Reproducing TCP Fast Open

Zhenhuan Su(su170@purdue.edu)
Wen-Ling Chi (chi64@purdue.edu)
Sandy Hsiao (hsiaos@purdue.edu)
Yufan Chen (chen4076@purdue.edu)
Junpei Ma (ma571@purdue.edu)

Motivation

- TCP data flows
  - Transmission Delay (bandwidth)
  - **Propagation Delay (data transfer time)**

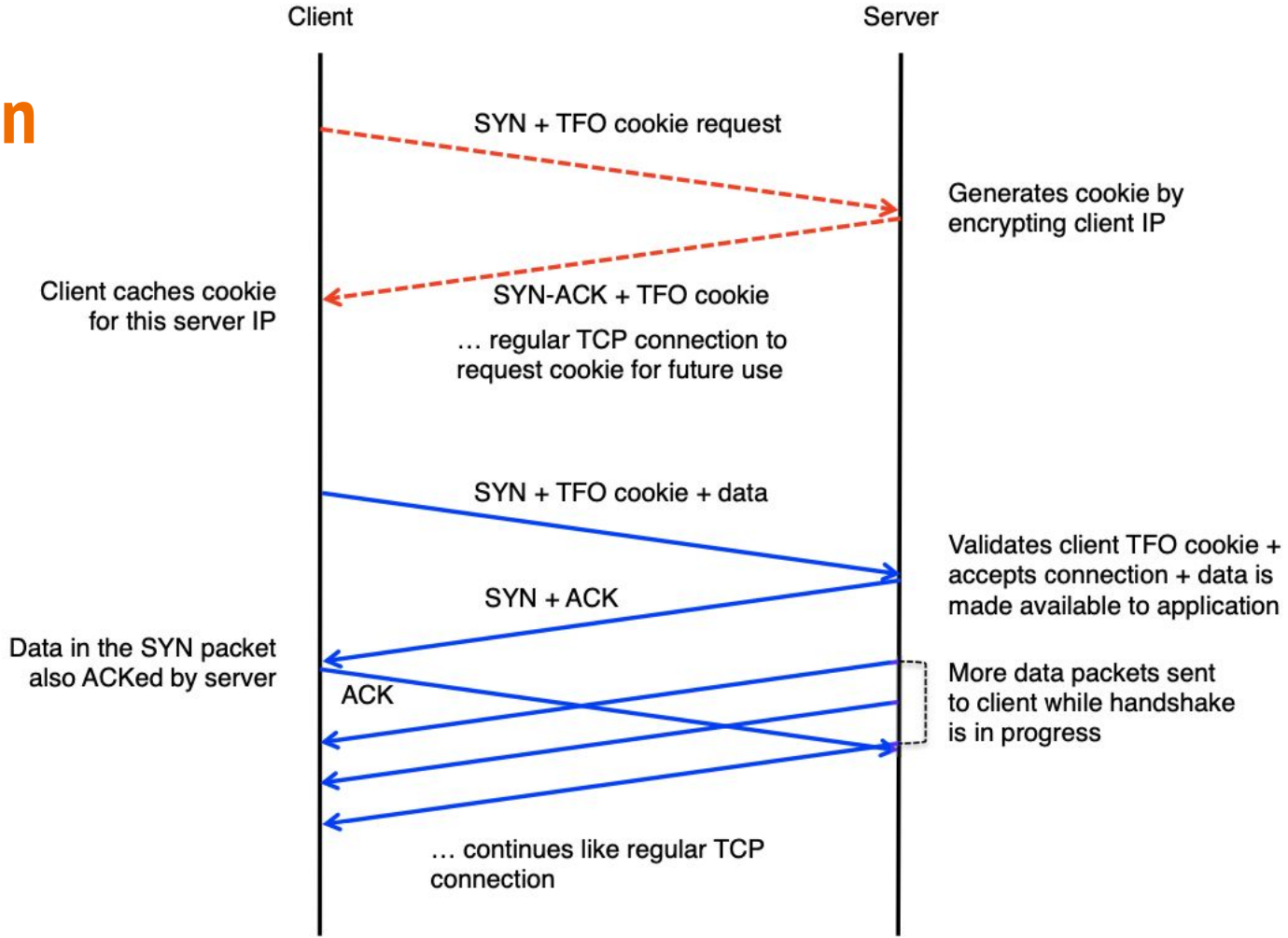
- **TCP Fast Open (TFO)**
  - A cryptographic cookie
  - Client and Server
  - Initial connection

Motivation

- Learning TCP 3-way handshake in course
  - Think how to reduce the time overhead
  - RTT: 1 >> 0

- Commonly used
  - Google Chrome
  - Some Apple's IOS versions
  - Microsoft Edge

