An Early Look at Multipath TCP Deployment in the Wild

Olivier Mehani[1] Ralph Holz[1] Simone Ferlin[1,2]
Roksana Boreli[1]
[1]NICTA, Sydney, NSW, Australia, {first.last}@nicta.com.au
[2]Simula Research Laboratory, Oslo, Norway, ferlin@simula.no

ABSTRACT

Multipath TCP was standardised in 2013 at IETF. It promises better use of network resources of multi-homed devices for capacity aggregation or seamless fail-over capabilities. The uptake has however been rather slow. Some operating systems support MPTCP out of the box, but little is known about their deployment on the server side. We built a scanning infrastructure to search for MPTCP-capable hosts on the Internet. In this study, we used the hosts on the Alexa Top 1M list to test the platform and gain insights into server support. We find that less than 0.1% of the hosts on the Alexa list currently support MPTCP. Interestingly, their geographic distribution is quite different from that of clients reported in other studies, with the majority of domains being in China. We also find that very few IPs actually expose multi-homing information, suggesting that these early deployments aim at providing reliability rather than capacity aggregation. We also identify some deployment issues.

Categories and Subject Descriptors

C.2.2 [**Computer Systems Organization**]: Computer-communication Networks—*Network Protocols*; C.2.2 [**Computer Systems Organization**]: Performance of Systems—*Measurement Techniques*

Keywords

MPTCP; deployment; measurement; transport; scan

Erratum (2015-10-21)

As mentioned in Section 6, we kept scanning the Top-1M Alexa list after this paper was published. It was pointed out to us that an issue we didn't expect might be confusing our data. As reported in [1], a number of middle-boxes on the Internet were found to mirror options they didn't support. In the case of our MPTCP scans, this means that the `MP_CAPABLE` TCP suboptions were added to return traffic from non MP-enabled hosts, leading to a number of false positives. We started monitoring this issue on 2015-09-09 (by inspecting that the sender's key was different in the reply), and found a consistent number of around 20 MPTCP-enabled servers, with the rest of the data being due to mirroring middleboxes. The data can be visualised at http://nicta.info/mptcp-deployment.

1. INTRODUCTION

With the rapid increase in the adoption of mobile and wireless technologies, a large number of devices now have multi-homing capabilities. Standard TCP is however not able to efficiently utilize a multi-connected infrastructure as it tightly couples the data stream to the source and destination IP addresses used to establish the connection.

Multipath TCP closes the gap between *multipath networks* and *single-path transport* by allowing the use of two (or more) network paths for a data session [2], thereby enabling higher overall throughputs and better connection resilience for applications and services. Multipath TCP has been standardised by the IETF [3], with implementations for a number of platforms: Linux, FreeBSD, iOS 7 and, most recently, Mac OS X Yosemite.

Beyond its ability to utilize multiple network paths simultaneously, the main benefit of MPTCP is its transparent integration with existing applications and most firewall/middle-boxes, where no change to either is necessary to accommodate multipath extensions. This is usually the largest obstacle to overcome for new transport protocols in the Internet [4]. Although the resilience of MPTCP has been demonstrated in a number of studies [5], [6], this issue is the subject of continuous (measurement) studies as new firewalls/middle-boxes deployed in the wild might have an unforeseen impact.

Despite such studies, there is limited knowledge about the scale of current MPTCP deployments. In this paper, we present an active measurement platform to bridge this gap, by scanning sets of hosts for MPTCP support. This allows us to observe current deployments not only in terms of their numbers, but also how end-hosts are making use of MPTCP, *e.g.*, resilience and/or capacity aggregation. We describe our preliminary results of a scan of the top-1M Alexa list. We plan to periodically re-run this scan to collect data about MPTCP deployment over time. We also expect our platform to allow us to better understand how multi-homed or dual-stack systems are deployed.

