공 학 석 사 학 위 논 문

소형 셀 통신지연을 최소화를 위한 강화학습 기반 사용자 접속과 캐시 교체 전략

2022년 2월

부 경 대 학 교 대 학 원

스마트로봇융합응용공학과

전 상 은

공 학 석 사 학 위 논 문

소형 셀 통신지연을 최소화를 위한 강화학습 기반 사용자 접속과 캐시 교체 전략

지도교수 홍 준 표

이 논문을 석사학위논문으로 제출함.

2022 년 2 월

부 경 대 학 교 대 학 원

스마트로봇융합응용공학과

전 상 은

전상은의 공학석사 학위논문을 인준함.

2022 년 2 월 25 일

위 원 장 공학박사 이 훈 (인)

위 원 공학박사 홍 준 표 (인)

위 원 공학박사 윤 상 석 (인)

목 차

그림 목차

표 목차

소형 셀 통신지연을 최소화를 위한 강화학습 기반
사용자 접속과 캐시 교체전략

전 상 은

부 경 대 학 교 대 학 원 스마트로봇융합응용공학과

요 약

이 논문은 캐시 사용이 가능한 small cell network (SCN)에서 콘텐츠 전송 지연을 최소화하기 위한 사용자 접속과 캐시 교체전략의 최적화 연구이다. 사용자가 콘텐츠를 요청할 때마다 central unit (CU)는 콘텐츠를 전달할 small cell base stations (SBSs) 중 하나를 선택하고 사용자의 위치, SBS의 캐시 상태 및 환경을 고려하여 선택한 SBS의 캐시를 업데이트 한다. 이 논문은 정책 최적화에서 캐시 적중률과 통신 신뢰성 간의 균형을 고려하여 Markov decision process (MDP)에 의해 위 순차 결정 문제를 공식화한다. 효과적인 정책을 도출하기 위해 deep Q-network (DQN) 알고리즘을 채택하고 문제 특성을 활용하여 DQN 알고리즘에 대한 새로운 신경망 구조 및 입력 벡터를 설계한다. 다양한 환경에서의 시뮬레이션 결과 제안된 기법이 주어진 환경에 적합한 효과적인 전략을 학습할 수 있음을 보여줌으로써 평균 지연 시간 측면에서 기존의 전략 뿐만 아니라 기본 DQN 알고리즘보다 우수한 성능을 보였다. 또한, 콘텐츠의 요청확률이 시간에 따라 변화하는 환경에서의 시뮬레이션 결과에서 제안한 기법의 학습 과정이 콘텐츠에 적절히 학습됨을 보여준다.

Ⅰ. 서 론

소형 셀 네트워크는 주로 모바일 장치와 높은 데이터 전송률 서비스에 대한 수요 증가로 인해 모바일 트래픽의 급속한 성장에 대처하기 위한 효과적인 네트워크 구조로 간주되고 있다 [1]. Small cell base stations (SBSs)의 밀집 배치와 그에 대한 데이터 오프로딩 (offloading)은 셀 크기를 줄여 공간 스펙트럼 재사용을 증가시킴으로써 네트워크 영역 내 처리량을 크게 향상시킨다 [2], [3]. 그러나 이러한 이점이 충분히 활용하려면 SBS들의 고속 백홀 (backhaul) 연결이 보장 되야 하기 때문에, 백홀 네트워크 구축은 네트워크 조밀화의 주요 문제로 간주된다. 구축의 용이성, 유연성 및 비용 효율성을 위해 SCN에 대한 대부분의 연구는 광섬유 백홀 대신 링크 용량과 안정성을 희생하여 밀리미터 파(mmWave) 기반 무선 백홀을 채택했다 [4]. 무선 백홀 링크의 병목 현상을 방지하고, 네트워크 처리량을 개선하기 위해 다양한 밀집 네트워크 시나리오에서 통신 자원 할당 및 전송 기술이 광범위하게 연구되었다 [4]−[8].

연결 중심 통신에서 비디오 스트리밍 서비스와 같은 콘텐츠 중심 통

신으로 패러다임이 전환됨에 따라 에지 (edge) 노드의 캐싱 (caching) 기능은 제한된 백홀 용량으로 인한 문제를 완화하는 강력한 도구로 큰 주목을 받았다 [9], [10]. 에지 노드에 캐시된 콘텐츠를 이용하여 트래픽을 상대적으로 저렴한 메모리 자원으로 오프로드 (offload)함으로써 백홀 네트워크의 트래픽 부담을 줄일 뿐만 아니라 원격 서버의 통신 지연 시간을 줄일 수 있다. 이러한 이득은 일반적으로 캐시 적중률 (cache hit-rate) 에 비례하기 때문에, 소수의 인기 있는 콘텐츠가 모바일 트래픽의 대부분을 차지하고 캐싱에 필요한 메모리 용량 단위당 가격이 지속적으로 하락하는 현 상황에서 에지 캐싱의 장점이 더욱 두드러지고 있다. [11]－[13]. SBS가 캐시 기능을 가진 단일 셀 네트워크에서 SBS는 캐시 적중률과 네트워크 성능을 극대화하기 위해 요청 확률이 가장 높은 일부 콘텐츠를 캐시해야 한다. 콘텐츠 요청확률에 대한 사전 정보가 없을 경우 콘텐츠 요청확률 학습 및 인기 콘텐츠 캐싱 프로세스는 multi-armed bandit (MAB) 문제에 대한 알고리즘을 통해 해결될 수 있다 [14]. 다중 소형 셀 네트워크에서, SBS의 콘텐츠 배치는 캐시 적중률을 개선하기 위해 콘텐츠 중심 사용자 접속을 추가로 고려해야 한다 [15], [16]. 다시 말해, 콘텐츠 중심 사용자 접속 (user association) 은 근처의 여러 SBS에서 캐시된 콘텐츠를 활용하여 사용

