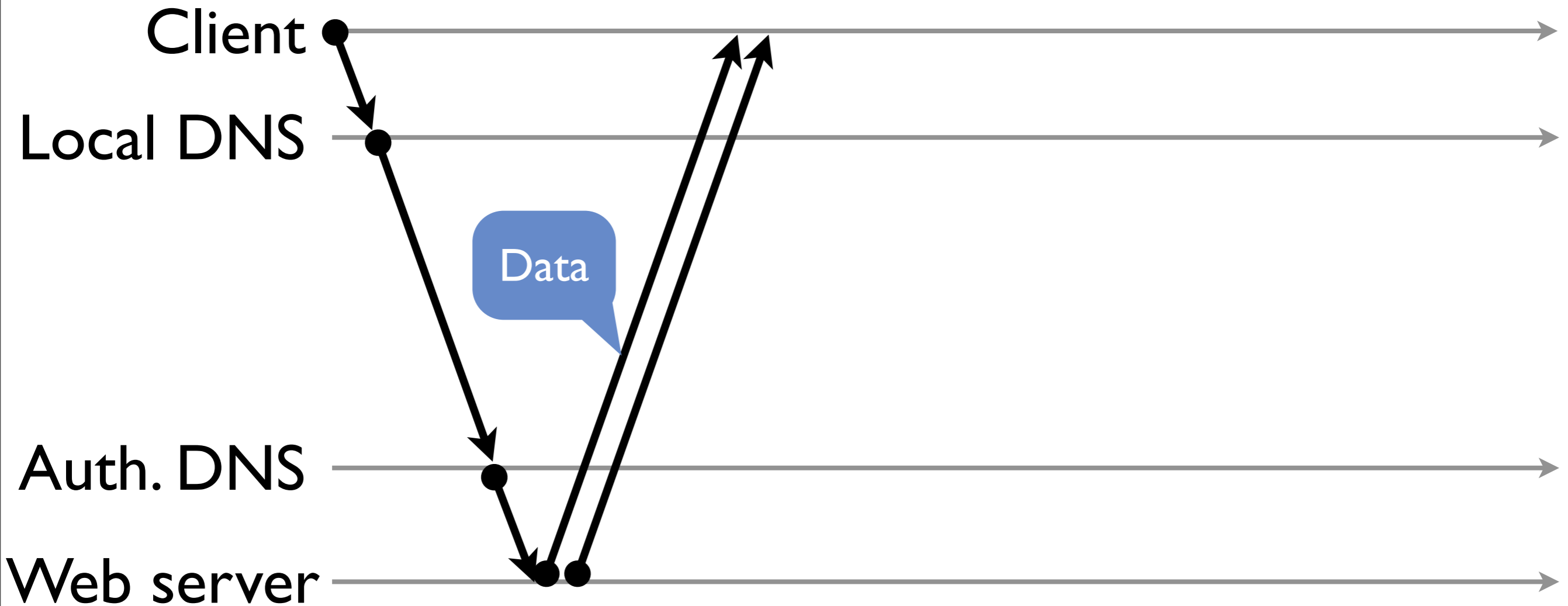
*Cl: connection information*

# Our Design: ASAP



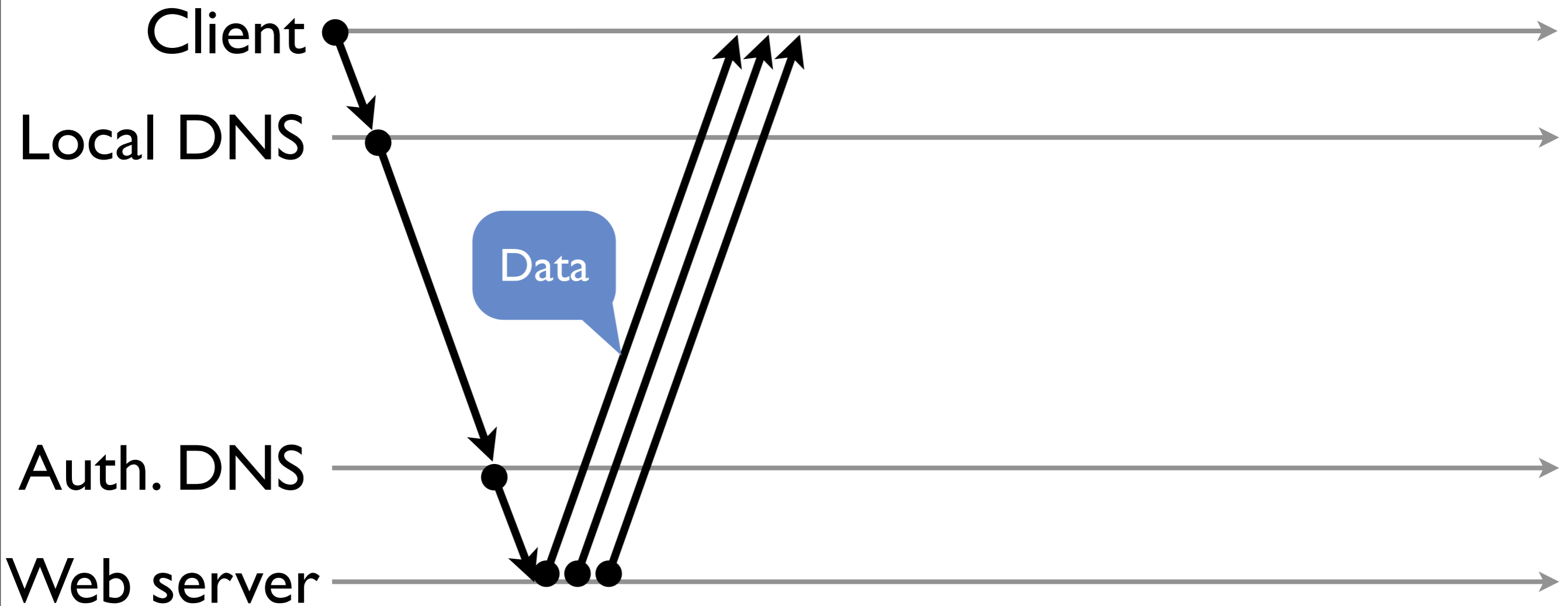
*Cl: connection information*

# Our Design: ASAP



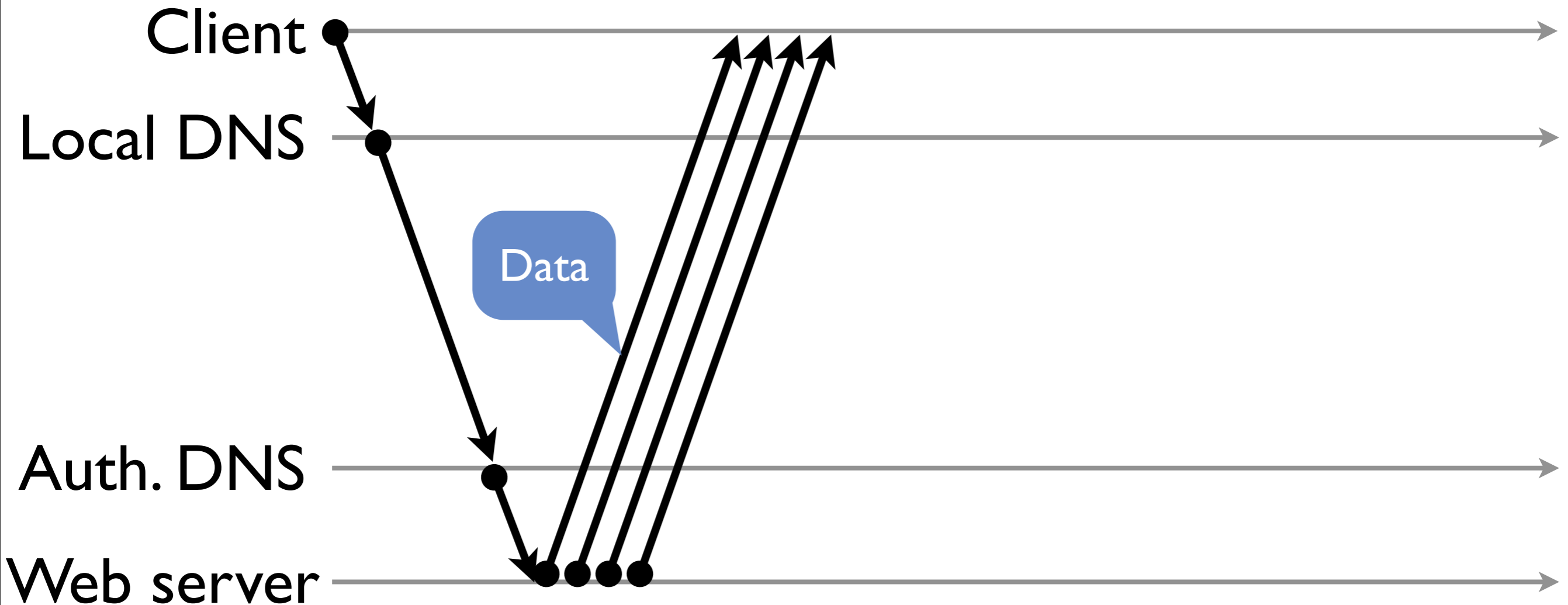
*Cl: connection information*

# Our Design: ASAP



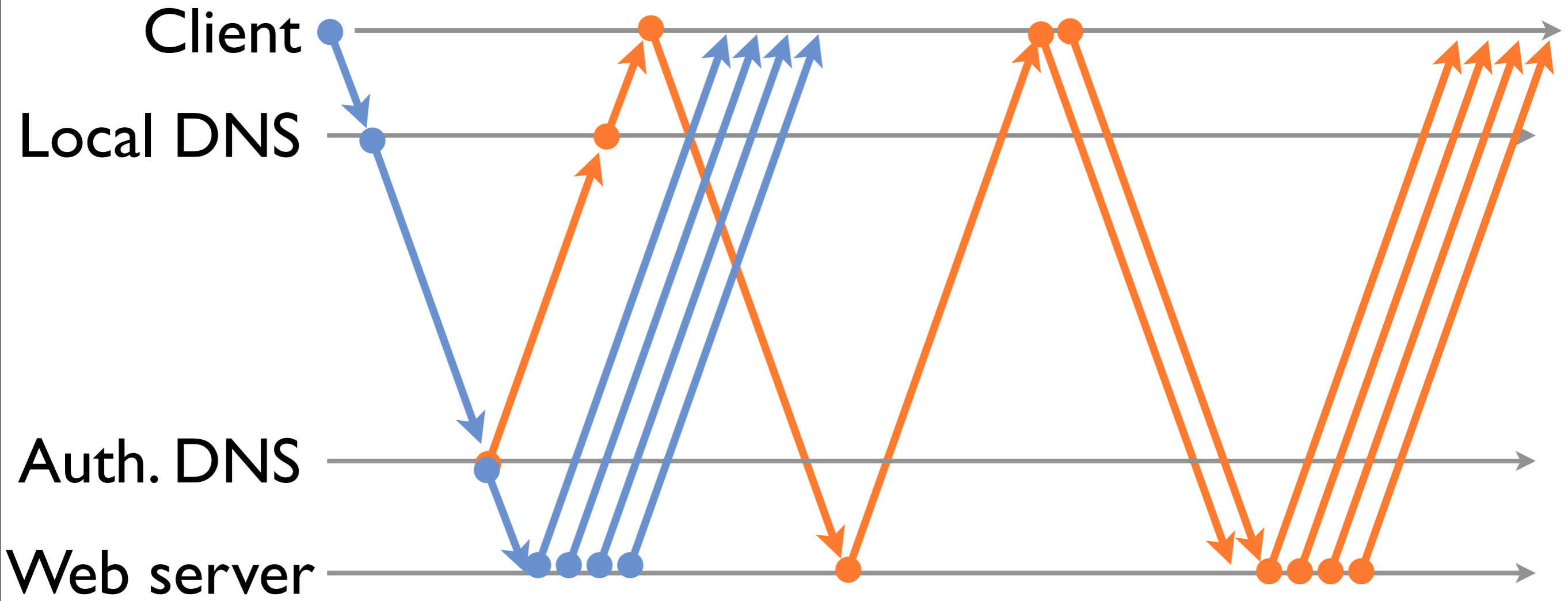
*Cl: connection information*

# Our Design: ASAP

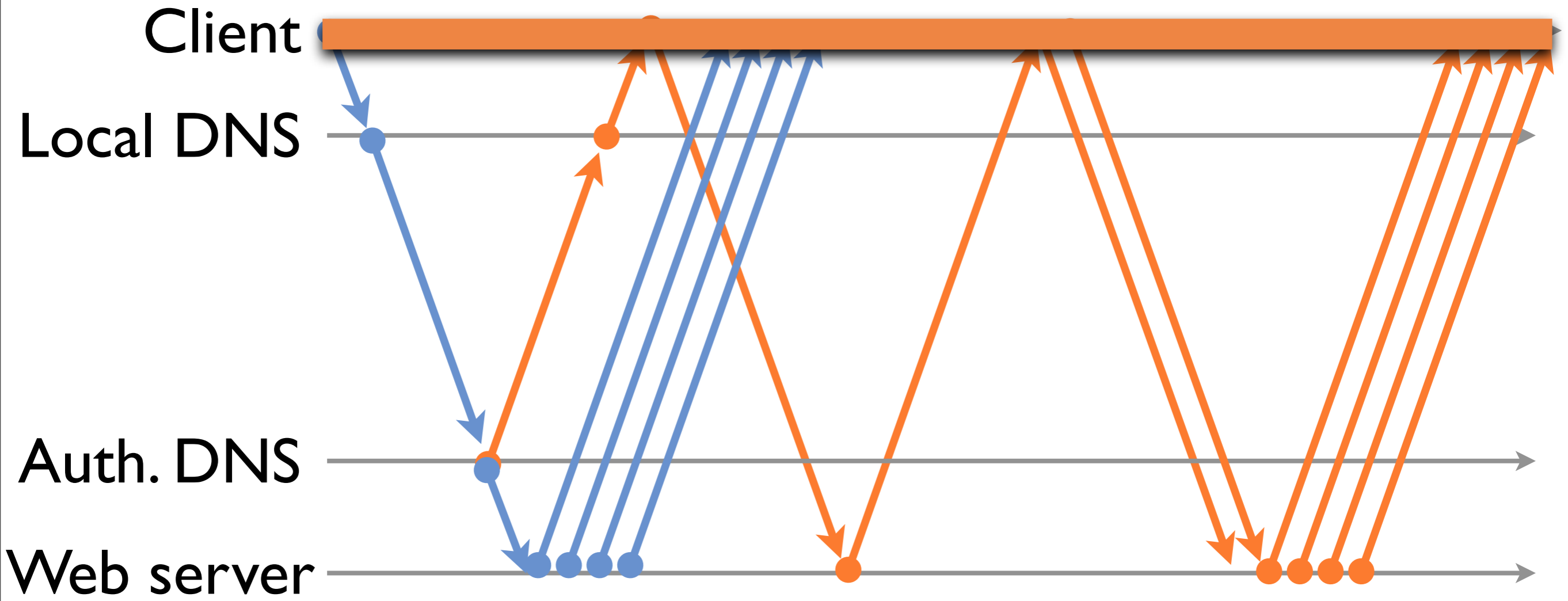


*Cl: connection information*

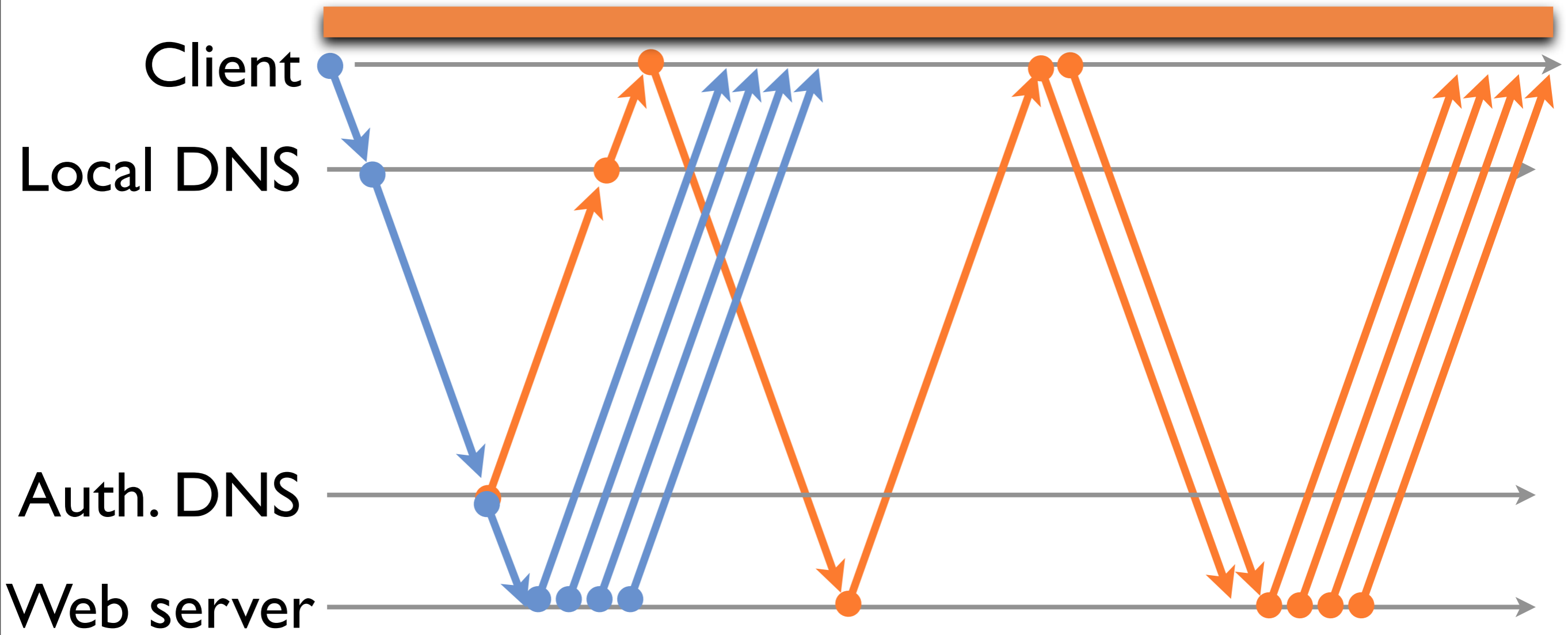
# ASAP Improvement



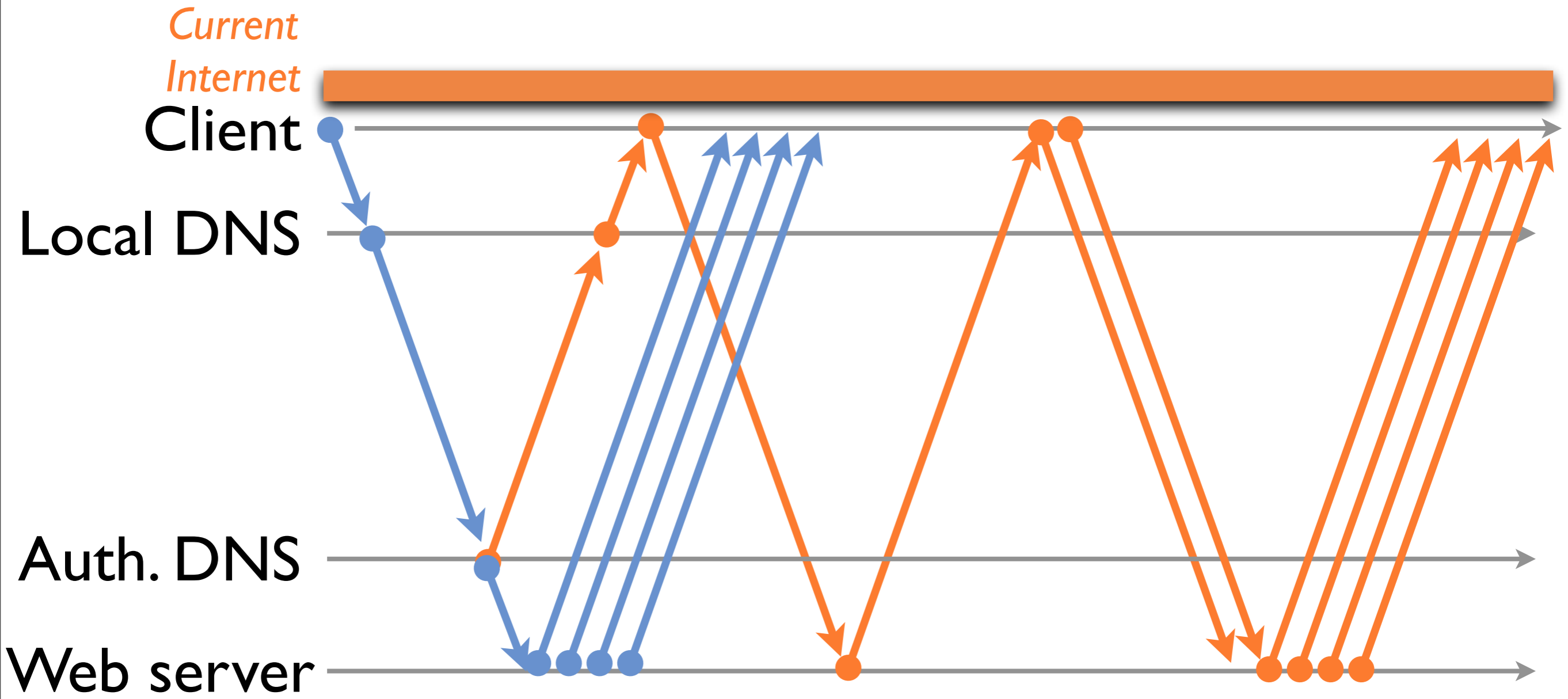
# ASAP Improvement



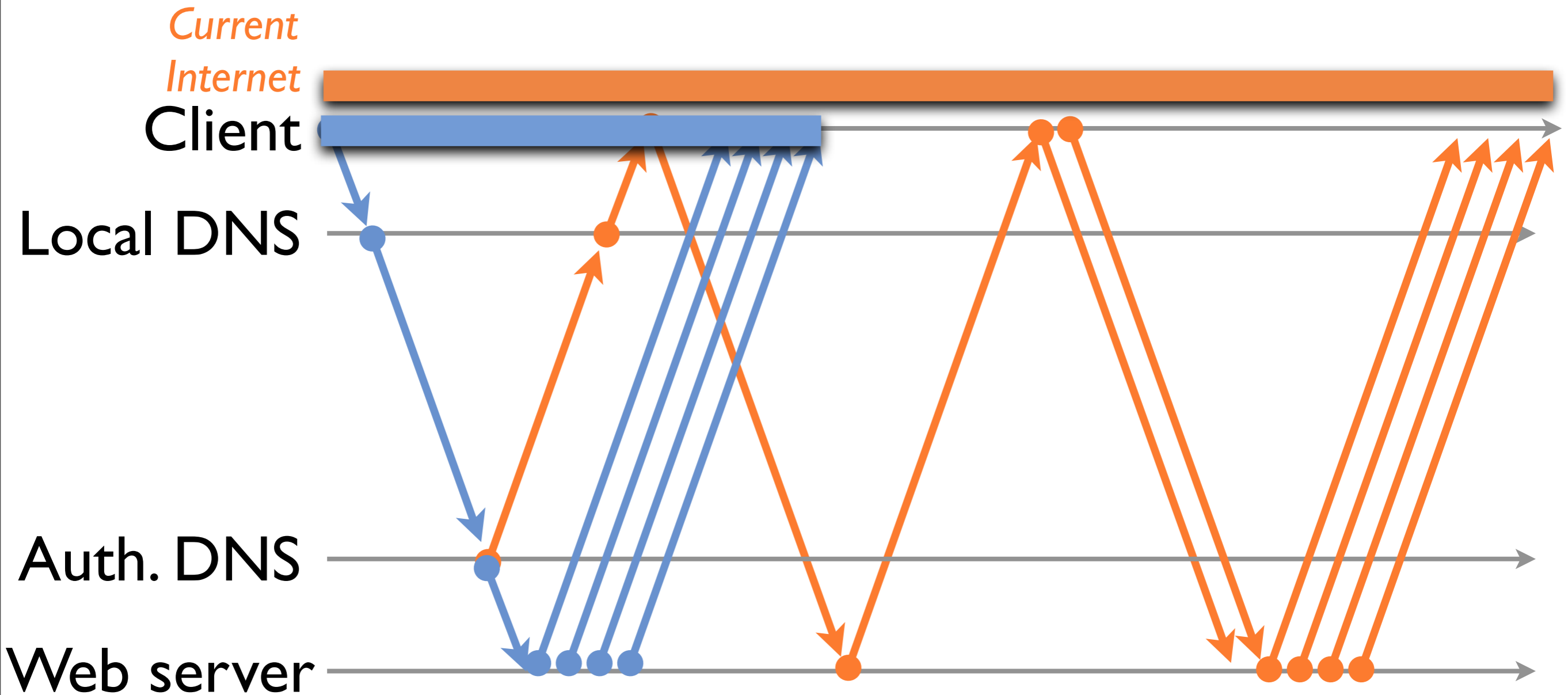
# ASAP Improvement



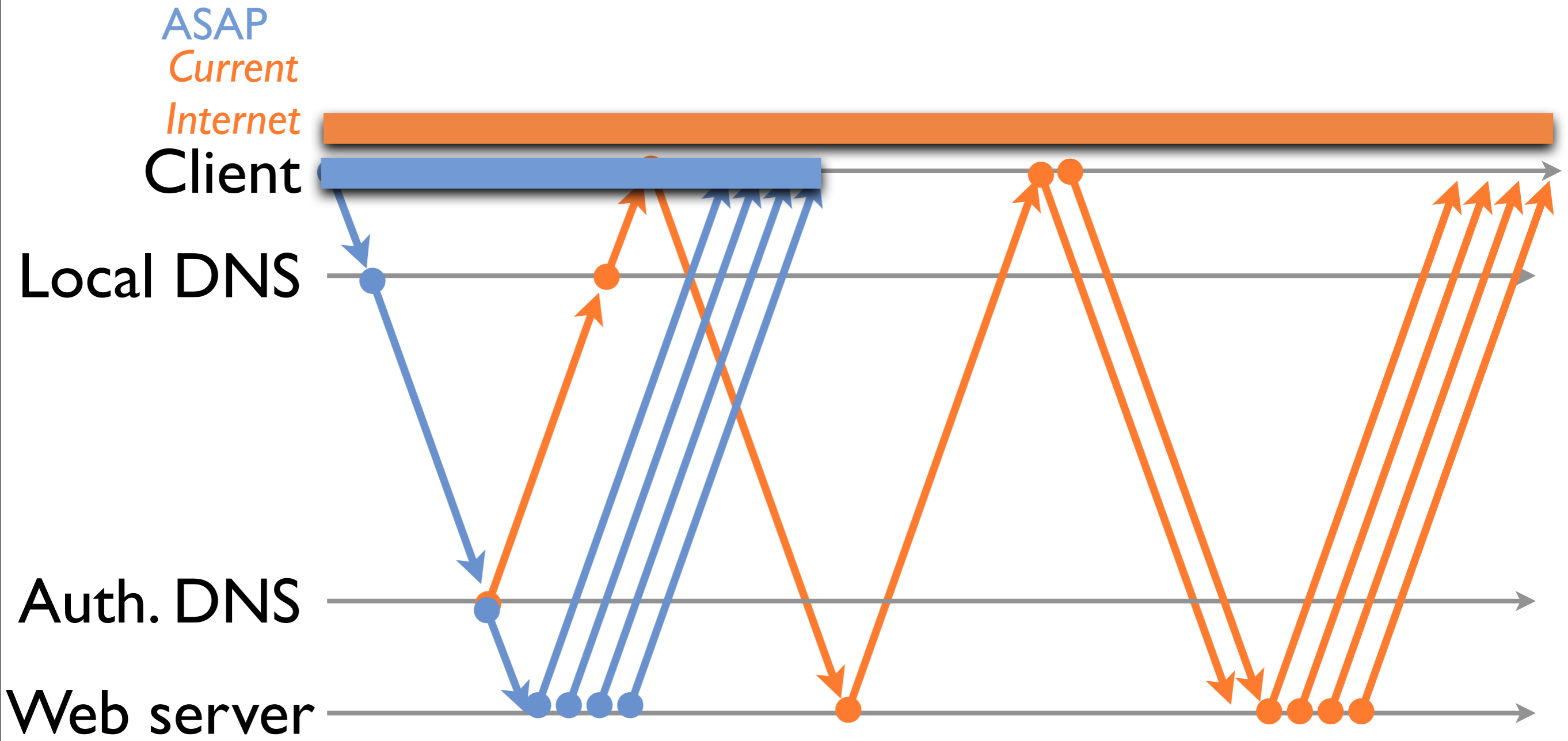
# ASAP Improvement



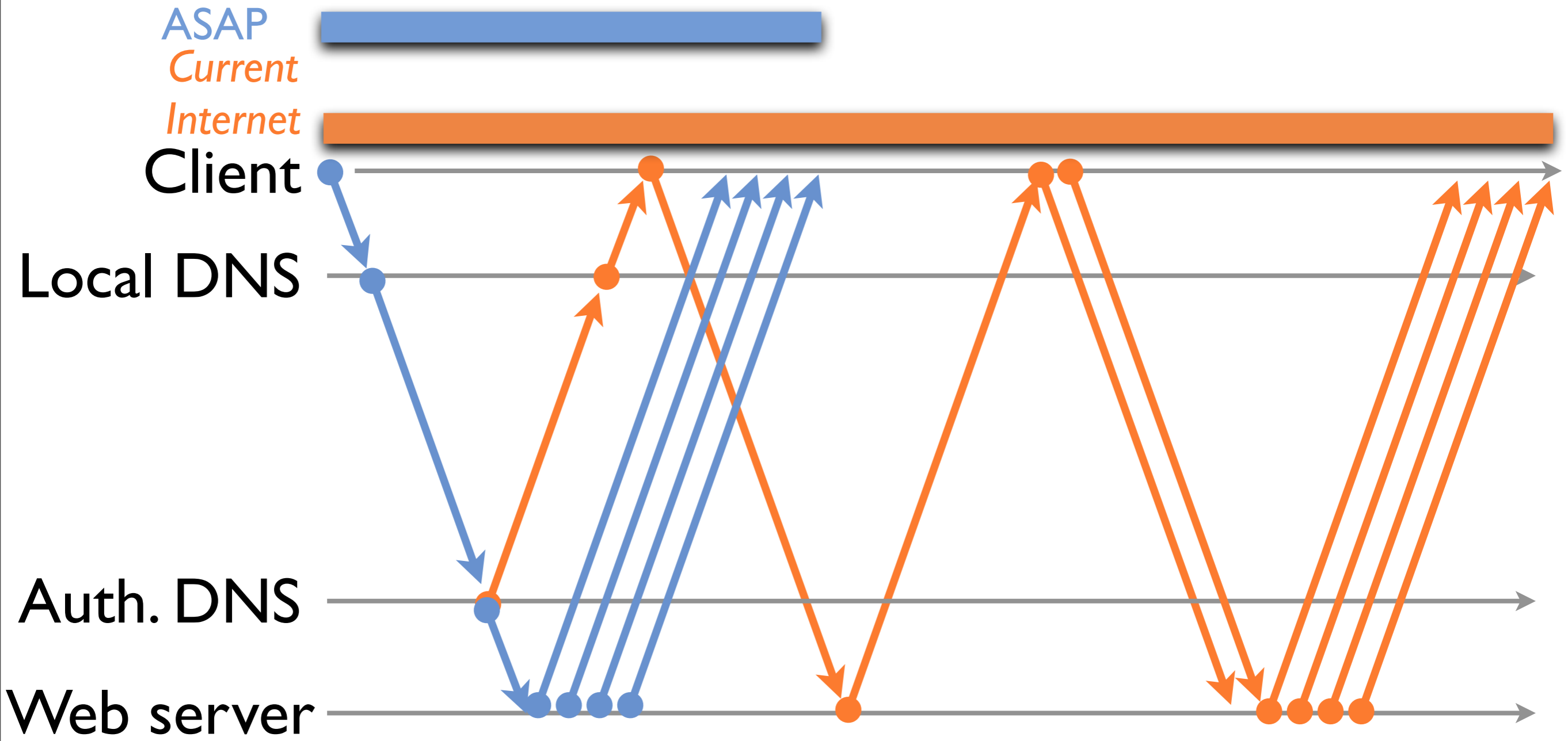
# ASAP Improvement



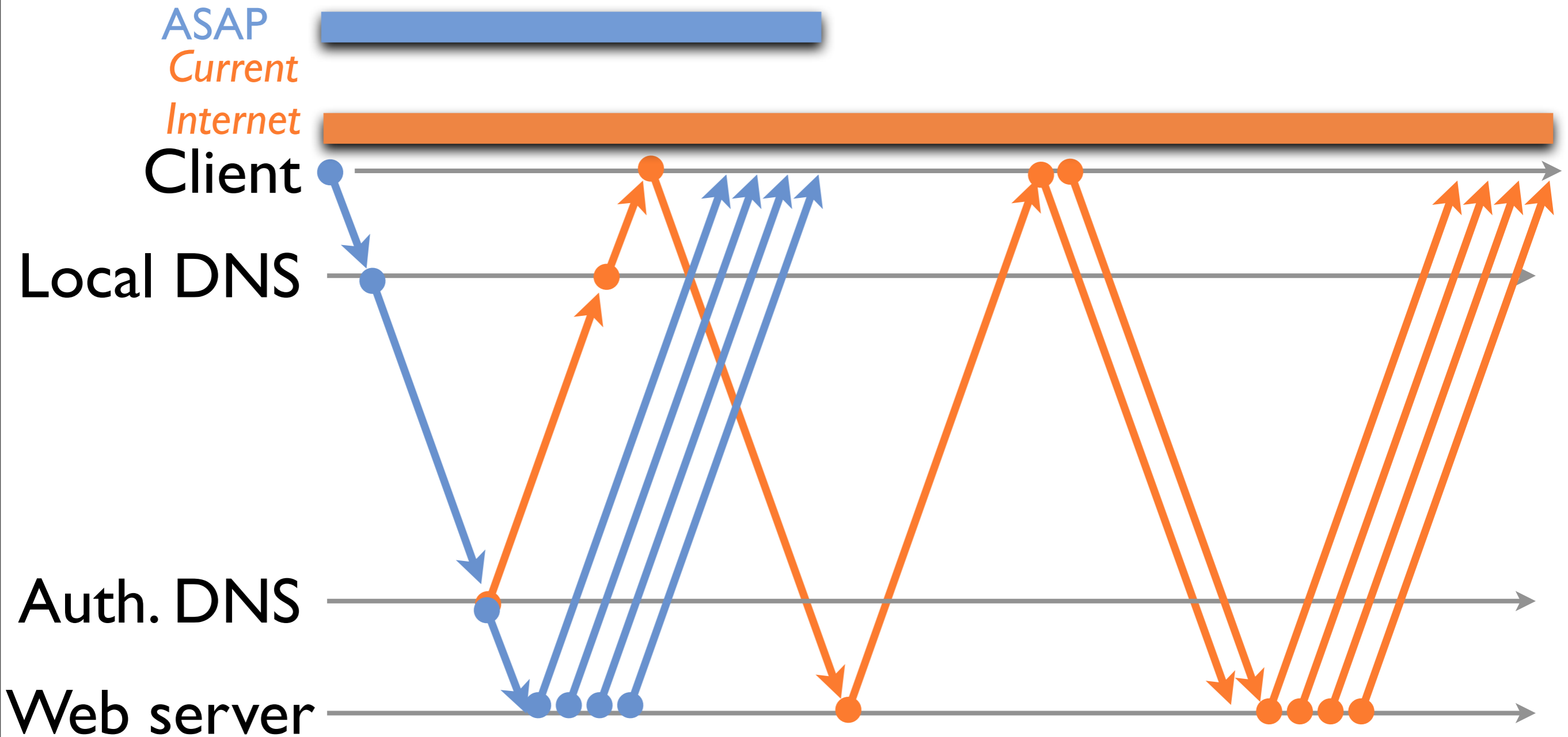
