The Cost of the "S" in HTTPS

Carnegie Mellon University
POLITECNICO DI TORINO
Telefónica

David Naylor[+], Alessandro Finamore[*], Ilias Leontiadis[✸],
Yan Grunenberger[✸], Marco Mellia[*], Maurizio Munafò[+],
Konstantina Papagiannaki[✸], and Peter Steenkiste[+]

MOTIVATION

The use of HTTPS is increasing and may become the default in HTTP 2.0. The privacy and security benefits of ubiquitous encryption are relatively clear, but what are the costs?

deployment / 1

HTTPS Usage Trends

We examine recent HTTPS usage trends by analyzing per-flow logs from a vantage point monitoring the traffic of about 25,000 residential ADSL customers in a major European residential ISP ("Res-ISP").
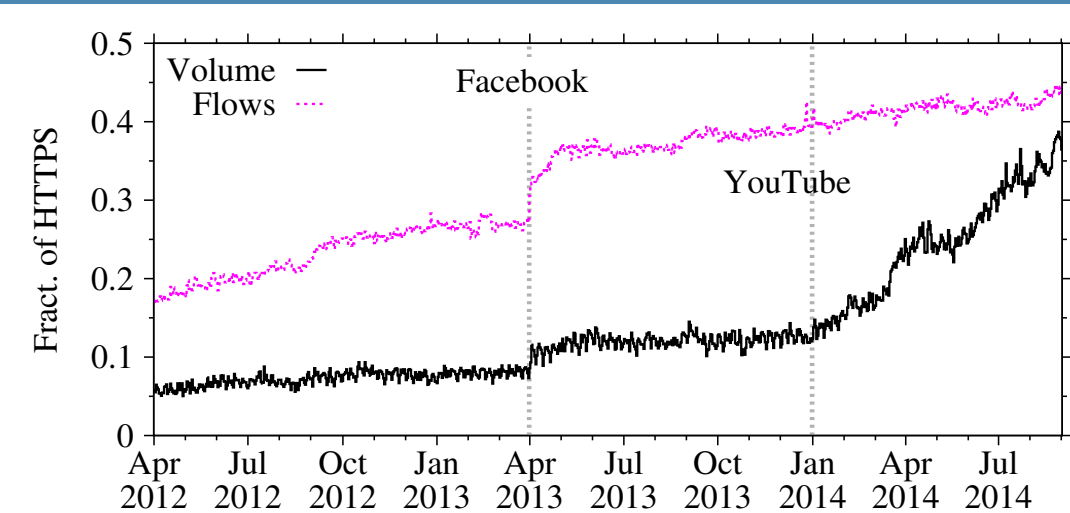
◄ *Flow and Volume Shares*

Flow Share: has more than doubled in two years.
Volume Share: growing more slowly, since large content is typically unencrypted, though YouTube is changing the landscape.
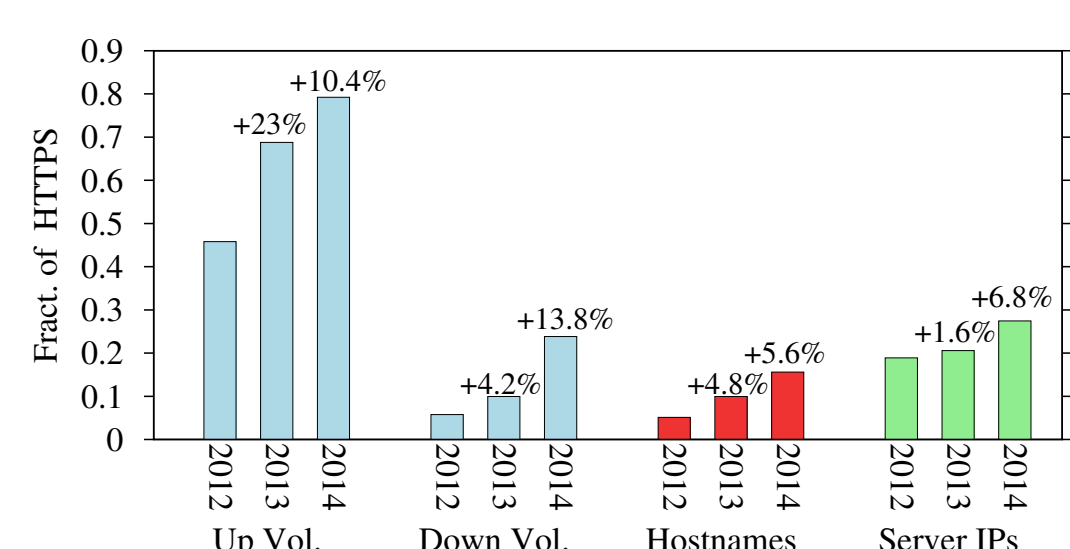
◄ *Upload and Download Volume*

