Rasch analysis of HTTPS reachability

George Michaelson
APNIC*
ggm@apnic.net

Matthew Roughan Jonathan Tuke
UoA‡ and ACEMS†
{matthew.roughan,simon.tuke}@adelaide.edu.au

Matt P. Wand
UTS§ and ACEMS†
matt.wand@uts.edu.au

Randy Bush
IIJ¶
randy@psg.com

***Abstract*—The use of HTTPS as the only means to connect to web servers is increasing. It is being pushed from both sides: from the bottom up by client distributions and plugins, and from the top down by organisations such as Google. However, there are potential technical hurdles that might lock some clients out of the modern web. This paper seeks to measure and precisely quantify those hurdles in the wild. More than three million measurements provide statistically significant evidence of degradation. We show this through statistical techniques, in particular Rasch analysis, which also shows that various factors influence the problem ranging from the client's browser, to their locale.**

I. Introduction

There is a growing push for "HTTPS Everywhere," where HTTPS, or more exactly HTTP over TLS (Transport Layer Security), is a more secure form of the standard Hyper-Text Transfer Protocol. It is more secure in that it provides:

1. server authentication using certificates, *i.e.,* a server can prove its identity;
2. a private communications channel, *i.e.,* it prevents eaves-dropping; and
3. data integrity, *i.e.,* it prevents standard man-in-the-middle attacks.

HTTPS Everywhere is the ubiquitous use of HTTPS in preference to HTTP for all services, not only those specifically requiring a secure connection.

The Electronic Frontier Foundation (EFF) is promulgating a browser extension to this effect [1] as a defence against spying, *e.g.,* from nation states in the post-Snowden era. Google supports the idea [2], and has announced that they will give search-rank priority to HTTPS sites [3]. And the increase in the number of clients accessing the Internet through wireless connections mandates encryption at the connection level. Reactions include the HTTPS-Only Standard [4], for the US Federal Government.

There is a performance cost documented [5]–[7] as far back as the 1990s. This cost arises primarily because the certificate exchange requires an additional round trip at the start of a connection. However, most HTTP requests don't require a full handshake, and with modern hardware the cryptography overhead is not critical. For example Doug Beaver from Facebook, stated *"We have found that modern software-based TLS implementations running on commodity CPUs are fast enough to handle heavy HTTPS traffic load without needing to resort to dedicated cryptographic hardware. We serve all of our [Facebook's] HTTPS traffic using software running on commodity hardware."* [8].

So on the face of it, HTTPS Everywhere is a "no brainer." There is even an "HTTP Shaming" web page.

HTTPS Everywhere seems to be happening. StatOperator [9] reported that the number of (the top million) sites using HTTPS as the default increased from around 103 to 219 thousand from 2016 to 2017. Google reports client usage statistics [10], [11], and they show similar steady growth from 2015 to the present.

However, there is an important question to answer before we convert the entire Internet to HTTPS: *Will there be people who are stranded behind port 80?*

We know that HTTPS is not an issue for many people (the current large-scale deployments of HTTPS prove that it mostly works), but there could be locations, or users of specific equipment that face challenges. Detailed reasons are given in Section II. They range from concern about the quality of the technology, to the rejection of compromised connections.

In this paper we provide evidence to inform the technical and policy debate concerning the deployment of secure web services, by measuring whether users can access HTTPS in the wild. We collected 3.3 million observations using APNIC's web advertising infrastructure [12], from which we found that there is sufficient evidence to show that HTTPS is *not* easily accessible to all Internet users.

A secondary concern of this paper is the statistical rigour necessary to allow such a statement to be made with confidence. The proportion of users that failed to make an HTTPS connection in our study was small. It has been common in the past to simply report numbers, but our goal is to provide statistically confident statements, despite a noisy and faint signal. The ability to detect such faint signals is important — a mere 0.1% of users now represents millions of individuals. We do so using both standard statistical tests, and a tool that has not been previously used in Internet measurement, but which may find many other applications: Rasch analysis [13], [14].

We found statistically significant evidence that there are clients that find HTTPS connections harder to complete than HTTP, and that this difficulty was influenced by origin autonomous system, browser, country of origin, and operating system, suggesting a range of causes.

*APNIC, South Brisbane, 4101, QLD, Australia.
†ARC Centre of Excellence for Math. & Stat. Frontiers.
‡University of Adelaide, Adelaide, 5005, SA, Australia
§University of Technology Sydney, Ultimo, 2007, NSW, Australia.
¶Internet Initiative Japan (IIJ) Research Lab, Tokyo, Japan.

ISBN 978-3-903176-08-9

II. Background and Related Work

A. Experimental Context

Simple web services with no protection against snooping or identity are typically conducted over TCP port 80, using the HTTP protocol. We call this 'port 80' service or HTTP.

Web services which are protected by Transport Layer Security (TLS) are usually conducted over TCP port 443, commonly called 'port 443' or HTTPS.

There have been many studies of HTTPS. However, they have focused on two main topics.

1. The certificate landscape, *e.g.,* see [15]–[17], in which the problems with certificate distribution have led to security holes, and consequent fixes[1].
2. Comparisons between HTTP and HTTPS performance, looking primarily at their latency difference, *e.g.,* see [5], [6], but also considering communications overhead and energy consumption [7].

As a consequence of using HTTPS, an additional handshake is needed to establish a connection. There can be no effective proxy-caching of the content, and filtering (*e.g.,* by firewalls) is hampered. HTTPS also uses cryptography which induces extra computational (and hence energy) costs, which may be trivial on a modern computer, but may be important on battery-operated devices, such as mobile phones.

A deeper consequence of the additional layer of complexity is the potential for failures. Surprisingly, studies of HTTPS appear to assume basic reachability, or more correctly, they appear to assume that HTTPS reachability, while perhaps not perfect, will be no worse than HTTP. However, it is not obvious that this will be so. A prominent browser maker asked if the Asia-Pacific Network Information Centre (APNIC) Labs ad-based measurement system [12] could see if a statistically significant number of users were unable to access TLS protected web resources.

