

소프트웨어 정의 네트워크에서의 시민감 트래픽에 대한 유전 알고리즘 기반 최적 전송 경로 탐색

이 승 환*, 주 현 태*, 김 황 남^o

Discovering Optimal Transmission Path Based on Genetic Algorithm for Time Sensitive Traffic in Software-Defined Networks

Seunghwan Lee*, Hyeontae Joo*, Hwangnam Kim^o

요 약

시민감 트래픽은 송/수신 시 낮은 지터, 패킷 손실률, 지연 등의 요구사항을 만족하는 트래픽을 말한다. 이러한 요구사항을 만족하기 위해 네트워크 내에서 최적의 전송 경로를 찾는 것이 필수적이다. 본 논문에서 우리는 소프트웨어 정의 네트워크(SDN)를 통해 네트워크를 구성하고, 네트워크에 대한 전역적인 시각을 갖고 제어할 수 있도록 하였다. 다음으로, 유전 알고리즘(GA)을 활용하여 시민감 트래픽 전송을 위한 최적 경로를 탐색하는 알고리즘을 제안하였다. 그리고 이 알고리즘이 동적 계획법(DP)을 기반으로 탐색하여 도출한 실제 최적 경로 수준의 정확성을 가짐과 동시에, 시간적으로 매우 높은 효율성을 보임을 확인하였다. 마지막으로, 이 알고리즘을 통해 찾은 경로를 SDN 토폴로지에 적용할 시, 기존 제어기에서 제공하는 트래픽 전송 경로에 비해 지터, 패킷 손실률, 지연이 확연히 낮아짐을 확인하였다.

키워드 : 소프트웨어 정의 네트워크, 시민감 트래픽, 유전 알고리즘, 최적 경로

Key Words : Software Defined Network (SDN), time sensitive traffic Genetic Algorithm (GA), optimal path

ABSTRACT

Time sensitive traffic refers to traffic that meets requirements such as low jitter, low packet loss rate, and low delay in transmission and reception. To meet these requirements, it is essential to find the optimal transmission path within the network. In this paper, we construct a network in the form of Software-Defined Network (SDN), and through it, we enable control with a global perspective on the network. Next, we propose an algorithm to explore the optimal path for time sensitive traffic transmission using genetic algorithms (GA). In addition, it was confirmed that this algorithm showed very high efficiency in terms of time while having the level of accuracy of the actual optimal path derived by searching through dynamic programming (DP). Finally, when the path found through this algorithm is applied to the SDN topology, it was confirmed that the jitter, packet loss rate, and delay are significantly lower than that of the transmission path provided by the controller.

* 본 연구는 방위사업청과 국방과학연구소가 지원하는 군집형 무인 CPS 특화연구실 사업의 일환으로 수행되었습니다(UD190029ED). 또한 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구입니다(No. 2020R1A2C1012389).

• First Author : Korea University Dept. of Smart Convergence, longkid1@korea.ac.kr, 학생회원

o Corresponding Author : Korea University Dept. of Electrical Engineering, hnkim@korea.ac.kr, 종신회원

* Korea University Dept. of Electrical Engineering, motern800@korea.ac.kr, 학생회원

논문번호 : 202303-050-B-RN, Received March 13, 2023; Revised April 24, 2023; Accepted May 2, 2023

I. 서 론

사이버 물리 시스템(Cyber Physical Systems: CPS)은 컴퓨팅 및 통신 코어에 의해 제어되는 물리 엔지니어링 시스템이다^[1]. 최근 CPS는 제조업, 에너지, 방산, 드론 등 다양한 산업의 분야들에 접목되고 있다. 특히, 드론은 최초의 군사적 목적 외 다양한 산업 및 학술적 목적으로 그 용도가 확장되고 있다^[2]. CPS는 연관된 여러 물리적 프로세스들을 제어하는 여러 센서, 액추에이터 및 CPS 컨트롤러들로 이루어져 있다. 그리고 이러한 센서나 액추에이터 등에 대한 정보들을 전송하기 위해 안정적인 기반 네트워크가 필요하다. 이는 지상이 아닌 공중에 무선 네트워크의 형태로 구현되기도 한다^[3]. 그리고 이러한 CPS 네트워크에서 처리되는 트래픽 중 상당수는 시민감 트래픽이며, 이는 지터(jitter), 패킷 손실율(packet loss rate), 지연(delay) 등에 의해 영향을 받을 수 있음을 의미한다^[4]. 그러므로, CPS 네트워크는 데이터 송수신 시 반드시 낮은 지터, 패킷 손실율, 지연 등의 요구사항을 만족해야 한다. 이러한 요구사항을 만족하기 위해 이러한 요소들을 최소화하는 최적의 트래픽 전송 경로를 찾는 것이 반드시 필요하다. 이를 위해 전체 네트워크 토폴로지의 구조 및 각 노드와 링크들의 상태에 대한 인지가 필요하며, 네트워크에 대한 유연한 설정이 가능해야 한다.

소프트웨어 정의 네트워크(Software-Defined Networks: SDN)를 활용하는 것은 이에 대한 적합한 솔루션이 될 수 있다. SDN은 네트워크의 제어부(control plane)와 전송부(data plane)를 분리하고, 제어부에서 프로그래밍된 네트워크 설정을 전송부에 직접적으로 제공한다^[5]. 즉, 모든 시민감 네트워크 트래픽과 네트워크 토폴로지에 대한 전역적 시각을 갖고 있는 하나의 제어기(controller)가 존재하며, 이를 통해 트래픽 전송 경로를 정하고 설정할 수 있다^[6]. 이는 중앙 집중화된 방식으로 전역적 시각을 갖고 전체 네트워크를 제어할 수 있게 한다. 이러한 네트워크에 대한 제어기의 완전한 인지는 네트워크의 트래픽 흐름을 최적으로 관리할 수 있게 한다^[7]. 네트워크의 설정에 대한 내용이 플러그 인 애플리케이션(app)의 형태로 제어기로부터 전체 네트워크에 배포되므로, 유연하고 동적인 네트워크 설정이 가능하다^[8]. 그림1은 H1에서 H2를 향해 전송된 트래픽이 S1-S3-S5의 경로를 통해 전송되도록 설정된 SDN 토폴로지의 예시이다.

