Detecting Cellular Middleboxes using Passive Measurement Techniques

Utkarsh Goel[1], Moritz Steiner[2], Mike P. Wittie[1],
Martin Flack[2], and Stephen Ludin[2]

[1] Department of Computer Science, Montana State University, Bozeman MT 59717
utkarsh.goel@montana.edu, mwittie@cs.montana.edu
[2] Akamai Technologies, Inc., San Francisco, CA 94103
{moritz, mflack, sludin}@akamai.com

Abstract. The Transmission Control Protocol (TCP) follows the end-to-end principle – when a client establishes a connection with a server, the connection is only shared by two physical machines, the client and the server. In current cellular networks, a myriad of middleboxes disregard the end-to-end principle to enable network operators to deploy services such as content caching, compression, and protocol optimization to improve end-to-end network performance. If server operators remain unaware of such middleboxes, TCP connections may not be optimized specifically for middleboxes and instead are optimized for mobile devices. We argue that without costly active measurement, it remains challenging for server operators to reliably detect the presence of middleboxes that split TCP connections. In this paper, we present three techniques (based on latency, loss, and characteristics of TCP SYN packets) for server operators **to passively identify Connection Terminating Proxies (CTPs) in cellular networks**, with the goal to optimize TCP connections for faster content delivery. Using TCP and HTTP logs recorded by Content Delivery Network (CDN) servers, we demonstrate that our passive techniques are as reliable and accurate as active techniques in detecting CTPs deployed in cellular networks worldwide.

Keywords: cellular, middleboxes, split TCP, network measurement

1 Introduction

The Transmission Control Protocol (TCP), Hyper Text Transport Protocol (HTTP) and secure HTTP (HTTPS) were originally designed with the assumption that clients communicate over end-to-end connections with servers. However, given the different types of networks involved in an end-to-end connection between cellular clients and servers (such as the radio network, the cellular backbone, and the public Internet), optimizing communication for each of these networks independently improves the overall performance of the end-to-end connections between clients and servers [5][10][11]. One of the techniques used by cellular carriers to improve the communication performance in their networks is to deploy Connection Terminating Proxies (CTPs) that split TCP connections between clients and servers [9][13]. CTPs allow cellular carriers to speed up TCP

transfers between devices and the cellular gateways to the Internet through TCP optimization, content caching, and bandwidth throttling.

Content Distribution Networks (CDNs), cloud providers, or other server providers on the Internet are mostly unaware of specific CTPs deployed by individual cellular carriers. As a result, servers may not optimize their connections for CTPs, but optimize connections for the mobile device instead. We believe that if server providers are made aware of the presence of CTPs, TCP configurations could be fine-tuned to improve content delivery to the middlebox and to the end-user [7]. However, without expensive active network measurements on mobile devices, it remains challenging for server operators to reliably detect the presence of CTPs and optimize connections accordingly [17].

In this study, we propose three techniques to **passively** detect the presence of CTPs in cellular networks, using TCP and HTTP logs recorded by Akamai's geographically distributed CDN servers. Our first technique compares **latency** estimated by clients and servers for TCP connections. The second technique compares the **packet loss** experienced by CDN servers for HTTP and HTTPS sessions. Our third technique analyzes characteristics of **TCP SYN** packets for connections to ports 80 (HTTP) and 443 (HTTPS). Although our evaluation is based on Akamai server logs, we argue that our techniques are not limited to CDN providers and also apply to other types of servers. The major contributions of this work are as follows:

- We perform the first large scale measurement study to passively detect the presence of CTPs deployed in cellular networks worldwide. Our study is based on data collected by Akamai CDN servers during January-July 2015. Our current dataset contains performance metrics from over a total of 14 million TCP connections from clients in different cellular networks.
- We propose three techniques for server operators to passively detect the presence of CTPs from TCP and HTTP server logs. Results from our measurements indicate that the use of CTPs is very popular among cellular carriers worldwide. In fact, carriers employ CTPs for splitting HTTPS sessions, in addition to splitting HTTP sessions.
- Using the collected data, we demonstrate that our techniques are reliable in detecting CTPs deployed in cellular networks across several countries. In Table 1, we compare the results of our passive techniques with the `Delayed Handshake (DH)` active measurement technique of CTP detection for cellular carriers in the US [17]. The tickmarks in the table indicate the presence of CTPs. We show that despite the fact that our passive measurement techniques do not generate probing traffic, they correctly detect CTPs as detected by active experiments in `DH` [17].

