FEASIBILITY STUDY ON MPTCP ACK VIA ALTERNATIVE PATH IN REAL NETWORK ENVIRONMENT

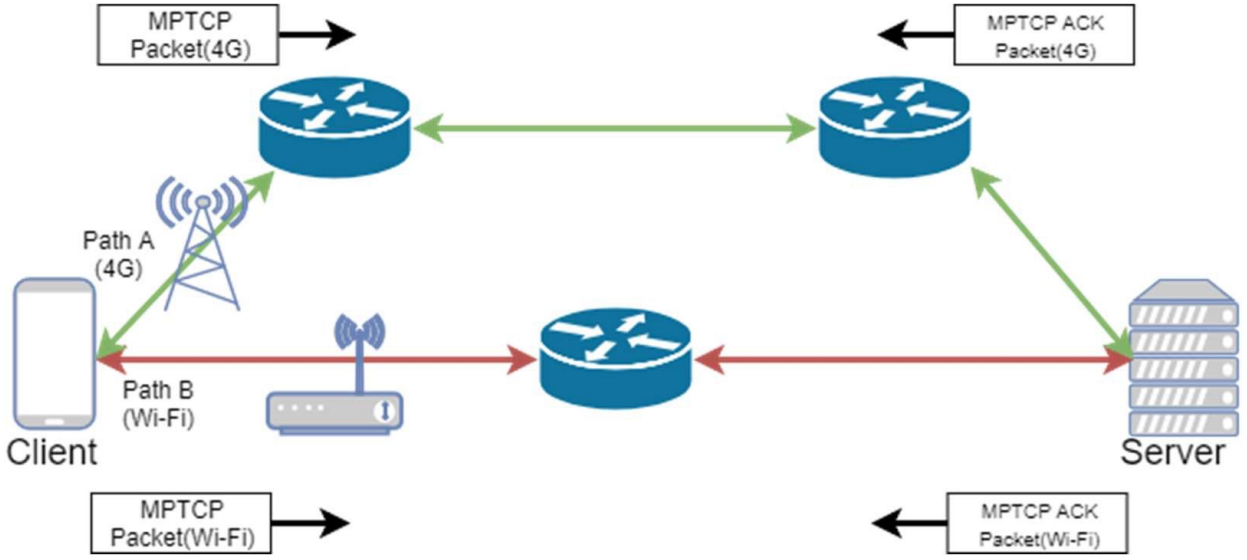
Hiroyuki KOIBUCHI, Hirotake ABE, Kazuhiko KATO
University of Tsukuba, Japan

Research Background

- **Cloud computing technology is getting a lot of attention.**
  - **Cloud computing services such as AWS, Azure, and GCP.**
  - **Data centers are located in various regions of the world.**
- **Data centers will be used for providing various services including big data analyses.**
  - **Handle large amounts of data among data centers across countries.**
  - **Improving communication performance is very important to improve the overall performance or service quality.**
- **Network protocols affect communication performance.**
  - **MPTCP (Multi Path TCP) is a protocol that can improve communication performance.**

MPTCP (Multi Path TCP)

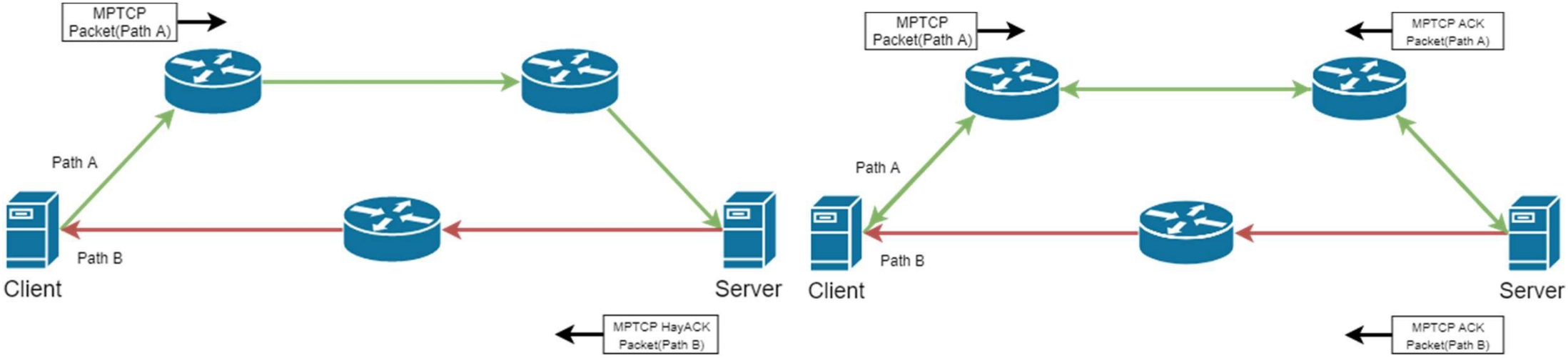
- A TCP extension that allow two or more TCP connections to be used simultaneously
  - Applications can regard those as a single connection (w/ standard socket API)
- Acknowledgment message for a data packet is returned through the same path as the data packet go
  - This behavior can cause performance problem in some cases.



Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, and Mark Handley. 2011. Improving datacenter performance and robustness with multipath TCP. SIGCOMM Comput. Commun. Rev. 41, 4 (August 2011), 266-277.

MPTCP extension to send ACK via alternative path

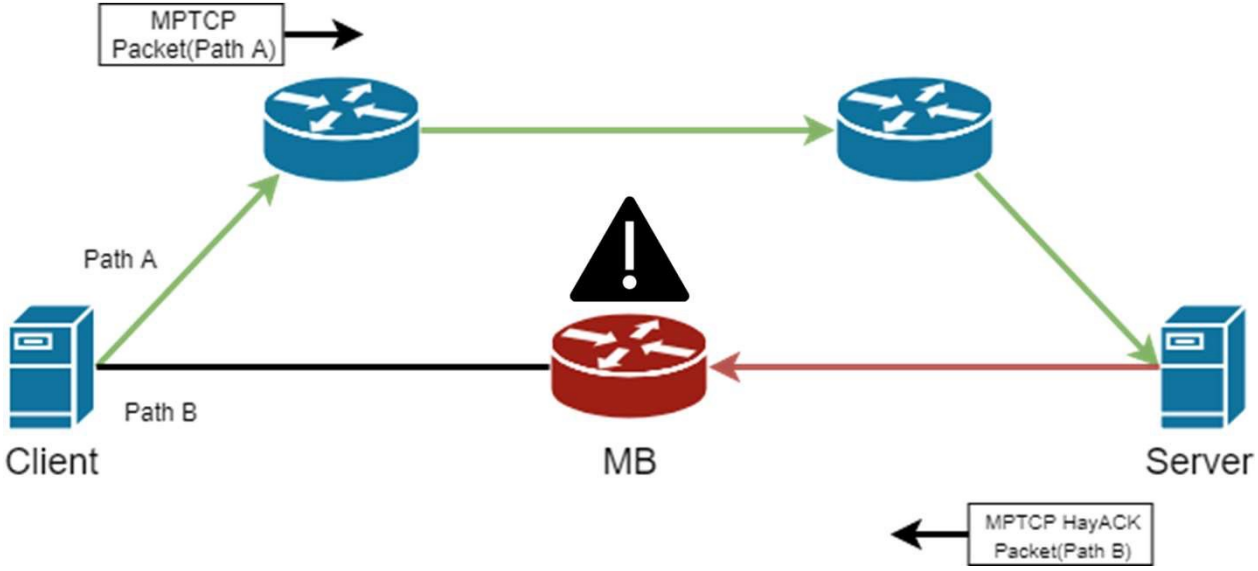
- **HayACK [CloudCom WS 2017]**
  - **Sends ACK packets via alternative path.**
  - **Optimized to cases where there is a big difference in RTT (Round Trip Time) between two paths.**
  - **Increased throughput by up to 1.72x, average 1.52x.**
    - **Experiments on ns-3 simulations.**



Yusuke Morikoshi, Hirotake Abe and Kazuhiko Kato. HayACK: Exploiting Characteristically Diverse Paths to Achieve Quick ACKing in MPTCP. International Workshop on the Future of Cloud Computing and Cloud Services (FutureCloud 2017, held in conjunction with IEEE CloudCom 2017), 8 pages. Hong Kong, China, December 2017.

Possible discard of ACK packets caused by Middleboxes

- **Middlebox (MB): General term for network appliances such as FW, NAT, VPN, TCP accelerator etc.**
  - **Most of those are dedicated for ensuring network security.**
  - **Discards "unusual" packets (e.g., inconsistent seq. no., unrelated to known TCP connections, etc.)**
- **Practicality of ACK via alternative path can be greatly affected by the deployment of MBs.**
  - **If it happens, extended MPTCP (like HayACK) needs to fall back to a single TCP connection.**



Research Objective

- **To survey how middleboxes are deployed in the current Internet that might interfere extended MPTCP that sends ACK via alternative path.**

Related work (1):

”Is it still possible to extend TCP?”

