ANALYSIS OF HTTPS OVERHEAD AND MINIMAL

WEB CERTIFICATE CHAIN OF TRUST

By

RAKESH RAVISHANKAR

Bachelor of Engineering in Instrumentation Technology

Visvesvaraya Technological University

Belguam, India

2010

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2015

ANALYSIS OF HTTPS OVERHEAD AND MINIMAL

WEB CERTIFICATE CHAIN OF TRUST

Thesis Approved:

Dr. Eric David Chan-Tin

Thesis Adviser

Dr. K. M. George

Dr. Nohpill Park

ACKNOWLEDGEMENTS

I am very thankful to my parents who supported me throughout my life.

I would like to thank my advisor Dr. Eric Chan Tin, for giving the opportunity to work on this topic. I am very grateful for the assistance and support he provided me.

I would like to thank Dr. K. M. George and Dr. Nohpill Park, for providing their valuable inputs and guiding me.

Finally, I would like to thank all people, who supported me directly or indirectly.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: RAKESH RAVISHANKAR

Date of Degree: JULY, 2015

Title of Study: ANALYSIS OF HTTPS OVERHEAD AND MINIMAL WEB
CERTIFICATE CHAIN OF TRUST

Major Field: COMPUTER SCIENCE

Abstract: The popularity of the web is indisputable. With the recent revelations about NSA spying and the increased need for privacy and security, the default use of secure web through TLS/SSL connections has been highlighted. However, the pushback against enabling secure web connections by default is due to the increase in communication and processing time.

In this work, we quantify the communication time for http and https download times for the most popular websites. The average download time over http non-persistent connection is 2.72 seconds while the average download time over https non-persistent connection is 3.156 seconds. The overhead in using encryption is thus only 436 milliseconds (about 4 round trip times on the Internet) or 16.1% for non-persistent connections. And for persistent connections the overhead is 15%. We thus make the case that https should be enabled by default due to the very low communications overhead. With the recent hacks and breaches at various certificate authorities and no-longer-trusted certificate authorities, we also quantified which certificate authorities are most popular on the Internet. By only trusting ten certificate authorities, a webbrowser can access almost 80% of https-enabled websites. The number of trusted certificate authorities can thus be reduced from thousands to a few dozen.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

HTTPS (Hyper Text Transfer Protocol Secure) is the secured communication protocol over the internet. Over a decade, HTTPS has evolved as the de facto standard for secure web access. HTTPS is mainly used for secure e-commerce transactions that transmits potential personal data over the network. HTTPS creates a secure channel and provides authentication services and secure data transmission layering. HTTPS is based on the TLS encrypted protocol and a supporting public key infrastructure (PKI) composed of thousands of certificate authorities (CAs). These CAs are trusted by web browsers to verify the identity of the web servers. Chapter 2 provides a brief overview of the web, HTTPS, and the web PKI.

With increasing concern over the government surveillance and people’s increasing awareness for privacy, it is important to enable SSL/TLS encryption to secure the information sent over the internet. Most of the websites do not enable https by default (for example, Google’s Gmail and Yahoo’s mail services enabled https by default in 2014). The many reasons are overhead in communications and processing and the costs in maintaining and purchasing signed certificates. There has been a push recently in making HTTPS the default for all web traffic, such as the Electronic Frontier Foundation's HTTPS Everywhere extension [8].

Every webbrowser and operating system trust thousands of root certificate authorities. These authorities sign web-server's certificate so that clients can verify they are visiting the actual webserver's site and not some other website. Recent breaches at root certificate authorities such as Diginotar [3] and authorities issuing certificates impersonating other websites such as Google [4] have highlighted the fragile nature of the web PKI. Other services such as CloudFlare [5] also act as man-in-the-middle content distribution network (CDN) to improve performance.

The two goals of this project are

1. To determine the overall download time for plain-text (http) and secure (https) web connections of popular websites to show that the usage of secure web connections does not add much overhead to the download time. The recommendation then would be to use https as the default protocol for all web connections.

2. To analyze the certificate authorities (CAs) of popular websites and to determine if there is a minimal number of trusted certificates by the web browser to access most websites.

To achieve the first goal, we downloaded the whole webpage (all the related files such as scripts and images) using both http and https connections. We use the PlanetLab [17] network to determine if geographical location affect the download time. This study shows that the difference in download time using a https versus a http non-persistent connection is about 400 milliseconds. This difference is the communications overhead in using a secure connection. The overhead percentage is about 16%, which is not very significant. We also downloaded whole webpages on persistent and non-persistent connections to determine if the https overhead is similar. The difference between non-persistent and persistent connections is about 1%, which is the reduction in communications overhead due to reusing connections. We expect the overhead to go down as more webservers start using HTTP/2.0 as it supports concurrency on a single TCP and SSL connection by using multiplexing to allow more than one request at a time to send and receive data on a single connection.

With regards to the second goal, we downloaded the certificate chain for the most popular websites and listed the most used root certificate authorities. We found that a few root certificate authorities are used for the majority of the webservers' certificates. Trusting only 10 certificate authorities will allow a user to access almost 80% of the websites on the Internet without any certificate warning.

Details of our experimental setup are given in Chapter 3. Chapter 4 provide more results about our experiments. An overview of the related work is given in Chapter 5 and Chapter 6 provides a discussion for future work.

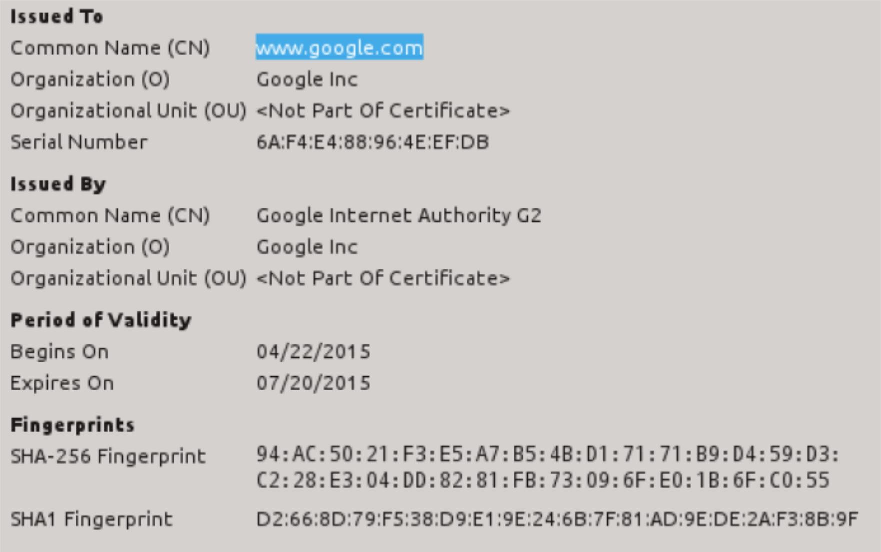
CHAPTER II

REVIEW OF LITERATURE

2.1 Background

A web page, static or dynamic is an information set that contains numerous types of information, which is able to be seen, heard or interact such as images(animated or static), audios and videos(different formats) by the web user. Each webpage has a specific HTML (Hyper Text Markup Language). A static webpage is pre-designed page that gets loaded whereas a dynamic webpage is generated by a web based application on the server side software or client side scripting. Overall web page load time can be divided as request, access and receive times and is the total time to download all the contents.

