

# Exploring HTTP Header Manipulation In-The-Wild

Gareth Tyson  
Queen Mary University of  
London  
gareth.tyson@qmul.ac.uk

Ignacio Castro  
Queen Mary University of  
London  
i.castro@qmul.ac.uk

Shan Huang  
Queen Mary University of  
London  
shan.huang@qmul.ac.uk

Vasile C. Perta  
Sapienza University of Rome  
perta@di.uniroma1.it

Steve Uhlig  
Queen Mary University of  
London  
steve.uhlig@qmul.ac.uk

Felix Cuadrado  
Queen Mary University of  
London  
felix.cuadrado@qmul.ac.uk

Arjuna Sathiaselan  
University of Cambridge  
arjuna@cl.cam.ac.uk

## ABSTRACT

Headers are a critical part of HTTP, and it has been shown that they are increasingly subject to middlebox manipulation. Although this is well known, little is understood about the general regional and network trends that underpin these manipulations. In this paper, we collect data on thousands of networks to understand how they intercept HTTP headers in-the-wild. Our analysis reveals that 25% of measured ASes modify HTTP headers. Beyond this, we witness distinct trends among different regions and AS types; *e.g.*, we observe high numbers of cache headers in poorly connected regions. Finally, we perform an in-depth analysis of the types of manipulations and how they differ across regions.

## 1. INTRODUCTION

HTTP underpins one of the most successful inventions in recent history: the World Wide Web. Although HTTP has received much attention, an aspect that remains understudied is that of *headers*. These are attribute-value pairs that are embedded within all HTTP messages. While they are well documented within standards, little is known of their practical usage at scale. This is exacerbated by the increasing propensity for *middleboxes* to manipulate headers (*e.g.*, for caching, monitoring, censorship). While it is known that users' private information can be exposed and tracked through middleboxes [5], we posit that such middlebox injections may also reveal a wealth of information about the networks engaged in the header manipulation. We therefore ask *which insights can be extracted from header manipulations performed across regions and networks?*

To gain an understanding of the use of HTTP headers, we begin by collecting data on almost 1 million websites (§2). We then present a novel methodology to collect large-scale data on global HTTP middlebox header manipulation. This involves creating a

measurement platform using the Hola peer-to-peer proxy network [2] (§3). Using this platform, we craft and forward HTTP requests via third party networks to a web server we control. By monitoring both the request and response endpoints, we can discover manipulations performed by these networks. Exploiting Hola, we launch over 400k HTTP queries from 143k vantage points in 3818 Autonomous Systems (ASes) — one of the largest studies of its kind. Unlike techniques using controlled infrastructures (*e.g.*, Planetlab), this provides unique visibility on a range of network types in countries rarely studied, *e.g.*, over 400 ASes in Africa (§4).

In this paper we explore the propensity of different network types and regions to manipulate HTTP headers, in terms of both frequency (§5), and content (§6). We find that header manipulation is remarkably widespread: hosts in 25% of measured ASes witness header modifications at least once. Despite this headline figure, our data shows that the density of middlebox injections varies dramatically across regions, with networks in technologically advanced countries abandoning the use of caches.

A common theme in our findings is the lack of standards adherence; we find thousands of new non-standard headers returned by web servers (§2), alongside networks injecting over 40 non-standard headers (§6). Although this form of extensibility could be considered desirable, our findings suggest that it runs the risk of bloating the protocol in a way that breaks down common understanding. For example, amongst other things, we observe middleboxes injecting cookies, disabling performance-enhancing features, caching against our dictates and adding private information into requests.

Opinion on whether or not this is damaging is divided, however, we believe that the Web community is at a juncture at which they should decide if they wish to support (*e.g.*, via mTLS [20]) or undermine this behaviour. This is particularly important considering recent efforts towards new web protocol standards [8, 13]. As such, this paper provides a unique insight into in-the-wild practices, and how future protocol decisions on header manipulation may affect operators in different regions.