- By Honda et al. [IMC' 11].
- They solicitated volunteers from all over the world and asks to send “unusual” TCP packets that have inconsistent sequence number.
- Surveyed 142 routes from 24 countries
  - (jp:41 routes, uk:22 routes, us:17 routes, etc.)
- Only 67% of packets with inconsistent sequence numbers are arrived at the destination.

M. Honda, Y. Nishida, C. Raiciu, A. Greenhalgh, M. Handley, and H. Tokuda, "Is it still possible to extend TCP?" in Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. ACM, 2011, pp. 181-194.

Related work(2):

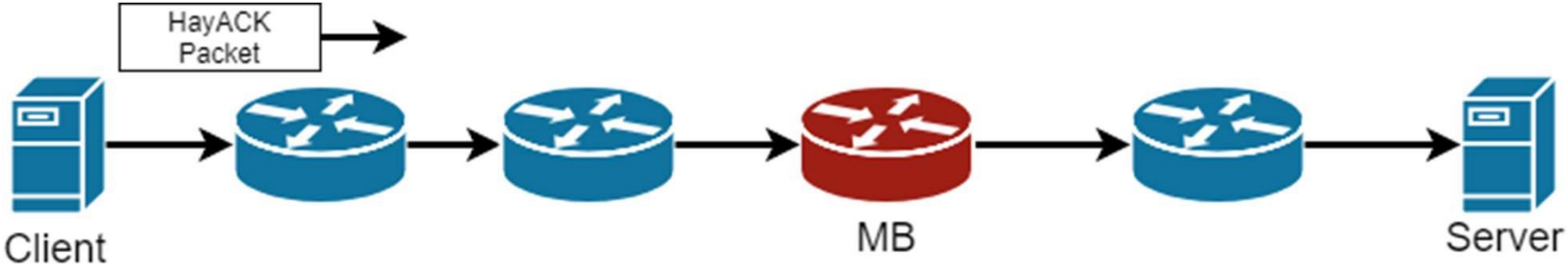
Study on the availability of ECN function

- By Mirja et al. [PAM' 13]
- They did a survey on compatibility of the Internet with ECN (Explicit Congestion Notification) packets
  - ECN introduces two additional flags to TCP packet format (ECE, CWR)
- Investigate the server's responses to ECN packets.
  - Send packets with SYN, ECE, and CWR flags
  - 30% of 100,000 servers responded to ECN packets.

Mirja Kühlewind, Sebastian Neuner, and Brian Trammell. 2013. On the state of ECN and TCP options on the internet. In Proceedings of the 14th internation al conference on Passive and Active Measurement (PAM'13), Matthew Roughan and Rocky Chang (Eds.). Springer-Verlag, Berlin, Heidelberg, 135-144.

Our method

- Let client hosts in all over the world send “unusual” ACK packet to server and we count how many of those packets are successfully arrived at our server.
  - Using a dedicated program made with Scapy, an API to tweak TCP/IP packets from a program in Python.
- Survey #1: (simpler)
  - Investigation with “spontaneous” ACK, which means the ACK that is not related to any known connections.
- Survey #2: (more complex)
  - Investigation with ACK packets that have inconsistent sequence number under “mimicked” TCP connections.



Survey Platform: VPNGate [NSDI' 14]

- Our challenge: how to collect as many client hosts as possible in a very limited time?
- Our solution: Use VPN service!
- VPNGate is a free-of-charge VPN service hosted by many (approx. 15k) volunteers all over the world.
  - Breaking a government censorship is their main objective.