The Secure HTTP (HTTPS) protocol uses SSL/TLS to establish a secure web communication between a client and a server. Both the TLS and SSL protocols use 'asymmetric' Public Key Infrastructure (PKI) system. X.509 certificate is an ITU-T standard for PKI and are installed on servers for authentication purpose. Every client (webbrowser and operating system) has a list of trusted root certificate authorities. The public key and certificate of these authorities are stored on disk. If a client visits a website, say Google, over https, the server will send to the client its certificate, signed by a root certificate authority. This certificate contains the public key needed to begin the secure session. The client will then verify that the certificate is authentic, that is, the certificate belongs to Google and that it is signed by a certificate authority it (client) trusts. An example of google.com's certificate is given in Figure 1.



Issued To	
Common Name (CN)	www.google.com
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	6A:F4:E4:88:96:4E:EF:DB
Issued By	
Common Name (CN)	Google Internet Authority G2
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>
Period of Validity	
Begins On	04/22/2015
Expires On	07/20/2015
Fingerprints	
SHA-256 Fingerprint	94:AC:50:21:F3:E5:A7:B5:4B:D1:71:71:B9:D4:59:D3: C2:28:E3:04:DD:82:81:FB:73:09:6F:E0:1B:6F:C0:55
SHA1 Fingerprint	D2:66:8D:79:F5:38:D9:E1:9E:24:6B:7F:81:AD:9E:DE:2A:F3:8B:9F

Figure 1: An example of a X.509 certificate issued to Google.

2.2 Related work

A significant amount of research has been performed in studying the extra overhead that is incurred with use of HTTPS over HTTP. The increased concern over security and privacy has amplified the adoption of secure connections for over 50% of all web connections.

Study by Goldberg et al. [11] in 1998 showed increase in encryption time by 22% on popular webservers with secure connections. A study by Naylor et al [15] to measure the cost of using HTTPS by loading webpages of Alexa top 500 websites 20 times. It showed overhead to be more than 500ms for 40% (over 3G network) to 90% (over fiber connection). A study by Coarfa et al [6] on performance cost of TLS summarized overhead factor of 3.4 to 9 with usage of HTTPS over an insecure connection and public key cryptography contributed 13% to 58% of this overhead. Other studies on the HTTPS overhead analysis [2, 14] have showed varying percentage of overhead. From all the studies it is observed that overhead is not consistent and varies with respect to the scans conducted. And most of the studies were performed from a single geographical location. Our study focuses towards analyzing https overhead for Alexa websites globally using PlanetLab nodes located in various geographical locations. We observed average overhead of only 16.1% using https over http non- persistent connections and overhead of 15% for persistent connections.

Butkiewicz et al. [19] characterized metrics such as images, scripts, server counts etc. in a web page affecting page download time. Ihm and Pai et al. [20] captured data set related to higher level characteristics of webpages and proposed a webpage analysis algorithm to group requests into streams and exploit the structure and size of the pages. We also analyzed the relation between page size and the download time.

Analysis of web certificates and certification authorities (CAs) have been performed previously. A study by Durumeric et al [7] analyzed HTTPS certificate ecosystem performing 110 Internet-wide scans over 14 months and classified more than 1,800 CAs vouching for the identity of websites. Holz et al. [13] performed on regular scans of high ranked websites (top 512 – 1024) of Alexa Top websites list that showed only 40% of certificates are absolutely valid. Another study by Arnbak et al [3] was concentrated on understanding the market and value chain for HTTPS using data set from [13]. Our experiment scans the 500,000 most popular websites of Alexa [1] to obtain root CAs and measure percentage of websites accessible when trusting fewer root CAs. We also performed this experiment in various geographical locations using PlanetLab nodes to determine if the Certificate Authority used varies with geographical location. It was noted that root CAs of websites did not change with geographical location.

CHAPTER III

IMPLEMENTATION DETAILS

As our website dataset, we used the most popular websites from Alexa [1], downloaded on October 20, 2014. We downloaded the X.509 certificates from the top 500,000 websites from Alexa [1]. To determine the download times for http and https connections, we used the top 100 websites from Alexa [1]. To determine if geographical location affects download time and certificate authorities, we leveraged the PlanetLab [2] network to download the webpages of the top 100 websites and the certificate list of the top 500 websites. Table 1 shows the 11 PlanetLab nodes used in our experiments.

PlanetLab Node	HTTPS Overhead
planetlab-01.vt.nodes.planet-lab.org (PL1)	21.74%
planetlab2.csee.usf.edu (PL2)	8.55%
pl-node-1.csl.sri.com (PL3)	7.29%
planetlab1.citadel.edu (PL4)	4.10%
planetlab1.csuohio.edu (PL5)	5.86%
plonk.cs.uwaterloo.ca (PL6)	15.84%
planetlab1.koganei.itrc.net (PL7)	19.10%
pl1.eng.monash.edu.au (PL8)	12.73%
planetlab0.otemachi.wide.ad.jp (PL9)	19.21%
planetlab1.cesnet.cz (PL10)	2.79%
planetlab1.cs.otago.ac.nz (PL11)	23.30%

Table 1: List of PlanetLab nodes used and their respective average overhead percentage in download times for https connections.

3.1 HTTPS Overhead

We used PhantomJS [16] to download the whole webpage (html file and related files such as images on the html page) Download time is calculated as the difference in the time from the http(s) request to a webpage to the time when all data of that webpage has been downloaded. For each website, we downloaded the webpage 100 times over an http non-persistent connection, then downloaded that webpage 100 times over an https non-persistent connection. For each connection, we ensured that the previous connection is completed and the cache is cleared, so that each connection acts as a completely new connection (non-persistent). If there is any redirection, our script follows the redirection; for example, http://facebook.com redirects to https://facebook.com. The experiments to download the whole webpage were started on March 20, 2015 and finished on March 21, 2015.

HTTP/1.1 specifies all connections to be persistent. Persistent connections allow multiple requests to use a single connection, eliminating need to perform SSL Handshake multiple times and thus reduce the communication time. Hence, we compared the https overhead for persistent and non-persistent connections by running the experiments on May 20, 2015. Algorithm 3.1. outlines the pseudocode for our experiments.

Algorithm 3.1: Determine overhead time of https connection over http connection

Input: List of top 100 websites, $L_{websites}$

Output: Files containing download times of each website for both https and http connection **OR** File containing overhead of https over http for all websites as part of input

Procedure:

```
for every website $w \in L_{websites}$ do
Prepare http URL (like http://google.com) or https URL (like https://google.com)
```

```
http(s) : for n = 0 to 100 do
    Connect to the URL
    Record download start time, $T_{start}$
    if server response code is 200
        download whole webpage
        record download end time, $T_{End}$ and save $W_{http(s)}[n] = T_{diff(s)} = T_{End} - T_{Start}$
    else
        add website to unresponsive website list
    end if
    increment n
    end for
    calculate average http and https download time, $T_{http} = W_{http} /100$ & $T_{https} = W_{https} / 100$
    record average download time, $T[w] = T_{https} - T_{http}$ (in unit: milliseconds)
end for
```

3.2 Certificate chain

For https connections, certificates are trusted by browsers to guarantee the identity of the webserver/website. We implemented custom java code to obtain the list of certificates provided by each website. From this list of certificates, we can then determine the number of root certificates that need to be trusted to access the Internet. For each website from the top 500,000 Alexa sites, we connected to the webserver to download the certificate list. Since some websites do not support https connections, we ignored these sites. Once the certificate chain is obtained, we extracted the root certificate. Our algorithm is shown in Algorithm 3.2.