# ASAP Improvement



# ASAP Improvement

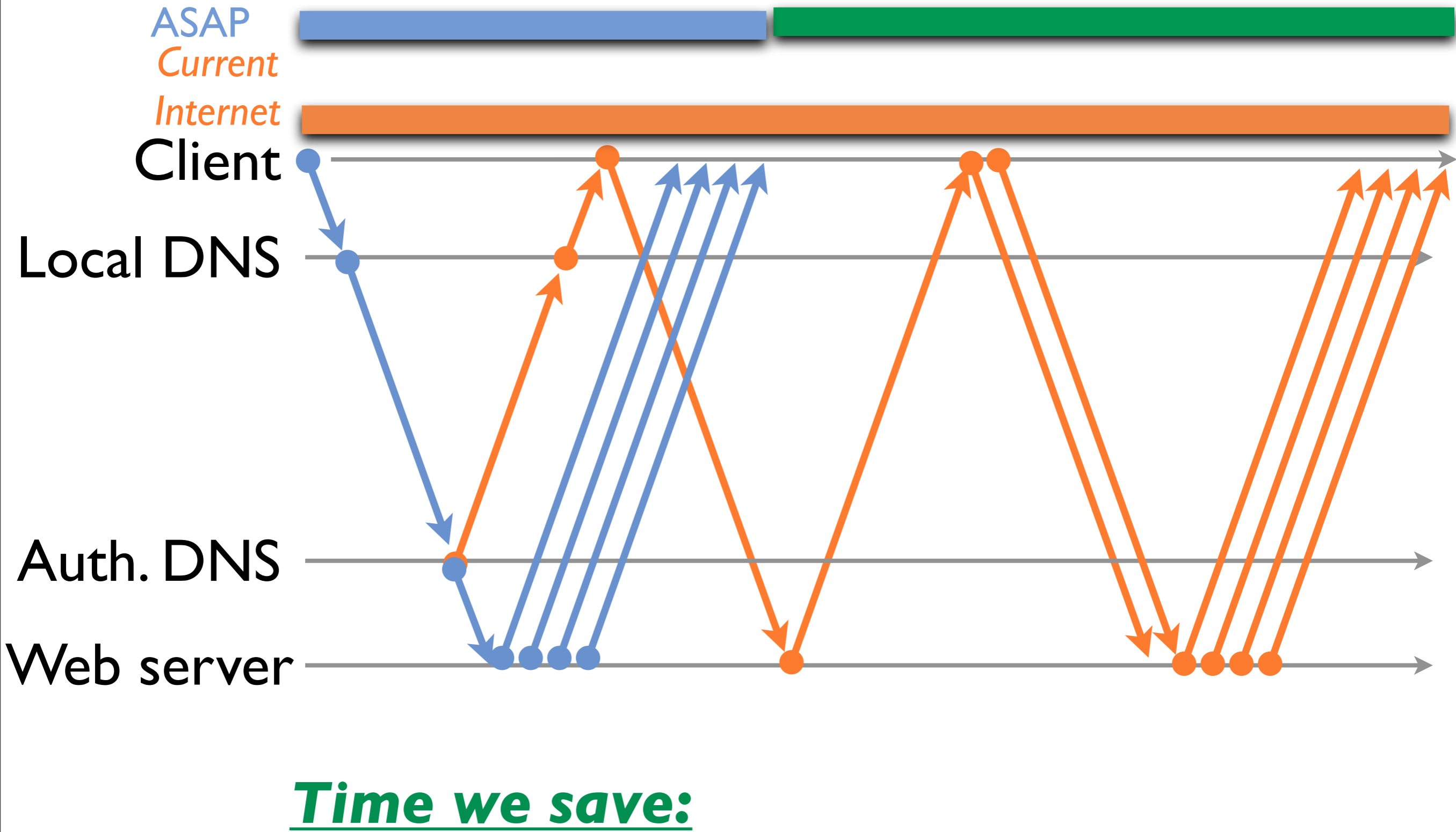


# ASAP Improvement

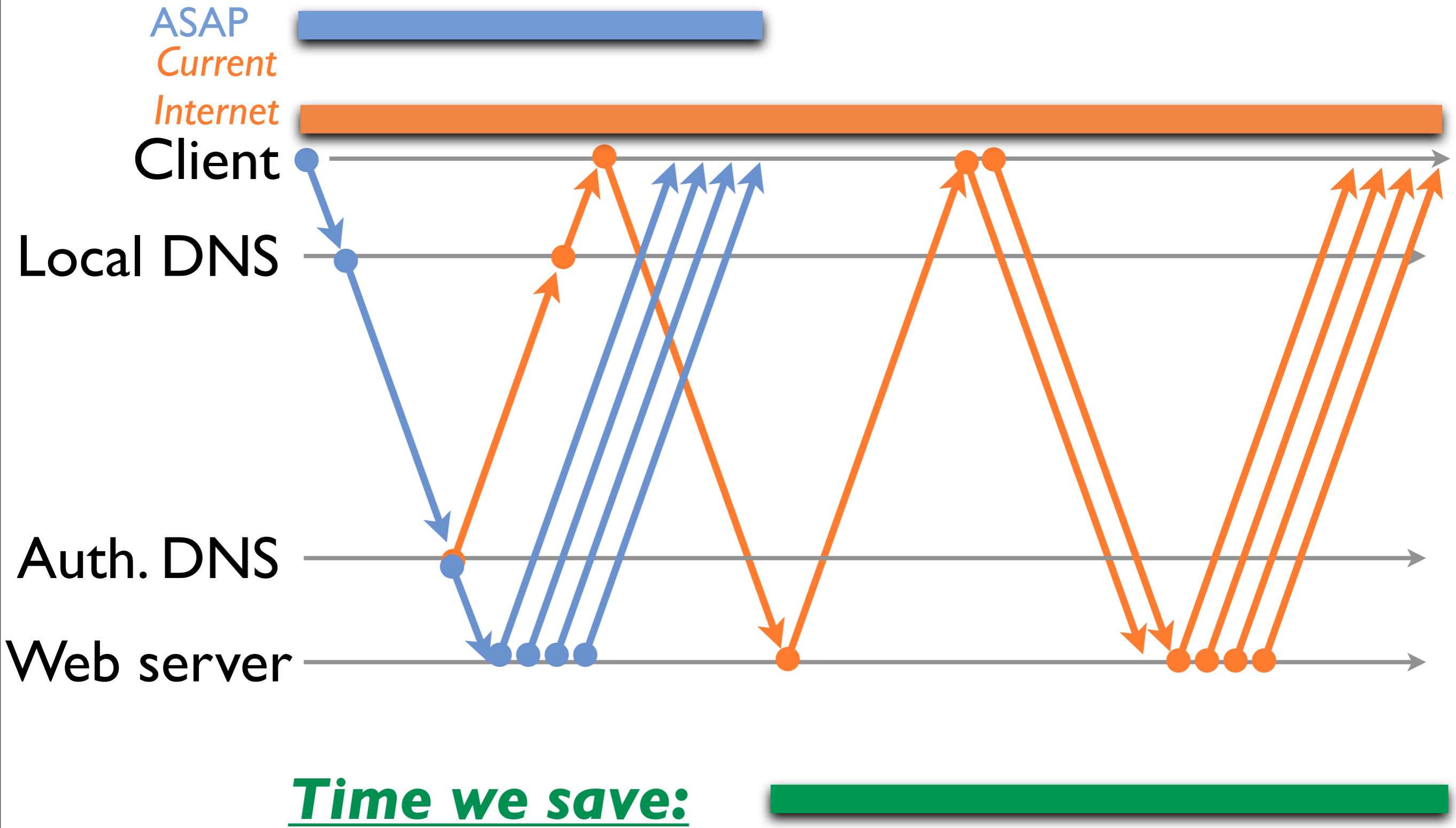


**Time we save:**

# ASAP Improvement



# ASAP Improvement



# Challenges and Solutions

# Challenges and Solutions

**Problem:** Breaks DNS caching

# Challenges and Solutions

**Problem:** Breaks DNS caching

- **Solution:** Client also sends a standard DNS query

# Challenges and Solutions

**Problem:** Breaks DNS caching

- **Solution:** Client also sends a standard DNS query

**Problem:** All queries traverse a single Auth. DNS

# Challenges and Solutions

**Problem:** Breaks DNS caching

- **Solution:** Client also sends a standard DNS query

**Problem:** All queries traverse a single Auth. DNS

- **Solution:** Another layer of Auth. DNS at servers

# Challenges and Solutions

**Problem:** Breaks DNS caching

- **Solution:** Client also sends a standard DNS query

