Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Andreas Germann

Measuring Internet Path Transparency for Transport Protocol Extensions

27.06.2016

Semester Thesis SA-2016-49
April 2016 to July 2016

Tutor: Prof. Dr. Laurent Vanbever
Supervisors: Brian Trammell, Mirja Kuehlewind

Acknowledgements

I thank my supervisors Prof. Dr. Laurent Vanbever, Brian Trammell and Mirja Kuehlewind for their support during my work on this project. Furthermore, I thank the team behind Pathspider for providing the software infrastructure for my measurements and for their support in writing my measurement probe.

Abstract

In this work, we measure the internet path transparency of the tranport protocol extension TCP Fast Open. TCP Fast Open aims to reduce the latency of webservices. It does that by allowing data in the payload of the initial [SYN] packet of a TCP handshake.
TCP Fast Open can be broken by middleboxes in the internet. For a set of path-diverse connections we want to find out how many of these connections are broken by a middlebox. With this data we estimate how usable TCP Fast Open is in todays internet infrastructure.
Our results show that currently the rollout of TCP Fast Open is rather limited with less than one in thousand webhosts supporting it. Where TCP Fast Open is implemented, it works quite well. In our measurement, more than 9 in 10 connections were successful.

Contents

Chapter 1

Introduction

Transport layer protocols are an essential building block in todays internet infrastructure. Based on the internet layer, they provide specific communication services to the application layer. These transport protocols are continuously extended to increase throughtput, decrease latency or implement new services. Examples for such extensions are Multipath TCP, Explicit Congestion Notification or TCP Fast Open.
Such extensions can be broken in the internet when packets are dropped or protocol options are stripped by middleboxes. This is a hindrance in the deployment of these extensions. In this work, we will measure how well TCP Fast Open connections work in the internet.
TCP Fast Open, we will call it TFO from now on, is an extension of TCP. It aims to reduce latency of webservices by sending application data in the initial [SYN] packet of the TCP three-way handshake. TFO was developped at Google and their team estimated that TFO could reduce latency by 15% on average. For queries that fit into the payload of a single packet, this can even reduce the latency by one rount-trip time. For example Google searches usually fit into one packet payload. Therefore, the server can already process the search after receiving the [SYN] packet. Another potential use is in DNS over TCP where DNS queries could be sent in the initial [SYN] packet.
TFO is based on a TCP option and can therefore be broken by middleboxes that stip this option. Also, TFO breaks on middleboxes that block [SYN] packets containing data. Therefore, we want to measure todays internet path transparency for TFO to see how well it works in the internet infrastructure.
To measure the path transparency of TFO we try to open TFO connections from different starting points to different end points. This gives us path-diverse connections for which we determine whether TFO works or whether some middlebox breaks it along the way. To do this, we use Pathspider[1], a tool for A/B testing which we extend for TFO testing. Pathspider has already been used to measure the path transparency of Explicit Congestion Notification[5].
Based on these results, we try to determine how often TFO is broken in the internet. This should allow a conclusion of how usable TFO is.

Chapter 2

Background

Our goal was to measure the internet path transparency of transport protocol extensions. These protocol extensions aim to improve the quality of service of existing transport protocols like for example TCP. Examples for such extensions are Multipath TCP, Explicit Congestion Notification or TCP Fast Open. In this work we measure the path transparency of TCP Fast Open.

2.1 TCP Fast Open

TFO is an extension of TCP that aims to reduce the latency of webservices. This is achieved by sending application data in the initial [SYN] packet of a TCP handshake. TFO was developped at Google and the developpers outline the potential of TFO in their work. They mention latency as an important quality factors in todays webservices and they identify the TCP handshake as one of the main bottlenecks. Their studies showed that most objects in HTTP connections have a size of only a couple of KB. In such connections, the TCP handshake is one of the main causes of latency. Their simulations show that TFO could reduce the average latency of HTTP transactions by 15%.[2] The reduction of latency is especially large when a query fits into one packets payload. In that case, the server can process the query right after receiving the [SYN] packet. In the ideal case, this reduces the latency by one round-trip time. Examples for this use are Google searches or DNS over TCP.[3]
TFO is based on a cookie that serves as an authenticatior for the client. The server only processes the data in a [SYN] packet if it contains a valid cookie. This prevents an attacker from exhausting the servers ressources by flooding it with spoofed packets which would all be processed if there was no authentication.
This cookie is created by the server and issued to the client in an initial TCP handshake. The cookie is a MAC created from a secret key and the clients IP address. It is transmitted within a TCP option field and has a length between 4 and 16 bytes When a client wants to set up a TFO connection, it requests a TFO cookie by starting a regular TCP handshake with an empty TFO option in the [SYN] packet. If the server supports TFO, it returns a TFO option with a cookie in the [SYN,ACK] packet. The client stores this cookie for later use and continues the TCP connection. After the client has received a cookie, it can use it by starting a new TCP handshake. This time the client sends the initial [SYN] packet with the cookie in the TFO option and the data in the payload. The server validates the cookie. If it is valid, the server acknowledges the [SYN] and the data in the [SYN,ACK] packet. If it is not valid, the server only acknowledges the [SYN]. If the data was acknowledged, the client finishes the TCP handshake and can send further data if necessary. If the data was not acknowledged, the client continues like in a regular TCP handshake and then retransmits the data.
The server acknowledges the [SYN] and the data by setting the acknowledgement number to the sequence number plus the length of the data plus one. The server does not acknowledge the data if it only increases the sequence number by one.[4]
Figure 2.1 shows the sequence diagrams of a TFO cookie request and how it is used.

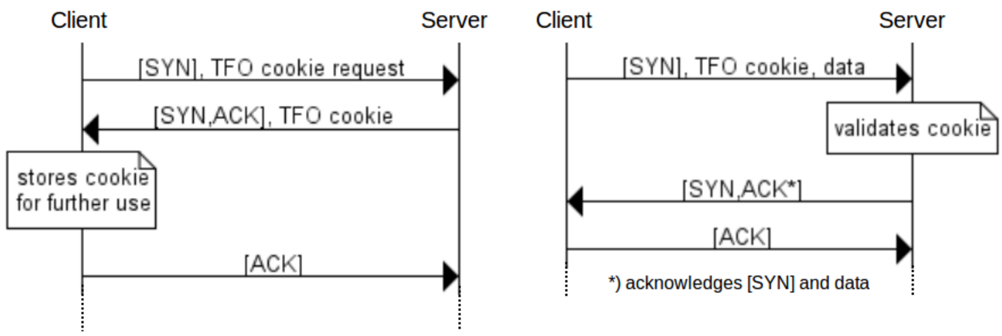

Figure 2.1: Sequence diagrams of the TFO cookie request (left) and the TFO data transmission (right)

2.2 Path Transparency

Internet path transparency is a measure to determine, how well a specific protocol or extension is passed through the internet. Or on the other hand, how many middleboxes in the internet break this protocol or extension.
A connection through the internet is not a single link from the sender to the receiver. A packet runs through multiple nodes until it has reached the receiver. Sender and receiver have hardly any say in the selection of the path or the nodes that pass on their packets. So an extension or protocol can be broken by middleboxes that they have no influence on and can not avoid.

2.3 Pathspider

For our measurements we used Pathspider[1]. Pathspider is a tool for A/B testing. It tries to open connections from point A to point B to determine whether this connection works. Pathspider itself is a framework that parallelizes connection attemps via multithreading. This allows a user to quickly test a lot of connections. Furthermore, Pathspider takes care of packet capture and flow handling. This allows an easy analysis of the flows of different connections. Within this framework, a user implements his own spider. This spider determines, which extension is tested and how the flows are analyzed. An already existing spider was ECNSpider that was designed to test connections with Explicit Congestion Notification.[5]

2.3.1 A/B Testing

A/B testing means that Pathspider checks whether a connection between A and B is possible for certain protocols or extensions. Point A is always the machine that runs Pathspider. From there, connections are tried to different points B. When testing a certain protocol or extension, Pathspider first tries to open a regular TCP connection to the endpoints to determine whether there actually is a path. This is done to prevent blaming an unreachable endpoint on the tested extension. In a second step, Pathspider tries the connection with the extension that is to be tested.
Figure 2.2 shows an example network that could be tested. As an example, there is one middlebox that breaks the tested extension and there is one completely broken link. The connection from A to B would work with both regular TCP and the extension. The connections from A to B’ would only work with regular TCP since the middlebox breaks the extension along the way. The connection from A to B” would work with neither since there is no usable path.

