

SDN 환경에서 효율적인 세그먼트 라우팅 구현

김 영 일*, 권 태 옥^o

Implementing of Efficient Segment Routing in SDN

Young-il Kim*, TaeWook Kwon^o

요 약

기존 네트워크 아키텍처의 한계를 극복하기 위해 등장한 소프트웨어 정의 네트워킹(SDN)은 중앙 컨트롤러를 통해 라우팅 관리를 보다 간단하고 효율적으로 만든다. SR(Segment Routing)은 소스 라우팅을 수행하는 유연하고 확장가능한 방법으로 패킷 헤더에 정렬 된 세그먼트 목록을 통해 네트워크의 정보 경로를 정의한다. SDN을 사용하는 환경에서 각 라우터의 성능은 거의 동일하지만 라우팅 알고리즘에 따라 자주 활용되는 경로에 패킷이 집중되는 경향이 있다. 해당 경로에 있는 라우터는 고장의 빈도가 비교적 높고 병목현상이 발생할 가능성이 높다. 본 논문에서는 SR이 있는 SDN에서 네트워크 내 자원인 라우터가 패킷을 균등하게 처리하여 관리자 입장에서 유휴 라우터가 없이 네트워크 내의 자원을 활용함과 동시에 라우터의 관리를 수월하게 하는 라우팅 알고리즘을 제안한다.

키워드 : 소프트웨어 정의 네트워킹, 세그먼트 라우팅, 라우팅 알고리즘, 네트워크 자원

Key Words : Software-Defined Networking(SDN), Segment Routing, Routing Algorithm, Network Resource

ABSTRACT

Software-Defined Networking (SDN), which has emerged to overcome the limitations of existing network architectures, makes routing management simpler and more efficient through a central controller. SR (Segment Routing) is a flexible and scalable way of doing source routing, and defines the information path of the network through a list of segments arranged in the packet header. In an SDN environment, the performance of each router is almost the same, but packets tend to be concentrated on routes that are frequently used depending on routing algorithms. Routers in that path have a relatively high frequency of failure and are more likely to become bottlenecks. In this paper, we propose a routing algorithm that allows the router, which is a resource in the network, to evenly process packets in the SDN with SR, so that the administrator can utilize the resources in the network without idle routers, and at the same time facilitate the management of the router.

1. 서 론

최근 인터넷과 관련 어플리케이션의 급성장에 따라 기존 네트워크는 점점 더 복잡해지고 운영과 유지 관리가 어려워졌다. 이러한 문제를 해결하기 위해 등장

한 소프트웨어 정의 네트워킹(Software Defined Networking, SDN)은 네트워크를 제어하기 위해 제어 평면과 데이터평면을 분리하고 제어평면의 컨트롤러를 소프트웨어적으로 프로그래밍 하여 필요한 네트워크 요소에 대한 세분화 된 기능을 제공한다. 세그먼트

* First Author : Korea National Defense University, fosung@gmail.com, 정회원

^o Corresponding Author : Korea National Defense University, kwontw9042@mnd.go.kr, 종신회원

논문번호 : 202107-166-B-RN, Received July 15, 2021; Revised September 1, 2021; Accepted September 1, 2021

라우팅(Segment Routing, SR)은 제어평면의 복잡한 작업 없이 효과적인 트래픽 엔지니어링을 가능하게 하기 위한 것으로^{1,2)}, 최근에 도입된 개념이다.

기존 네트워크의 라우팅 알고리즘은 보통 OSPF(Open Shortest Path First), IS-IS(Intermediate System to Intermediate System)를 사용하는데 이 알고리즘들은 링크상태를 기준으로 경로를 선택하게 된다. 최단 경로를 선택하기 위해 Dijkstra의 SPF 알고리즘을 사용하는데 결국 고정된 네트워크에서는 링크상태의 변화가 없으면 라우팅 테이블은 일정하게 유지되고 매번 같은 경로를 선택하여 패킷을 처리하게 된다. 이러한 문제는 해당 경로에 있는 네트워크 자원인 라우터에 병목현상이 발생할 가능성이 높고 과부하에 따른 장애가 빈번할 것이라고 예상된다. 이러한 문제를 해결하기 위해 SDN 환경에서 네트워크 자원의 균등한 활용을 위한 세그먼트 라우팅 알고리즘을 제안한다.