**Problem:** All queries traverse a single Auth. DNS

- **Solution:** Another layer of Auth. DNS at servers

**Problem:** Eliminating 3-way handshake makes DoS attacks easier

# Challenges and Solutions

**Problem:** Breaks DNS caching

- **Solution:** Client also sends a standard DNS query

**Problem:** All queries traverse a single Auth. DNS

- **Solution:** Another layer of Auth. DNS at servers

**Problem:** Eliminating 3-way handshake makes DoS attacks easier

- **Solution:** Hand reusable certificates to clients

# Challenges and Solutions

**Problem:** Breaks DNS caching

- **Solution:** Client also sends a standard DNS query

**Problem:** All queries traverse a single Auth. DNS

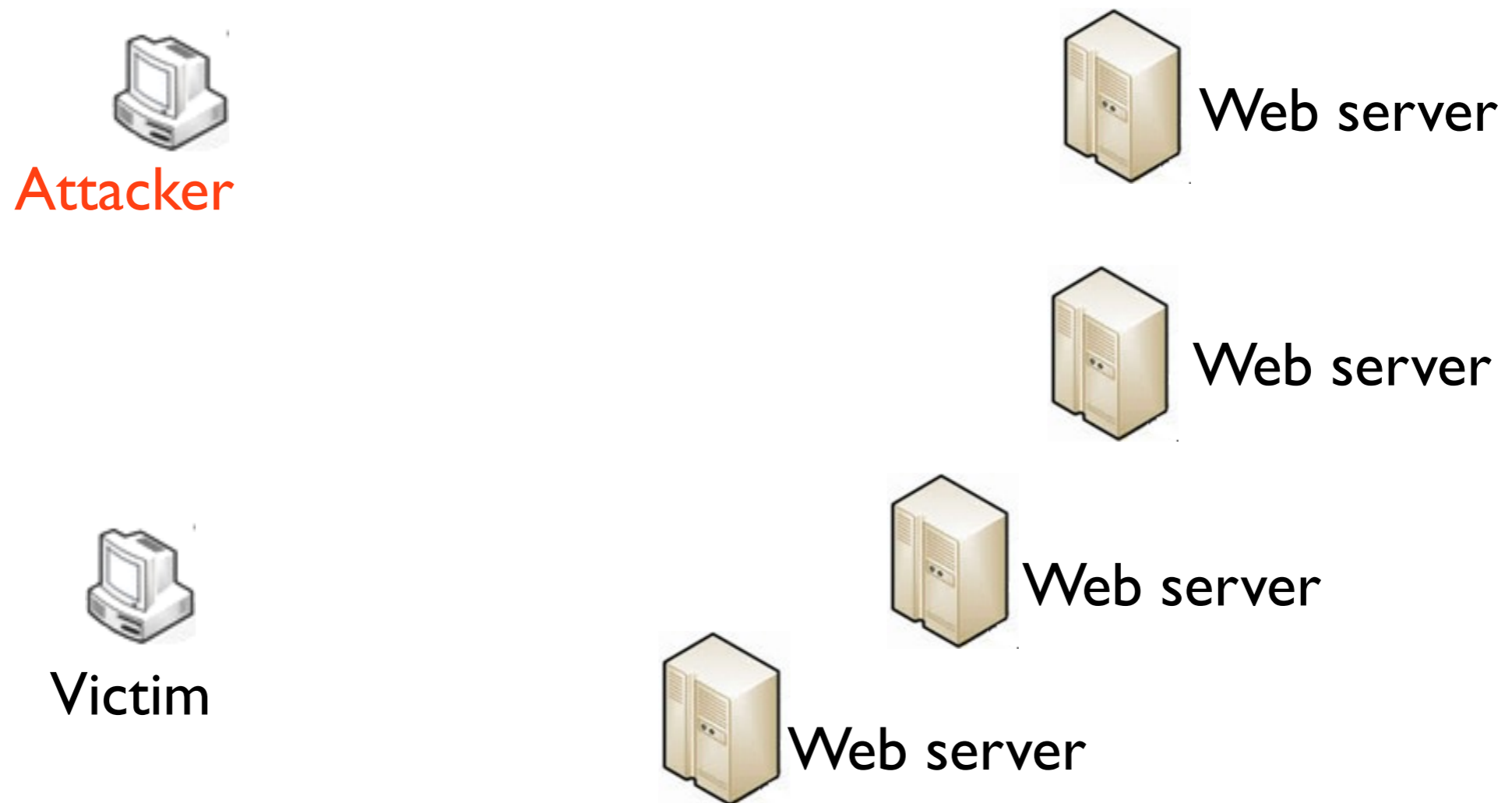
- **Solution:** Another layer of Auth. DNS at servers

