An Analysis of Changing Enterprise Network Traffic Characteristics

David Murray, Terry Koziniec, Sebastian Zander, Michael Dixon, Polychronis Koutsakis
School of Engineering and Information Technology
Murdoch University
{D.Murray, T.Koziniec, S.Zander, M.Dixon, P.Koutsakis}@murdoch.edu.au

***Abstract*—Studies on the composition and nature of Internet protocols are crucial for continued research and innovation. This study used three different methods to investigate the presence and level of support for various Internet protocols. Internet traffic entering and exiting a university network was passively captured, anonymised and analysed to test protocol usage. Active tests probed the Internet's most popular websites and experiments on the default behaviour of popular client, server and mobile operating systems were performed to reconcile the findings of the passive data collection. These results are valuable to research areas, such as those using emulations and simulations, where realism is dependent on the accuracy of the underlying assumptions about Internet traffic. Prior work is leveraged to explore changes and protocol adoption trends. This study shows that the majority of Internet traffic is now encrypted. There has also been an increase in large UDP frames, which we attribute to the Google QUIC protocol. Support for TCP options such as Selective Acknowledgements (SACK) and Maximum Segment Size (MSS) can now be assumed. Explicit Congestion Notification (ECN) usage is still marginal, yet active measurement shows that many servers will support the protocol if requested. Recent IETF standards such as Multipath TCP and TCP Fast Open have small but measurable levels of adoption.**

Keywords*—*Internet measurement, Internet statistics, TCP, UDP, HTTPS, QUIC, Explicit Congestion Notification (ECN), MSS, SACK, WSOPT, TSOPT, Multipath TCP, TCP Fast Open

I. Introduction & Prior Work

Internet research is frequently conducted with simulators, such as NS2, OPNET, NetSim, and many scientific papers have been proposed and refined using these methods. It is therefore crucial that simulations and experimental emulations are performed using models that reflect the current Internet. The importance of measurement studies has been voiced by key figures within the Internet research community [1].

Internet measurement studies can be categorised as active or passive. Active measurement is performed by probing servers or other Internet hosts. Prior active measurement studies [2], [3] have been able to probe popular Internet servers to approximate the level of support for various protocols. Although active tests are used to quantify IETF standards adoption of IP and TCP options in servers, passive measurement is more likely to describe the actual state or level of protocol support being used across a network today.

Passive studies usually capture data moving across a network where terms of service issues for users, as well as the legal and ethical ramifications of user data [4], must be carefully navigated. Many prior studies exist [5], [6], [7], [8], [2], [3], using a range of active and passive techniques but frequent reassessment is required to track the changing environment. In addition to updating the community on protocol and packet statistics, a number of questions were of particular interest.

- The Internet is currently transitioning from HTTP to HTTPS websites; is the majority of web browsing now encrypted?
- Google's internally developed congestion control mechanism, QUIC, operates over UDP. Has the widespread adoption of Google services, and Android devices running the Chrome web browser, increased the quantity of UDP traffic?
- What are the Maximum Segment Size (MSS) values requested in TCP SYN messages? What is the distribution of sizes?
- Apple has begun mandating Explicit Congestion Notification (ECN) support in recent software development toolkits. Is there a visible increase, in ECN, above the negligible values measured in prior work [2], [3], [5]?
- Multipath TCP [9], standardised in 2013 and TCP Fast open, [10], standardised in 2014, are significantly more recent than ECN. Is there any measurable usage of these protocols?

This paper is structured as follows. Section II describes how the data was collected, in an ethical manner, without compromising user anonymity or privacy. Section III describes the usage of ports, the distribution of frame sizes and the occurrence of TCP options. These results are reconciled against an experiment evaluating the default protocol configuration in a variety of server, client and mobile operating systems. The conclusion highlights the key contributions of the paper and future work.

II. Methodology

A. Packet header capture & anonymisation

To capture traffic, the professional IT communications staff configured a switchport analyser (SPAN) port to mirror traffic entering and exiting the University. As these interfaces operate at 1 Gb/s, no special hardware was required to capture these packets. A rackmount server, running Ubuntu 16.04 and TCPdump, captured and anonymised packets.

