

일정한 트래픽 패턴이 보이는 장소에서 인공지능을 통한 OSPF 프로토콜 개선 및 최적화

홍민, 채형주, 김형래, 정주명, 오혁준*
광운대학교, *광운대학교

kkh020329@kw.ac.kr, chj3023@kw.ac.kr, *hj_oh}@kw.ac.kr

Optimization of OSPF Protocol Using Artificial Intelligence in Environments with Predictable Traffic Patterns

Min Hong, Hyeongju Chae, Hyeongrae Kim, Joomyung Jung, Hyukjun Oh*
Kwangwoon Univ., *Kwangwoon Univ.

요약

본 논문은 시간대별로 트래픽 패턴이 변화하는 네트워크 환경에서 Open Shortest Path First(OSPF) 프로토콜의 자원 비효율 문제를 해결하기 위해 인공지능 기반 링크 cost 조정 기법을 제안한다. Cisco Modeling Labs 시뮬레이션과 Wireshark 기반 패킷 분석을 통해 트래픽 데이터를 수집하고, RandomForestRegressor를 활용해 링크 cost를 예측하였다. 링크 부하 분산 및 병목 구간 처리 능력이 향상되었으며, 평균 대역폭은 약 6.58MB/s 개선되었다.

I. 서 론

현대 네트워크 환경은 시간과 장소에 따라 급격하게 변화하는 트래픽 패턴에 직면해 있으며, 이에 따라 효율적인 트래픽 관리와 자원 활용이 점점 더 중요해지고 있다. 특히 공단 지역의 원룸촌이나 학교 주변과 같은 특수 지역에서는 시간대에 따라 트래픽 양이 크게 변동하기 때문에, 이를 고려한 정교한 라우팅 전략이 요구된다[1].

(OSPF)는 대표적인 링크 상태 기반의 동적 라우팅 프로토콜로, 네트워크 토플로지의 변화에 실시간으로 반응하여 경로를 재계산하는 기능을 제공한다[2]. 그러나 OSPF는 트래픽 양이나 부하 상황과 무관하게 일정 주기로 전체 라우팅 정보를 갱신하며, 모든 링크에 대해 동일한 방식으로 cost 계산을 수행한다. 이로 인해 트래픽이 적은 시간대에도 불필요하게 CPU 자원을 소모하고, 트래픽이 집중되는 시간대에는 혼잡을 효과적으로 완화하지 못하는 비효율성이 존재한다.

본 연구는 이러한 문제점을 해결하기 위해 인공지능 기반 트래픽 예측 및 cost 조정 기법을 제안한다. 트래픽 데이터를 분석하여 시간대별 패턴을 학습하고, 이를 기반으로 OSPF의 링크 cost 값을 조정함으로써 트래픽 분산을 유도하고 차원 소모를 최소화한다. 결과적으로 네트워크의 전반적인 성능과 효율을 향상시키는 것이 본 연구의 목표다.

실험은 Cisco Modeling Labs를 이용해 분기점이 존재하는 네트워크 토플로지를 구성하고, Wireshark 및 Python을 활용해 각 링크의 트래픽 데이터를 수집 및 분석한다. 평균 패킷 수, 전송 지연 시간 등 주요 지표를 feature로 사용하여 AI 모델을 학습시키고, 이를 통해 트래픽이 특정 링크에 집중되지 않도록 cost 값을 조절한다.

마지막으로, 제안하는 방식이 적용된 OSPF 네트워크와 기존 OSPF 네트워크 간의 CPU 사용량과 트래픽 분

산 효과를 비교 분석함으로써, 제안 기법의 실효성과 기여도를 검증한다.

II. 본론

1. 시뮬레이션 환경 및 네트워크 토폴로지 설계

초기에는 Cisco Packet Tracer를 이용해 시뮬레이션을 시도했으나, 패킷 데이터를 직접 추출하는 데 한계가 있어 CML로 전환하였다. 시간대별 트래픽 양의 차이를 시뮬레이션하기 위해 분기점이 존재하고 다수의 PC가 연결된 네트워크 토플로지 그림 1과 같이 설계하였다. 라우터 간 서브넷 분리, OSPF 설정, default route 설정 등을 통해 현실적인 시나리오를 구현하였고, 각 링크의 가용성과 라우팅 경로를 ping을 통해 확인하였다.