자 요청을 처리할 수 있기 때문에, 모든 SBS에서 요청 확률이 가장 높은 콘텐츠들을 캐시 하는 것이 다중 소형 셀 네트워크에서 캐시 적중률을 최대화하는 최적의 콘텐츠 배치 전략이 아닐 수 있다. 이에 따라 사용자와 SBS의 배치가 주어졌을 경우 중앙 집중적인 네트워크 성능 향상을 위해 SBS 콘텐츠 배치가 연구되었다 [17]–[19]. 초기에 콘텐츠 요청확률을 알 수 없는 경우 SBS 간의 분산 콘텐츠 배치 및 공유를 위해 MAB 문제에 대한 알고리즘을 활용한 연구가 [20]에서 연구되었다. 콘텐츠 배치 외에도 캐시 적중률과 통신 신뢰성 사이에서 균형을 맞추기 위해 사용자−SBS 링크 거리와 SBS의 캐시 콘텐츠를 함께 고려하여 사용자 접속도 신중하게 결정해야 한다 [21], [22]. [23], [24]에서는 콘텐츠 배치와 사용자 접속의 최적화를 iterative 알고리즘을 통해 처리하는 기법이 연구 되었다.

일반적으로 콘텐츠 배치 전략은 업데이트 주기 마다 사용자 배치, 콘텐츠 요청확률 등 환경 변화에 따라 캐시된 콘텐츠를 일괄적으로 업데이트 한다. 이러한 일괄적인 업데이트는 동적 환경 변화에 적응하기 어려울 수 있기 때문에, 높은 사용자 이동성과 non-stationary 콘텐츠 요청확률 분포가 있는 시나리오에선 캐시를 활용한 성능이득이 제한적일 수 있다. 동적 환경을 위한 효과적인 캐시 전략으로, 사용자가 콘텐

츠를 요청할 때마다 캐시를 업데이트하는 캐시 교체 기법이 효과적일 수 있다. Least recently used (LRU) 및 least frequently used (LFU)과 같은 contents delivery network (CDN)에서의 기존 캐시 교체 전략을 SCN의 실시간 캐시 업데이트에 작동 할 수도 있지만 캐시 교체에서 무선 네트워크 구조를 추가적으로 고려할 경우 네트워크 성능을 더욱 향상시킬 수 있다. 그러나 캐시 교체 문제는 콘텐츠 배치와 달리 순차적 의사결정 문제 (sequential decision problem)로 볼 수 있어 기존의 최적화 이론을 적용해결하기 어려운 문제점이 있다. 순차적 구조를 반영하기 위해 SCN의 캐시 교체 문제는 Markov decision process (MDP)로 공식화 하여 강화 학습 기반 접근방식으로 해결하는 기법이 최근 활발히 연구되고 있다 [25]–[28]. 이러한 연구는 강화 학습에서 도출된 캐시 교체 전략이 고정 노드 구축 및 연결에 대한 분산 캐시 상태를 추가로 고려함으로써 LRU 및 LFU 기반 전략보다 더 낮은 백홀 사용을 달성할 수 있다는 것을 보여주었다. 콘텐츠 배치의 경우와 마찬가지로, 사용자 접속을 최적화 하는 것은 사용자에게 서비스를 제공하기 위해 사용 가능한 캐시를 확장함으로써 캐시 교체의 성능 향상을 이끌어낼 수 있다. 그러나 캐시 교체에 대한 기존 연구 중 무선 페이딩 채널과 저장된 콘텐츠 정보를 함께 고려해 사용자 접속을 결정하는 연구는 아직

이루어진 바가 없다. SCN에서 기대 콘텐츠 전송 지연 시간을 최소화하기 위한 캐시 교체 및 사용자 접속 문제를 연구한다. 여기서 콘텐츠의 캐시와 전달은 청크 (chunk) 단위로 수행된다. 고려하는 환경 시나리오에서 딥 페이딩으로 채널 상태가 좋지 않으면 무선 백홀 및 액세스 링크에서 재전송과 함께 전송 지연이 발생할 수 있다. 또한 지연 시간은 재전송 수 뿐만 아니라 관련 SBS 캐시의 가용성에 따라 달라지기 때문에 사용자 접속 및 캐시 교체의 최적화에서 통신 신뢰성과 캐시 적중률 사이에 trade-off 관계가 존재한다. 이러한 trade-off 관계와 캐시 교체의 특성을 고려하여, 이 문제는 MDP에 의해 공식화하고 다양한 상황에 적응할 수 있는 전략을 도출하기 위해 DQN 알고리즘을 적용하는 방법을 제안한다. 또한 학습을 용이하게 하기 위해 콘텐츠 요청에 대한 연속적인 청크 전달 및 SBS 캐시 간의 상호 관계와 같은 문제 특성을 기반으로 deep neural network (DNN)와 해당 입력 피처 (feature) 의 새로운 구조를 설계한다. 시뮬레이션 결과를 통해 DQN 알고리즘에서 도출된 전략은 기존 위치 기반 및 콘텐츠 기반 사용자 접속보다 평균 지연 시간을 단축할 수 있음을 보였다. 이는 제안된 DQN 알고리즘이 효과적인 캐시 교체 및 사용자 접속 전략을 학습하여 통신 신뢰성과 캐시 적중률의 균형을 유지함으로써 기대 지연 시간