**Problem:** Eliminating 3-way handshake makes DoS attacks easier

- **Solution:** Hand reusable certificates to clients

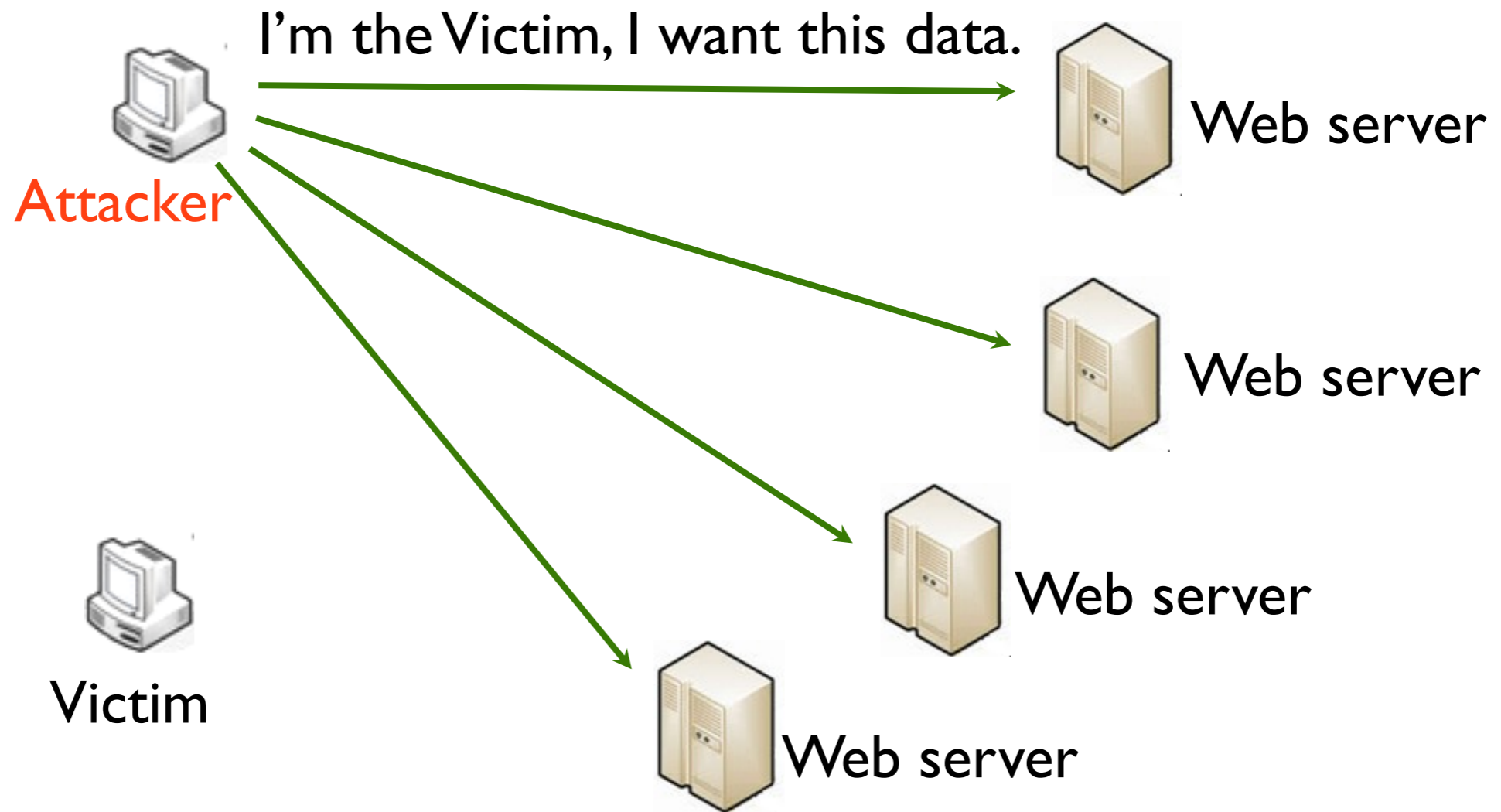
# Security Problem: DoS Attack

- Reflection/Amplification DoS attack



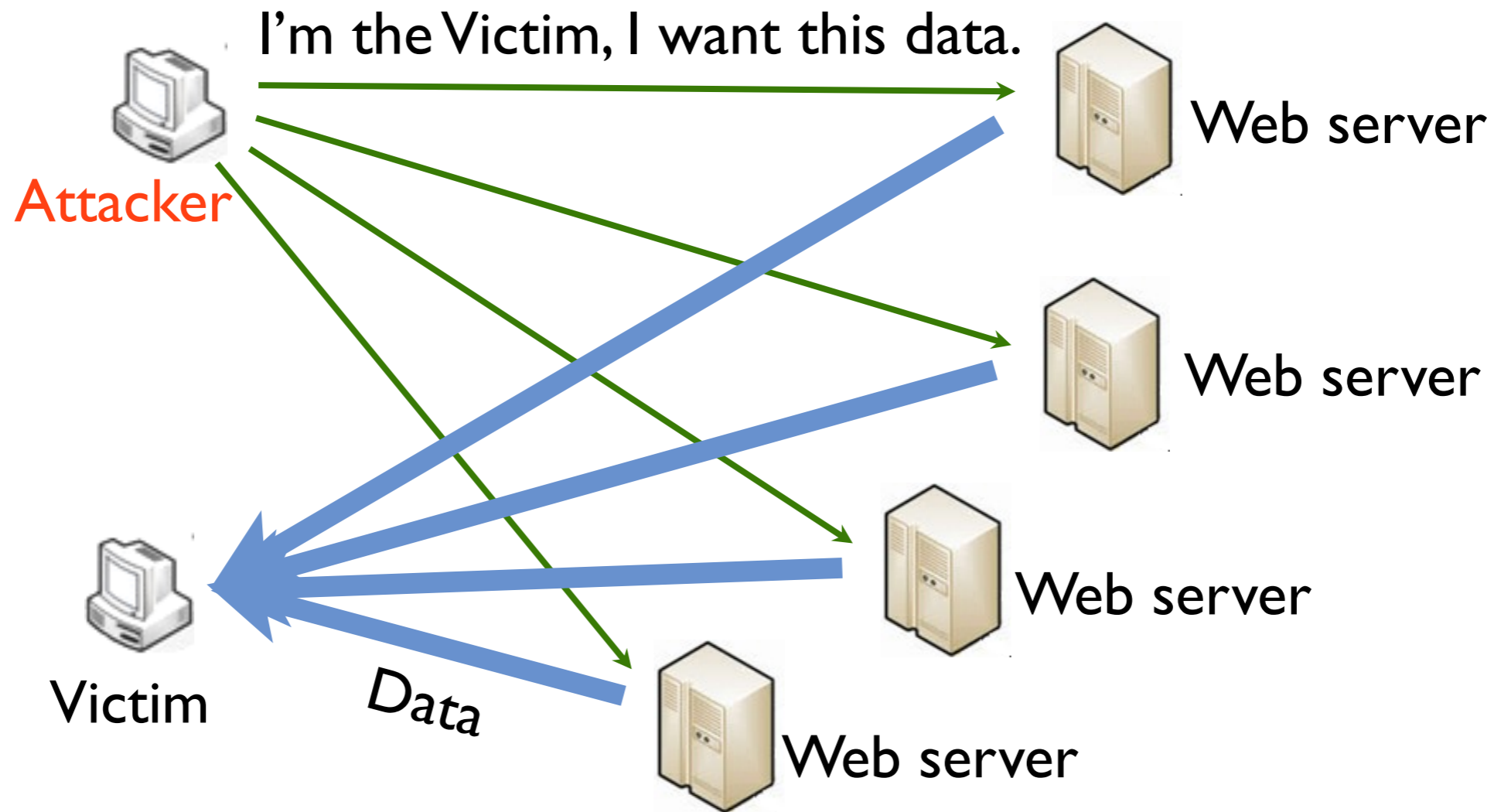
# Security Problem: DoS Attack

- Reflection/Amplification DoS attack



# Security Problem: DoS Attack

- Reflection/Amplification DoS attack



# Security Mechanism

# Security Mechanism

**Goal:** Servers verify client's address without adding an RTT

# Security Mechanism

**Goal:** Servers verify client's address without adding an RTT

**Solution:** Cryptographic proofs

# Security Mechanism

**Goal:** Servers verify client's address without adding an RTT

**Solution:** Cryptographic proofs



Provenance  
Verifier (PV)

# Security Mechanism

**Goal:** Servers verify client's address without adding an RTT

**Solution:** Cryptographic proofs



Provenance  
Verifier (PV)

*Choices of PV:*

- Web server
- CDN
- Trusted 3<sup>rd</sup> party
- ...

# Security Mechanism

**Goal:** Servers verify client's address without adding an RTT

**Solution:** Cryptographic proofs

Client



Provenance  
Verifier (PV)

*Choices of PV:*

- Web server
- CDN
- Trusted 3<sup>rd</sup> party
- ...

# Security Mechanism

**Goal:** Servers verify client's address without adding an RTT

**Solution:** Cryptographic proofs

Client



Provenance  
Verifier (PV)



Web Server

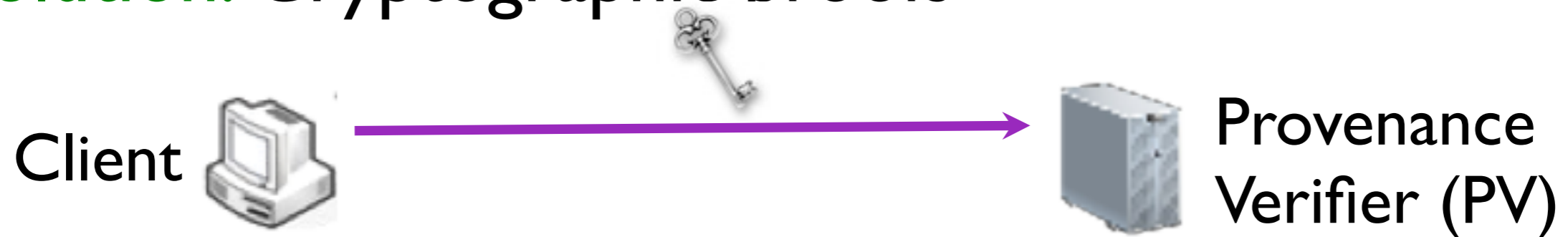
*Choices of PV:*

- Web server
- CDN
- Trusted 3<sup>rd</sup> party
- ...

# Security Mechanism

**Goal:** Servers verify client's address without adding an RTT

**Solution:** Cryptographic proofs



*Choices of PV:*

- Web server
- CDN
- Trusted 3<sup>rd</sup> party
- ...

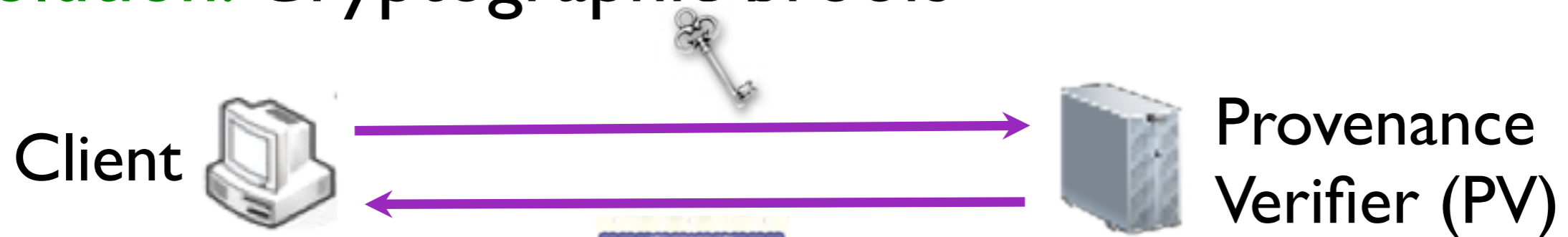


Web Server

# Security Mechanism

**Goal:** Servers verify client's address without adding an RTT

**Solution:** Cryptographic proofs



Certificate



Web Server

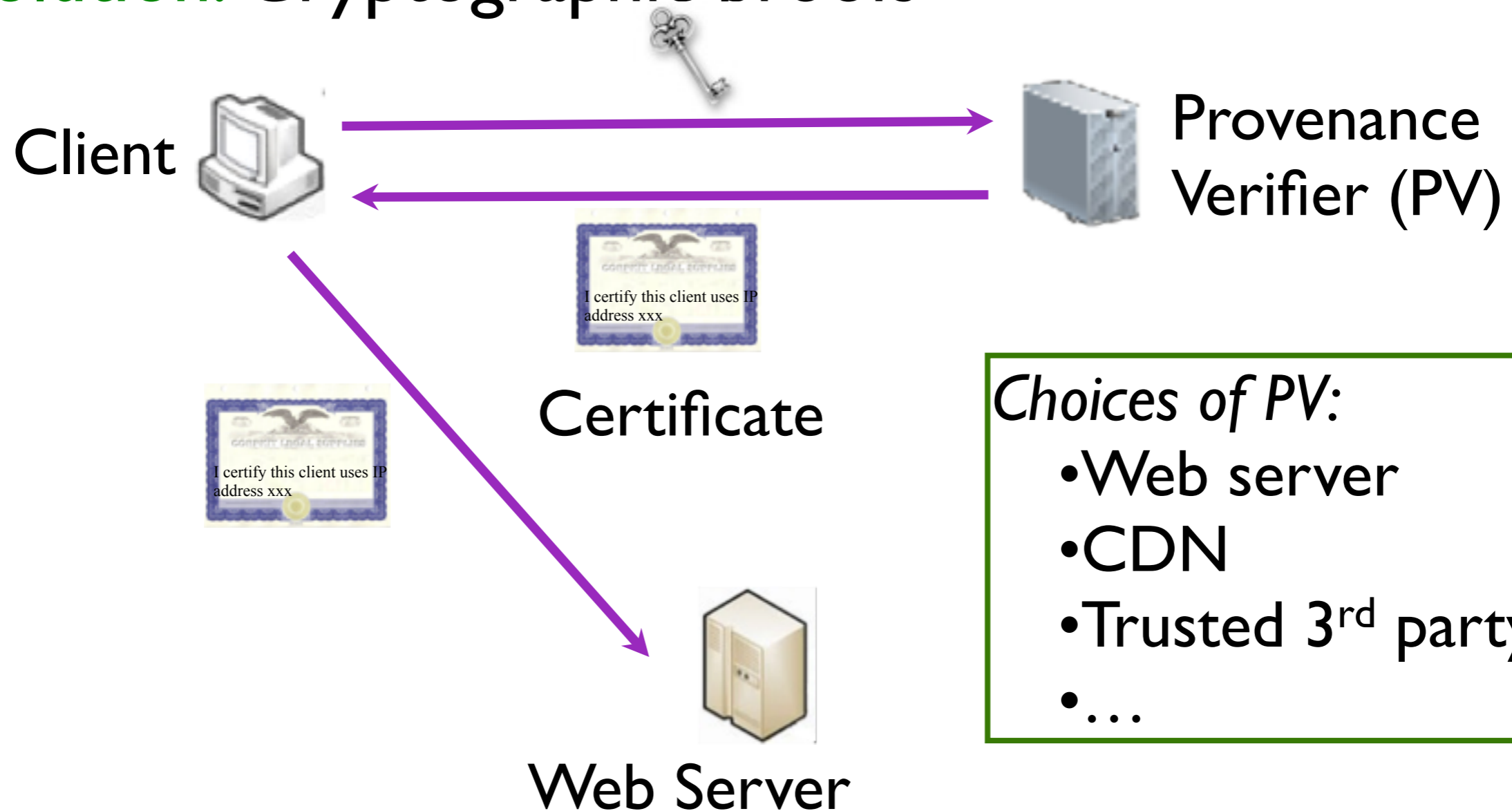
*Choices of PV:*

- Web server
- CDN
- Trusted 3<sup>rd</sup> party
- ...

# Security Mechanism

**Goal:** Servers verify client's address without adding an RTT

**Solution:** Cryptographic proofs



*Choices of PV:*

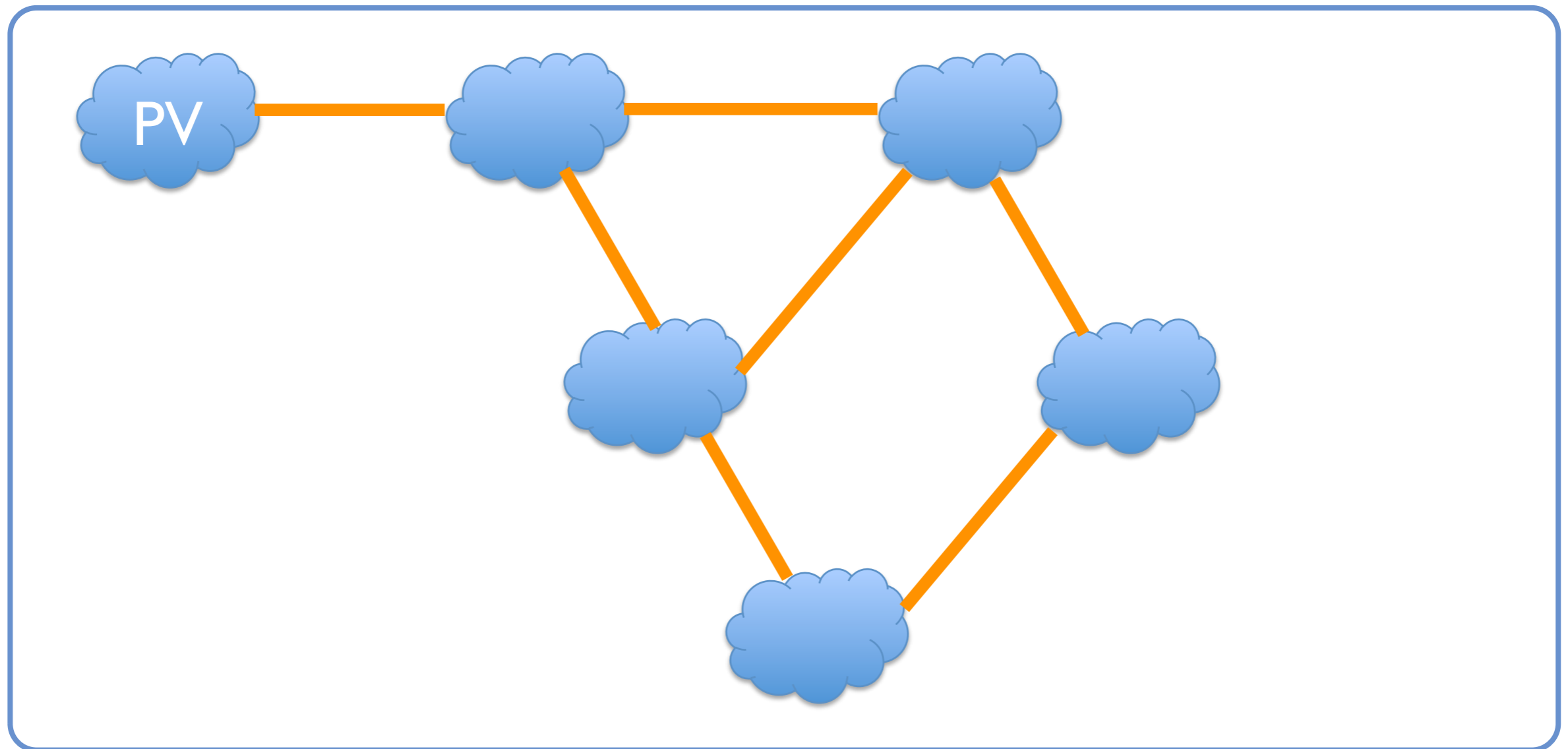
- Web server
- CDN
- Trusted 3<sup>rd</sup> party
- ...