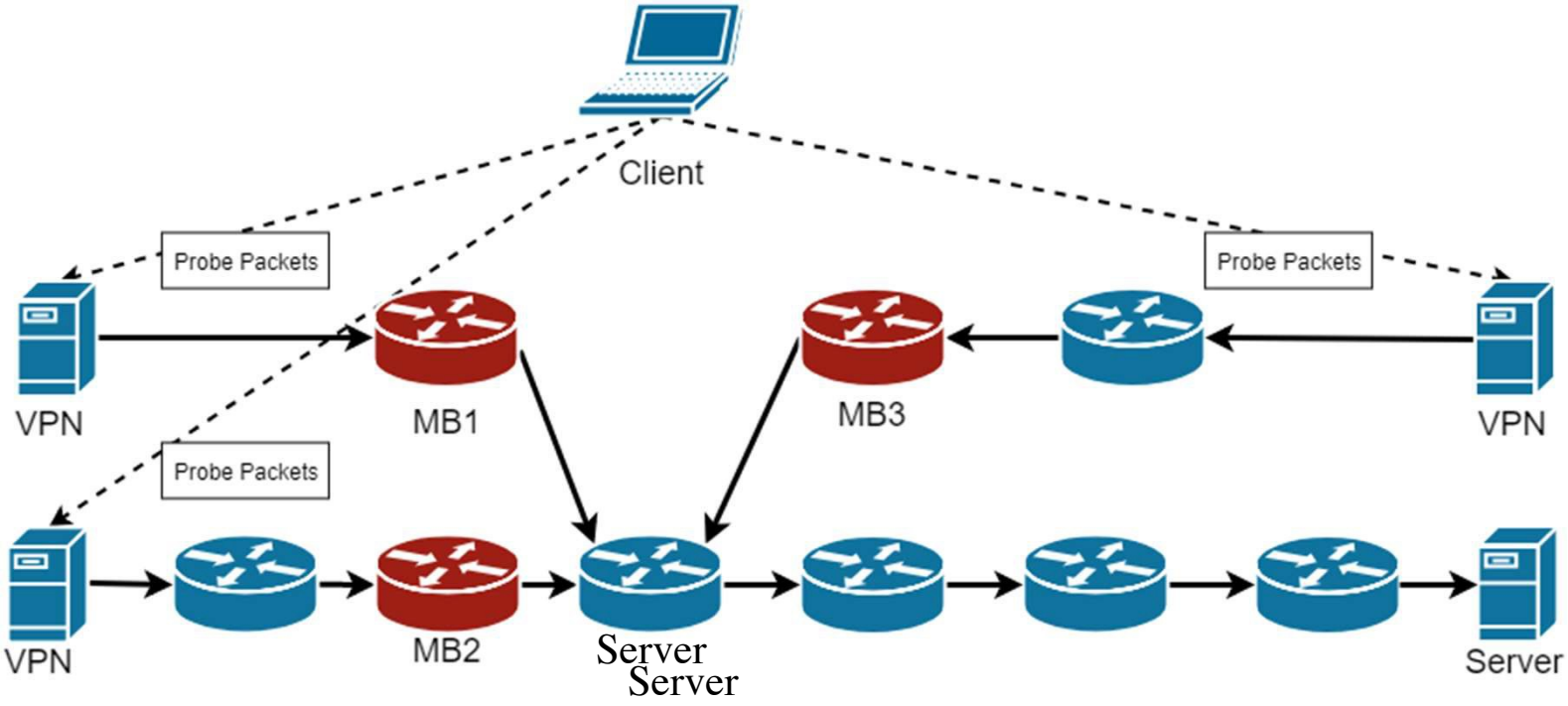
Daiyuu Nobori and Yasushi Shinjo. 2014. VPN gate: a volunteer-organized public VPN relay system with blocking resistance for bypassing government censorship firewalls. In Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation (NSDI'14). USENIX Association, Berkeley, CA, USA, 229-241.

How it works with VPNGate

- Our client host picks one of the VPNGate servers and establishes a VPN connection to the selected server.
  - They provide the latest list of available servers on their website (vpngate.net)
- Our client host sends various packet to our server via the VPN connection.
- Our server counts how many packets sent from our client host are successfully arrived.