Algorithm 3.2: Obtain CA list

Input: List of top MMM websites, $M_{websites}$

Output: File list of root certificate authority for each website

Procedure:

```
for every website $w \in \mathbf{M}_{websites}$ do
    Connect to https://w using HttpsURLConnection object
    if server response code == 200
        Download certificate chain
        Extract root certificate and certificate authority details
    end if
end for
```

Our experiment was run starting from February to April 2015 on a machine connected to a university network to download the root certificate authority for the top 500,000 Alexa websites. To check if a website over a secure connection is authorized by the same CA when accessed from different geographical locations, we ran a similar experiment on PlanetLab nodes to obtain the CA list for the top 500 Alexa websites. A smaller set of websites is used because if the certificate chain is the same for these websites, then it is likely the same for other websites.

CHAPTER IV

FINDINGS

4.1 HTTPS Overhead

4.1.1 Non-persistent connection

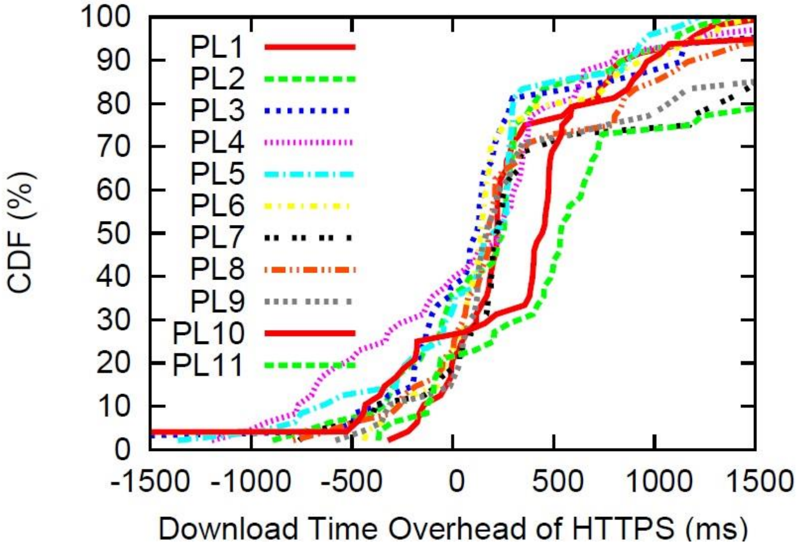


Figure 2. CDF of the communications overhead in milliseconds for all 11 PlanetLab nodes.

Figure 2 shows the communication overhead in milliseconds for 11 PlanetLab nodes used in first run. We measure the download time of whole contents of the webpage rather than just one round trip time. As seen on the graph, for few websites secured connection is faster than normal http connection. This is likely due to SPDY [10] implementation of https which attempt to decrease the overhead in the HTTP protocol and uses compression. However, for most of websites (above 30%)

using secure connections does increase the overall download time. The median download time overhead is about 200 milliseconds, which is not a lot as the average round trip time latency on the Internet is about 90 milliseconds [12].

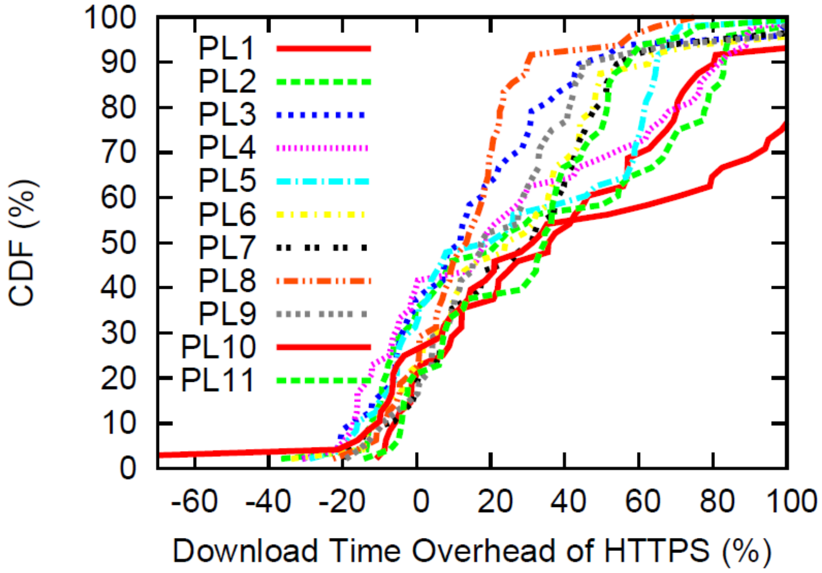


Figure 3. CDF of the communications overhead percentage for all 11 PlanetLab nodes.

Figure 3 shows the percentage download time overhead when using https. Again, it can be seen all PlanetLab nodes experience a similar overhead percentage with some nodes having a long "tail". The median overhead percentage is at around 20%.

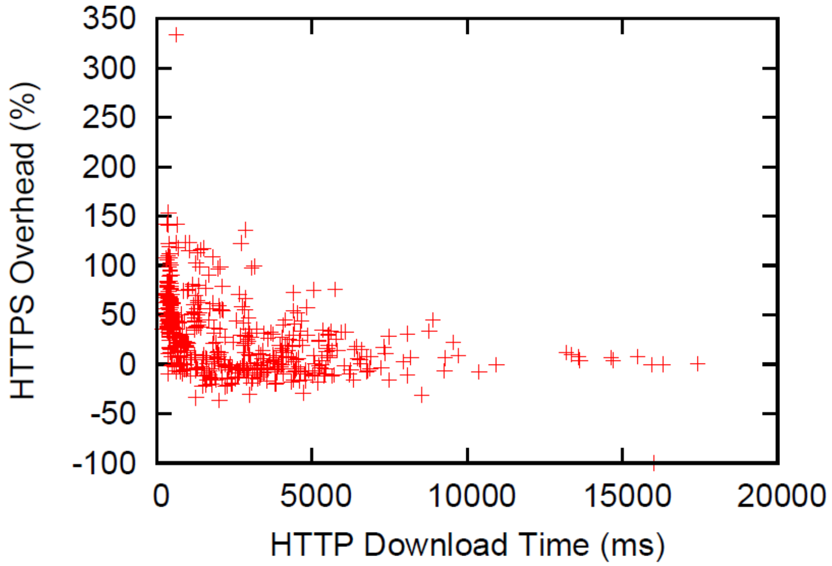


Figure 4. The HTTPS overhead percentage compared to the HTTP download time.