This paper is structured as follows: Section 2 summarises Multipath TCP's signalling integration into TCP. Section 3

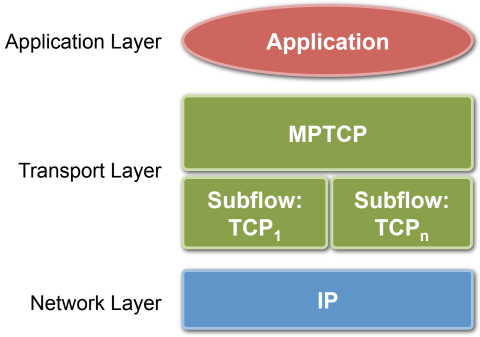


Figure 1: The MPTCP Architecture

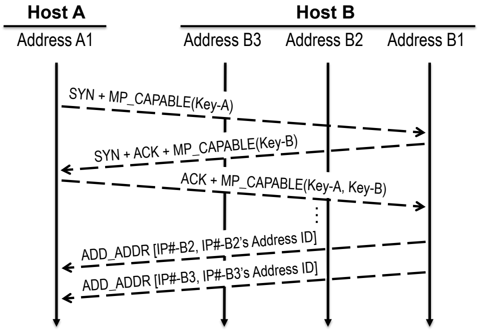

Host A
Host B
Address A1
Address B3
Address B2
Address B1
SYN + MP_CAPABLE(Key-A)
SYN + ACK + MP_CAPABLE(Key-B)
ACK + MP_CAPABLE(Key-A, Key-B)
ADD_ADDR [IP#-B2, IP#-B2's Address ID]
ADD_ADDR [IP#-B3, IP#-B3's Address ID]

Figure 2: MPTCP's Signalling through TCP Options in the TCP's Three-Way Handshake [3]

describes our measurement infrastructure. The results are presented in Section 4 and discussed in Section 5.

2. RELATED WORK

Multipath TCP is a major protocol extension to TCP that supports the transmission of a single data stream across different interfaces. It can therefore increase goodput and resilience of a data stream by efficiently pooling the network's resources [3]. Resource pooling is achieved by presenting a regular TCP socket to the application, which underneath leverages multiple additional TCP connections (sub-flows) on different endpoints, *e.g.*, 3G and WLAN, as shown in Figure 1. These sub-flows together form a Multipath TCP connection. MPTCP signalling uses TCP options on the sub-flows, and looks like regular TCP to middle-boxes. It is therefore readily deployable in today's Internet.

2.1 Multipath's TCP Control Plane

To use Multipath TCP, both end-hosts must enable MPTCP in the system. An MPTCP session always start as a regular TCP session. As shown in Figure 2, additional options are exchanged to negotiate multipath capability. First, an `MP_CAPABLE` sub-option with some additional flags is added to the SYN. If MPTCP-capable, the receiving end-host replies with the same sub-option set in its SYN+ACK.

After confirming MPTCP capability, the end-hosts still need to discover additional endpoints and create more sub-flows. The remote end-host usually advertises additional IP addresses (IPv4 and/or IPv6) with `ADD_ADDR` sub-options after the first sub-flow's three-way handshake has completed.

How and when new sub-flows are opened, *e.g.*, IP address advertisements to remote end-hosts and number of MPTCP sub-flows per endpoint, is defined by MPTCP's path manager [3]. In this paper, we use the default path manager of the Linux implementation, *fullmesh*.[1] With this path manager, MPTCP will create a full mesh of sub-flows among the available IP addresses from both end-hosts: for each `ADD_ADDR` received from the destination, the initiator sends an `MP_JOIN` sub-option in the handshake of the sub-flows from its own addresses to join them to the existing MPTCP session.

After the second sub-flow has been established, MPTCP's data plane is now in charge of keeping track of the data sent on each sub-flow. This requires an additional sequence number level `DSS` (data sequence signal) on top of TCP's regular sequence number to re-assemble it. Although `DSS` usage has diverse goals [4], it is relevant for MPTCP to control data retransmission and guarantee connectivity in the face of network failures, handovers [7] or simply to keep track between sub-flows, *e.g.*, with a transfer on dual-stack systems [5]. This remains completely transparent to applications, which only see a regular TCP socket (see top of Figure 1).