을 최소화할 수 있음을 나타낸다. 뿐만 아니라 제안된 방식은 DNN 구조와 입력 피처 설계를 통해 평균 지연 시간 뿐만 아니라 훈련 프로세스의 수렴 속도 측면에서 기존 DQN 알고리즘을 능가함을 보았다. 이는 문제의 특성에 기반한 DNN 구조와 입력 기능 설계가 DNN 매개 변수 업데이트에서 불필요한 전파를 제한하여 적절한 훈련을 용이하게 하기 때문이다. 이러한 이점을 바탕으로 제안된 계획은 콘텐츠 요청확률이 시간에 따라 변화하는 non-stationary 시나리오에서 환경 변화에 빠르게 적응할 수 있다.

본 연구의 주요 기여도는 다음과 같이 정리할 수 있다.

1. 제안 기법은 사용자 접속과 캐시 교체의 최적화 문제를 MDP 형태로 공식화하고 통신 신뢰성과 캐시 적중률 사이의 trade-off 관계를 정의한 첫 연구이다.
2. 문제의 특성을 기반으로 새로운 DNN 구조와 입력 피처를 설계하여 DQN 알고리즘의 수렴 속도를 향상시킬 뿐만 아니라 기존 방식보다 낮은 평균 지연 시간을 달성하는 정책을 학습할 수 있게 한다. 또한 콘텐츠 요청 확률 분포가 바뀌면 전략을 빠르게 업데이트할 수 있다.
3. 시뮬레이션 결과는 제안된 기법이 모든 시뮬레이션 환경에서 기

존 사용자 접속 및 캐시 교체 기법보다 향상된 지연 성능을 보인다. 또한, 제안된 기법은 캐시 용량이 증가함에 따라 기존 기법과의 성능 격차가 커진다는 점에서 사용자 접속 및 캐시 교체의 최적화가 캐시 용량이 큰 향후 SCN에 중요 할 것임을 보였다.

4. 훈련된 전략의 동작은 다양한 상황에서 기대 지연 시간 최소화를 위해 효과적인 사용자 접속 및 캐시 교체 방법에 대한 통찰력 있는 정보를 제공한다.

논문의 나머지 부분은 다음과 같이 구성되어 있다. II 장 에서는 시스템 모델을 소개한다. III 장 에서는 사용자 접속 및 캐시 교체 문제가 MDP의 형태로 공식하며, MDP 해결을 위한 기본 DQN 알고리즘이 제시된다. IV 장은 DQN 알고리즘의 훈련 과정을 용이하게 하기 위한 DNN의 새로운 구조와 입력 피처 매트릭스를 설계하며, 제안된 기법의 성능은 V 장의 집중 시뮬레이션을 통해 평가된다. 마지막으로 VI 장으로 논문을 마무리한다.

Ⅱ. 시스템 모델

그림. 1와 같이 단일 central unit(CU)와 캐시가 가능한 K 개의 SBS가 있는 SCN을 고려한다. CU는 커버리지 (coverage) 내에 위치한 사용자 콘텐츠 요청을 효율적으로 처리하기 위해 전체 네트워크를 관리하고, SBS는 사용자에게 요청 콘텐츠를 전달한다. 각 지점의 위치를 나타내기 위해 2차원 좌표 $\mathcal{C} = \{(x, y): |x| \leq x_{max}, |y| \leq y_{max}\}$를 가정하며, 적용 범위는 좌표 집합으로 정의된다. CU는 적용범위의 중심에 위치한다고 가정하며, CU의 좌표는 원점 (0, 0)으로 표시된다. 사용자는 $\mathcal{C}$에 속하는 임의의 좌표에 위치한다. 사용자와 가장 가까운 SBS 사이 최대 링크 거리를 최소화하기 위해 K SBS는 규칙적으로 격자 중앙에 배치된다. 그림. 1은 $K = 4$를 사용한 네트워크 배치의 예를 보여준다. 사용자 콘텐츠 요청은 Zipf's law를 따르는 것으로 가정한다. 즉, 사용자가 요청확률이 높은 r번째 콘텐츠를 요청할 확률은 $q_r = \frac{1/r^s}{\sum_{l=1}^{L} 1/l^s}$로 나타낼 수 있으며, 여기서 $s > 0$은 Zipf의 지수를 나타내고 L은 사용자가 요청할 수 있는 콘텐츠의 수를 나타낸다. 각 콘텐츠는 동일한 크기의 N_{p} 청크로 분할되고 청크 단위로 전송 및 캐시한다.