II. 관련 연구

2.1 Software Defined Networking(SDN)

ONF(The Open Networking Foundation)에서는 SDN을 네트워크의 제어평면과 중앙 집중화된 컨트롤러가 여러 대의 전송 장비를 통제하는 데이터 전송평면에 대한 물리적인 분할로 정의하고 있다³⁾. 기존 네트워크의 제어기능이 하드웨어에 있었다면 SDN에서는 소프트웨어를 통해 논리적 또는 가상적인 실체로서 네트워크를 관리하고 제어한다. 이러한 방법을 통해 관리하는 네트워크를 중앙에서 제어하여 정책을 적용하는데 있어 유리한 점이 많다. 또한 프로그래밍을 통해 필요한 기능이나 새로운 라우팅 알고리즘 등을 적용할 수 있다.

2.2 Segment Routing(SR)

세그먼트 라우팅은 소스 라우팅 기반의 기술로 패킷을 보내기 전에 송신측에서 수신측까지 가는 경로를 모두 리스트(세그먼트)로 만들어 패킷의 헤더에 넣어 보내는 방식이다⁴⁾. 기존 IP/MPLS 네트워크와 달리 포워딩 테이블을 채우기 위한 정보교환 프로토콜이 필요하지 않기 때문에 제어 평면의 작업이 크게 줄어들고 단순해졌다⁵⁾. 제어 평면의 아키텍처는 분산형, 중앙집중형, 하이브리드형으로 나뉘는데 앞서 살펴본 SDN과 연계하여 중앙 컨트롤러가 라우팅 정책에 의해 세그먼트를 계산하는 방법이 많이 사용되어⁶⁾ 연구가 많이 이루어지고 있다.

III. 제안 알고리즘

본 논문에서 제안하는 라우팅 알고리즘은 그림 1에서 보는 것과 같이 처리할 패킷이 생성되면 경로를 계산하게 되는데, 먼저 누적시간 동안 해당 노드의 패킷 처리량(handled packet)을 전체 노드의 처리량(total packet)으로 나눈 값에서 $1/\text{total node}$ 를 빼준다. $1/\text{total node}$ 를 빼는 이유는 각 노드가 동등하게 패킷을 처리했을 때를 기준으로 더 많이 패킷을 처리하면 양의 가중치를 주고, 적게 패킷을 처리하면 음의 가중치를 주기 위함이다. 예를 들어 5개의 노드가 있다면 동등하게 패킷을 처리하면 20%를 기준으로 잡아 양과 음의 가중치를 계산할 수 있게 된다. 그 다음으로 모든 노드의 Usage가 동일하다면 기존의 SPF(Shortest Path First)를 활용하여 경로를 선택하고, Usage가 다르다면 SPF와 Usage 가중치를 더한 값을 기준으로 경로를 선택하게 된다. 여기서 기존의 Link cost와 Usage cost를 더할 때 Usage cost에 적절한 추가 가중치인 K의 값이 필요하다. 먼저 모든 링크의 bandwidth는 동일하다고 가정하고, 각 노드가 처리한 패킷은 hp_x , 전체 노드의 처리량은 tp , 노드의 수는 n 으로 하고 임의의 a에서 b로 가는 경로 $R_1(a-c-b)$, $R_2(a-d-e-b)$ 가 있다고 가정한다. 각 경로의 cost를 RC_1 , RC_2 라고 하면, 수식 1과 수식 2로 나타낼 수 있다.