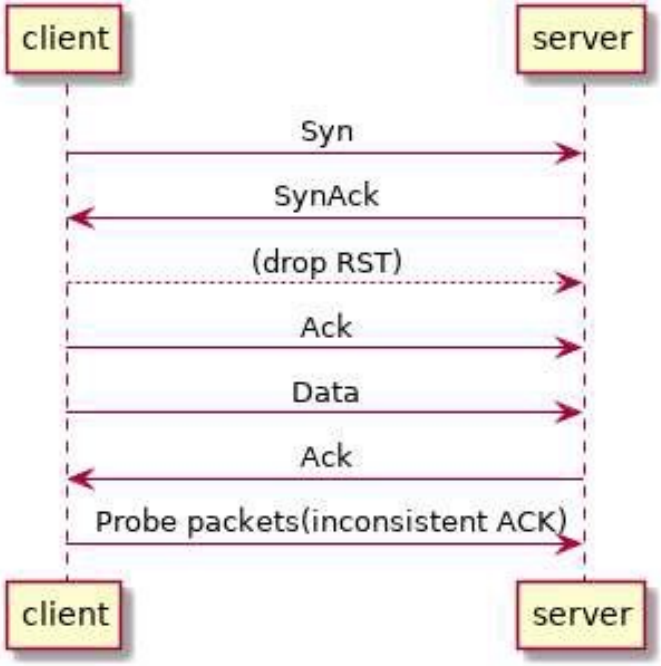
How survey #1 (”spontaneous”) works

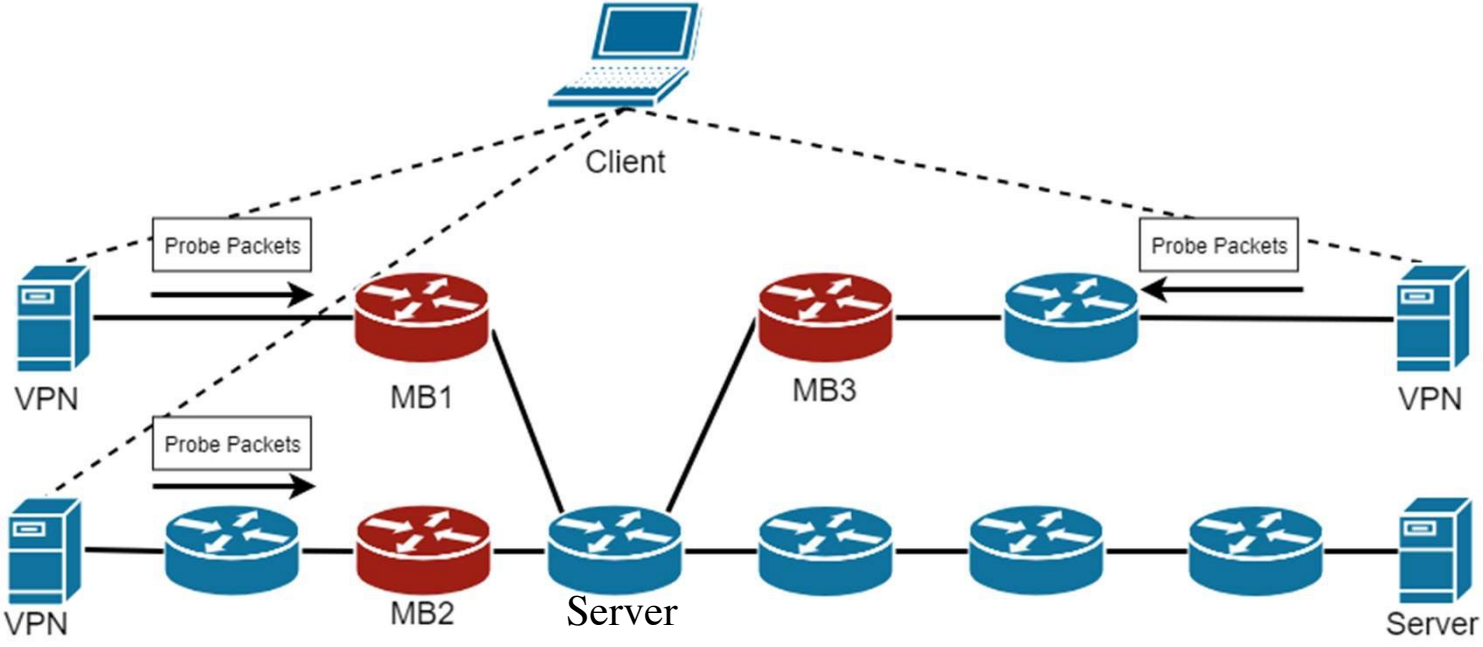
- **Send survey packets from various IP addresses without connections.**
  - **If a packet is sent via VPN, its sender address become VPN server’s, not client host’s.**



How survey #2 (“mimicked”) works

- **Send survey packets from various IP addresses after establishing connections using mimicked 3-way handshake with Scapy.**





Packet types used in the surveys

- Survey #1: “spontaneous” (simpler)
  - Ports: 80(http), 443(https), 34343
  - Flags: Syn, Ack, Syn+Ack, Syn+Ece, Ack+Ece
- Survey #2: “mimicked” (more complex)
  - Port: 80 (http)
  - Flags: Ack, Syn+Ack, Ack+Ece, Ack+Cwr, Ack+Cwr+Ece
- We used ECN flags because we thought it might affect the results.