HTTPS accounts for **80% of the upload volume** in 2014, but only **25% of the download volume**, since privacy-sensitive information tends to be uploaded more than downloaded. HTTPS download volume is accelerating, however, driven in part by YouTube.
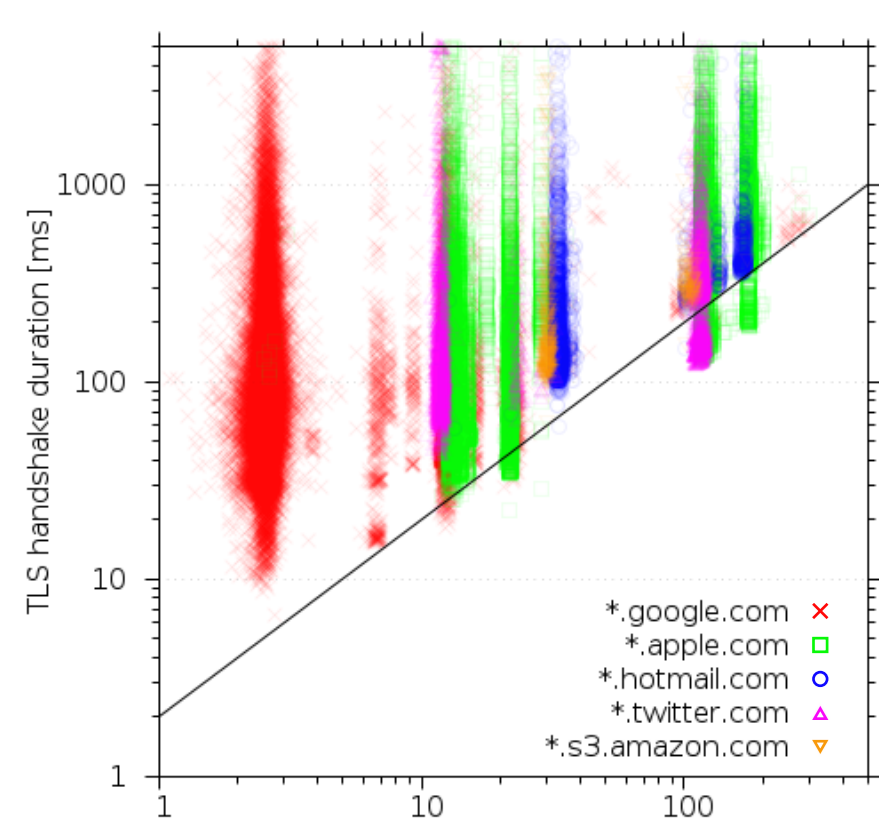
TAKEAWAY

HTTPS accounts for 50% of all HTTP connections and is no longer used solely for small objects, suggesting that the cost of deployment is justifiable and manageable for many services.

