The Case for Prefetching and Prevalidating TLS Server Certificates

Emily Stark, Lin-Shung Huang, Dinesh Israni, Collin Jackson, Dan Boneh

February 8, 2012

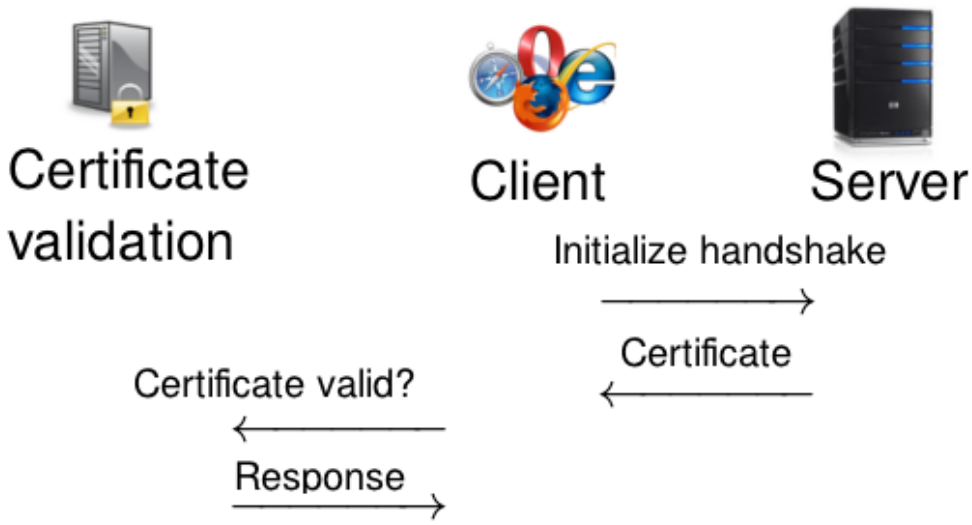
Transport Layer Security

- Want to secure traffic between web browsers and servers
- One problem is latency from TLS handshake

