Journal of Institute of Control, Robotics and Systems (2023) 29(1):000-000
https://doi.org/10.5302/J.ICROS.2023.0085 ISSN:1976-5622 eISSN:2233-4335

손실발생 모바일 SDN에서의 룰 캐싱 기반 신속 라우팅

Fast Routing with Rule Caching in Lossy Mobile SDN

김 장 한[1], 박 형 석[1], 박 경 준[1*]
(Janghan Kim[1] , Hyung-Seok Park[1] and Kyung-Joon Park[*])
[1]Department of Electrical Engineering and Computer Science DGIST

Abstract: UAVs (Unmanned Aerial Vehicles) and UGVs (Unmanned Ground Vehicles) have been actively employed for tasks that are challenging for humans in exhibition scenarios and disaster sites. Swarm unmanned systems comprising UAVs and UGVs experience dynamic additions or removals of unmanned aircrafts. In this paper, we propose a rule caching-based recovery technique for achieving rapid forwarding rule establishment for additional UAVs deployed for recovery in wireless SDN-based swarm unmanned systems. To validate the proposed algorithm, we conduct experiments by setting up a testbed using Odroid and a controller. Our results demonstrate that the proposed method improves forwarding rule establishment performance by up to 50.3%.

Keywords: Software-defined networking, lossy network, forwarding rule, caching

I. 서론

최근 전시 상황이나 재난 현장에서 UAV (Unmanned Aerial Vehicle)나 UGV (Unmanned Ground Vehicle)를 이용하여 작전 지역의 효율적 임무 수행을 위한 군집 무인 시스템에 대한 연구가 활발히 이루어지고 있다[1]. 군집 무인 시스템은 중앙에서 제어하는 GCS (Ground Control Station)와 UGV나 UAV와 같은 무인 로봇으로 구성된다. UAV나 UGV는 임무를 수행하면서 획득한 센서 데이터를 GCS로 전달하고 GCS는 임무 수행에 관련된 다양한 센서 데이터를 얻기 위해 UAV와 UGV를 적절히 배치한다.

군집 무인 시스템에서 UAV나 UGV들은 실시간 이동성을 가지고 있기 때문에 동적 토폴로지를 형성한다. 따라서 무인기 간 통신을 위하여 AODV, OLSR과 같은 애드혹 라우팅을 이용한 무선 네트워크를 주로 이용한다. 그러나 전시 상황같은 미션 크리티컬한 환경에서 중요한 데이터를 전송함에 있어서 애드혹 무선 네트워크는 상당히 많은 자원이 필요하다. 본 논문에서는 이러한 문제점을 해결하기 위해 실시간 모니터링이 가능한 SDN (Software-Defined Networking)을 이용하여 군집 무인 시스템을 구성하였다.

중앙 제어 구조인 SDN은 중앙 제어 장치인 컨트롤러와 연결된 모든 스위치들을 관리한다. 컨트롤러는 LLDP (Link Layer Discovery Protocol) 패킷을 자신과 연결된 모든 스위치에게 전달하여 실시간으로 토폴로지를 모니터링한다[2]. 이를 통해, 컨트롤러는 자신과 연결된 스위치에 최적의 포워딩 룰을 계산하여 모든 스위치에 전달함으로써 효과적으로 스위치를 관리할 수 있다.

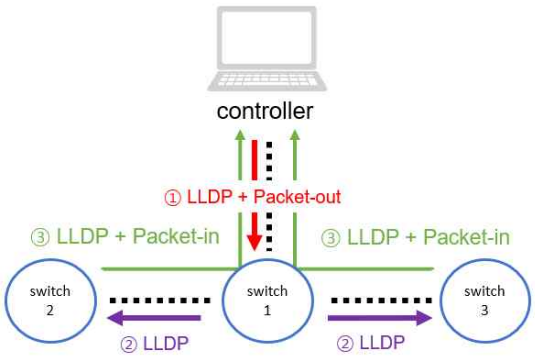


그림 1. SDN 기반 네트워크의 토폴로지 탐색 과정
Fig. 1. Topology discovery process in SDN-based networks

