https://doi.org/10.7840/kics.2023.48.4.413

확정적 네트워크에서의 동적 처리순위를 활용한 강화학습 기반 스케줄러

류 지 혜•, 박 규 동*, 권 주 혁**, 정 진 우°

Reinforcement Learning-Based Scheduler with Dynamic Precedence in Deterministic Networks

Jihye Ryu•, Gyudong Park*, Juhyeok Kwon**, Jinoo Joung°

요 약

스마트 인더스트리, 메타버스, 디지털 트윈, 군사용 어플리케이션 등에서 확정적 데이터 전달을 요구하고 있다. 본 논문은 일반적으로 통용되는 플로우들의 클래스 혹은 우선순위와는 별도로, 네트워크 상황과 중요도에 따라 플로우 별로 동적으로 처리순위(precedence)를 할당하고, 이에 따라 스케줄링 알고리즘을 결정하는 강화학습 기반의 스케줄링 프레임워크를 제안한다. 이를 실증하기 위해서 두 개의 처리순위 큐가 존재하는 환경을 상정하여, 강화학습 에이전트가 지정된 기준에 따라 플로우들의 처리순위를 지정하며 스케줄링 알고리즘을 선택하는 두 가지의 행동(action)을 취한다. 네트워크 특성에 따라 다양한 기준으로 처리순위를 결정할 수 있다. 본 연구에서는 플로우가 요구하는 마감기한(deadline)을 처리순위 결정의 중요한 기준으로 사용하였다. 딥러닝 기반의 강화학습 모델인 DDQN(Double Deep Q-Network)을 활용하여, 고정된 길이의 결정 주기마다 네트워크의 상태(state)를 관측하고 행동을 선택함으로써 처리순위를 결정한다. 본 연구의 환경에 맞게 개발한 네트워크 시뮬레이터를 통해 DDQN 에이전트가 여러 휴리스틱 알고리즘과 비교하여 높은 성능을 보이는 것을 확인하였다.

Key Words : reinforcement learning, deep learning, deterministic networking, Q-learning, double deep Q-network, precedence

ABSTRACT

Smart industry, metaverse, digital-twin, and military applications require deterministic data delivery in large scale networks. This paper proposes reinforcement learning-based scheduling that assigns dynamically different precedences to the flows, in addition to the flow's class or priority, and determines the scheduling algorithm according to the flow's precedence. In the proposed reinforcement learning-based scheduling algorithm with two precedence queues, the reinforcement learning agent takes two actions that assigns the precedence of flows according to a specified criterion and selects a scheduling algorithm. Depending on the purpose of the

※ 이 논문은 2022년 정부(방위사업청)의 재원으로 국방과학연구소의 지원을 받아 수행된 연구임(911194202)
• 상명대학교 지능정보공학과, Sangmyung University, Department of Artifical Intelligence & Informatics, 학생회원
° Corresponding Author : 상명대학교 휴먼지능정보공학과, Sangmyung University, Department of Human-centered AI, jjoung@smu.ac.kr, 정회원
* 국방과학연구소 국방첨단과학기술연구원 지휘통제체계단, Agency For Defense Development, Advanced Defense Science & Technology Research Institute - Command and Control Systems, 정회원
** 상명대학교 지능정보공학과, Sangmyung University, Department of Artifical Intelligence & Informatics, 학생회원
논문번호 : 202212-311-B-RN, Received December 29, 2022; Revised February 13, 2023; Accepted February 14, 2023

network, any factor with high importance could be a criterion for determining the precedence. In this study, the deadline required by the flow is designated as the major factor for precedence decision. By utilizing DDQN (Double Deep Q-Network), a deep learning-based reinforcement learning model, the precedence and the scheduling algorithm are determined by observing the state of the network and selecting an action at each decision period with a fixed length. In the network simulator developed for the study, it was confirmed that the DDQN agent showed better performance than various heuristic algorithms.

Ⅰ. 서 론

실시간성의 보장이 필요한 증강현실, 스마트 인더스트리, 군사용 네트워크와 같이 시간 민감형 네트워크를 목표로 확정적 네트워크와 관련된 연구가 진행되어오고 있다[1]. 본 연구에서는 군용 네트워크에서 제안된 처리순위(precedence)의 개념[2]을 활용하여, 타임슬롯 기반의 네트워크에서 플로우가 요구하는 마감기한을 보장하는 프레임워크를 개발한다. 본 연구에서 개발하는 스케줄링 프레임워크를 다양한 토폴로지나 크기의 네트워크에 적용할 수 있도록 고려한다.

처리순위는 같은 우선순위 혹은 같은 클래스의 플로우에 대해서 다양한 조건에 따라 동적으로 추가적인 우선권을 할당할 수 있는 개념이다. 노드에서의 큐 할당은 처리순위에 따라 이루어질 수 있다. 플로우의 처리순위가 높을수록 높은 우선순위의 큐에 인입되고, 선택된 스케줄링 알고리즘에 따라 서비스된다. 타임슬롯 기반의 네트워크 환경을 가정하지만, 매 타임슬롯의 초기 시점에서의 행동 선택이 이루어지려면 연산의 빈도가 매우 커지므로, 고정된 수의 타임슬롯 동안 선택된 처리순위 할당 알고리즘과 스케줄링 알고리즘을 유지하는 방법으로 연산 빈도를 낮추었다. 처리순위 결정과 스케줄링 알고리즘 결정의 두 개 action은 딥러닝 기반 강화학습모델인 DDQN 에이전트가 수행한다. 각 플로우 별 패킷 생성주기의 최소 공배수인 결정 주기(decision period)마다 DDQN의 학습 및 행동 선택이 이루어진다.