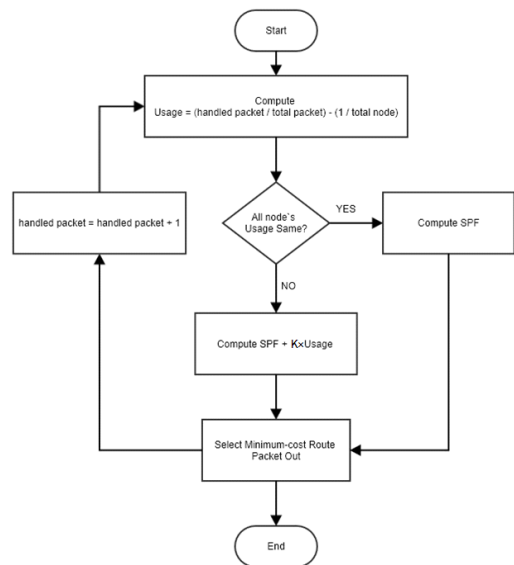


그림 1. 제안 알고리즘 흐름도
Fig. 1. Algorithm flowchart of proposed technique

$$RC_1 = \text{Link cost}_1 + K * \text{Usage}_{a \rightarrow c \rightarrow b} \quad (1)$$

$$RC_2 = \text{Link cost}_2 + K * \text{Usage}_{a \rightarrow d \rightarrow c \rightarrow b} \quad (2)$$

$$\text{Usage}_{a \rightarrow c \rightarrow b} = \frac{hp_a + hp_c + hp_b - (\text{Link cost}_1 + 1) \times \frac{1}{n}}{tp} \quad (3)$$

$$\text{Usage}_{a \rightarrow d \rightarrow c \rightarrow b} = \frac{hp_a + hp_d + hp_c + hp_b - (\text{Link cost}_2 + 1) \times \frac{1}{n}}{tp} \quad (4)$$

여기서 각 Usage 값은 수식 3과 수식 4로 나타낼 수 있다. 기존의 Link cost에 대비해서 Usage 가중치가 역할을 하기 위한 K 값의 범위를 구하려면 Link cost가 작은 RC₁이 RC₂ 보다 크기가 같도록 해야한다. 앞선 예를 들어 계산해보면 수식 5와 같이 나타낼 수 있고 Link cost₂-Link cost₁은 임의의

$$K \geq \frac{\text{Link cost}_2 - \text{Link cost}_1}{\frac{hp_c - (hp_d + hp_e)}{tp} + (\text{Link cost}_2 - \text{Link cost}_1) \times \frac{1}{n}} \quad (5)$$

상수 p가 되고, $\frac{hp_c - (hp_d + hp_e)}{tp}$ 는 패킷 처리량에 따라 -1에서 1 사이의 값을 갖는다. 이를 계산하면 K는 최소한 노드의 수인 n보다 큰 값을 가져야 기존의 Link cost와 같은 효과를 낼 수 있으나 Delay와 Hop count가 증가될 것으로 예상된다. 기존의 SPF와 비슷한 성능을 내면서 네트워크의 자원을 균등하게 사용할 수 있는 가중치 K를 실험을 통해 구해보고 그 성능을 평가해보도록 할 것이다.

IV. 실험 및 분석

4.1 실험 환경

제안하는 라우팅 알고리즘은 가상 네트워크 환경에서 구현하여 효과를 평가하였다. 실험 환경은 CPU AMD Ryzen 7 3700X 3.59 GHz, OS Windows 10, RAM 16GB, Virtual Box, Ubuntu 20.04.1 가상머신 환경에서 진행하였고, 사용한 프로그램은 미니넷 2.2.2 에뮬레이터와 Ryu 컨트롤러를 사용하였다. 참고한 소스코드는 세그먼트 라우팅을 고안한 단체인 Netgroup Research Group에서 진행하고 있는 ROSE 프로젝트 중 SDN-TE-SR-tools^[7] 코드를 활용하여 제안한 알고리즘으로 수정하여 실험환경을 구성했다.