SDN 컨트롤러가 네트워크 토폴로지를 탐색하는 과정은 그림 1에서의 1 번 단계와 같다. 컨트롤러와 연결된 1 번 스위치에게 LLDP 패킷을 포함한 Packet-out 메시지를 전송한다. 이 후. 2 번 과정과 같이 Packet-out 메시지를 받은 스위치는 다시 자신과 연결된 2, 3 번 스위치에게 LLDP 패킷을 전파한다. 이러한 2 번 단계를 통해 LLDP 패킷을 수신한 2, 3 번 스위치는 LLDP 패킷과 함께 Packet-in 메시지를 컨트롤러로 다시 전송하고, 이를 통해 컨트롤러는 네트워크 토폴로지를 인식하게 된다. 이러한 방식을 통해 컨트롤러는

*Corresponding Author
Manuscript received 20xx.xx.xx; revised 20xx.xx.xx; accepted 20xx.xx.xx
김장한: 대구경북과학기술원 전기전자컴퓨터공학과 석사과정(jhjy96@dgist.ac.kr, ORCID 0009-0007-2685-6471)
박형석: 대구경북과학기술원 전기전자컴퓨터공학과 박사과정(hyungseok@dgist.ac.kr,ORCID 0000-0002-4127-079)
박경준: 대구경북과학기술원 전기전자컴퓨터공학과 교수(kjp@dgist.ac.kr,ORCID 0000-0003-4807-646)
※ 본 연구는 방위사업청과 국방과학연구소가 지원하는 군집형 무인 CPS 특화연구실 사업의 일환으로 수행되었습니다(UD220005VD).

주기적으로 LLDP 패킷을 전송하여 네트워크 토폴로지를 계속하여 감지한다.

기존의 SDN 기반 네트워크에서, 포워딩 룰을 계산하는 방법은 reactive forwarding과 proactive forwarding을 사용한다. reactive forwarding은 필요할 때만 경로를 결정하는 On-demand 라우팅 방식으로, 데이터 송신자가 목적지로 데이터 패킷을 전송할 때마다 경로를 계산하여 패킷을 전송한다. 이로 인해 초기 지연 시간이 발생하는 단점이 있다. 반면에, proactive forwarding은 미리 라우팅 정보를 보유하고 있는 방식으로, 데이터 송신자가 데이터 패킷을 전송할 때 이미 계산되어 있는 라우팅 테이블에 등록된 경로를 사용하여 즉시 데이터를 전송한다. 그러나 동적 토폴로지를 가진 네트워크 환경에서는 주기적으로 라우팅 정보를 업데이트해야 하므로 많은 자원이 필요한 단점이 있다.

군집 무인 시스템은 전시 상황이나 재난 상황에서 많이 활용되지만, 의도치 않은 상황으로 인해 무인기의 추가 및 탈락이 빈번하게 발생한다[3]. SDN 기반 군집 무인 시스템에서 무인기의 추가 및 탈락이 발생할 경우, 임무 수행 데이터를 신속하게 전송하기 위해 빠른 데이터 전달 경로를 구축하는 것이 중요하다. 그러나 컨트롤러와 추가/탈락한 스위치 사이의 홉 수가 증가하면 LLDP 패킷을 전송하여 네트워크 토폴로지를 파악하는데 많은 자원이 요구되어 추가/탈락한 스위치에 전송하는 데이터 전송시간이 증가한다. 따라서 추가/탈락한 스위치가 생길 경우, 각 무인기에 새로운 포워딩 룰을 신속하게 구축하는 연구가 필요하다.

본 논문에서는 SDN 기반 군집 무인 시스템에서 무인기가 추가될 경우, 캐싱을 이용하여 임무 수행 데이터 전달 경로를 신속하게 구축하는 방법을 제안한다. 제안하는 기법은 추가하는 무인기에 구축할 포워딩 룰을 미리 말단 노드에 캐싱한다. 또한 캐싱하는 스위치 이외에 모든 스위치에 추가되는 스위치로의 uplink와 downlink 포워딩 룰을 캐싱한다. 이후 추가된 무인기가 연결될 경우 캐싱했던 룰을 전달하여 신속히 임무수행 데이터 전달 경로를 구축하면서 다른 스위치 또한 uplink/downlink 포워딩 룰을 자동으로 구축한다. 제안 기법의 성능을 검증하기 위해 Odroid 10대를 linear 방식으로 연결하여 종단 간 거리를 10 홉으로 설정하였다. 검증 결과 제안한 캐싱 룰 알고리즘을 사용할 경우 최대 50.3% 빠르게 새로운 스위치로 포워딩 룰을 구축할 수 있었다.