본 연구에서는 확정적 네트워크 기반 확장성 있는(scalable) 대규모 네트워크 구현을 위해 전체 네트워크의 상황이 아닌 개별 노드의 큐 상태만으로 패킷들의 요구사항을 지킬 수 있도록 처리순위 큐 할당, 스케줄링 알고리즘을 결정하는 DDQN 스케줄링 프레임워크를 제안한다. 전체적으로 DDQN의 학습은 그림 1과 같은 과정을 에피소드의 최대 수만큼 반복하며 이루어진다. 하나의 에피소드 시작인 구성(configuration) 단계에서는 트래픽을 동적으로 조성하기 위해 플로우를 구성하는 랜덤 파라미터들의 결정이 이루어진다. 이에 따라 패킷들의 생성 및 노드에서의 큐 할당과 스케줄링이 이루어지며, DDQN에서의 샘플 데이터 관측과 memory replay와 같은 학습 프로세스들이 수행된다. 패킷 전송이 완료되면 이와 같은 과정을 다시 반복하게 된다. 본 연구에서의 시뮬레이션에서는 플로우의 마감기한까지 남은 시간에 따라 처리순위를 결정하였지만, 일반적으로는 아래의 그림 2와 같이 다양한 기준에 따라 결정할 수 있다. 트래픽의 상황이 여유로울 때는 플로우의 클래스와 같은 기존의 우선순위 큐 할당 기준을 채택하다가, 혼잡이 예상되면 가장 우선해야 하는 기준에 따라 처리순위를 바꿀 수 있다.

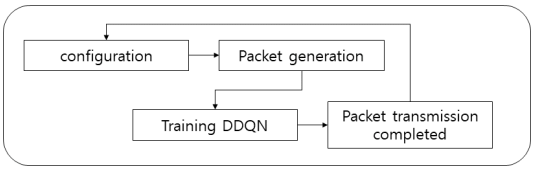

그림 1. 학습 과정의 도식화
Fig. 1. workflow of the training simulation

기존 연구[13]에서 증명된 바와 같이, 강화학습 기반 스케줄링 방식을 개별 노드에서 독립적으로 행동할 수 있도록 학습했기 때문에, 다양한 토폴로지에서도 스케줄링이 가능하다. 이렇듯 강화학습 방식의 스케줄링을 활용하면 다방면의 효과가 예상되지만, 강화

	Criteria
Flow level (note: F1~F4 is static, while F5 is dynamic)	F1: Total path cost F2: Path length (special case of F1 where every path cost is one) F3: Urgency (e.g. Requested E2E delay bound) F4: Other measure of importance (e.g. differentiated reward, like the Military service) F5: Flow service history (state). e.g. token bucket
Packet level	P1: Experienced delay so far P2: Delay budget left (Delay bound – P1) P3: Expected latency from now on • calculated from congestion status ahead • remaining hop (remaining path cost)

그림 2. 처리순위 할당 기준
Fig. 2. precedence queue assignment criteria at flow-level or packet-level

학습을 실제 네트워크 장비에 탑재하기 위해서는 에너지 효율 향상과 추론시간 단축과 같은 과제들이 남아있다. 따라서 강화학습 기반 스케줄러의 실현을 위해서는 딥러닝 연산의 비용을 고려하는 것 또한 중요하다. 본 연구는 간단한 입력데이터와 적은 인공신경망의 파라미터를 사용하였으며, 인공신경망 모델의 추론 빈도를 낮추는 방법을 사용해 해당 문제를 고려하였다.

본 연구에서는 시뮬레이션을 통해, 상황에 따라 처리순위가 높은 플로우에 적은 종단간 지연시간(end-to-end delay)을 보장할 수 있음을 입증하였다

Ⅱ. 관련 연구

2.1 Background

IEEE TSN TG와 IETF DetNet WG에서는 매우 낮은 패킷 손실률와 지연시간 및 지터 등 데이터 전달의 확정성을 확보하기 위한 기술의 표준화를 추진하고 있다[3]. 시간 민감성 통신의 수요 증가에 따라 스케줄링 기술의 개선이 필요한데, 이에 대한 잠재적인 해결책으로 딥러닝 기반의 강화학습 분야를 활용하는 연구 사례가 늘고 있다[4]. [5~8]은 딥러닝 기반의 강화학습을 통해 네트워크 스케줄링을 수행하였고, 라우팅[6] 및 자원 할당[9,10] 연구 사례도 확인할 수 있다. 이렇듯 네트워크 분야의 다양한 문제를 강화학습을 통해 해결하고자 하는 시도가 이루어지고 있다. 데이터 전달의 확정성을 확보하기 위해 본 연구 이전에 진행했던 연구[13]에서도 DDQN기반의 스케줄링이 타임슬롯 기반의 환경에서 기존 알고리즘의 성능을 능가하는 결과를 보였던 바 있다.

2.2 강화학습

강화학습은 기계학습의 지도, 비지도학습 방식과 달리 에이전트를 특정 환경에서 정보에 따른 행동 정책(policy)을 배우도록 하는 학습 방법이다. 기존에는 Q-table을 사용하는 Q-learning을 사용하다가, 딥러닝의 발전과 함께 인공신경망을 강화학습에 활용하게 되면서 강화학습의 발전이 이루어졌다. 강화학습에 딥러닝을 최초로 도입한 DQN(Deep Q-Network) 연구는 [11]에서 확인할 수 있다. 본 연구에서는 DQN에서 더 발전한 모델인 DDQN을 사용하여 에이전트를 학습하였다[12]. 강화학습의 학습 과정과 수식의 유도에 대한 설명은 그림 3과 [13]을 참조하라.