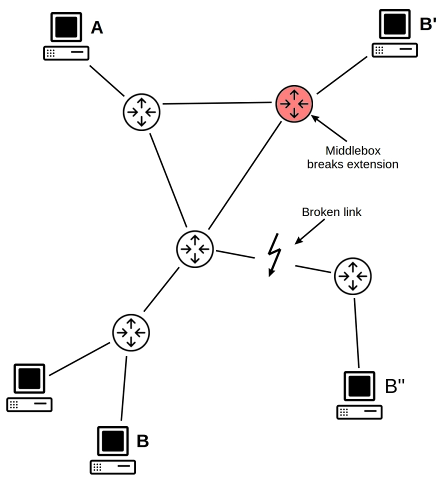

Figure 2.2: Example of a network with broken links and extension-breaking middleboxes

2.3.2 Spiders

Pathspider is a framework where a user can run his own tests. Pathspider takes care of the parallelization of connections, the capturing of the packets at the network interface and the assigning of targets from the target-queue to the parallel threads. A user-defined probe that is run in this framework is called a spider.
A spider consists of four parts. These parts are defined by the user to fit his need. This spider is then run within Pathspider. The first part is the worker that opens and closes connections to the targets that were assigned to it. Multiple workers are run in parallel to increase the testing rate. The workers are definded by the user since he might want to open the connections in a specific way, for example with certain socket options. The second part is the configurator. The configurator applies system-wide changes that might be necessary to open a connections with a specific extension. This configurator is synchronized with the workers and changes the settings between the connections with and without the extension. An example for this is a *sysctl* call that turns on Explicit Congestions Notification in the kernel. The third part is the observer. The observer captures packets at the network interface and orders them into flows. These flows are processed with user-defined functions that allow the user to analyze the flow according to his wishes. The last part is the merger that merges the list of the attempted connections with the results from the observer.
For our measurement, we implemented TFOSpider to run tests for TFO with Pathspider.
Figure 2.3 shows the parts of Pathspider and how they work together.

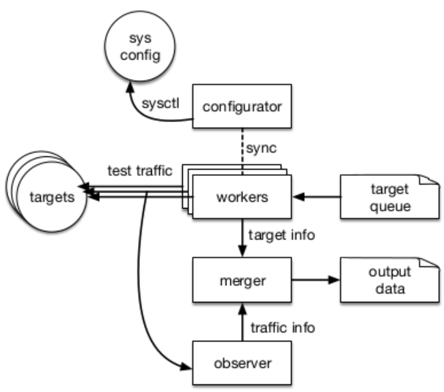

Figure 2.3: Elements of Pathspider and how they work together[1]

Chapter 3

TFOSpider

For our measurements we implemented TFOSpider. It works in the Pathspider framework and determines the workability of connections with TFO. To do that, it first opens a regular TCP connections to make sure that there is a path to the target. Then, it opens a TFO connection to see whether TFO works on that path.

3.1 Connections

The connections of the TFOSpider are opened by the workers. These workers run in parallel to increase the testing rate in the measurement.
In a first step, the workers open connections to the targets with regular TCP. These connections determine whether the target is actually reachable.
In the second step, the workers open connections to the targets with TFO. The configurator is not used in the TFOSpider since there are no system-wide changes necessary between regular TCP and TFO connections. For each TFO connection, two separate connections are opened. In the first connection, TFOSpider requests a TFO cookie from the target server. This means it sends a [SYN] packet with an emptry TFO option. If the target server supports TFO, it answers with a [SYN,ACK] packet that contains a TFO option with a cookie. If not, the server answers with a regular [SYN,ACK] packet. TFOSpider then finishes the handshake and tears down the connection. In the second connection, TFOSpider tries to send data in the initial [SYN] packet with the previously received cookie. The server then asweres with a [SYN,ACK] packet that ideally acknowledges the data. After completing the handshake, the connection is torn down again. The cookie handling and the potential retransmission of not acknowledged data is performed by the kernel. The application socket is not involved in the process and therefore receives no information on the TFO status or a retransmission. To determine whether TFO worked we need the flow analysis.
Figure 3.1 shows a flowcharts of the regular TCP connection and the TFO connections attempted by TFOSpider.

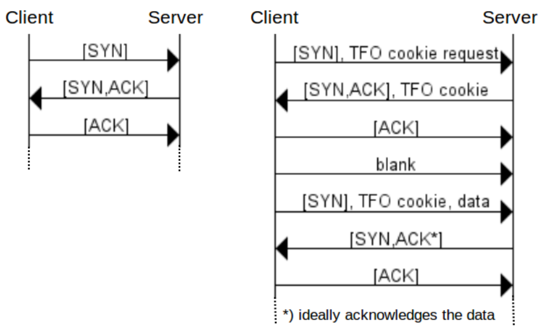

Figure 3.1: Sequence diagrams of the TCP connection (left) and the two TFO connections (right)

3.2 Flow Analysis

During these connections, the observer captures the packets that are sent and received. With our observer functions we try to determine whether TFO has worked or whether the connections has fallen back to regular TCP. For the analysis we are only interested in the second TFO connection. The necessary information about the first TFO connection can be derived from the second.
The flow of the second connection is analyzed packet by packet. We first look for a [SYN] packet containing a TFO cookie and data. If there is one, it means that the endpoint has issued a cookie in the first connection. If there is none, the endpoint has not issued a cookie. If we have found a TFO data packet, we look for a retransmission of the data. If there is a retransmission, the TFO data transmission failed at some point. If there is no retransmission, we look whether the data has been acknowledged. If it has been acknowledged, TFO has worked. The case where there is no retransmission and no acknowledgement should not occur since it would imply a failure of the TCP connection.
This analysis is able to distinguish between the three cases of *No Cookie Recieved*, *Data Transmission Failed* and *TFO Working* even though there are more different cases (See Section 3.3). The reason for this is the limitation of available ressources for processing the flows. A more detailed analysis would increase the demand of ressources.
Figure 3.2 shows a flowchart of how the state of a TFO connection is assessed.

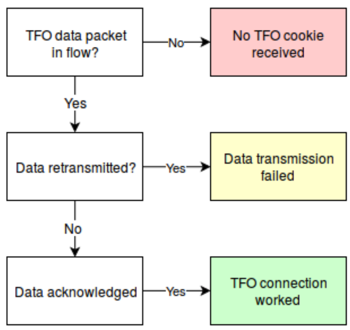

Figure 3.2: Flowchart of the status assessment of a TFO connection

3.3 Failures of TFO

In our work, we observed three different ways that a TFO connection can fail. Each of these three cases can have different causes, but due to our limited capabilities of observation we can only give potential explanaitions for the failure. The observation was limited to capturing packets at the network interface of the sending machine and in limited cases on the receiving machine. TFO server functions were not implemented on our machines, so a full TFO connection could not be set up to them for testing.

3.3.1 No Cookie Received

The first and most frequent case is when the client does not receive a TFO cookie. This can have two causes. The first being that the server simply does not support TFO. In that case it will ingore the TFO option in the [SYN] packet and continue the TCP handshake.
Another possible reason for not receiving a TFO cookie is a middlebox that stripps the TFO option from the packets. In that case, the server only receives the [SYN] of a regular TCP handshake and will respond accordingly. We determined this as the reason why TFO does not work from within the ETH network (See Section 4.1.2). We sent a TFO cookie request to another one of our machines and observed the incomming packets there. These packets were all stripped of the TFO option.
Figure 3.3 shows possible sequence diagrams of the two described scenarios.

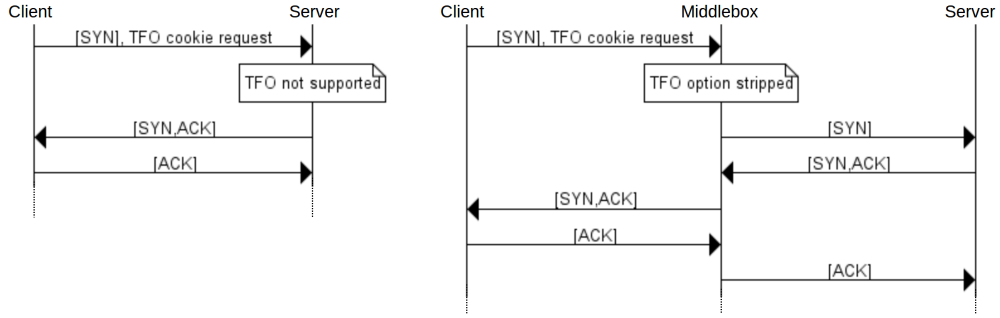

Figure 3.3: Possible sequence diagrams of the case *No Cookie Received*; TFO not supported (left), TFO option stripped (right)

3.3.2 Data Packet Corrupted

The second case is when the client recieves a cookie but the TFO data packet is corrupted. In this case, the client sends a [SYN] packet with the cookie and the data and recieves a [SYN,ACK] packet that only acknowledges the [SYN]. The server does this by setting the acknowledgement number of the [SYN,ACK] packet to the [SYN] packets sequence number plus one instead of increasing the sequence number by one and the length on the payload data. In this case, the client continues with the regular TCP handshake and retransmits its data. Note that in this case, the TFO cookie request has already worked across all involved middleboxes, so these middleboxes only damage the data packets but not with the packets of the TFO cookie request. Due to the limited observation capabilities we could not determine how exactly the data packet was changed. But our observations showed two slightly different subcases.
In the first subcase, the [SYN,ACK] packet is just a regular answer of a TCP handshake. An explanaition for this case is a middlebox that strips the data packet of its payload and possibly TFO option. Therefore, the server only sees the [SYN] and acknowledges it. It does not matter whether the middlebox stripped the TFO option since the option itself is not specifically acknowledged.
In the second subcase, the [SYN,ACK] packet itself contains a TFO cookie. This can be explained with a middlebox that strips the data from the [SYN] packet and reduces its TFO option to a cookie request by removing the cookie from the option.
Figure 3.4 shows possible sequence diagrams for the two subcases.