2.2 Known Multipath TCP Deployments

Although MPTCP is well suited to today's Internet, not much is known about its deployment in the wild. Some research work has studied the incentives for deployment [8]. More recently, [6] studied traffic captures over a short period of time, with the aim to study proper operation of MPTCP in the wild.

There are implementations for the most common free operating systems, such as Linux[2] or FreeBSD[3]. Apple has also included MPTCP in iOS, but limited its use to the traffic generated by Siri, and to fail-over, *e.g.*, change from Wi-Fi to a 3G/4G network. Recent news suggest that MPTCP is no longer limited to Siri, but remains to be used only for fail-over purposes.[4] We also found that MPTCP had been added to the latest Mac OS X (10.10.3 Yosemite) and turned on by default[5]. A survey of implementations [9] also identified a few commercial actors using MPTCP: Networks, Netscaler, and Tessares.

The Linux MPTCP website provides a cumulative report of all MPTCP-capable hits to their web server.[6]

3. EXPERIMENTAL SETUP

Our goal is to identify MPTCP deployments in the wild. We built a scanning platform to establish (MP)TCP connections to selected addresses and inspect replies for MPTCP sub-options. We performed a pilot study on the Top 1M list of popular Web sites as identified by Alexa[7].

[1] http://multipath-tcp.org/pmwiki.php/Users/ConfigureMPTCP

[2] http://www.multipath-tcp.org

[3] http://caia.swin.edu.au/urp/newtcp/mptcp

[4] https://support.apple.com/en-au/HT201373

[5] As shown by `sysctl -a | grep mptcp`

[6] "HoneyMap": http://multipath-tcp.org/honeymap/map.html

[7] http://www.alexa.com/

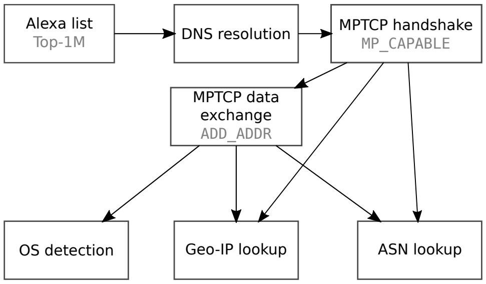


Figure 3: Tool chain of scanners and lookup tools employed in our experiments.

Figure 3 shows our tool chain. The chain is executed once a day, with one exception: as the detection of remote operating systems is a lengthy and also very intrusive process, this is only done occasionally and started manually.

Alexa list.

The tool chain begins with a download of the Alexa Top 1M list.[8] This list is sometimes inconsistent and must thus be cleaned of artefacts (such as URLs or IP addresses instead of domain names). The result is a list of second-level domains, which we then extend by duplicating every entry and adding a `www` prefix to the duplicate. We obtain roughly 1.98 million DNS names this way.

DNS resolution.

In the next step, we use GNU `adns`[9] to resolve the domain names to IP addresses. `adns` works as a stub resolver that we use to query all A resource records.

MPTCP port scan.

The next step is to run the `zmap` scanner [10] on TCP port 80 (HTTP). We implemented a probe module for `zmap` that sets the `MP_CAPABLE` MPTCP sub-option in the outgoing probe and checks whether a response carries the option, too. This indicates a server where MPTCP is enabled.

TCP handshake.

The next tool in the chain scans the thus obtained IP addresses a second time: it connects to them with `wget` and uses the `tshark` tool[10] to read TCP options and, in particular, the list of additional IP addresses on which a server claims to be reachable (`ADD_ADDR`). As a result of this step, we now hold a complete list of IP addresses that were either identified directly as MPTCP-enabled or via announcement by a server.

The next steps are all carried out in parallel.

ASN lookup.