DDQN은 상태(state)에 대한 최적의 행동(action)을 추정하는 것이 목표이다. 딥러닝의 인공신경망을 통해

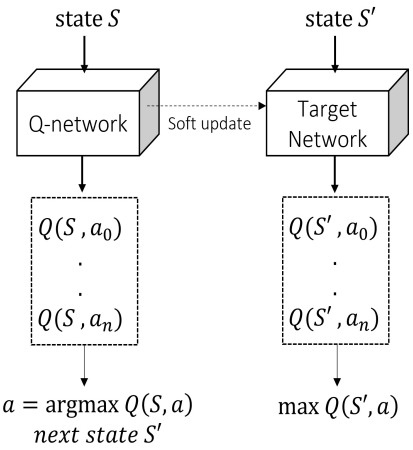

그림 3. DDQN 학습 과정[13]
Fig. 3. Training process of DDQN[13]

특정 상태에서 선택할 수 있는 모든 행동별 Q-value라는 가치 값을 잘 추정하도록 학습하는 것이고, 이 추정값에 대한 정답(target)데이터는 에이전트가 실제로 행동한 것에 대한 보상(reward) 값을 기반으로 계산한다. 계산된 정답데이터와 추정된 Q-value의 오차를 역전파함으로써 신경망을 업데이트하고 에이전트의 예측 성능을 향상해나간다. 이와 같은 학습을 위해 다양한 입력 데이터가 필요한데, 이는 여러 번의 시뮬레이션을 통해 얻어진다. 에이전트가 학습하는 환경은 데이터를 생성하는 역할을 하는 셈이다. 지도학습에서 준비된 데이터 셋을 입력하는 것과 같은 이치로, 데이터를 가공하고 전처리하는 과정이 필요한 것과 마찬가지로 에이전트가 환경에서 얻은 정보에서 행동 선택에 도움을 주는 형태로 변환해 상태(state)에 입력해주는 것이 필요하다.

Ⅲ. 시스템 모델링

3.1 네트워크 환경

DDQN 에이전트의 확장성과 일반화를 위해, 네트워크의 트래픽이 고정적이지 않고 유동적으로 생성되는 환경을 가정하였다. 네트워크의 시뮬레이터는 파이썬 기반의 이산 이벤트 시뮬레이션 환경을 생성할 수 있는 SimPy 프레임워크로 구현하였다. 시뮬레이터는 네트워크의 노드, 링크와 같은 요소들을 모듈화 하여 구성하였고, 패킷의 생성 및 전송과 큐 인입과 같은 프로세스들을 모듈 내부에 구현하여 시뮬레이션을 수행하였다. DDQN 에이전트가 학습하는 시뮬레이션 환경은 그림4와 같다. 해당 환경에서는 두 개의 처리

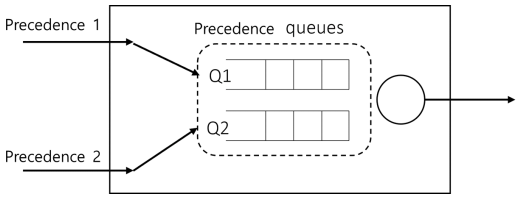

그림 4. DDQN 학습을 위한 개별 노드의 구조
Fig. 4. Structure of the single node for DDQN

순위 큐가 존재하고 더 높은 처리순위일수록 낮은 번호의 처리순위 큐에 할당된다.

본 연구에서 시뮬레이션을 진행한 네트워크 환경과 네트워크 파라미터들은 표 1에서 확인할 수 있다. 해당 파라미터들은 네트워크의 다양한 트래픽 패턴을 만들기 위해 임의의 값을 할당하였다. 여러 번의 시뮬레이션을 통해 다양한 외부 환경을 경험하게 하여, 특정 환경에만 적응하는 것이 아닌 다양한 환경에서 활용할 수 있는 스케줄링을 구현하기 위함이다. 네트워크에서는 총 8개의 플로우가 생성되며, 각 플로우들은 고정된 주기, 마감기한을 가지고 있다. 이때, 마감기한은 패킷이 생성 시점부터의 종단 지연시간에 대한 요구 조건을 의미한다. 주기와 마감기한 모두 임의로 생성되며, 각 플로우들은 시뮬레이션이 시작되는 초기 타임슬롯 시점에 전송이 시작되지 않고 주기의 배수인 일정 타임 스텝 이후 전송이 시작된다. 이 전송 시작 시점을 dt_n로 명시하였다.

표 1. configuration 단계에서 결정되는 네트워크 파라미터
Table 1. Parameter description

Parameter	Description	Quantity / Range
N	플로우의 총 개수	8
p_n	n번째 플로우의 생성 주기, 타임슬롯 단위	12~24
h_n	n번째 플로우의 남은 홉 수, 타임슬롯 단위	1~5
dt_n	n번째 플로우의 전송 시작 타임슬롯	0~30
b_n	n번째 플로우의 최대 버스트(burst)	$2\sim\frac{U}{N}p_n$
d_n	n번째 플로우의 마감기한	8~12
np_n	n번째 플로우의 패킷 갯수	60
dp	스케줄링 결정의 업데이트 주기, p_n의 최소 공배수	$lcm(p_n)$
U	에피소드동안 생성되는 트래픽의 이용률(utilization), $\sum_{i=0}^{n}\frac{b_i}{p_i}$	0.8~1.0