So, what are the possible concerns? They range widely; the following is an incomplete list.

1. A browser or OS may be too old to perform TLS at the current specification. The web server used in this experiment did not offer older approaches, such as RC4 cryptography, so there is a chance that pre-TLS 1.x browsers will fail. However, the older standards are no longer considered secure, and it is our view that providing a false appearance of security is worse than providing none. It might be tempting to tell users to "catch up", but this is infeasible on mobile networks that sell captive locked phones left behind on "old cold" protocol variants.
2. Some modern browsers use intermediate systems to speed up or cache data. Opera, for instance, deployed a world-wide "anycast" cloud of intermediates to offer speed-up services, performing tasks such as JPG compression, to make the web faster. It is possible that this service notionally works with TLS, but that it works badly for flows it has in port 80 that move up into TLS because the state doesn't exist. Other well-meaning intermediary systems might break such up-lifts.
3. The additional overhead of the extra handshake makes the session more vulnerable to network problems, and hence less stable.
4. TLS protects against the threat of bad actor man-in-the-middle attacks. If an on-path attacker intercepts the session and attempts to hijack an aspect of the content, TLS should prevent the flawed connection. However, if such attacks are prevalent, they become DoS attacks on the HTTPS service.
5. A firewall along the path might block encrypted traffic as a matter of course. Though most firewalls allow port 80 traffic, they sometimes block all other ports. This might be considered misconfiguration, but misconfiguration is not uncommon [18].
6. Firewalls or other middle-boxes may perform their own hijacking of a connection through installed certificates on user machines.
7. Flaws in implementations or configuration [19], [20].

Our approach uses a cross-site reference within an advertisement in order to create a measurement. The underlying idea is not new. It has been used to measure DNSSEC and IPv6 deployment, among other features, *e.g.,* [21] (or for a more general review see [22]). However, our approach differs in several respects from [22]. The most important is that it performs a pair of measurements: a control based on HTTP, and an actual measurement of HTTPS, the focus. As far as we are aware, past studies have typically lacked a control, and therefore have been hard to interpret statistically.

However, APNIC's measurement infrastructure also differs from other approaches in that we use (paid) web-advertising to instantiate the tests (details below). Additionally, all fetches are to an APNIC-managed server, avoiding the major ethical controversies of past experiments (see Section III-D for more discussion of this issue).

B. Simple Statistical Background

Here we lay out the key statistical background. The material is somewhat tutorial, but as these techniques are not commonly applied in the Internet measurement context, we feel it is valuable to be precise about the methods and their rationales. We start by defining terminology:

Observation: the collected responses of a single client's connection attempts (see Section III-A for details).

Sample: the set of all observations.

Measurement: a particular feature of an observation, for instance, whether a successful HTTPS GET was completed. We also call these *response variables*, and denote them by random variables (RV) $Y^{(j)}$, where $j \in \{\text{HTTP}, \text{HTTPS}\}$ is the *treatment*, and the measurement

$$Y^{(j)} = \begin{cases} 1, & \text{if measurement } j \text{ succeeds,} \\ 0, & \text{otherwise.} \end{cases}$$

The sample is the collection of instances $\{y_i^{(j)}\}$ of this RV.

[1] TLS security is predicated on valid certificates, and there have been significant problems resulting from this weakness in the past. However, Certificate Transparency mitigates many of these issues [16].

Test: a statistical test applied to the data.
Categorical variable: one that takes a set of discrete values.
Predictor: a variable, also called a *covariate*, whose value *may* influence the outcome of the measurements.

We make a distinction here between a measurement and a test, the latter meaning a *hypothesis test* to discriminate between a *null-hypothesis* H_0 and its alternative H_1. The advantage of a hypothesis test is that it is consistent and repeatable with strict, precisely-defined assumptions and interpretation. Through their use we can avoid making common errors, such as over-interpreting limited evidence.

The test is conducted with respect to a significance level, α, chosen at the outset of the experiment. Here we use the common choice of $\alpha = 0.05$. This sets the Type I error probability (the chance we reject H_0 incorrectly). The Type II error probability (the probability we fail to reject H_0 when it is false) is determined by the power of the test on the particular data. Thus we cannot control for it, but can ensure it is small by providing enough observations.

We calculate a test statistic, determine from this a p-value, and then reject the null-hypothesis if the p-value falls below α. The common interpretation of the p-value is that it is the probability, given the null-hypothesis is true, of observing the at least the given test statistic. Hence, a small p-value can be taken as evidence that the null-hypothesis is invalid. However, we must be careful of this interpretation, because of the underlying statistical nature of the problem.

When the null-hypothesis is true, we would expect to see a uniform distribution of p-values over a set of repeated experiments, which includes some values $<\alpha$, leading to Type I errors. In order to avoid incorrect inferences in repeated experiments, we should try to control the *Family-wise error rate* (FWER) not the *Per-comparison error rates* (PCER). We shall do so here using the *Bonferroni correction* [23], in which α is divided by the number of tests in the family. We should note that this is rather conservative, and that there are other more complex procedures available [23], but we deliberately use a conservative FWER here.

Our experiment is a *matched pairs* experiment. That is, the pair of measurements is conducted on the same client, the question of interest being whether some users have more trouble with HTTPS than HTTP. This cannot be answered simply by comparing the proportions of successes for each measurement, because in a matched pair experiments the measurements are very likely correlated. Simply plotting the two probabilities, while useful in an explanatory sense, would not take these correlations into account.

However, the inclusion of our control experiment makes it possible to ask this question in the formal context of hypothesis testing using *McNemar's test* [24], with hypotheses:

- H_0 is that $p_1 = p_2$; and
- H_1 is that $p_1 \neq p_2$;

where p_j is the probability that the jth measurement of any particular observation is successfully completed. Rejecting the null implies significant evidence that the difficulty of the two measurements is different.