load time / 2
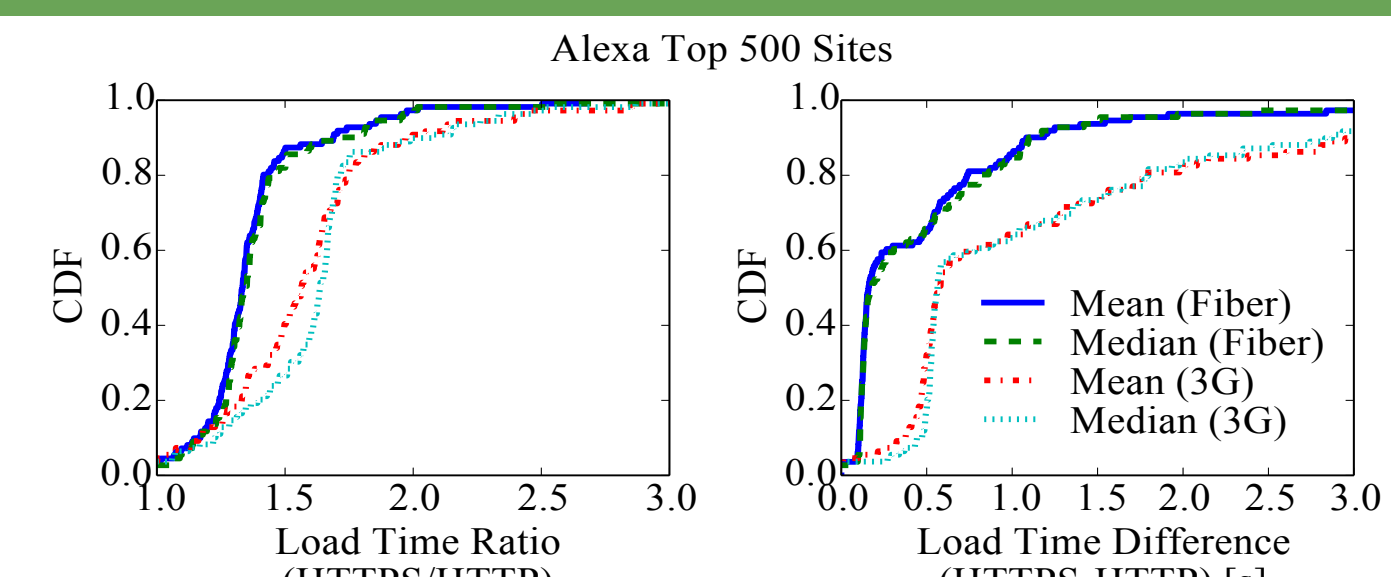
▲ *Page Load Time*

Inflation on 3G:	Inflation on fiber:
> 500 ms for 90%	> 200 ms for 50%
> 1.5 s for 25%	> 500 ms for 40%

◄ *TLS Handshake Latency*

→ Vertical clusters of points likely represent data centers.

→ Regardless of RTT, each cluster contains samples with long handshakes (e.g., several seconds).

→ Only 30% of the connections used TLS fast negotiation.

Source: Res-ISP, April 3, 2014
(External RTT: RTT between vantage point and remote server)

4% of the clients experience at least **one handshake > 300 ms.** Of these, **50% (75%)** have an internal **RTT of 51 ms (97 ms).** (So a spotty connection is not to blame.)

TAKEAWAY

The extra latency introduced by HTTPS is not negligible, especially in a world where one second could cost 1.6 billion in sales.
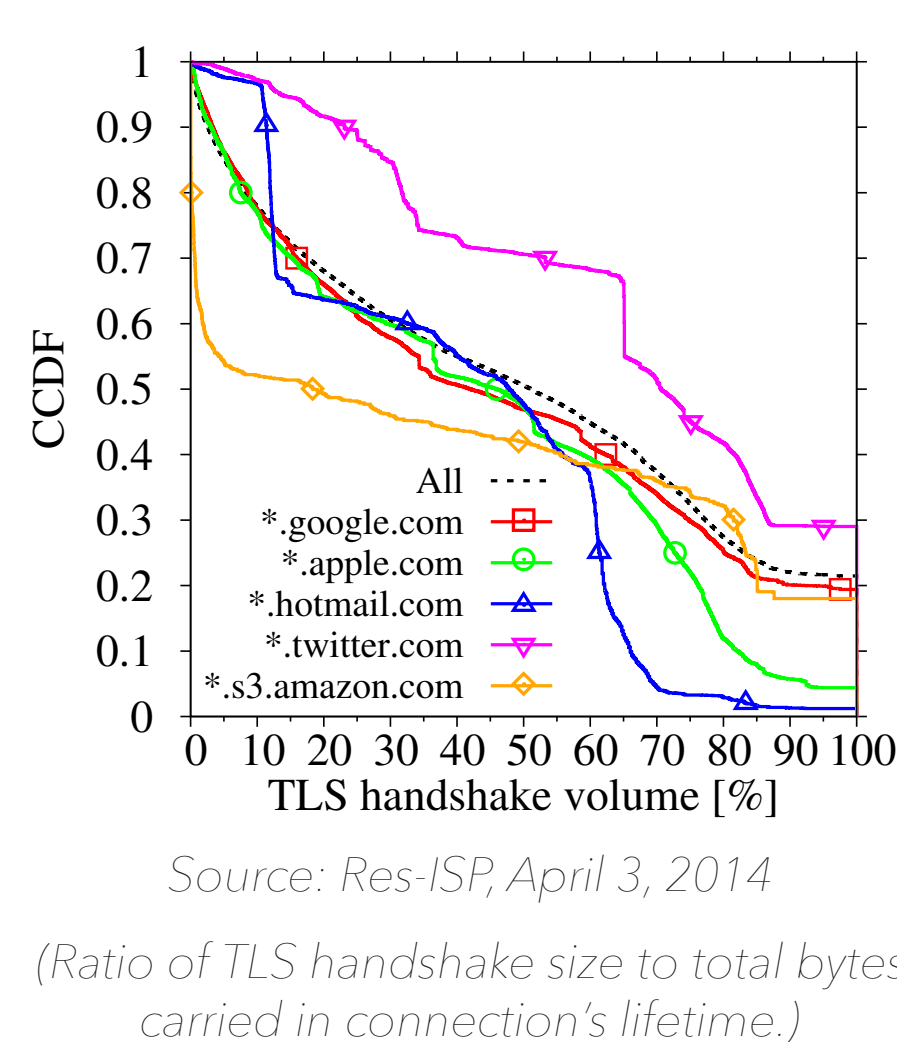
data usage / 3