## 2. A PRIMER ON HTTP HEADERS

Before studying middlebox header interference, it is important to understand (*i*) what HTTP headers are, and (*ii*) how they are currently used by web servers. HTTP was developed in 1991 with the advent of HTTP 0.9. Unlike subsequent versions, HTTP 0.9 was





This still leaves the small possibility that some end hosts were injecting headers traditionally injected by in-network middleboxes (e.g., by installing Squid locally). To identify this, we separated all samples into their origin ASes and extracted ASes for which we have over 5 samples. We then searched for peers that were the only nodes in the AS to make a given change. The rationale was that such peers may be injecting headers locally, thereby differentiating themselves from the remaining samples in the AS. We then manually inspected these manipulations, discovering the presence of various headers containing things like localhost or 127.0.0.1. Hence, we compiled a simple set of rules to filter such fetches from the dataset (removing 1016 samples). Following this step, out of 405k fetches, only 88 requests and 92 responses remained that were exclusively manipulated by a single node in an AS. These showed no explicit indications of local manipulation. We are therefore confident that any remaining modifications were introduced somewhere along the intermediate peer  $\rightarrow$  server path and not by the peer itself. Our data is available at [26].

### 3.2.3 Peer Metadata

To get an idea of the networks we have sampled, we augment each Hola peer with metadata. First, we geolocate every peer IP address. It is well known that individual geolocation databases contain errors [21]. Hence, we use majority voting from 10 separate databases.<sup>2</sup> These IPs correspond to either the public address of the peer or, alternatively, an intermediate TCP-terminating middlebox (generally these are in the same network, but not always). Further, we limit our granularity to country-level geolocation, which reports high accuracy [24]. We then map each peer to their respective AS using the same databases. Finally, we tag each peer with its AS *type*. We experimented with three off-the-shelf AS classifiers: (i) The CAIDA AS Rankings [1]; (ii) A classifier provided by Dhamdhare *et al.* [11]; and (iii) A classifier provided by Dimitropoulos *et al.* [12]. The first provided high recall, but only a coarse (3 category) classification, whilst the other two provided much lower recall but a slightly finer-grained classification. In this paper we utilise the third classifier, as this offered similar recall to (ii) but with a finer grained classification.

## 3.3 Data Limitations and Ethics

It is important to outline key limitations in the dataset. Most obvious is the use of a third party system (Hola), which could introduce unexpected behaviour. To mitigate this risk, we have extensively tested Hola to identify unusual behaviour. Fortunately, the only anomalies seen (e.g., injecting X-Hola-Error headers) can be cleaned from the data. Another consideration is that our measurements are not an unbiased sample of networks around the globe. As a general-purpose tool, the bias induced by more Western “tech-savvy” crowd sourced measurement platforms may be reduced though. We also acknowledge that Hola cannot tell us *where* the manipulations happen. That said, by filtering locally injected headers, we can be confident that any modifications were performed on the peer  $\rightarrow$  server path. Of course, this leaves the possibility that the middlebox exists in a different network to the client, however, this still reveals a network that is subjected to manipulations. Hence, we temper our analysis with this consideration. Finally, it is worth noting that the use of Hola may raise ethical questions, as we are forwarding requests through users’ machines. To mitigate these, we *only* forward requests to our own web server (containing a “Hello

<sup>2</sup>OpenIPMap; MaxMind; Whois; RIRs’ allocation files for RIPE, APNIC, ARIN, AFRINIC and LACNIC; Team Cymru; and Reverse DNS lookups, which were used to infer the location based on city, country codes (CCs) or airports in the reverse names.

World” page). As such, there is no risk of triggering censorship. Further, Hola informs its users about how it operates. Hence, users are already aware that they operate as proxy points. We have obtained IRB approval.

## 4. CHARACTERISING HOLA

We begin by briefly describing Hola’s scale and distribution, as captured by our measurements. In total, we have collected data covering 143,288 IP addresses in 3818 ASes. To add context, this can be compared against other state-of-the-art measurement platforms, e.g., Dasu (2431 ASes [22]), although it should be noted that each platform supports very different features. Figure 2 presents the density of (i) Hola users, as measured by IP addresses; and (ii) unique ASes hosting Hola nodes. It can be seen that Hola possesses vantage on every region in the world: 216 countries and territories. Consequently, the beauty of Hola, compared to other platforms, is that it can provide very wide vantage on these global users. This is an attractive property, as we see data regarding many areas that often are excluded from such studies (e.g., Africa). In terms of IP addresses, the best represented country is the US, whereas, in terms of ASes, Ukraine is best represented. At first, we thought this may be a geolocation error, however, we manually confirmed the veracity. This is perhaps a product of the differing ISP market structures seen in these countries. But, to address this imbalance, we primarily characterise manipulations on a *per-AS* basis (rather than *per-IP*), to reduce the bias introduced by highly populous ASes.