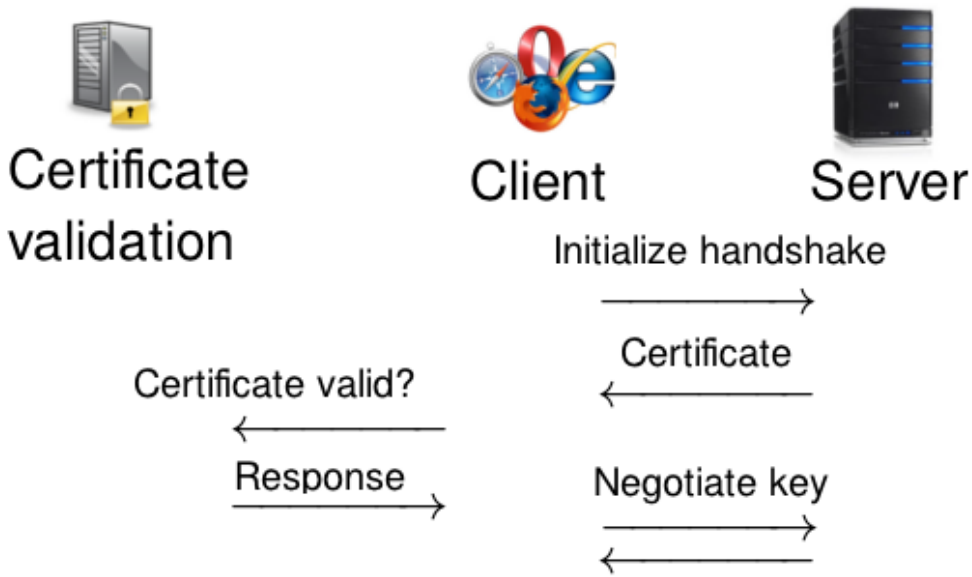
The TLS handshake

The TLS handshake

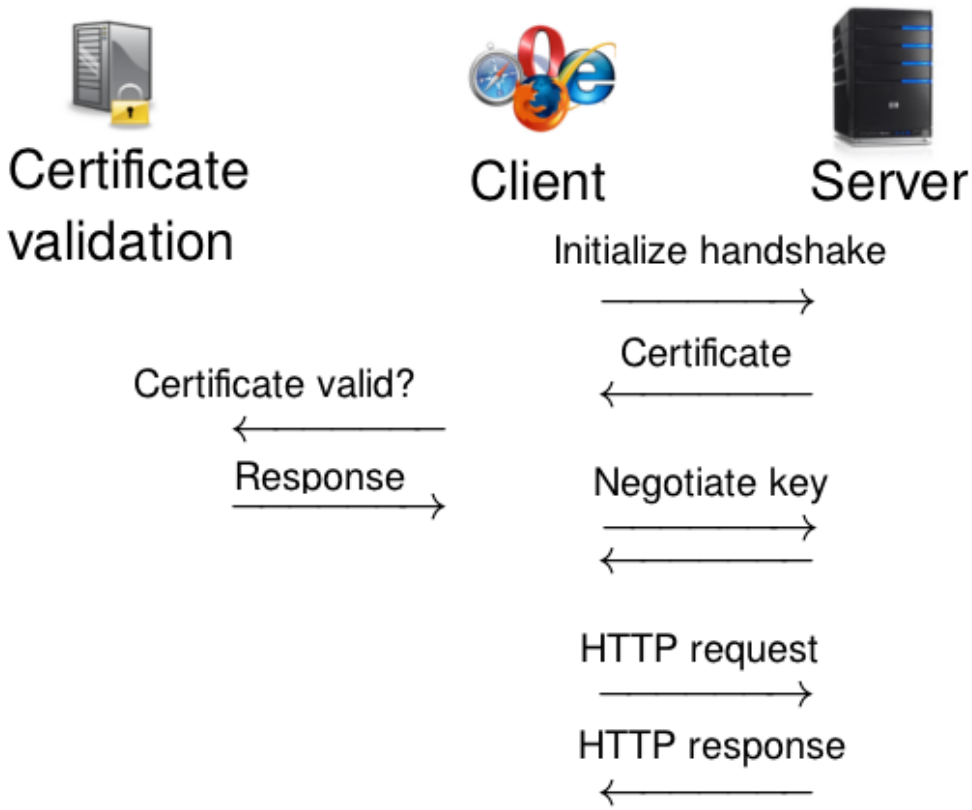
The TLS handshake



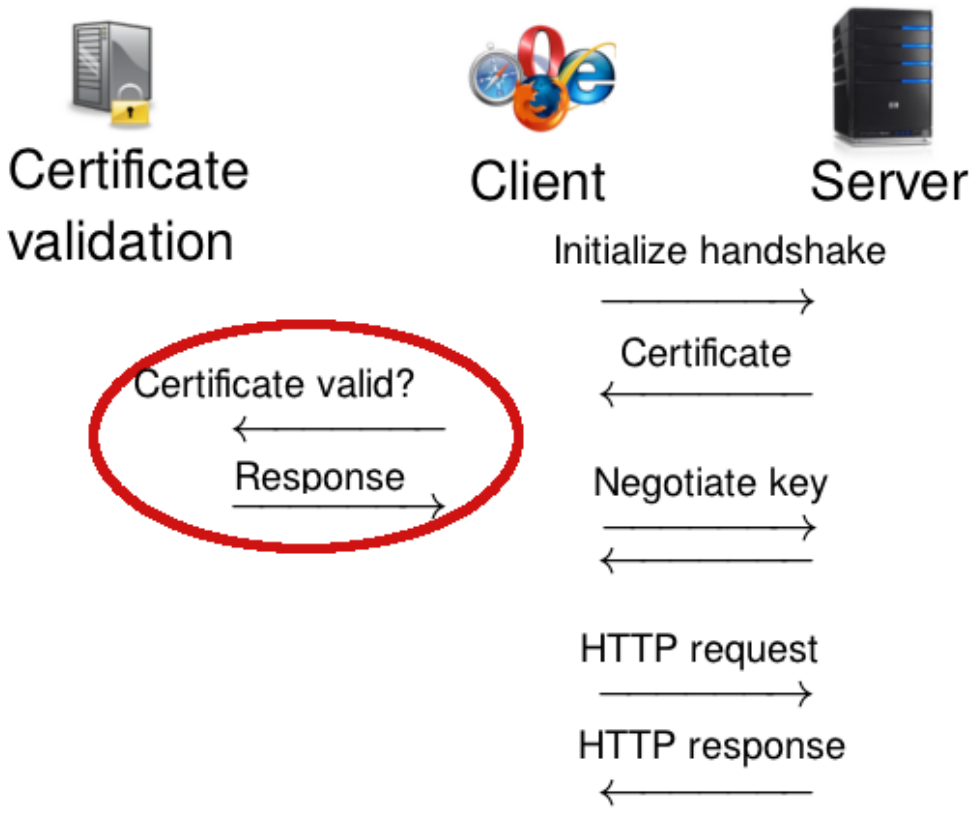
The TLS handshake



The TLS handshake



The TLS handshake



Certificate validation: OCSP

- Online certificate status protocol

Certificate validation: OCSP

- Online certificate status protocol
- Server certificate specifies OCSP responder