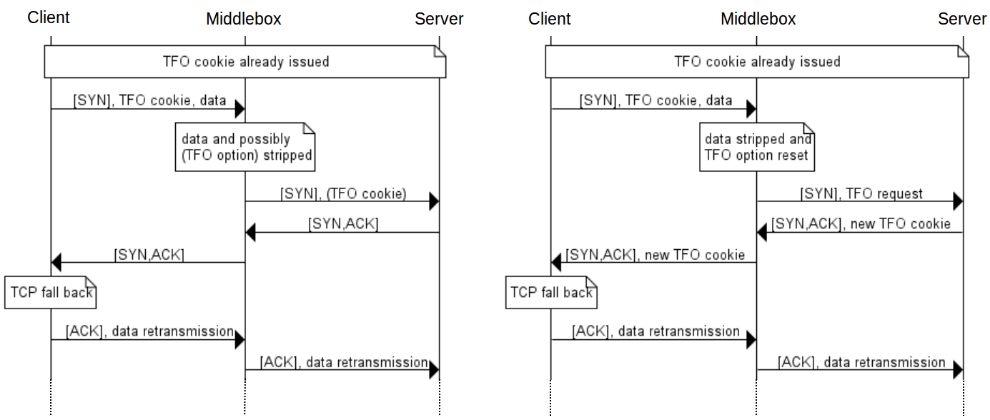

Figure 3.4: Possible sequence diagrams of the case *Data Packet Corrupted*; data (and TFO option) stripped (left), data stripped and TFO option reduced (right)

3.3.3 Data Packet Lost or Ignored

The last case is when the client receives a cookie but get no response for the data packet. Here, the client sends the [SYN] packet with the cookie and the data but does not receive any response. In that case, the client sends a new [SYN] packet without cookie or data, initializing the fall back to a regular TCP connection.
A reason for this case can be a middlebox that drops TFO data packets completely. Therefore, the server does never receive any data packet to respond to. Another reason could be that the server receives the packet but views the cookie as invalid or expired. This could happen if a middlebox changes the cookie and thereby invalidates it. If the cookie is viewed as expired, the TFO implementation on the server is probably false since we request and use the cookie within seconds. Again note that in this case, the TFO cookie request has already worked across all involved middleboxes, so these middleboxes only drop or change the data packets but pass the packets of the TFO cookie request.
Figure 3.5 shows possible sequence diagrams for the two scenarios.

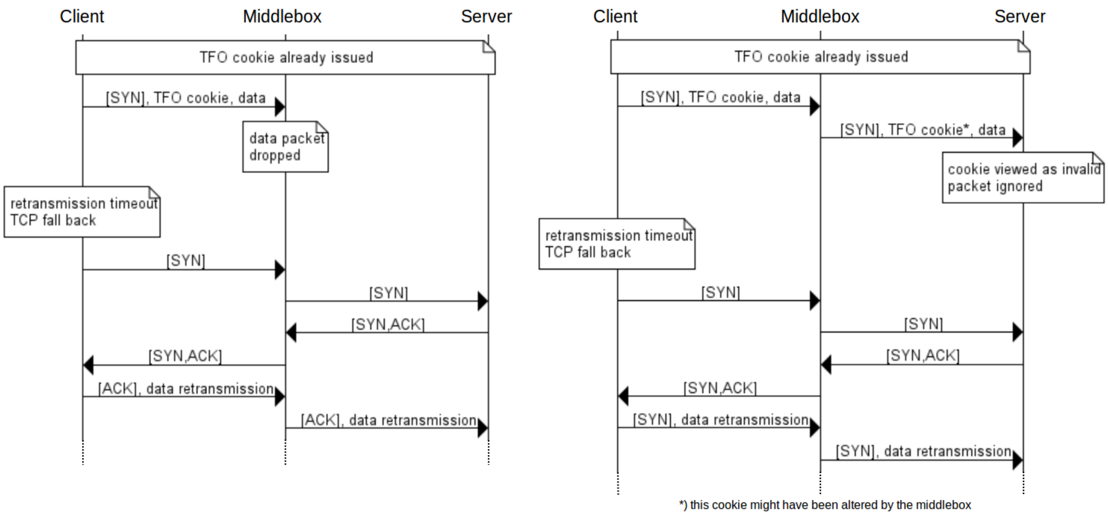

Figure 3.5: Possible sequence diagrams of the case *Data Packet Lost or Ignored*; data packet dropped by middlebox (left), data packet ignored by server (right)

3.4 Experimental TFO Option

In our work we also found out that some servers do support TFO but use and old experimental cookie. The regular TFO option is the TCP option with the identifier 34 (0x22) in the first byte. The experimental option has the identifier 254 (0xFE) for experimental in the first byte followed by the length of the option and the identifier 249, 137 (0xF9, 0x89) for TFO.[6] We saw that our kernel tried a TFO connection with the experimental option after an unsuccessful attempt with the regular one. So these hosts have already ignored a cookie request with the regular TFO option.
Upon closer inspection, we found 14 servers that used the experimental option. Four of those have also used the regular option at some point. Therefore, those four were also amongst the endpoints for the measurement. It's interesting to note that some of those hosts also worked with regular TFO options from some places (See Section 4.1.3).
Figure 3.6 shows the two possible TCP options for TFO.

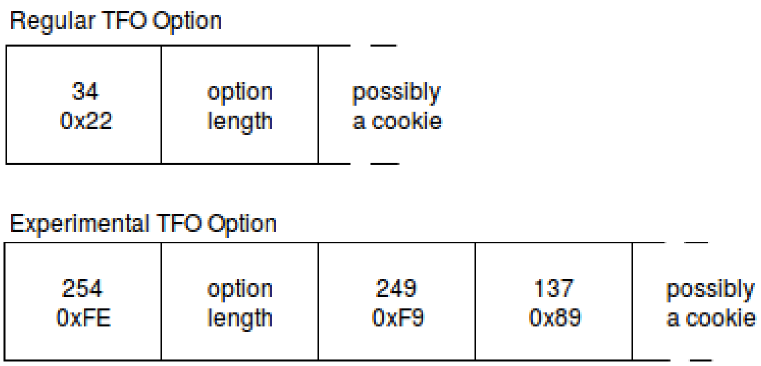

Figure 3.6: Regular and experimental TCP options for TFO

Chapter 4

Measurement

With our measurement probe we wanted to measure the internet path transparency of TFO. We are interested how many paths in the internet contain middleboxes that break TFO. For this measurement we needed a large set of path-diverse connections to try out.
The results from this measurement are analyzed with respect to different factors to give as much insight into the functioning of the TFO connections as possible.

4.1 Measurement Setup

As starting points for these connections served 7 machines that ran Pathspider. They were located in 7 different countries. As endpoints, we used TFO capable servers of websites which we took from the Alexa Top Million list. These starting and endpoints should give us as many path-diverse connections as possible.

4.1.1 Starting Points (A)

The starting points or points A in A/B testing are the machines that run Pathspider. The connections that Pathspider tries originate from here.
For this work we had a total of 7 different starting points. The first starting point was a private DSL operated by Swisscom in Switzerland. The remaining 6 were Digital Ocean servers located in Frankfurt (Germany), London (United Kingdom), Toronto (Canada), Amsterdam (Netherlands), San Francisco (United States) and Singapore (Singapore).
With these 7 starting points we were able to achieve a good path diversity having 7 different places to start connections from.

4.1.2 Access Networks

When we were experimenting with TFO and looking for the starting point in Switzerland, we saw that the access network often already breaks TFO.
The first example was the ETH network. TFO cookie requests were never successful when originating from within the ETH network. Closer inspection revealed that some firewall in the network strips the TFO option from any outgoing packet. So here TFO already breaks in the client network (See Section 3.3).
The second example was a private internet access run by Swisscom in Zurich over copper cable. Here, TFO cookie could be received without a problem, but TFO data transmissions were only successful over IPv6. IPv4 [SYN] packets containing data seemed to be blocked. The sender retransmitted a [SYN] packet without data to start a TCP handshake. This problem was resolved when we switched to a fiber optic access als run by Swisscom and changing the home router.
On another private internet access also run by Swisscom in St.Gallen over copper cable, TFO worked without any problems on IPv4 and IPv6.
This already shows that the path transparency "in the internet" is not the only problem for TFO.

TFO can already break in the clients network. But this is a problem that the client should be able to resolve by himself.

4.1.3 Endpoints (B)

For the connection endpoints or points B in A/B testing we used servers of websites. We tested the websites from the Alexa Top Million list for TFO capabilities and used the capable ones for our measurement.
The Alexa Top Million list is a list of the top million most visited websites. This gave us a large list of potential endpoints. Resolving these websites for IP addresses gave 655621 unique addresses. It is less than a million since some websites resolved to the same IP address. We removed duplicate addresses since they would not add any path diversity and distort the measurements. Amongst those 655621 addresses were 629084 IPv4 and 26537 IPv6 addresses.
For those 655621 addresses we checked whether their servers would issue a TFO cookie. This gave us the endpoints to which we then tried TFO connections. This list of TFO capable hosts contained 362 IP addresses of which 330 were IPv4 and 32 were IPv6.
These numbers show that the implementation of TFO on webhosts is still rare, with only 330 IPv4 addresses (0.052% of all IPv4 addresses), 32 IPv6 addresses (0.12% of all IPv6 addresses) and 362 addresses in general (0.055% of all addresses).
These hosts are spread over 16 countries with the majority located in the United States.
A list of all the endpoints is included in Appendix A.
Table 4.1 shows the distribution of the hosts by country.