A bug we found in VPNGate

- VPNGate uses SoftetherVPN as its building block.
- SoftetherVPN (v4.31 or earlier) had a bug in handling TCP packets with ECN flags.
  - It discards every TCP packets with both Syn and ECN flags!
    - Ironically, older SoftetherVPN might be a MB that prevents deploying new protocols… ☹
  - We reported the bug and they fixed it in v4.32
- We had to discard some of results related to ECN flags but others were unaffected.

Results of survey #1 "spontaneous" (including affected results)

- **Only 11% of VPNGate nodes were fixed at the time we conducted the survey.**

country	IP	S:80	S:443	S:34343	SE all	SA all	AE all	A all
jp	73	73	73	73	14	0	12	12
kr	76	76	76	76	4	0	0	0
us	6	6	6	6	0	0	0	0
hk	1	1	1	1	0	0	0	0
id	1	1	1	1	0	0	0	0
ve	1	1	1	1	0	0	0	0
vn	1	1	1	1	0	0	0	0
my	1	1	1	1	0	0	0	0
ph	1	1	1	1	0	0	0	0
ro	1	1	1	1	0	0	0	0
th	1	1	1	1	0	0	0	0
SUM	163	163	163	163	18	0	12	12
%	100	100	100	100	11	0	7	7

S:Syn, A:Ack, E:Ece

Results of survey #1 “spontaneous” (w/o affected results)

- n=18. (was 163 before filtering)
- We are still able to know few things even from these results.

S:Syn, A:Ack, E:Ece

country	IP	S:80	S:443	S:34343	SE all	SA all	AE all	A all
jp	14	14	14	14	14	0	12	12
kr	4	4	4	4	4	0	0	0
us	0	0	0	0	0	0	0	0
hk	0	0	0	0	0	0	0	0
id	0	0	0	0	0	0	0	0
ve	0	0	0	0	0	0	0	0
vn	0	0	0	0	0	0	0	0
my	0	0	0	0	0	0	0	0
ph	0	0	0	0	0	0	0	0
ro	0	0	0	0	0	0	0	0
th	0	0	0	0	0	0	0	0
SUM	18	18	18	18	18	0	12	12
%	100	100	100	100	100	0	67	67

- 67% of “sponta-neous” Ack was not filtered.
- Adding ECN flags to Ack does not affect at all.

Results of survey #2 “mimicked”

- **Not affected by the buggy VPNGate because Syn was already sent without ECN flags by mimicked 3-way handshake**

S:Syn, A:Ack, E:Ece, C:Cwr

- **Mimicking can improve acceptance of ACK with inconsistent seq. no. (67% => 82%)**
- **Again, adding ECN flags to Ack does not affect at all.**

country	IP	A：80	AE:80	SA：80	AC：80	AEC:80
KR	44	43	43	0	43	43
JP	31	21	21	0	21	21
VN	3	2	2	0	2	2
RS	2	1	1	0	1	1
AR	1	1	1	0	1	1
IN	1	1	1	0	1	1
CL	1	1	1	0	1	1
NO	1	0	0	0	0	0
TH	1	0	0	0	0	0
KH	1	0	0	0	0	0
LV	1	1	1	0	1	1
合計	87	71	71	0	71	71
%	100	82	82	0	82	82

Summary of the results

- **Acceptance rate of "unusual" ACK packets looks increasing in last decade.**
  - **In 2011: 67% (by Honda et al.)**
  - **In 2021: 82% (this work)**
- **Mimicking 3-way handshake can increase the acceptance rate.**
  - **67% (w/o handshake) => 82% (w/ handshake)**
- **ECN flags does not affect the drop rate.**

Conclusion

- We conducted surveys on acceptance rate of TCP ACK packets that have inconsistent sequence numbers.
  - It happens when we use extended MPTCP like HayACK.
- Our results show a sign that the situation of interference caused by middleboxes is getting better.
  - Broader chance that newer protocols like HayACK can work.

ACKnowledgment (that I should not drop)

- **This work was supported by JSPS KAKENHI Grant Number 18K11235.**

That‘s all. Thank you for listening!