To better understand the nature of Hola’s network sampling, Table 1 provides a breakdown of the types of networks seen (based on the classifier in §3.2.3). It only lists statistics on the ASes that were successfully classified. Customer indicates a variety of commercial networks that do not offer residential Internet access (e.g., web hosts), whilst Network Information Centres (NICs) are ASes that host important infrastructure (e.g., root domain name servers). Full details of the AS meanings and methodology can be found in [12]. The analysis confirms a wide sample of networks types. We argue this diversity is highly attractive for studying header manipulations, as it gives a wide vantage on global behaviour. More generally, this also confirms Hola’s efficacy as a powerful platform for web data collection.

## 5. WHO MANIPULATES HEADERS?

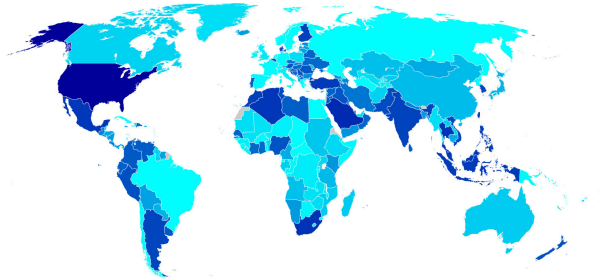
### 5.1 Measuring Across Networks

First, we inspect how header manipulations occur across the sampled ASes. 21% of the ASes have requests manipulated, compared to 19% for responses. Overall, 25% contain sessions that manipulate headers at least once. We discover that classifying ASes in this manner, however, is not straightforward. To highlight this, Figure 3 presents the top 100 most sampled ASes that contain modifications to headers; each AS is separated into requests that were manipulated and requests that were not. Curiously, it can be seen that many ASes contain both modified and non-modified requests. Evidently, this is problematic when classifying an entire AS. Inspection suggests that this occurs for a variety of reasons, most likely due to a diversity of paths and middlebox deployments within a large AS and its interconnected networks (as well as misconfiguration [25]).

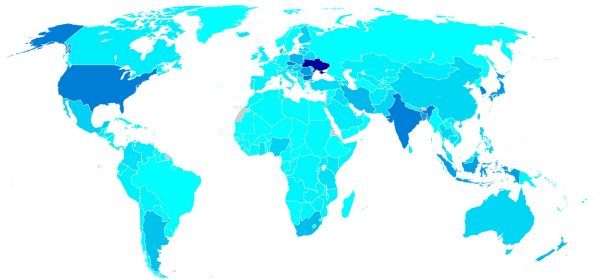
This also raises the question of what *types* of ASes witness these changes. To answer this, we separate all ASes into the classifications presented in §3.2.3. Figure 4 breaks down ASes into their types, and then presents CDFs of the percentage of samples in each AS that modify headers. Table 1 further provides statistics on the

AS Classification	#ASes	#Samples Modified		#Samples Not Modified		%Samples modified		% ASes that modify at least once		Mean % Samples Modified Per AS	
		Req	Res	Req	Res	Req	Res	Req	Res	Req	Res
Net Info Centre	41	992	585	8059	8466	10.96	6.46	34.15	39.02	7.39	7.50
Customer	348	2432	978	29086	30536	7.72	3.10	15.52	15.80	5.58	4.72
Regional ISP	798	9395	4184	159589	164798	5.56	2.48	32.08	29.82	7.06	4.40
Tier-1	15	47	21	2998	3024	1.54	0.69	26.67	33.33	10.33	7.38
University	62	9	7	1009	1011	0.88	0.69	4.84	3.23	1.97	1.94

Table 1: Statistics per AS types (excluding samples that could not be classified into AS type).



(a) Number of IP addresses per country (max 2803)



(b) Number of ASes per country (max 325)

Figure 2: Map of IP and AS samples from Hola; dark blue represents highest density.

behaviour of ASes within each group. It can be seen that the different AS types exhibit a range of trends. University ASes stand out as having very few modifications (3% of ASes for responses, 5% for requests). Overall, networks classified as NICs are the most likely to manipulate headers (39% of ASes for responses, 34% requests). While it is hard to definitively state the reasons, it is well known that university networks tend not to transparently intercept traffic [10]. NICs and hosting centres, on the other hand, may deploy dedicated infrastructure to optimise their activities. It can similarly be seen that ISPs tend to have high proportions of manipulations. 32% of regional ISPs inject headers into requests at least once. Again, this reflects commonly understood business models, in which these edge networks may wish to reduce their egress traffic by utilising cache middleboxes. Studying middleboxes without an appreciation of the types of networks sampled therefore could be quite misleading.