All IP addresses of MPTCP-enabled hosts are fed into two tools: `pyasn`[11] and `cymruwhois`. `pyasn` works by downloading the Routing Information Base (RIB) from the time of the scan from Routeviews[12] and using it to determine the Autonomous System (AS) holding the IP prefix that contained the IP address in question at the time of the scan. `cymruwhois` allows us to determine the AS operator and the country of registration. It queries the free WHOIS service by Team Cymru.[13]

Table 1: Number of MPTCP-capable servers (our scan) and clients (HoneyMap) per country. Only countries with MPTCP-enabled servers are listed.

Country code	Servers	Ratio of servers [%]	Clients	Ratio of clients [%]
CN	319	90.11	842	2.65
US	12	3.39	6742	21.25
HK	5	1.41	163	0.51
FI	4	1.13	1319	4.16
DE	3	0.85	4493	14.16
JP	2	0.56	2918	9.20
TH	2	0.56	16	0.05
TR	2	0.56	5	0.02
CA	1	0.28	809	2.55
AU	1	0.28	311	0.98
NO	1	0.28	102	0.32
VN	1	0.28	54	0.17
CL	1	0.28	14	0.04

Geolocation lookup.

We use the free Maxmind databases (country and city)[14] to obtain the approximate geolocation of the hosts.

OS scan.

We use the `nmap`[15] tool to determine the operating system and kernel version of the remote, MPTCP-enabled host.

4. RESULTS

Unless otherwise stated, the results presented here are based on a scan run on 2015-05-29. For this dataset, we resolved 1,991,262 domain names to 452,008 unique IP addresses. Out of these hosts, 428,895 responded, and 353 (0.08%) advertised multipath capabilities. Data from other scans is qualitatively similar.

4.1 Where Are MPTCP-capable Hosts?

Figure 4 shows the geographic repartition of MPTCP-enabled servers as found by our scans. For comparison purposes, we also include the localisation of MPTCP-enabled clients as reported by the MPTCP HoneyMap (containing 31,729 records). Surprisingly, there is little overlap. China, for example, is found to have the largest number of servers (we checked the full Alexa list to confirm there was no initial bias towards China) (90.11%), but very few clients (2.65%). Table 1 extends this analysis for other countries.

This is confirmed in Table 2, which lists the number of MPTCP-enabled hosts per Autonomous Systems. Alibaba

[8] https://s3.amazonaws.com/alexa-static/top-1m.csv.zip
[9] http://www.chiark.greenend.org.uk/~ian/adns/
[10] https://www.wireshark.org/docs/man-pages/tshark.html
[11] https://pypi.python.org/pypi/pyasn
[12] http://www.routeviews.org/
[13] http://www.team-cymru.org/Services/ip-to-asn.html#whois
[14] http://dev.maxmind.com/geoip/geoip2/geolite2/
[15] https://nmap.org/