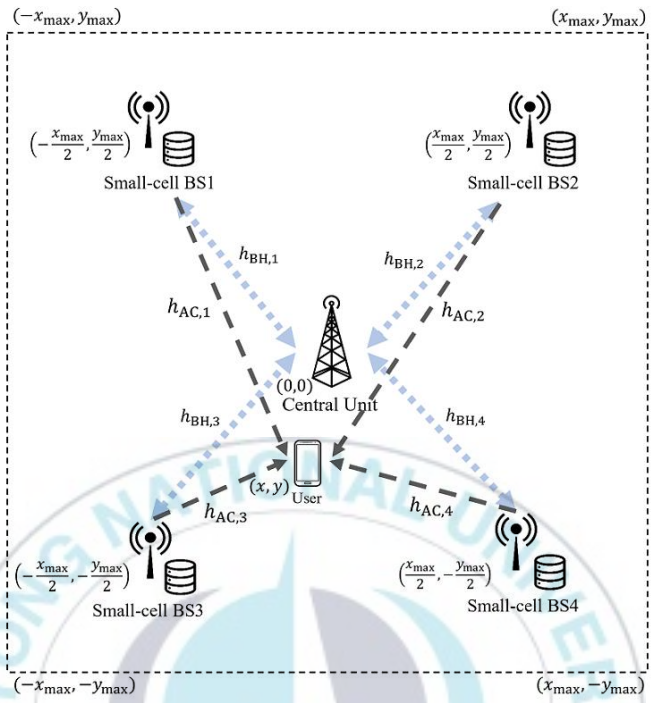

그림. 1 시스템 모델

CU는 모든 L 콘텐츠에 접근할 수 있다고 가정하지만 SBS k의 캐시용량 $\mathcal{M}_k$은 메모리 용량을 초과할 수 없다 $|\mathcal{M}_k| \leq M$. 따라서, SBS는 관련 사용자가 요청한 콘텐츠 청크가 메모리에 캐시 되어 있지 않을 경우 무선 백홀을 통해 CU로부터 청크를 수신해야 한다. CU는 SBS의 배포, 각 SBS의 캐시된 청크 및 요청하는 사용자 위치를 알고 있다고 가정한다. 이러한 정보를 바탕으로 CU는 요청된 콘텐츠의 청크를 전달하기 위해 SBS를 선택하고, 요청된 청크가 접속 SBS에서 캐시 되어 있지 않은 경우 새 청크로 교체할 캐시 청크를 결정 해야 한다. 다시 말

해, 요청된 각 콘텐츠의 청크 전달과정에서 CU는 사용자 접속 및 캐시 교체를 함께 제어하여 연속 콘텐츠 요청을 처리하는 지연 시간을 줄일 수 있다.

장기적인 channel state information (CSI) 만 CU에서 사용할 수 있고 SBS k는 사용자에게 i 번째 콘텐츠 요청의 청크 n인 청크 $\mu_{i,n}$를 전달하기 위해 선택된다고 가정한다. 또한 접속 SBS가 청크 $\mu_{i,n} \notin \mathcal{M}_k$ 을 캐시 하고 있지 않으면 무선 백홀 링크를 통해 CU에서 청크를 가져와 캐시 된 청크 중 하나를 새 청크로 교체해야 한다. 무선 백홀 채널을 고려하면, 전송 실패 확률은 다음과 같이 나타낼 수 있다.

$$p_{\mathrm{BH},k} = \Pr\left[\log_2\left(1+\left|h_{\mathrm{BH},k}\right|^2\rho_{\mathrm{BH}}\right) \le R\right], \quad (1)$$
$$= 1 - e^{-\left(2^R-1\right)/\rho_{\mathrm{BH}} d_{\mathrm{BH},k}^{-\alpha_{\mathrm{BH}}}},$$

이때, $h_{\mathrm{BH},k} \sim \mathcal{CN}\left(0, d_{\mathrm{BH},k}^{-\alpha_{\mathrm{BH}}}\right)$는 CU에서 SBS k 까지의 링크 거리 $d_{\mathrm{BH},k}$ 및 경로 손실 지수 α_{BH} 에 따른 페이딩 채널이득을 나타낸다. 백홀 전송률과 청크 전송의 신호 대 잡음비 signal-to-noise ratio (SNR) 도 각각 R 와 ρ_{BH}로 표시된다. 신뢰할 수 있는 전송을 위해 CU는 SBS로 성공적으로 전송될 때까지 청크를 다시 전송한다. 따라서, 성공적인 전달을 위한 전송 수 $n_{\mathrm{BH},k}$ 는 확률 $1-p_{\mathrm{BH},k}$ 을 따르는 기하분포의 무작위 변수로 간주될 수 있다. SBS k의 청크 수신이 성공한 후 CU의 결정에

따라 캐시된 청크 중 하나를 새로 수신된 청크와 교체하여 $\mathcal{M}_k$가 업데이트된다. 이어서, SBS k는 액세스 링크를 통해 청크 $\mu_{i,n}$을 요청 사용자에게 전달할 수 있다. 무선 백홀 링크와 유사하게, 액세스 링크의 청크 전송은 확률과 함께 실패할 수 있다.

$$p_{\mathrm{BH},k} = p_{\mathrm{AC},k} = 1 - e^{-(2^R-1)/p_{\mathrm{AC}} d_{\mathrm{AC},k}^{-\alpha_{\mathrm{AC}}}}, \quad (2)$$

이때, $d_{\mathrm{AC},k}$는 SBS k에서 사용자까지의 링크 거리를 나타내며 p_{AC}및 α_{AC}는 액세스 링크에서의 SNR 및 경로 손실 지수를 나타낸다. 성공적인 청크 전달을 위한 전송 수 $n_{\mathrm{AC},k}$는 성공 확률 $1 - p_{\mathrm{AC},k}$인 기하분포를 따른다. 액세스 링크의 전송실패 확률 $p_{\mathrm{AC},k}$는 콘텐츠를 요청하는 사용자 위치에 따라 달라지는 반면, 백홀 링크의 전송실패 확률 $p_{\mathrm{BH},k}$은 SBS의 고정 배치로 인해 일정하다는 점에 유의해야한다.