본 논문은 2장에서 관련 연구 내용을 설명하고 기여점을 설명한다. 3장에서는 제안한 알고리즘에 대하여 설명하고 4장에서 실험 환경 설계 및 실험 결과에 대하여 설명하며 5장에서 결론을 맺는다.

II. 관련 연구

SDN에서 신속한 데이터 전달 경로 구축에 대한 방법으로 캐싱을 이용한 연구는 크게 3가지 방식으로 진행되어 왔다[4,5,6]. 첫 번째로 Luo et al. [4]은 SDN 기반 네트워크에서 포워딩 룰에 대해 캐싱을 이용하여 효율적으로 종단 간 시간지연을 감소시켰다. 그러나 [4]에서 연구한 SDN 기반 네트워크는 새로운 스위치가 추가되었을 때의 상황을 고려하지 않고 유선 SDN 기반 네트워크 상황에 대해서 시뮬레이션을 실행하였다. M Dong et al. [5]은 스위치가 이동하게 되어 다른 스위치에 연결이 될 경우 캐싱을 이용하여 비교적 신속한 포워딩 룰을 구축하도록 연구하였다. 그러나 이 논문에서는 기존에 연결되어 있던 스위치가 이동하는 상황에 대한 룰 캐싱 연구로 새로운 스위치에 대한 정보를 모르는 상황을 고려하지 않았다. Janghan Kim et al. [6]에서는 추가되는 스위치에 대한 룰을 캐싱하여 추가된 스위치에 대한 포워딩 룰을 신속하게 구축하는 연구를 진행하였다. 그러나 이 연구는 추가되는 스위치에 대한 정보를 컨트롤러와 연결되있던 스위치들이 알고 있다는 가정을 하여 uplink와 downlink에 대한 구축경로를 고려하지 않았다.

본 논문에서는 새로운 스위치에 대한 정보를 알아낸 뒤 새로운 스위치에 포워딩 룰을 신속하게 구축한다. 또한 새로운 스위치에 대한 uplink/downlink 포워딩 룰을 모든 스위치에게 설치한다. 이를 통해 모든 스위치는 새로운 스위치에 대한 정보가 없어도 새로운 스위치가 연결되자마자 데이터를 보낼 수 있다. 본 논문에서 제안하는 RCNS (Rule Caching for New Switch) 알고리즘의 성능을 증명하기 위하여 테스트베드를 구축하여 검증한다.

III. RCNS 알고리즘

3.1 포워딩 룰 캐싱

제안하는 알고리즘을 통하여 새로운 스위치에 구축할 포워딩 룰을 계산하여 자신과 연결된 스위치에게 전달하여 캐싱한다. 또한 새로운 스위치가 기존에 컨트롤러와 연결된 모든 스위치에 데이터를 전달할 수 있도록 uplink/downlink에 대한 포워딩 룰을 전달한다.

첫 번째로 새로운 스위치에 대한 룰은 새로운 스위치가 추가되어 말단 스위치와 연결되었을 경우를 가정하여 각각의 말단 스위치에 새로운 스위치에 설치할 포워딩 룰을 계산하여 생성한다. 생성된 룰은 각각 말단 스위치에 전달되고 말단 스위치는 전달받은 포워딩 룰을 저장한다.

두 번째로 새로운 스위치와 기존 스위치가 통신하도록 uplink/downlink에 대한 룰을 컨트롤러가 계산한다. 새로운 스위치와 연결되는 스위치에 따라 다음 홉으로 가는 uplink와 이전 홉으로 가는 downlink에 대한 포워딩 룰을 계산하여 각 스위치에 전달하고 각 스위치는 전달받은 uplink와 downlink 포워딩 룰을 저장한다.

3.2 추가되는 스위치와 연결

추가되는 스위치가 인접한 1 홉 거리의 스위치와 연결되기 위해 추가되는 스위치는 자신의 존재를 브로드캐스팅 방식으로 알린다. 또한 인접 스위치를 제외한 모든 스위치에게 추가되는 스위치가 자신과 연결되었다는 것을 알리기 위해 멀티캐스팅 방식을 사용하였다.