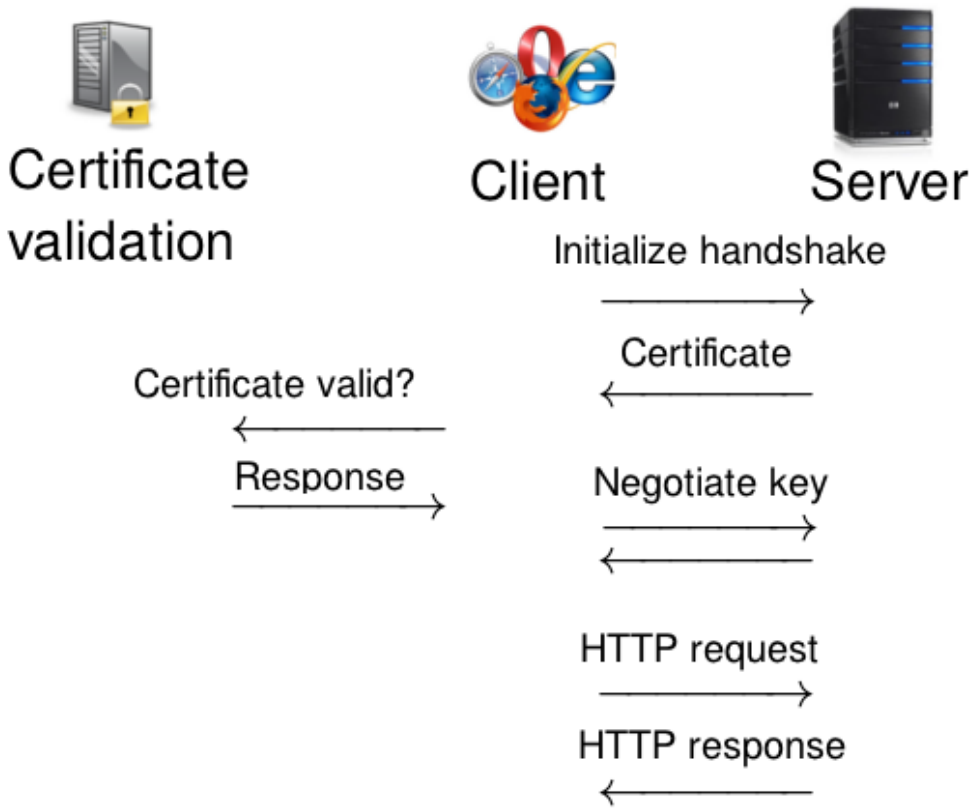
Certificate validation: OCSP

- Online certificate status protocol
- Server certificate specifies OCSP responder
- Clients asks responder whether cert is valid