C. Rasch Modelling and Analysis

Hypothesis tests are an important starting point, but they only tell us "if" but not "how much?" This paper further proposes the use of Rasch analysis, an approach within the broader area of *Item Response Theory* (IRT). It is best illustrated by its application to the analysis of exams. An exam consists of a list of m questions, performed by n students. Each student answers each question either correctly, or not, forming the binary response variables $Y_i^{(j)}$, where i is the student (in our context an observation) and j is the question (a measurement).

Rasch modelling is one of the most popular strategies within IRT [13], [14]. It posits that there are *latent* variables, namely:

1. the ability or proficiency of student i, denoted α_i; and
2. the difficulty of question j, denoted β_j;

that determine the probability that student i answers question j correctly. The variables are latent in that we do not know them *a priori*.

In its simplest case, *i.e.,* a dichotomous response, we have response variables $Y_i^{(j)}$, which are Bernoulli random variables indicating a successful answer to a question, whose probabilities are modelled as

$$p_i^{(j)} = \mathbb{P}\left\{Y_i^{(j)} = 1\right\} = \frac{\exp\left(\alpha_i - \beta_j\right)}{1 + \exp\left(\alpha_i - \beta_j\right)}. \tag{1}$$

The *logistic* function above has an inverse called the *logit* function. Applying the logit to (1) gives

$$\text{logit}(p_i^{(j)}) = \log\left(p_i^{(j)}/(1 - p_i^{(j)})\right) = \alpha_i - \beta_j, \tag{2}$$

a linear relationship between the logit and the parameters.

Rasch modelling's enduring appeal within IRT [13], [14] arises because:

- it simplifies the relationships so that reasonable estimates can be made, even though we have only one instance of each student attempting each question;
- unlike the conventional statistical paradigm, where parameters are fit to data, and accepted or rejected based on the accuracy of the fit, in Rasch modelling the objective is to obtain "data" that fit the model, *i.e.,* the latent predictor variables; and
- the Rasch model embodies the *principle of invariant comparison*, in which (broadly speaking) the effect on the outcome of a question is separated into the affect of the respondent, and the question's difficulty.

The approach is not limited to modelling examinations, but can be applied to a set of observations such as we have. However, in traditional *dichotomous* Rasch analysis, each student answers each question one of two ways (correctly or incorrectly). Here, the naïve approach would be to consider each observation as a "student" with two questions (the HTTP and HTTPS measurements), but then we would have well above the number of students typically considered, blowing up the computational load for most algorithms. Moreover, this would be banal, as performance at this level of granularity is immaterial to us.

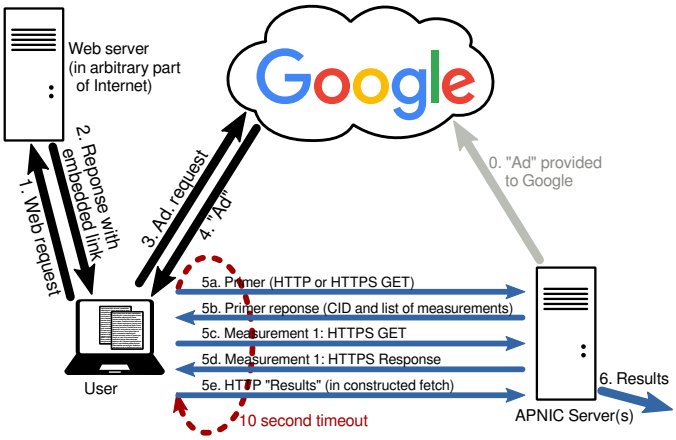

Fig. 1: *The observation process: numbers indicate sequence.*

In practice, we would like to group measurements into meaningful partitions, but we then depart from the standard dichotomous Rasch model.

There are at least two alternative approaches; we might think of these partition's subsets as either being comprised of:

1. a group of similar students, who have an underlying property in common (usually we assume members of the group have proficiencies that are random variables with a common mean and variance); or
2. a group of repeated measurements of a single "student" who corresponds to the particular subset, and the responses are now binomial random variables corresponding to the number of correct measurements within the subset.

The two assumptions lead naturally to different algorithms, and as the second is non-standard, we leave analysis its details until Section IV.

III. Experimental Method

A. Measurements

APNIC Labs uses web advertising to measure browser behaviour worldwide [12]. The advertisement is written in HTML5 and fetches multiple pixels in the various protocol exchanges under test (DNS, TCP/UDP, IP, TLS). The system is 100 lines of JavaScript, gzip compressed to 5kb of data, which is a small cost in web-page loading.

The process is illustrated in Figure 1. The observations start (at 5a) with a *primer* query initiated by the advertisement served to the user's browser via standard advertisement infrastructure. The primer query is an HTTP GET, and the body of the response is a set of measurements to be performed. Each measurement is a discrete URL with the unique client identity (CID) encoded in it, and is fetched under a ten-second timeout via an asynchronous JavaScript web fetch; on completion of a measurement, the time is recorded. On completion of all measurements, or the ten-second timer, a *result* web query is sent, which encodes the measurement results in the query argument as a sequence of labels, showing the time or 'null' if they did not complete inside the time limit.

The web logs show whether a primer/result pair was valid, and if so, we analyse the results. Observations without primer and result success are filtered from the sample. The goal in discarding these is to focus on the measurements with the highest signal to noise ratio — measurements without a valid pair indicate problems other than a failure of HTTPS, and hence don't add much information.

TABLE I: *Experiment duration, and number of observations. Analysis focuses on the 3.3 million experiments initiated with HTTP (with a subsequent HTTPS GET).*

Duration (days)	Unique client IDs (millions)	Valid responses (millions)	HTTPS init. (millions)	HTTP init. (millions)
25	192.5	132.4	129.1	3.3

The primary goal of these measurements was to collect information about ability to perform HTTPS. Google requires that advertisements placed over a TLS-secured session remain in TLS. Thus we could not recruit TLS users into a test of insecure web access. However, we were permitted to take an HTTP session and include fetches of web elements over TLS. Therefore, our observations measure HTTP users who were asked to fetch a web asset over TLS, thus detecting their ability to upgrade to TLS, which is not precisely a raw HTTPS access.