첫 번째로 브로드캐스팅 방식은 그림 2의 1 번 과정과 같이 추가되는 1 번 스위치가 가장 가까운 이웃 스위치에

Journal of Institute of Control, Robotics and Systems (2023) 29(1):000-000
https://doi.org/10.5302/J.ICROS.2023.0085 ISSN:1976-5622 eISSN:2233-4335

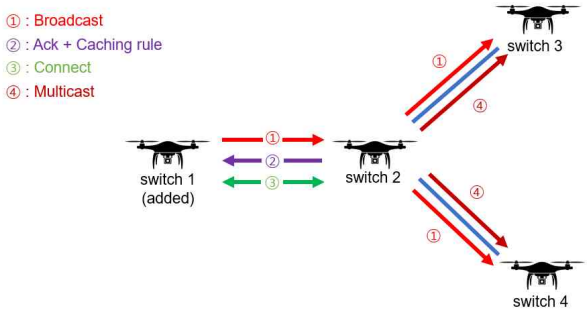


그림 2. 추가되는 스위치에 대한 Alert 기법
Fig. 2. Alert mechanisms for newly added switches

게 자신의 정보를 알리기 위해 사용된다. 가장 먼저 브로드캐스팅으로 전달된 패킷을 수신한 2 번 스위치가 그림 2의 2 번 과정같이 ACK 메시지와 함께 캐싱된 룰을 추가된 스위치에게 전송한다. ACK 메시지를 받은 1 번 스위치는 2 번 스위치와 연결됨과 동시에 전달받은 포워딩 룰을 자신에게 구축한다. 동시에 2 번 스위치는 1 번 스위치에 임무 데이터를 전달할 수 있는 포워딩 룰을 구축하게 된다. 또한 저장되어 있던 uplink/downlink 포워딩 룰을 동시에 설치하게 되어 1 번 스위치에 대한 uplink/downlink 포워딩 룰을 구축하게 된다.

두 번째로 2 번 스위치는 멀티캐스팅 방식으로 그림 2의 4 번 과정과 같이 자신을 제외한 모든 스위치에게 1 번 스위치가 연결되었다고 알린다. 모든 스위치들은 멀티캐스팅 방식으로 받은 알림으로 인해 1 번 스위치에 대한 정보를 알아내어 미리 저장한 uplink/downlink 포워딩 룰을 자신에게 설치한다. 따라서 모든 스위치가 새로운 스위치로의 데이터 전달 경로 구축이 완성된다.

IV. 실험

1. 실험 환경 및 시나리오

본 논문에서는 우리가 제안하는 알고리즘을 성능검증하기 위해 실험환경을 구성하였다. 그림 3과 같이 스위치로 사용할 10대의 Odroid와 컨트롤러로 사용할 노트북 1대를 이용하여 실험하였다. 다수의 무인기로 이루어진 군집 무인 시스템에서 형성된 토폴로지는 무인기 간 멀티홉 관계를 가진다. 이 실험에서는 무인기 간 멀티홉이 최대 10 홉 거리의 상황을 가정하여 진행하였다. 또한 전시 상황이나 재난 상황의 경우, 외부 요인에 의한 통신 간섭으로 패킷 손실이 비교적 많이 일어나는 상황을 가정하여 loss rate를 {0, 10, 20, 30, 40}% 로 설정 후 연결시간을 측정하였다. 각 Odroid는 OVS (Open vSwitch)를 설치하였고 그림 4와 같이 10대의 Odroid가 종단간 10 홉이 되도록 유선으로 linear 연결된 토폴로지를 사용하였다. 9대의 Odroid는 컨트롤러와 연결하였으며 마지막 10 번째의 Odroid는 추가되는 스위치로 가정하여 컨트롤러와 연결하지 않았다. 10대의 Odroid는 유선으로 연결하였지만 제안한 알고리즘은 유•무선 상관