# Security Mechanism (cont'd)

**Problem:** Eavesdropping

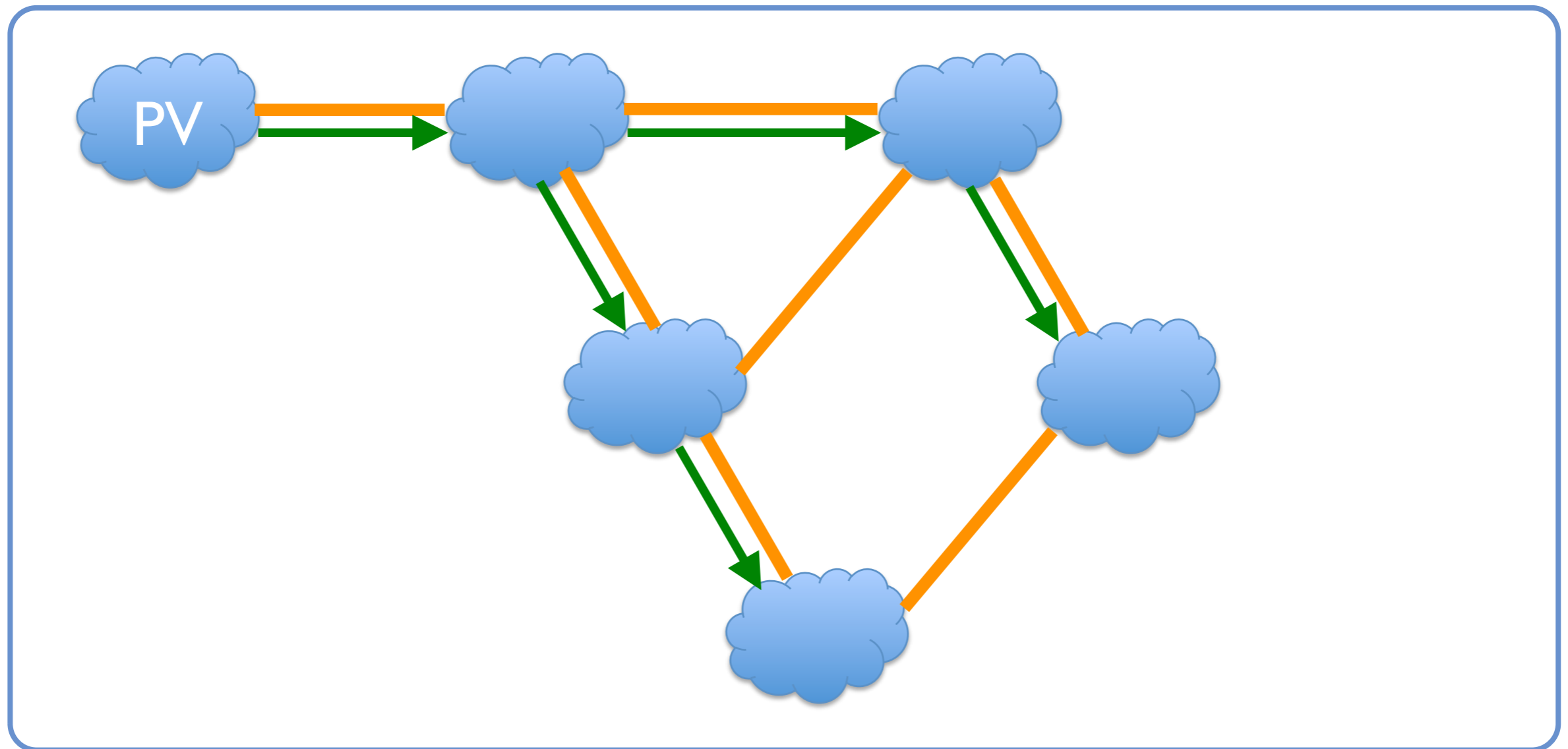
# Security Mechanism (cont'd)

## Problem: Eavesdropping



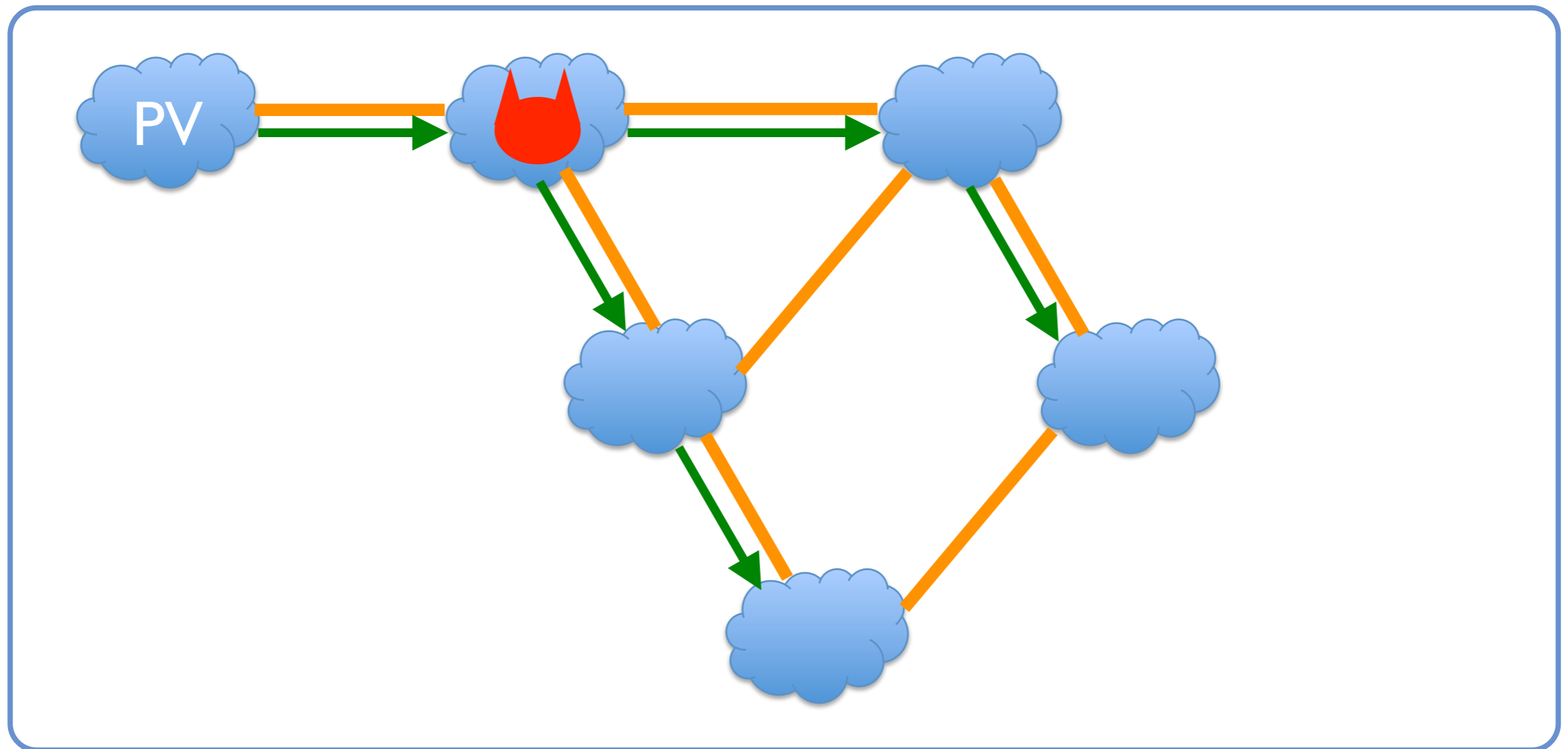
# Security Mechanism (cont'd)

## Problem: Eavesdropping



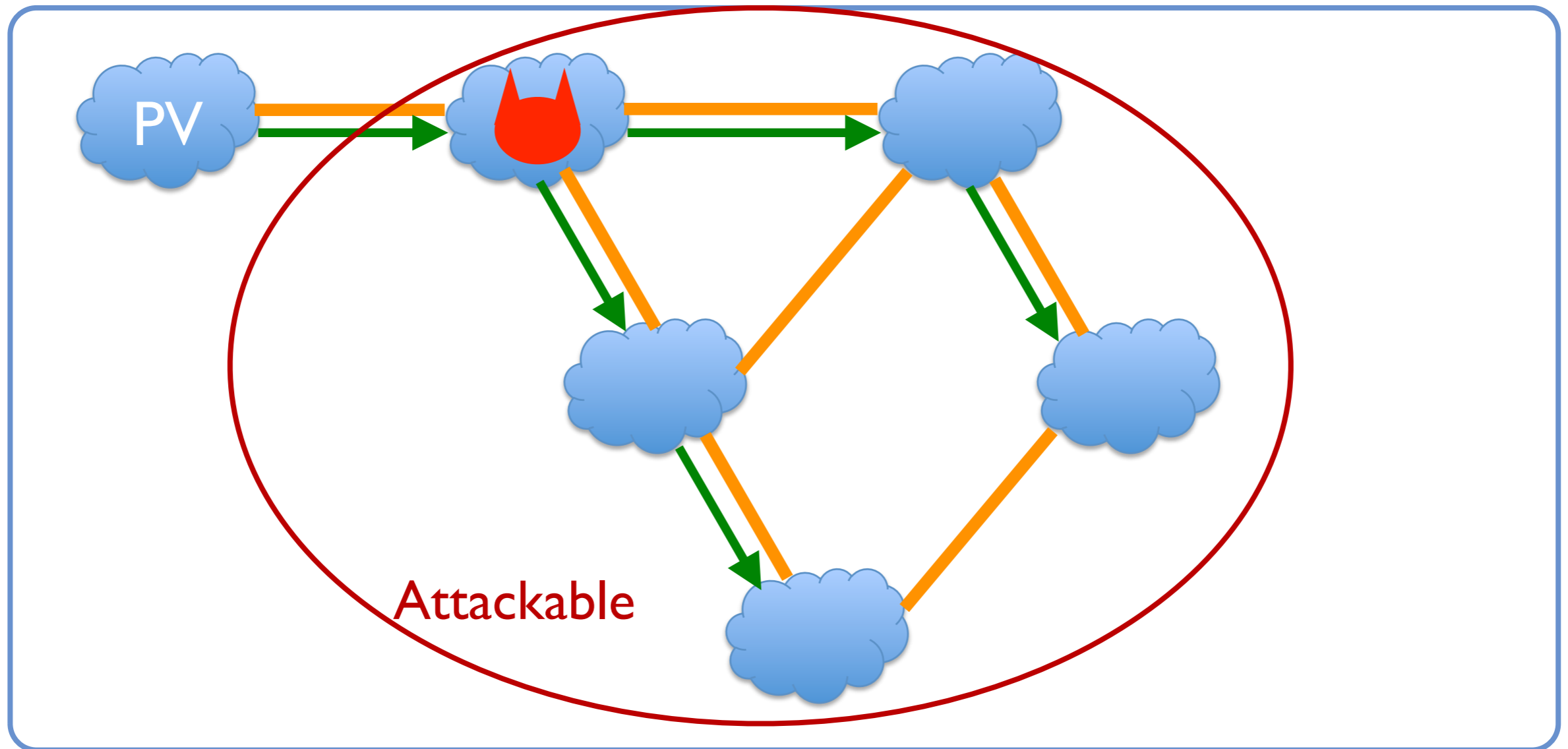
# Security Mechanism (cont'd)