We focused on connections initiated over HTTP because this HTTP signal provides a "control." The priming process and the HTTP control measurement follow an identical connection path. Hence, if the observation is valid, the client has demonstrated the ability to perform an HTTP GET; therefore failures of subsequent HTTP GETs provide an indication of the "noise" in the system, *i.e.,* the baseline rate of random loss against which we should measure HTTPS connection failures.

The data were collected between the 10th of November and 4th of December, 2016. Table I shows the total set of client IDs, and the number of valid responses. A large number of connection attempts defaulted to initiating over HTTPS. Table I shows the decrease in the number of experiments as we progress through HTTPS to only HTTP initiations.

Initial exploratory analysis suggested that a signal existed, but at an intensity that could not be easily measured. The situation is analogous to experiments conducted on mice who are genetically modified to have cancer. We wished to measure factors that affected a situation with small probability, and so we inflate the probability of seeing the phenomena of interest. In our case, we focused on observations where the initial connection was HTTP, because these were the cases where failures of HTTPS were most often expected to occur.

As noted, it is a standard statistical approach to collect data in this way, but we must note that the observation is not representative of a "typical" Internet user. For instance, were we to measure a failure rate of 1% on these observations, this does not mean that the general population has a 1% failure rate. However, the question of interest here is not the absolute value of the failure rate, but whether HTTPS is "harder" than HTTP, and what factors affect the failure rate.

More formally, the main goal was to measure success/failure for sessions upgrading to TLS and to see if those sessions which could not upgrade to TLS were still successful on port

80. In other words: *"are there stranded users?"*

Google's infrastructure does not carry forward the referring site, and DHCP can reallocate IP addresses, so we cannot be certain that there were no repeats. However, the advertising infrastructure is intended to reach many discrete individuals, so the number of repeats should be very small. We also removed the small number of obvious duplicates from the data. There is some complexity in this process, resulting from apparent fetches from the same IP address that cannot be resolved due to the potential presence of middle-boxes such as Network Address Translators (NATs). We preserved entirely unique requests for the primer, but removed additional fetches without a new primer. As a result, we cannot claim that there are no duplicated observations, but they should be minimal.

B. Data Collected

The experiment logged all of the web fetches, using domain names directed to APNIC-managed DNS and web servers. We also captured the packet flow to relevant services: port 80, port 443, ICMP, DNS, as well as any fragmented IP state.

The combination of web logs, DNS logs, and packet captures allowed us to collate experiments by their IP address and identity in the DNS name, and as presented to the web. Thus we were able to derive the exact sequences of events in any observation.

In the case of this experiment, the data were processed into the form of a series of flags indicating (1) the success or failure of each stage, and (2) whether the measurement succeeded within a timeout. The delays were recorded in each case up to 120 seconds, but for our purposes we recorded success if the measurement completed within a timeout of 10 seconds.

In the data analysed, unique client IDs were assigned to anonymise the data. We used code which harvests system entropy and time, to obtain probably unique (modulo birthday paradox) non-sorted 96-bit numbers. We then mapped them into hex (see [25], for the code that was embedded in the NGINX [26] web server).

C. Classification of Covariates

The secondary goal here was to identify the qualities behind the quantities: *i.e.,* can we understand these users in terms of browser type, ISP, economy, or operating system, in order to identify specific problem causes? In practice, this is important because the goal behind APNIC's participation in such experiments goes beyond simply finding problems. Ideally, the experiment should also help develop strategies to remediate any problems found.

For the purposes of the analysis, a set of qualities was identified, which we felt were simple, easy to reproduce by other people, and provided useful groupings for understanding causes. These qualities were:

- country,
- region (based on United Nations sub-regions [27]),
- origin Autonomous System Number (ASN),
- browser, and
- Operating System (OS).

We used the daily BGP table collected at AS4608 to map IP addresses to origin ASN. There are well-known problems in such a mapping. However, those problems are most prevalent in infrastructure addresses, and we measured "eye-balls" here; inter-AS links do not browse [28].

Likewise, mapping of eye-balls to geographic locations is more accurate [28] than mapping arbitrary IP addresses to geographic locations. In this paper, we used MaxMind [29] data to geolocate the IP addresses, but only at the country/region levels, and so expected a reasonably low error rate.

We also logged each client's user-agent string, which provides details of the client's browser, OS, and device. To collect and parse the information we used the Python *uabrowser* library [30]. It is known that the user-agent string is spoofed in some cases, for instance the ToR browser bundle does so by default (*e.g.,* it pretends to be running on Windows, regardless of the underlying OS). However, there is no easy way to avoid this problem at present, and it remains a caveat on the browser- and OS-level results.

We also considered categorising the client's device-type, but this was too noisy to be useful at this stage, due to the large number of uniquely identified device types by vendor and version-string.

For each of these categories, we collated them into a series of unique values, and then used a one-way random relabelling to anonymise the categories. There might be enough data to perform some act of deanonymisation, to obtain values for some categories, However, it is important to note that this level of blinding was not intended for the protection of individual privacy (already protected through the client ID anonymisation, and unlikely to be compromised by the additional coarse-grained categorisations). Rather it was intended to allow the statistical analysis to proceed, unbiased by preconceived notions of the likely results.

D. Ethical Concerns

Google's advertising infrastructure was used here, and so we reviewed compliance with Google and APNIC lawyers, and complied with Google's legal restrictions on the measurements. In particular, these excluded use of Personally Identifying Information (PII), which in any case we did not require or want. End-user IP addresses were only used for ASN and regional classification, and were then anonymised via a one-way-mapping.

As in many Internet measurement experiments the nature of the measurement technique precluded voluntary recruitment. However, we strictly followed any suggestion that the users wished to opt out of such studies. For instance, end users who had enabled 'do not track', who had disabled JavaScript, or who ran ad-blocking software were not recruited. Also, as far as possible, no users were repeatedly asked to run the experiment, in order to place minimum load on any one user.