4.2 실험 방법

실험은 미니넷을 통하여 그림 2와 같이 (1)-(3)-(2), (1)-(4)-(5)-(2)를 경유하는 다중 경로 토폴로지를 구축

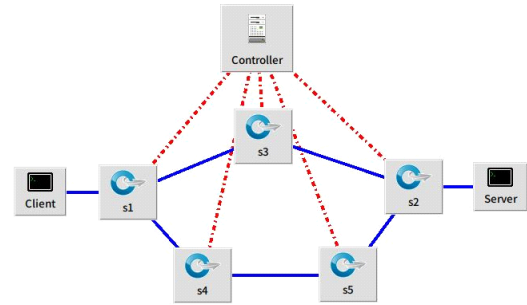


그림 2. 실험 토폴로지
Fig. 2. Topology

하고, 비교할 라우팅 메트릭은 기존의 SPF(Link cost), 제안하는 기법인 Link cost에 Usage를 더한 Link+Usage cost, 마지막으로 Usage로만 메트릭을 갖는 Usage cost로 총 3가지 방법을 비교한다. Client는 패킷을 생성하여 서버로 전송하여 Delay를 측정하고, 패킷을 모두 처리한 후 각 라우터의 Usage를 비교한다.

Client가 생성하는 패킷은 10,000개로 하였으며, Usage는 Request, Reply, Packet_In, Packet_Out 등 각 라우터가 처리하는 모든 패킷을 카운트한다. Usage를 갱신하는 시간은 Legacy network의 라우팅 테이블 갱신 시간과 같은 30초로 설정하였다. 각 라우터별 링크의 Bandwidth는 1Gbps로 동일하게 설정하였다. 실험은 각 메트릭 별로 10번을 반복하고 각 결과의 평균값으로 비교한다. 적절한 가중치 K를 구하는 방법은 K=1, K=n, 2n, 3n 으로 하여 성능을 비교한다.

4.3 실험 분석

실험을 통해 적절한 가중치 K 값을 찾고, 해당 K 값을 이용하여 기존의 SPF와 Usage만을 사용한 것과 제안 알고리즘 간의 성능을 분석해 보았다. 그림 3을 보면 적절한 K의 값을 찾기 위해 값을 바꿔가며 실험을 진행했고 처리한 패킷의 양이 그래프와 같은 결과가 나왔다. 앞서 예상한 것과 같이 K가 n보다 같거나 작을 때는 Link cost와 거의 비슷한 패킷 처리량을 보여준다. K가 2n, 3n 씩 증가하면 점점 패킷 처리량이 평균에 수렴해가는 것을 볼 수 있다. 이것을 표준편차로 나타내면 표 1과 같은 결과가 나온다. 표준편차는 그 값이 작을수록 평균에 수렴한다는 것을 의미하는데, SPF와 K가 1, n일 때는 거의 비슷한 값을 가지는데 비해 2n 이상이 되면 그 값이 크게 감소하는 것을 볼 수 있다.

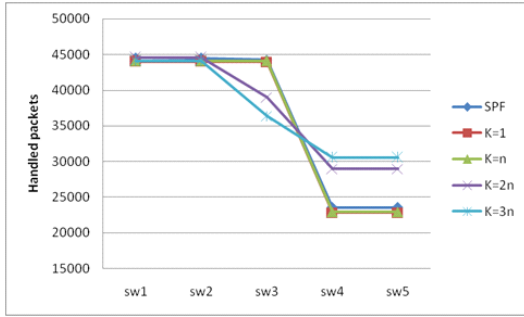


그림 3. 라우터별 처리 패킷량(K 가중치)
Fig. 3. Average handled packets per router(K weight)

표 1. 가중치 K별 처리패킷량 표준편차
Table 1. Standard deviation of handled packets by weight K

	SPF	K=1	K=n	K=2n	K=3n
Standard deviation	11435.73	11573.01	11600.58	7870.22	6770.79

하지만 K가 커질수록 Delay에 대한 성능이 떨어진다. 표 2는 K 값에 따른 Delay 측정 결과를 보여준다. 표는 총 10,000개의 패킷을 처리한 총 시간을 나눠 1개의 패킷이 처리되는 평균 시간을 나타낸다. K가 1일 때는 SPF 대비 약 97.6%의 성능을 보여주고, K가 n에는 96.3%, 2n에는 95.5%, 3n은 87%의 성능으로 측정되었다. 이를 통해 기존 SPF와 비슷한 성능을 내지만 네트워크 자원을 균등하게 활용할 수 있는 적절한 K의 값은 2n으로 확인하였다.