그림 3. 실험에 사용된 Odroid 와 컨트롤러
Fig. 3. Odroid and controller used in an experiment

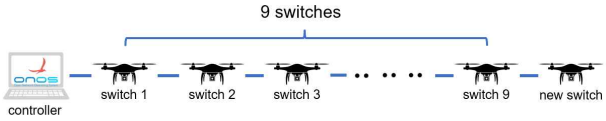


그림 4. 실험 환경에 적용한 토폴로지
Fig. 4. Topology used in an experiment

표 1. 실험환경 파라미터
Table 1. Testbed parameter

Parameter	Value
Ubuntu	18.04 LTS
Odroid	Odroid xu4 * 10
relay node	10
loss rate	0 10 20 30 40
mesurement tool	Netcat
forwarding application	reactive forwarding
ONOS version	2.3.0 - rc

없이 동작할 수 있도록 설계하여 무선 SDN 기반 네트워크에서도 적용할 수 있다. 컨트롤러는 ONOS 컨트롤러를 설치하여 연결된 스위치들을 모니터링하였다.

추가되는 스위치 상황을 실행하기에 앞서, 컨트롤러는 ONOS 컨트롤러에 기본적으로 설치되어있는 reactive forwarding 어플리케이션과 제안 알고리즘을 실행시켜 캐싱될 정보들이 모든 스위치에 전달하였다. 이 과정에서 모든 스위치에 포워딩 룰을 캐싱하기 때문에 각 스위치의 오버헤드가 증가할 수 있다는 단점이 있다. 이후, 추가되는 스위치가 연결되면 Netcat 프로그램을 이용하여 포워딩 룰이 구축되는 시간을측정하였다. 실험에 사용된 파라미터는 표 1과 같이 적용하여 실험하였다.

2. 실험 결과

추가되는 스위치는 인접 스위치와 연결되면서 Netcat의 서버를 열게 된다. 동시에, 멀티캐스팅을 통해 uplink/downlink 포워딩 룰을 구축한 10 홉 거리의 스위치에서 추가된 스위치에게 Netcat 연결 후, 데이터 전송 완료 시간을

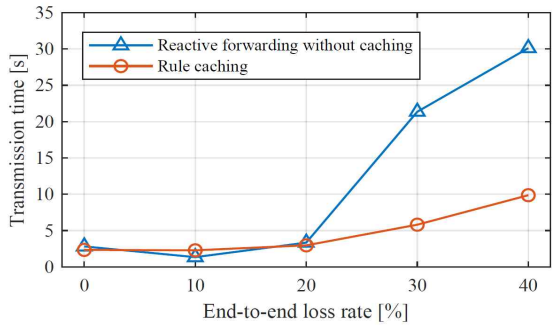


그림 5. 각 loss rate에 따른 종단간 데이터 전송시간
Fig. 5. End-to-end transmission time for each loss rate

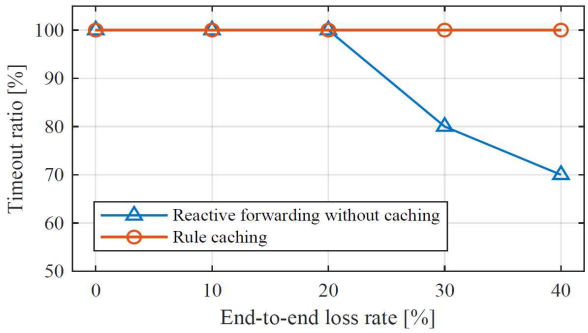


그림 6. 각 loss rate에 따른 연결 성공 확률
Fig. 6. Probability of successful connections for each loss rate

측정하였다. 이 과정에서 종단 간 loss rate를 {0, 10, 20, 30, 40}% 까지 변경해가면서 각 loss rate마다 10 번의 횟수를 거쳐 평균값을 구하였다. 비교군으로 ONOS가 기본으로 제공하는 어플리케이션인 reactive forwarding 앱을 실행시키고 나서 측정한 시간과 제안 알고리즘을 같이 사용한 시간을 비교하였다.

그림 5는 reactive forwarding 어플리케이션만 사용하여 측정한 평균값과 제안한 알고리즘을 같이 사용하여 측정한 평균값을 비교한 그래프를 나타낸다. 만약 Netcat 연결이 timeout이 되어 연결 실패될 경우 평균값 계산 시 제외하였다. 그림 5와 같이 loss rate가 {0, 10, 20}%일 경우, reactive forwarding 어플리케이션만 실행하여 측정한 시간과 제안한 룰 캐싱 알고리즘을 같이 사용하여 측정한 시간이 뚜렷한 차이가 나지 않는다. 그러나 loss rate가 {30, 40}%일 경우, 제안한 룰 캐싱 알고리즘을 같이 사용한 측정값이 {50.3, 42.9}% 만큼 신속하게 임무 수행 데이터 전달 경로를 구축하였다.