다음으로, 시민감 트래픽의 전송을 위한 최적 경로 도출이 필요하다. 이를 위해 본 논문에서는 유전 알고리즘(Genetic Algorithms: GA)을 활용한 시민감 트래픽

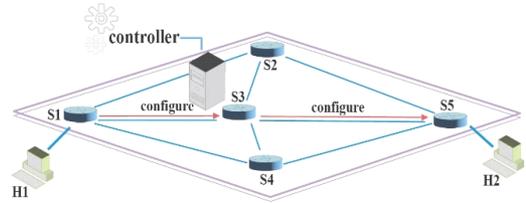


그림 1. SDN 네트워크 토폴로지
Fig. 1. An SDN topology

전송 경로 탐색 알고리즘을 제안한다. 지터, 패킷 손실율, 지연의 3가지 지표에 대해 각 목적식(objective function)을 정의하고, 이에 대한 가중합(weighted sum)을 통해 어떤 경로에 대한 비용(cost)을 정의한다. 그리고 이 비용을 최소화하는 것으로 문제를 정의한다.

본 논문의 2장에서는 관련 선행 연구 사례들과 유전 알고리즘의 개요에 대해 간단히 소개하고, 유전 알고리즘을 활용한 시민감 트래픽 전송을 위한 최적 경로 도출 알고리즘을 제안한다. 3장에서는 알고리즘의 성능을 평가한다. 동적 계획법(Dynamic Programming: DP)^[9]을 활용한 완전 탐색을 통해 도출되는 실제 최적 경로와 제안 알고리즘을 통해 도출되는 최적 경로 간의 비교를 수행한다. 그리고 제안한 알고리즘을 통해 도출되는 경로를 SDN 토폴로지에 적용하고, 기존 SDN 제어기 내의 app을 통한 트래픽 전송 경로와 제안한 알고리즘에 의한 전송 경로를 비교함으로써, 제안 알고리즘의 효용성을 확인한다. 마지막 4장에서 본 논문의 전체 내용을 정리하고 향후 연구 방향에 대해 제시하며 논문을 마무리한다.

II. 본 론

2.1 관련 연구

본 장에서는 네트워크 트래픽 전송 경로 제시에 관한 선행 연구 사례들을 소개한다. 또한, 시민감 트래픽을 성공적으로 전송하기 위한 연구 사례들도 함께 소개한다.

주헌태 외 3인(2022)^[10]은 심층 강화학습을 적용한 대역폭 할당을 통해 SDN 환경 내에서 시민감 트래픽 전송이 성공적으로 이루어질 수 있도록 하는 연구를 수행했다. SDN 제어기의 학습을 통해 시민감 트래픽과 일반 트래픽이 함께 전송되는 SDN 환경 내에서 일반 트래픽의 전송을 방해하지 않으면서 시민감 트래픽의 성공적인 전송을 보장하도록 하였다. Wang 외 7인(2021)^[11]은 향상된 개미 집단 최적화(Improved Ant

Colony Optimization: IACO)를 활용하여 시민감 트래픽을 스케줄링하는 연구를 수행하였다. 이 연구는 이를 통해 IEEE Time-Sensitive Networking Working Group^[12]의 시민감 네트워킹(Time-Sensitive Networking: TSN) 요구 조건을 만족한다고 주장한다. 다만, 이 연구들은 보다 복잡하고 다양한 노드 간 링크 상태들이 존재하는 네트워크 토폴로지에 대한 고려가 이루어지지 않았다. 따라서 이 연구들의 방법론을 보다 복잡한 네트워크 토폴로지에 적용했을 시에도 결과가 적절히 도출될 것이라 확신하기 어렵다.

한편, 네트워크 트래픽을 보다 효율적으로 전송하기 위한 경로를 제시하기 위한 연구는 다양하게 이루어져 왔다. Obeidat와 Al-shlabi(2022)^[13]는 유전 알고리즘을 활용하여 네트워크 트래픽의 전송 경로를 도출하는 알고리즘을 제안하였다. 이는 트래픽 전송 시 발생하는 지연을 줄이고, Dijkstra 알고리즘 등을 활용한 최단 경로 도출 알고리즘에 비해 계산 복잡도를 낮추는 효과가 있음을 보였다. Irfan 외 3인(2019)^[14]은 Dijkstra 알고리즘을 활용하여 SDN 토폴로지 내에서 트래픽 최단 전송 경로를 도출하여 지연을 낮추는 라우팅 알고리즘을 제시 하였다. 그리고 이를 실제 SDN 제어기 내 App으로 개발하여 적용함으로써, 기존 제어기 내 App을 통한 전송 경로에 비해 지연을 낮추는 효과가 있음을 확인하였다. 위의 연구들은 트래픽 전송 시 지연을 낮추는 데 목표가 있는 연구들이다. 시민감 트래픽은 지연 뿐 아니라 지터, 패킷 전송률에도 매우 민감한데, 위의 연구들에 지연 외의 지표들이 고려되지 않았다. 따라서, 위 연구들이 시민감 트래픽 전송에 적합한 솔루션을 제시해 준다고 확신할 수 없다.

본 논문에서는 유전 알고리즘을 활용하여 시민감 트래픽의 성공적인 전송을 위한 최적 전송 경로를 도출하는 알고리즘을 제안한다. 시민감 트래픽의 성공적인 전송을 위해서는 지연 뿐만 아니라 지터, 패킷 전송률이 모두 감소해야 한다. 따라서, 이후 장들에서 제안 알고리즘을 통해 도출되는 전송 경로를 통해 트래픽을 전송할 시 지터, 패킷 전송률, 지터가 모두 감소함을 실험을 통해 증명한다.