반면에 선택한 SBS k가 요청된 청크 $\mu_{i,n} \in \mathcal{M}_k$를 이미 캐시한 경우 SBS는 백홀 링크를 사용하지 않고 액세스 링크를 통해 사용자에게 청크 $\mu_{i,n}$을 전송할 수 있다. 따라서 청크 교체로 인한 $\mathcal{M}_k$ 변화가 없다.

청크 $\mu_{i,n}$을 사용자에게 전달하기 위한 통신 지연 시간이 백홀 및 액세스 링크의 각 전송에 대해 발생할 수 있다는 것을 고려할 때, 다음과 같이 나타낼 수 있다.

$$\tau_{i,n} = \begin{cases} n_{\mathrm{BH},k}\tau_{\mathrm{BH}} + n_{\mathrm{AC},k}\tau_{\mathrm{AC}}, & \mu_{\{i,n\}} \notin \mathcal{M}_k \\ n_{\mathrm{AC},k}\tau_{\mathrm{AC}}, & \mu_{\{i,n\}} \in \mathcal{M}_k \end{cases} \quad (3)$$

여기서 τ_{BH}와 τ_{AC}는 각각 백홀 링크와 액세스 링크의 전송 장애로 인한 지연 시간을 나타낸다. 요청된 청크가 접속 SBS에 이미 캐시 되어 있는 경우 백홀 링크 지연 시간이 발생하는 것을 방지할 수 있다. 따라서 T 번의 연속된 콘텐츠 요청을 처리하기 위한 누적 지연시간은 다음과 같이 나타낼 수 있다.

$$\tau_{\mathrm{sum}} = \sum_{i=1}^{T}\sum_{n=1}^{N_p}\tau_{i,n}, \quad (4)$$

(1), (2), (3)에서 볼 수 있듯이, 전송 실패와 캐시의 가용성이 지연 시간 τ_{sum}을 결정하는 두 가지 주요 요인으로 네트워크 지연 시간 감소에서 두 요소 사이 trade-off 가 있다. 예를 들어, CU가 전송 장애를 최소화하기 위해 거리 기반 사용자 접속을 채택할 경우, 사용자 요청을 처리하기 위해 사용 가능한 캐시는 사용자에게 가장 가까운 SBS의 청크로 제한된다. 반대로 CU가 캐시 적중률을 극대화하기 위해 협력적인 캐싱과 콘텐츠 기반 사용자 접속을 결정하면 액세스 링크의 긴 통신 거리로 인해 전송 장애가 자주 발생할 수 있다. 따라서 누적 지연 시간을 줄이려면 사용자 접속 및 캐시 교체 간 최적화가 중요하다.

Ⅲ. 사용자 접속 및 캐시 교체 학습

누적 지연 시간 합계를 최소화하는 사용자 접속 및 캐시 교체 문제를 기존 오프라인 최적화 접근 방식으로는 해결하는 것은 다음과 같은 이유로 어렵다. i) 향후 정보 없이 사용자 위치, 요청된 내용 및 SBS의 캐시된 청크 정보만으로 사용자 접속 및 캐시 교체에 대한 일련의 결정을 내려야 한다. ii) 교체결정에 따라 SBS의 캐시 청크가 결정되며, 이는 후속 의사결정에 영향을 미친다. 결과적으로, 각 교체결정에서 현재 상황의 청크 전송 뿐만 아니라 향후 청크 전송에 미치는 영향도 함께 고려해야 한다.

3.1. MDP와의 문제 일반화

앞서 언급한 문제의 특징을 반영하기 위해, MDP로 문제를 공식화 할 수 있다. 표현의 단순성을 위해 $t = N_p(i-1) + n$을 i 번째 요청된 콘텐츠의 n 번째 청크를 전달 하기위한 의사결정 단계로 정의한다. 따라서 기호에서 첨자 i와 n은 t로 대체될 수 있다 $\mu_{i,n} \rightarrow \mu_t$. 이 문제를 설명하는 데 사용된 MDP은 다음과 같은 네 가지 요소로 구성되어 있다.

- State : 시간 단계 t의 상태는 tuple로서 정의한다.

$$s_t = (\mu_t, (x_{\text{user},t}, y_{\text{user},t}), \mathcal{M}_{all,t}), \quad (5)$$

여기서 $(x_{\text{user},t}, y_{\text{user},t})$는 t단계에서 요청하는 콘텐츠 사용자 좌표를 나타내고 $\mathcal{M}_{all,t} = (\mathcal{M}_{1,t}\, \mathcal{M}_{2,t}, \dots, \mathcal{M}_{K,t})$ 는 t단계에서 캐시상태의 tuple을 나타낸다. $(x_{\text{user},t}, y_{\text{user},t})$ 는 연속 단계 $N_p(i-1)+1 \le t \le N_p i$, 고정된 N_p 에 대한 i번째 청크를 사용자에게 전송한다. $\mathcal{S}$를 가능한 모든 상태의 집합으로 정의한다.