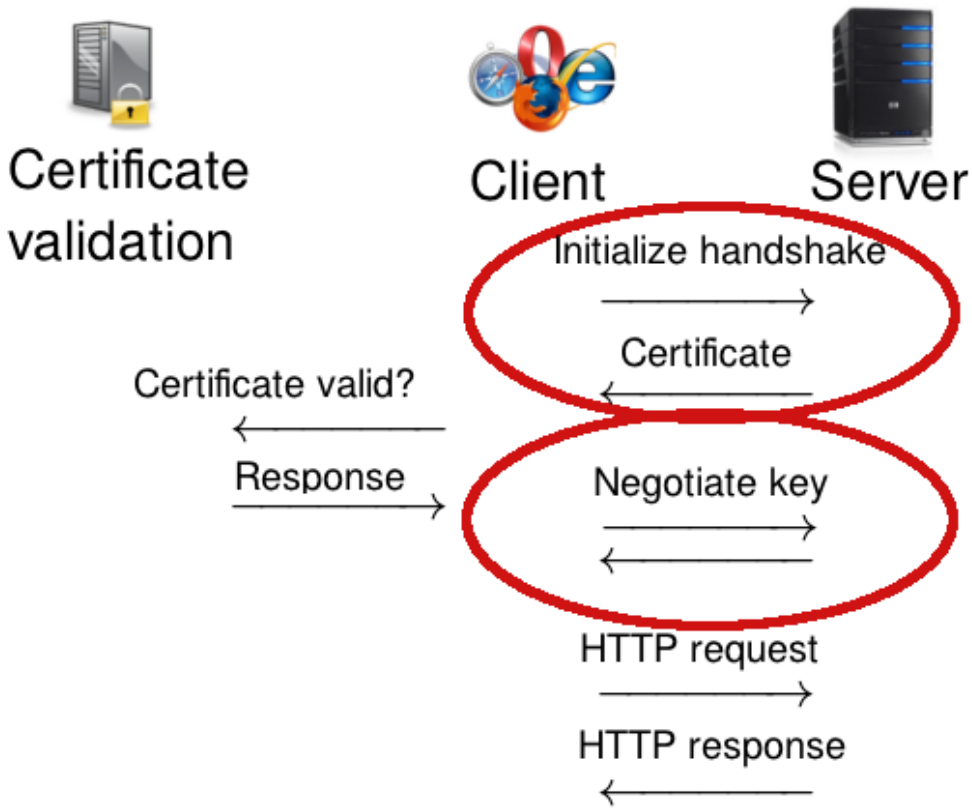
Certificate validation: OCSP

- Online certificate status protocol
- Server certificate specifies OCSP responder
- Clients asks responder whether cert is valid
- Responder specifies how long response is valid for

The TLS handshake



The TLS handshake



Removing round trips

Previous proposal, TLS Snap Start

- Zero round trip handshake

Removing round trips

Removing round trips

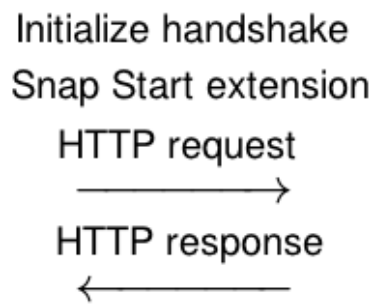
Initialize handshake
Snap Start extension
HTTP request
→
HTTP response
←

Snap Start challenges

- Client must know server certificate
  - Cached from previous visit

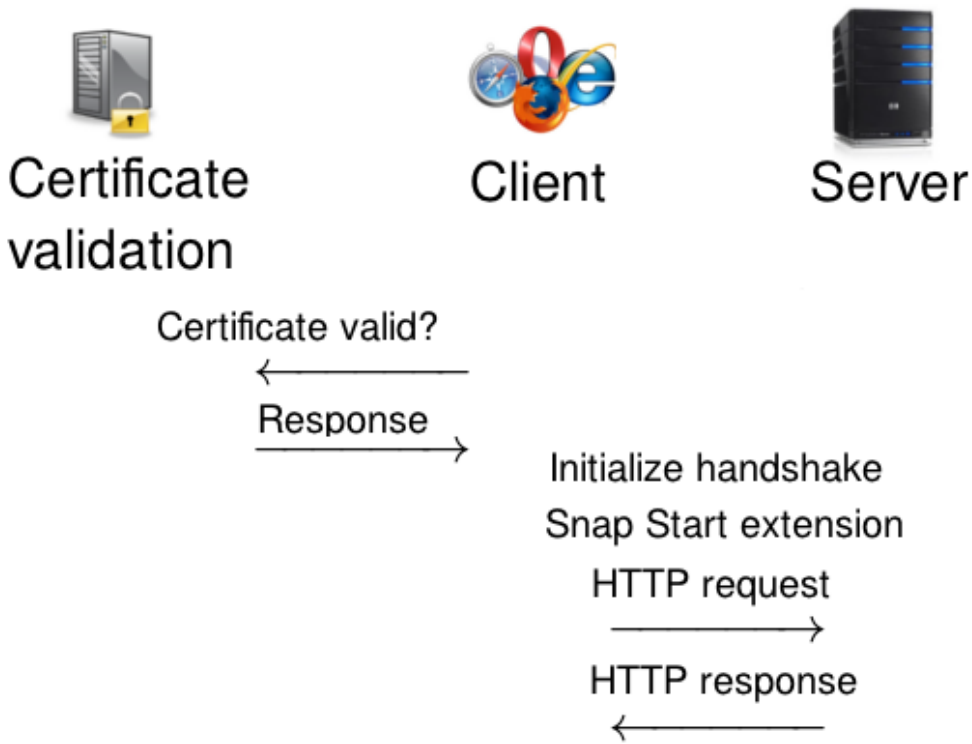
Snap Start challenges

- Client must know server certificate
  - Cached from previous visit
- Revocation checking is still slow