Only the first 100 bytes of each frame was captured. Any user payload data in these 100 bytes was zeroed and

IP addresses were anonymised using TCPRewrite [11]. The research staff were only provided access to the anonymised and payload removed packet capture files. Professional IT communications staff performed all tasks where actual IP addresses or fragments of packet payloads were present. The project was approved by the Murdoch University Human Research Ethics Subcommittee (Approval Number: 2015/255).

The capture location is also important in this study. The capture point was near the edge of the network where packets are entering or exiting the network over the Internet. Subsequently, internal LAN traffic and many internal DNS lookups do not feature in the capture. As the capture point is behind the firewall, many network attacks or anomalies may have been dropped and are not measured in this study.

Measurement and analysis of the data was performed with custom C programs on a Linux server. A single 24 hour period, storing only the header data to disk, amounted to approximately 500GB of data. After using bunzip to compress this data, each day amounted to approximately 80GB on disk. The numbers in this paper are presented as percentages or ratios because the absolute numbers are large. Unless stated otherwise, the averages are based on a seven day period in May 2017.

In addition to passive packet header capture and anonymisation, active measurement was also performed to investigate the adoption of Explicit Congestion Notification (ECN). Finally to reconcile the results of the passive and active research, an experiment on the default protocol setting of a range of mobile and computer based operating systems was performed.

B. Dataset context

The dataset from a University Internet connection may differ from an Internet router and discussion of these differences is appropriate. The University network carries a traffic mix from standard managed operating devices and privately administered devices. The dataset does not include student or staff accommodation and is limited to the packet headers of traffic on campus or from users connecting to University resources via the Internet. While a large proportion of these machines may run Windows, the University procures many Mac laptops and desktops. Free internet access, via Eduroam, is offered to students and staff where a range of iOS and Android devices are used. There is also a wired staff network which predominantly connects staff computers. Network access is more permissive than some commercial work environments, with most social media or video streaming tolerated. On the contrary, we expect network usage to be more work oriented than a home broadband environment, where greater use of Netflix or Amazon Prime type streaming services is expected.

III. Results

The data analysis in this paper begins with transport layer ports; an indicator of the traffic profile and the applications used. After investigating the transport layer ports, the size of frames and the Maximum Segment Size (MSS) values requested in TCP SYN messages are explored. The final section investigates and describes the remaining TCP options including, SACK (Selective Acknowledgements), Window Scale (WSOPT), Timestamps (TSOPT), Explicit Congestion Notification (ECN), Multi Path TCP (MPTCP) and TCP Fast Open (TFO).

TABLE I. Top 5 TCP and UDP Ports

TCP Port	Percent	UDP Port	Percent
443 HTTPS	51.3 %	443 QUIC	6.5 %
80 HTTP	22.7 %	4500 Cisco VPN	0.7 %
22 SSH	2.7 %	53 DNS	0.6 %
23 Telnet	2.4 %	50725 Dynamic	0.3 %
993 IMAP	0.8 %	58735 Dynamic	0.3 %

A. Transport and Application Layer Statistics

By investigating transport layer ports, estimates on the types of applications are possible. Table I shows the most frequently used TCP and UDP ports. Note that port 443 is the most used in both TCP and UDP transport layers. Table I is based on the number of packets transmitted and received rather than the number of bytes transferred, however, with only a few exceptions, packets and bytes correlate strongly.

TCP port 433 represents HTTPS web based Internet use, which is increasingly replacing HTTP, port 80. In the Alexa website list [12] eight of the ten most popular websites use HTTPS by default. The two pages lacking HTTPS support are predominantly used in non-English speaking countries. Some traffic that enters the university network would be destined for the main University web page, which is currently unencrypted, but most of the student and staff resources are password protected HTTPS Sites.

A comparison with a 2011 dataset shows that the Internet has switched from HTTP to HTTPS. In 2011 [5] only 5% of web traffic[1] used HTTPS, but in 2017 71% of web traffic used HTTPS. With many browsers now indicating that HTTP based websites are insecure, this trend is expected to continue.

Similar to the TCP results, port 443 is also the most commonly used port over UDP. This traffic is likely to be the Google QUIC protocol, which implements congestion control and encryption over UDP. Currently, QUIC may account for users running the Google Chrome web browser, when visiting Google sites such as YouTube, Google search and Gmail. While some of the UDP 443 traffic could be VPNs, we expect that this would be a small component. Quantifying the QUIC and VPN usage may only be possible through analysis of the source and destination IP addresses. For ethical reasons, this is not feasible because the packet headers stored are anonymised.