Carrier	**Latency**	**Packet Loss**	**TCP SYN**	**DH [16]**
AT&T	✓	✓	✓	✓
Verizon W.	✓	✓	✓	✓
Sprint	✓	✓	✓	✓
T-Mobile	✓	✓	✓	✓

Table 1: Comparison of results from our passive techniques with previous work [17] that uses active experiments, for cellular networks in the US.

The rest of the paper is organized as follows. In Section 2, we discuss related work on detecting cellular middleboxes. In Section 3, we present our methodology. In Section 4, 5, and 6, we discuss how server operators could detect CTPs by using latency estimated by clients and servers, packet loss observed on the server-side, and inspecting `TCP SYN` packets, respectively. In Section 7, we offer discussion of our results. Finally, we conclude in Section 8.

2 Related Work

Several studies have investigated the characteristics, performance benefits and deployment locations of CTPs in cellular networks. Weaver *et al.* and Xu *et al.* investigated the characteristics of transparent Web proxies in cellular networks using active experiments on mobile devices [16][17]. Other studies looked at the performance benefits of TCP splitting proxies to improve Web communications in cellular networks [6][9][13]. Ehsan *et al.* measured the performance gains of CTPs for Web caching and packet loss mitigation in satellite networks [8]. A study by Wang *et al.* characterized implications of cellular middleboxes on improving network security, device power consumption and application performance [15]. Our work, in contrast to these studies, focuses on detecting CTPs using passive measurement techniques, instead of active experiments.

3 Data Collection Methodology

To verify that our latency-based technique reliably detects CTPs in cellular networks worldwide, we used the webpage timing data collected by Akamai's Real User Monitoring system (RUM) [3], which leverages the Navigation Timing API on the client browser [1]. The data includes the time to establish TCP connections for both HTTP and HTTPS sessions. Akamai's RUM also records TCP latency estimated by CDN servers for HTTP and HTTPS session. To investigate whether our packet loss-based technique reliably detects CTPs, we used TCP logs recorded by CDN servers deployed worldwide and extracted the number of packets retransmitted by the server for both HTTP and HTTPS sessions. Finally, to investigate whether our `TCP SYN`-based technique detect CTPs, we collected TCP-dumps on CDN servers for several hours and captured `SYN` packets for connection requests to port 80 (HTTP) and 443 (HTTPS).

4 Detecting CTPs from Client and Server-side Latency

When a CTP splits an end-to-end connection between clients and CDN servers, the latency estimated by clients should be higher than latency estimated by CDN servers. This is because the latency observed by the client will include the radio and cellular backbone latency (∼tens of milliseconds [2]). Whereas the latency estimated by CDN servers would include the latency on the wired public Internet and is likely to be low (∼5 ms), as CDNs have wide deployment of servers inside many cellular networks.

In this section we analyze the TCP latency estimated by clients and servers for TCP connections (both HTTP and HTTPS sessions) using two different methods. First, we compare the latency from both client and server endpoints to identify networks where the latency experienced by clients is significantly

CC	Carrier	Protocol	Hits	Client RTT			Server RTT			Proxy?
				p25	p50	p75	p25	p50	p75	
US	AT&T	HTTP	1.7M	37	47	67	3	4	8	✓
US	AT&T	HTTPS	686K	45	60	89	52	75	114	X
US	Verizon W.	HTTP	1.9M	36	45	69	5	10	21	✓
US	Verizon W.	HTTPS	471K	44	60	87	48	65	87	X
US	T-Mobile	HTTP	2.1M	40	59	85	19	68	157	Limited
US	T-Mobile	HTTPS	459K	45	65	98	59	94	180	–
US	Sprint	HTTP	1.4M	39	52	78	3	12	28	✓
US	Sprint	HTTPS	275K	47	63	93	52	72	118	X
US	Clearwire	HTTP	96K	75	93	128	75	95	139	X
US	Clearwire	HTTPS	39K	75	92	137	82	100	143	X
CA	Bell Canada	HTTP	63K	38	50	69	49	78	151	–
CA	Bell Canada	HTTPS	17K	38	49	73	57	85	157	–
CA	Rogers	HTTP	97K	37	51	86	41	64	110	–
CA	Rogers	HTTPS	30K	37	52	87	48	72	119	–
CA	Telus	HTTP	65K	34	43	60	9	19	49	✓
CA	Telus	HTTPS	16K	43	58	83	47	66	104	X
CA	Sasktel	HTTP	10K	27	41	83	23	33	75	X
CA	Sasktel	HTTPS	2K	43	63	116	59	100	230	–
CA	Videotron	HTTP	7K	44	55	71	44	58	91	X
CA	Videotron	HTTPS	4K	46	58	86	50	70	120	X
MX	Uninet	HTTP	41 K	83	113	183	142	267	571	–
MX	Uninet	HTTPS	8 K	79	109	177	163	256	446	–

Table 2: Distribution of TCP latency estimated by clients (Client RTT) and servers (Server RTT) for IPv4-based cellular networks in North America.

higher than latency experienced by servers – which indicates that a CTP is being used for a connection. Second, we compare the latency for HTTP and HTTPS sessions only from the server-side to identify networks where servers experience significantly different latencies for HTTP and HTTPS sessions – which indicates that a CTP is used for one type of connections.