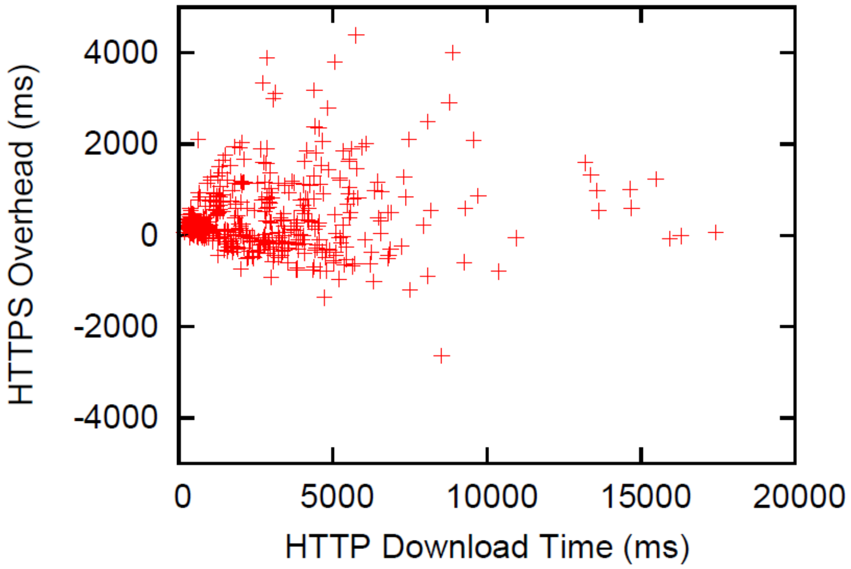


Figure 5: The HTTPS overhead in milliseconds compared to the HTTP download time.

To further understand the high overhead percentage, we plotted the http download time in millisecond against its respective https download time overhead percentage. As shown in Figure 4, the high overhead percentage occurs for really small (very fast) download times over http connections. However, as shown in Figure 5, the absolute value of the overhead for https connection is about the same regardless of the download time over http connections (with the exceptions of a few outliers).

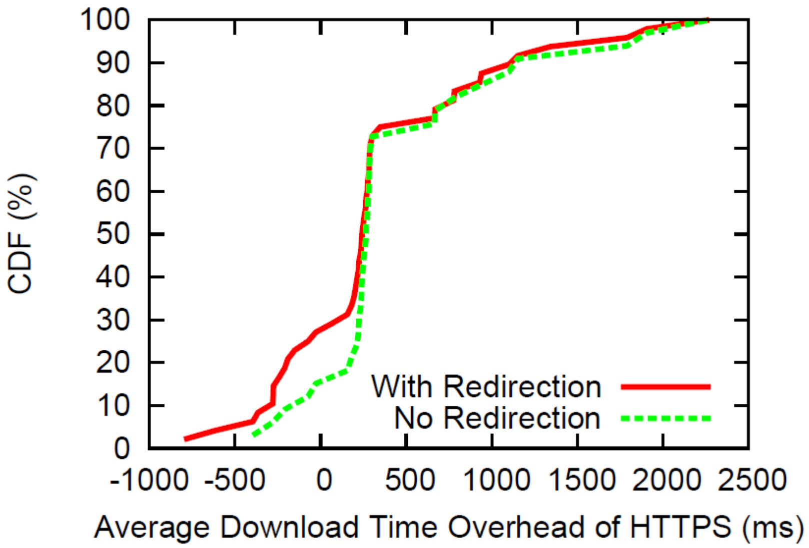


Figure 6: Average download time overhead of https versus http in milliseconds. The graph also shows the overhead if redirection from http to https and from https to http are ignored.

A few websites such as google.com and facebook.com perform redirection to their secure sites if a user visits their site over http. Figure 6 shows the average download time overhead when ignoring redirection. The Figure also shows the average download time overhead when not ignoring redirection. The Figure shows that redirection does not significantly impact the download time overhead when using a secure connection: both lines are almost exactly the same. The median overhead is about 200 milliseconds, the 75th percentile is about 350 milliseconds, and the 90th percentile is about 1.1 seconds. Figure 7 shows a similar graph with the overhead percentage instead of absolute values. For most of the experiments, the overhead when using https is not significantly higher than when using http.

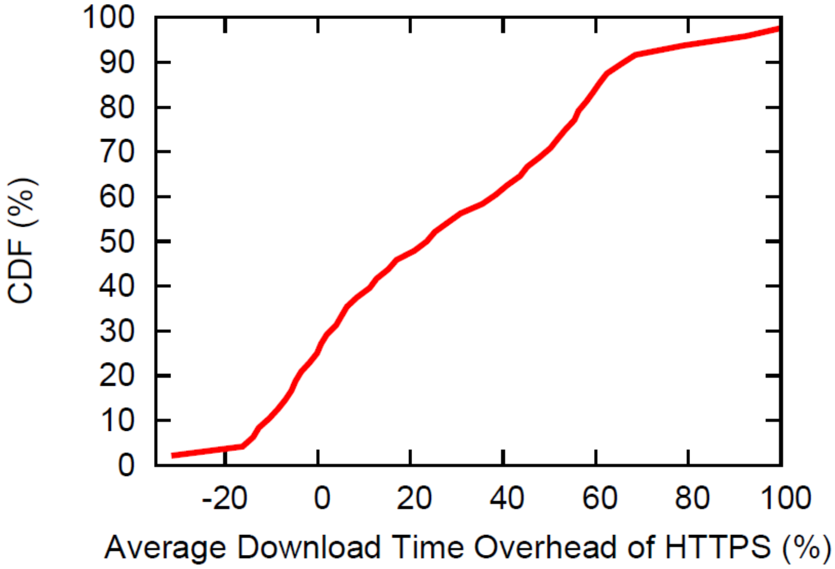


Figure 7: Average download time overhead percentage of https.

This result shows that the communications overhead in using https connections is not significant when compared to using http connections. A user can easily wait a few hundred milliseconds for the rest of a webpage to load. This overhead will decrease as more webservers start supporting HTTP/2.0, which implements SPDY.

4.1.2 Persistent vs Non-persistent connection

PlanetLab Node	HTTPS Overhead	
	Non-persistent	Persistent
planetlab-01.vt.nodes.planet-lab.org (PL1)	31.30%	29.67%
planetlab1.citadel.edu (PL4)	21.61%	4.36%
planetlab1.csuohio.edu (PL5)	26%	23.04%
plonk.cs.uwaterloo.ca (PL6)	14.40%	22.08%
planetlab1.koganei.itrc.net (PL7)	10.11%	-0.46%
pl1.eng.monash.edu.au (PL8)	21.70%	20.82%
planetlab0.otemachi.wide.ad.jp (PL9)	27.00%	20.00%
planetlab1.cesnet.cz (PL10)	2.79%	5.10%

Table 2: List of PlanetLab nodes used for second run and their respective average overhead percentage in download times for https persistent and non-persistent connections.

In this sub-section, we present results for set of data collected on May 20, 2015. This includes a) comparison of https communications overhead on persistent and non-persistent connections b) relation between whole webpage size to download time for non-persistent connections.

During this run we could connect only to 8 of the 11 PlanetLab nodes mentioned in Table 1.We could not connect to nodes - PL2, PL3 & PL11 (due to temporary shutdown of nodes).

Persistent connections, also called keep-alive connections, reduces the web page connection time and eventually reduce download time. In HTTP/1.0 it was an additional feature to be enabled. But, in HTTP/1.1 all connections are considered persistent unless they are specified to not be persistent. Persistent connection enables HTTP pipelining of requests and responses and reduces network congestion by reducing number of packets caused by TCP connections.