The University VPN, mainly recommended for off campus workers, uses UDP port 4500 and accounts for only 0.7% of the traffic. The third highest port is port 53 and represents DNS. The proportion of DNS frames is lower than expected, 0.6% of the total, because the University's DNS server will respond over the LAN and the resulting traffic will not traverse the Internet link where the capture point is located. Peer-to-peer traffic was not detected, and we would suggest that any use is marginal.

Interestingly, since 2011 there has been a slight decrease in the number of UDP messages. It was expected that the use of QUIC, as well as VPNs, would have increased the

[1]Our working definition of web traffic is HTTP and HTTPS

TABLE II. USAGE OF TRANSPORT LAYER PROTOCOLS

Protocol	Percent in 2011	Percent in 2017
TCP	84.35% packets	87.56% packets
UDP	13.92% packets	11.72% packets
TCP	92.00% bytes	89.55% bytes
UDP	08.00% bytes	09.91% bytes

TABLE III. USAGE OF TCP OPTIONS

TCP Option	Percent in 2011	Percent in 2017
MSS	96.59%	99.99%
SACK	94.00%	99.26%
WSOPT	63.97%	88.48%
TSOPT	39.29%	41.83%
ECN	00.00%	03.92%
Fast Open	Not Measured	00.11%
Multi Path	Not Measured	00.07%

proportion of UDP frames. Although the percentage of UDP frames decreased in 2017, the percentage of bytes carried by UDP transport increased. In 2011 UDP transferred 7.5% of the bytes which increased to 10% in 2017.

B. Packet and Frame Sizes

The distribution of packet sizes in 2011 and 2017 are shown in Fig 1 and 2 respectively. Since 2011 Internet packet sizes have become increasingly bimodal [7], [5]. An even greater proportion of frames now exist in the less than 100 byte and greater than 1400 byte categories. The proportion of UDP frames sized between 1300 and 1400 bytes in 2017 has also increased [7], [5] as a result of QUIC.

C. TCP Options

This section reviews the adoption and use of TCP options. In the case of well-entrenched TCP options, such as; SACK (Selective Acknowledgments), Maximum Segment Size (MSS), Window Scale (WSOPT) and Timestamps (TSOPT) the results are compared with historical data. Emerging options, such as: Explicit Congestion Notification (ECN), Multi Path TCP (MPTCP) and TCP Fast Open (TFO) are discussed with particular emphasis on the adoption challenges. A summary of the data on TCP options, found in our passive measurement study is, shown in Table III below.

1) MSS: The Maximum Segment Size (MSS) TCP option is used by end nodes to state the maximum supported packet size. Unless an MSS is specified in the TCP SYN, the 536 byte MSS value as defined in RFC 1122 is used. A prior study in 2011 found that the MSS was requested in 96.6% TCP SYNs [5]. This study found that 99.99% of TCP SYNs specified an MSS.

Table IV, compares the MSS values observed in this dataset and with prior research from 2011 [5]. The results show that over 90% of connections specify a value between 1301 and 1460, with 30% specifically requesting 1460. As an overall trend, the requests for MSS values between 1001 and 1300 bytes shrank and requests for MSS values between 1301 and 1460 bytes grew. These results confirm the histograms, illustrating the distribution of frame sizes, in Fig 1 and 2.

2) SACK: TCP Selective Acknowledgements (SACK) are a modification of the cumulative ack behaviour of TCP. The ack field, within the TCP header, holds the sequence number

TABLE IV. MSS OPTION VALUES

MSS Size	Percent in 2011	Percent in 2017
0-1000	00.04%	00.33%
1001-1300	23.69%	07.93%
1301-1459	46.27%	60.10%
1460	26.54%	31.59%
1461-1600	00.02%	00.02%
1601-8000	00.01%	00.01%
8001-9000	00.00%	00.02%
$>$ 9000	00.01%	00.01%

of all the data successfully received. Under this scheme, a burst of packet losses will take multiple round trip times to recover, as only a single packet loss can be communicated. SACK is a TCP option that can specify the blocks which have been successfully received and therefore may signal multiple packet losses within a single acknowledgement. As a result, SACK enables TCP to more efficiently and quickly recover from multiple packet losses within the same window.