In Table 2, we show the distribution (25th, 50th, and 75th percentile) of network latency measured by the client (**Client RTT**) and by the server (**Server RTT**) for major cellular networks in North America. The column **CC** represents the country code of each network. Column **Hits** represents the number of unique TCP connections behind latency distributions. The column **Proxy?** indicates whether our techniques detect CTPs for a given cellular carrier. For example, for AT&T network in the US, the **Client RTT** for HTTP sessions is almost 10 times the **Server RTT**, which indicates that servers are communicating with a device only 4 ms away. Since 4 ms is too low for an end-to-end connection over a cellular network [2], we argue that servers communicate with CTPs deployed in AT&T network (as indicated by ✓ in the Proxy column). In the case of HTTPS sessions in AT&T, we observe that **Client RTT** and **Server RTT** are similar, which indicates that there is no CTP for HTTPS sessions in the AT&T network (as indicated by X in Proxy column). Further, when we look at only the **Server RTT** for HTTP and HTTPS sessions, we see that servers experience significantly higher latency for HTTPS sessions, which further confirms that AT&T does not employ CTPs for splitting HTTPS sessions. Ta-

CC	Carrier	Protocol	Hits	Client RTT			Server RTT			Proxy?
				p25	p50	p75	p25	p50	p75	
CN	China Mobile	HTTP	85 K	34	61	101	46	77	128	X
CN	China Mobile	HTTPS	24 K	49	81	132	57	93	170	X
TW	HiNet	HTTP	53 K	33	48	70	35	50	91	X
TW	HiNet	HTTPS	18 K	33	48	77	38	58	103	X
CN	ChinaNet	HTTP	4 K	45	81	149	33	83	167	X
CN	ChinaNet	HTTPS	5 K	207	342	471	118	144	215	–
CN	China Unicom	HTTP	8 K	55	90	150	70	119	209	X
CN	China Unicom	HTTPS	4 K	70	109	187	82	127	213	X
HK	China Mobile	HTTP	9 K	32	53	93	34	60	110	X
HK	China Mobile	HTTPS	3 K	32	48	91	39	57	108	X
IN	Vodafone	HTTP	304 K	58	128	367	33	59	170	–
IN	Vodafone	HTTPS	191 K	80	131	349	102	244	553	–
KR	Korea Telecom	HTTP	28 K	29	35	43	30	40	51	X
KR	Korea Telecom	HTTPS	25 K	30	38	56	37	43	65	X
JP	SoftBank	HTTP	44 K	30	40	55	3	8	13	✓
JP	SoftBank	HTTPS	8 K	37	47	64	41	49	62	X
MY	TM Net	HTTP	13 K	57	75	120	65	113	397	X
MY	TM Net	HTTPS	3 K	60	82	129	83	136	380	X
AE	Eitc	HTTP	4 K	123	153	217	139	159	221	X
AE	Eitc	HTTPS	3 K	139	159	233	140	161	228	X
AE	Etisalat	HTTP	4 K	30	37	49	3	5	29	✓
AE	Etisalat	HTTPS	3 K	33	40	52	35	42	57	X

Table 3: Distribution of TCP latency estimated by clients (Client RTT) and servers (Server RTT) for cellular networks in Asia.

bles 3, 4, and 5 show the application of the latency technique to detect CTPs in cellular networks in Asia, Europe, and Oceania and South America, respectively.

While employing our latency-based techniques to detect CTPs in cellular networks worldwide, we made five observations on the behavior of CTPs. First, we observe that for **p25** of HTTP sessions in T-Mobile USA network, the latency experienced by clients and servers is significantly different, which indicates a presence of CTPs HTTP sessions in T-Mobile network. However, for **p50** of the HTTP sessions, the two latencies are similar – indicating no presence of CTPs for HTTP sessions in T-Mobile network. To investigate this surprising behavior of T-Mobile network, we classified our data based on server locations and domain names. Table 6 shows the distribution **Client RTT** and **Server RTT** for HTTP sessions for different domain names across different locations in the US. We observe that for clients connecting to servers in CA and VA, CTPs are used on per domain basis. For example, the HTTP latency estimated by servers in CA to download webpages associated with a clothing website is significantly lower than latency estimated for a ticketing website. We see similar trends at other locations in the US and across several domain names. Next, we observe that T-Mobile employs CTPs for HTTP sessions only at a few locations in the US. For example, in Table 6 the latency experienced by clients connecting to servers in TX indicate that T-Mobile does not use a CTP for terminating HTTP sessions for any domain name. Thus we argue that T-Mobile's deployment of CTPs in the US is different across different locations and domain names. Based on these observations, we label the **Proxy?** column in Table 2 as 'Limited'.