The above indicates that networks might deploy header-manipulating middleboxes in only a subset of locations. To expand on this, we can also check if the manipulated samples seen within an AS always make the *same* modifications. To explore this, we subset samples to leave only those in which we detect manipulation. We

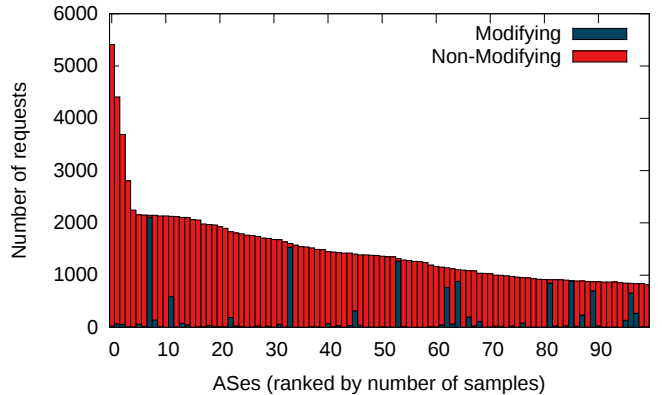


Figure 3: Number of requests for the top 100 ASes sampled that modify headers at least once (manipulated vs. not manipulated samples per AS).

then map them into their respective ASes, and filter out any ASes in which we only have a single IP address. For each AS,  $i$ , we extract the full set of headers that are manipulated,  $H^i$ . For each header,  $h \in H^i$ , we calculate the percentage of samples in the AS where  $h$  is modified. For example, if all samples in an AS modify the `Server` header, we assign that AS a `Server` value of 100%. By averaging the percentages within an AS for all manipulated headers,  $\bar{H}^i$ , a single value can be obtained per-AS. Figure 5 presents a CDF of the per-AS averages. It can be seen that only 44% of ASes exclusively record the *same* modifications for all requests, whilst this is 32% for responses. This leaves a significant fraction of samples from an AS that vary the headers that are manipulated. This confirms that users in individual ASes are subject to a mix of manipulations, and the heterogeneity of these changes is high.

## 5.2 Measuring Across Regions

Anecdotally, it is known that different regions have different styles of network deployment. With our data it is possible to evaluate the prevalence of header-manipulating middleboxes across different regions. We sample all continents: Africa (442 ASes), Asia (1114), Europe (1581), Middle East (209), North America (394), Oceania (154) and South/Central America (182).<sup>3</sup> Figure 6 presents the fraction and number of ASes per country (top 30 countries) that record manipulated headers. Absolute numbers are shown with the bars (right axis), whilst the fraction of modifying ASes is shown by the line (left axis). We only include countries where we have sufficient sampling, of at least 10 ASes. Broadly speaking, we see sim-

<sup>3</sup>Note that some ASes have a presence in multiple continents, and therefore these samples are split between regions.

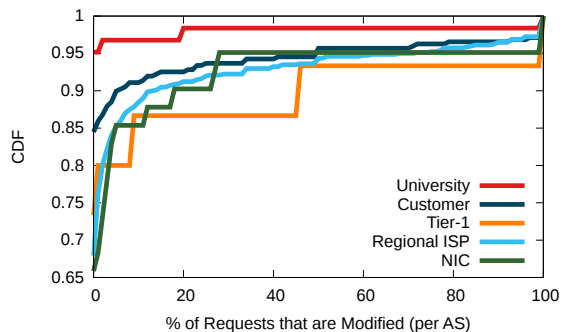


Figure 4: CDF of the percentage of requests that are modified on a per AS basis. Data is separated into network types.