Caching & Compression

Encryption **prevents proxies from caching and compressing content.** We analyze logs from a transparent proxy in a major European mobile carrier.

| Compression Ratio:	Cache Hit Ratio:
28.5%	**14.9%**
Avg. daily per-user savings:	Avg. daily ISP savings:
2.1 MB	**16 TB**

◄ *TLS Handshake Volume*

→ Many TLS connections are not heavily used. For half, the handshake is over 50% of the total data exchanged.

→ "Light" services like Twitter are least efficient; "heavy" services like Amazon S3 are most.

→ Some services mask latency by "pre-opening" connections; if the connection is never used, overhead is 100%.

Source: Res-ISP, April 3, 2014
(Ratio of TLS handshake size to total bytes carried in connection's lifetime.)

TAKEAWAY

Most users are unlikely to notice significant jumps in data usage due to loss of compression, but ISPs stand to see a large increase in upstream traffic due to loss of caching.

energy consumption / 4

Synthetic Benchmarks
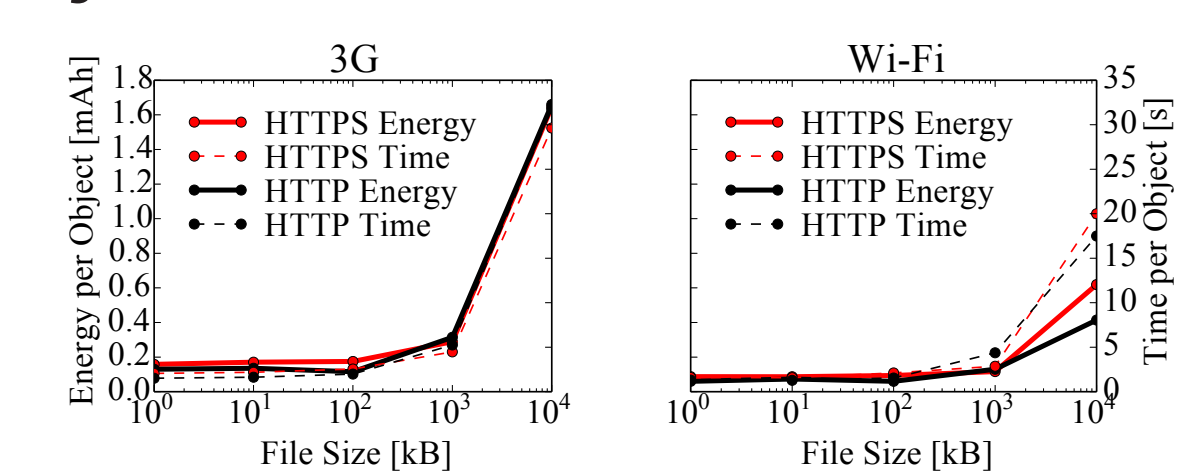

▲ *Energy Consumption*

TLS' cryptographic operations have no noticeable impact on battery life.

Results from instrumented Galaxy S II.