Country	IPv4	IPv6	Total
Bulgaria	2	0	2
Canada	1	1	2
Czech Republic	1	1	2
Germany	1	0	1
Spain	7	0	7
France	5	3	8
Ireland	0	2	2
Italy	1	0	1
South Korea	1	0	1
Lituania	3	0	3
Netherlands	1	1	2
Norway	3	0	3
Romania	2	0	2
Russia	7	0	7
Singapore	1	0	1
United States	294	24	318
Total	330	32	362

Table 4.1: TFO capable hosts by country

4.1.4 Google Endpoints

Queries in the website registries for those addresses revealed that out of those 362 addresses, 304 (84.0% of all addresses) belonged to hosts affiliated to Google. These are 278 IPv4 addresses (84.2% of all IPv4 addresses) and 26 IPv6 addresses (81.2% of all IPv6 addresses). 302 of these addresses are located in the United States and 2 in Ireland. In the United states, they make up 95.0% of all addresses and in Ireland 100%.
This shows that at the moment the server side rollout of TFO is very much limited to Google. One one hand this is not surprising since Google is the developper of TFO. But shows that aside from Google, few other companies have rolled it out on their servers.

4.1.5 Experimental Option

As mentioned in Section 3.4, we also discovered 14 IPv4 hosts that did issue a cookie with the experimental TFO option. Four of those were amongst the 362 endpoints since they also issued a TFO cookie using the regular TFO option at some point. Amongst the four hosts in the main measurement, two were located in Lituania, one in the United States and one in Singapore. Amongst the remaining 10, two were located in the United States, 6 in the Netherlands, one in France and one in Australia.
Since we found out about these hosts too late, we did not have the time to update TFOSpider for including a measurement for those hosts so we only have results for the four that are amongst the regular hosts.
A list of these hosts is included in Appendix A.

4.2 Results

In our measurement on the 27th of June 2016 we checked connections from 7 starting points to 362 endpoints. That are 2534 path-diverse connections. In this Section, we analyze the results from this measurement. A complete list of the results is included in Appendix A.

4.2.1 Results by Connections

Amongst the 2534 connections were 2310 on IPv4 and 224 on IPv6. The number of connections that did not work on regular TCP were 7. These were the connections to one host with an IPv4 address that seemed to be under DDoS protection and constantly changing his address. Then, there were 127 connection where no TFO cookie could be received with 109 connections on IPv4 and 18 on IPv6. There were 31 connections where a TFO cookie was issued but the data transmission failed. All of those connections were on IPv4. And finally, there were 2369 connections that fully worked with TFO. Of those, 2163 were on IPv4 and 206 on IPv6.
These results show that more than 9 in 10 TFO connections were successful. Based on these numbers, we conclud that in an over all perspective, the path transparency for TFO is quite good.
Table 4.2 shows the numbers of the different cases.

	Total Connections	TCP Connection Failed	No TFO Cookie Recieved	Data Transmission Failed	TFO Worked
Total	2534	7 (0.3%)	127 (5.0%)	31 (1.3%)	2369 (93.5%)
IPv4	2310	7 (0.3%)	109 (4.7%)	31 (1.2%)	2163 (93.6%)
IPv6	224	0	18 (8.0%)	0	206 (92.0%)

Table 4.2: Results by Connections

4.2.2 Results by Starting Points

For our measurement we had 7 starting points where the attempted connections originated from. These were located in St. Gallen (Switzerland, CH), Frankfurt (Germany, DE), London (United Kingdom, UK), Toronto (Canada, CA), Amsterdam (Netherlands, NL), San Francisco (United States, US) and Singapore (Singapore, SG). From all these points we tried connections to all the 362 endpoints. From all the starting points, between 92.8 and 94.8% of the TFO connections worked. For IPv4 these numbers were between 93.0 and 94.2% and for IPv6 between 90.6 and 100%.
These numbers show that from none of our starting points, TFO connection work considerably better than from the others. Therefore, we conclude that the path transparency for TFO is more or less equal for all starting points. Along with the hight percentages of successful TFO connections we also conclude that the path transparency is quite good for all starting points.
Table 4.3 shows the numbers of working TFO connections by starting point.

	Total Connections	TCP Connection Failed	No TFO Cookie Recieved	Data Transmission Failed	TFO Worked
CH	362	1 (0.3%)	2 (0.6%)	16 (4.4%)	343 (94.8%)
... IPv4	330	1 (0.3%)	2 (0.6%)	16 (4.8%)	311 (94.2%)
... IPv6	32	0	0	0	32 (100%)
DE	362	1 (0.3%)	22 (6.1%)	1 (0.3%)	338 (93.4%)
... IPv4	330	1 (0.3%)	19 (5.8%)	1 (0.3%)	309 (93.6%)
... IPv6	32	0	3 (9.4%)	0	29 (90.6%)
NL	362	1 (0.3%)	22 (6.1%)	1 (0.3%)	338 (93.4%)
... IPv4	330	1 (0.3%)	19 (5.8%)	1 (0.3%)	309 (93.6%)
... IPv6	32	0	3 (9.4%)	0	29 (90.6%)
UK	362	1 (0.3%)	15 (4.1%)	9 (2.5%)	337 (93.1%)
... IPv4	330	1 (0.3%)	12 (3.6%)	9 (2.7%)	308 (93.3%)
... IPv6	32	0	3 (9.4%)	0	29 (90.6%)
US	362	1 (0.3%)	22 (6.1%)	0	339 (93.6%)
... IPv4	330	1 (0.3%)	19 (5.8%)	0	310 (94.0%)
... IPv6	32	0	3 (9.4%)	0	29 (90.6%)
CA	362	1 (0.3%)	22 (6.1%)	1 (0.3%)	338 (93.4%)
... IPv4	330	1 (0.3%)	19 (5.8%)	1 (0.3%)	309 (93.6%)
... IPv6	32	0	3 (9.4%)	0	29 (90.6%)
SG	362	1 (0.3%)	22 (6.1%)	3 (0.8%)	336 (92.8%)
... IPv4	330	1 (0.3%)	19 (5.8%)	3 (0.9%)	307 (93.0%)
... IPv6	32	0	3 (9.4%)	0	29 (90.6%)

Table 4.3: Results by Starting Points

4.2.3 Results by Endpoints

In the measurement we tried connections to 362 webserver. These were the endpoints of the connections. Amongst these endpoints were 330 IPv4 and 32 IPv6 hosts. The connections to 335 of these endpoints (92.5%) worked for TFO from all starting points. Of those 335 addresses, 306 were IPv4 (92.7% of all IPv4 addresses) and 29 IPv6 (90.6% of all IPv6 addresses). One of the IPv4 addresses that work from all entry points is amongst the 14 addresses that issued TFO cookies with the experimental option at some point. Furthermore, one IPv4 address did not work with regular TCP for all starting points. This hosts seemed to be under DDoS protection at the time.

For the remaining 23 IPv4 and 3 IPv6 addresses we observed 7 other cases.

- 8 IPv4 addresses: CH receives cookie ; DE, NL, UK, US, CA, SG receive no cookie
- 7 IPv4 addresses: CH, UK receive cookie ; DE, NL, US, CA, SG receive no cookie
- 4 IPv4 and 3 IPv6 addresses: CH works ; DE, NL, UK, US, CA, SG receive no cookie
- 1 IPv4 address*: CH, DE, NL, UK, US, CA work ; SG receives cookie
- 1 IPv4 address: US, CA work ; CH, DE, NL, UK, SG receive cookie
- 1 IPv4 address*: DE, NL, UK, US, CA, SG work ; CH recieves no cookie
- 1 IPv4 address*: DE, NL, US work ; UK, CA, SG receive cookie ; CH receives no cookie

*) These three addresses are amongst the 14 addresses that issued TFO cookies with an experimental option at some point.

These numbers show that most hosts are reachable with TFO from all starting points. Even more so if we note that some of the hosts were not reachable from anywhere which could be caused by a problem in the servers network and not "in the internet". In this case, the servers administration should be able to solve this problem by themselfes. From this data we conclude that for most of TFO supporting hosts, TFO works from all starting points.

4.2.4 Results by Countries

In our measurement, the connections had starting points in 7 countries and endpoints in 16. Amongst the 7 countries of the starting points we saw in Section 4.2.2 that for all of them between 90 and 95% of the connections were successful. In this section we want to look at the situation for the countries of the endpoints.
The first thing to notice is that 318 of the 362 addresses are located in the United States. This is probably due to Google being the developper of TFO and an American company. Since the rollout of TFO is mostly limited to Google, a lot of TFO supporting servers are in the United States. Amongst the 318 US hosts, 316 could be reached with TFO from all starting points. For the other two hosts, the success of TFO depended on the starting point.
For the other countries we saw one group where all TFO connections worked from all starting points as well. These countries are Bulgaria, the Czech Republic, Ireland, Italy, South Korea, the Netherlands and Norway.
Hosts in Canada and Germany were only reachable from Switzerland. For all other starting points we did not even receive a cookie. With Spanish and Romanian hosts TFO connections did not work from any starting points. They only issued cookies to some.
The results for France, Lituania, Russia and Singapore do not allow a general statement on whether connections to their hosts work with TFO. These countries have hosts that work from all starting points, hosts that work from none and hosts with dependency on the starting point.
From these numbers we conclude that hosts in some countries work better than in others. A reason for this could be the internet infrastructure in the respective countries.
Table 4.4 shows the number of connections that worked with different levels of success. The first number tells the number of connections that worked with TFO, the second number gives the number of connections where only a cookie was issued but the data transmission failed and the last number tells the number of connections where no cookie was issued at all.