이제 앞서 구한 가중치 K의 값을 통해 제안 알고리즘과 기존 SPF와 Usage만 사용한 방법과의 결과를 비교 분석해보았다. 먼저 Usage의 결과값을 살펴보면, 기존의 SPF의 Link cost를 메트릭으로 한 결과는 그림 4와 같다. 토폴로지에서 각 링크는 동일한 Bandwidth를 가지고 있기 때문에 (1)-(3)-(2)의 Link cost는 2, (1)-(4)-(5)-(2)는 3으로 계산되어 패킷은 (1)-(3)-(2) 경로로 전달이 된다. 그리고 각 링크의 Bandwidth는 변화가 없기 때문에 모든 패킷은 (1)-(3)-(2) 경로로 전달된다. 그렇기 때문에 3번 라우터의 Usage가 4·5번 라우터보다 약 20,000개의 패킷을 더 처리한 것을 알 수 있다. 그리고 전체 처리한 패킷량을 전체 라우터 수로 나눈 평균 패킷량은 약

표 2. 가중치 K별 Delay 측정 결과
Table 2. Delay measurement result by weight K

Delay(ms)	SPF	K=1	K=n	K=2n	K=3n
Avg	0.0533	0.0546	0.0553	0.0558	0.0612

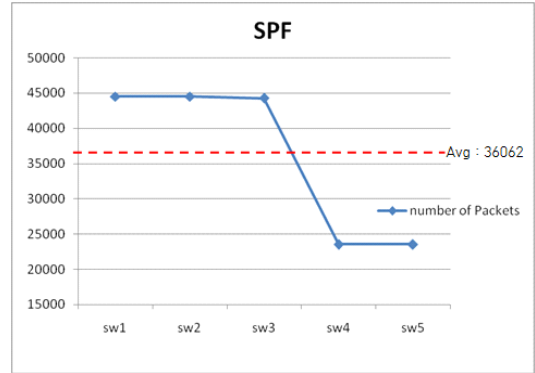


그림 4. 라우터별 처리 패킷량(SPF)
Fig. 4. Average handled packets per router(SPF)

36,000개인데 모든 라우터는 평균과 약 10,000개 정도씩 차이가 나며 표준편차는 약 11435.73의 값으로 계산된다.

제안 알고리즘의 결과는 그림 5와 같다. 앞서 SPF와 비교했을 때 3번과 4·5번 라우터간의 처리 패킷량의 격차가 20,000개에서 약 10,000개 정도로 줄어들었고, 각 라우터 전체적으로 처리한 패킷의 양이 평균에 수렴해가는 모습이 보였고 표준편차는 약 7870.229의 값으로 계산됐다. 이는 Link cost에 Usage를 더했을 때 기존 Link cost의 차이인 1을 기준으로 Usage 가중치에 따라 일정 패킷을 처리하면 (1)-(3)-(2) 경로에서 (1)-(4)-(5)-(2) 경로로 업데이트 되고, 반대로 일정 패킷을 처리하고 Usage가 갱신되면 (1)-(3)-(2) 경로로 다시 업데이트되는 현상이 발생하게 된다. 그렇기 때문에 Link cost 보다는 좀 더 전체 평균 패킷량에 좀 더 수렴한 것을 볼 수 있다.

마지막으로 Usage cost의 결과는 그림 6과 같으며, 3, 4, 5번 라우터의 처리 패킷량이 거의 동일해진 것



그림 5. 라우터별 처리 패킷량(Proposed)
Fig. 5. Average handled packets per router(Proposed)

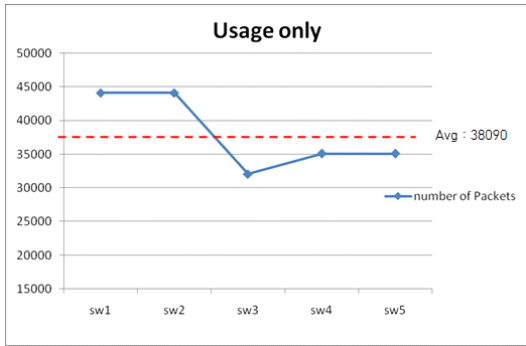


그림 6. 라우터별 처리 패킷량(SPF)
Fig. 6. Average handled packets per router(SPF)