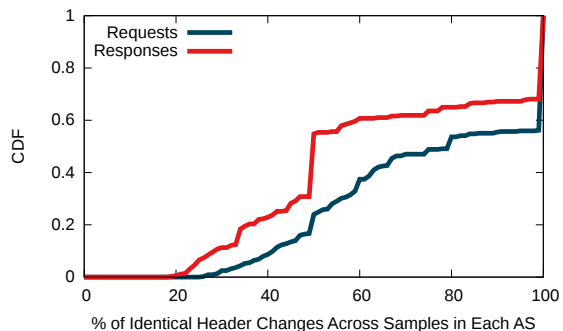
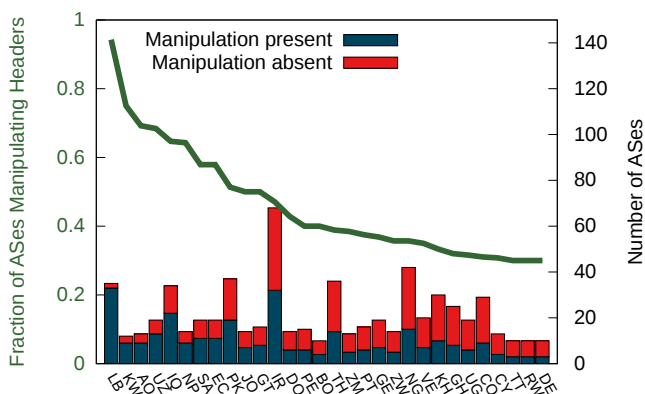


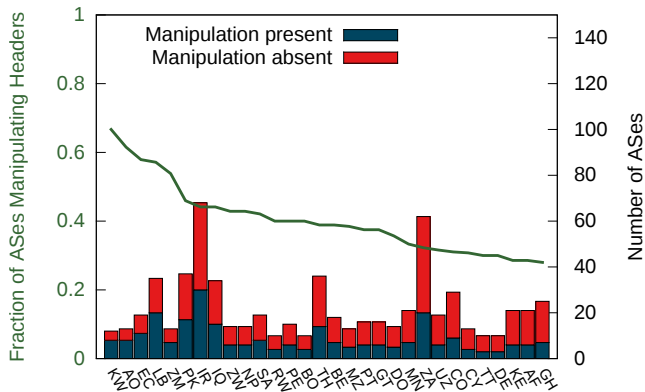
Figure 5: CDF of percentage of Hola peers in an AS that see the same headers changed. Each AS is represented by the average taken across all headers manipulated.

ilar fractions for both request and response manipulation per country (correlation 0.73), however, there are noteworthy discrepancies, *e.g.*, in Jordan, only 21% of ASes manipulate responses, compared to 50% that manipulate the requests. This confirms the need to use both endpoints to detect interference.

The reasons for these countries exhibiting such trends are likely diverse. To get an idea of these, we compare them against well known technological, economic and societal measures. To this end, we collect metrics from (i) The Web Index: a composite ranking that measures a country’s online capabilities; and (ii) The World Bank: an organisation that compiles economic data. Figure 7 presents the Spearman correlation coefficient for the fraction of ASes that manipulate headers in a given country *vs.* the metrics taken from the Web Index and the World Bank. Due to space constraints, we do not delineate the nature and methodology behind each metric; instead, full details can be found at [31, 32]. However, it can be seen that there is a strong *negative* correlation between a country’s header-manipulating middlebox deployment and its positions in these metric rankings. High middlebox deployments are correlated with less developed countries (as measured via the metrics presented). This can generally be observed too, *e.g.*, 36% of African ASes manipulate requests compared to just 8% in Europe. This is somewhat counterintuitive as one might expect more developed countries to have larger infrastructure deployments. The most correlated factors in Figure 7 pertain to how widely available services are to individuals in a country. It appears that nations with less developed infrastructures rely more heavily on the use of middleboxes (most prominently caches). To explain this, we contacted several operators in both Europe and Africa, who confirmed



(a) Request headers



(b) Response headers

Figure 6: The bars show the *number* of ASes that modify (a) request, and (b) response headers per country. The green line shows the *fraction* of ASes that modify per country.

the veracity of our findings. The main reason listed by European operators was the progressive reductions in network transit prices, alongside greater peering via Internet eXchange Points [6]. In conjunction with higher line rates, this meant that such operators may actually have to pay more for running multi-Gbps web caches than simply contacting the origin via peering or transit. This, however, was not the case for African operators, who still complained of exorbitant transit costs and a distinct lack of peering [14]. Another frequently cited reason by European operators was the deployment of dedicated provider-specific caches in their networks (*e.g.*, Google Caches, Netflix Appliances). Considering the bulk of traffic handled by these websites, the need for augmentary transparent caches was radically reduced. But, again, the presence of these in Africa is still limited [15]. These reasons have meant that the business case for transparent caching in developed regions has reduced, whereas it is still strong in developing countries.