Snap Start challenges

- Client must know server certificate
  - Cached from previous visit
- Revocation checking is still slow



Problem

- TLS imposes extra latency due to retrieving and validating server certificate
  - How to obtain certificate to do Snap Start handshake?
  - How to validate without extra latency?

Contribution

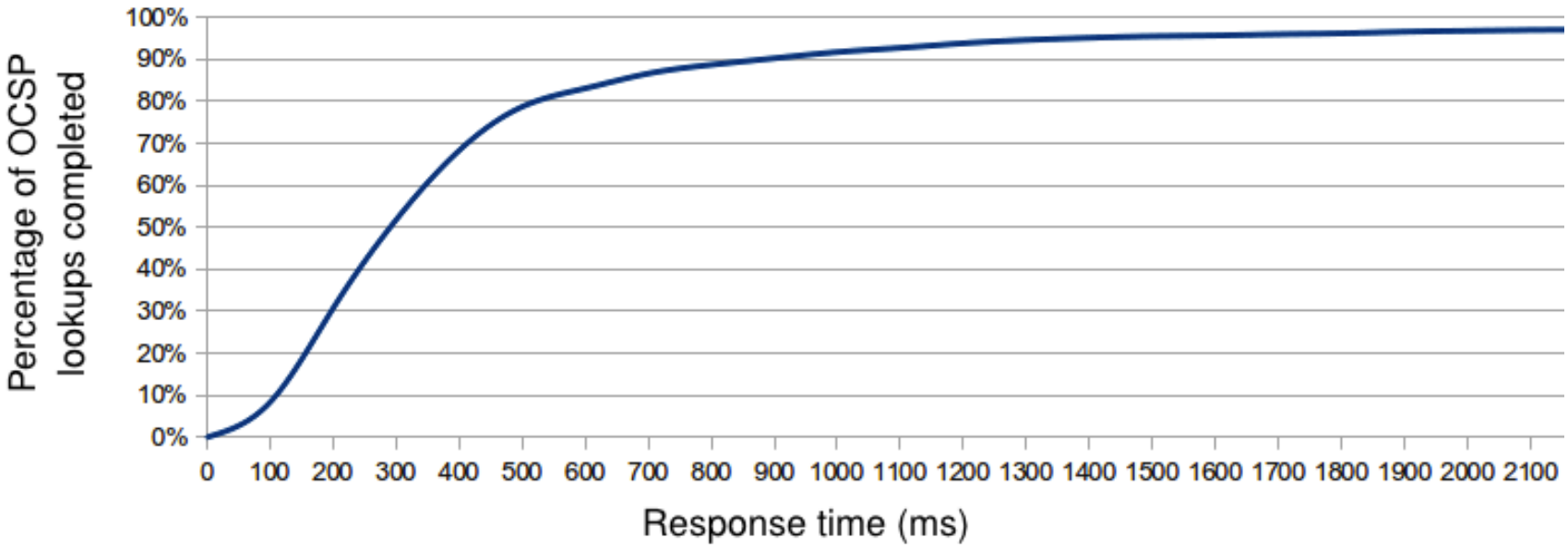
- Real world study of OCSP response times
- Certificate prefetching and prevalidation
  - Propose four prefetching strategies
  - Analysis of effectiveness
  - Prototype implementation

How long does OCSP take in the real world?