을 볼 수 있다. 이는 Link cost가 없이 각 라우터가 처리한 패킷량만을 기준으로 경로를 선정하기 때문에 (1)-(3)-(2) 경로와 (1)-(4)-(5)-(2) 경로가 번갈아가면서 업데이트 되어 패킷이 처리됨에 따라 Usage가 거의 동일해지고 평균 패킷량에 전체적으로 수렴하게 되어 표준편차는 약 5649의 값으로 나타났다. 즉, 각 라우터의 처리 패킷량을 기준으로 네트워크 자원인 라우터의 균등한 사용에 유리한 방법은 Usage > Link+Usage > Link 순으로 볼 수 있다. 하지만 각 라우터의 Usage가 비슷해도 Delay에서 많이 차이가 나면 활용하기 어렵다. 표 3은 라우팅 메트릭에 따른 Delay 측정 결과를 보여준다. 표는 총 10,000개의 패킷을 처리한 시간을 나눠 1개의 패킷이 처리되는 평균 시간을 나타낸다.

각 10번씩 실험을 한 결과를 모두 작성하였으며 평균적으로 SPF는 1개의 패킷 당 0.0533ms의 Delay가 발생했으며, 제안 알고리즘은 1개 패킷 당 0.0558의

표 3. 각 메트릭별 Delay 측정 결과
Table 3. Delay measurement result for each metric

Delay(ms)	SPF	Proposed	Usage only
1st	0.05	0.05	0.063
2nd	0.049	0.05	0.048
3rd	0.057	0.044	0.048
4th	0.054	0.051	0.159
5th	0.052	0.055	0.064
6th	0.053	0.062	0.062
7th	0.056	0.058	0.159
8th	0.052	0.058	0.057
9th	0.052	0.065	0.064
10th	0.058	0.065	0.161
Avg	0.0533	0.0558	0.0885

Delay가, 마지막으로 Usage only는 1개 패킷 당 0.0885의 Delay가 발생한 것을 볼 수 있다. 즉 제안 알고리즘은 Link cost 대비 약 95.5%의 성능을 보여 주고, Usage only는 약 60%의 성능을 보여준다.

SPF와 제안 알고리즘은 거의 비슷한 Delay가 나온 반면, Usage only는 아무래도 앞선 메트릭들보다는 (1)-(3)-(2) 경로와 (1)-(4)-(5)-(2) 경로를 번갈아가며 비슷한 양의 패킷을 처리하다보니 Delay가 높은 것을 알 수 있다. 여기서 중요한 점은 제안 알고리즘의 Delay가 SPF와 큰 차이가 없이 사용자가 느끼지 못할 정도의 속도라는 점이다. 이는 Usage라는 메트릭을 도입하는 것이 네트워크 자원의 균등한 활용을 함과 동시에 Delay도 기존의 네트워크와 큰 차이가 없음을 보여준다.

실험 결과를 종합해보면 Usage만을 고려한다면 당연히 Usage 만을 사용하는 것이 유리하지만, Delay까지 고려한다면 제안알고리즘을 활용하는 것이 전체 네트워크의 자원을 균등하게 활용하면서 기존 네트워크의 Delay와도 큰 차이가 없는 네트워크를 구축할 수 있다는 것을 보여준다.

V. 결 론

본 논문에서는 기존 네트워크에서 네트워크 자원의 불균형적인 사용에 따른 문제를 해결하기 위해서 향후 표준화가 될 수 있는 세그먼트 라우팅에 대한 라우팅 알고리즘을 제시하였다. 먼저 실험을 통해 해당 알고리즘에 필요한 적절한 가중치 K에 대한 값에 대해 찾았으며 해당 알고리즘에 대한 성능을 평가하였다. 제안한 기법은 기존 Delay를 줄이려는 방법보다는 네트워크 자원을 모두 균등하게 활용하기 위한 방안으로 네트워크 관리 측면에서 효율을 높이는데 초점을 맞춘 것이지만, 실험 결과에서 볼 수 있듯이 Usage 메트릭을 추가하여도 Delay에는 큰 차이가 없지만 각 라우터의 처리 패킷량은 좀 더 평균에 수렴하여 전체 네트워크를 관리하는데 효율적인 것을 실험을 통해 확인할 수 있었다.