Video Playback

We tested energy consumed while playing four YouTube videos. On HTTPS, the connection traverses our carrier's proxy; on HTTPS it does not.
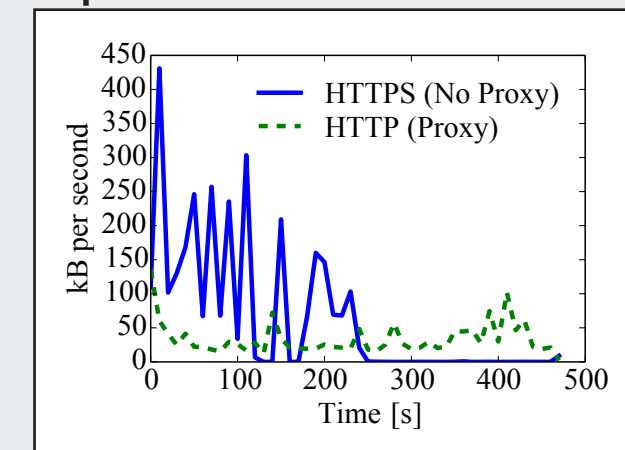
NOTE: Experiments use YouTube's desktop site, since the mobile site does not support HTTPS.
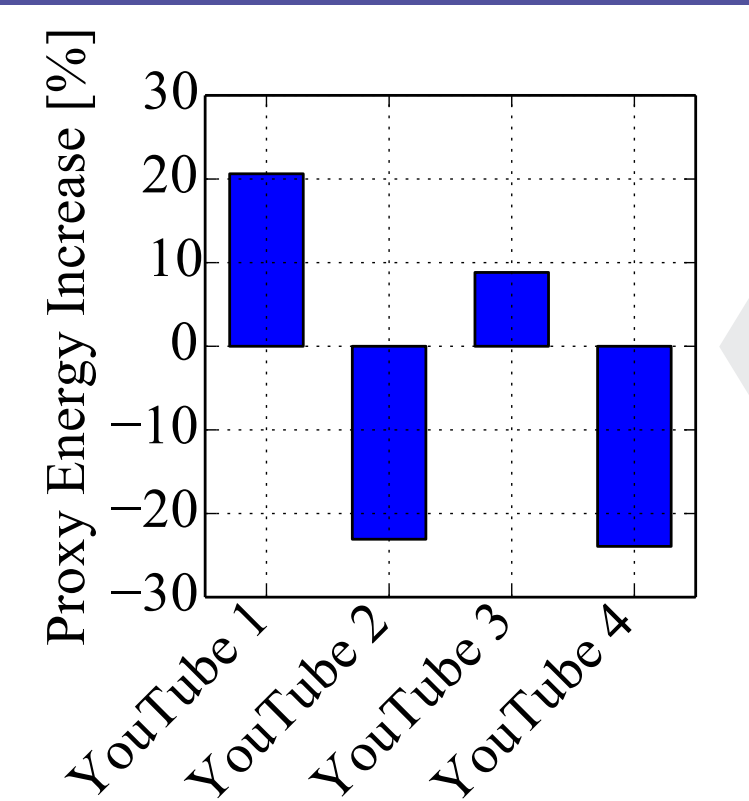
▲ *Increase on HTTP (w/ proxy) over HTTPS (w/o proxy)*

❶ **Proxy can hurt**
Proxy throttles download rate, using more energy due to increased radio uptime.

❷ **Proxy can help**
Proxy modifies GETs to request `mp4` encoding in place of `webm`. Our phone has hardware support for `mp4`.

◄ *Throttling*

TAKEAWAY

HTTPS' cryptographic operations have almost no impact on energy costs, but the loss of proxies can significantly impact battery life (positively and negatively).

value-added services / 5

Many middleboxes **rely on packet contents** and become **blind in the presence of encryption**:

- caching proxies
- compression proxies
- parental filters
- firewalls
- intrusion prevention systems
- forensics
- transcoders
- app-level load balancers
- ad/tracking cookie blocking

We've seen proxies **can help**:

→ Reduce user data usage

→ Reduce ISP data usage

→ Reduce energy consumption

Another Example: **Parental Filtering**

Internet Watch Foundation Blacklist:

5% pure domain or subdomain

95% of the list is useless if filter cannot see the full URL

TAKEAWAY

Though difficult to quantify, the loss of in-network services is potentially substantial; some of that functionality could be equally well performed on the client, while some may require a total rethink.