The advantages are more obvious with HTTPS or HTTP over SSL/TLS. There, persistent connections may reduce the number of costly SSL/TLS handshake to establish security associations, in addition to the initial TCP connection set up. As most websites uses HTTP/1.1 it is important to measure the webpages download time using persistent and non-persistent connections simultaneously to compare https communications overhead of both.

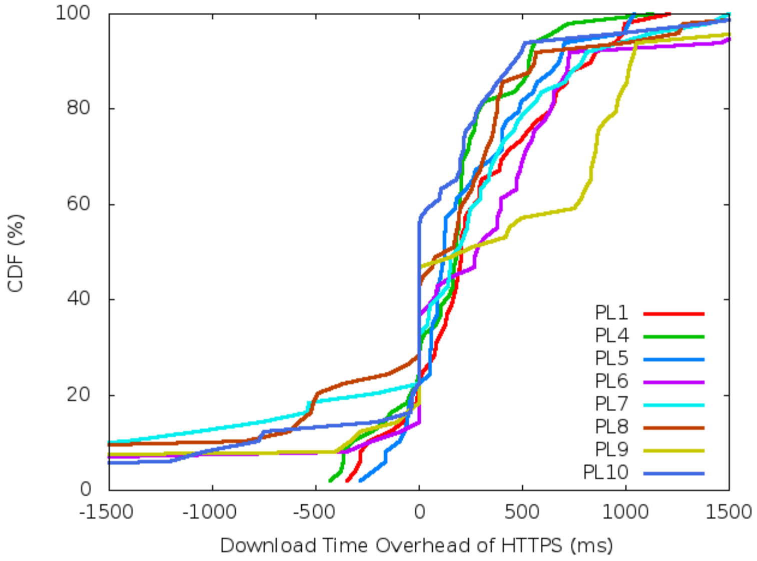


Figure 8. CDF of the communications overhead in milliseconds for non-persistent connection.

Figure 8 shows the communications overhead in milliseconds recorded for non-persistent connections. Compared to data in Figure 2, median download time is 180 ms i.e. 20 ms (10%) lesser than the median of the initial run.

Figure 9 shows overhead in percentage for non-persistent connections. Around 20% websites downloaded faster (less time) on secured connections. The median overhead percentage is at around 20% and same as data in Figure 3. Table 2 shows the overhead percentage for persistent and non-persistent connections.

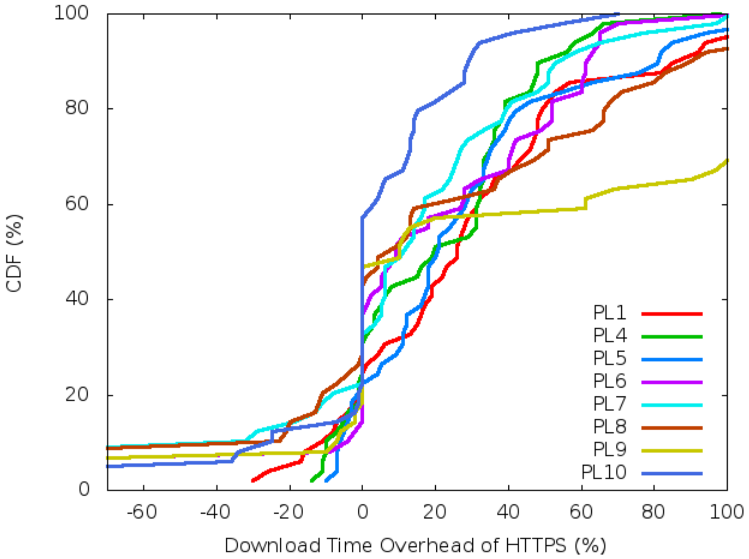


Figure 9. CDF of the communications overhead in percentage for non-persistent connection.

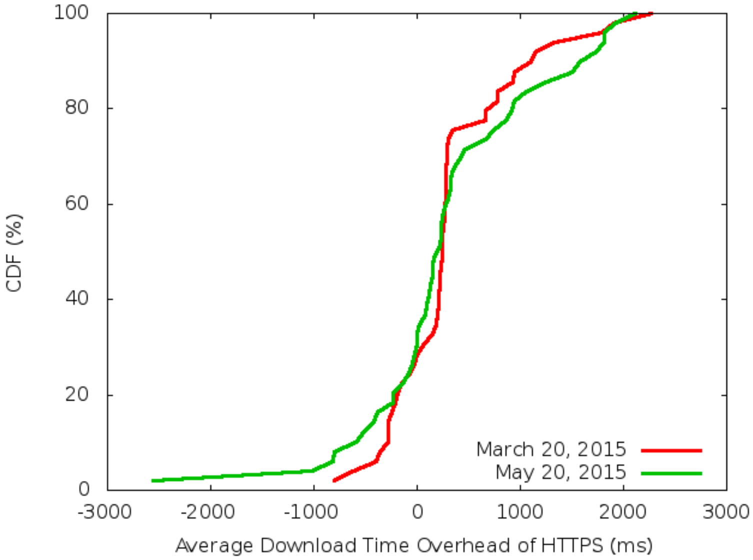


Figure 10. CDF of the average communications overhead in milliseconds for non-persistent connection runs.

Figure 10 shows average communications overhead in milliseconds both runs of non-persistent connections. In both the runs, around 20% of all websites downloads faster on https connections over http. The difference of average overheads for non-persistent connections is 13 ms, which is insignificant. This shows that new experiment run on May 20 is same as the previous experiment from March 20/21.

Figure 11 shows communications overhead in milliseconds for persistent connections. Downloading same webpage multiple times on persistent connection reduces the average download time. First download is usually slower compared to subsequent downloads of same webpage. The median download time recorded for persistent connections is 90 milliseconds. It is equal to average round trip time latency on the Internet. This is 50% less than median of non-persistent connections. Around 40% websites downloaded faster on persistent connections.

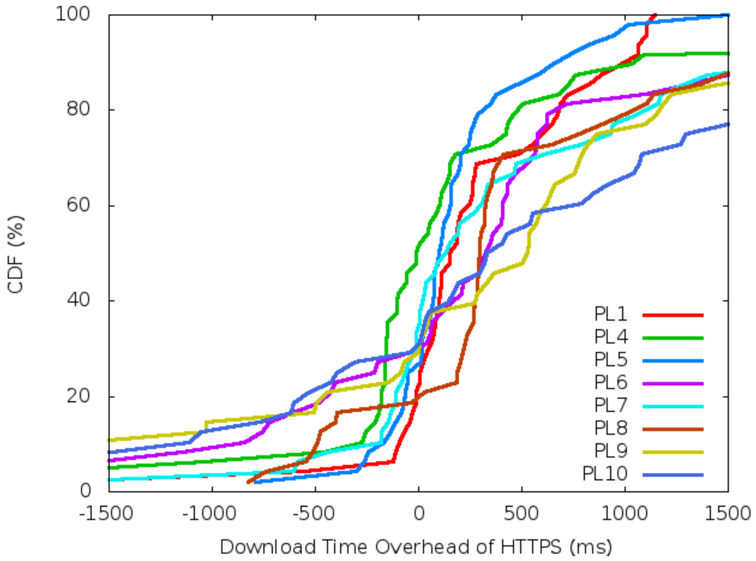


Figure 11. CDF of the communications overhead in milliseconds for persistent connection.