	CH	DE	NL	UK	US	CA	SG
Bulgaria	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
Canada	2/0/0	0/0/2	0/0/2	0/0/2	0/0/2	0/0/2	0/0/2
Czech Republic	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
Germany	1/0/0	0/0/1	0/0/1	0/0/1	0/0/1	0/0/1	0/0/1
Spain	0/7/0	0/0/7	0/0/7	0/0/7	0/0/7	0/0/7	0/0/7
France	7/1/0	3/0/5	3/0/5	3/0/5	3/0/5	3/0/5	3/0/5
Ireland	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
Italy	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
South Korea	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
Lituania	2/1/0	2/1/0	2/1/0	2/1/0	3/0/0	3/0/0	1/2/0
Netherlands	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
Norway	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0
Romania	0/2/0	0/0/2	0/0/2	0/2/0	0/0/2	0/0/2	0/0/2
Russia	2/4/1	2/0/5	2/0/5	2/4/1	2/0/5	2/0/5	2/0/5
Singapore	0/0/1	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
United States	316/1/1	317/0/1	317/0/1	316/2/0	317/0/1	316/1/1	316/1/1

Table 4.4: Results by Countries: [Number of Connections Working with TFO/Number of Connections with Only Cookie Issued/Number of Connections without Cookie Issued]; Green - All Connections Working, Yellow - Some Connections Working, Red - No Connections Working

4.2.5 Results by IP Versions

In our measurement, we tried connections on IPv4 and IPv6. We had 330 IPv4 and 32 IPv6 endpoints to which we tried connections from our 7 starting points.
With respect to the individual connections, 2163 out of 2310 (93.6%) IPv4 and 206 out of 224 (92.0%) IPv6 connections were successful with TFO.
Also with respect to the different starting points, the share of successful TFO connections was

between 90 and 95% for both IPv4 and IPv6. The only exception was Switzerland with 100% of IPv6 connections being successful.
And finally with respect to the different endpoints, 306 out of 330 (92.7%) IPv4 endpoints and 29 out of 32 (90.6%) IPv6 endpoints were successfully connected to with TFO.
It is worth mentioning that only for some IPv4 connections a TFO cookie was issued but the data transmission failed. This case was not observed with IPv6 connections. Also, there were only IPv4 addresses that issued TFO cookies with the experimental option at some point.
From these numbers we conclude that the path transparency for TFO does not heavily depend on the IP version that is used. Both IPv4 and IPv6 connections were comparably successful. Only the rollout of TFO on IPv6 hosts is slightly more advanced with 0.12% of IPv6 hosts supporting TFO as compared to 0.053% of IPv4 hosts.

4.2.6 Results for Google

As mentioned in Section 4.1, 278 IPv4 (84.2% of all IPv4 addresses) and 26 IPv6 (81.3% of all IPv6 addresses) addresses are affiliated to Google. That are 304 out of 362 (84.0%) addresses over all. In our measurement we saw that those 304 addresses all worked from all starting points.
On the other hand, the percentage of successful connections to not Google-affiliated hosts is less than 60%. Of all connections to not Google-affiliated hosts, 241 out of 406 (59.4%) were successful with 217 out of 364 (59.6%) IPv4 and 24 out of 42 (57.1%) IPv6 connections.
Out of these 58 hosts only 31 (53.4%) were reachable with TFO from all starting points. Amongst those were 28 out of 52 (52.8%) IPv4 and 3 out of 6 (50.0%) IPv6 hosts.
This shows that while Google has rolled out TFO successfully, other companies have a more limited rollout and also have more problems with the path transparency for TFO.

Chapter 5

Summary

In this work we wanted to measure the internet path transparency for the transport protocol extension TCP Fast Open. TFO was developped by Google and aims to reduce latency by sending data in the initial [SYN] packet of a TCP handshake. The path transparency determines how many middleboxes break TFO by for example dropping packets or stripping TCP options. The path transparency therefore gives a clue on how well TFO works in the current internet infrastructure.
We measured the path transparency with A/B testing. This means that we tried to establish a connection from starting point A to endpoint B, first with regular TCP and then with TFO. The first connection determines whether there is a path from A to B and the second connection checks whether TFO works on that path.
For our measurement we used 7 starting points in 7 different countries and tried connections to 362 webhosts in 16 different countries. The endpoint addresses contained both IPv4 and IPv6 addresses. These 2534 paths gave us the maximum of available path diversity.
We then analyzed the results from these connections to some conclusion on how well TFO packets can pass through the internet.

5.1 Conclusions

While setting up the measurement we saw that TFO can not only fail "in the internet" but also in the clients and the servers network. Another thing we saw was that the rollout of TFO is quite limited with less than 1 in 1000 hosts from the Alexa One Million list supporting TFO.
But in the measurement we saw that for the supporting servers TFO does work quite well with more than 9 in 10 connections being successful. Also, more than 9 in 10 target servers were reachable from all starting point. We also saw that the success of a TFO connection rarely depends on its starting point. But we saw that the location of the target server might have an influence since we discovered that servers in certain countries worked better than others.
We also saw that Google as the developper of TFO is also the company that runs most of the TFO capable servers. And their servers were successfully connected to from all starting points. This shows that Google has rolled out TFO on its servers very successfully. For the remaining not Google-affiliated servers, this means that the percentage of successful TFO connections to them drops below 60%.
Finally, we saw that TFO works comparably on IPv4 and IPv6 so the path transparency is comparable for both IP versions.

5.2 Outlook

In the future, a more thorough measurement could be performed with more starting points and/or more endpoints. More starting points could be achieved by organizing more machines in different locations. More endpoints are only accessable if TFO is rolled out on more servers.
Also, further work could go into more details about hosts that still use the experimental TFO

option.
Another possibility would be to determine the packets route along with the measurement. This could be done by running *traceroute* alongside the measurement for TFO. By combining the results, one could try to pinpoint the nodes in the internet that break TFO.
Also, further work could implement tests for other transport protocol extensions like for example Accurate Explicit Congestion Notification or Multipath TCP.

Appendix A

Measurement Data

This appendix contains the resulting data of our measurement.

A.1 TFO Fully Working

The 335 hosts in this list are those that were reachable with TFO from all entry points. "rank" denotes the rank of the host in the Alexa One Million list. "google" denotes whether the host is affiliated to Google. A star denotes the hosts that have used the experimental TFO option at some point.

rank	host	ip	country	google
1	www.google.com	2a00:1450:400e:801::2004	US	1
1	www.google.com	216.58.212.164	US	1
2	www.youtube.com	2a00:1450:400e:801::200e	US	1
2	www.youtube.com	216.58.212.174	US	1
11	www.google.co.in	2a00:1450:400e:801::2003	US	1
11	www.google.co.in	216.58.212.163	US	1
43	www.blogspot.com	2a00:1450:400e:802::2009	US	1
43	www.blogspot.com	216.58.212.201	US	1
84	www.googleusercontent.com	2a00:1450:400e:802::2001	US	1
84	www.googleusercontent.com	216.58.212.193	US	1
270	www.google.cn	2a00:1450:4013:c01::a0	US	1
270	www.google.cn	74.125.136.160	US	1
782	www.googleapis.com	2a00:1450:4013:c00::5f	US	1
782	www.googleapis.com	74.125.136.95	US	1
908	www.humblebundle.com	74.125.34.32	US	1
996	www.khanacademy.org	72.14.249.132	US	1
1126	www.googleadservices.com	216.58.212.130	US	1
1140	dudeplaygames.com	78.140.190.67	US	0
1158	www.evite.com	74.125.34.176	US	1
1226	digitalworldads.com	78.140.190.102	US	0
1518	www.servingclks.com	130.211.9.122	US	1
1599	mobilepinbox.com	78.140.190.71	US	0
1661	www.epochtimes.com	107.178.245.188	US	1
1788	www.lindaikejisblog.com	2a00:1450:4013:c01::79	US	1
1788	www.lindaikejisblog.com	74.125.136.121	US	1
2436	abema.tv	107.178.244.89	US	1
2465	www.withgoogle.com	2a00:1450:400e:801::2011	US	1
2465	www.withgoogle.com	216.58.212.241	US	1
2559	www.express.com.pk	107.178.244.221	US	1
3576	www.gmail.com	2a00:1450:400e:801::2005	US	1
3576	www.gmail.com	216.58.212.165	US	1
3630	www.googlegroups.com	2a00:1450:4013:c01::8b	US	1
3630	www.googlegroups.com	74.125.136.139	US	1
4156	www.virustotal.com	74.125.34.46	US	1
4486	www.media.wix.com	107.178.254.135	US	1
4784	www.dcard.tw	130.211.12.255	US	1
4979	www.googleblog.com	2a00:1450:400e:801::2001	US	1
4979	www.googleblog.com	216.58.212.225	US	1
5436	w20sweeplnd.com	78.140.190.109	US	0
5831	www.dopr.net	72.14.248.124	US	1
7629	www.admob.com	2a00:1450:4013:c01::64	US	1
7629	www.admob.com	74.125.136.100	US	1
7742	www.lezhin.com	72.14.246.137	US	1
8278	d35classicapp.com	78.140.190.78	US	0
8444	www.waze.com	130.211.9.172	US	1
8537	www.chromium.org	216.239.32.27	US	1
8744	www.googleplex.com	2a00:1450:4013:c01::81	US	1
8744	www.googleplex.com	74.125.136.129	US	1