## 6. WHAT MODIFICATIONS ARE MADE?

The previous section has shown how regions and networks differ in their frequency of header manipulation. Next, we inspect the *content* of the changes made. To explore this, we manually classify all manipulated headers into functional categories using various resources (*e.g.*, RFCs, W3C, IANA, blogs). Table 2 presents the 5 categories, including the number of injections/modifications we



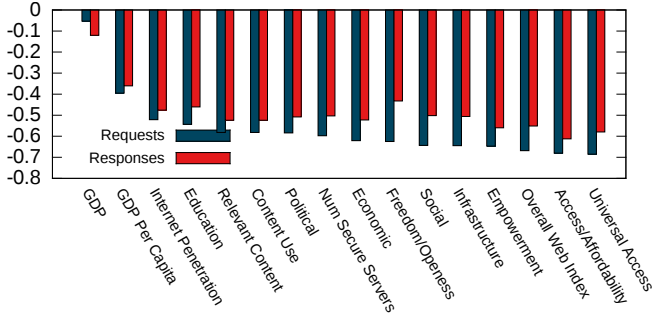


Figure 7: Correlation coefficient for fraction of ASes that manipulate in a country vs. metrics taken from the Web Index and World Bank (only countries with 10+ AS samples).

Header Type	#Headers in Category		Total #Headers injected/modified	
	Request	Response	Request	Response
Cache	4	9	8419	3799
Operational	12	9	5090	63
Feature	8	3	639	1884
Information	1	5	20	20
Unknown	4	3	10	41

Table 2: Number of headers in each category, and number of instances of headers being injected/modified (count based on unique IP addresses).

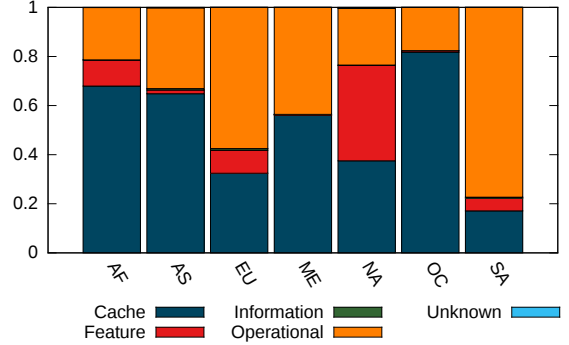
see globally; a full list of the classifications can be found in [26]. Figure 8 separates all IP samples into their continents, and presents the fraction of manipulations that fall into each category. We emphasise that we are *not* trying to classify middleboxes themselves; instead, our focus is on the individual headers they manipulate. We now describe each category, and their regional presence.

## 6.1 Caching Headers

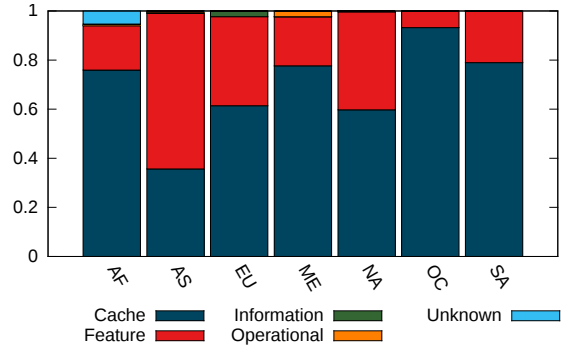
Caching headers are those that add information relating to cache operations (*e.g.*, cache hits); these account for a significant fraction of manipulations. In-line with earlier discussion (§5.2), we find that regions with higher transit prices tend to have a higher proportion of caching headers [6]. For instance, 93% of IPs (79% of ASes) sampled in Oceania (OC) witness cache-related *response* manipulations (OC has amongst the most expensive transit in the world [6]); it is also common in other regions with high transit prices, *i.e.*, Africa (AF), South/Central America (SA) and the Middle East (ME). In contrast, well-networked regions with reduced transit prices such as Europe (EU) and North America (NA), show the fewest cache-related manipulations. Figure 8 also shows that continents that see high proportions of cache-related *response* manipulations, also see similar trends for their requests. The only notable exception is SA, which sees a high proportion of cache headers injected into responses (79% of IPs, 77% of ASes), but few in requests (17% IPs, 18% ASes); the exact reasons for this trend are unclear.<sup>4</sup>

Next, to inspect the specific header attributes, Figure 9 presents the global most frequently modified headers. Many at the top are cache related. This is largely because of a few specific headers that are regularly manipulated. 5% of measured ASes edit our preset

<sup>4</sup>We are *not* stating that SA does not have caches — only that it is not injecting a high proportion of cache request headers.