2.2 유전 알고리즘의 개요

유전 알고리즘(GA)은 광범위한 최적화 문제에 적용시킬 수 있는 휴리스틱(heuristic) 탐색 접근법이다^[15]. GA를 통해 해를 탐색하는 과정은 크게 초기 세대(initial population) 해집합 형성, 선택(selection), 교배(crossover), 돌연변이(mutation)로 이루어져 있다^[16-18]. 이 과정들을 통해 매번 새로운 세대의 후보 해집합을

형성하고, 세대를 거듭하며 해집합은 최적해에 가깝게 진화하게 된다. 그리고 전체 세대 수만큼의 해집합이 생성 되면 마지막 세대의 해집합에서 해를 선택하고 탐색을 종료한다. 이 때, 각 세대의 해집합 내의 후보 해에 해당하는 원소들을 염색체(chromosome), 그 후보 해를 이루는 각 요소들을 유전자(gene)라고 명명한다^[19]. 즉, 본 논문에서는 각 후보 경로들이 염색체이며, 각 경로 내에 속하는 노드들이 유전자라 할 수 있다.

선택(selection)은 어떤 세대의 유전자들을 적합도(fitness)에 따라 평가하고 다음 세대의 유전자들의 집합을 생성하기 위해 현재 세대의 유전자들 중 일부를 선택하여 복사하는 과정이다. 다시 말해, 교배(crossover)과 돌연변이(mutation) 연산을 수행하기 위한 염색체를 선택하는 과정이다. 이 과정을 통해 적합도가 높은 염색체들은 자식(offspring) 염색체를 생성하여 최적해에 포함되어 있을 가능성이 높은 자신의 우성 유전자들을 다음 세대에 상속할 확률이 높아진다.

교배(crossover)는 선택된 두 개의 부모 염색체들이 공통으로 갖고 있는 유전자 중 하나를 임의로 교차점으로 정한 후, 이를 기준으로 나뉘어진 각 부염색체(sub-chromosome)들을 교환하여 자식 염색체를 생성하는 과정이다^[18].

그림 2는 교배 과정을 나타낸 예시이다. 두 부모 염색체들은 3과 5의 2개의 공통 유전자를 갖고 있다. 이들 중 임의로 3을 선택하여 교차점으로 정하고 이를 기준으로 각 부모 염색체들의 부분 염색체들을 교환하여 새로운 자식 염색체를 생성하는 과정을 보여준다. 교배를 통해 염색체들은 다음 해집합 세대를 이루는 자식 염색체를 생성한다. 이는 세대를 거듭할수록 적합도가 높아져 최적해에 가까운 염색체가 선택될 수 있도록 도움을 준다.

돌연변이(mutation)는 어떤 염색체 내의 하나 이상의 유전자를 임의로 선택하여 그 값을 변경하는 연산을 말한다. 이는 다양한 유전자를 가진 염색체들을 만들어

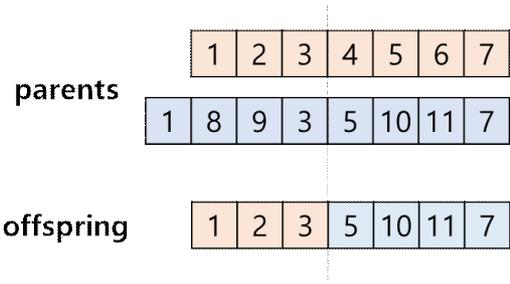


그림 2. 교배(crossover) 연산
Fig. 2. A crossover operation

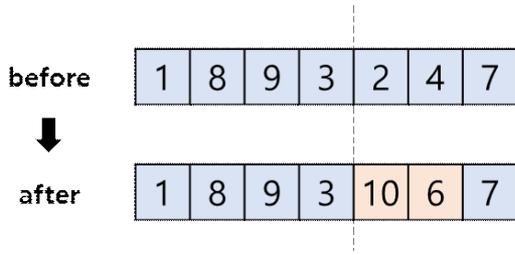


그림 3. 돌연변이(mutation) 연산
Fig. 3. A mutation operation

낼 수 있게 함으로써, 전체 공간을 탐색할 수 있도록 한다¹⁹⁾. 즉, 탐색 과정에서 지역 최적값(local optimum)에 빠져 실제 최적해를 찾지 못하게 되는 문제를 방지해주는 것이다. 그림 3은 어떤 염색체 내의 2, 4의 값을 가진 유전자들을 선택하여 이를 10, 6의 값으로 바꾸는 돌연변이 연산을 보여준다.

2.3 제안 최적 경로 탐색 알고리즘

2.3.1 문제 정의

본 논문은 시민감 트래픽의 전송을 위한 최적 경로를 찾는 네트워크 라우팅에 관한 문제를 다룬다. 이를 위해 네트워크 토폴로지를 그림4와 같이 비방향 그래프 $G = (V, E)$ 의 형태로 치환한다. 이 때, 네트워크의 노드(node)와 노드 간 링크(link)들이 각각 정점과 간선에 대응된다. 출발 정점을 S , 목적지 정점을 D 로 가정했을 때, “ S 에서 D 로 향하는 2번 이상 중복된 정점이 없는 경로”들의 집합을 해집합 \mathbb{P} 로 정의한다. 그림 4에서는 $S - V_1 - V_2 - V_3 - D, S - V_1 - V_2 - D$ 등이 해집합 \mathbb{P} 의 원소들에 해당 해당한다고 할 수 있다.

한편, 앞서 언급한 것처럼, 어떤 간선 $E = (V_i, V_j)$ 는 네트워크 노드 간 링크에 대응된다. 그러므로, 그래프 $G = (V, E)$ 의 각 간선에는 지터, 패킷 손실율, 지연의 정보가 부여된다. 이 때, 지연은 전송 노드에서 패킷이 전송된 시간부터 목적지 노드가 패킷을 완전히 수신한 시간 간격을 의미한다²⁰⁾. 지터는 전송되는 패킷들에 대한 지연 변동값으로 정의된다²¹⁾. 패킷 손실은 전송

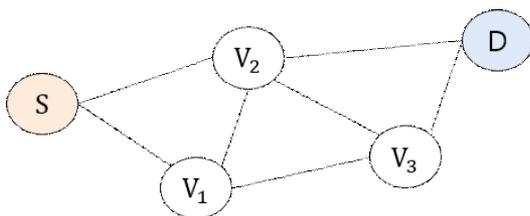


그림 4. 비방향 그래프 $G = (V, E)$
Fig. 4. An undirected graph $G = (V, E)$

노드에서 목적지 노드로 전송된 패킷이 전송에 실패함을 의미²²⁾하고, 패킷 손실율은 전송된 여러 개의 패킷들 중 이러한 손실이 발생한 비율을 의미한다. 그리고 본 논문에서 최종적으로 해결하고자 하는 문제는 이러한 지터, 패킷 손실율, 지연을 최소화할 수 있는 그래프 상 경로를 찾는 문제로 정의할 수 있다.

