SDN-based Controller Switching for Resilience of Drones

Young-Min Kwon and Kyung-Joon Park

Department of Information and Communication Engineering,
Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Republic of Korea
Email: {kym9102, kjp}@dgist.ac.kr

***Abstract*— In CPS, the network connecting the cyber and the physical systems is critical for ensuring stability of the physical system. A controller switching scheme using software-defined networking is proposed as a recovery method under network failure in CPS. We apply the controller switching scheme to the drone control system and validate its performance with empirical study.**

***Keywords*— Cyber Physical Systems, SDN, drones, controller switching, resilience**

I. Introduction

Recently, the term of cyber-physical systems (CPS) has gained a great interest and substantial research has been conducted [1, 2]. In CPS, networks play a critical role in ensuring reliability and stability of systems. In particular, networks deliver sensing and control messages between the cyber and the physical systems. Consequently, networking is crucial to ensure control performance of CPS.

In this work, we propose a controller switching scheme using software-defined networking for recovery from network failure in CPS [3]. Then, we validate the proposed architecture with a drone testbed.

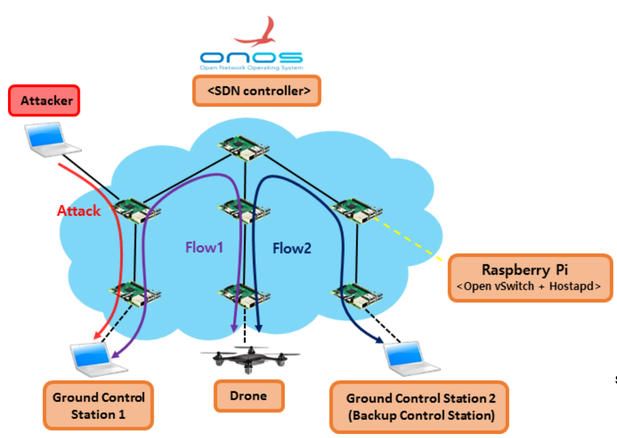

Fig. 1. Configuring a testbed for controller switching experiments.

II. Controller Switching in Drone Control System

A. Testbed Configuration Environment

In order to make an environment in which the drones are controlled via an SDN, a testbed is constructed as shown in figure 1. The testbed is implemented using SDN for switching of ground control station (GCS). We use ONOS as the SDN controller [4]. In addition, openvSwitch, a software-based switch, and hostapd, which performs function of an access point (AP), are installed in RaspberryPi-3 [5, 6]. We use Pixhawk X8 drones [7].

B. Experiments

The experimental scenarios are as follows: In the situation where GCS1 (Flow1) controls the drone, an attacker attacks GCS1 by using ICMP flooding. Here, we assume that the attacker has already entered the network. When a sender sends a request packet, it responds by sending a reply packet. By exploiting this feature, ICMP flooding attack overwhelms the receiver by flooded request packets. GCS1 that has been attacked by ICMP flooding will become busy replying ICMP packets, which blocks proper communication for drone control. Therefore, in order to guarantee the reliability of the drone control, the GCS2 (Flow2), which is a backup GCS, needs to take over the control function in real time using SDN.

										Mavlink Message Set				
	Time	system value								Field Name	value			
	2017-01-22T14:09:05.330	FE	C	0	0	4D	1	1	2	mavlink_system_time_t	time_unix_usec	0	time_boot_ms	14774552
	2017-01-22T14:09:05.340	FE	E	0	0	4E	1	1	9E	mavlink_mount_status_t	pointing_a	-20746	pointing_b	0
	2017-01-22T14:09:05.350	FE	16	0	0	4F	1	1	C1	mavlink_ekf_status_report_t	velocity_variance	0	pos_horiz_variance	4.56E-11
End Point →	2017-01-22T14:09:05.350	FE	20	0	0	50	1	1	F1	mavlink_vibration_t	time_usec	1.48E+10	vibration_x	0.02732
	2017-01-22T14:09:06.000	FE	9	0	0	1B	FF	BE	0	mavlink_heartbeat_t	custom_mode	0	type	6
	2017-01-22T14:09:07.000	FE	9	0	0	1C	FF	BE	0	mavlink_heartbeat_t	custom_mode	0	type	6
	2017-01-22T14:09:08.000	FE	9	0	0	1D	FF	BE	0	mavlink_heartbeat_t	custom_mode	0	type	6

(a) Drone log record of GCS1

										Mavlink Message Set				
	Time	system value								Field Name	value			
	2017-01-22T14:09:02.003	FE	9	0	0	AB	FF	BE	0	mavlink_heartbeat_t	custom_mode	0	type	6
	2017-01-22T14:09:03.005	FE	9	0	0	AC	FF	BE	0	mavlink_heartbeat_t	custom_mode	0	type	6
	2017-01-22T14:09:04.007	FE	9	0	0	AD	FF	BE	0	mavlink_heartbeat_t	custom_mode	0	type	6
	2017-01-22T14:09:05.010	FE	9	0	0	AE	FF	BE	0	mavlink_heartbeat_t	custom_mode	0	type	6
Start Point →	2017-01-22T14:09:05.781	FE	1A	0	0	74	1	1	1B	mavlink_raw_imu_t	time_usec	1.89E+09	xacc	0
	2017-01-22T14:09:05.781	FE	16	0	0	75	1	1	74	mavlink_scaled_imu2_t	time_boot_ms	14775034	xacc	10
	2017-01-22T14:09:05.796	FE	E	0	0	76	1	1	1D	mavlink_scaled_pressure_t	time_boot_ms	14775034	press_abs	1015.14

(b) Drone log record of GCS2

Fig. 2. Drone log record of GCSs.

In order to check the switching delay, we investigate the drone log file recorded in GCSs. As shown in Figure 2, GCS1 termination time is 5.350 seconds and GCS2 connection time is

5.781 seconds. Hence, the switching delay can be obtained as the difference of these two values. As a result of 10 experiments, we observe that the average switching time takes around 564 ms. For Pixhawk drones, the heartbeat message for drone control is delivered with a period of 1 second. If the heartbeat message is missed for more than 3 times, it will operate in failsafe mode. Our experimental results show that the proposed SDN-based architecture can switch the feedback controller within the period of heartbeat messages, which can guarantee reliability of drone control under network attack scenarios.

III. Conclusions

In this work, we have proposed a real-time recovery scheme for network failure in CPS, which can provide resilience of CPS. We have validated the proposed scheme with a drone testbed. In our future works, we plan how to detect the attack flow and when to perform controller switching.

Acknowledgment

This work was partly supported by Unmanned Vehicles Advanced Core Technology Research and Development Program Through the Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science, ICT and Future Planning, the Republic of Korea (NRF-2016M1B3A1A01937599) and Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIP; 2014-0-00065, Resilient Cyber-Physical Systems Research).

References

[1] K.-J. Park, R. Zheng, and X. Liu, " Cyber-physical systems: Milestones and research challenges," Computer Communications, vol. 36, issue 1, pp. 1-7, December 2012.

[2] K.-J. Park, J. Kim, H. Lim, and Y. Eun, "Robust path diversity for network quality of service in cyber-physical systems," IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2204-2215, November 2014.

[3] S. Yoon, J. Lee, Y. Kim, S. Kim, and H. Lim, “Fast controller switching for fault-tolerant cyber-physical systems on software-defined networks,” IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2017) - Fast abstract, Christchurch, New Zealand, Jan. 22-25, 2017.

[4] http://onosproject.org

[5] http://openvswitch.org

[6] https://w1.fi/hostapd/

[7] ardupilot.org/planner/index.html