(a) Breakdown of request header types



(b) Breakdown of response header types

Figure 8: Fraction of header manipulations in each category. Counts based on unique IP address samples.

**Cache-Control *response* header.** The majority of these changes (99%) add extra settings (must-revalidate and no-store), whereas others change the existing ones, *e.g.*, increasing the max-age setting from 0 to 60. We also note that 3 ASes completely remove the Cache-Control header set by the server. This enables both browser and in-network caching against our will. Other prominently injected headers include X-Cache (9% of ASes), and X-Cache-Lookup (6.9% of ASes). These non-standard headers are used by a variety of cache implementations to report cache hits (making it trivial to detect caches). We note they may be used for attacks [16]. Their presence also allows us to check for middleboxes that might be caching uncacheable content against the instructions of our server. We find one South American AS doing this. Another cache-related header (injected in 18 ASes) is *Age*, which states the age of the object in the cache. 110 samples have this added, with 96 setting values greater than 0, further revealing caches that store uncacheable content.

## 6.2 Operational Headers

Operational headers are those that pertain to infrastructural operations other than caching, *e.g.*, tagging middleboxes, firewalls, recording IPs. Globally, we only observe 18 Middle Eastern, 6 Asian and 3 African ASes injecting operational *response* headers, all of which are non-standard (various bespoke functions, *e.g.*, vendor specific firewalls).

Far more networks inject operational *request* headers though. In fact, in SA, the vast majority of request manipulations are operational. The high frequency is due to two key headers. The most frequently injected one is *Via*, as dictated by RFC 2616. This header



across regions. For instance, our results show that well connected regions such as Europe expose fewer caching headers than regions such as Africa, where transit is costly. While revealing, it should also be noted that exposing this information is a potential security threat, due to the frequent use of old and vulnerable middlebox software [7].

Our future work will focus on exploring how these trends evolve. Interestingly, many HTTP/2.0 browser implementations are following an encrypt everything model, which will undermine some middlebox functions. This is perhaps concerning as our work indicates a widespread dependence on their functionality. Hence, ISPs may endeavour to find ways around this [27]. This is particularly the case for security and performance oriented middleboxes, which may be considered critical to business operations. Hence, we believe that the continued monitoring of this process could offer fascinating insight into how network operators react and optimise to changes in Web protocols.