2.3.2 적합도 함수(fitness function) 정의

본 절에서는 유전 알고리즘을 활용하여 2.2.1절에서 정의한 문제에 대한 최적해를 찾기 위해 적합도 함수(fitness function)를 정의한다. 적합도 함수는 어떤 염색체의 선택에 대한 적합도를 평가하기 위한 함수로서, 유전 알고리즘의 성능을 좌우하는 매우 중요한 요소이다¹⁸⁾.

먼저, 2.2.1절에서 언급했던 “출발 정점 S 와 목적지 정점 D 를 연결하는 2번 이상 중복된 정점이 없는 경로”는 염색체에 해당하며, 이들의 집합인 \mathbb{P} 는 하나의 세대에 해당한다. 다음으로, 그래프의 각 간선마다 부여된 지터, 패킷 손실율, 지연의 3가지 값들 각각에 대하여 목적식(objective function)을 정의한다. 그리고 이들의 가중합(weighted sum)을 통해 어떤 염색체에 대한 비용(cost)을 정의할 수 있다.

어떤 염색체 $P = P_0 - P_1 - P_2 - \dots - P_n \in \mathbb{P}$ 에 대하여 지터에 대한 비용을 나타내는 목적식 $C_j(P)$ 는 수식 (1)과 같이 정의한다. 수식 (1)에서 θ_1 는 지터에 대한 최대 허용치를 나타낸다. 본 논문에서 θ_1 는 10으로 정한다. 그리고 $J(P)$ 는 P 에 속하는 간선들에 대한 지터들의 총합으로 정의한다.

$$C_j(P) = \begin{cases} \exp\left(\frac{J(P) - \theta_1}{\theta_1}\right) & \text{if } J(P) > \theta_1, \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$J(P) = \sum_{i=0}^{n-1} j(P_i, P_{i+1})$$

다음으로, 패킷 손실율에 대한 비용을 나타내는 목적식 $C_L(P)$ 은 수식 (2)와 같이 정의한다. 수식 (2)에서 θ_2 는 패킷 손실율에 대한 최대 허용치를 나타낸다. 본 논문에서 θ_2 는 0.1로 정한다. 여기서 $L(P)$ 는 P 에 속하는 간선들에 부여된 패킷 손실율의 총합으로 정의한다.

$$C_L(P) = \begin{cases} \exp\left(\frac{L(P) - \theta_2}{\theta_1}\right) & \text{if } L(P) > \theta_2, \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$L(P) = \sum_{i=0}^{n-1} l(P_i, P_{i+1})$$

지연에 대한 비용을 나타내는 목적식 $C_D(P)$ 은 수식 (3)과 같이 정의한다. 수식 (3)에서 θ_3 은 지연에 대한 최대 허용치를 나타낸다. 한편, 지연의 경우 본 논문에서는 지터와 패킷 손실율에 비해 상대적으로 후순위로 고려하는 값이므로, 목적식에서의 비용도 후자들과 달리 선형적으로 증가하도록 설계한다. 또한, 지연은 지터와 패킷 손실율과 달리 본 논문에서 성능 평가를 위해 명시적으로 정하는 최대 허용치가 존재하지 않는다. 그러므로 θ_3 은 0으로 정한다. $D(P)$ 은 P에 속하는 간선들에 부여된 지연들의 총합으로 정의한다.

$$C_D(P) = \begin{cases} D(P) - \theta_3 & \text{if } D(P) > \theta_3 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$D(P) = \sum_{i=0}^{n-1} d(P_i, P_{i+1})$$

다음으로, 수식 (4)와 같이 이들에 대한 가중합을 통해 어떤 염색체 P에 대한 비용 $C(P)$ 를 정의한다. 수식 (4)에서 w_1, w_2, w_3 은 지터, 패킷 손실율, 지연에 대한 비용의 상대적인 중요도에 따라 부여되는 가중치이다.

$$C(P) = w_1 C_J(P) + w_2 C_L(P) + w_3 C_D(P) \quad (15)$$

본 논문에서는 지연값을 지터, 패킷 손실율에 비해 후순위로 고려하였다. 따라서, 본 논문의 실험에서는 $w_1 = 2, w_2 = 2, w_3 = 1$ 의 가중치를 각각 부여하였다. 마지막으로, 염색체 P에 대한 적합도 함수 f_P 및, 본 논문에서 해결하고자 하는 문제의 최종 목표는 아래 수식 (5)와 같이 정의할 수 있다.

$$f_P = (C(P))^{-1} \quad (5)$$

maximize(f_P), subject to $P \in \mathbb{P}$

이는 세대 내 비용의 역수가 제일 큰 염색체를 선택하는 것이 목표임을 나타낸다. 이는 다시 말하면, 비용이 가장 낮은 염색체를 선택하는 최적화 문제를 말한다 고 할 수 있다.

2.3.3 최적 경로 탐색 알고리즘 설계

본 절에서는 앞선 2.2.1절과 2.2.2절에서 정의한 내용을 바탕으로 최적 경로 탐색 알고리즘을 설계한다. 선택(selection)을 통해 부모 염색체들을 생성하고, 교배(crossover)와 돌연변이(mutation) 연산을 통해 자식 염색체들을 생성한다. 이렇게 새로운 세대를 형성하면 해당 세대 내 최적 경로에 해당하는 염색체 및 그 비용

알고리즘 1. 알고리즘의 진행 과정
Algorithm 1. Processes of the algorithm

Algorithm 1: Processes of the algorithm

Inputs:

- G : total number of populations
- N : number of chromosomes per population
- p_1 : mutation probability
- \mathcal{P}_k : kth population
- \mathcal{S}_k : costs of chromosomes in \mathcal{P}_k
- P_k : kth chromosome in a population
- P_{child} : result of the crossover operation
- $P_{mutation}$: result of the mutation operation