시뮬레이션 시작 전 구성 단계에서 랜덤 파라미터들이 결정된다. 플로우들의 남은 홉 수 또한 임의로 결정하였다. 본 연구에서는 1 타임슬롯을 1500byte로 고정된 사이즈의 패킷 1개가 지나갈 수 있는 크기로 정의하였다. 따라서, h_n로 명시된 남아있는 홉 수 또한 타임슬롯 단위를 가질 수 있다. 단위 시간당 전송률을 의미하는 이용률 U를 랜덤하게 결정함으로써 다양한 네트워크 환경을 시뮬레이션할 수 있으며, 주기마다 1개의 패킷을 전송하는 대신 b_n의 한도 내에서 임의의 수만큼 패킷을 전송하게 되므로, 다소 불규칙적인 패킷 도착 분포(packet arrival rate)를 보일 수 있다. 랜덤 파라미터들은 여러 번의 시뮬레이션에서 DDQN이 학습할 수 있는 환경을 구성하기 위해 네트워크가 너무 혼잡하거나 여유롭지 않은 수치를 경험적으로 결정하였다. 패킷의 최대 버스트(burst)를 결정하는 과정은 다음과 같이 유도할 수 있다:

각각의 플로우들은 패킷을 전송할 때 U 수준의 패킷 rate R을 지켜야 하므로 U와 N에 대해서 R은

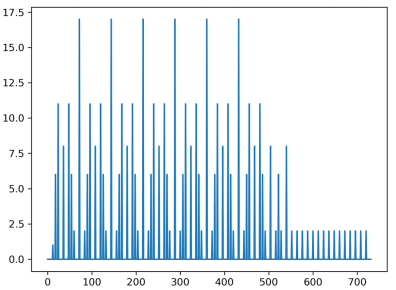

그림 5. 타임슬롯별 패킷 생성량
Fig. 5. Packet generation amount per timeslot

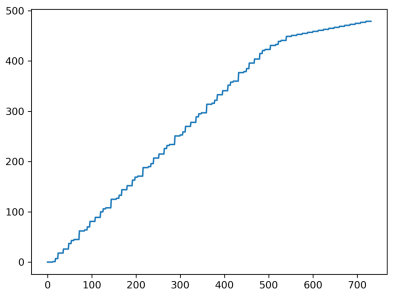

그림 6. 타임슬롯별 패킷 누적 생성량
Fig. 6. Accumulated packet generation amount

다음과 같다. $R=\frac{U}{N}$. 이 때, $R=\frac{b_n}{p_n}$ 으로 나타낼 수 있으므로, $\frac{U}{N}=\frac{b_n}{p_n}$ 이 성립한다. 따라서 $b_n=\frac{U}{N}p_n$ 이다.

연구 환경에서 b_n 은 2 이상 $\frac{U}{N}p_n$ 이하 임의의 수를할당받게 된다. 이러한 과정을 통해 생성된 트래픽의 패턴을 시각화하면 그림 5와 같다. 그림 5는 타임슬롯 마다 생성된 패킷 그래프를 나타내며, 그림 6은 생성된 누적 패킷 그래프를 나타낸다.

3.2 강화학습 환경

앞서 언급한 것과 같이, 본 연구에서 제안하는 DDQN 기반 스케줄링은 확장성을 위해 개별 노드에서 관측할 수 있는 상태만으로 학습해야 하고, 처리순위 큐 할당, 스케줄링 방식 두 개의 행동을 결정한다. 큐에서 관측할 수 있는 가장 중요한 정보로 패킷들의 생성부터 현재시점까지의 지연시간이 있으며, 앞으로 목적지까지 소요될 것으로 예상되는 지연시간이 있다. 이를 통해 개별 패킷들의 종단 간 지연시간을 예측할 수 있다. 따라서 연구에서는 추정 종단 간 지연시간(estimated end-to-end delay)를 ET 라고 명시하여, DDQN학습 및 테스트 시 상태의 관측 및 보상 지급, 성능 측정에 있어서 ET를 사용한다. 강화학습에서는 환경의 MDP 요소들인 상태, 행동, 보상의 적절한 설계가 매우 중요하다. 해당 요소들은 DDQN의 메모리 버퍼에 저장되는 데이터 샘플들을 구성하므로, 지도학습에서의 데이터 수집 및 전처리가 결과에 많은 영향을 주듯이 강화학습에서도 상태, 행동, 보상의 설계가 모델의 성능에 중요한 역할을 한다. 또한, 연구에서 스케줄링 결정이 수행되는 주기는 dp라고 명시한 바 있으므로 강화학습의 모든 프로세스가 발생하는 타임스텝은 dp이다. 연구에서 정의한 상태, 행동, 보상의 수식에 사용된 기호들은 표 2에서 확인할 수 있다.

표 2. 기호 설명
Table 2. Symbol description

Symbol	Quantity
ET_i^p	p번 째 처리순위 큐의 i번 째로 대기중인 패킷의 estimated End-to-end delay
ω_i^p	p번 째 처리순위 큐의 i번 째로 대기중인 패킷의 타임슬롯 budget, $d_i^p - ET_i^p$
L_p	p번 째 처리순위 큐에서 대기중인 패킷의 수
F	dp동안 플로우 처리순위를 결정하는 heuristic algorithms의 집합
H	dp동안 스케줄링 policy를 결정하는 heuristic algorithms의 집합
α	처리순위 1에 대한 reward에 할당하는 hyper parameter
β	처리순위 2에 대한 reward에 할당하는 hyper parameter
χ_p	p번 째 처리순위의 패킷들의 마감기한 이내 전송 확률

3.2.1 상태(state)

연구에서 상태를 정의할 때 단순하면서도 핵심적인 정보를 함축하고 있는 요소를 우선적으로 고려하였다. 연구의 목적은 패킷들의 마감기한을 엄수하는 것이기 때문에, 큐의 복잡성과 지연시간을 고려한 정보들을 상태로 설정하였다. 또한 해당 에피소드에서의 이용률 정보를 추가하여 네트워크 상황에 대한 정보를 추가하였다. 상태 s들의 집합인 S는 다음과 같이 정의하였다:

$$S=[L_1, L_2, \min(\omega_i^1), \min(\omega_i^2), U]$$

3.2.2 행동(action)