그림 6은 reactive forwarding 어플리케이션만 사용하였을 때 Netcat 연결시간이 timeout이 되어 연결 실패된 확률을 나타낸다. 제안한 룰 캐싱 알고리즘을 같이 사용할 경우 Netcat 연결이 모든 loss rate 환경에서 100% 확률로 성공하였다. 그러나 reactive forwarding만 사용하였을 경우 {0, 10, 20}%의 loss rate를 설정하였을 때 100% 확률로 연결에 성공하였지만 30%의 loss rate 설정부터 Netcat 연결에 실패하는 경우가 발생했다.

loss rate 설정 후 reactive forwarding만 사용할 경우, 연결 실패가 일어나는 이유는 그림 7의 (a)와 같이 주기적으로 보낸 LLDP가 loss가 일어나면서 토폴로지 파악을 실패한다.. 이로 인해 그림 7의 (b)와 같이 특정 스위치의 연결이 끊기게 되면서 ONOS의 비정상적인 토폴로지로 인식하게 된다. 결국, 추가되는 스위치에 대한 정상적인 포워딩 룰을 구축하기까지 오랜 시간이 걸려 timeout 현상이 발생된다. 그러나 제안한 룰 캐싱 알고리즘을 같이 사용할 경우, uplink/downlink 포워딩 룰을 스위치에 구축하기 때문에 ONOS에서 토폴로지 탐색을 위한 LLDP 패킷이 손실되더라도 추가되는 스위치에 대한 포워딩 룰이 구축되어 있어 연결할 수 있다. 이로 인해 제안한 룰 캐싱 알고리즘을 같이 사용할 경우 종단 간 임무 수행 데이터 전달 경로를 구축하여 데이터 손실 없이 전달 가능하며 최대 50.3% 신속한 데이터 전달이 이루어질 수 있다.

V. 결론

최근 전시 상황이나 자연재해와 같이 인간이 임무를 수행하기 어려운 환경에서 군집 무인 시스템을 활용하고 있다. 본 논문에서는 군집 무인 시스템에 SDN을 도입하여 임무 수행 데이터를 전송한다. 특히 SDN을 도입한 군집 무인

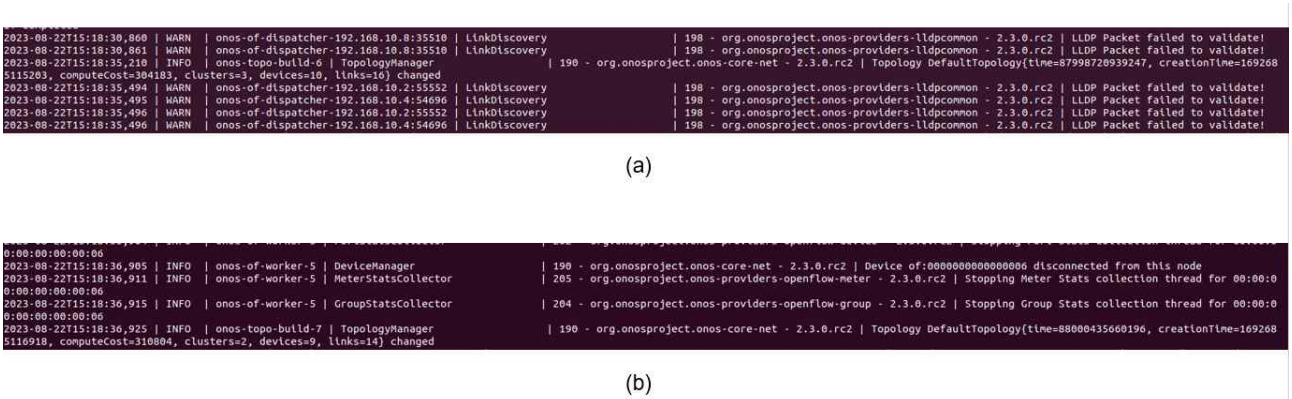


그림 7. ONOS 컨트롤러 log; (a) LLDP 패킷손실 log, (b) 6 번 스위치 연결 끊김 log
Fig. 7. ONOS controller log; (a) LLDP loss packet log, (b) 6th switch disconnected log

시스템에서 의도치 않은 무인기 고장 시 이를 복구하기 위하여 새로운 스위치가 추가될 경우, 룰 캐싱을 이용하여 신속한 임무 수행 데이터 전달 경로를 구축하는 연구를 하였다.