9958	www.freenom.com	130.211.15.224	US	1	
10113	number1hazard.com	78.140.190.76	US	0	
10852	www.ingress.com	72.14.249.180	US	1	
11093	www.totalvpn.com	130.211.5.134	US	1	
11156	www.instapage.com	107.178.243.80	US	1	
11709	www.googlecode.com	2a00:1450:4013:c01::52	US	1	
11709	www.googlecode.com	74.125.136.82	US	1	
12858	www.corotos.com.do	107.178.245.189	US	1	
13563	www.mysupermarket.co.uk	107.178.248.114	US	1	
15294	www.smarterweblife.com	107.178.240.45	US	1	
15507	www.mantan-web.jp	107.178.252.115	US	1	
15959	www.jeeadv.ac.in	107.178.247.41	US	1	
16115	www.balatarin.com	107.178.241.59	US	1	
16530	gamez4tops.com	78.140.190.69	US	0	
16704	www.offersupply.com	130.211.11.108	US	1	
17259	www.ehost.com	107.178.253.18	US	1	
17545	www.uberconference.com	72.14.249.61	US	1	
17776	www.vertalen.nu	107.178.243.255	US	1	
20594	www.justapinch.com	130.211.10.17	US	1	
21201	www.workana.com	107.178.252.201	US	1	
22250	www.porch.com	107.178.243.140	US	1	
24147	www.conjur.com.br	107.178.251.137	US	1	
24193	ato.mx	107.178.247.200	US	1	
24465	www.submenow.com	209.141.38.179	US	0	
24527	www.pigeonsandplanes.com	130.211.12.86	US	1	
24582	www.bq.com	107.178.246.4	US	1	
25392	www.googletagmanager.com	2a00:1450:400e:802::2008	US	1	
25392	www.googletagmanager.com	216.58.212.168	US	1	
26397	www.meethue.com	74.125.34.101	US	1	
26586	www.dariknews.bg	94.156.102.47	BG	0	
27076	www.swap.com	74.125.34.8	US	1	
27804	www.johnnys-net.jp	72.14.248.39	US	1	
28633	shop3stockoff.com	78.140.190.75	US	0	
29570	www.pdfconverterhq.appspot.com	2a00:1450:400e:803::2011	US	1	
29570	www.pdfconverterhq.appspot.com	173.194.65.141	US	1	
29819	www.livehouse.in	107.178.253.92	US	1	
30855	www.wedmegood.com	107.178.240.38	US	1	
30897	www.sonicdrivein.com	72.14.246.104	US	1	
31540	www.incognitovpn.com	107.178.245.151	US	1	
32730	www.mypcbackup.com	107.178.252.163	US	1	
35333	www.gqueues.com	72.14.248.247	US	1	
37515	www.focus.ua	107.178.241.206	US	1	
38819	www.blogspot.com.cy	2a00:1450:4013:c01::bf	US	1	
38819	www.blogspot.com.cy	74.125.136.191	US	1	
39140	www.frip.com	107.178.252.19	US	1	
39463	www.raterhub.com	72.14.249.104	US	1	
39712	www.juventus.com	130.211.6.243	US	1	
39950	www.tadpoles.com	72.14.249.45	US	1	
40478	www.twpro.jp	107.178.241.124	US	1	
40655	www.2ality.com	216.239.32.21	US	1	
40714	www.eventshigh.com	107.178.245.125	US	1	
41465	www.fanpagekarma.com	72.14.248.225	US	1	
44821	www.justcloud.com	107.178.253.242	US	1	
45185	www.saferweb.com	107.178.247.71	US	1	
45186	www.kinsta.com	107.178.240.207	US	1	
45359	www.rogerhub.com	107.178.242.73	US	1	
46560	www.episodeinteractive.com	72.14.247.145	US	1	
46635	www.instantdomainsearch.com	130.211.25.142	US	1	
47903	www.thelayoff.com	72.14.247.238	US	1	
48521	www.staatsloterij.nl	74.125.34.110	US	1	
48840	www.padpadblog.com	216.239.38.21	US	1	
49437	www.snapengage.com	72.14.249.2	US	1	
50667	www.googlesciencefair.com	74.125.34.43	US	1	
52193	www.lovemondays.com.br	130.211.9.55	US	1	
54164	www.mibqyyo.com	107.178.247.220	US	1	
56195	www.tiltbrush.com	216.239.32.29	US	1	
57098	www.fullstory.com	72.14.248.62	US	1	
57676	www.photographyisnotacrime.com	107.178.246.8	US	1	
58652	www.pixoto.com	74.125.34.19	US	1	
58730	www.exchange4media.com	64.32.5.147	US	0	
60584	www.prepladder.com	130.211.7.54	US	1	
60943	www.balagannna.com	216.239.34.21	US	1	
64698	www.inbox.lt	195.216.236.10	LV	0	*
66709	www.netotrade.com	130.211.26.124	US	1	
67115	takusuta.com	107.178.243.152	US	1	
68797	www.twreporter.org	107.178.250.210	US	1	
68816	www.non.li	107.178.244.41	US	1	
69840	www.stickermanager.com	107.178.247.49	US	1	
70491	www.dereferer.me	2001:19f0:6800:13e4:5400:ff:fe20:89e8	FR	0	
70491	www.dereferer.me	62.210.220.53	FR	0	
73827	www.elsoldemexico.com.mx	107.178.246.250	US	1	
75521	www.unionbank.co.il	107.178.244.151	US	1	
77607	www.kreativeideer.com	185.91.65.142	NO	0	
77771	www.whoscall.com	130.211.11.41	US	1	
77928	www.talentclue.com	130.211.5.242	US	1	
80243	www.franchisehelp.com	72.14.247.209	US	1	
80662	www.farmigo.com	72.14.249.46	US	1	

82702	www.ilbuongiorno.it	94.23.67.102	IT	0
83299	www.cocina-casera.com	107.178.252.240	US	1
84504	www.dartlang.org	72.14.249.27	US	1
86793	www.2016cbseresultsnic.in	216.239.36.21	US	1
87537	www.mapio.net	130.211.6.80	US	1
90190	www.matterhackers.com	72.14.246.64	US	1
90221	www.keyingredient.com	72.14.249.9	US	1
90249	www.kumpulansoaltest.blogspot.co.id	2a00:1450:400e:803::2001	US	1
90249	www.kumpulansoaltest.blogspot.co.id	173.194.65.132	US	1
91846	www.dialpad.com	72.14.247.98	US	1
92948	www.odrive.com	72.14.247.154	US	1
93493	www.small-improvements.com	72.14.247.197	US	1
93544	www.fancyhands.com	74.125.34.100	US	1
93641	www.radioradicale.it	107.178.246.30	US	1
95044	www.junyiacademy.org	72.14.248.120	US	1
95941	www.shopifiedapp.com	130.211.7.175	US	1
100594	www.thinkwave.com	72.14.248.245	US	1
101082	www.getyarn.io	130.211.5.232	US	1
101562	www.cart-checkout.com	72.14.248.11	US	1
104763	www.aladdin.ie	72.14.246.48	US	1
106190	www.mysupermarket.co.il	130.211.6.189	US	1
109999	www.filewatcher.com	5.135.181.170	FR	0
113767	www.loudr.fm	72.14.248.104	US	1
116929	www.teleport.org	107.178.247.63	US	1
117569	www.iaa.gov.il	130.211.12.52	US	1
118644	www.solestruck.com	72.14.249.7	US	1
123640	www.jobright.com	107.178.246.253	US	1
126966	www.ambev.com.br	130.211.16.61	US	1
128925	www.go1buy1.com	72.14.249.21	US	1
130996	www.porterfinance.com	130.211.12.165	US	1
133084	www.terminals.io	130.211.4.46	US	1
135177	www.codeskulptor.org	2a00:1450:4013:c01::80	US	1
135177	www.codeskulptor.org	74.125.136.128	US	1
137161	www.s-e-s.co	198.251.89.79	US	0
144801	www.careeravenue.com	107.178.248.36	US	1
148680	www.drrr.com	130.211.4.165	US	1
156948	www.ifoodie.tw	107.178.252.150	US	1
157090	www.vetconnectplus.com	74.125.34.51	US	1
164792	www.hundredrooms.com	130.211.9.153	US	1
168941	www.zipcloud.com	107.178.240.66	US	1
172616	www.adbuddiz.com	72.14.247.229	US	1
175680	www.fantia.jp	130.211.10.110	US	1
178093	www.crozdesk.com	130.211.5.215	US	1
178394	www.aporta.org.mx	130.211.13.223	US	1
185692	www.mapadaprova.com.br	107.178.254.93	US	1
186536	www.nextivafax.com	130.211.15.213	US	1
186856	www.adyoulike.com	74.125.34.106	US	1
188210	www.jjgames.com	107.178.241.12	US	1
188966	www.matchclub.dk	130.211.4.148	US	1
192305	www.koukni.cz	2a00:1ed0:1:500::c	CZ	0
192305	www.koukni.cz	93.185.104.72	CZ	0
194897	www.sitenizolsun.com	130.211.9.176	US	1
195013	www.googlegoro.com	74.125.34.25	US	1
196117	www.itviet360.com	74.125.130.121	US	1
198653	www.lmfdb.org	130.211.24.123	US	1
199941	www.showpad.com	107.178.247.254	US	1
203662	www.ebl.com.bd	107.167.27.69	US	0
205477	www.mystudentdashboard.com	107.178.255.210	US	1
212060	www.cubeslam.com	74.125.34.151	US	1
213241	www.symphonytools.com	72.14.246.163	US	1
213288	www.pinsforme.com	107.178.244.218	US	1
214361	www.blockly-games.appspot.com	2a00:1450:400c:c01::8d	US	1
214361	www.blockly-games.appspot.com	74.125.195.141	US	1
216259	www.sublinet.com	107.178.245.35	US	1
220816	www.siteblog.com	107.178.240.12	US	1
221327	www.bablic.com	107.178.241.14	US	1
226936	www.ideahost.com	107.178.243.24	US	1
232903	www.battlefieldbr.com	74.125.196.121	US	1
233668	www.xcams.de	64.15.124.119	US	0
234867	www.the-alba.com	188.227.74.41	RU	0
237051	www.formy-i-blanki.ru	74.125.29.121	US	1
238892	www.etsyonsale.com	72.14.246.82	US	1
239216	www.fromtheboxoffice.com	107.178.251.43	US	1
246653	www.boxc.com	107.178.250.251	US	1
249901	www.roomle.com	107.178.255.238	US	1
256087	www.dersteknik.com	64.233.166.121	US	1
261468	www.phantomjscloud.com	107.178.252.155	US	1
266990	www.social-lunch.jp	72.14.248.254	US	1
267051	www.miraie.me	107.178.247.223	US	1
278317	www.metacdn.com	74.125.34.50	US	1
279038	www.maxiforex.ru	130.211.4.235	US	1
281144	www.kindara.com	74.125.34.137	US	1
292475	www.webinmerkezi.com	173.194.65.121	US	1
295026	www.gudog.com	130.211.16.122	US	1
298881	www.joblift.de	130.211.8.209	US	1
303938	www.signal.bg	94.156.102.56	BG	0
304393	www.hebban.nl	72.14.249.59	US	1