제안하는 모델에서의 행동은 2차원으로 구성되어 있음을 언급하였다. 본 연구에서는 처리순위 큐 할당을 결정하는 휴리스틱, 스케줄링 방식을 결정하는 휴리스틱 두 종류의 선택 가능 알고리즘이 있고, DDQN 에이전트가 관측한 상태에 따라 dp동안 해당 휴리스틱 알고리즘들을 유지하게 된다. 이를 통해 DDQN 에이전트는 dp라는 주기마다 큐의 상황을 관측, 적절한 휴리스틱들을 동적으로 결정하는 방법을 학습하게 된다. 처리순위 큐의 할당 기준은 위 그림 2에서 언급하였다. 연구의 학습 환경에서는 플로우의 처리순위 할당 시 플로우의 마감기한과 남은 홉 수의 차이를 플로우별 오름차순으로 정렬하여, 플로우 수준의 긴급도(urgency)를 처리순위 할당의 기준으로 설정하였다. 이에 따른 처리순위 큐 할당 휴리스틱은 긴급도에 따라 오름차순으로 정렬된 플로우들 중에서 상위 2개 플로우는 무조건 처리순위 1을 할당하고, 추가적으로 f 개의 플로우들을 더 처리순위 1에 할당할지 결정하는 알고리즘이다. 즉, 해당 휴리스틱 알고리즘은 총 $2+f$개가 된다. 이 때 $f \in F$ 임을 알 수 있다. 하지만 군용 네트워크망과 같이 확정적 네트워크를 사용하는 환경에서는 긴급도 외에도 송/수신자의 권한 및

처리순위를 할당하고자 하는 여러 평가 기준이 존재할 수 있다. 이 점을 고려하여, 처리순위의 할당 기준을 변경하고자 할 때 따로 DDQN 에이전트를 학습할 필요 없는 휴리스틱을 설계해 다양한 환경에 적용하기에 용이하도록 설계하였다. DDQN 에이전트가 선택할 수 있는 스케줄링 방식은 총 5개의 알고리즘들로 구성하였다.

1) strict priority (SP)
2) weighted round robin (WRR)
3) Heuristic1 (H1) : $\arg\min_p(\frac{\omega_0^p}{h_0^p})$
4) Heuristic2 (H2) : $\arg\min_p(d_0^p)$
5) Heuristic3 (H3) : $\arg\min_p(\omega_0^p)$

이중 SP는 기존 네트워크 알고리즘으로, 우선순위가 더 높은 큐의 패킷을 먼저 서비스하는 알고리즘이다. WRR은 우선순위 큐들을 차례대로 서비스하는 round robin 알고리즘에서, 할당된 가중치만큼 높은 우선순위 패킷을 조금 더 많이 서비스할 수 있는 알고리즘이다. 3) ~ 5)는 연구에서 설정한 휴리스틱으로, 각 처리순위 큐의 첫 번째 패킷의 특정 값을 비교해 더 작은 패킷을 서비스하는 알고리즘이다. DDQN 모델에서의 행동은 1차원으로 선택할 수 있게 설계되어 있기 때문에, 2차원의 action을 1차원으로 맵핑 시키는 절차가 필요하다. DDQN 에이전트가 선택할 수 있는 action a의 집합을 수식으로 나타내면 결과적으로 다음과 같다:

$$A = \{f \times h \mid f \in F, h \in H\}$$

3.2.3 보상(reward)

다음으로 보상은 dp 동안 패킷이 마감기한 안에 도착할 확률에 따라 지급한다. 또한 처리순위에 따른 가중치를 두어 보상에 차별성을 두도록 설계하였다. α는 처리순위 1 패킷들의 마감기한 이내 도착 확률에 할당되는 가중치이고, β는 처리순위 2의 패킷들의 마감기한 이내 도착 확률에 할당되는 가중치이다. 마감기한 이내 도착 확률은 dp 이내에 전송한 총 패킷 수에 대한 마감기한 만족 패킷수의 비율을 의미한다. 마감기한의 만족 여부는 실제 종단 지연시간이 아닌 ET 기반이다. 보상 r를 수식으로 정의하면 다음과 같다:

$$r = \alpha\chi_1 + \beta\chi_2$$

보상은 dp동안 각 처리순위 큐에서 평균적으로 마감기한 내에 패킷을 서비스할 확률에 대한 수식으로 정의하였다. 단순히 처리한 패킷의 수로 보상을 설계하게 되면, dp의 길이나, 개별 노드에 도착한 패킷의 수와 같이 에이전트가 제어할 수 없는 환경에 의해 보상 값이 달라지게 되기 때문이다.

에이전트가 학습 프로세스를 실행하는 타임스텝은 dp 단위임을 위에서 언급하였다. 정리하자면, 타임스텝 t에서 $< s_t, a_t, s_{t+1}, r_{t+1} >$ 은 DDQN 에이전트가 관측할 수 있는 데이터 샘플이 되고, 이는 메모리 버퍼에 저장되고 재생(replay)되어 학습이 이루어진다. 여기서 r_{t+1}은 t에서의 결과로 주어진 보상이기 때문에, $t+1$에서 지급되는 보상이다.