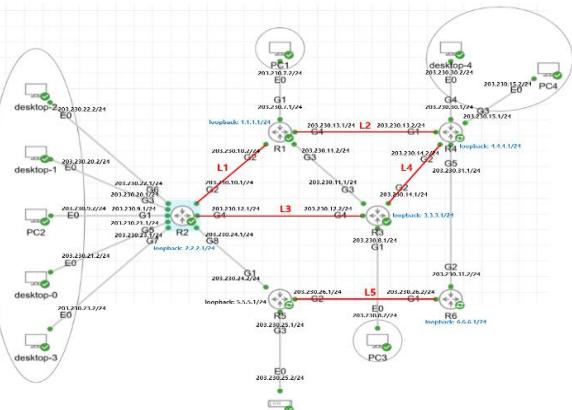


그림 1. 구성한 네트워크 토플로지

2. 트래픽 데이터 수집 및 주요 링크 식별

Wireshark를 통해 주요 링크(L1~L5)에 대한 패킷 데이터를 수집하고, 다양한 트래픽 패턴에 대해 시뮬레이션을

진행하였다. 시나리오는 그림 1과 같이 총 5가지로 분류되며, 각 케이스는 특정 노드 간의 통신 패턴에 따른 링크 부하의 변화를 나타낸다. 이러한 시나리오 기반 데이터는 각 링크별 평균 패킷 길이, 전송 시간 차이(diff), 시간당 바이트 수 등의 feature로 가공되었으며, 이를 통해 링크별 트래픽 상태를 정량화할 수 있었다.

3. 인공지능 모델을 통한 cost 예측

트래픽 데이터로부터 RandomForestRegressor 모델을 이용하여 각 링크의 최적 cost 값을 예측하였다[3]. 이 모델은 비선형적인 관계를 효과적으로 학습할 수 있어 링크 사용량과 효율적인 분산 간의 상관관계를 잘 반영하였다. 결과적으로 예측된 cost는 각 링크의 실시간 트래픽 수준을 반영하는 방향으로 설정되었다.

4. 개선된 OSPF와 기존 OSPF 비교

예측된 cost를 적용한 개선형 OSPF와 기존 OSPF 간의 트래픽 분산 효과를 비교하였다. 그림 2.에서 보이는 것처럼 기존 OSPF에서는 대부분의 트래픽이 L5 링크를 회피하고 짧은 경로에 집중되었으나, 그림 3.에서 보이는 것처럼 개선형 OSPF에서는 L5를 포함한 다양한 경로로 트래픽이 분산되었다. 또한, 각 라우팅 경로(route)에 대해 시간당 평균 전송량인 평균 대역폭 사용량(Average Bandwidth Usage)을 산출하였다. 이를 위해 패킷 전송 시작 간의 시간 차(Δt_i)와 각 패킷의 길이(L_i)를 기반으로 단위 시간당 전송량(B_i)을 계산하고 이를 평균내어 경로 전체의 평균 대역폭을 도출하였다.

$$\bar{B} = \frac{1}{N} \sum_{i=1}^N \left(\frac{L_i}{\Delta t_i} \right) \quad (1)$$

개선형 OSPF는 약 6.58MB/sec의 추가 이득을 보였으며, L5를 지나는 트래픽 양에서도 유의미한 차이를 확인할 수 있었다.

5. CPU 사용량 분석

비록 초기 목표 중 하나였던 CPU 사용량 감소는 cost 계산이 수동적이고 routing table 자체는 여전히 OSPF 구조를 따르기 때문에 큰 차이를 보이지는 않았지만, 트래픽 분산의 효과 측면에서는 확실한 개선이 이루어졌다.