TCP Fast Open

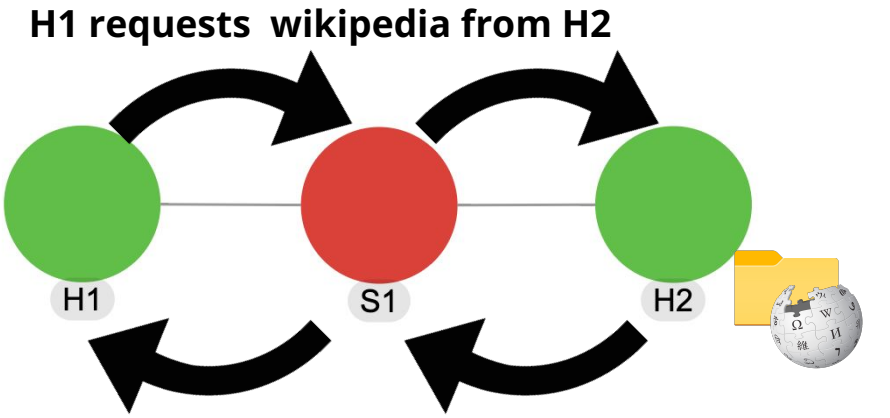


Results from original paper

Page	RTT(ms)	PLT : non-TFO (s)	PLT : TFO (s)	Improv.
amazon.com	20	1.54	1.48	4%
	100	2.60	2.34	10%
	200	4.10	3.66	11%
nytimes.com	20	3.70	3.56	4%
	100	4.59	4.30	6%
	200	6.73	5.55	18%
wsj.com	20	5.74	5.48	5%
	100	7.08	6.60	7%
	200	9.46	8.47	11%
TCP wikipedia page	20	2.10	1.95	7%
	100	3.49	2.92	16%
	200	5.15	3.03	41%

Table 1: Average page load time (PLT) in seconds for various pages for an emulated residential broadband user with a 4Mbps/256Kbps link. In all tests, the standard deviations of the PLT are within 5% of the average except for amazon.com with 20ms RTT (7%).

Design with Mininet



- Delay
- Bandwidth
- Enable TFO

Methods

Two approaches

1. Python Mininet API + mget + socketserver
2. Reuse course docker (mininet/controller) + python socket

Evaluate

1. Fixed **delay** with **different bandwidth** for both TCP/TFO
2. Fixed **bandwidth** with **different delay** for both TCP/TFO

Metric: Page load time(PLT)

Result Comparison: paper vs. method 1

Page	RTT(ms)	PLT : non-TFO (s)	PLT : TFO (s)	Improv.
amazon.com	20	1.54	1.48	4%
	100	2.60	2.34	10%
	200	4.10	3.66	11%
nytimes.com	20	3.70	3.56	4%
	100	4.59	4.30	6%
	200	6.73	5.55	18%
wsj.com	20	5.74	5.48	5%
	100	7.08	6.60	7%
	200	9.46	8.47	11%
TCP wikipedia page	20	2.10	1.95	7%
	100	3.49	2.92	16%
	200	5.15	3.03	41%

Control variable: Bandwidth = 10 Mb/s

site	RTT(ms)	Non-TFO(s)	TFO(s)	Improvement
amazon	20	17.179	16.321	5%
	100	55.604	42.255	24%
	200	108.312	81.231	25%
newyorktimes	20	3.676	3.150	14%
	100	14.356	10.342	28%
	200	28.362	19.558	31%
wsj	20	5.926	5.748	3%
	100	15.965	14.227	11%
	200	30.182	23.136	23%
wikipedia	20	0.91	0.803	12%
	100	3.761	2.859	24%
	200	7.356	5.222	29%

Left: original table; Right: method 1

Comparison between 2 methods: Same bandwidth

Control variable: Bandwidth = 10 Mb/s

site	RTT(ms)	Non-TFO(s)	TFO(s)	Improvement
amazon	20 100 200	17.179 55.604 108.312	16.321 42.255 81.231	5% 24% 25%
newyorktimes	20 100 200	3.676 14.356 28.362	3.150 10.342 19.558	14% 28% 31%
wsj	20 100 200	5.926 15.965 30.182	5.748 14.227 23.136	3% 11% 23%
wikipedia	20 100 200	0.91 3.761 7.356	0.803 2.859 5.222	12% 24% 29%

Control variable: Bandwidth = 10 Mb/s

Site	RTT(ms)	Non-TFO(s)	TFO(s)	Improvement
amazon	20 100 200	19.058 80.71 159.11	18.486 68.603 127.3	3% 15% 20%
newyorktimes	20 100 200	9.652 41.246 81.342	8.59 33.407 61.003	11% 19% 25%
wsj	20 100 200	8.042 23.401 43.194	7.801 21.526 36.713	3% 8% 15%
wikipedia	20 100 200	2.563 10.775 21.169	2.332 8.834 16.302	9% 18% 23%

Left: method1;Right: method 2

Comparison between 2 methods: Same delay

Control variable: delay = 10ms

Site	Bandwidth(Mb/s)	Non-TFO(s)	TFO(s)	Improvement
amazon	5 10 15	17.655 17.179 14.278	24.885 16.321 13.819	10% 5% 3%
newyorktimes	5 10 15	5.173 3.676 3.260	3.983 3.161 2.869	23% 14% 12%
wsi	5 10 15	10.019 5.926 4.660	9.417 5.748 4.565	6% 3% 2%
wikipedia	5 10 15	1.166 0.91 0.836	0.991 0.801 0.755	15% 12% 10%

Control variable: delay = 10ms

Site	Bandwidth(Mb/s)	Non-TFO(s)	TFO(s)	Improvement
amazon	5 10 15	22.745 19.179 18.203	20.926 18.604 17.839	8% 3% 2%
newyorktimes	5 10 15	10.931 9.627 9.324	9.401 8.568 8.39	14% 11% 10%
wsj	5 10 15	12.05 8.018 6.807	11.327 7.777 6.7893	6% 3% 1%
wikipedia	5 10 15	2.885 2.602 2.478	2.597 2.368 2.280	10% 9% 8%

Left: method 1;Right: method 2

Thank you