308485	www.bqreaders.com	107.178.243.211	US	1
312244	www.sende.rs	130.211.6.54	US	1
318158	ussto.com	130.211.8.135	US	1
318492	www.huge-options.com	130.211.4.80	US	1
324010	www.pins.fi	107.178.249.96	US	1
326927	www.food.co.il	72.14.249.52	US	1
328434	www.sener.es	107.178.255.169	US	1
329806	www.lidota.vn	173.194.72.121	US	1
331944	www.slownet.ne.jp	72.14.249.135	US	1
333368	www.cmsantagostino.it	130.211.6.173	US	1
340178	www.google-melange.com	74.125.34.4	US	1
349513	www.hostclear.com	107.178.244.42	US	1
351334	www.hudora.de	72.14.249.116	US	1
356999	www.videopixie.com	74.125.34.60	US	1
358366	www.hitbiju.ru	95.213.164.227	RU	0
366666	www.cocycles.com	130.211.6.110	US	1
371357	www.grifo210.com	107.178.242.219	US	1
371619	www.constituteproject.org	72.14.246.113	US	1
376853	www.pancake.vn	130.211.6.117	US	1
386275	www.alvexo.com	107.178.246.64	US	1
389897	www.tropicaltrade.com	130.211.11.224	US	1
390264	www.buysellpassport.com	107.178.253.32	US	1
392142	www.toyga.com	130.211.6.86	US	1
395032	www.addressreport.com	72.14.248.98	US	1
403094	www.siersaglik.com	130.211.8.12	US	1
403499	www.cincymls.net	130.211.8.106	US	1
413770	www.projectnoah.org	72.14.246.152	US	1
415409	allabout-japan.com	107.178.255.23	US	1
418715	www.leonax.net	107.178.254.89	US	1
420425	www.longine.jp	72.14.246.22	US	1
428367	widevine.com	2a00:1450:400e:803::200e	IE	1
434024	www.beautyscenery.com	64.233.183.121	US	1
434305	www.censys.io	72.14.246.220	US	1
435030	www.bbfdirect.com	72.14.249.47	US	1
445105	www.twentyfourseven.co.il	107.178.253.104	US	1
474884	shipment.co	107.178.240.249	US	1
475664	www.dashub.com	130.211.5.74	US	1
476814	www.hipotels.com	130.211.12.118	US	1
485234	notifuse.com	130.211.15.163	US	1
488258	www.kastatic.org	72.14.247.4	US	1
490414	www.gorgias.io	107.178.242.172	US	1
505275	www.smartamp.com	72.14.247.198	US	1
505616	www.point.ml	107.178.252.41	US	1
514443	www.tinytap.it	107.178.243.43	US	1
542554	www.docketalarm.com	72.14.246.24	US	1
570197	www.okthere.com	72.14.249.95	US	1
570887	www.prosent.no	185.91.65.52	NO	0
580502	www.my.ga	107.178.244.196	US	1
582504	www.weotta.com	130.211.6.172	US	1
583331	www.uprisingtech.com	72.14.248.157	US	1
584252	www.santaclarasystems.com	72.14.248.184	US	1
585250	www.lierac.fr	107.178.245.155	US	1
590475	www.gene-media.com	107.178.248.4	US	1
593708	www.weblogzwolle.nl	2a02:2770:6:0:216:3eff:fe2a:9b33	NL	0
593708	www.weblogzwolle.nl	37.252.123.204	NL	0
609406	www.sofiaapartament.com	107.178.255.126	US	1
614503	www.kjvvv.kr	211.34.105.42	KR	0
615313	www.realmassive.com	72.14.247.126	US	1
616503	www.dot.cf	107.178.245.235	US	1
616808	www.keyhole.com	72.14.249.189	US	1
624022	www.kesuskim.com	2001:4860:4802:32::15	US	1
628281	www.grandpad.net	72.14.249.82	US	1
630962	www.blossom.io	74.125.34.52	US	1
632624	www.auntbertha.com	72.14.249.163	US	1
639088	www.zeetings.com	72.14.247.61	US	1
653070	www.thedailyaztec.com	208.98.43.109	US	0
653669	www.pexe.so	107.178.248.247	US	1
667435	www.googletagservices.com	2a00:1450:400e:800::2002	US	1
667435	www.googletagservices.com	216.58.212.226	US	1
668093	www.recaptcha.net	2a00:1450:401b:801::2004	US	1
668093	www.recaptcha.net	216.58.209.68	US	1
673310	www.kubity.com	72.14.247.39	US	1
677901	www.phyto.fr	130.211.7.180	US	1
678911	www.essayassay.com	72.14.247.2	US	1
680924	www.mdais.org	130.211.15.44	US	1
684774	www.rinkak.com	74.125.34.226	US	1
686124	www.rgames.jp	107.178.241.131	US	1
700159	www.beatstage.com	130.211.20.201	US	1
701858	www.yenitrt.com	130.211.6.222	US	1
704910	www.profissa.net	107.178.252.55	US	1
706043	www.pulse.me	74.125.34.28	US	1
716601	www.noracomunicacao.com.br	64.233.186.121	US	1
717215	www.arrow-arrow.com	130.211.7.224	US	1
718994	www.wondermall.com	130.211.10.85	US	1
772902	www.arbor.io	107.178.243.32	US	1
774475	www.keepy.me	130.211.10.57	US	1
776063	www.sweetprotection.com	185.91.65.141	NO	0

776491	www.gemr.com	107.178.241.221	US	1
787457	www.traderoo.com.au	72.14.249.217	US	1
796116	www.sbrick.com	107.178.253.203	US	1
804593	www.wemood.co	130.211.12.131	US	1
825487	www.omegaoptions.com	107.178.251.156	US	1
827190	www.ad4screen.com	130.211.7.4	US	1
834072	www.porijazz.fi	107.178.242.239	US	1
836496	www.globaltechpost.com	107.178.253.254	US	1
842037	www.disneytickets.co.uk	107.178.253.145	US	1
860982	www.ianlewis.org	107.178.246.40	US	1
874941	www.freeseoblog.ru	173.194.71.121	US	1
883495	www.vetconnectplus.fr	72.14.246.176	US	1
884891	www.micomercialtoddy.com	130.211.4.116	US	1
888329	anonanalytics.com	2a00:1450:400c:c0a::79	IE	1
909088	www.ad-2.me	107.178.241.54	US	1
909964	www.spacial.jp	107.178.250.190	US	1
919061	www.oulala.com	107.178.244.148	US	1
921416	www.zjogos.com.br	130.211.5.27	US	1
961335	www.edgbaston.com	107.178.254.198	US	1
964466	www.gdog168.com	107.178.254.17	US	1
965414	www.lenstag.com	72.14.246.21	US	1
978374	www.tiebow-tie.com	74.125.28.121	US	1
983780	www.mizancudito.com	74.125.21.121	US	1
987011	www.fibrenoire.ca	107.178.254.112	US	1
988944	www.looklet.com	130.211.4.11	US	1
990470	www.haychoigame.com	74.125.71.121	US	1