Ⅳ. 실험 결과

4.1 기존 알고리즘 간의 비교

본 연구에서 선택할 수 있는 스케줄링 알고리즘은 총 5개이다. DDQN 학습에 앞서, 에이전트가 선택할 수 있는 스케줄링 알고리즘끼리의 스코어 비교를 진행하였다. 연구에서 성능의 지표로 사용된 스코어는 에피소드 동안 얻은 r의 평균을 의미한다. 각 1000번씩의 시뮬레이션을 진행하였으며 시뮬레이션 결과는 아래 그림 7과 같다. 그림에서 가로축은 스코어, 세로축은 휴리스틱 알고리즘의 종류를 의미한다. WRR과 SP는 우선순위를 보장하기 위해 제안된 알고리즘이며 그 외 휴리스틱 알고리즘들은 단순히 패킷의 지연시간에 관련된 요소들만으로 스케줄링하는 알고리즘이

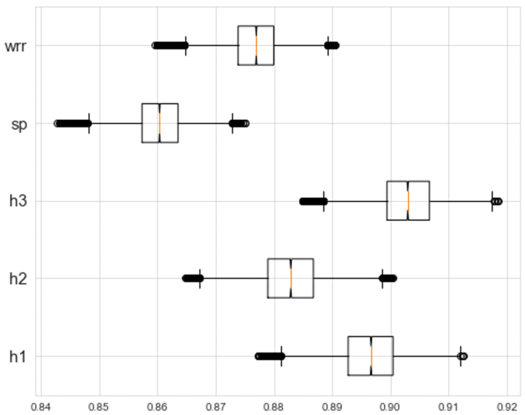

그림 7. 휴리스틱 알고리즘간의 스코어 비교
Fig. 7. Score comparison between heuristic algorithms

다. 각 알고리즘들을 시뮬레이션할 때 처리순위 휴리스틱 f=2로, 처리순위 큐 1에 플로우 4개, 처리순위 큐 2에 플로우 4개가 고정적으로 할당된다. 제안된 선택할 수 있는 휴리스틱 알고리즘 중에서는 H3이 가장 높은 스코어를 보였음을 알 수 있다. 여기서 α와 β는 각각 0.5이다. 즉, 처리순위의 차별성을 두지 않고 오로지 휴리스틱 알고리즘의 패킷 스케줄링 성능만을 비교하였다. 또한 1000회의 시뮬레이션 모두 랜덤 파라미터의 구성이 진행되기 때문에 모두 다른 패턴으로 트래픽이 생성되는 환경이지만, 시뮬레이션 내부의 랜덤 시드(seed)를 모두 같은 값으로 고정하였기 때문에 각각의 알고리즘마다 모두 같은 패턴의 순서대로 시뮬레이션을 수행하였다. 시뮬레이션의 U는 0.8~1.0으로 설정하였으며, 스코어의 최댓값은 1.0이다. 최고 성능을 보인 H3 알고리즘에서 패킷들을 마감기한 안에 전송할 확률은 평균 약 0.90~0.91이 되는 것을 확인할 수 있다.

4.2 DDQN 학습 결과

그림 8은 20000회의 DDQN 학습 에피소드 동안의 학습 곡선(learning curve)을 보여준다. 비교를 위해 최고 성능의 휴리스틱인 H3 알고리즘의 결과를 에피소드별로 DDQN과 함께 나타내었다. 해당 그래프는 각 에피소드에서 달성한 스코어의 이동평균을 나타내며, 이동평균의 윈도우 크기는 500이다. 학습 곡선에서 확인할 수 있듯이 약 10000 에피소드부터 14000 에피소드까지 DDQN이 최고 성능 휴리스틱보다 높은 성능을 보인 이후, 점차 스코어가 감소하는 결과를 보인다. 이는 DDQN 모델이 과적합 되었을 가능성이 있으므로, 성능 검증에서는 500 에피소드의 윈도우 동안 가장 높은 성능을 달성한 구간에서의 DDQN 모델의 가중치를 다른 알고리즘들의 성능 비교 평가에 활용하였다. 그림 8에서의 H3과 DDQN의 스코어 분포에 다소 변동이 존재하는 것은 에피소드마다 여러 랜덤 파라미터들이 결정되기 때문이다. 네트워크 환경의 이용률과 패킷 버스트의 설정으로 인해 변동이 있는 트래픽을 보이게 되고, 이에 따라 학습이 어려워지게 된다. 이는 연구에서 특정 네트워크 환경에 과적합 되지 않으면서 일반적인 네트워크 환경으로의 확장 및 적응을 위해 의도한 부분이다.

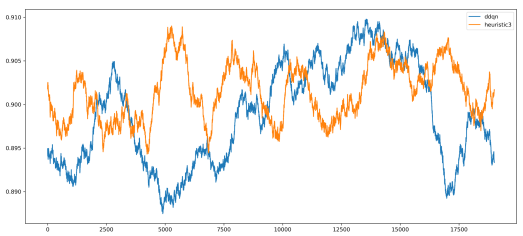

그림 8. 최적 휴리스틱을 사용한 경우 DDQN의 학습 곡선
Fig. 8. DDQN learning curve with best heuristic

그림 9는 학습된 DDQN과 그림 7에서 시뮬레이션한 스케줄링 알고리즘들의 스코어 비교 결과이다. dp마다 상태를 관측하고 스케줄링 알고리즘과 처리순위 할당 휴리스틱을 동적으로 선택하는 DDQN 알고리즘과 스케줄링 알고리즘 단독 동작의 비교 그래프이다. 그림 8에서와 마찬가지로 각각 1000회의 시뮬레이션을 진행하였다. 처리순위1에 0.7의 가중치, 처리순위 2에 0.3의 가중치를 주었을 때 DDQN이 가장 높은 평균 스코어를 보였음을 확인할 수 있다. DDQN은 outlier가 존재하지 않는 것 또한 확인할 수 있다. 그림 10은 이용률이 비교적 낮고, 보상의 계산 시 처리순위에 주어지는 가중치를 0.5, 0.5로 공평하게 설정한 시뮬레이션의 결과이다.