CC	Carrier	Protocol	Hits	Client RTT			Server RTT			Proxy?
				p25	p50	p75	p25	p50	p75	
DE	DTAG	HTTP	22K	39	50	75	5	8	14	✓
DE	DTAG	HTTPS	13K	53	79	125	34	46	93	–
DE	Vodafone	HTTP	57K	39	51	82	7	11	16	✓
DE	Vodafone	HTTPS	17K	49	64	100	53	70	128.5	X
ES	Telefonica	HTTP	65K	55	92	372	10	18	30	✓
ES	Telefonica	HTTPS	136K	108	149	218	14	22	35	Limited
ES	UNI2	HTTP	43K	41	57	96	38	62	141	X
ES	UNI2	HTTPS	121K	43	59	102	45	64	115	X
ES	Vodafone	HTTP	91K	30	43	72	6	15	30	✓
ES	Vodafone	HTTPS	223K	35	49	76	39	55	90	X
ES	Jazztel	HTTP	9 K	56	75	127	61	90	233	X
ES	Jazztel	HTTPS	17 K	56	73	109	66	87	147	X
FR	Bouygues	HTTP	75K	28	37	57	2	4	38	✓
FR	Bouygues	HTTPS	26K	30	39	59	35	47	79	X
FR	France Telecom	HTTP	37K	37	48	73	1	6	13	✓
FR	France Telecom	HTTPS	17K	40	56	94	1	7	39	✓
FR	SFR	HTTP	41K	37	50	82	3	7	33	✓
FR	SFR	HTTPS	15K	44	62	103	48	72	142	X
FR	Free	HTTP	23 K	43	59	92	40	59	90	X
FR	Free	HTTPS	10 K	45	63	116	26	42	71	–
GB	Telefonica	HTTP	186K	49	71	109	7	11	23	✓
GB	Telefonica	HTTPS	40K	59	85	150	48	72	115	X
GB	Vodafone	HTTP	115K	41	56	89	7	14	57	✓
GB	Vodafone	HTTPS	24K	49	68	111	54	76	145	X
IT	H3G	HTTP	49K	55	73	116	60	81	157	X
IT	H3G	HTTPS	14K	55	77	142	65	93	221	X
IT	Tim	HTTP	55K	39	57	94	6	12	41	✓
IT	Tim	HTTPS	13K	46	67	110	53	80	167	X
AT	France Telecom	HTTP	8 K	41	57	80	59	97	210	–
AT	France Telecom	HTTPS	3 K	43	59	87	66	101	219	–
AT	H3G	HTTP	9 K	40	57	79	58	94	205	–
AT	H3G	HTTPS	4 K	41	59	88	62	98	225	–
AT	T-Mobile	HTTP	10 K	33	48	72	5	15	48	✓
AT	T-Mobile	HTTPS	3 K	40	58	83	52	76	131	X
NL	Vodafone	HTTP	8 K	33	39	61	2	2	16	✓
NL	Vodafone	HTTPS	2 K	35	43	80	37	46	71	X
SE	Vodafone	HTTP	8 K	37	45	59	62	97	175	–
SE	Vodafone	HTTPS	39 K	37	46	58	65	89	142	–
TR	Turk Telecom	HTTP	34 K	54	83	150	39	80	143	X
TR	Turk Telecom	HTTPS	9 K	49	72	138	50	80	145	X
TR	Vodafone	HTTP	16 K	40	59	116	9	51	85	Limited
TR	Vodafone	HTTPS	4 K	55	92	128	64	102	152	X

Table 4: Distribution of TCP latency estimated by clients (Client RTT) and servers (Server RTT) for cellular networks in the Europe.