These measurements were also less contentious than others that have applied similar cross-origin requests *e.g.,* [22]. TLS is used ubiquitously for banking, login, end-user tracking by less responsible advertisers, "bread crumbs" and web-site

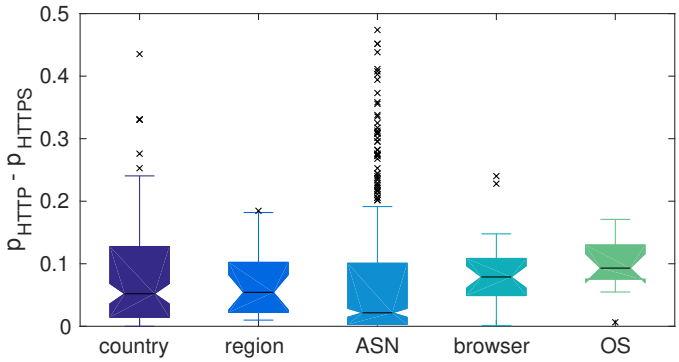

Fig. 2: *Box and whisker plot of difference $p_{HTTP} - p_{HTTPS}$ by covariate, showing the interquartile range (shaded region) the confidence interval for the estimate of the median (notch), Tukey's 1.5 IQR, (whiskers) and outliers (crosses). Note that the lower quartile is always positive, as are all whiskers except those for ASN, suggesting that HTTPS* is *harder than HTTP. Also, if covariates had no effect on the result, these should all have similar interquartile range and median, the differences therefore suggest some structure.*

logistics. Evidence suggests that the rate of TLS in the public web is high (above 50% [7]) and very likely significantly higher given the age of that study, and that it has been rapidly increasing in recent years [9], [10]. Therefore, the simple presence of a request to fetch a web asset over TLS does not represent a high-risk activity.

Moreover, the measurement site to which the advertisement redirected requests is innocuous, belonging to a regional address registry (APNIC), so we were not able to discern any reasonable risk to participants from such a connection.

In this experiment, those researchers not employed by APNIC were exposed only to anonymised data, except for those statistics reported here.

IV. Analysis

In this section, we discuss the results of the analyses. We will start with "broad brush" simple hypothesis tests, then focus on those same tests, applied to country, region, ASN, OS and browser. This will be followed by a more comprehensive Rasch model, which analyses the data as a whole.

Figure 2 shows a box and whisker plot [31], [32] of the differences organised by the various predictors. Note that the lower quartile is always positive, as are all whiskers except those for ASN, suggesting that HTTPS *is* harder than HTTP. Our task is to determine whether this effect is statistically significant.

Also, if covariates had no effect on the result, these should all have similar interquartile range and median, the differences therefore suggest some structure.

A. Standard Statistical Tests

We applied McNemar's test with a significance level of $\alpha = 0.05$, applying the appropriate Bonferroni corrections when conducting a set of multiple tests (*i.e.,* we used significance α/n for a family of n tests). Note that in some cases, *e.g.,* when we were testing against ASN, n was quite large, and so the actual threshold was very small. Less conservative corrections exist (for instance the Sidak or Holm-Bonferroni) but the results here are conclusive without needing the extra power gained through these more accurate corrections.

TABLE II: *Statistical tests applied to the whole dataset. Note that very small p-values are reported via a bound.*

Test	p-value	Accept/Reject
Fisher	$< 2 \times 10^{-16}$	Reject null
McNemar	$< 2 \times 10^{-16}$	Reject null

The results, shown in Table II, for the test applied to the whole dataset was a p-value less than 10^{-16} strongly supporting a difference in the two measurements. This finding must be qualified: although the two measurements are matched they occur in order, and hence, there may be some effect on the second measurement resulting from the state created by the first. So we must understand that this experiment concerns lifting a connection up from HTTP to HTTPS, not an arbitrary HTTPS connection (see the detailed notes in Section III).

It is important, also, to verify that this is not caused by some confounding effect of the covariates. If the covariates were truly irrelevant, we would expect that interquartiles and medians should be the same (within the ranges of natural variation shown by the confidence in intervals in the case of the median), and hence Figure 2 provides evidence that the covariates are important.

Therefore we now consider what part the covariates (country, region, ASN, OS and browser) play. We chose, at least initially, to be conservative by only analysing groupings with at least 500 observations. It is quite possible that smaller groupings would have been amenable to analysis, but we had no need (here) to describe the relationships between all of the rarer groupings, as our goal was to ascertain whether the overall result was supported on a finer level of granularity.

The number 500 was chosen through an initial exploratory analysis, which noted that some of the probabilities in question were quite close to 1, and hence statistical rules of thumb required a moderately large number of observations. As we did not know exactly what these probabilities were *a priori*, we chose a conservative lower bound. We also found, as Table III shows, that excluding the groups with a small number of observations excluded only a small percentage of the data.

The results can be seen in Figure 3, which shows histograms of the distributions of p-values over the set of tests grouped by the various categorical covariates described above. The important fact to note is that most of the p-values are small. We cannot see (at the resolution of the plot) whether the p-values fall below the threshold, so Table III summarises the tests, showing that in a large proportion of the cases we should reject the null-hypotheses. Thus we have significant evidence for a difference in most of the groupings.

Interestingly, ASN is the grouping with the lowest proportion of rejected null-hypotheses, while we might have expected that ASN would have a larger effect on the network aspects of the problem. However, remember the large Bonferroni correction in this case, which leads to a very conservative test.

Hypothesis testing could be expanded here in several ways.