시뮬레이션 결과를 통해 적절한 파라미터를 설정하는 경우 DDQN은 처리순위의 보장에 있어서 H3을 포함한 다른 알고리즘들보다 더 좋은 성능을 보임을 알 수 있다. 무조건 높은 우선순위의 패킷을 전송하는 SP보다도 높은 스코어를 보였기 때문에, DDQN은 처리순위1은 물론 처리순위2의 패킷도 마감기한을 더 잘 보장한다는 것으로 해석할 수 있다. 그림 10과 같이 처리순위에 주어지는 가중치가 모두 같을 때는 DDQN의 성능이 최고 성능 휴리스틱인 H3과 비슷한 수준에 도달하는 결과를 보였다. 본 연구에서 DDQN 스케줄링의 최우선 역할은 처리순위가 높은 플로우의 마감기한을 만족시키는 것이었기 때문에, 네트워크의

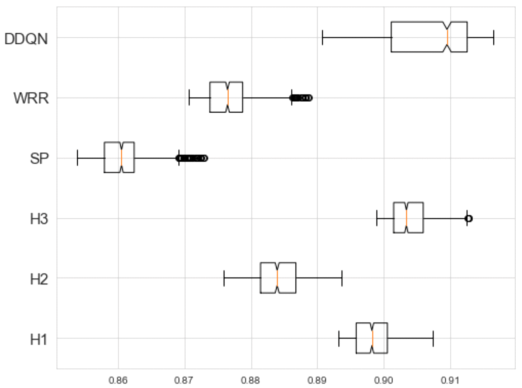

그림 9. U = 0.8~ 1.0, α = 0.7, β = 0.3일 때 스코어
Fig. 9. score comparison when U = 0.8~ 1.0, α = 0.7, β = 0.3

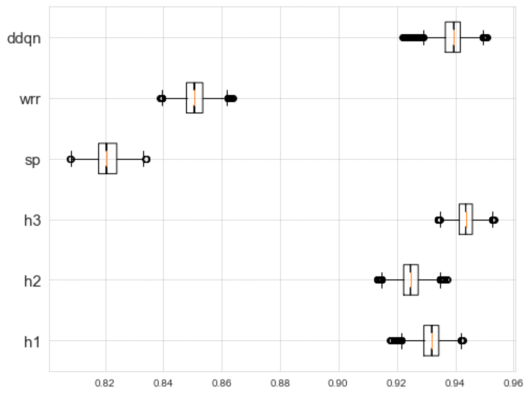

그림 10. U = 0.7~ 0.9, α = 0.5, β = 0.5일 때 스코어
Fig. 10. score comparison when U = 0.7~ 0.9, α = 0.7, β = 0.3

전체 트래픽의 마감기한을 만족하는 것도 중요하지만 이용률이 거의 100%에 육박하는 매우 혼잡한 상황에 있어서 높은 처리순위 트래픽의 마감기한을 잘 만족하는 것이 중요하다. 이를 위해 보상함수에 각 처리순위에 대한 가중치를 할당하였으며, 결과적으로 그림 9에서 DDQN 알고리즘이 타 알고리즘 및 휴리스틱에 비해 높은 비율로 처리순위가 높은 트래픽들을 deadline 안에 전송하였음을 확인하였다.

V. 결 론

점점 증가하는 어플리케이션의 실시간성을 보장하기 위해서 새로운 솔루션을 찾으려는 시도가 이어지고 있고, 여기에 딥러닝 기반의 강화학습이 가능성 있는 대책으로 부상하고 있다. 딥러닝은 음성, 언어, 이미지 처리와 같은 분야에서 성능을 이미 입증한 바 있다.

본 연구에서는 군용 네트워크에서 제안된 처리순위의 개념을 활용한 DDQN 스케줄링을 고안하였다. 다양한 휴리스틱 알고리즘과의 성능 비교 결과, DDQN 스케줄링은 평균적으로 최대 성능의 알고리즘과 비슷한 정도의 성능을 보였지만, 높은 처리순위의 플로우에 대해서는 마감기한 만족률을 더 높은 비율로 충족시켰다. 이는 강화학습 기반 스케줄링이 패킷의 종단 지연시간에 대한 요구사항을 잘 만족시켜줄 수 있을 뿐만 아니라, 처리순위가 높은 데이터에 대해 더 높은 수준의 성능을 제공할 수 있음을 시사한다.

본 연구에서는 네트워크에 딥러닝 기반의 연산을 적용하기 위해서 적절한 네트워크 시뮬레이터를 구현하였다. 확장 가능한 대규모 네트워크에 적용하기 위해, 에이전트에 다양한 네트워크 상황을 학습해 볼 수 있도록 에피소드마다 변하는 이용률, 주기, 마감기한 등의 임의의 파라미터를 설정하였다. 이는 시뮬레이션 환경에 편차로 작용하여 학습 곡선의 변동폭 요인이 되었지만, DDQN 에이전트가 특정 네트워크 상황에 대한 과적합을 방지하고 다양한 상황을 학습하는 데에 도움을 주었다. 에피소드마다 환경의 편차가 너무 큰 경우, 트래픽이 매우 여유롭거나 혹은 극도로 혼잡한 상황이 발생하기 때문에 DDQN 학습 시 손실함수(loss)가 폭주하고 제대로 학습하지 않는 결과를 초래하기 때문에 에이전트가 학습할 수 있는 적절한 랜덤 파라미터들의 설정이 중요한 역할을 하였다.