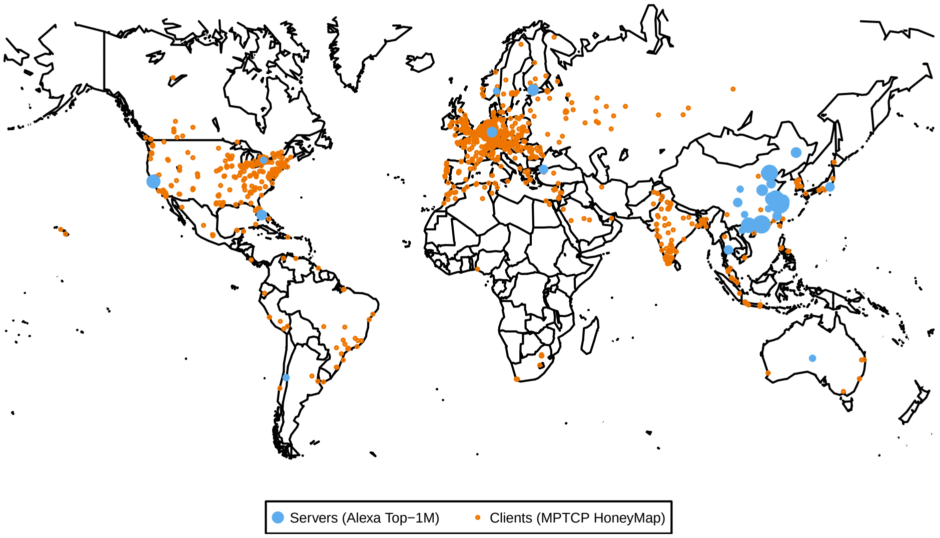


Figure 4: Geographic repartition of MPTCP-enabled servers from the Alexa Top-1M. The radius of the server dots is log log-proportional to the number of servers found in a given location.

is the main user of MPTCP , with 45.92% of the detected servers, followed by ChinaNet (20.85%), accross a few ASes.

4.2 Who Are They?

Next, we look at which domains the MPTCP-enabled IPs serve. Figure 5 shows a CDF of the Alexa rank of the MPTCP-capable domains. We find that the 353 IPs we identified serve content for 520 (0.05%) unique domains (conflating the `www` subdomain). Table 3 lists the first 8 domains in the Alexa list found to support MPTCP.

Most of the domains are however not fully MPTCP-capable across their endpoints, with 68 out of 85 multi-addressed domains having both MPTCP-capable and -incapable addresses (as early as the 17th, at Alexa rank 7765).

4.3 What Are They?

We now consider the additional information obtained during the data exchange. We are mainly interested in information about multi-homing and support for multiple address families, which can be obtained through `ADD_ADDR` headers on packets following the handshake.

In this respect, we barely found any endpoint advertising a multi-homed-configuration: 1 advertised IPv4 addresses, and 0 for IPv6. This host advertised 2 IPv4 addresses. One was in the same AS as the IP we scanned, though it was not part of the list of addresses resolved from Alexa. More interestingly, the second address was in a private range, and therefore not globally routable. In similar scans performed during the previous days, we saw minor variations in those results: one host was seen advertising an IPv6 address; another advertised up to 6 IPv4 addresses; and a handful of other hosts exposed IP addresses in various private ranges.

An additional observation we could make in those exchanges is that our scanning machine sometimes sent packets with an `MP_FAIL` sub-option (in 6 cases).

Finally, we tried to identify the operating systems of those endpoints we found to support MPTCP. This turned out to be a relatively frustrating exercise as the accuracy of `nmap` was generally relatively low. We also found this phase to take a prohibitively long time, even with the limited number of hosts and a time limit for each. We therefore do not run this phase everyday. The results presented thereafter are from 2015-05-25. The average accuracy, as reported by the tool, was 86.4% with a standard deviation of 1.79. Often, we had more than one candidate with equal accuracy for the same IP address. We thus chose to compute the occurrences of operating system as follows. We computed a list of distinct operating system candidates[16] and then summed up how often a candidate appeared in our data set. Every time a candidate appeared in the results for one IP address, we weighted the occurrence by interpreting the accuracy as a percentage (*i.e.*, 85% is interpreted 0.85) and dividing by the number of candidates for that IP address. The result was added to the total occurrence for that candidate OS. Table 4 shows those with an occurrence higher than 10.

The first thing to note here is that none of the strings that identify the operating systems point to kernels that we would readily identify as known implementors of MPTCP. Patched Linux kernels, in particular, are missing. The second noteworthy observation is that many Apple devices are among the results. In the unfiltered data set, we find 331 candidates that contain the strings “Apple iOS” or “Apple iPhone”. At first sight, this may seem plausible. However, iOS 4 is not known to support MPTCP. We can speculate this might be misidentification of an iOS 7 device—but even then, the question would remain of why this device would reply to a port scan on port 80. This leaves the interesting cases of devices from network equipment manufacturers. Allied Telesis alone appears 1137 times in the raw dataset as a candidate, and Cisco CSS more than 300 times (note these

[16] By “operating system candidate”, we refer to the “operating system family” output of `nmap`.