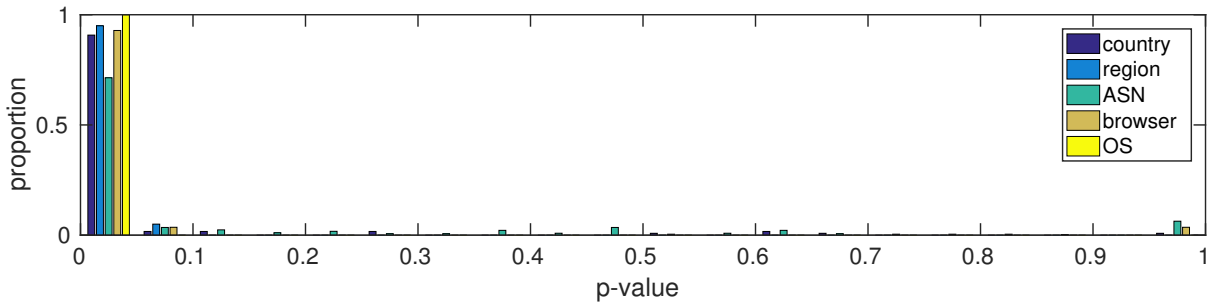

Fig. 3: *The distributions of p-values for the McNemar tests, applied across country, region, ASN, browser and OS. The distributions in all cases vary dramatically from a uniform distribution, with values heavily skewed in the direction of $p = 0$.*

TABLE III: *Hypothesis tests summaries for different covariates. $\tilde{N}$ is the number of groups left after excluding those with fewer than 500 observations. The "% of data" is that retained by this filtering. And the final column reports the proportion of McNemar tests for which we reject the null hypothesis over the $\tilde{N}$ groups.*

covariate	$\tilde{N}$	% of data	McNemar
country	119	99.6	0.840
region	20	100.0	0.950
ASN	458	93.1	0.555
browser	28	99.9	0.929
OS	14	100.0	1.000

Multiple-comparisons could be applied, for instance, to test differences between countries or some other covariate. However, in doing so there would be $O(n^2)$ comparisons for n countries, and these hypothesis tests are not all independent of each other, complicating the test procedure greatly. Moreover, much of this theory has been developed in domains where the each measurement requires a physical or social experiment, and therefore it seeks to make best use of a limited set of costly measurements. We have many measurements, and so these refinements are not needed. Instead, in our next step we opt to apply an approach called *Rasch analysis*.

B. Rasch Analysis

The disadvantage of the previous tests is they provide only a yes/no answer (or really a yes/maybe answer), while we would like, for instance, to be able to say how large the difference is. Here we use Rasch modelling to perform this analysis, but as we are not interested in the per-observation performance we use a grouping strategy. We start by defining G_k to be the kth group of observations determined by a covariate. We consider here two modelling approaches.

The first model still follows (2), but now we assume that the proficiency of each student is distributed as $\alpha_i \sim N(\lambda_k, \sigma^2)$, for $i \in G_k$, where λ_k is the group mean proficiency, and σ^2 is the common standard deviation within groups. The task is then to estimate λ_k and σ^2. The careful reader will note the additional assumptions introduced by this model.

The second approach takes a simpler model, that

$$\text{logit}(p_i^{(j)}) = \alpha_k - \beta_j. \tag{3}$$

for $i \in G_k$, We now only estimate a group proficiency α_k, not individual proficiencies. This has the disadvantage that it might not be able to fit the data as accurately, but it frees from distributional assumptions.

The former approach has been developed further, in that exact results are known, and there are off-the-shelf solvers using *Marginal Maximum Likelihood Estimation* (MMLE). We use IRTm [33], [34], a Matlab toolbox allowing quite general models to be estimated. Apart from its additional assumptions, in MMLE each categorical variable with m categories is deconstructed into m binary variables, each an indicator for one possible state of the original variable. For instance, the 119 countries in our data result in constructing a covariate vector consisting of 119 binary elements, leading to an estimation procedure taking considerable memory and time.

The results are illustrated below, in conjunction with those of the second approach, in which we assume each group consists of a set of repeated measurements. However, the standard Binomial Rasch models assumes each measurement is repeated a fixed number of times. For instance, in partial-credit Rasch models [35], a student may obtain some proportion of the marks for a question, but each student answers the same question, with the same total possible marks. But in our groupings, the number of "total marks" would vary, depending on the number of client observations that fall into the group. This case does not appear to have been treated in the literature, and hence we wrote our own Alternating Least Squares (ALS) algorithm (also in Matlab) to estimate the parameters.

The algorithm alternates between fitting the α_k and the β_j values, keeping the other parameters fixed. It also needs an additional fixed point of reference (because the variables α_k and β_j are not otherwise uniquely determined), which we fix, without loss of generality, by $E[\alpha_k] = 0$.

We assessed the two approaches in this (somewhat non-standard) application by comparing computation times, and Root-Mean-Square (RMS) fitting errors, as shown in Figure 4. All computations were made on an 8 core, Intel i7-6900K 3.2 GHz, running Linux Mint 18, and Matlab R1016b (the largest case for the MMLE algorithm did not complete within 24 hours and so is excluded), The ALS algorithm is orders of magnitude more accurate and faster, and so in what follows we focus on the ALS approach. Note also that the estimation errors in ALS reach a maximum in the order of 5%, which is reasonable given the problem of interest.

Ideally, we could group by all covariates at once. However,

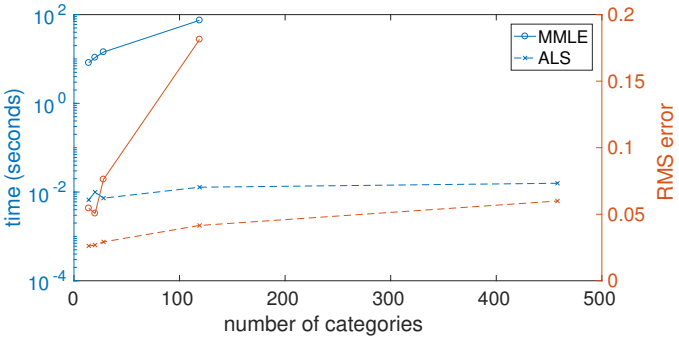

Fig. 4: *Computation times (blue) and RMS errors (red) for the two approaches to Rasch modelling.*

this results a *very* large number of covariates in the MMLE approach, while in the ALS approach, we end up with very few observations in many of the bins, due to the combinatorial number of bins. Thus we analyse each of the categories as separate groupings. As before, we consider only groupings with at least 500 observations.