Wolfgang's 2006 [7] study found that 91% of all TCP SYN segments specified SACK capability. Later work in 2011 [5] found that 94% of SYNs used the SACK permitted TCP option. The results of this study show that now 99.26% of TCP SYN messages specify SACK capability.

3) Window Scale: The *Window Size*, or WSOPT, field in the TCP header is only 16 bits long and limits the advertised size of a TCP window to 65,535 bytes. The *Window Scale* TCP option (WSOPT), defined in RFC 1323, operates as a multiplier on the *window size* and is required for fast transfers over high latency networks. Research in 2011 found that the Window Scale option was present in 63.9% of TCP SYN messages [5]. These results suggest that support for WSOPT has grown to 88.48%.

4) Timestamps: Timestamps (TSOPT) were defined alongside the Window Scale option in RFC 1323. Timestamps are used to calculate the RTT and are used for Protection Against Wrapped Sequence (PAWS). The 32 bit TCP sequence number may only represent up to 4 GB of data, and therefore ambiguity, over which segment an ack refers to, must be clarified. PAWS determines to which 4GB sequence a replayed packet belongs. Prior research in 2007 found 14.5% [7] of SYNs advertised the Timestamps option and in 2011, this had grown to 39.2% [5]. In 2017 TSOPT only marginally increased to 41.8%. The reasons for the marginal increase in usage of TSOPT is further explained in a followup experiment on the default options of a variety of operating systems found in section III-F.

5) ECN: Since the design of TCP in the 1970's, detecting congestion between endpoints has been implied through packet loss. Any packet loss is detected and then signalled through duplicate acknowledgements, which travel in the reverse direction to the data. Signalling congestion using duplicate acknowledgements, remains the primary mechanism, but often occurs too late and when buffers have already grown too large. ECN provides a mechanism for routers to inform TCP senders of congestion before the need for packet loss occurs[13].

Floyed et al.'s initial work on ECN [13] was implemented into Linux kernel version 2.4.20; released in November 2002. A 2004 study on TCP options by Pentikousis and Badr [6] found that ECN deployment, in servers, was marginal

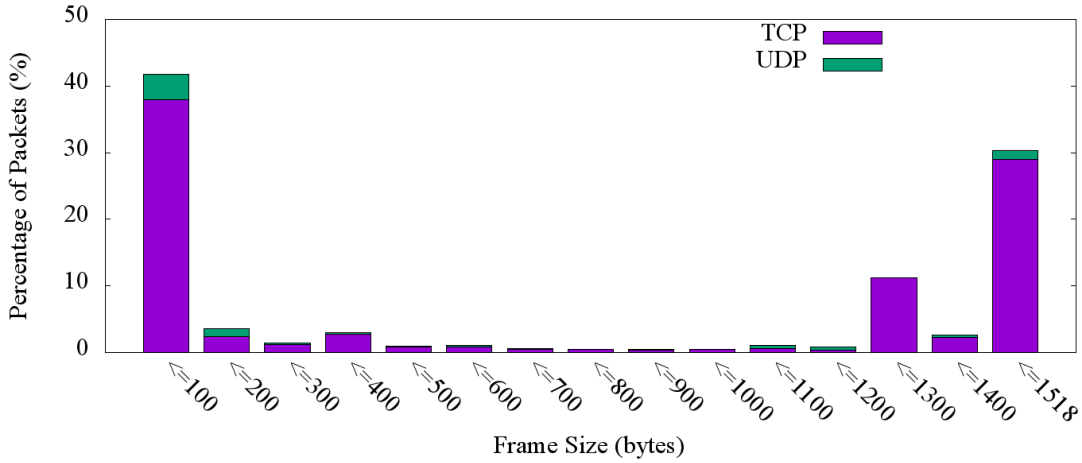

Fig. 1. Distribution of UDP and TCP packet sizes in 2011

Fig. 2. Distribution of UDP and TCP packet sizes in 2017

(0.15%). Furthermore, the proportion of traffic marked with CE (Congestion Experienced) was close to zero [6]. An active measurement study, by Medina et al. in 2005, found that 2.1% of web servers were ECN capable [2]. Thus, even when requested, only 2.1% of servers could use ECN to signal congestion. The analysis of some passive header data in 2012 [5], found that the use of ECN markings in headers was non-existent.