향후, 평균적인 스코어를 증가시키는 것을 목표로, 스코어의 분포를 줄여 어떤 상황에서든 강인하고 안정적인 강화학습 기반 스케줄링을 구현하는 방향으로 연구를 계획하고 있다. 본 연구에서 제시한 강화학습 메커니즘은 실제 네트워크 장비에 설치하기 위해서 다소 비용이 발생할 수 있다는 부분이 어려움으로 남아있다. 네트워크에 딥러닝을 도입하는 연구가 상용화되기 위해서는 다양한 벤치마크 데이터셋과 표준화된 시뮬레이션 프레임워크가 필요한 상황이며, 딥러닝의 예측(prediction)을 실시간으로 스케줄링에 활용하기 위한 모델 경량화 및 에너지 효율성 측면에서의 연구가 필요하다. 향후 연구에서는 딥러닝 기반 강화학습 스케줄링의 일반화에 초점을 둘 계획이다.

References

[1] Y. Choi, et al., "Ultra-high-precision network technology trend for ultra-immersive/high-precision service," *Electron. and Telecommun. Trends*, vol. 36, no. 4, pp. 34-47, Aug. 2021. (https://doi.org/10.22648/ETRI.2021.J.360404)

[2] Y. Xue, et al., "A framework for military precedence-based assured services in GIG IP networks," *IEEE Military Commun. Conf.*, Orlando, FL, USA, Oct. 2007. (https://doi.org/10.1109/MILCOM.2007.4454862)

[3] N. Finn, et al., "Deterministic networking architecture," *RFC* 8655, Oct. 2019. (https://www.hjp.at/doc/rfc/rfc8655.html)

[4] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," in *IEEE Commun. Surv. & Tuts.*, vol. 21, no. 3, pp. 2224-2287, third quarter 2019. (https://doi.org/10.1109/COMST.2019.2904897)

[5] S. Chinchali, et al., “Cellular network traffic scheduling with deep reinforcement learning,” *Thirty-Second AAAI Conf. Artificial Intell.*, vol. 32, no. 1, 2018. (https://doi.org/10.1609/aaai.v32i1.11339)

[6] S. Chilukuri and D. Pesch, “RECCE: Deep reinforcement learning for joint routing and scheduling in time-constrained wireless networks,” in *IEEE Access*, vol. 9, pp. 132053-132063, 2021. (https://doi.org/10.1109/ACCESS.2021.3114967)

[7] T. Zheng, et al., “Deep reinforcement learning-based workload scheduling for edge computing,” *J. Cloud Computing*, vol. 11, no. 3, pp. 1-13, 2022. (https://doi.org/10.1186/s13677-021-00276-0)

[8] D. Ghosal, S. Shukla, A. Sim, A. V. Thakur, and K. Wu, “A reinforcement learning based network scheduler for deadline-driven data transfers,” *2019 IEEE GLOBECOM*, pp. 1-6, Waikoloa, HI, USA, Dec. 2019. (https://doi.org/10.1109/GLOBECOM38437.2019.9013255)

[9] X. Xiong, K. Zheng, L. Lei, and L. Hou, “Resource allocation based on deep reinforcement learning in iot edge computing,” in *IEEE J. Sel. Areas in Commun.*, vol. 38, no. 6, pp. 1133-1146, Jun. 2020. (https://doi.org/10.1109/JSAC.2020.2986615)

[10] S. Mollahasani, et al., “Actor-critic learning based qos-aware scheduler for reconfigurable wireless networks,” *IEEE Trans. Network Sci. and Eng.*, vol. 9, no. 1, pp. 45-54, 2021. (https://doi.org/10.1109/TNSE.2021.3070476)

[11] V. Mnih, et al., “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv: 1312.5602,* 2013. (https://doi.org/10.48550/arXiv.1312.5602)

[12] H. van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-Learning,” in *Proc. AAAI Conf. Artificial Intell.*, vol. 30, no. 1, pp. 2094-2100, 2016. (https://doi.org/10.1609/aaai.v30i1.10295)

[13] J. Ryu, J. Kwon, and J. Joung, “Timeslot scheduling with reinforcement learning using a double deep q-network,” *J. KICS*, vol. 47, no. 7, pp. 944- 952, 2022. (https://doi.org/10.7840/kics.2022.47.7.944)

류 지 혜 (Jihye Ryu)

2020년 8월 : 상명대학교 휴먼지능정보공학과 학사
2020년 9월~2023년 2월 : 상명대학교 지능정보공학과 석사
<관심분야> 네트워크, 강화학습, 딥러닝

박 규 동 (Gyudong Park)

1994년 2월 : 홍익대학교 컴퓨터공학과 졸업
1996년 2월 : 홍익대학교 컴퓨터공학과 석사
2014년 2월 : 홍익대학교 컴퓨터공학과 박사
1996년 1월~1998년 12월 : 국방정보체계연구소 연구원
1999년 1월~현재 : 국방과학연구소 연구원
<관심분야> 상호운용성, 네트워크, 인공지능, 보안
[ORCID:0000-0001-7484-5426]

권 주 혁 (Juhyeok Kwon)

2020년 8월 : 상명대학교 휴먼지능정보공학과 학사
2020년 9월~2022년 8월 : 상명대학교 지능정보공학과 석사
2022년 9월~현재 : 상명대학교 지능정보공학과 박사
<관심분야> 유무선통신, 네트워크, 임베디드 시스템

정 진 우 (Jinoo Joung)

1992년 2월 : KAIST 전자공학과 학사
1994년 8월 : NYU 전기전자공학과 Master
1997년 8월 : NYU 전기전자공학과 Ph.D.
1997년 10월~2005년 2월 : 삼성전자 종합기술원
2005년 3월~현재 : 상명대학교 휴먼지능정보공학과 교수
<관심분야> 유무선통신, 네트워크, 임베디드 시스템
[ORCID:0000-0003-3053-9691]