- Action : 시간 단계 t의 액션은 tuple로서 정의한다.

$$a_t = (k, \check{\mu}_k), \quad (6)$$

여기서 k는 청크 전달을 요청하는 콘텐츠 사용자와 접속된 SBS의 인덱스를 나타내며, $\check{\mu}_k \in \mathcal{M}_{k,t}$ 는 새로운 μ_t로 교체되는 청크를 나타낸다. $\mathcal{A}$를 가능한 모든 작업의 집합으로 정의한다.

- Transition probability : $t \bmod N_p \neq 0$이면 요청된 콘텐츠의 전송이 완료되지 않아 다음 청크 μ_{t+1}는 동일한 사용자에게 전송한다. 따라서 $t \bmod N_p = 0$이면, $a_t = (k, \check{\mu}_k)$ 에서의 작용은 다음과 같이 구성된 s_t에서 s_{t+1}로 상태로 전환한다.

$$\mu_{t+1} = \mu_{i,t+1},$$
$$\left(x_{\text{user},t+1}, y_{\text{user},t+1}\right) = \left(x_{\text{user},t}, y_{\text{user},t}\right),$$
$$\mathcal{M}_{k',t+1} = \begin{cases} \mathcal{M}_{k',t}, & \text{if } k' \neq k \\ \mathcal{M}_{k',t}\{\check{\mu}_k\} \cup \{\mu_t\}, & \text{if } k' = k \end{cases} \quad (7)$$

반면에 $t \bmod N_p = 0$ 이고 t가 마지막 단계가 아니면 새 사용자가 요청한 콘텐츠를 전달하기 시작한다. 사용자 요청 콘텐츠는 기록에 따라 독립적으로 변경되므로 요청 사용자 위치 $(x_{\text{user},t+1}, y_{\text{user},t+1})$와 청크 μ_{t+1}은 이전 요청과 독립적으로 결정된다. 따라서, $a_t = (k, \check{\mu}_k)$에서의 작용은 다음과 같이 구성된 s_t에서 s_{t+1}로 상태 전환으로 이어진다.

$$\mu_{t+1} = \mu_{i+1,t},$$
$$\left(x_{\text{user},t+1}, y_{\text{user},t+1}\right) \in \mathcal{C},$$
$$\mathcal{M}_{k',t+1} = \begin{cases} \mathcal{M}_{k',t}, & if\ k' \neq k \\ \mathcal{M}_{k',t}\{\check{\mu}_k\} \cup \{\mu_t\}, & if\ k' = k \end{cases} \quad (8)$$

다음 청크 μ_{t+1}은 새로운 요청 콘텐츠 $\mu_{i+1,t}$의 초기 청크가 된다. 사용자 위치$\left(x_{\text{user},t+1}, y_{\text{user},t+1}\right)$는 $\mathcal{C}$ 에 속하는 독립적인 무작위 좌표이다. 캐싱 세트는 (7)과 같은 방식으로 액션에 의해 업데이트한다.

- Reward : 시간 단계 t의 액션 a_t 이후 보상을 받을 수 있다.

$$r_{t+1} = -\tau_t, \tag{9}$$

(3)에서 볼 수 있듯이 보상은 관련 SBS k와 캐시 집합 $\mathcal{M}_k$에 따라 분포가 달라지는 랜덤 변수이다. 감쇄 계수 $\gamma \in [0,1]$ 에서 시간 단계 t 의 보상은 다음과 같이 정의한다.

$$G_t = \sum_{t'=1}^{TN_p} \gamma^{t'-t} r_{t'+1} = -\sum_{t'=1}^{TN_p} \gamma^{t'-t} \tau_{t'}, \tag{10}$$

따라서, MDP 프레임워크에서 목표는 $t \in \{1, \ldots, TN_p\}$에 대한 기대보상 (10)을 최대화하기 위해 정책 (policy) 이라고 불리는 최적의 조치 선택 방법을 도출하는 것이다. 기대보상의 최대화는 T 연속 콘텐츠 요청을 처리하기 위한 기대 지연 시간의 최소화와 동일하다.

3.2. Deep Q-learning

MDP에 대한 솔루션을 얻기 위한 도구로 Q-learning을 사용하여 누적 지연 시간을 최소화하기 위한 효과적인 사용자 접속과 캐시 교체 전략을 도출한다. 매 시간 단계 t마다 Q-value이라고 하는 평균 보상이

state-action 쌍을 평가하기 위해 계산되고 전략은 Q-value를 기반으로 업데이트된다. 작용 $a \in \mathcal{A}$ 및 상태 $s \in \mathcal{S}$ 에 대한 Q-value은 다음과 같이 정의한다.

$$Q(s,a) = \mathbb{E}[G_t | s_t = s, a_t = a], \quad (11)$$

MDP에서는 가능한 캐싱 세트와 연속적인 사용자 좌표를 가진 상태 $\binom{LN_p}{M}^K$가 무수히 많기 때문에 가능한 모든 state-action 쌍에 대한 Q-value을 계산하고 저장하기 어렵다. 따라서 DNN을 활용하여 state-action 쌍과 이에 상응하는 기대 보상 사이의 관계를 근사화 하는 DQN 알고리즘을 [29] 활용한다. 즉, DQN은 모든 state-action 쌍에서의 Q-value을 계산하지 않고, (12)와 같이 모든 state-action 쌍의 실제 Q-value과 근접한 DNN 기반 action-value 함수를 구성한다.

$$Q(s,a) \approx Q(s,a;\theta), \quad (12)$$

여기서 $\theta \in \mathbb{R}^D$는 DNN 모델의 매개 변수를 나타낸다.