Outputs:

- P^* : optimal path
- C^* : minimal cost

Steps:

1. INIT $C^* = infinite\ value$
 2. for $k = 1$ to G do
 3. SET $\mathcal{P}_k = kth\ population$
 4. **Get the costs of chromosomes in \mathcal{P}_k :**
 for chromosome in \mathcal{P}_k do
 $C_k = get\ k^{th}\ chromosome's\ cost$
 $\mathcal{S}_k.add(C_k)$
 end for
 5. **Update minimal cost if needed:**
 if $\min(\mathcal{S}_k) < C^*$ then
 $C^* = \min(\mathcal{S}_k)$
 $P^* = argmin_{P \in \mathcal{P}_k}(\mathcal{S}_k)$
 end if
 6. **selection operation:**
 $\mathcal{P}_{k+1} = selection(\mathcal{P}_k)$
 7. **crossover & mutation operation:**
 while $length(\mathcal{P}_{k+1}) \neq N$ do
 select 2 chromosomes : P_m, P_n
 $P_{child} = crossover(P_m, P_n)$
 $\mathcal{P}_{k+1}.add(P_{child})$
 with the probability of p_1 do
 $P_{mutation} = mutation(P_{child})$
 $\mathcal{P}_{k+1}.add(P_{mutation})$
 end while
 8. end for
-

을 업데이트하며, 이 과정을 전체 세대 수만큼 반복한다. 알고리즘의 진행 과정에 대한 의사 코드는 알고리즘 1과 같다.

설계된 알고리즘에서 탐색체는 출발 정점에서 목적지 정점까지 도달하는 경로를 나타낼 수 있도록 $P = P_0 - P_1 - P_2 - \dots - P_n$ 와 같은 구조의 순서쌍으로 표현된다. 유전자는 실제 존재 가능한 경우만 고려되며, 각 유전자의 길이는 다를 수 있다.

한편, 탐색체의 생성은 아래 알고리즘 2에서 제시된 과정을 통해 이루어진다. 이는 초기 세대를 생성하거나 돌연변이 연산을 수행하기 위해 필요한 과정이다.

알고리즘 2. 임의의 경로 생성
Algorithm 2. A random path generation

```

Algorithm 2: A random path generation
Inputs:
 $P_C$  : current path

Outputs:
 $P$  : a newly generated path

Steps:
1. INIT  $i = \text{length}(P_C)$ 
2. while  $P_i \neq \text{dst vertex}$  do
3.   if nodes connected to  $P_i = \emptyset$  then
        $P_C = \text{NONE}$ 
     end if
4.   for node in nodes connected to  $P_i$ 
       if NOT node  $\in P_C$ 
          $P_C.add(\text{node})$ 
       end if
     end for
5.    $i = i + 1$ 
6. end while
    
```

III. 실험 및 성능 평가

3.1 제안 알고리즘의 그래프 적용

본 장에서는 본 논문에서 제안한 알고리즘을 그림 5와 같이 SDN 토폴로지를 치환한 그래프에 적용하고, 이에 대한 성능 평가를 수행한다. 각 그래프의 간선에는 지터(ms), 패킷 손실율(%), 지연(ms)의 속성을 부여한다. 또한, 다양한 크기의 SDN 토폴로지에 대한 제안

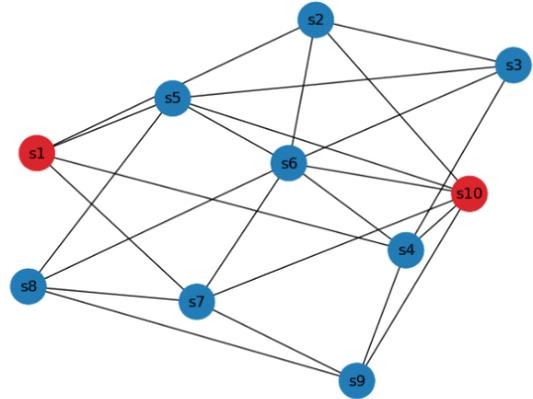


그림 5. 정점 10개 그래프
Fig. 5. A graph of 10 vertices

알고리즘의 확장성을 확인하기 위해 노드 10개, 30개, 50개가 존재하는 3개의 그래프들에 각각 알고리즘을 적용한다.

성능 평가를 위해 동적 계획법(DP)를 활용한 완전 탐색을 통해 도출되는 실제 최적 전송 경로와 제안 알고리즘을 통해 도출되는 최적 전송 경로를 비교한다. 이는 2장 2.3.2절의 수식 (4)를 통해 계산되는 경로에 대한 비용을 비교함으로써 수행한다. 이를 통해 제안 알고리즘을 통해 도출되는 최적 전송 경로의 정확도를 평가할 수 있다. 제안 알고리즘은 각 그래프에 100번씩의 시뮬레이션을 수행한 결과의 평균을 활용한다. 결과는 아래의 그림 6 및 표 1과 같다. 표 1에서의 Accuracy는 DP를 통해 도출되는 경로에 대한 비용 대비 GA를 활용한 제안 알고리즘을 통해 도출되는 경로에 대한 비용의 비율을 백분율(%)로 나타낸 것으로, 제안 알고리즘의 경로 정확도를 나타낸다.

다음으로, DP를 활용하여 실제 최적 전송 경로를 도

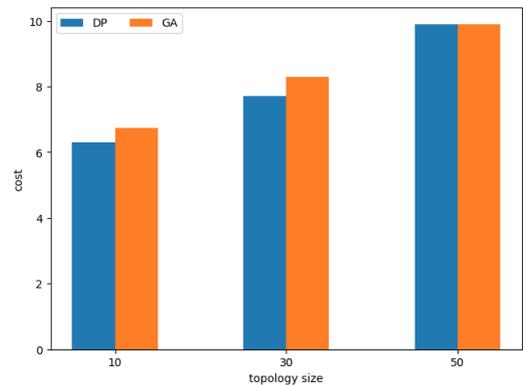


그림 6. 제안 알고리즘의 경로 정확도
Fig. 6. Path accuracy of the proposed algorithm

표 1. 제안 알고리즘의 경로 정확도
Table 1. Path accuracy of the proposed algorithm

Topology size	Algorithm		Accuracy (%)
	DP	GA (proposed)	
10	6.3	6.75	92.86
30	7.7	8.3	92.21
50	9.9	9.9	100