The first detail to consider is the β_j values, indicating the difficulty in completing the two measurements (HTTP, and HTTPS). The estimated values are shown in Table IV, along with their difference. Larger values indicate additional difficulty with a measurement. The positive values of the difference indicate additional difficulty in the HTTPS measurements compared to HTTP. From all points of view, HTTPS is more difficult than HTTP.

We also see some consistency, namely, the differences in β_j are similar for location (country, region and ASN), and for end-point software (OS and browser), as you might expect. Notably the former group seems to have a larger impact on success than the latter, so it appears that while a client's device is important, the location from which one accesses the Internet is more important.

The second set of parameters to examine are the α_i values, namely, the ability or proficiency of a particular covariate group to perform any of the measurements, large values being better. Figure 5 shows the distribution of α_i values for the region, OS, and browser covariates. We see that they might be coarsely considered to follow a Normal distribution. The data by OS fit this assumption least well, but remember that there are only 14 values here, and we expect to see some natural variation here, because of measurement noise.

Note that we do not draw, from these values, inferences about the particular quality of HTTPS in a particular country (or other grouping). The α_k variables record the ability of a group to perform both HTTP and HTTPS measurement. This parameter separates out the "noise" inherited from the quality of Internet connections through a particular country from the HTTP v HTTPS question.

However, we also see outliers, here defined as those values that fall more than 1.96 times the standard deviation from the mean, *i.e.,* outside the 95th percentiles. These are not extreme outliers, but there may be some interest in these, so we have reversed the mapping (for these outliers only).

TABLE IV: *ALS estimates of Rasch "difficulty" parameters with different groupings. Larger values indicate a smaller chance of measurement success. Note the increase in difficulty for HTTPS.*

	country	region	ASN	browser	OS
β_{HTTP}	-5.26	-4.91	-6.07	-3.94	-3.99
β_{HTTPS}	-2.92	-2.74	-3.80	-2.29	-2.12
Difference	2.34	2.16	2.27	1.65	1.86

- **country**: five positive outliers: Suriname, Macau, Cyprus, Latvia, Korea; and one negative: Macedonia.
- **region**: no outliers.
- **ASN**: there is a list of 22 positive outliers, but only one negative: AS58539 (listed as China Telecom).
- **browser**: positive outlier: Amazon Silk and no negatives.
- **OS**: one positive outlier: ChromeOS and no negatives.

Some of these might be slightly surprising – for instance, many may not have expect Suriname to be in the list of positive outliers. However, it should be remembered that the measurement methodology filters participants who successfully complete the primer and results query successfully. So this result really says that, those who have a good connection, have a very good connection, *i.e.,* , if they complete the primer and result, they are very likely to be able to complete the other measurements.

Similarly, the positive outliers ChromeOS and Amazon Silk (the Kindle Fire's browser) are perhaps indications of consistency amongst all such devices, because of the stronger constraints on these devices. For instance, Amazon Silk routes requests through remote proxy servers powered by Amazon EC2, which provide high-performance connection speeds and computing power not normally available to a mobile form factor, and apparently improve the consistency of responses.

Hence, though these outliers may be interesting, the underlying point is not that any particular location or device group has a given α_k, so much as the parameter allows us to disentangle these effects from those of the two measurements (HTTP and HTTPS), and thus see the latter in isolation.

In summary, there are two main conclusions to be drawn.

1. The measurements show that there is significantly more difficulty in performing HTTPS than HTTP measurements. The difference is often small, necessitating some extra care in order to determine whether the difference is significant.
2. There are country, OS, and browser differences, mainly important through a small set that exhibits more extreme variations from the norm.

One last important insight is that the dependence on the covariates indicates that the results are not an artefact of APNIC's measurement infrastructure, as that would remove dependency on point of origin effects.

V. Conclusion and Further Work

Using a large set of measurements (data provided at bandicoot.maths.adelaide.edu.au/HTTPS/), and detailed statistical modelling we have shown that a small cohort of users in the real world will be adversely affected if HTTPS is adopted universally. That cohort is not a large proportion of Internet users, but those users deserve our attention.

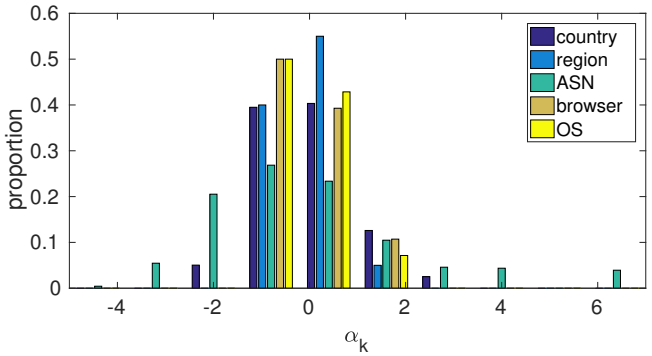

Fig. 5: *Histograms of α_k values for ALS algorithm. Positive values indicate a favourable probability of measurement success.*

We have categorised measurements by country and region, their provider (origin ASN), browser and operating system, and shown that all of these factors affect a client's facility with HTTPS. The range of factors points to a range of causes for the blockages. The browser/OS combination suggests a technological problem, but the other covariates suggest problems based in the network near the clients.

In the future, we plan to further investigate, and use the details of the analysis with extensions to understand better correlations in covariates, to help focus efforts onto relevant development to mitigate the problem.

The use of careful statistical methods was vital in this study. The underlying signal is weak, and hence required "amplification" and careful analysis so as to be able to make confident statements.

Acknowledgements

We would like to thank the Australian Research Council for funding through the Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS), and grant DP110103505.

The Javascript code used by APNIC originates in a library written by Emile Aben, RIPE-NCC.

APNIC Labs has received support and in-kind assistance from Google, ICANN, RIPE-NCC and ISC in its experiments.