CC	Carrier	Protocol	Hits	Client RTT			Server RTT			Proxy?
				p25	p50	p75	p25	p50	p75	
AU	Vodafone	HTTP	106 K	31	40	62	2	3	13	✓
AU	Vodafone	HTTPS	64 K	38	51	94	36	48	87	X
NZ	Vodafone	HTTP	7 K	30	49	71	2	11	27	✓
NZ	Vodafone	HTTPS	6 K	38	59	99	37	61	115	X
BR	Telefonica	HTTP	560 K	51	108	273	58	120	309	X
BR	Telefonica	HTTPS	63 K	40	78	165	51	100	212	X
PY	Telefonica	HTTP	13 K	180	217	289	186	237	430	X
PY	Telefonica	HTTPS	3 K	184	221	297	202	262	428	X

Table 5: Distribution of TCP latency estimated by clients (Client RTT) and servers (Server RTT) for cellular networks in Oceania and South America.

State	Domain Type	Client RTT			Server RTT			Proxy?
		p25	p50	p75	p25	p50	p75	
CA	Clothing website	37	51	75	2	3	3	✓
CA	e-Commerce website	40	56	80	2	2	3	✓
CA	Health Care website	40	56	90	40	80	175	X
CA	Ticketing website	37	49	65	43	93	186	X
VA	Clothing website	39	57	80	2	2	2	✓
VA	e-Commerce website	46	68	89	2	2	2	✓
VA	Health Care website	44	64	90	27	63	121	X
TX	Clothing website	54	72	96	49	93	204	X
TX	Health Care website	56	75	97	61	107	211	X
TX	Ticketing website	50	70	90	33	67	111	X
TX	Movies website	56	71	91	88	156	301	X

Table 6: Distribution of HTTP latency estimated by clients (Client RTT) and servers (Server RTT) for T-Mobile across different domains & locations.

The second observation we make is that cellular networks in the US use CTPs for TCP connections over their IPv4 networks, but not over their IPv6 networks. Since we did not observe statistically significant IPv6 traffic from cellular carriers deployed outside of the US, we restrict this observation to cellular carriers in the US only. In Table 7, we show the distribution of TCP latency for IPv6 networks deployed by major US carriers, estimated by clients and CDN servers. We observe that clients in Verizon Wireless connecting to CDNs over IPv6 network experience latency similar to that estimated at the server for HTTP sessions. However, from Table 2, we observe that Verizon clients connecting to CDN servers over its IPv4 network experience much higher latency than experienced by the CDN servers, for HTTP sessions – indicating the presence of CTP for HTTP sessions in its IPv4 network. Therefore, we argue that Verizon employs CTPs for HTTP sessions in its IPv4 network and not in its IPv6 network.

The third observation we make is that some networks use CTPs to split HTTPS sessions. Using our measurement data, we identified a cellular carrier in France that employs CTPs to split HTTPS sessions. In Table 4, we show that for France Telecom, the **Server RTT** for HTTPS sessions is significantly lower than the **Client RTT**, therefore we believe that France Telecom uses CTPs to split HTTPS sessions. Telefonica in Spain is another cellular carrier

CC	Carrier	Protocol	Hits	Client RTT			Server RTT			Proxy?
				p25	p50	p75	p25	p50	p75	
US	AT&T	HTTP	15 K	37	45	60	2	3	16	✓
US	AT&T	HTTPS	4 K	43	58	87	47	62	93	X
US	Verizon W.	HTTP	232 K	46	62	84	43	66	83	X
US	Verizon W.	HTTPS	81 K	46	62	87	50	69	90	X
US	T-Mobile	HTTP	295 K	42	60	85	4	24	59	Limited
US	T-Mobile	HTTPS	82 K	47	68	96	49	67	99	X

Table 7: Distribution of TCP latency estimated by clients (Client RTT) and servers (Server RTT) for IPv6 cellular networks in North America.

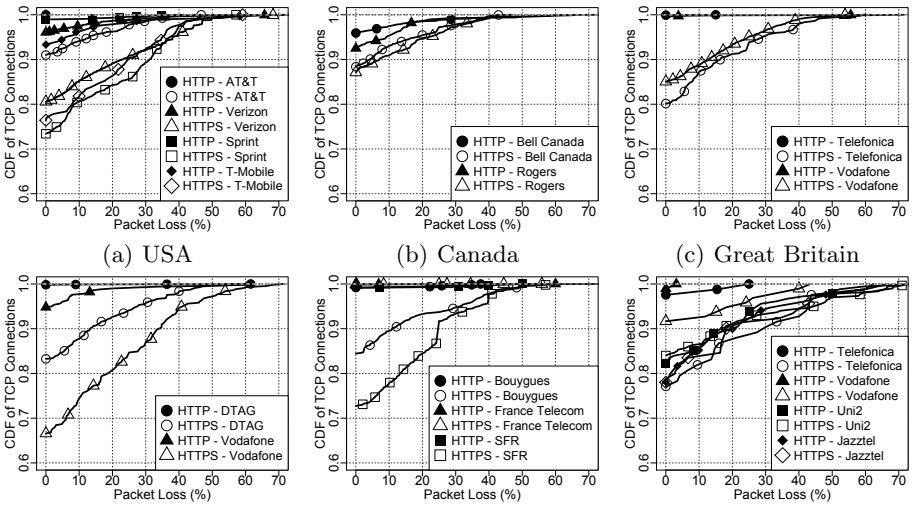
for which we observe that CTPs split HTTPS sessions, as the latency estimated by CDN servers is lower than latency estimated by clients. Further, Telefonica's recent design of `mcTLS` protocol indicates that ISPs work towards deploying CTPs for HTTPS sessions [12], likely to support content caching and connection optimization for secure connections [14].