입력 피처 $\mathrm{X}(s)$ 는 상태 s의 수치 표현을 DNN 모델 처리에 용이하게 하도록 정의된다. 입력 특징 $\mathrm{X}(s)$의 경우, DNN 모델 $\zeta(\mathrm{X}(s);\theta)$의 상태 s에서 모든 $|\mathcal{A}|$ 동작의 대략적인 Q-value을 반환하도록 훈련된다. 즉, Q-value $Q(s,a)$는 DNN 모델 $\zeta(\mathrm{X}(s);\theta)$의 출력 벡터의 일부이다. 매개변수 θ를 훈련시키기 위한 Q-network 알고리즘은 알고리즘 1에

제시되어 있다. 알고리즘 1의 7번째 문장에서 $w_a \sim \mathcal{N}(0, \sigma_w^2)$ 는 미개척 동작 탐색을 위한 인공 잡음를 나타낸다. 알고리즘 1에서 훈련된 파라미터로, 전략은 현재 상태 $s \in \mathcal{S}$ 의 Q-value가 최대값을 가지는 동작 $a \in \mathcal{A}$을 선택하여 결정한다.

$$\pi(a|s) = \begin{cases} 1, \ if \ a = \underset{a' \in \mathcal{A}}{\text{argmax}} \, Q(s, a'; \theta) \\ 0, \ \text{otherwise} \end{cases}, \qquad (13)$$

Ⅳ. 사용자 접속 및 캐시 교체를 위한 신경망 설계

Universal approximation theorem에 따르면 신경망 모델은 node의 숫자가 충분히 많다면 2개의 layer 만으로도 모든 연속 함수를 충분히 근사할 수 있지만[30] 정확한 Q-value 근사를 위해 매개 변수 θ를 훈련하는 것은 간단하지 않다. 따라서 매개변수 훈련이 용이하도록 문제의 특성을 고려하여 입력 특징 X와 DNN 모델 $\zeta(\mathrm{X}(s))$를 신중하게 설계해야 한다.

제안된 설계에서, 우리는 입력 특징 X를 state-action 함수 행렬을 $\left(\mu_t, \left(x_{\mathrm{user},t}, y_{\mathrm{user},t}\right), \mathcal{M}_{\mathrm{all},t}\right)$ 구성한다. 구체적으로, 모든 콘텐츠의 청크는 $j \in \{1, \ldots, LN_p\}$에 의해 인덱스 되고 $l \in \{1, \ldots, L\}$ 콘텐츠는 $j \in \{(l-1)N_p + 1,\ (l-1)N_p + 2, \ldots, lN_p\}$ 에 의해 인덱스 되는 청크로 구성된다. 인덱싱을 기반으로 각 청크 j는 one-hot encoding 벡터 $\mathbf{q}_j^T \in \{0,1\}^{1\times LN_p}$ 로 표현된다. 다시 말해, 요청된 청크 j는 j번째 엔트리에서 1을 제외한 모든 엔트리가 0인 벡터 $\mathbf{q}_j$로 표시된다. K SBS의 캐시 상태를 나타내기 위해 k번째 행과 j번째 열의 항목 $\mathbf{M} \in \{0,1\}^{K\times LN_p}$가 SBS k에 캐시된 경우 1이고 그렇지 않은 경우 0인 행렬 $\mathbf{M}$을 정의한다. SBS의 메모리 용량 때문에 매트릭스 $\mathbf{M}$의 k번째 행에 있는 항목은

Algorithm 1 Deep Q-Network [29]

Initialize replay memory $\mathcal{B}$
Initialize parameters θ randomly
Initialize target parameters $\bar{\theta} \leftarrow \theta$
for episode = 1, E **do**
  for $t = 1, N_{\mathrm{p}}T$ **do**
    Observe state s_t
    Select action $a_t = \arg\max_{a \in \mathcal{A}} Q(s_t, a; \theta) + w_a$
    Observe reward r_t and next state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{B}$
    Sample mini-batch of (s_k, a_k, r_k, s_{k+1}) from $\mathcal{B}$
    $g_k \leftarrow \begin{cases} r_k & , k+1 = N_{\mathrm{p}}T \\ r_k + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a'; \bar{\theta}) & , k+1 \neq N_{\mathrm{p}}T \end{cases}$
    Perform a stochastic gradient descent step on $(g_k - Q(s_k, a_k; \theta))^2$ with respect to θ
    Reset target parameters $\bar{\theta} \leftarrow \theta$
  end for
end for

$\sum_{j=1}^{LN_p} m_{k,j} \leq M, k \in (1, 2, \ldots, K\}$을 만족한다. K 액세스 링크의 전송 실패 확률은 벡터 $\mathbf{P}_{AC} = [P_{AC,1}, P_{AC,2}, \ldots, P_{AC,K}]^T$로 나타낸다. 요청된 청크 $\mathbf{q}_j$, 캐시 상태 $\mathbf{M}$ 및 전송 실패 확률 P_{AC}를 기반으로 k번째 행과 j번째 열의 항목이 다음과 같은 피처 행렬 $X \in \mathbb{R}^{K \times LN_p}$을 정의한다.

$$x_{k,j} = \theta_1^{(0)}\left(1 - p_{\mathrm{AC},k}\right) + \theta_2^{(0)} q_j + m_{k,j} \qquad (14)$$

여기서 $\theta_1^{(0)}, \theta_2^{(0)} \in \mathbb{R}$는 의사 결정 시 전송 신뢰성 $1 - p_{AC}$, 요청된 청크 $\mathbf{q}_j$, 캐시 상태 $\mathbf{M}$의 가중치를 주기 위한 훈련 가능한 매개변수이다.