SDN 기반 네트워크에서 룰 캐싱을 이용하여 전달 경로를 구축하는 대부분의 연구는 무선 SDN 구조를 고려하지 않았으며 새로 연결된 스위치에 대한 정보를 알고 있다는 가정하에서 룰 캐싱을 이용한다. 본 논문에서는 새로운 스위치에 대한 정보를 모르는 상태에서 새로운 스위치가 연결되었을 때 룰 캐싱을 이용하여 신속히 포워딩 룰을 구축하는 방법을 제안하였다. 제안하는 방법은 새로운 스위치와 연결된 인접 스위치가 새로운 스위치에 구축할 포워딩 룰을 직접 전달하여 ONOS 컨트롤러의 개입 없이 새로운 스위치에 포워딩 룰을 구축할 수 있다. 실제 실험환경을 구축하여 검증하였고, 실험 결과 기존 SDN에서 사용하는 포워딩 룰 구축 방법보다 최대 50.3% 신속하게 임무 수행 데이터 전달 경로를 구축하였고, 종단간 연결 실패 없이 전송가능하였다.

향후, 룰 캐싱에 필요한 스위치의 수를 줄여 스위치 간의 전달 데이터를 감소시키는 연구를 진행할 계획이다. 또한 선형 토폴로지 이외에 다양한 토폴로지에 적용하여 성능을 검증할 예정이다.

REFERENCES

[1] A. Girma et al., "IoT-enabled autonomous system collaboration for disaster-area management," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 5, pp. 1249-1262, September 2020.
doi: 10.1109/JAS.2020.1003291

[2] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14-76, Jan. 2015.
doi: 10.1109/JPROC.2014.2371999

[3] H. Park, S. Lee and K. Park, "Wireless SDN Self-Recovery for Unmanned Swarm Cyber-Physical Systems," *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, Jeju, Korea, Republic of, 2021, pp. 87-90.
doi: 10.23919/ICCAS52745.2021.9649967.

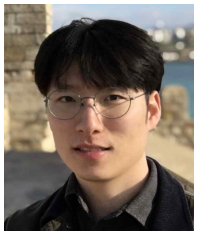
[4] L. Luo, R. Chai, Q. Yuan, J. Li and C. Mei, "End-to-End Delay Minimization-Based Joint Rule Caching and Flow Forwarding Algorithm for SDN," *IEEE Access*, vol. 8, pp. 145227-145241, 2020.
doi: 10.1109/ACCESS.2020.3013770.

[5] M. Dong, H. Li, K. Ota and J. Xiao, "Rule caching in SDN-enabled mobile access networks," in IEEE Network, vol. 29, no. 4, pp. 40-45, July-August 2015.
doi: 10.1109/MNET.2015.7166189.

[6] Janghan Kim, Hyung-Seok Park, Kyung-Joon Park."Fast Recovery using Rule Caching in SDN based Wireless Multi-hop Networks" *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*,(2023):671-672.
https://dbpia.co.kr/journal/articleDetail?nodeId=NODE11487225

김 장 한

2022년 충남대 정보통신공학과 졸업.
2022년~현재 대구경북과학기술원 전기전자컴퓨터공학과 석사과정 재학 중
관심분야는 SDN, 네트워킹 서비스



박 형 석

2018년 한밭대 컴퓨터공학과 졸업.
2020년 대구경북과학기술원 정보통신융합전공 석사.
2020년~현재 대구경북과학기술원 전기전자컴퓨터공학과 박사과정 재학 중
관심분야는 Software-defined networking, robot operating system, data distribution service, networked control systems, and resilient cyber-physical systems

박 경 준

1998년 서울대학교 전기공학부 졸업.
2000년 서울대학교 전기공학부 석사.
2005년 서울대학교 전기컴퓨터공학부 박사.
2011년~현재 대구경북과학기술원 전기전자컴퓨터공학과 교수.
관심분야는 Cyber-Physical Systems