The fourth observation we make is that for some carriers, the **p75** of **Server RTT** is similar to **p25** of **Client RTT**, when the **p25** and **p50** of **Server RTT** indicate the presence of CTPs in that carrier. For example, the **p75** of **Server RTT** for HTTP sessions in Etisalat network in Table 3, suggests that CTPs may not be used for splitting all HTTP sessions. We speculate that when CTPs get overloaded, client requests are likely not sent to CTPs and instead sent directly to servers. As a result servers occasionally experience (unproxied) latency of end-to-end connections to mobile devices. To deal with such occasional instances, TCP stacks of servers should interpret such connections as direct connections to mobile devices.

Finally, the fifth observation we make is that for a few cellular carriers the **Server RTT** is either higher or lower than **Client RTT** by at least 80 ms for **p75**. Specifically, if we observe **Server RTT** to be higher than **Client RTT**, we speculate that CTPs are deployed near the gateway and Internet egress points are far from the gateway. If we observe **Server RTT** to be lower than **Client RTT**, we speculate that CTPs are near to both egress points and gateways but clients connect to gateways far in the network. For such cellular carriers we place a **'-'** in the **Proxy?** column in Tables 2, 3, 4, and 5. We argue that for such cellular carriers, passive techniques in the following sections may be used to detect the presence of CTPs.

5 Detecting CTPs from Packet Loss on the Server-side

In previous section, we discussed how server operators could use latencies measurements by clients and servers to detect the presence of CTPs. In this section, we are interested in verifying another technique, based on packet loss, to passively detect CTPs across cellular networks worldwide using measurement data collected by Akamai CDN servers. Since we observe TCP latency estimated by CDN servers to CTPs is significantly low, we argue that CTPs and CDN servers are usually deployed within the same or nearby datacenters. Therefore, when a CTP is employed to split connections, the number of packets retransmitted by servers should be lower than packets retransmitted for connections where CTPs



(a) USA (b) Canada (c) Great Britain

(d) Germany (e) France (f) Spain

Fig. 1: Distribution of packet loss over HTTP and HTTPS sessions for cellular networks in different countries. For visibility, we reduced the number of symbols on each line.

are not used. Following this assumption, in Figure 1, we show the distribution of packet loss observed during our tests for thousands of HTTP and HTTPS sessions. Our first goal is to identify networks where packet loss observed by CDN servers is higher for one type of connections and not others. We also aim to determine whether results from using packet loss correlate with our CTP detection in the previous section. Due to space limitations, we show distribution of packet loss for only a few cellular carriers in North America and Europe.

In Figure 1(a), we show the distribution of packet loss observed for HTTP and HTTPS sessions in four major cellular carriers in the US. Specifically, in the case of Verizon, AT&T, and Sprint networks, we observe that for HTTP sessions CDN servers experience low packet loss, whereas for HTTPS sessions CDN servers experience significantly higher packet loss – indicating the presence of CTPs for HTTP sessions. The results for these networks agree with our observations from using latency-based technique. However, in the case of T-Mobile, we see that the packet loss for HTTP sessions is slightly higher compared to other networks. We speculate that the packet loss for HTTP sessions in T-Mobile network are influenced by T-Mobile's policy to employ CTPs at only a few locations and domain names in the US (Table 6).

Next, we compare the packet loss observed for connections in a network where CTP is not employed, the Rogers network in Canada, as detected by our latency-based technique in Table 2, with a network where our latency-based technique could not detect the presence of CTPs, the Bell Canada network in Canada. In Figure 1(b), we show that for both HTTP and HTTPS sessions in Bell Canada and Rogers networks, CDN servers observe similar packet loss. We speculate that

either CTPs are not employed in the Bell Canada network or CTPs are present but CTPs experience same network conditions as Rogers network without CTPs.