A.2 TCP Not Working

This is the host that was not reachable with TCP from any starting point.

rank	host	ip	country	google
690653	www.almadia.pl	186.2.165.154	RU	0

A.3 TFO Partially Working

This list shows the remaining 26 hosts. "ch_tfo", "de_tfo", "nl_tfo", "uk_tfo", "us_tfo", "ca_tfo" and "sg_tfo" show the status of the TFO connection coming from Switzerland, Germany, the Netherlands, the United Kingdom, the United States, Canada and Singapore. A "0" denotes that no TFO cookie was issued, a "1" denotes that a cookie was issued but the data transmission failed and "2" denotes success of the TFO connection. A star denotes the hosts that have used the experimental TFO option at some point.

rank	host	ip	country	google	ch_tfo	de_tfo	nl_tfo	uk_tfo	us_tfo	ca_tfo	sg_tfo	
24857	www.mapfre.es	195.53.217.36	ES	0	1	0	0	0	0	0	0	
29608	www.macif.fr	195.25.232.194	FR	0	1	0	0	0	0	0	0	
45783	www.mail.ee	195.216.236.230	LV	0	2	2	2	2	2	2	1	*
66237	www.kyivpost.com	2001:41d0:d:1662::1	FR	0	2	0	0	0	0	0	0	
77058	www.tigerrr.com	2001:41d0:b:6c8::1	FR	0	2	0	0	0	0	0	0	
77058	www.tigerrr.com	178.33.126.200	FR	0	2	0	0	0	0	0	0	
66237	www.kyivpost.com	151.80.102.98	FR	0	2	0	0	0	0	0	0	
95474	www.verti.es	212.170.33.117	ES	0	1	0	0	0	0	0	0	
93018	www.fundacionmapfre.org	195.235.248.195	ES	0	1	0	0	0	0	0	0	
124264	www.mapfre.com	195.53.217.193	ES	0	1	0	0	0	0	0	0	
186822	www.jubilacionypension.com	195.235.248.34	ES	0	1	0	0	0	0	0	0	
198617	www.mapfretecuidamos.com	195.235.248.46	ES	0	1	0	0	0	0	0	0	
268922	www.teamskeetimages.com	64.210.135.72	US	0	0	2	2	1	2	1	1	*
282993	www.imoney.ph	128.199.73.182	SG	0	0	2	2	2	2	2	2	*
378667	www.magenmagic.com	148.251.46.105	DE	0	2	0	0	0	0	0	0	
422610	www.20speed.co	186.2.165.170	RU	0	1	0	0	1	0	0	0	
614808	www.sohomod.com	2607:5300:60:6d2c::1	CA	0	2	0	0	0	0	0	0	
614808	www.sohomod.com	167.114.118.44	CA	0	2	0	0	0	0	0	0	
526862	www.truesocks.net	186.2.165.181	RU	0	1	0	0	1	0	0	0	
517075	www.thinktheology.co.uk	198.251.81.181	US	0	1	0	0	1	0	0	0	
672504	www.runechat.com	186.2.165.159	RU	0	1	0	0	1	0	0	0	
626572	www.mapfre.net	195.53.217.43	ES	0	1	0	0	0	0	0	0	
766647	www.regiocollege.nl	186.2.165.146	RU	0	1	0	0	1	0	0	0	
807760	www.ptichki.club	103.249.70.21	RO	0	1	0	0	1	0	0	0	
782917	www.inbox.eu	195.216.236.200	LV	0	1	1	1	1	2	2	1	
813315	www.x4b.net	103.249.70.250	RO	0	1	0	0	1	0	0	0	

A.4 Experimental TFO Option

The 14 hosts in this list are the ones that used the experimental TFO option at some point. A star denotes the hosts that also used the regular TFO option and are also in the regular list of hosts.

rank	ip	host	country	google	
2775	40.127.139.14	www.meduza.io	US	0	
29763	89.108.106.190	www.fanat1k.ru	NL	0	
45018	144.76.82.156	www.samba.org	NL	0	
45783	195.216.236.230	www.mail.ee	LV	0	*
64698	195.216.236.10	www.inbox.lt	LV	0	*
165034	61.147.80.168	www.ceve-market.org	AU	0	
268922	64.210.135.72	www.teamskeetimages.com	US	0	*
276262	140.211.167.135	www.enlightenment.org	US	0	
282993	128.199.73.182	www.imoney.ph	SG	0	*
353553	151.80.139.61	www.soniaperonaci.it	NL	0	
554991	188.166.208.247	www.imoney.sg	NL	0	
593839	217.159.186.253	www.rocs.ru	NL	0	
826004	198.50.194.216	miconcinemas.com	FR	0	
946823	62.210.127.111	www.seashepherd-shop.com	NL	0	

Appendix B

Original Problem

This is the original presentation of the problem as advertised by the Communication Systems Group at ETH Zurich.

Measuring Internet Path Transparency for Transport Protocol Extensions

Recent years have seen the definition of new transport protocols, and extensions of existing transport protocols, to provide new services to applications in the Internet. The Fast Open extension to TCP, for example, reduces connection setup latency by allowing data to be sent in the first packet. However, the widespread deployment of middleboxes, which can interfere with traffic they don't understand, makes it hard to deploy these new protocols.
The risk of middlebox meddling to the functionality of a protocol extension, or to connectivity when the protocol extension is enabled, can be measured through simple A/B testing: attempt to connect from a measurement point to a target both with and without the extension enabled, and observing the differences. A/B testing can be done from multiple vantage-points in the Internet to help to differentiate impairments to connectivity and functionality that are due to devices in the Internet core from those nearer to the target.
This A/B testing has already been done for the Explicit Congestion Notification extension to TCP/IP [1] using the Pathspider tool [2], developed in previous semester and master theses at TIK. It works by alternating connections to a target while alternating system-level configuration parameters to enable the desired extension. The Pathspider approach can be generalized to other extensions, in order to build a more comprehensive picture of the deployability of extensions other than ECN.
This semester's thesis will develop extensions to the existing Pathspider tool, written in Python, to test additional features, for example TCP Fast Open [3] and Accurate ECN [4], and build a measurement probe for using Pathspider to test support for these features in the Internet. The thesis report will include the results of a pilot measurement study of Internet hosts to determine support and deployability of these features.

References:

1. B. Trammell, M. Kuehlewind, D. Boppart, I. Learmonth, G. Fairhurst, and R. Scheffenegger, Enabling Internet-Wide Deployment of Explicit Congestion Notification, in Proceedings of the Passive and Active Measurement Conference 2015, Hong Kong SAR, China (http://ecn.ethz.ch/ecn-pam15.pdf)

2. Pathtools (https://github.com/mami-project/pathspider)

3. RFC 7413 (https://tools.ietf.org/html/7413)

4. More Accurate ECN Feedback in TCP (https://datatracker.ietf.org/doc/draft-ietf-tcpm-accurate-ecn/)

Bibliography

[1] PATHspider: A tool for active measurement of path transparency
Iain R. Learmonth, University of Aberdeen, Scotland; Gorry Fairhurst, University of Aberdeen, Scotland; Brian Trammell, ETH Zurich, Switzerland; Mirja Kuehlewind, ETH Zurich, Switzerland
July 2016
https://irtf.org/anrw/2016/anrw16-final13.pdf
Last accessed: 27.07.2016

[2] TCP Fast Open
Sivasankar Radhakrishnan, Google; Yuchung Cheng, Google; Jerry Chu, Google; Arvind Jain, Google; Barath Raghavan, ICSI
2011
http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37517.pdf,
Last accessed: 27.07.2016

[3] RFC7766: DNS Transport over TCP - Implementation Requirements
J. Dickinson, Sinodun; S. Dickinson, Sinodun; R. Bellis, ISC; A. Mankin, Verisign Labs; D. Wessels, Verisign Labs
March 2016
https://tools.ietf.org/html/rfc7766
Last Accessed: 27.07.2016

[4] RFC7413: TCP Fast Open
Sivasankar Radhakrishnan, Google; Yuchung Cheng, Google; Jerry Chu, Google; Arvind Jain, Google
December 2014
https://tools.ietf.org/html/rfc7413
Last Accessed: 27.07.2016

[5] Enabling Internet-Wide Deployment of Explicit Congestion Notification
Brian Trammell, ETH Zurich, Switzerland; Mirja Kuehlewind, ETH Zurich, Switzerland; Damiano Boppart, ETH Zurich, Switzerland; Iain Learmonth, University of Aberdeen, Scotland; Gorry Fairhurst, University of Aberdeen, Scotland Richard Scheffenegger, NetApp, Inc.
March 2015
http://ecn.ethz.ch/ecn-pam15.pdf
Last Accessed: 27.07.2016

[6] IANA - Transmission Control Protocol (TCP) Parameters
http://www.iana.org/assignments/tcp-parameters/tcp-parameters.xhtml
Last Accessed: 27.07.2016