Table 2: Number of MPTCP-enabled IPs per Autonomous Systems

AS name	Count
CNNIC-ALIBABA-CN-NET-AP Hangzhou Alibaba Advertising Co.,Ltd.,CN	154
CHINATELECOM-GUANGDONG-IDC Guangdong,CN	51
CHINANET-BACKBONE No.31,Jin-rong Street,CN	44
CHINA169-BJ CNCGROUP IP network China169 Beijing Province Network,CN	17
CHINA169-BACKBONE CNCGROUP China169 Backbone,CN	13
CHINANET-JS-AS-AP AS Number for CHINANET jiangsu province backbone,CN	11
CHINANET-IDC-BJ-AP IDC, China Telecommunications Corporation,CN	9
CHINANET-SH-AP China Telecom (Group),CN	9
CNNIC-ALIBABA-CN-NET-AP Alibaba (China) Technology Co., Ltd.,CN	9
CHINATELECOM-HE-AS-AP asn for Hebei Provincial Net of CT,CN	4
CNNIC-BAIDU-AP Beijing Baidu Netcom Science and Technology Co., Ltd.,CN	4
SANOMA-AS Sanoma Data Oy,FI	4
(Other)	25

Table 3: Most popular domains supporting MPTCP.

Domain	Rank	Endpoints
baidu.com	5	3
tmall.com	19	1
hao123.com	24	4
alibaba.com	55	2
alipay.com	100	2
cnzz.com	549	1
etao.com	740	2
aliyun.com	1582	1

are candidates, and one IP address may yield many candidates).

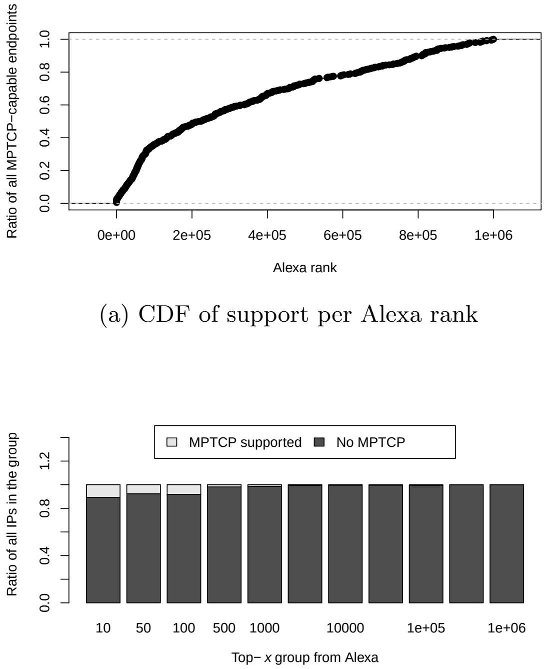
5. DISCUSSION

The data we collected during this scanning campaign shows a very limited uptake of MPTCP on server-side. Nonetheless, there are a few findings worth discussing.

5.1 Difference in Location and Numbers

The vast majority of endpoints supporting MPTCP are located in China. An early adopter of the protocol appears to be the Alibaba group, a Chinese e-commerce platform, is the entity with the largest deployments both for its most-visited domains and sub-domains and its ASes. More generally, our results suggest that more popular (*i.e.*, better ranked) sites are paying more attention to MPTCP than worse ranked sites, with Chinese domains in the lead.

The number of MPTCP-capable endpoints is also much lower than that of clients from the HoneyMap project. We attribute this mainly to the difference in collection methods, data schema, and vantage points. The HoneyMap is a list of all visitors to the `multipath-tcp.org` website where clients supported MPTCP. Considering its objective, it is to be expected that this site would attract a larger-than-average number of MPTCP clients. The list does not contain date and time information of the visit and is therefore a cumulative view of all visits to date. It does not capture visits by non-MPTCP clients, either. Conversely, the dataset we presented is a daily snapshot of websites sorted by global popularity. We therefore believe that it represents a more accurate view of current deployments of MPTCP at large.