Figure 12 shows communications overhead percentage obtained on persistent connections. The median overhead percentage is around 15%, less by 5 % of non-persistent connections. Due to use of persistent connection to same webpage, subsequent connection time reduces but data transfer time remains same. So, the reduction of overhead percentage when persistent connections are used compared to non-persistent connections is substantial and as expected.

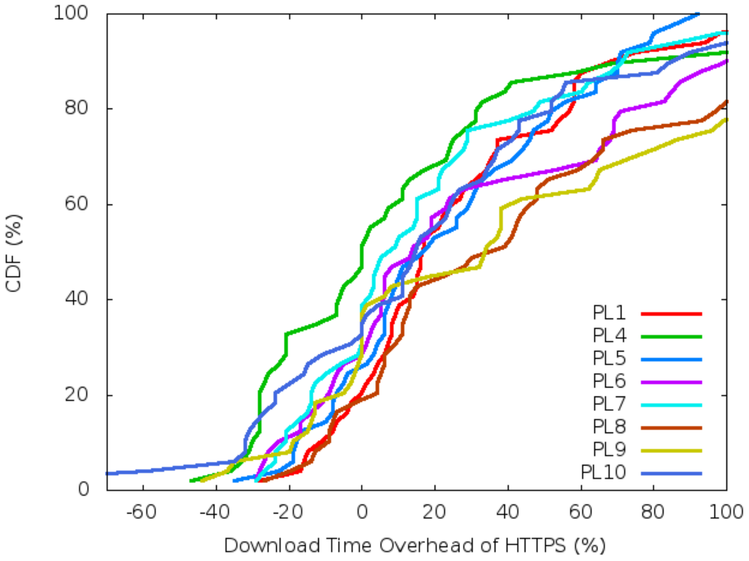


Figure 12. CDF of the communications overhead in percentage for persistent connection.

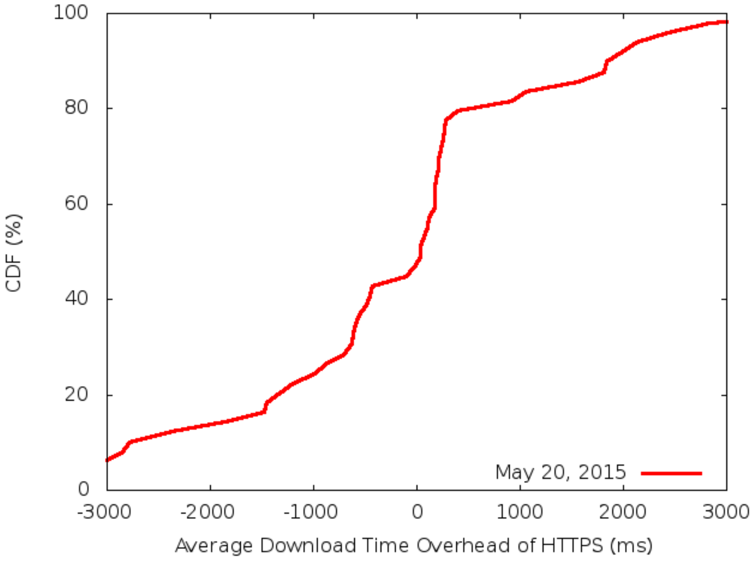


Figure 13. CDF of the average communications overhead in milliseconds for persistent connection.

Figure 13 shows average communications overhead in milliseconds for persistent connections. Around 40% of websites, download faster on https on persistent connection compared to 20% of websites that download faster on non-persistent connections. As more websites enable all its connections to be persistent as per HTTP/1.1 specification, the percentage of websites that download faster on https will increase.

Figure 14 shows average communications overhead percentage for 8 PlanetLab nodes obtained on both non-persistent and persistent connections. As mentioned earlier, 20% and 40% websites downloaded in less time on non-persistent and persistent connections respectively. With this data, it can been judged that use of persistent connections is advantageous in reducing whole page download time.

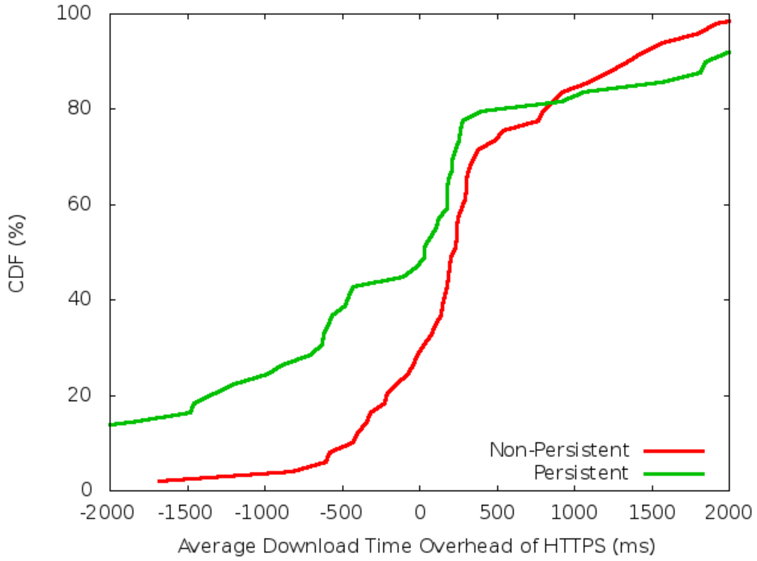


Figure 14. CDF of the average communications overhead in milliseconds for 8 PlanetLab nodes.

This result shows that using persistent connection webpages can be loaded faster than on non-persistent connection. Most of the web browsers open multiple concurrent persistent connections for same webserver to retrieve embedded objects and thus reduce the download time of whole webpage compared to use of non-persistent connection which opens new connection for each object. Figure 15 shows average overhead in milliseconds for all the experiments performed.

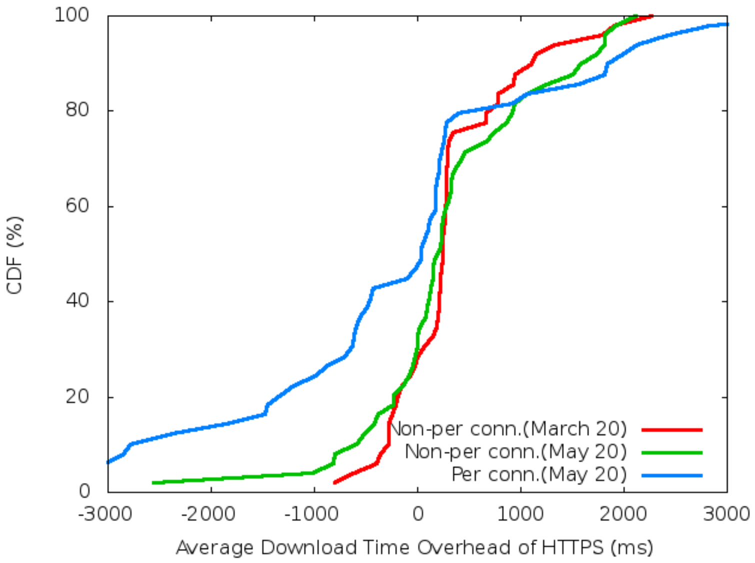


Figure 15. CDF of the average communications overhead in milliseconds for all experiments.