- Experimental setup
  - OCSP response times collected from users running Perspectives browser extensions
  - 242 clients, 4474 certificates, 24 responders

OCSP in the wild

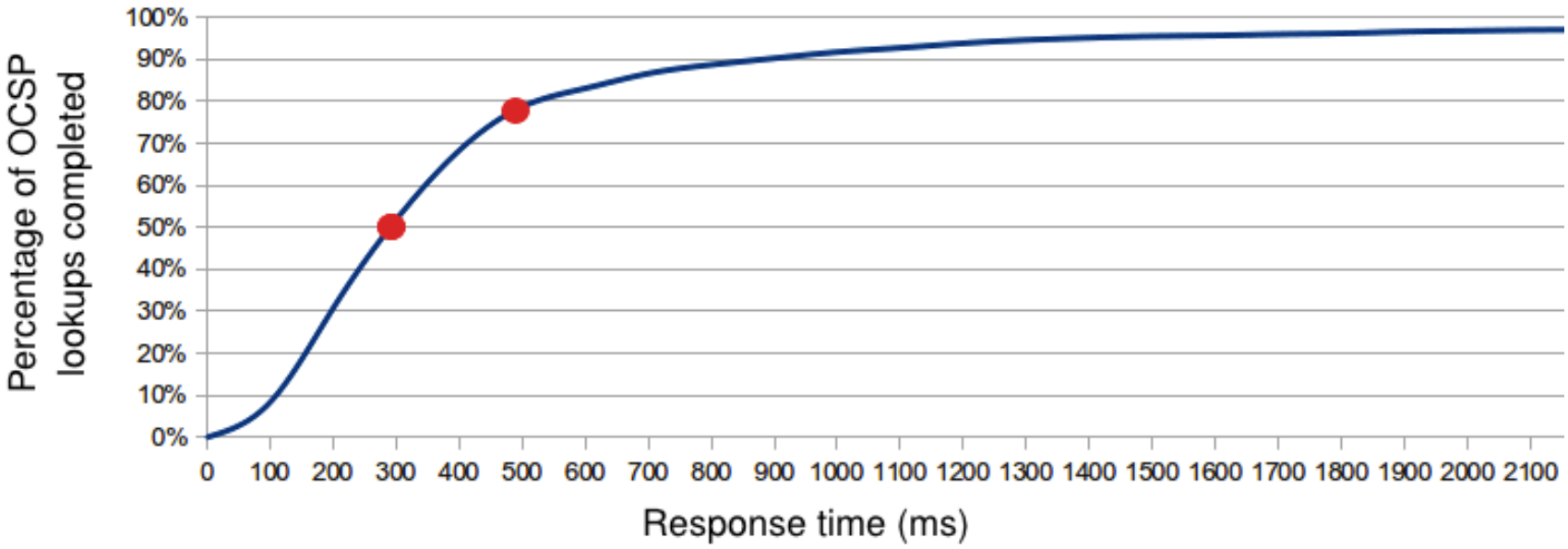
CDF of OCSP response time:



Median: 291 ms, mean: 498 ms

OCSP in the wild

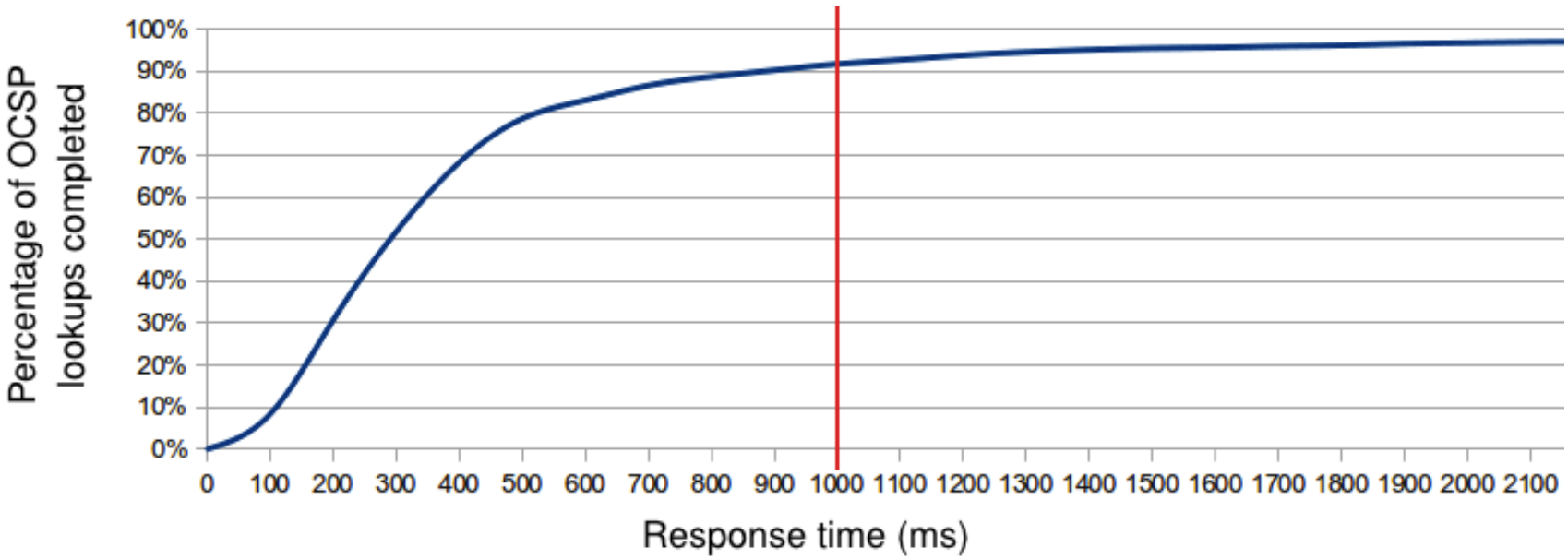
CDF of OCSP response time:



Median: 291 ms, mean: 498 ms

OCSP in the wild

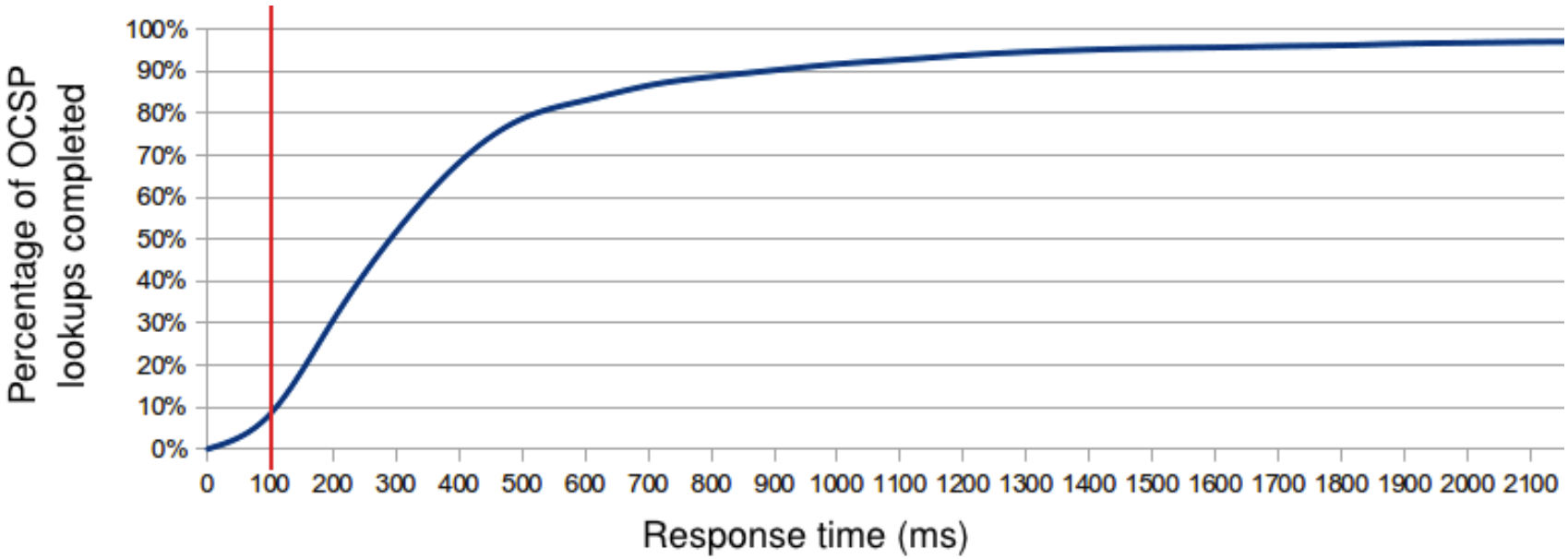
CDF of OCSP response time:



Median: 291 ms, mean: 498 ms

OCSP in the wild

CDF of OCSP response time:



Median: 291 ms, mean: 498 ms

Design

- Prefetch certificates
  - Enables Snap Start handshakes more frequently
- Prevalidate certificates
  - Removes OCSP lookup from critical path

Design questions

- When to prefetch? When to prevalidate?
- How to obtain certificate?