(a) CDF of support per Alexa rank

(b) MPTCP support in top-domains groups of increasing sizes

Figure 5: Distribution of MPTCP-capable endpoints per domain.

5.2 Multi-homed and Dual-Stacked Hosts

One result we were expecting from these scans was to gain more insight into multi-homed and dual-stacked hosts. While we confirmed that our assumption that we can learn about additional addresses of a host by observing MPTCP traffic is correct, we found little positive information to identify multi-connected hosts.

Only half a dozen hosts advertised additional IP addresses across our daily scans. A few were publicly routable IPv4 addresses, and only one was an IPv6 address. This would suggest that most hosts are not multi-homed nor dual-stacked. However, our current sample size of MPTCP-capable hosts is extremely small at this point and cannot provide strong confidence yet.

A scan targeting a wider range of IP addresses would increase the size of our MPTCP sample, from which more confidence could be gained about multi-homed hosts. We note, however, that this might introduce a sampling bias of its

Table 4: Operating system detection, applied to MPTCP-capable IPs.

Name	Occurrence
Allied Telesis AT-8000S; Dell PowerConnect 2824, 3448, 5316M, or 5324; Linksys SFE2000P, SRW2024, SRW2048, or SRW224G4; or TP-Link TL-SL3428 switch	64.58
Cisco CSS 11501 switch	44.14
Yamaha RX-V2067 or RX-V3900 audio receiver	35.10
Cisco SG 300-10, Dell PowerConnect 2748, Linksys SLM2024, SLM2048, or SLM224P, or Netgear FS728TP or GS724TP switch	34.60
Sagem My du@l radio 700 Internet radio	31.88
3Com Baseline Switch 2924-SFP or Cisco ESW-520 switch	28.86
Allied Telesyn AT-AR410 router	24.29
Apple iOS 4.3.3	16.28
Linksys SRW2000-series switch	15.12
HP 9100c Digital Sender printer (J3113A)	10.98

own, as administrators deploying MPTCP machines could be more enclined to provide them with multiple uplinks to exercise capacity-aggregation features.

The fact that the vast majority of MP-capable hosts did not expose any additional endpoints suggests that capacity-aggregation is not the main incentive. Rather, this might be an attempt to better support mobile clients, such as iOS devices.

5.3 Deployment Issues

In conducting our MPTCP data exchange test to collect ADD_ADDR options, we identified two issues which might prove problematic in further MPTCP deployments.

First, multi-homed hosts tended to advertise private IP addresses, presumably from a management or otherwise internal network. This creates a potential security risk, already mentioned in [11], as it allows to discover and enumerate private networks. While we cannot assert that every multi-homed host with an internal network address also advertised it, this is an issue that system administrators will have to address. One way to mitigate the problem is to configure the MPTCP path-manager to only expose public addresses.

Perhaps the most unexpected observation in our scans are the results from the OS fingerprinting. We suspect the many implausible OSes to be artefacts due to middle-boxes on the path to the endpoints, which alter the flow of TCP packets and their headers. This hypothesis is supported by our observation of packets with the MP_FAIL MPTCP option. This option is generally used to signal a flow for which both correspondent nodes cannot agree on a match of their sequence numbers, which is often due to middle-boxes or other semi-transparent proxies (which would act as the endpoint after the handshake).

Besides obvious OS misidentification in the OS fingerprinting, some interesting results stand out, such as Allied Telesis, Cisco, or Linksys. These companies are known to produce middle-boxes. The presence of these boxes on the path might hinder MPTCP use, as they would limit the number of paths available between two otherwise MPTCP-capable endpoints.

Unfortunately, our current scanning pipeline does not allow us to test the middle-box hypothesis any further. Extensions allowing to do so would include comparing TCP RTTs to network latency as measured with `ping`, as well as collecting `traceroute` or Tracebox[17] data.