One of the problems or impediments to ECN adoption are middleboxes which may drop or mangle packets with ECN marks. The term middlebox refers to firewalls, intrusion detection systems, NATs, virtual private network gateways, load balancers, and WAN accelerators. Honda et al.'s study in 2011 [14] found that 86% of Internet Paths allow traffic using TCP extensions.

The connectivity problems caused by middleboxes resulted in passive ECN implementations. This means that support is present but not used in outgoing requests. In mid 2015 Apple announced that the new iOS dev kits would mandate ECN and IPv6 compliance. In 2016, 5% of all iOS 9.3 and OSX El Capitan v 10.11.5 connections would request ECN [15]. Apple's statistics stated that, in September 2015, 56% of the top 1 million web sites supported ECN. In June 2016, they claimed this number had increased to 83%.

Analysis of the captured data in this paper shows the proportion of flows performing ECN today. The results suggest that only 3.51% of TCP headers requested ECN. The ECN protocol is negotiated at the transport (TCP) layer but signalled at the network (IP) layer. After negotiation, congestion is marked in the IP DSCP header flags. The data suggests that a marginal level, 0.73% of IP packets, had the ECN capable flag set. ECN adoption is therefore still increasing very slowly, but this may rapidly change. Many installations provide passive support; they will only perform ECN if requested by the other client.

To determine the level of passive support, a longitudinal analysis of the Alexa top websites [12] was recorded. A script probed the most visited 25 sites on the Internet using Mozilla Firefox in safe browsing mode. TCPDump saved pcap files and C code analysed ECN support. The level of ECN support was assessed on the proportion SYN-ACK-ECN replies that received. Modern websites involve many TCP transactions with different servers, including advertisers. Therefore support ECN for most websites is not binary. Table V shows the

TABLE V. PASSIVE ECN SUPPORT IN THE ALEXA TOP 25 WEBSITES

Year	Average ECN Support
2012	08.5%
2015	28.4%
2016	54.1%
2017	70.2%

percentage of ECN capable TCP flows captured using these techniques. As these measurements were manually started, there are unfortunately some missing years. While the passive analysis reveals that ECN support is absent, active analysis suggests that many mainstream websites now include ECN support, a finding that is further supported by the substudy on operating systems in section III-F.

D. Multi Path TCP

Multipath TCP (MPTCP) is considerably more recent than ECN, with an IETF draft published as an experimental standard in 2013 [9]. MPTCP allows devices to use multiple network interfaces for a single TCP connection. In a data centre environment, this may allow a server to utilise two or more network adapters to increase throughput. There are also many benefits for mobile devices where WiFi and 3G/4G networks can be employed to provide additional bandwidth and redundancy as a device moves through different networks.

Apple has implemented MPTCP in iOS 7 since 2013 and in Mac OSX 10.10 since 2014. Support has existed in FreeBSD since 2013 [16]. Similar to ECN, MPTCP is negotiated in TCP options in the three-way handshake and requires support on both the server and the client. The Linux implementation, at time of writing, has not yet been integrated into a kernel release. Apple has selectively used MPTCP on all their platforms for Siri. The passive measurement study, described in this paper, measured the proportion of TCP SYN messages with MPTCP capability. The results suggest that only 0.06% of TCP SYN messages are using MPTCP. Given that only Apple clients use Siri, the result is unsurprising but will be important to track in the future.

While MPTCP is very promising from a resource utilisation perspective and may increase the robustness of mobile device Internet connections, it is not without issues. It introduces new challenges for congestion control algorithms that must now assess and fairly share the available resources across multiple paths. MPTCP also is also significantly more complex for end host stacks [17].

E. TCP Fast Open

Web transactions often consist of many small TCP flows. An impediment to lowering the transfer completion time is not bandwidth, but latency, and TCP's three-way handshake exacerbates any delays. TCP Fast Open (TFO) can reduce the delay of opening subsequent flows by one round trip time [18] through the transmission of a cookie in the TCP SYN. TFO became a IETF experimental standard in 2014 RFC 7413 [10].

Support for TFO is required on the server as well as the client OS and browser. Linux has supported TFO since kernel 3.7 and Firefox version 55. Google have provided support in Android, Chrome OS and their web browsers Google Chrome and Chromium. Apple has included support for iOS 9, and OSX 10.11 and Microsoft have support in Windows 10 and Microsoft Edge.