When to prefetch

Ideas borrowed from DNS prefetching:

- DNS prefetching triggers are effective for certs

When to prefetch

Ideas borrowed from DNS prefetching:

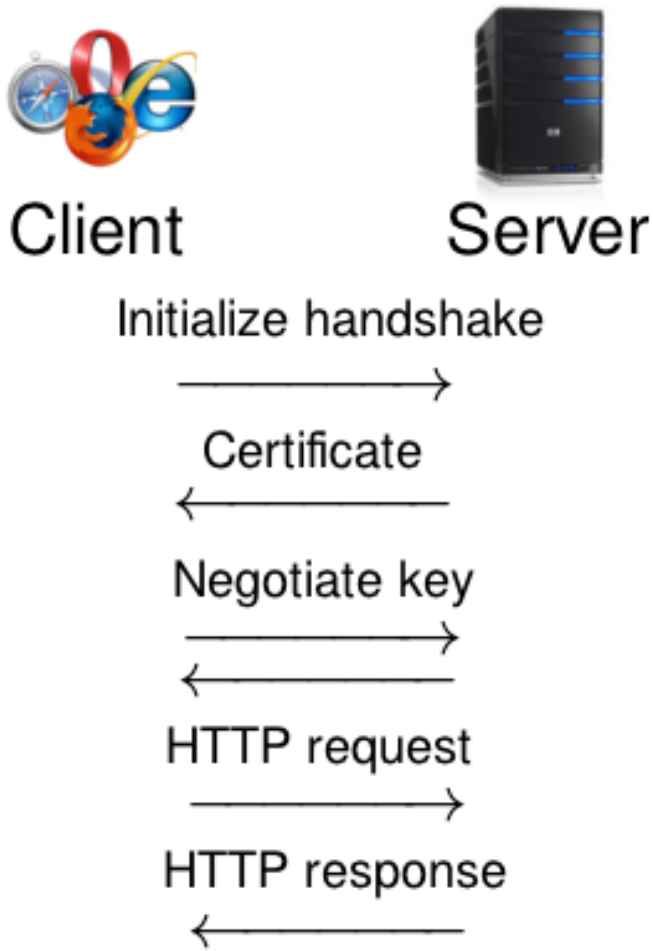
- DNS prefetching triggers are effective for certs
- Could be deployed with HTML hints for effective prefetching

How to prefetch

- Goal: Obtain server certificate
- Challenge: Full TLS handshake is expensive
- Four proposed methods that are more efficient

Prefetching methods

Option 1: Truncated handshake



Prefetching methods

Option 1: Truncated handshake

Prefetching methods

Option 1: Truncated handshake

- No public key crypto!
- Server admin does nothing
- But implementation requires new API in TLS layer

Prefetching methods

Option 2: HTTP GET request

e.g., to `http://www.domain.com/cert`

- Much less load than full TLS handshake, but still impacts the server

Prefetching methods

What if we want no additional load on server?

Prefetching methods

What if we want no additional load on server?

Option 3: Retrieve from CDN

- HTTP GET request, avoid hitting web server

Prefetching methods

What if we want no additional load on server?

Option 4: Retrieve from DNS

- DNS TXT record can store certificate
- No impact on web server

Prevalidation

- After prefetching cert, prevalidate it
  - Normal OCSP lookup

Prototype

- Prefetching and prevalidating in Chromium
- Piggyback on DNS prefetching architecture
- DNS and HTTP GET prefetching

Analysis

- How much does prefetching and prevalidating affect handshake latency?

Handshake latency

- Normal TLS handshake: **122 ms**

  ↓ Remove round trips by prefetching cert and using Snap Start

- Snap Start, unvalidated cert: **83 ms**

  ↓ Remove OCSP validation by prevalidating cert

- Snap Start, prevalidated cert: **30 ms**

Server: Ubuntu 10.04, 256MB, Apache 2.2.17, Client: Ubuntu 10.04, 1GB RAM

HTTP GET request: 16 ms

Conclusions

- OCSP latency matters, especially when handshakes have fewer RTTs
- Need prefetched certificate to **enable Snap Start** and for **OCSP prevalidation**
- 4 proposed strategies for prefetching certs
- Reduce TLS handshake by two RTTs and OCSP response time (factor of 4 in our experiments)