4.1.3 Download time vs page size

Page download time depends on many factors. Size of the webpage is one of the element. It varies with the content type on the webpage. Figure 16 shows the relation between download times and page size for non-persistent connections. Whole webpage size is almost same for download on normal and secure connection. With https trend line being 20% higher than of http trend line, it reiterates the fact of communication overhead of https being 20%. Thus, whole webpage size has less significance on https overhead measured.

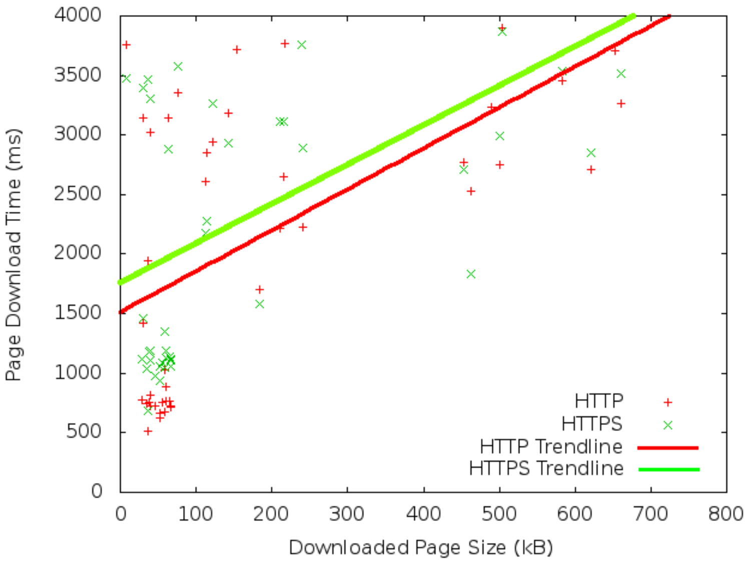


Figure 16. Page download time compared to page size

4.2 Certificate Chain

Table 3 shows the 9 most used root certificate authorities when querying the top 500,000 websites from Alexa. Not all the websites support a secure https connection. Out of these 500,000 websites, only 97,415 supported https (webserver listening on port 443 and returning a valid certificate chain). Out of these 97,415 websites, 18,944 (almost 20%) had their certificates signed by the AddTrust root certificate authority. The goal is to decrease the number of trusted root certificate authorities.

Figure 17 shows the percentage of the 97,415 websites that can be accessed over a secure https connection when trusting only a subset of the root certificate authorities. When only one root CA is trusted, 20% of websites can be accessed. When five root CAs are trusted, 59% of websites can be accessed. When ten root CAs are trusted, almost 80% of the websites can be accessed. By “access”, we mean that the webbrowser will not show any warning about the certificate.

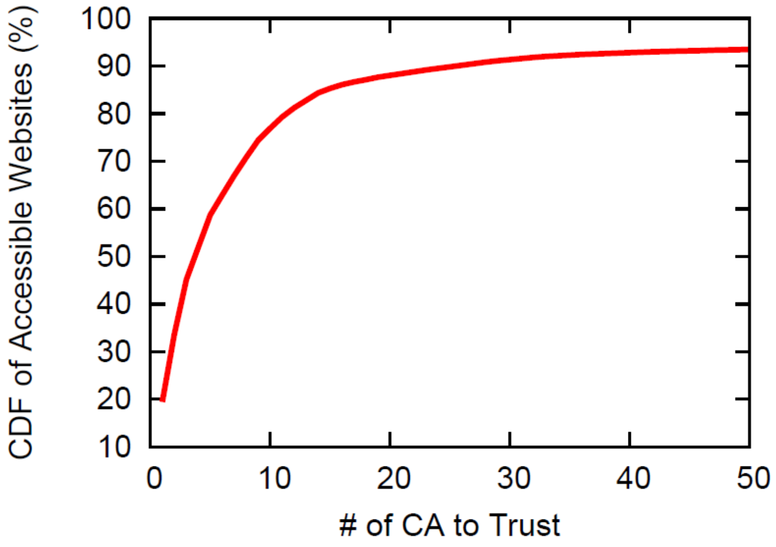


Figure 17. Percentage of websites that are accessible over secure connections when "trusting" varying number of root certificate authorities (sorted by most popular certificate authority).

AddTrust	18944
GlobalSign	13788
Equifax	11285
Go Daddy	6621
GeoTrust	6557
VeriSign	4083
Parallels	4048
DigiCert	3697
Thawte	3470

Table 3: Top 9 root certificate authorities for the most popular 500,000 Alexa websites, obtained from our university.

Equifax	83
DigiCert	42
VeriSign	38
GeoTrust	27
GlobalSign	27
Go Daddy	24
AddTrust	17
Thawte	12
GTE Corporation	10

Table 4: Top 9 root certificate authorities for the most popular 500 Alexa websites, obtained from 11 PlanetLab nodes. Each node fetched exactly the same certificate chains.

To determine if geographic location plays a role in the certificate list obtained, we ran the same experiment on the 11 PlanetLab nodes mentioned in Table 1 but downloaded the certificate list for the top 500 websites from Alexa (only 313 supported https). Table 4 shows the 9 most used root certificate authorities. We note that each PlanetLab node obtained the same certificate lists. The table shows that Equifax is the most used root certificate authority. We also note that Table 3 and Table 4 differ slightly in their ranking, but almost the same root certificate authorities can be found in both tables.

AddTrust AB	14753
GeoTrust Inc.	8299
GlobalSign nv-sa	7568
The Go Daddy Group	7377
Equifax	7362
VeriSign	6953
DigiCert Inc	3733
Thawte Consulting cc	3478
GoDaddy.com	2157

Table 5: Top 9 root certificate authorities for the most popular 500,000 Alexa websites, removing websites that belong to same company

Certain companies like Google and Amazon have multiple websites based on countries, for example google.com and google.co.uk. Certificates for these websites that belong to the same company are issued by the same certificate authority. Of popular 500,000 Alexa websites with only 97,145 supporting secure connections, 16,829 websites belong to one or other company already considered.

Table 3 shows the top 9 certificate issuing authorities for 500,000 websites. Table 5 shows the top 9 certificate issuing authorities for 80,316 websites (removing 16,829 websites out of 97,145 that belong to same company). Table 3 and Table 6 differ in ranking of certificate authorities. Foe example, all websites of Google Inc. is authorized by Equifax CA. Considering google.com and neglecting other google websites significantly changed the ranking of Equifax.