Although support is present in many platforms and browsers, at the time of writing it is not enabled by default in Google Chrome, iOS or OSX. Furthermore, Firefox 55 was not a stable release at the time of the data capture. Therefore, the current usage of TFO is limited, with only 0.11% of TCP SYN messages requesting TFO. Given the widespread support in clients, this may change quickly and will be important to track in future work. TFO may suffer from similar issues as ECN and MPTCP if middleboxes clear unknown or new TCP options.

TABLE VI. DEFAULT TCP OPTIONS FOR COMMON CLIENT OPERATING SYSTEMS

OS	Version	ECN	SACK	WSOPT	TSOPT
Windows	8.1 and 10		•	8	
	7		•	4	
	XP		•	128	•
Apple	OSX 10.12.5	5%	•	32	•
Apple	iOS 9.3.5	5%	•	16	•
Windows	10 Mobile		•	8	
Android	4.1.2 and 6		•	64	•
Chrome OS	48		•	128	•

TABLE VII. DEFAULT TCP OPTIONS FOR COMMON SERVER OPERATING SYSTEMS

OS	Version	ECN	SACK	WSOPT	TSOPT
Ubuntu Linux	12.04-17.04 LTS	Passive	•	128	•
Windows	Server 2016	•	•	8	
	Server 2012	•	•	128	
	Server 2008		•	4	

F. Operating System Default Options

To reconcile the findings in the passive capture, a follow-up experiment measured the default ECN, SACK, Window Scale and Time Stamp capabilities of client operating systems. Different operating systems running on a virtual machine opened an Apache splash page hosted on a Ubuntu 16.04 LTS server. The TCP options proposed in the TCP SYN were then summarised and are shown in Table VI and VII. For servers and Windows desktop operating systems, previous generations were examined to observe any shifts in the use of particular options.

In both current and legacy OSs we found SACK to be universally enabled. ECN is enabled on both Linux and Windows server versions from 2012. In the case of Linux, ECN support is passive, with Linux negotiating ECN capability when requested by an incoming SYN but not requesting it if the server is the initiator. In practice, whether or not a TCP flow is ECN capable, is driven by the client. In contrast to server OS, there is minimal support for ECN on the client side. We did not observe any clients proposed the use of ECN in their SYN but note that Apple states that 5% of TCP sessions will initiate ECN in current OSs on an experimental basis[15].

All operating systems support WSOPT, but the actual scaling value varies widely. Linux versions consistently use a scaling factor of 7 (128x) while both Windows server and desktop platforms have changed dramatically from one version to the next. For researchers using simulations involving Long Fat Networks (LFN) which could include both satellite

and multi-gigabit enterprise LANs, assumptions regarding the client and server OS combinations are likely to have a significant influence on throughput for large transfers.

Examining the support for TCP Time Stamps, it is a case of Microsoft and the rest. Since Windows 7, Microsoft no longer supports TSOPT on any platform. In contrast, the other operating systems we examined, across all platforms, enable Time Stamps. Microsoft's withdrawn support for TSOPT is the reason why the level of usage, measured in the passive data analysis, was stagnant. It is suspected that the reason why TSOPT might be removed is because it may be used to reveal the uptime. This may be a used to interpret the last time the machine was rebooted and thus the service pack level [19].

IV. Conclusions & Future Work

This study has captured, anonymised and analysed data entering and exiting a large university network. Active measurement, probing real world Internet servers as well as measuring the default behaviour of a wide range of server, desktop and mobile operating systems has also been performed. The results of this measurement study provide some new insights.

Web traffic is quickly transitioning from HTTP to HTTPS with some implications for the Internet. Caching frequently requested content may have diminishing value and law enforcement may find reassembly of files, from captured network traffic, valueless.

The distribution of Internet packet sizes has become increasingly bimodal. The majority of packets are either less than 100 bytes or between 1400 and 1518 bytes. Very few TCP SYN messages stated that they would support a segment size above 1500 bytes, and few packets are sized between 100 and 1400 bytes. The proportion of large UDP frames has increased, which we attribute to the Google QUIC protocol.