## 9. REFERENCES

- [1] The CAIDA UCSD AS classification dataset. [http://www.caida.org/data/as\\_classification](http://www.caida.org/data/as_classification).
- [2] Hola. <https://hola.org/>.
- [3] Luminati FAQ. <https://luminati.io/faq>.
- [4] Verizon injecting perma-cookies to track mobile customers, bypassing privacy controls. <https://www.eff.org/deeplinks/2014/11/verizon-x-uidh>.
- [5] AT&T stops using invasive perma-cookies, but it may turn them back on. <http://www.wired.com/2014/11/att-hits-pause-privacy-busting-perma-cookie-test>, 2014.
- [6] The relative cost of bandwidth around the world. <https://blog.cloudflare.com/the-relative-cost-of-bandwidth-around-the-world>, 2014.
- [7] Squid: Security vulnerabilities. [https://www.cvedetails.com/vulnerability-list/vendor\\_id-823/Squid.html](https://www.cvedetails.com/vulnerability-list/vendor_id-823/Squid.html), 2016.
- [8] CARLUCCI, G., DE CICCO, L., AND MASCOLO, S. Htt over udp: an experimental investigation of quic. In *Proc. ACM SAC* (2015).
- [9] CHUNG, T., CHOFFNES, D., AND MISLOVE, A. Tunneling for transparency: A large-scale analysis of end-to-end violations in the internet. In *Proc. ACM IMC* (2016).
- [10] DETAL, G., HESMANS, B., BONAVENTURE, O., VANAUBEL, Y., AND DONNET, B. Revealing middlebox interference with tracebox. In *Proc. ACM IMC* (2013).
- [11] DHAMDHERE, A., AND DOVROLIS, C. Twelve years in the evolution of the internet ecosystem. *IEEE/ACM Transactions on Networking (ToN)* (2011).
- [12] DIMITROPOULOS, X., KRIOUKOV, D., RILEY, G., ET AL. Revealing the autonomous system taxonomy: The machine learning approach. In *Proc. PAM* (2006).
- [13] ELKHATIB, Y., TYSON, G., AND WELZL, M. Can spy really make the web faster? In *IFIP Networking* (2014).
- [14] FANOU, R., FRANCOIS, P., AND ABEN, E. On the diversity of interdomain routing in africa. In *Proc. PAM* (2015).
- [15] FANOU, R., TYSON, G., FRANCOIS, P., AND SATHIASEELAN, A. Pushing the frontier: Exploring the african web ecosystem. In *Proc. WWW* (2016).
- [16] HUANG, L.-S., CHEN, E. Y., BARTH, A., RESCORLA, E., AND JACKSON, C. Talking to yourself for fun and profit. In *Proc. Workshop on Web Security and Privacy* (2011).
- [17] KLYNE, G. Message headers. <http://www.iana.org/assignments/message-headers/message-headers.xhtml>, 2015.
- [18] KREIBICH, C., WEAVER, N., NECHAEV, B., AND PAXSON, V. Netalyzer: Illuminating the edge network. In *Proc. ACM IMC* (2010).
- [19] NAYLOR, D., FINAMORE, A., LEONTIADIS, I., GRUNENBERGER, Y., MELLIA, M., MUNAFÒ, M., PAPAGIANNAKI, K., AND STEENKISTE, P. The cost of the S in HTTPS. In *Proc. ACM CoNEXT* (2014).
- [20] NAYLOR, D., SCHOMP, K., VARVELLO, M., LEONTIADIS, I., BLACKBURN, J., LÓPEZ, D. R., PAPAGIANNAKI, K., RODRIGUEZ RODRIGUEZ, P., AND STEENKISTE, P. multi-context TLS (mctls): Enabling secure in-network functionality in TLS. In *Proc. ACM SIGCOMM* (2015).
- [21] POESE, I., UHLIG, S., KAAFAR, M. A., DONNET, B., AND GUEYE, B. Ip geolocation databases: Unreliable? *SIGCOMM CCR* (2011).
- [22] SANCHEZ, M. A., OTTO, J. S., BISCHOF, Z. S., CHOFFNES, D. R., BUSTAMANTE, F. E., KRISHNAMURTHY, B., AND WILLINGER, W. A measurement experimentation platform at the internet's edge. *IEEE Transactions on Networking (ToN)* (2014).
- [23] SCOTT, W., BHORASKAR, R., AND KRISHNAMURTHY, A. Understanding open proxies in the wild. Technical Report. <http://netlab.cs.washington.edu/squid/paper.pdf>, 2015.
- [24] SHAVITT, Y., AND ZILBERMAN, N. A geolocation databases study. *IEEE Journal on Selected Areas in Communications* (2011).
- [25] SHERRY, J., HASAN, S., SCOTT, C., KRISHNAMURTHY, A., RATNASAMY, S., AND SEKAR, V. Making middleboxes someone else's problem: network processing as a cloud service. *SIGCOMM CCR* (2012).
- [26] TYSON, G. Dataset. <http://bit.ly/1qg7PT4>.
- [27] VALLINA-RODRIGUEZ, N., AMANN, J., KREIBICH, C., WEAVER, N., AND PAXSON, V. A tangled mass: The android root certificate stores. In *Proc. ACM CoNEXT* (2014).
- [28] VALLINA-RODRIGUEZ, N., SUNDARESAN, S., KREIBICH, C., AND PAXSON, V. Header enrichment or ISP enrichment? Emerging privacy threats in mobile networks. In *Proc. HotMiddlebox* (2015).
- [29] VALLINA-RODRIGUEZ, N., SUNDARESAN, S., KREIBICH, C., WEAVER, N., AND PAXSON, V. Beyond the radio: Illuminating the higher layers of mobile networks. In *Proc. MobiSys* (2015).
- [30] WEAVER, N., KREIBICH, C., DAM, M., AND PAXSON, V. Here be web proxies. In *Proc. PAM* (2014).
- [31] Web index. <http://thewebindex.org>, 2016.
- [32] World bank data repository. <http://data.worldbank.org>, 2016.
- [33] ZHOU, Z., AND BENSON, T. Towards a safe playground for HTTPS and middle boxes with QoS2. In *Proc. HotMiddlebox* (2015).