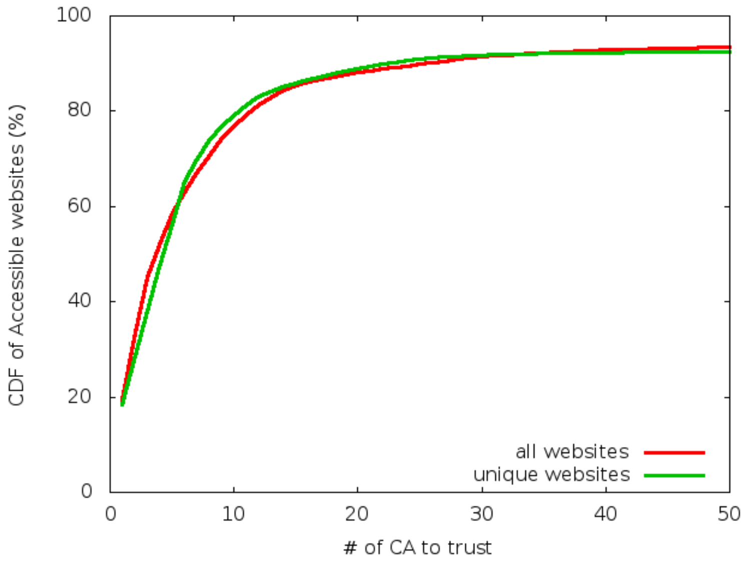


Figure 18. Percentage of unique domain websites that are accessible over secure connections (sorted by most popular certificate authority)

Figure 18 depicts a) percentage of 1345 websites (all websites) and b) percentage of 80316 websites (unique websites) that can be accessed over secure connection trusting 50 root certificates. With respect to unique websites, trusting 1 CA nearly 18% of websites can be accessed. Trusting 15 CAs almost 85% of websites can be accessed. The median is around 67%. The percentage of all websites trusting same root certificates is almost same with median of 68%. This shows that even after removing duplicate websites (belonging to same company) does not affect the popularity of certificate authorities. And even in this scenario, by trusting only 10 CAs we can still access 80% of the websites.

4.3 Summary

From the above experiments for page download time, we observed, on an average of 16.1% communications overhead with use of https over http non-persistent connections. For initial run, the average was 12.78% and for later 19.4%. The change in the average may be accounted to loss of few PlanetLab nodes and varying internet speed. With persistent connections, we observed https communications overhead of 15%. Compared to 16.1% overhead of non-persistent connections, persistent connections incur a lower overhead.

Web page (all contents – HTML, CSS, scripts, images. etc) size highly affect page download time. From our experiments related to download time vs page size, 500KB webpage is downloaded in 3 seconds. We noticed, an average 15 KB difference between page sizes of https and http, this negligible difference cannot be accounted for increase in https communications overhead.

Root certificates are a critical part of how encrypted connections validate the website connecting to. We learnt that the number of root certificate authorities trusted in webbrowsers and operating systems can be reduced to mitigate the effect of breaches at root certificate authorities or the possibility of rogue root certificate authorities. A more flexible approach to managing trusted certificate authorities also needs to be explored, where users can easily choose to delete or add new trusted certificate authorities.

CHAPTER V

CONCLUSION AND FUTURE WORK

The results of our experiments achieved our two goals: 1) https overhead is not significant and https should be enabled by default, and 2) the number of trusted certificate authorities can be decreased and a client can still access the majority of websites on the Internet without any certificate warning.

We now provide a discussion and avenues for future work. Supporting https is a server choice, but using https instead of an http connection is a client choice. For example, all http requests can be rewritten as https requests if https is supported. However, there is no reason for a server or company not to enable https. The processing overhead is minimal and we have shown that the communications overhead is also not significant; we expect the overhead to decrease as HTTP/2.0 and SPDY become more widely adopted. With respect to the cost of buying SSL certificates, free certificates are currently available [18]. The Let's Encrypt program [9], sponsored by major companies like Mozilla, Akamai, Cisco, and Electronic Frontier Foundation, plans to create a new certificate authority which will provide free certificates to users.

The list of thousands of trusted certificate authorities include companies from all over the world. Local companies tend to use local certificate authorities to issue certificates to them, that is, other companies in the same country. Also, some certificate authorities delegate the certificate issuing process to many other certificate authorities.

As future work, we plan to explore ways to create decentralized certificate authorities similar to PGP's web of trust and to propose a system that gives the user control over which certificate authorities to trust.

We hope that this work will start a conversation on enabling https by default and the trust assigned blindly to many certificate authorities, and spur further research in this area.

REFERENCES

[1] Alexa. http://www.alexa.com/, Accessed 2015.

[2] G. Apostolopoulos, V. Peris, and D. Saha. Transport layer security: how much does it really cost? In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 717-725 vol.2, Mar 1999.

[3] A. Arnbak, H. Asghari, M. Van Eeten, and N. Van Eijk. Security collapse in the https market. *Queue*, 12(8):30:30{30:43, Aug. 2014.

[4] China CA issues certificates for Gmail and Google. http://arstechnica.com/security/2015/04/01/google-chrome-will-banish-chinese-certificate-authority-for-breach-of-trust/, Accessed March 2015.

[5] CloudFlare. https://www.cloudflare.com/, Accessed 2015.

[6] C. Coarfa, P. Druschel, and D. S. Wallach. Performance analysis of tls web servers. *ACM Trans. Comput. Syst.*, 24(1):39{69, Feb. 2006.

[7] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman. Analysis of the https certificate ecosystem. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 291{304, New York, NY, USA, 2013. ACM.

[8] EFF HTTPS Everywhere. https://www.eff.org/HTTPS-EVERYWHERE, Accessed 2015.

[9] L. Encrypt. https://letsencrypt.org/, Accessed 2015.

[10] S. A. experimental protocol for a faster web. https://www.chromium.org/spdy/spdy-whitepaper, Accessed 2015.

[11] A. Goldberg, R. Buff, and A. Schmitt. A comparison of http and https performance.

Courant Institute of Mathematical Science, NYU, 1998.

[12] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: estimating latency between arbitrary internet end hosts. In *IMW: ACM SIGCOMM Workshop on Internet measurement*, 2002.

[13] R. Holz, L. Braun, N. Kammenhuber, and G. Carle. The ssl landscape: A thorough analysis of the x.509 pki using active and passive measurements. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 427{444, New York, NY, USA, 2011. ACM.

[14] H. Metwalley, S. Traverso, M. Mellia, S. Miskovic, and M. Baldi. Crowdsurf: Empowering informed choices in the web. *CoRR*, abs/1502.07106, 2015.

[15] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafo, K. Papagiannaki, and P. Steenkiste. The cost of the "s" in https. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, pages 133{140, New York, NY, USA, 2014. ACM.

[16] PhantomJS. http://phantomjs.org/, Accessed 2015.

[17] PlanetLab. http://planet-lab.org, Accessed 2015.

[18] StartSSL. https://www.startssl.com/, Accessed 2015.

[19] M. Butkiewicz, H. V. Madhyastha, and V. Sekar. Understanding website complexity: measurements, metrics, and implications. *In Proc. of the SIGCOMM conference on Internet Measurement Conference (IMC), 2011.*

[20] S. Ihm and V. S. Pai. Towards Understanding Modern Web Traffic. *In IMC* 2011.

VITA

Rakesh Ravishankar

Candidate for the Degree of

Master of Science

Thesis: ANALYSIS OF HTTPS OVERHEAD AND MINIMAL WEB CERTIFICATE CHAIN OF TRUST

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in July, 2015.

Completed the requirements for the Bachelor of Engineering in Instrumentation Technology at Visvesvaraya Technological University, Belguam, India in 2010.