An examination of the default TCP options used by mainstream operating systems aligned closely with the shifts observed in real world traffic. Some options such as SACK and MSS can now be assumed, however, there are some open questions. The Windows Scaling option was enabled in all current and legacy operating systems examined and yet 11.5% of SYN segments had no WSOPT present. The discrepancy between the operating system study and the passive study may be a result of legacy servers that do not support the option, or, middleboxes that filter WSOPT. The presence of the latter could have implications for the adoption of many options-based TCP enhancements. Timestamps, or TSOPT, support neither increased or decreased when compared with previous measurement in 2011. In the follow up study, on the default options in operating systems, this was a result of Microsoft removing support for the option due to security concerns.

ECN adoption on the Internet is still marginal, despite several generations of out-of-the-box support for both Windows and Linux servers. While clients are in a strong position to opt in, only Apple is tentatively enabling ECN. The prolonged uptake of ECN is juxtaposed with Google QUIC. Shifting congestion control to the application layer may result in significantly faster changes and Google's position at the server and client side further increase the rate of change. Apple has tested and experimented with numerous TCP options such as ECN, MPTCP and TCPFO and it will be important to track the split between user space QUIC and kernel based TCP in the future [20].

Acknowledgment

We would like to thank and acknowledge Mr Jarren Beveridge and Mr Myles Kennington for supporting this project. Without their help, this study would not have been possible.

References

[1] S. Floyd and E. Kohler, "Internet research needs better models", *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 29–34, January 2003.

[2] A. Medina, M. Allman, and S. Floyd, "Measuring the evolution of transport protocols in the internet", *ACM Computer Communication Review*, 2005.

[3] S. Bauer, R. Beverly, and A. Berger, "Measuring the state of ecn readiness in servers, clients,and routers", in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, New York, NY, USA, 2011, IMC '11, pp. 171–180, ACM.

[4] W. John, S. Tafvelin, and T. Olovsson, "Review: Passive internet measurement: Overview and guidelines based on experiences", *Comput. Commun.*, vol. 33, pp. 533–550, March 2010.

[5] D. Murray and T. Koziniec, "The State of Enterprise Network Traffic in 2012", in *8th Asia-Pacific Conference on Communications (APCC)*, 2012, APCC.

[6] K. Pentikousis and H. Badr, "Quantifying the deployment of tcp options - a comparative study", *IEEE Communications Letters*, vol. 8, no. 10, pp. 647–649, Oct 2004.

[7] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed", in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA, 2007, IMC '07, pp. 111–116, ACM.

[8] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. Diot, "Packet-level traffic measurements from the sprint ip backbone", *Network, IEEE*, vol. 17, no. 6, pp. 6 – 16, nov.-dec. 2003.

[9] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, " TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, 2013.

[10] Y. Cheng, J. Chu, S. Radhakrishnan, and A. Jain, "TCP Fast Open", RFC7413, 2014.

[11] AppNeta, "tcprewrite", Online: http://tcpreplay.synfin.net/wiki/tcprewrite, 2017.

[12] Alexa, "The top 500 sites on the web", Online:http://www.alexa.com/topsites, 2017.

[13] S. Floyd, "Tcp and explicit congestion notification", *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 5, pp. 8–23, Oct. 1994.

[14] M. Honda, Y. Nishida, C. Raiciu, A. Greenhalgh, M. Handley, and H. Tokuda, "Is it still possible to extend tcp?", in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, New York, NY, USA, 2011, IMC '11, pp. 181–194, ACM.

[15] S. Cheshire, "Networking for the modern internet", Internet:http://devstreaming.apple.com/videos/wwdc/2016/714urluxe140lardrb7/714/714_networking_for_the_modern_internet.pdf, 2016.

[16] G. Armitage, N. Williams, L. Stewart, R. V. Valli, and J. But, "Multipath tcp", Online: http://caia.swin.edu.au/newtcp/mptcp/, 2017.

[17] O. Bonaventure, M. Handley, and C. Raiciu, "An overview of multipath tcp", *USENIX login*, vol. 37, no. 5, 2012.

[18] S. Radhakrishnan, Y. Cheng, J. Chu, A. Jain, and B. Raghavan, "Tcp fast open", in *Proceedings of the 7th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2011.

[19] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting", *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, April 2005.

[20] M. Honda, F. Huici, C. Raiciu, J. Araujo, and L. Rizzo, "Rekindling network protocol innovation with user-level stacks", *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 52–58, Apr. 2014.