출하는 데 소요되는 시간과 제안 알고리즘을 활용하여 최적 전송 경로를 도출하는 데 소요되는 시간을 비교한다. 시간의 단위는 초(seconds)이다. 이를 통해 제안 알고리즘의 시간적 효율성을 평가한다. 제안 알고리즘은 100번의 시뮬레이션을 수행한 결과의 평균을 활용한다. 그 결과는 그림 7 및 표 2와 같다. 그림 7에서의 그래프에서는 시각적으로 직관적인 비교를 위해 DP와 GA를 활용한 제안 알고리즘에 소요된 시간에 각각 로그(log)를 취함으로써, log scale로 표현되었다. 표 2에서의 Efficiency는 DP를 통한 경로 도출 시간 대비 GA를 활용한 제안 알고리즘을 통한 경로 도출 시간의 비율을 백분율(%)로 나타낸 것으로, 제안 알고리즘의 시간적 효율성을 나타낸다.

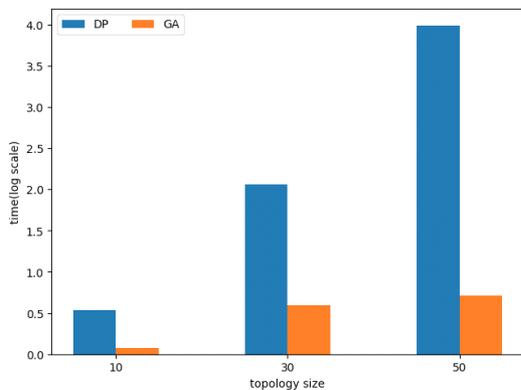


그림 7. 제안 알고리즘의 시간적 효율성 그래프
Fig. 7. Time efficiency of the proposed algorithm

표 2. 제안 알고리즘의 시간적 효율성
Table 2. Time efficiency of the proposed algorithm

Topology size	Algorithm		Efficiency (%)
	DP	GA (proposed)	
10	3.43	1.19	65.30
30	114.68	3.91	96.59
50	9770.35	5.17	99.95

위의 결과들은 제안 알고리즘의 성능이 매우 우수함을 보여주고 있다. 제안 알고리즘을 통해 도출되는 전송 경로는 최소 92.21%에서 최대 100%의 높은 정확도를 보인다. 이는 제안 알고리즘에 의해 도출되는 전송 경로가 실제 최적 전송 경로와 사실상 동일한 수준의 정확도를 보임을 증명한다. 한편, 제안 알고리즘은 최적 전송 경로를 도출하는데 최소 65.30%에서 최대 99.95%의 시간 단축을 보인다. 이를 통해 제안 알고리즘이 비약적인 시간적 효율성 향상을 보인다는 것도 확인할 수 있다.

3.2 실제 SDN 토폴로지에의 적용

본 장에서는 제안 알고리즘을 그래프에 적용했을 때 도출한 최적 전송 경로를 실제 SDN 토폴로지에 적용하는 실험을 수행한다. 본 논문에서는 Mininet^[23] 에뮬레이터를 활용하여 SDN 토폴로지를 생성하고, 이를 ONOS^[24] 제어기에 연동하여 활용한다. 그림 8은 ONOS와 Mininet을 연동하여 생성한 SDN 토폴로지의 예시이다.

ONOS는 트래픽의 전송을 목적으로 개발된 Forwarding App (fwd)이 존재한다. fwd는 Hop-Count를 최소로 하는 경로를 선택하여 트래픽을 전송하도록 프로그래밍되어 있다^[14]. 먼저 fwd를 통해 기본적으로 정해진 전송 경로를 통해 트래픽을 전송할 시의 지터, 패킷 손실율, 지연을 각각 측정한다. 예를 들어, Mininet을 활용하여 총 N개의 스위치가 존재하는 SDN 토폴로지를 형성할 경우, 각 스위치는 s1에서 sN까지 순서대로 명명된다. 이 때, s1과 sN에 각각 연결된 가상 호스트 간의 데이터 전송을 통해 각 지표를 측정한다. 호스트 간 ping test를 통해 매 초 ms 단위의 측정값이 도출되는데, 이 값의 절반이 지연값에 해당한다. 또한, iperf3 툴을 활용하여 지터와 패킷 손실율을 측정한다.

다음으로, 제안 알고리즘을 통해 도출된 최적 전송 경로를 SDN 토폴로지에 적용하고, 이 경로를 통해 트

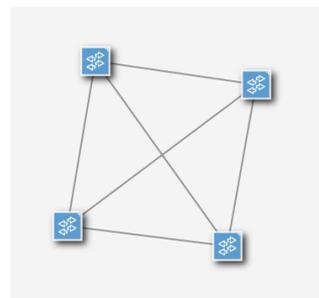
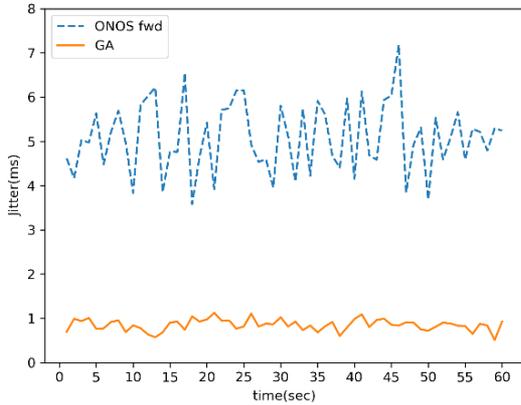
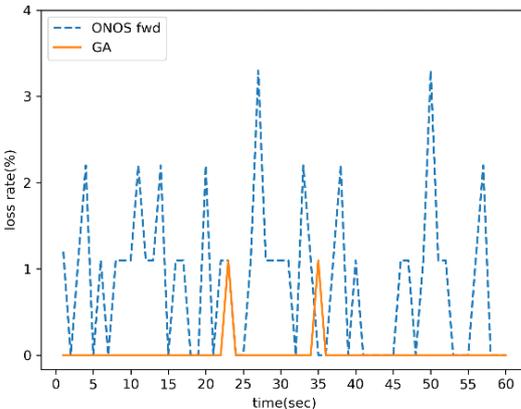