또한 어느정도 높은 Delay를 감수 할 수 있는 네트워크에서는 Usage만을 적용하는 것이 네트워크 관리 측면에서는 좀 더 효율이 있을 것으로 예상되는 결과도 확인할 수 있었다. 미래 네트워크는 초지연에 초점을 두고 Delay를 낮추는 기법들을 찾고 있지만, 반대로 초지연과 크게 연관이 없는 네트워크는 오히려 Usage를 활용한다면 네트워크의 자원을 관리하는데 이점이 있을 수 있다고 생각한다.

향후 연구로 세그먼트 라우팅에서는 여러 정책을 통한 라우팅이 가능한데 Usage를 정책에도 적용하여 특정 상황에서 Usage를 좀 더 가중치를 주고 적용하여 세그먼트를 생성하여 라우팅하는 기법을 연구하여 그 효과를 확인하고자 한다.

References

[1] C. Filsfils, et al., *Segment routing architecture*(2018), Retrieved Jul. 13, 2021, from <https://datatracker.ietf.org/doc/html/rfc8402>.

[2] C. Filsfils, et al., *Segment routing with MPLS data plane*, Retrieved Jul. 13, 2021, from <https://datatracker.ietf.org/doc/rfc8660>.

[3] *Software-Defined Networking (SDN) Definition*, Retrieved Jul. 13, 2021, from <https://opennetworking.org/sdn-definition/>.

[4] J. Y. Kim, "Study on segment routing technology using path computation element," *IEIE*, pp. 1742-1744, Jeju, Korea, Jun. 2015.

[5] A. Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini, and P. Castoldi, "SDN and PCE implementations for segment routing," *2015 20th Eur. Conf. Netw. and Optical Commun.(NOC)*, pp. 1-4, London, UK, 2015.

[6] Z. N. Abdullah, I. Ahmad, and I. Hussain, "Segment routing in software defined networks: A survey," *IEEE Commun. Surv. & Tuts.*, vol. 21, no. 1, pp. 464-486, 2019.

[7] S. Salsano and P. L. Ventre, *SDN-TE-SR-tools*, Retrieved Jul. 13, 2021, from <https://github.com/netgroup/SDN-TE-SR-tools>.

[8] Y. Choi and T. Kwon, "Design and implementation of The EIGRP based on the SDN," *J. Korea Multimedia Soc.*, vol. 22, no. 2, pp. 178-185, Feb. 2019.

[9] M.-C. Lee and J.-P. Sheu, "An efficient routing algorithm based on segment routing in software-defined networking," *Comput. Netw.*, vol. 103, pp. 44-55, 2016.

[10] L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano, and S. Salsano, "Traffic engineering with segment routing: SDN-Based architectural design and open source implementation," *2015*

Fourth Eur. Wkshps. Softw. Defined Netw., pp. 111-112, Bilbao, ESP, 2015.

[11] A. Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini, and P. Castoldi, "Experimental demonstration of segment routing," *J. Lightw. Technol.*, vol. 4, pp. 205-212, 2016.

[12] P. L. Ventre, et al., "Segment routing: A comprehensive survey of research activities, standardization efforts, and implementation results," *IEEE Commun. Surv. & Tuts.*, vol. 23, no. 1, pp. 182-221, 2021.

김 영 일 (Young-il Kim)



2013년 2월 : 육군사관학교 정보
과학과 졸업
2020년 3월~현재 : 국방대학교
컴퓨터공학과 석사과정
<관심분야> 네트워크, 라우팅
[ORCID:0000-0003-1182-1982]

권 태 옥 (TaeWook Kwon)



1986년 2월 : 육군사관학교 컴퓨
터공학과 졸업
1995년 2월 : 미국 해군대학원
컴퓨터공학과 석사
2001년 2월 : 연세대학교 컴퓨터
공학과 박사
2007년~현재 : 국방대학교 컴퓨
터공학과 교수

<관심분야> 네트워크, Sensor Network, CCN, SDN,
NFV
[ORCID:0000-0003-2880-9058]