We now extend our discussion and compare packet loss observed by CDN servers for connections in major cellular carriers in the UK, Germany, France, and Spain. Similarly to carriers in the US, in Figure 1(c) and 1(d), we show that packet loss observed by servers for HTTP sessions is significantly lower than packet loss observed for HTTPS sessions – indicating the presence of CTPs for HTTP sessions, similar to our observations from using latency-based technique. For cellular carriers in France in Figure 1(e), we observe that packet loss for HTTPS sessions in France Telecom network is similar to packet loss for HTTP sessions, with both being almost zero. This indicates that CTPs are employed by France Telecom for splitting both HTTP and HTTPS sessions – validating our observations from using latency-based technique.

Finally, in Figure 1(f), we show distribution of packet loss observed by CDN servers for major cellular carriers in Spain. We observe that for Vodafone and Telefonica networks, the packet loss for HTTP sessions is much lower than packet loss for HTTPS session – indicating the presence of CTPs for only HTTP connections, similar to our observations from using latency-based technique. For Uni2 and Jazztel, however, we observe that packet loss for both HTTP and HTTPS is similar. This indicates that CTPs are used for both HTTP and HTTPS sessions, similar to our observations from using latency-based technique. One exception to our results is for Telefonica. Using the latency technique we identified that Telefonica could be a potential carrier where CTPs are used to terminate HTTPS sessions. However, the high packet loss for HTTPS sessions indicates that CTPs are not used for splitting HTTPS sessions. To disambiguate the presence of CTPs, we propose another technique that relies on analyzing the characteristics of TCP SYN packets, which we discuss next.

6 Detecting CTPs from TCP SYN Characteristics

Our third technique is based on analyzing `TCP SYN` packets to detect the presence of CTPs in cellular networks. Our active experiments on understanding characteristics of `TCP SYN` packets generated by different types of mobile devices have revealed that the advertised `Initial Congestion Window Size (ICWS)`, `TCP Timestamp` in the TCP options header, and `Maximum Segment Size (MSS)` values are different across different types of mobile devices. We also observed that these values are different even when the same device connects to Wi-Fi and cellular network. Based on this observation, our goal is to identify whether analyzing `TCP SYN` packets (captured passively for HTTP and HTTPS sessions) have the same `ICWS`, `MSS`, and an increasing TCP Timestamp value, which would indicate that `SYN` packets are likely being generated by a single machine (a CTP), instead of from multiple mobile devices with different hardware.

Results from our analysis of `TCP SYN` packets indicate that for all observed `TCP SYN` packets on port 80 from cellular carriers for which our latency and packet loss-based techniques suggest presence of CTP for HTTP sessions, the `ICWS` and `MSS` fields in the `TCP SYN` packets have the same value and the TCP Timestamp option have monotonically increasing values with a near constant

skew – indicating the presence of CTPs for splitting HTTP sessions. For `TCP SYN` packets (generated from networks for which our latency and packet loss-based techniques suggest absence of CTPs for HTTPS sessions) to port 443 of CDN servers, we observed varying values of `ICWS`, `MSS`, and TCP Timestamp – indicating that the `TCP SYN` packets are likely generated by different mobile devices, instead of CTPs. We also verified our technique to be reliable for cellular carriers that employ CTPs for HTTPS sessions. For example, for France Telecom network in France we observed that the characteristics of all observed `TCP SYN` packets to port 443 were similar – indicating the presence of CTPs for HTTPS connections. For Telefonica in Spain, we did not observe similar characteristics of observed `TCP SYN` packets to port 443 – indicating absence of CTPs for splitting HTTPS sessions. Based on our findings on Telefonica's CTPs for HTTPS sessions from our latency, loss, and SYN-based techniques, we argue that active measurements may be needed to reliably detect CTPs. Finally, based on the data collected we did not find networks where `ICWS` and `MSS` values were similar but CTP was not detected using latency packet loss based techniques.

7 Discussion

We believe that one can leverage the use of our latency-based technique to identify the cellular latency offered by carriers where CTPs are present. We argue that for such carriers, **Client RTT** is a reliable indicator of the cellular latency, comprising of the sum of radio latency and latency within the cellular backbone. Specifically, if 4G is widely deployed by a cellular carrier, the latency offered by 4G would be reflected in both **p25** and **p75** of **Client RTT**. Further, if 3G is more widely deployed than 4G, then the latency offered by 4G would be reflected in the **p25** and latency offered by 3G would be reflected in **p75** of **Client RTT**. For example, for Telefonica in Spain, Sensorly's [4] signal strength data suggests a wide deployment of 3G, but little deployment of 4G. Therefore, in Table 4, the **p25** of **Client RTT** for HTTP sessions (55 ms) reflects Telefonica's latency over its 4G network, whereas the **p75** latency of 372 ms reflects its 3G latency. Further, the Etisalat network in AE (in Table 3) has wide deployment of 4G (based on Sensorly data), thus the HTTP latency shown in both **p25** (30 ms) and **p75** (49 ms) of **Client RTT** represents the latency offered by Etisalat's 4G network. For other cellular networks with CTPs also, we verified that using Sensorly's data and **Client RTT** together allows cellular latency estimation in a given carrier.