## Problem: Eavesdropping



# Security Mechanism (cont'd)

## Problem: Eavesdropping



# Security Mechanism (cont'd)

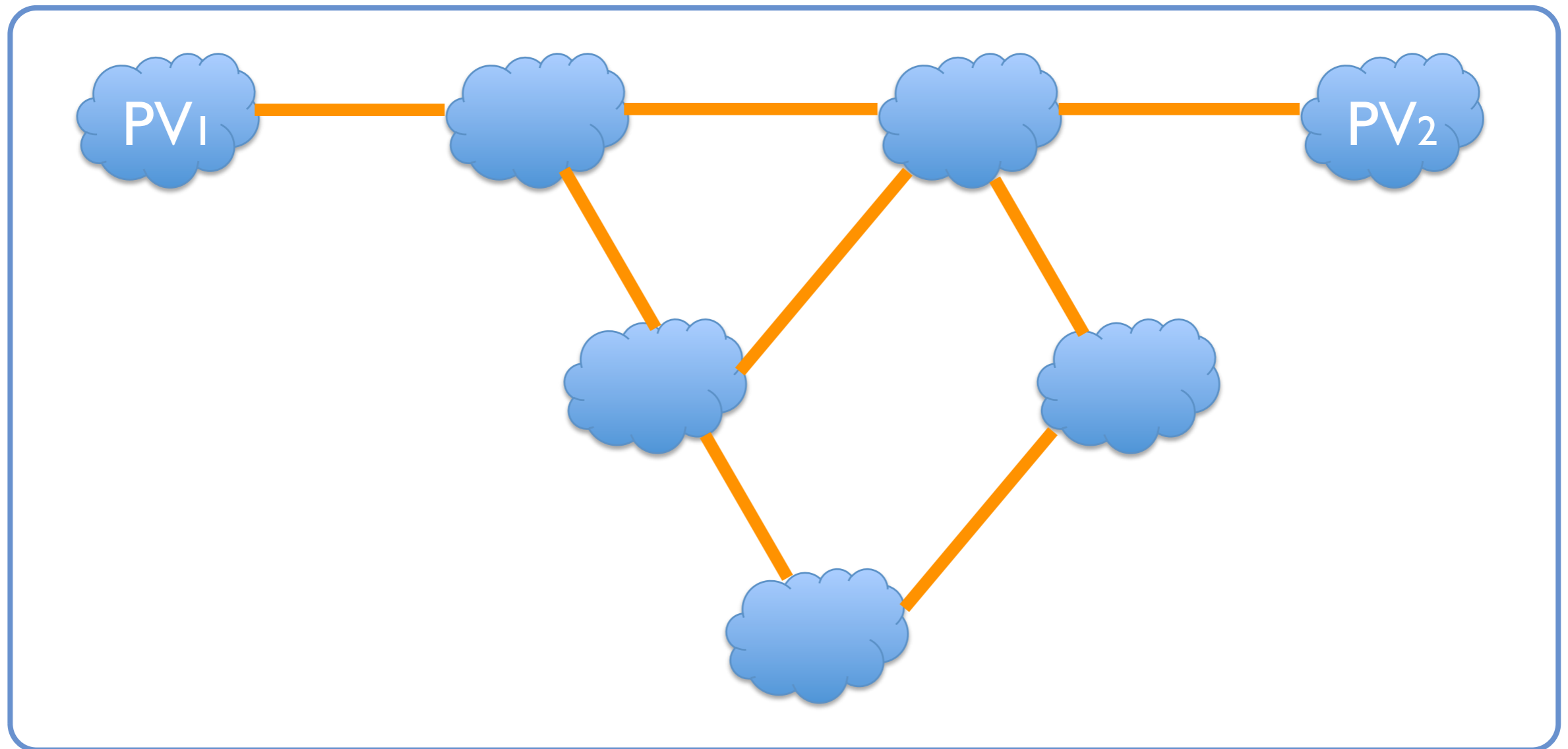
**Problem:** Eavesdropping

**Solution:** Multiple PVs

# Security Mechanism (cont'd)

**Problem:** Eavesdropping

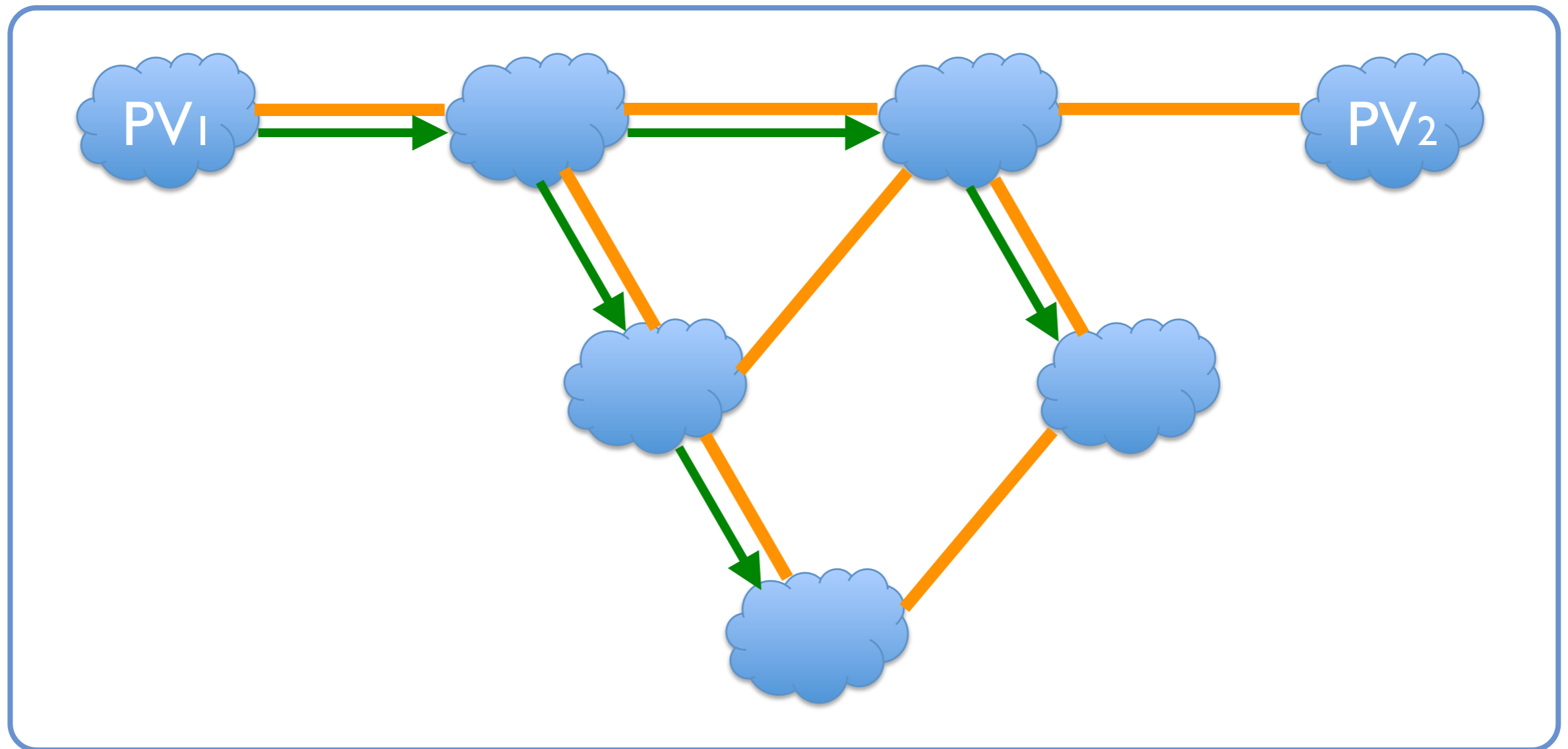
**Solution:** Multiple PVs



# Security Mechanism (cont'd)

**Problem:** Eavesdropping

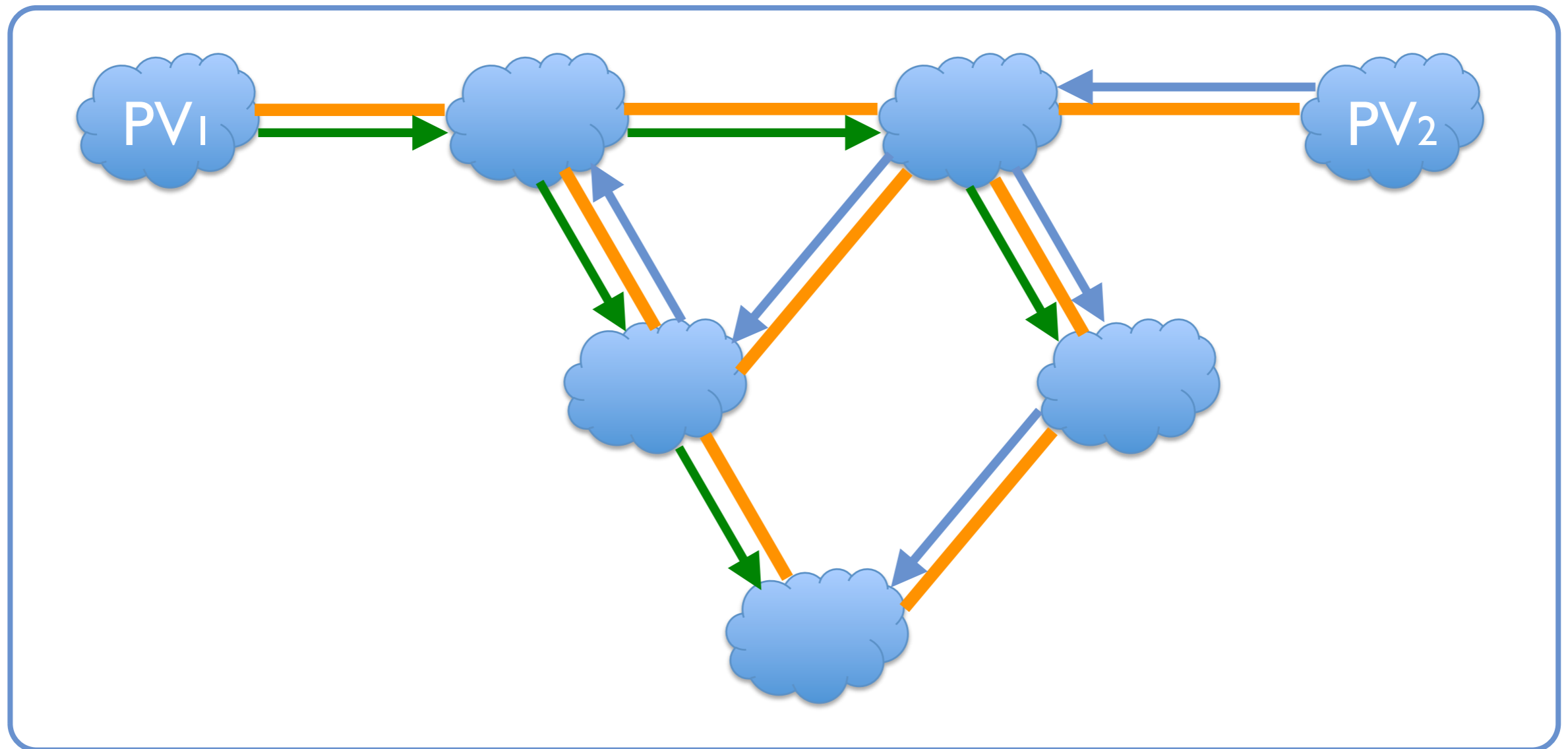
**Solution:** Multiple PVs



# Security Mechanism (cont'd)

**Problem:** Eavesdropping

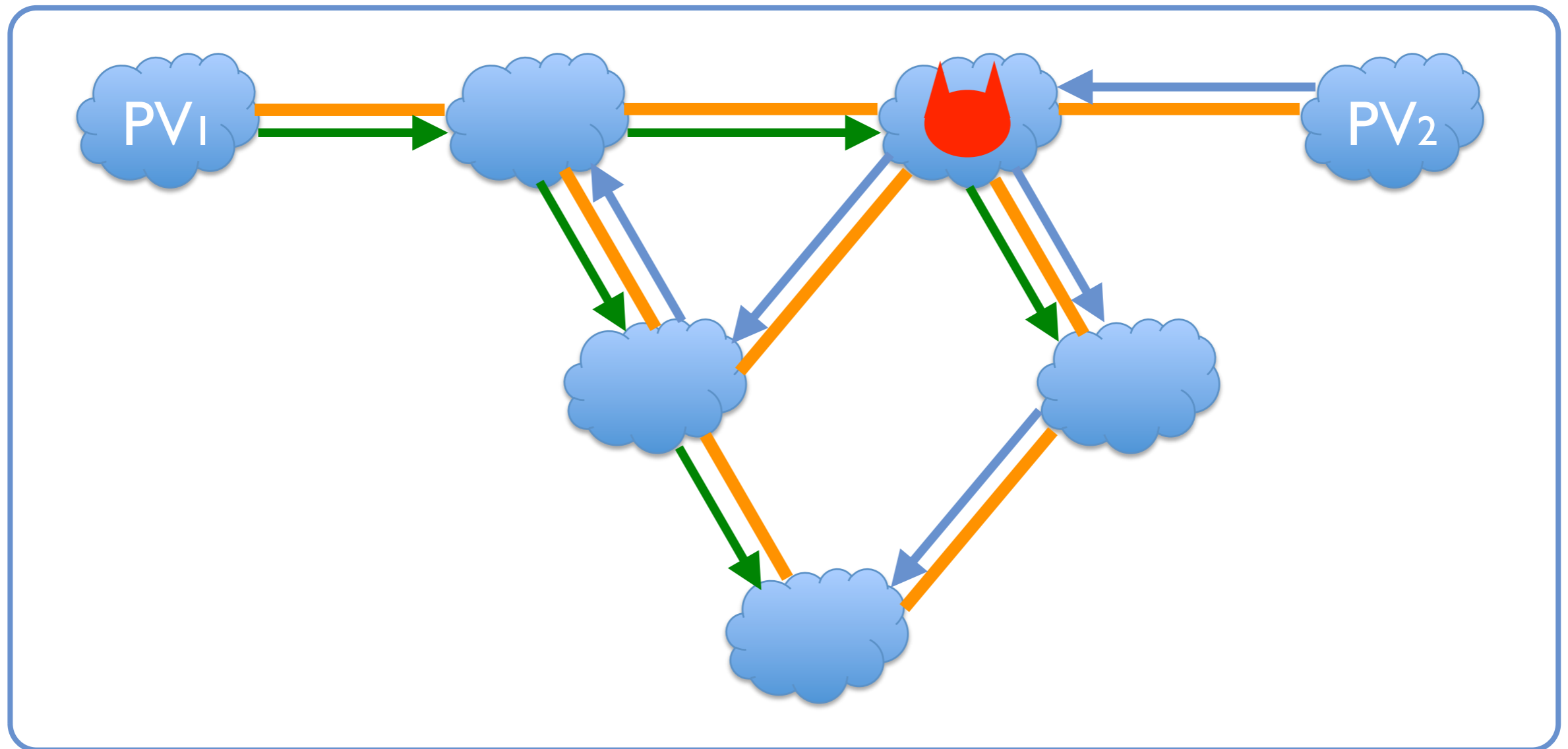
**Solution:** Multiple PVs



# Security Mechanism (cont'd)

**Problem:** Eavesdropping

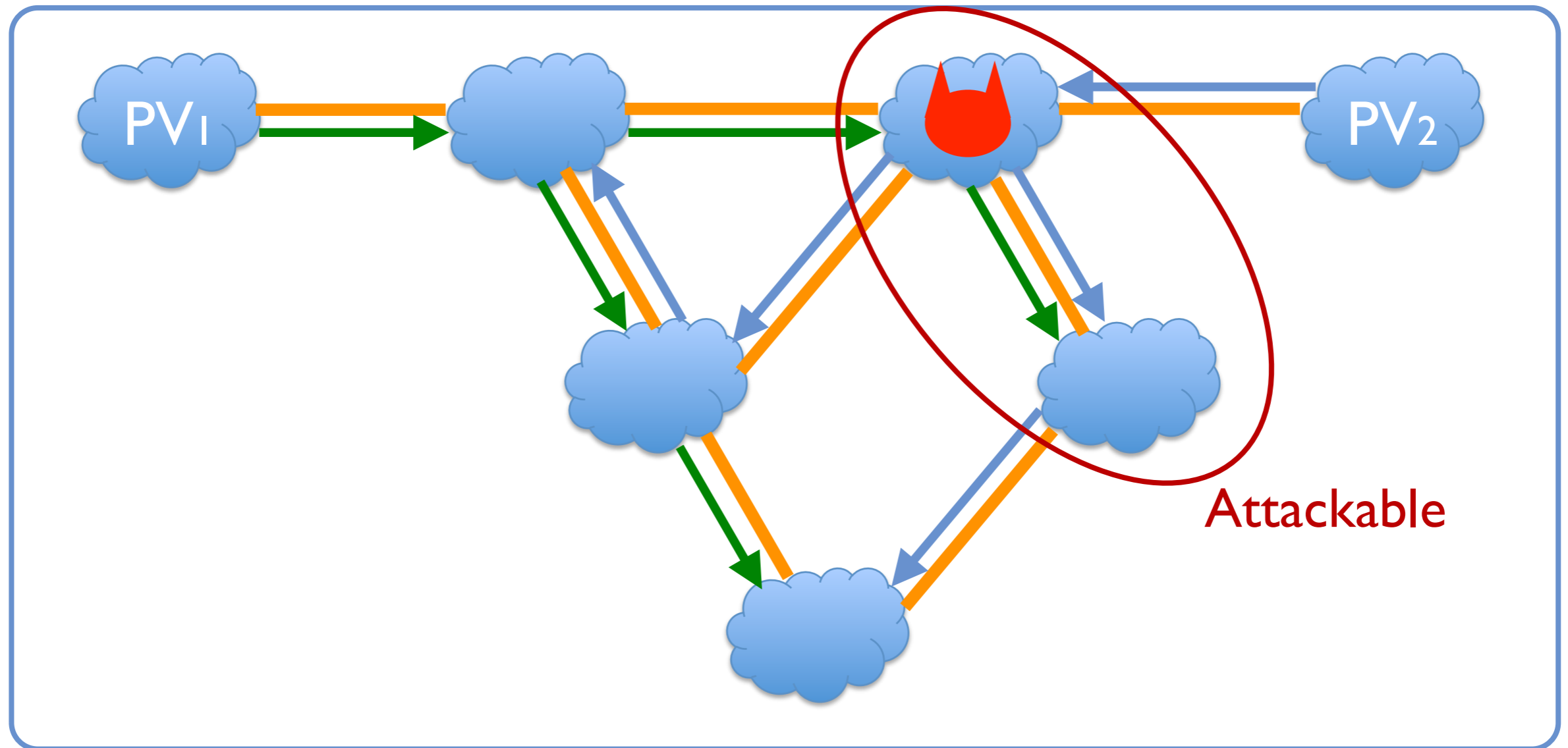
**Solution:** Multiple PVs



# Security Mechanism (cont'd)

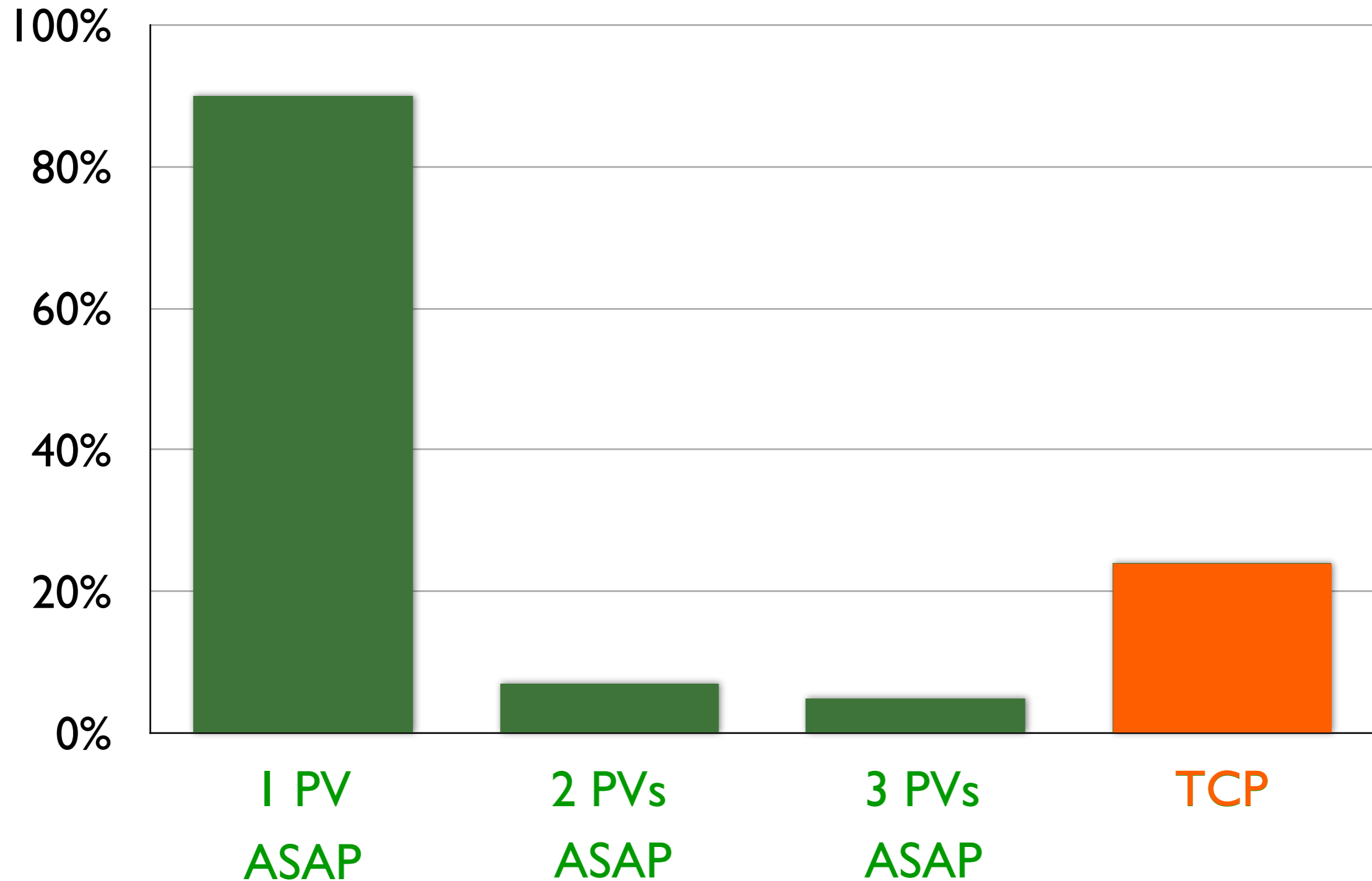
**Problem:** Eavesdropping

**Solution:** Multiple PVs



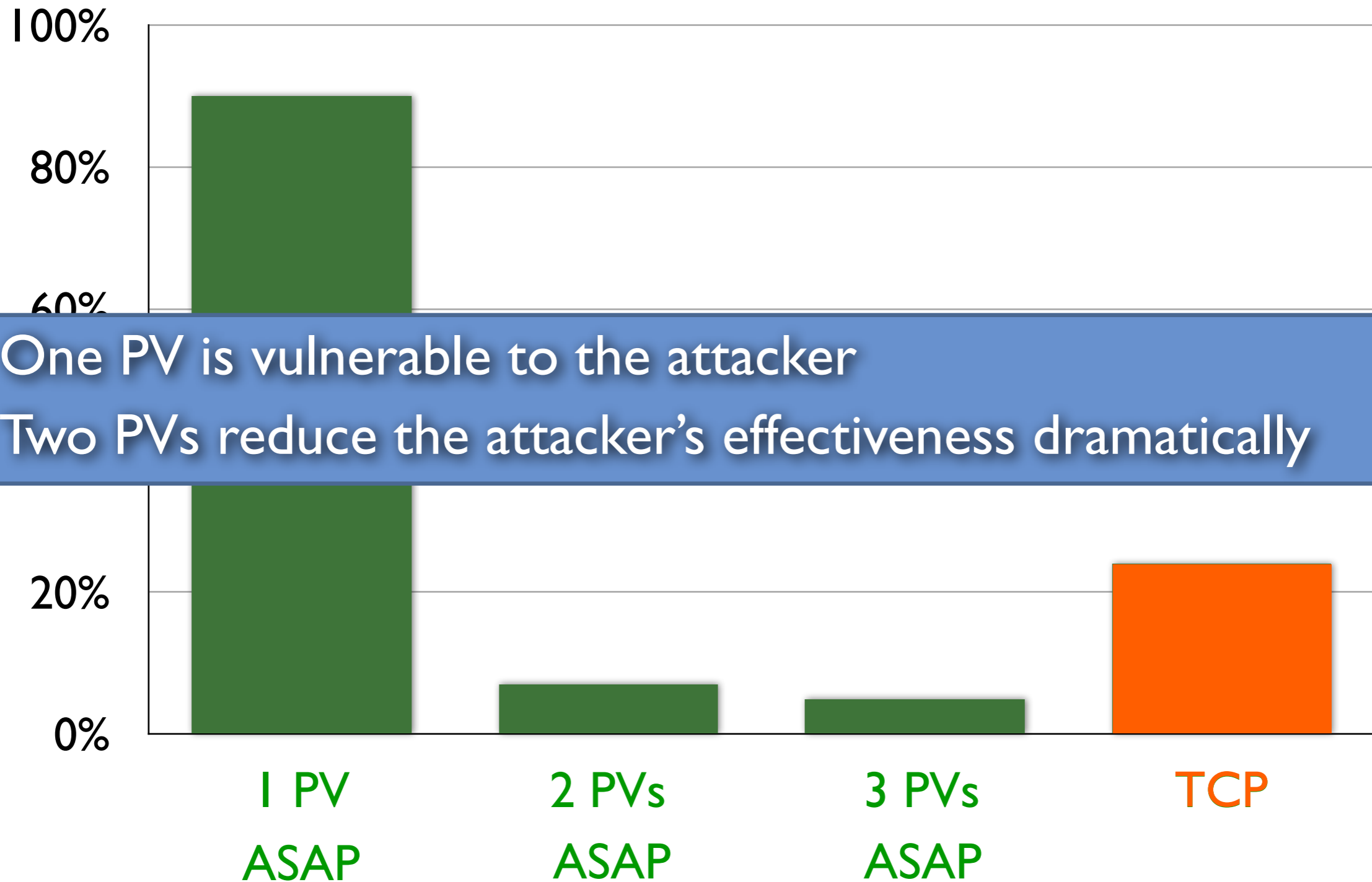
# Results: Eavesdropping Defense

## Attackability



# Results: Eavesdropping Defense

## Attackability



- One PV is vulnerable to the attacker
- Two PVs reduce the attacker's effectiveness dramatically