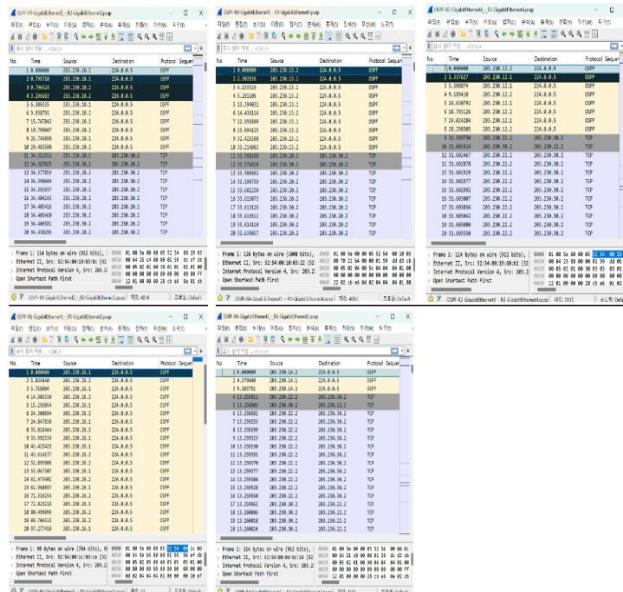


그림 2. 기존 OSPF에서 L1~L5의 Packet data

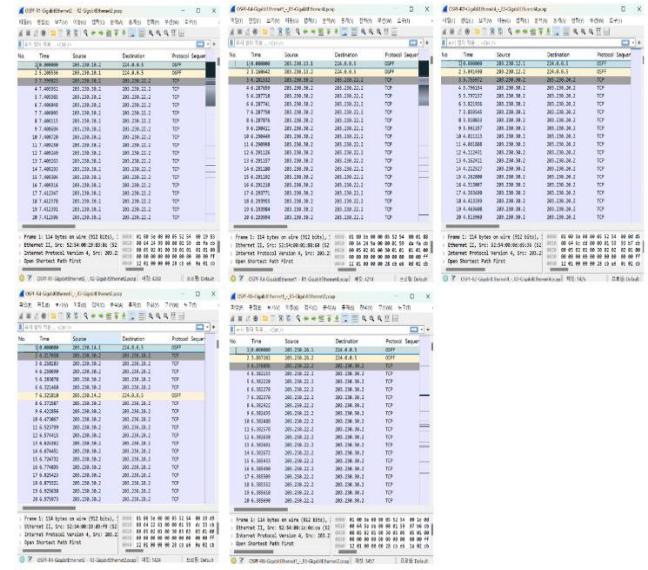


그림 3. 개선형 OSPF에서 L1~L5의 Packet data

III. 결론

본 연구는 인공지능 기반 트래픽 분석을 통해 OSPF 프로토콜의 링크 cost를 동적으로 조정함으로써, 네트워크 트래픽 분산의 효율성을 향상시키는 방안을 제안하였다. 다양한 시뮬레이션 시나리오와 트래픽 데이터 기반의 머신러닝 적용을 통해, 기존 OSPF보다 병목 구간의 해소와 링크 활용의 균형을 더욱 효과적으로 달성할 수 있음을 실험적으로 확인하였다. 특히, L5와 같은 경로가 활성화되면서 전체 트래픽이 집중되지 않고 고르게 분산되었으며, 평균 대역폭 사용률이 향상되는 결과를 얻었다. 그러나 본 방식은 트래픽 패턴이 일정한 시간적 규칙성을 보이는 환경에서 특히 효과적이며, 트래픽 변화가 불규칙하거나 예측 불가능한 환경에서는 적용 효과가 제한적일 수 있다는 점에서 한계를 가진다. 또한, cost 조정 자체가 외부에서 수동으로 이루어진 점, 그리고 라우팅 알고리즘 구조상 CPU 자원 소모를 근본적으로 줄이지는 못했다는 점도 향후 개선 과제로 남는다.

ACKNOWLEDGMENT

이 연구는 과학기술정보통신부의 지원을 받아 한국연구재단(NRF)의 연구사업(No. NRF-2021M1B3A3102358)을 통해 수행되었습니다.

참고 문헌

- [1] Y. Park et al. "Research Trends for Next-generation Radio Access Networks" 2025.
- [2] John Moy, "OSPF Version 2", RFC 2328, IETF, 1998. [\[https://datatracker.ietf.org/doc/html/rfc2328\]](https://datatracker.ietf.org/doc/html/rfc2328)
- [3] Leo Breiman, "Random Forests", Machine Learning, vol. 45, no. 1, pp. 5-32, 2001. DOI: 10.1023/A:1010933404324