8 Conclusions

Connection Terminating Proxies (CTPs) have been a great area of interest for many cellular carriers in the past. These proxies allow for optimizing TCP connections between servers and client devices. In this paper, we propose three techniques to passively identify the presence of CTPs, based on latency, loss, and `TCP SYN` characteristics. We also conduct an extensive measurement study based on Akamai server logs to demonstrate that our techniques can reliably detect CTPs in cellular networks worldwide. Based on our measurement results, we argue that server operators could use our suggested techniques to detect

CTPs using server logs only and optimize communications for different cellular networks with the goal of faster content delivery to end-users.

Acknowledgments

We thank Ruomei Gao, Chris Heller, Ajay Kumar Miyyapuram, and Kanika Shah for their invaluable insights on refining our data collection process. We also thank National Science Foundation for supporting this work through grant NSF CNS-1555591.

References

1. Navigation Timing. `http://w3c.github.io/navigation-timing/`, Aug. 2015.
2. NSF Workshop on Achieving Ultra-Low Latencies in Wireless Networks. `http://inlab.lab.asu.edu/nsf/files/WorkshopReport.pdf`, Mar. 2015.
3. Real User Monitoring. `https://www.akamai.com/us/en/resources/real-user-monitoring.jsp`, Aug. 2015.
4. Unbiased Wireless Network Information. `http://www.sensorly.com`, Aug. 2015.
5. J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. `https://tools.ietf.org/html/rfc3135`, Jun. 2001.
6. A. Botta and A. Pescape. Monitoring and measuring wireless network performance in the presence of middleboxes. In *Conference on Wireless On-Demand Network Systems and Services*, Jan. 2012.
7. N. Dukkipati, T. Refice, Y. Cheng, J. Chu, T. Herbert, A. Agarwal, A. Jain, and N. Sutin. An Argument for Increasing TCP's Initial Congestion Window. *SIGCOMM CCR*, 40(3), Jun. 2010.
8. N. Ehsan, M. Liu, and R. J. Ragland. Evaluation of Performance Enhancing Proxies in Internet over Satellite, Jan. 2003.
9. V. Farkas, B. Hder, and S. Novczki. A Split Connection TCP Proxy in LTE Networks. In *Information and Communication Technologies*, Aug. 2012.
10. C. Gomez, M. Catalan, D. Viamonte, J. Paradells, and A. Calveras. Web browsing optimization over 2.5G and 3G: end-to-end mechanisms vs. usage of performance enhancing proxies. *Wireless Communications and Mobile Computing*, Feb. 2008.
11. M. Ivanovich, P. Bickerdike, and J. Li. On TCP performance enhancing proxies in a wireless environment. *IEEE Communications Magazine*, Sept. 2008.
12. D. Naylor, K. Schomp, M. Varvello, I. Leontiadis, J. Blackburn, D. Lopez, K. Papagiannaki, P. R. Rodriguez, and P. Steenkiste. Investigating Transparent Web Proxies in Cellular Networks. In *ACM SIGCOMM*, Aug. 2015.
13. M. Necker, M. Scharf, and A. Weber. Performance of Different Proxy Concepts in UMTS Networks. In *Wireless Systems and Mobility in Next Generation Internet*, Jun. 2004.
14. M. Thomson. Blind Proxy Caching. `https://httpworkshop.github.io/workshop/presentations/thomson-cache.pdf`, Jul. 2015.
15. Z. Wang, Z. Qian, Q. Xu, Z. Mao, and M. Zhang. An Untold Story of Middleboxes in Cellular Networks. In *ACM SIGCOMM*, Aug. 2011.
16. N. Weaver, C. Kreibich, M. Dam, and V. Paxson. Here Be Web Proxies. In *Passive and Active Measurements Conference*, Mar. 2014.
17. X. Xu, Y. Jiang, T. Flach, E. Katz-Bassett, D. Choffnes, and R. Govindan. Investigating Transparent Web Proxies in Cellular Networks. In *Passive and Active Measurements Conference*, Mar. 2015.