# Deployability

ASAP requires changes to some parts of the Internet

But only to devices under the client's and server's control

- Auth. DNS, End host clients, Server

This simplifies deployability

# Experimental Results

# Experimental Results

ASAP has both benefits and costs

# Experimental Results

ASAP has both benefits and costs

- Improved latency, at the cost of some additional processing overhead

# Experimental Results

ASAP has both benefits and costs

- Improved latency, at the cost of some additional processing overhead

Metrics

# Experimental Results

ASAP has both benefits and costs

- Improved latency, at the cost of some additional processing overhead

## Metrics

- Reduced latency

# Experimental Results

ASAP has both benefits and costs

- Improved latency, at the cost of some additional processing overhead

## Metrics

- Reduced latency
- Processing overhead

# Experimental Results

ASAP has both benefits and costs

- Improved latency, at the cost of some additional processing overhead

Metrics

- Reduced latency
- Processing overhead

Implementation

# Experimental Results

ASAP has both benefits and costs

- Improved latency, at the cost of some additional processing overhead

## Metrics

- Reduced latency
- Processing overhead

## Implementation

- Linux Kernel

# Experimental Results

ASAP has both benefits and costs

- Improved latency, at the cost of some additional processing overhead

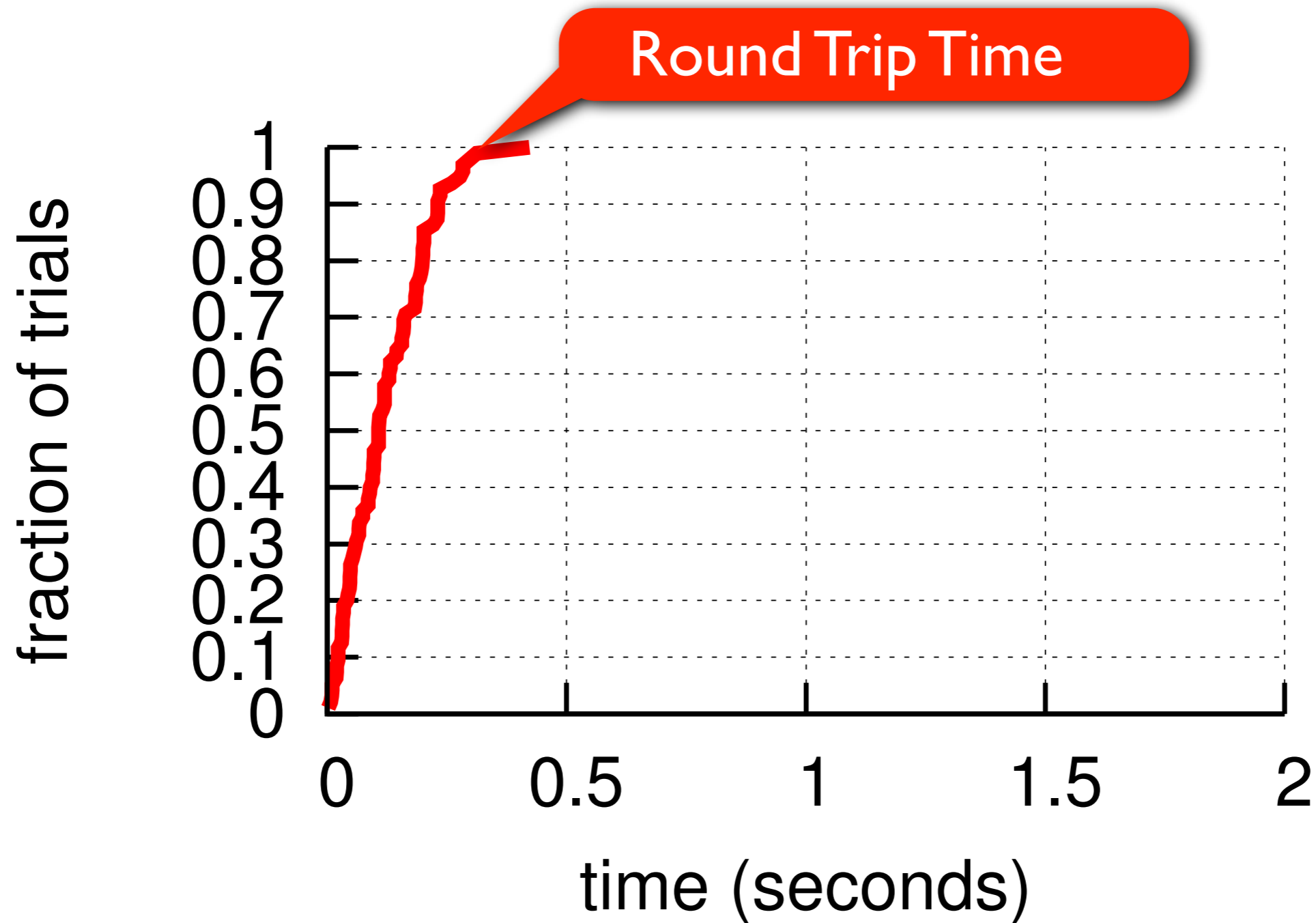
## Metrics

- Reduced latency
- Processing overhead

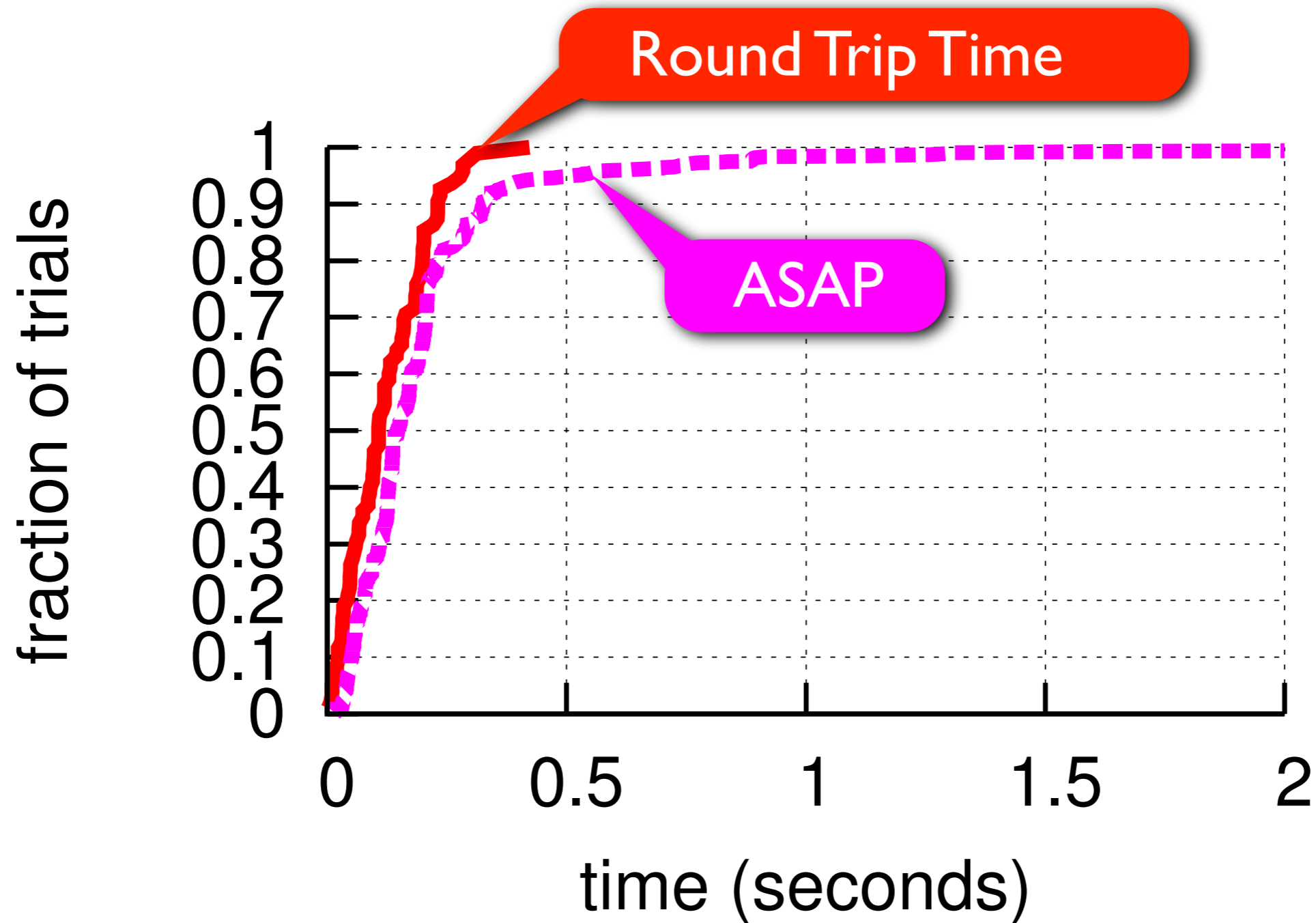
## Implementation

- Linux Kernel
- User-space on planet-lab

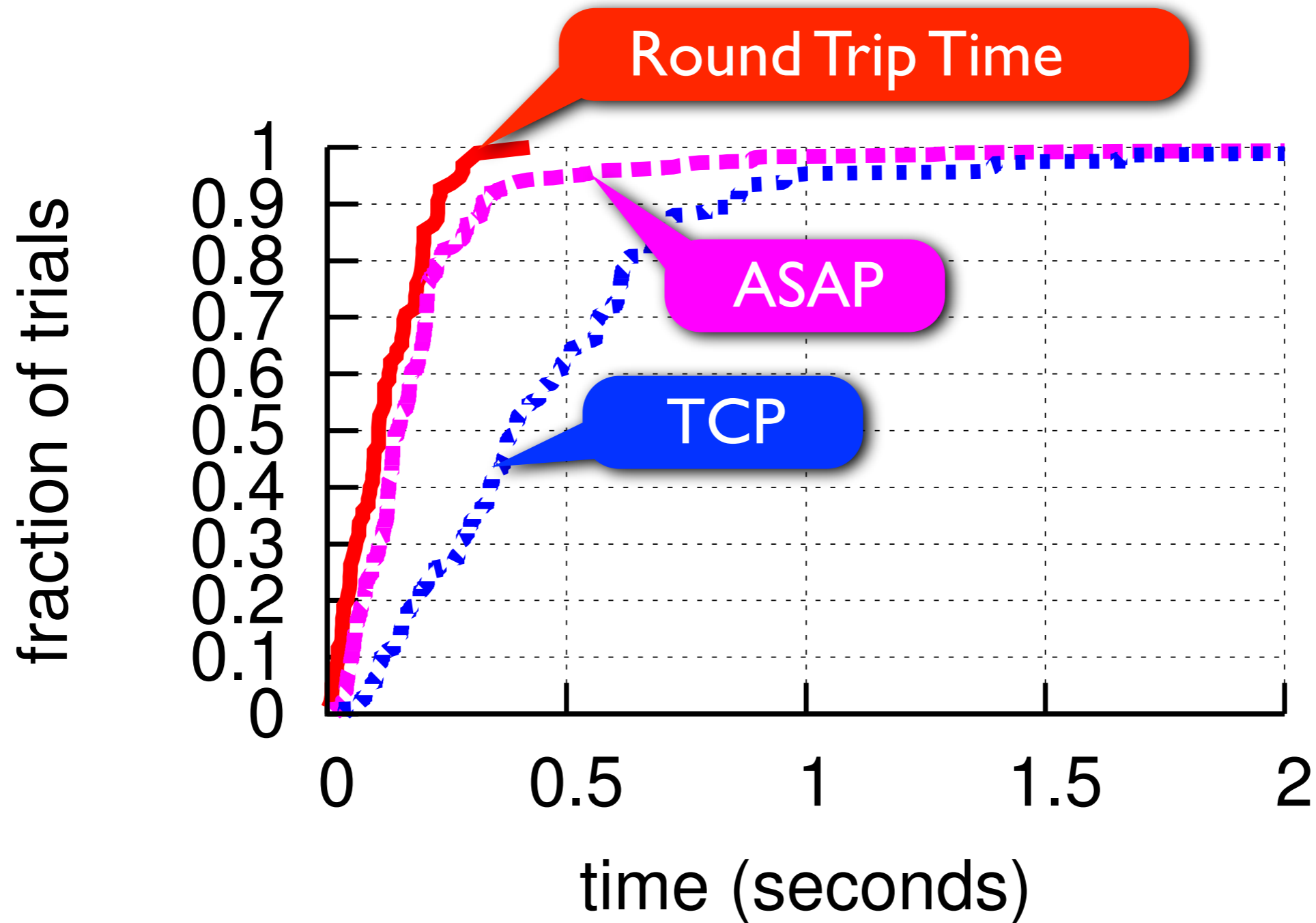
# Results: Latency of a Download



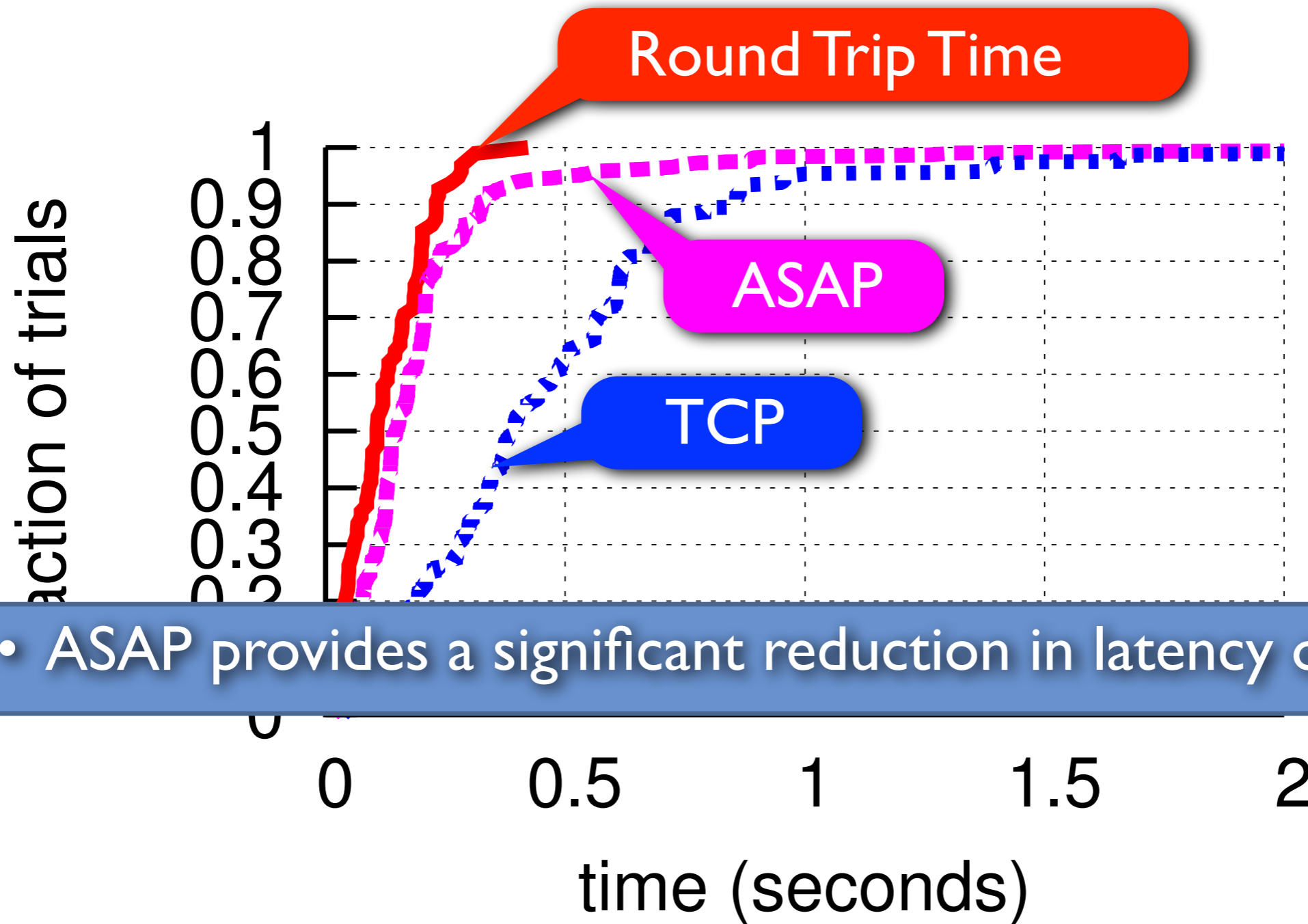
# Results: Latency of a Download



# Results: Latency of a Download

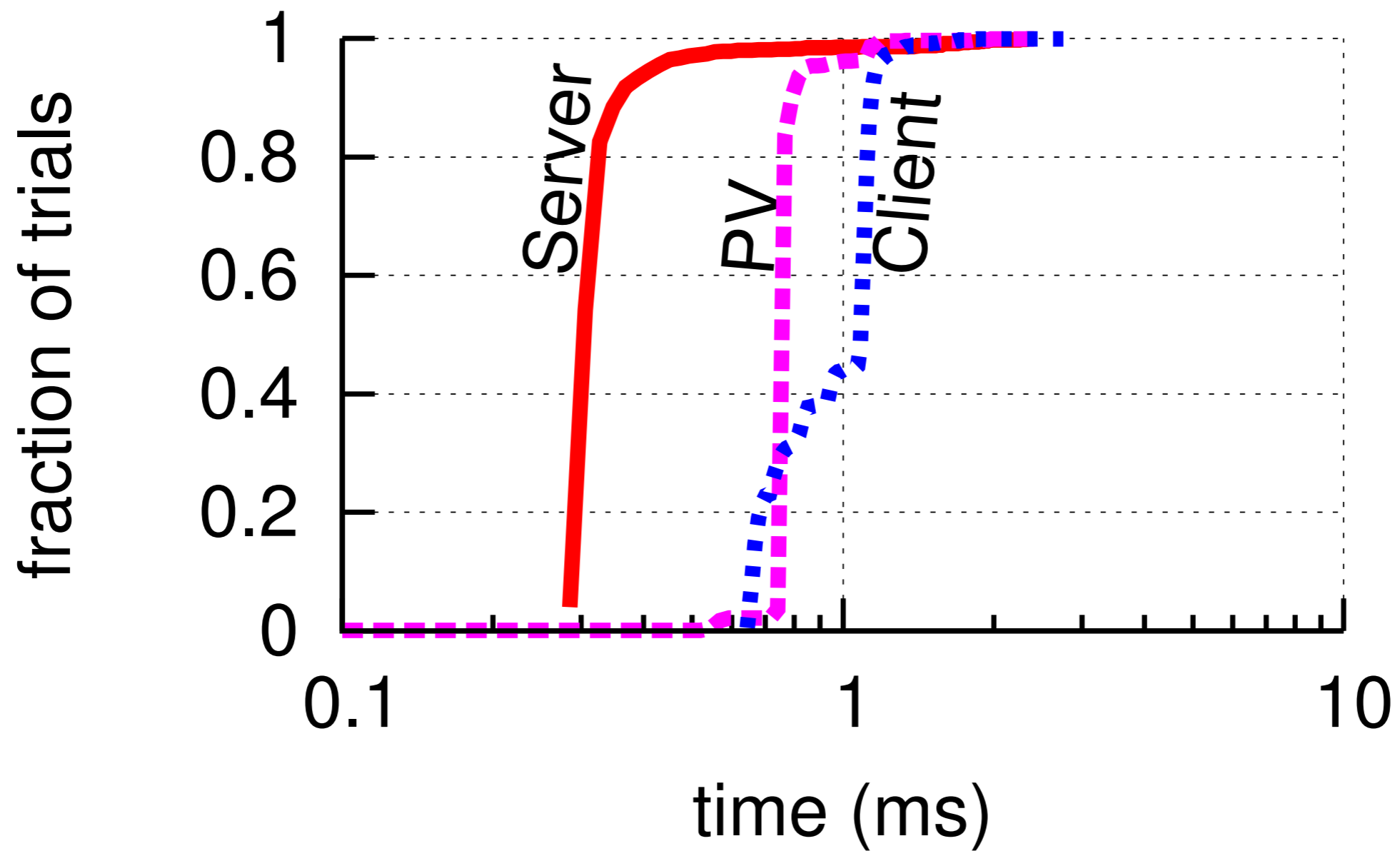


# Results: Latency of a Download

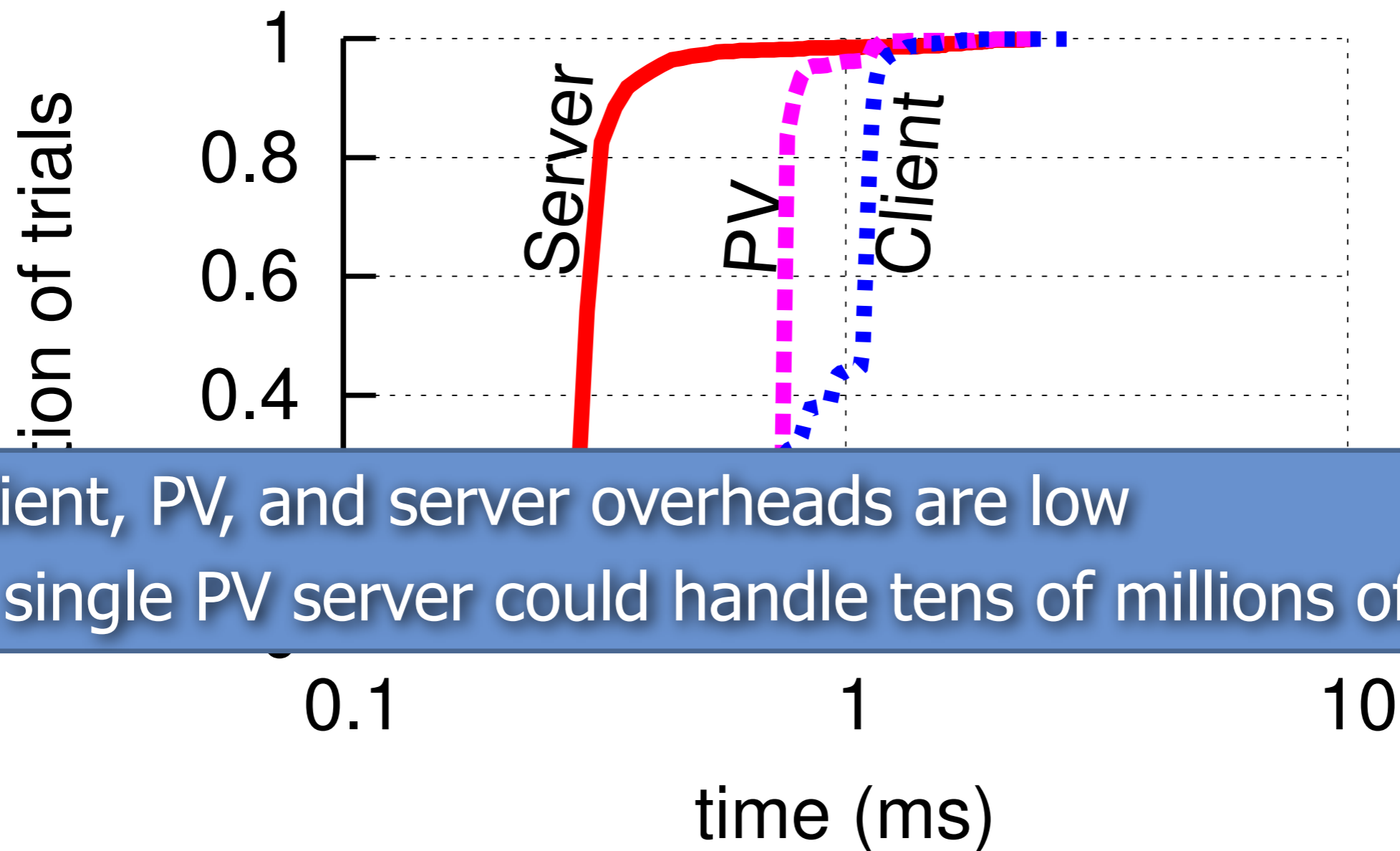


- ASAP provides a significant reduction in latency over TCP