그림 8. ONOS-Mininet을 통해 생성한 SDN 토폴로지
Fig. 8. SDN Topology created with ONOS-Mininet

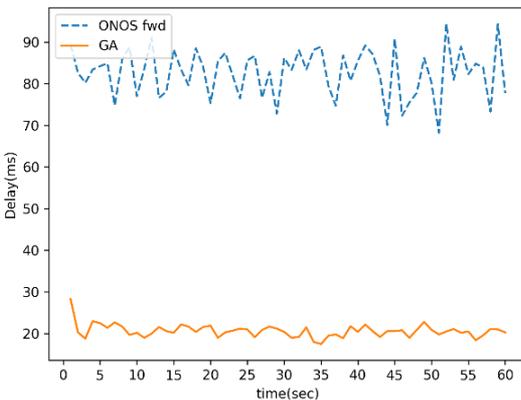
래픽을 전송할 시의 지터, 패킷 손실율, 지연을 측정한다. 이를 위해 3.1장의 실험 및 성능 평가에 활용되었던, 50개의 정점이 존재하는 그래프의 형태대로 Mininet을



(a) Comparison of jitter



(b) Comparison of packet loss rate



(c) Comparison of delay

그림 9. Fwd와 제안 알고리즘 간 각 성능 지표 비교
Fig. 9. Comparison of each performance indicator between paths fwd and proposed algorithm

활용하여 SDN 토폴로지를 형성한다. 그리고 ONOS를 활용하여 Mininet을 통해 형성한 SDN 토폴로지에 제안 알고리즘을 통해 도출된 최적 전송 경로를 적용한다. 이를 통해, fwd를 통해 기본적으로 정해지는 트래픽 전송 경로와 제안 알고리즘의 최적 전송 경로 간 지터, 패킷 손실율, 지연을 비교한다. 이를 통해 제안 알고리즘을 실제 SDN 토폴로지에 적용했을 시의 효용성을 평가할 수 있다. 아래 그림 9와 표 3에서 그 결과를 확인할 수 있다. 그림 9에서의 각 그래프는 iperf3와 ping test를 활용하여, fwd와 제안 알고리즘에 의해 정해진 경로에 대한 지터, 패킷 손실율, 지연을 60초 간 측정값의 양상을 비교하여 보여준다. 표 3에서의 Performance Indicator는 fwd를 통한 경로와 제안 알고리즘을 통한 경로에 대한 지터, 패킷 손실율, 지연의 60초 간 값의 평균이다. 그리고 Improvement rate는 fwd를 통한 경로 대비 제안 알고리즘을 통한 경로에 대한 각 지표의 비율을 백분율(%)로 나타낸 것으로, 제안 알고리즘의 각 지표에 대한 개선도, 즉 알고리즘의 성능을 나타낸다.

위의 결과들을 통해 제안 알고리즘을 통해 도출되는 최적 전송 경로를 적용 시 fwd에 비해 모든 지표에서 상당한 개선이 이루어짐을 확인할 수 있다. 특히, 패킷 손실율은 0.85%에서 0.04%로의 감소를 보이며 제안 알고리즘을 통해 본 논문에서 기준으로 정했던 0.1% 이내를 충족하게 됨을 보여 준다.

표 3. 각 성능 지표 개선
Table 3. Improvement of each performance indicator

Performance Indicator (60 sec avg)	Path		Improvement rate (%)
	fwd	GA (proposed)	
jitter	5.08	0.85	83.27
loss rate	0.85	0.04	95.29
delay	82.68	20.65	75.02

IV. 결론

본 논문에서는 유전 알고리즘을 활용하여 SDN 환경에서의 시민감 트래픽 최적 전송 경로를 도출하는 알고리즘을 제안하였다. 제안 알고리즘은 정확한 최적 전송 경로를 도출할 수 있음과 동시에 매우 높은 시간적 효용성을 가짐을 확인하였다. 또한, 제안 알고리즘을 통해 도출되는 최적 전송 경로를 실제 SDN 토폴로지 환경에 적용하여 지터, 패킷 손실율, 지연이 각각 83.27%,

95.29%, 75.02% 줄어듦을 확인하였다. 따라서, SDN 환경에서 시민감 트래픽 전송을 위해 제안 알고리즘을 활용할 경우 높은 효용성을 보일 것으로 기대된다.

다만, 본 논문에서의 실험은 실제 네트워크 환경이 아닌 Mininet을 통해 에뮬레이션된 환경에서 수행된 것이다. 실제 네트워크 환경은 에뮬레이션된 환경에 비해 트래픽이 동적으로 발생하며, 그에 따라 상대적으로 훨씬 예측하기 어렵다는 특성을 갖는다. 따라서, 이러한 변수가 존재하는 실제 네트워크 환경을 구성하고 제안 알고리즘을 적용할 시에도 에뮬레이션 환경만큼의 효용성을 보일지에 대한 검증이 필요하다. 따라서, 향후에는 가상화된 노드들이 아닌 실제 PC 등을 활용하여 네트워크 토폴로지를 구성하고, 트래픽 발생 확률에 대한 모델링을 포함하여 환경을 구성한 후 제안 알고리즘을 적용함으로써, 실제 네트워크 환경에서의 제안 알고리즘의 효용성에 대한 연구를 진행할 계획이다.