References

[1] "HTTPS everywhere," on-line: downloaded April 24th, 2017, https://www.eff.org/https-everywhere.

[2] "HTTPS everywhere," Google I/O, https://docs.google.com/presentation/d/15H8Sj-Zol1tcum0CSylhmXns5r7cvNFtzYrcwAzkTjM/edit#slide=id.g12f3ee71d_10.

[3] "HTTPS as a ranking signal," https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html, August 2016.

[4] "The HTTPS-Only standard," on-line: downloaded April 24th, 2017, https://https.cio.gov/.

[5] A. Goldberg, R. Buff, and A. Schmitt, "A comparison of HTTP and HTTPS performance," in *CMG98*, 1998, http://www.cs.nyu.edu/artg/research/comparison/comparison.html.

[6] C. Coarfa, P. Druschel, and D. S. Wallach, "Performance analysis of tls web servers," *ACM Trans. Comput. Syst.*, vol. 24, no. 1, pp. 39–69, Feb. 2006. [Online]. Available: http://doi.acm.org/10.1145/1124153.1124155

[7] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafò, K. Papagiannaki, and P. Steenkiste, "The cost of the 'S' in HTTPS," in *10th ACM International on Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, New York, NY, USA, 2014, pp. 133–140. [Online]. Available: http://doi.acm.org/10.1145/2674005.2674991

[8] D. Beaver, "HTTP2 expression of interest," on-line: downloaded April 24th, 2017, July 2012, http://lists.w3.org/Archives/Public/ietf-http-wg/2012JulSep/0251.html.

[9] "HTTPS usage statistics on top websites," on-line: downloaded April 24th, 2017, https://statoperator.com/research/https-usage-statistics-on-top-websites/.

[10] "HTTPS usage," on-line: downloaded April 24th, 2017, https://www.google.com/transparencyreport/https/metrics/?hl=en.

[11] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, "Measuring HTTPS adoption on the web," in *USENIX Securit*, 2017.

[12] G. Michaelson and G. Huston, "Experience with large-scale end user measurement techniques," in *Telecommunication Networks and Applications Conference (ATNAC), 2014 Australasian*. IEEE, 2014, pp. 1–5.

[13] B. Wright and M. Mok, *Introduction to Rasch Measurement: Theory, Models, and Applications*. Journal of Applied Measurement, 2004, ch. An Overview of the Family of Rasch Measurement Models, jampress.org/irmch1.pdf.

[14] B. D. Wright, "Solving measurement problems with the Rasch model," *Journal of Educational Measurement*, vol. 14, no. 2, pp. 97–116, 1977. [Online]. Available: http://www.jstor.org/stable/1434010

[15] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS certificate ecosystem," in *ACM Sigcomm Internet Measurement Conference*, 2013, pp. 291–304. [Online]. Available: http://doi.acm.org/10.1145/2504730.2504755

[16] B. Laurie and C. Doctorow, "Secure the internet," *Nature*, vol. 491, pp. 325–6, 2012.

[17] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the HTTPS protocol," *IEEE Security Privacy*, vol. 7, no. 1, pp. 78–81, Jan 2009.

[18] D. Ranathunga, M. Roughan, H. Nguyen, P. Kernick, and N. Falkner, "Case studies of SCADA firewall configurations and the implications for best practices," *IEEE Transactions on Network and Service Management*, 2016, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7529047&isnumber=5699970.

[19] "SSL/TLS - typical problems and how to debug them," on-line: downloaded April 24th, 2017, https://maulwuff.de/research/ssl-debugging.html.

[20] S. Fahl, Y. Acar, H. Perl, and M. Smith, "Why Eve and Mallory (also) love webmasters: A study on the root causes of SSL misconfigurations," in *ASIA CCS*, 2014.

[21] M. Casado and M. J. Freedman, "Peering through the shroud: The effect of edge opacity on IP-based client identification," in *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation*, ser. NSDI'07, 2007, pp. 13–13. [Online]. Available: http://dl.acm.org/citation.cfm?id=1973430.1973443

[22] S. Burnett and N. Feamster, "Encore: Lightweight measurement of web censorship with cross-origin requests," in *ACM SIGCOMM*, 2015, pp. 653–667. [Online]. Available: http://doi.acm.org/10.1145/2785956.2787485

[23] J. P. Shaffer, "Multiple hypothesis testing," *Annu.Rev.Psychol.*, vol. 46, pp. 561–584, 1995.

[24] A. Agresti, *Categorical Data Analysis*, 2nd ed. Wiley, 2002.

[25] "ngx_txid," https://github.com/APNIC-Labs/ngx_txid, accessed May 16th, 2017.

[26] "nginx," http://nginx.org/, accessed May 16th, 2017.

[27] "Standard country or area codes for statistical use (M49)," https://unstats.un.org/unsd/methodology/m49/, accessed May 16th, 2017.

[28] A. H. Rasti, N. Magharei, R. Rejaie, and W. Willinger, "Eyeball ASes: From geography to connectivity," in *ACM Sigcomm IMC*, Melbourne, Australia, 2010.

[29] "GeoIP databases & services: Industry leading IP intelligence," https://www.maxmind.com/en/geoip2-services-and-databases, accessed May 16th, 2017.

[30] "A Python implementation of the UA parser," https://github.com/ua-parser/uap-python, accessed May 16th, 2017.

[31] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978. [Online]. Available: http://www.jstor.org/stable/2683468

[32] H. Wickham and L. Stryjewski, "40 years of boxplots," had.co.nz, Tech. Rep., 2012, http://vita.had.co.nz/papers/boxplots.html.

[33] J. Braeken and F. Tuerlinckx, "Investigating latent constructs with item response models: a MATLAB IRTm toolbox," *Behavior Research Methods*, vol. 414, no. 4, pp. 1127–37, 2009.

[34] "IRTm," https://ppw.kuleuven.be/okp/software/irtm/.

[35] G. M. Masters, "A Rasch model for partial credit scoring," *Psychometrika*, vol. 47, no. 2, pp. 149–174, 1982.