# Results: Computational Overhead



# Results: Computational Overhead



- Client, PV, and server overheads are low
- A single PV server could handle tens of millions of clients

# Related Work: TCP Fast Open

	ASAP	TCP Fast Open
Similarities	Bypass three way handshake Use a security proof verifying address ownership	
Differences	Piggyback connection setup on name resolution Require changes to Auth. DNS	Name resolution not involved
	Certificates for security proof (can be used across domains)	Cookies for security proof (easier to compute)

# Conclusions

# Conclusions

**ASAP merges functions of DNS and TCP, and eliminates 3WH to reduce latency of interactive requests**

# Conclusions

ASAP merges functions of DNS and TCP, and eliminates 3WH to reduce latency of interactive requests

- Reduces latency by up to 2 RTTs

# Conclusions

ASAP merges functions of DNS and TCP, and eliminates 3WH to reduce latency of interactive requests

- Reduces latency by up to 2 RTTs
- Retains protection against attacks

# Conclusions

ASAP merges functions of DNS and TCP, and eliminates 3WH to reduce latency of interactive requests

- Reduces latency by up to 2 RTTs
- Retains protection against attacks

Implementation:

# Conclusions

ASAP merges functions of DNS and TCP, and eliminates 3WH to reduce latency of interactive requests

- Reduces latency by up to 2 RTTs
- Retains protection against attacks

Implementation:

- User-space and Kernel-space

# Conclusions

ASAP merges functions of DNS and TCP, and eliminates 3WH to reduce latency of interactive requests

- Reduces latency by up to 2 RTTs
- Retains protection against attacks

Implementation:

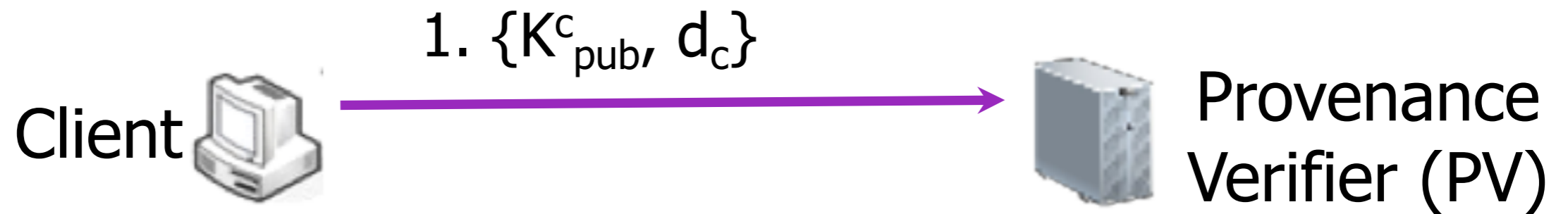
- User-space and Kernel-space
- Available at: <http://www.cs.illinois.edu/~wzhou10/asap.tar.gz>

# Backup: Security Mechanism



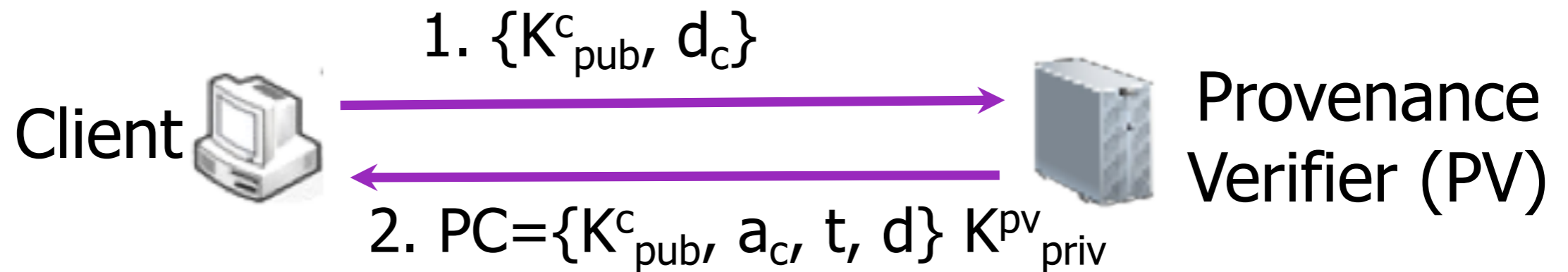
Web Server

# Backup: Security Mechanism



Web Server

# Backup: Security Mechanism



Web Server

# Backup: Security Mechanism