References

- [1] L. Monostori, et al., "Cyber-physical systems in manufacturing," *CIRP Annals*, vol. 65, no. 2, pp. 621-641, 2016.
(<https://doi.org/10.1016/j.cirp.2016.06.005>)
- [2] S. Yoo, et al., "Poster: A multi-drone platform for empowering drones' teamwork," *The 21st Annu. Int. Conf. MobiCom*, 2015.
(<https://doi.org/10.1145/2789168.2795180>)
- [3] S. Park, K. Kim, H. Kim, and H. Kim, "Formation control algorithm of multi-uav-based network infrastructure," *Appl. Sci.*, vol. 8, no. 10, pp. 1740, 2018.
(<https://doi.org/10.3390/app8101740>)
- [4] N. G. Nayak, F. Durr, and K. Rothermel, "Incremental flow scheduling and routing in time-sensitive software-defined networks," *IEEE Trans. Ind. Informatics*, vol. 14, no. 5, pp. 2066-2075, 2018.
(<https://doi.org/10.1109/TII.2017.2782235>)
- [5] D. Gopi, S. Cheng, and R. Huck, "Comparative analysis of SDN and conventional networks using routing protocols," *Int. Conf. CITS*, pp. 108-122, Jul. 2017.
(<https://doi.org/10.1109/CITS.2017.8035305>)
- [6] N. G. Nayak, F. Durr, and K. Rothermel, "Time-sensitive software-defined network (TSSDN) for real-time applications," *24th Int. Conf. Real-Time Netw. and Syst.*, pp. 193-202, Oct. 2016.
(<https://doi.org/10.1145/2997465.2997487>)
- [7] S. Sezer, et al., "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36-43, 2013.
(<https://doi.org/10.1109/MCOM.2013.6553676>)
- [8] K. Ahmed, J. O. Blech, M. A. Gregory, and H. Schmidt, "Software defined networking for communication and control of cyber-physical systems," *2015 IEEE 21st Int. Conf. Parallels and Distrib. Syst.*, Dec. 2015.
(<https://doi.org/10.1109/ICPADS.2015.107>)
- [9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th Ed., MIT Press, pp. 362-400, 2022.
- [10] H. Joo, S. Lee, S. Lee, and H. Kim, "Optimizing time-sensitive software-defined wireless networks with reinforcement learning," *IEEE Access*, vol. 10, pp. 119496-119505, 2022.
(<https://doi.org/10.1109/ACCESS.2022.3222060>)
- [11] Y. Wang, et al., "A time-sensitive network scheduling algorithm based on improved ant colony optimization," *Alexandria Eng. J.*, vol. 60, no. 1, pp. 107-114, 2021.
(<https://doi.org/10.1016/j.aej.2020.06.013>)
- [12] *IEEE Time-Sensitive Networking Task Group*, from <http://www.ieee802.org/1/pages/tsn.html>.
- [13] M. Obediat and M. Al-shalabi, "An efficient approach towards network routing using genetic algorithm," *Int. J. Comput. Commun. & Contr.*, vol. 17, no. 5, 2022.
(<https://doi.org/10.15837/ijccc.2022.5.4815>)
- [14] T. Irfan, R. Hakimi, A. C. Risdianto, and E. Mulyana, "ONOS intent path forwarding using Dijkstra algorithm," *2019 ICEEI*, Jul. 2019.
(<https://doi.org/10.1109/ICEEI47359.2019.8988853>)
- [15] O. Kramer, *Genetic Algorithm Essentials*, Springer, pp. 11-19, 2017.

- (https://doi.org/10.1007/978-3-319-52156-5_2)
- [16] X. Sun, J. Wang, W. Wu, and W. Liu, "Genetic algorithm for optimizing routing design and fleet allocation of freeway service overlapping patrol," *Sustainability*, vol. 10, no. 11, 2018.
(<https://doi.org/10.3390/su10114120>)
- [17] C. R. Reeves, *Genetic Algorithms*, in M. Gendreau, J. Y. Potvin, *Handbook of Metaheuristics*, Springer, 2009.
(https://doi.org/10.1007/978-1-4419-1665-5_5)
- [18] C. Ahn, R. S. Ramakrishna, and C. Kang, "A new genetic algorithm for shortest path routing problem," *J. KICS*, vol. 27, no. 14, pp. 1215-1227, 2002.
- [19] K. Sastry, D. E. Goldberg, and G. Kendall, *Genetic Algorithms*, in E. K. Burke, G. Kendall, *Search Methodologies*, Springer, 2014.
(https://doi.org/10.1007/0-387-28356-0_4)
- [20] Internet Engineering Task Force(IETF), *RFC 2679*(1999), from <https://www.rfc-editor.org/rfc/rfc2679>
- [21] Internet Engineering Task Force(IETF), *RFC 3393*(2002), from <https://www.rfc-editor.org/rfc/rfc3393>
- [22] Internet Engineering Task Force(IETF), *RFC 2680*(1999), from <https://www.rfc-editor.org/rfc/rfc2680>
- [23] *The Mininet documentation*(2021), Retrieved Mar. 02, 2023, from <https://github.com/mininet/mininet/wiki/Documentation>.
- [24] Open Networking Foundation(ONF), *ONOS Guides*(2016), Retrieved Mar. 02, 2023, from <https://wiki.onosproject.org/display/ONOS/Guides>.

이 승 환 (Seunghwan Lee)



2020년 8월 : 고려대학교 컴퓨터
학과 졸업
2022년 3월~현재 : 고려대학교
스마트융합학과 석사과정
<관심분야> 유무선 네트워크, 소
프트웨어 정의 네트워크
[ORCID:0009-0003-7623-1841]

주 현 태 (Hyeontae Joo)



2017년 2월 : 서울시립대학교 전
자전기컴퓨터공학부 졸업
2020년 3월~현재 : 고려대학교
전기전자공학과 석박사 통합
과정
<관심분야> 무인이동체, 무선 네
트워크, 가상화

[ORCID:0000-0002-3753-364X]

김 황 남 (Hwangnam Kim)



1992년 2월 : 부산대학교 컴퓨터
공학과 (공학사)
1994년 2월 : 서울대학교 컴퓨터
공학과 (공학석사)
2004년 2월 : 미국 Urbana-Cham-
paign 소재 Illinois 주립대학
컴퓨터공학과 (공학박사)

1994년~1999년 : LG전자 주임연구원
2004년~2005년 : 미국 Urbana-Champaign 소재
Illinois 주립 대학 Post Doctorate Fellow
2005년~2006년 : 삼성전자 책임연구원
2012년~2012년 : 미국 LA 소재 California 주립대학
방문연구원
2006~현재 : 고려대학교 전기전자공학과 교수
<관심분야> 모바일 컴퓨팅, 블록체인 플랫폼, 무인이동
체, 군집 지능

[ORCID:0000-0003-4322-8518]