6. CONCLUSION

We have described a scanning platform which we use to test hosts for MPTCP capability. Our procedure then establishes a data connection with the detected hosts and collects information exchanged in MPTCP sub-options. The most interesting such option is ADD_ADDR which a host can use to advertise other IPv4/v6 addresses under which it is reachable. Beyond MPTCP adoption, we intend to use this platform to establish a map of multi-homed and/or dual-stacked hosts.

As a pilot study, we scanned IP addresses from the Top 1M hosts from the Alexa lists. We found that about 0.1% of both IPs and domains were served by MPTCP-capable endpoints. A large part of these deployments were located in China. We found very few multi-homed hosts, which we took as an indication of an effort to better support mobile clients rather than to aggregate capacity. Moreover, the few cases where we found further advertised addresses tended to raise security concerns such as exposing private IP addresses. We also observed oddities in the packet exchanges that we hypothesise to be due to semi-transparent middle-boxes.

Our future work in this project is three-fold. First, we will continue to take periodic snapshots of the Alexa Top 1M and extend some scans to wider ranges, towards /0. We will set-up a web dashboard at http://nicta.info/mptcp-deployment that allows to explore this data. This will allow us to get a present a chart of MPTCP deployment over time. This is very relevant at a time when Apple has just enabled MPTCP for all applications, in both mobile and desktop OSes. Second, we will extend the scanning platform to address the shortcomings identified in this study, such as the need for more measurements geared towards identifying middle-boxes. Finally, we want to complement this study with an analysis of live traffic to capture cases where both clients and servers are MPTCP-capable and establish a fully functional multipath session.

References

[1] G. Detal, B. Hesmans, O. Bonaventure *et al.*, "Revealing middlebox interference with Tracebox", in *IMC 2013*, Barcelona, Spain, Oct. 2013, ISBN: 978-1-4503-1953-9.

[2] C. Raiciu, C. Paasch, S. Barre *et al.*, "How hard can it be? designing and implementing a deployable multipath TCP", in *NSDI 2012*, 2012.

[3] A. Ford, C. Raiciu, M. Handley and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses", RFC 6824, Jan. 2013.

[4] M. Honda, Y. Nishida, C. Raiciu *et al.*, "Is it still possible to extend TCP?", in *IMC 2011*, Berlin, Germany, Nov. 2011.

[5] I. Livadariu, S. Ferlin, O. Alay *et al.*, "Leveraging the IPv4/IPv6 identity dualityby using multi-path transport", in *GI 2015*, Hong Kong, Apr. 2015.

[17] http://www.tracebox.org/

[6] B. Hesmans, H. Tran-Viet, R. Sadre and O. Bonaventure, “A first look at real Multipath TCP traffic”, in *TMA 2015*, vol. 9053, Barcelona, Spain, Apr. 2015. DOI: 10.1007/978-3-319-17172-2_16.

[7] C. Paasch, G. Detal, F. Duchene *et al.*, “Exploring mobile/wifi handover with multipath tcp”, in *CellNet 2012*, Helsinki, Finland, Aug. 2012, ISBN: 978-1-4503-1475-6. DOI: 10.1145/2342468.2342476.

[8] H. Warma and H. Hämmäinen, “Researching Multipath TCP adoption”, in *AIMS 2010*, vol. 6155, Jun. 2010. DOI: 10.1007/978-3-642-13986-4_8.

[9] P. Eardley, “Survey of MPTCP implementations”, Internet-Draft draft-eardley-mptcp-implementations-survey-02.txt, Jul. 2013.

[10] Z. Durumeric, E. Wustrow and J. A. Halderman, “Zmap: Fast internet-wide scanning and its security applications”, in *USENIX Security 2013*, Washington, D.C., USA, 2013.

[11] C. Pearce and P. Thomas, “Multipath TCP — breaking today’s networks with tomorrow’s protocol”, in *BlackHat USA 2014*, Las Vegas, NV, USA, Aug. 2014.