그림. 2 는 2개의 1D-convolution (Conv) 계층과 3개의 fully connected

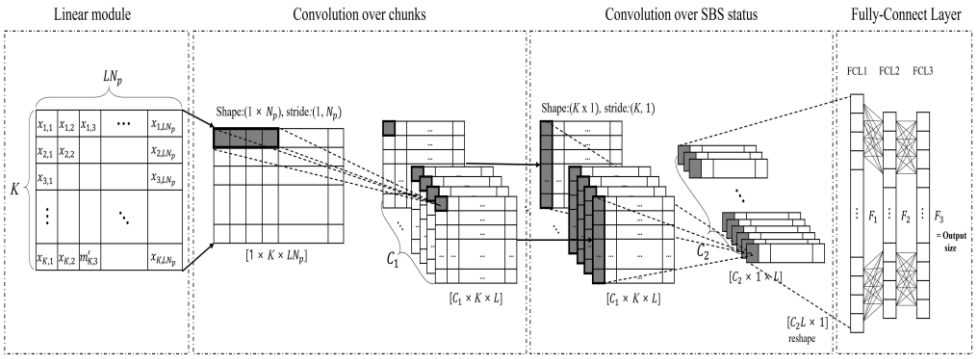

그림. 2 제안하는 DNN 구조

(FC) 계층으로 구성된 제안된 DNN 구조를 보여준다. 제안된 DNN 구조에서 Conv 계층은 문제의 두 가지 주요 특징을 기반으로 설계한다. 먼저 콘텐츠 전달 과정에서 요청된 콘텐츠에 대한 N_p개의 청크가 연속적으로 사용자에게 전달되어야 한다. 즉, $j \bmod N_p \neq 0$인 경우 j 청크를 전달한 후 청크 $j+1$을 전달해야 한다. 이러한 이유로 요청 내용 중 전송되지 않은 나머지 청크의 캐시 상태는 전송 전에 다른 청크로 교체하지 않도록 하는 것이 중요하다. 반면, $j \bmod N_p = 0$ 이면 청크 $j+1$의 전달은 청크 j의 전달과 독립적이다. 이러한 콘텐츠의 N_p 청크 간 상관관계를 바탕으로 제안된 DNN의 첫 번째 레이어로 콘텐츠 크기에 맞춘 C_1 필터가 있는 Conv 레이어를 도입한다. 구체적으로, 각 필터 $c_1 \in \{1, 2, \ldots, C_1\}$는 훈련 가능한 매개변수 $\theta_{c_1}^{(1)} \in \mathbb{R}^{1 \times N_p}$로 구성

표. 1 제안된 DNN 규격

	Conv1	**Conv2**	**FC1**	**FC2**	**FC3**
입력 크기	$K \times LN_p$	$C_1 \times K \times L$	$C_2 L$	N_{L1}	N_{L2}
필터 크기	$1 \times N_p$	$K \times 1$	−	−	−
행, 열의 stride	$1, N_p$	$K, 1$	−	−	−
필터 수	C_1	C_2	−	−	−
활성 함수	Leaky ReLU				
출력 크기	$C_1 \times K \times L$	$C_2 \times L$	F_1	F_2	KLN_p

된 행 벡터이며, 열과 행을 따라 피처 행렬 X를 사용한 convolution 연산의 진행범위는 각각 N_p와 1이다. 둘째, 교체할 캐시 청크를 선택할 때 여러 SBS에서 중복된 청크 캐싱은 캐시 사용 SCN의 캐시 적중률을 관리하는 데 중요한 정보이다. 캐시가 중복적으로 저장 되어있는 청크를 교체하면 캐시 적중률을 증가시킬 수 있다. 상태 정보에서 중복적인 캐시 상태 추출을 용이하게 하기 위해 제안된 DNN의 두 번째 계층으로 C_2 필터가 있는 Conv 계층을 도입한다. 구체적으로, 각 필터 $c_2 \in \{1, 2, \ldots, C_2\}$는 훈련 가능한 매개변수, $\theta_{c_2}^{(2)} \in \mathbb{R}^{K \times 1}$로 구성된 열 벡터이며, 첫 번째 레이어의 출력의 convolution 연산의 보폭 (stride) 은 1이다. 일반적인 CNN (convolution neural network)과 달리

Conv 레이어 뒤에는 pooling 레이어가 없다. 두 개의 Conv 레이어에 이

어 세 개의 FC 레이어가 추출된 기능을 기반으로 현재 상태의 모든 작업의 대략적인 Q-value을 생성한다. 3개의 FC 계층의 폭은 F_1, F_2, $F_3 = KLN_p$ 이다. 입력 x에서 $\max\{x, 0.01x\}$을 계산하는 leaky rectified linear unit (ReLU)은 모든 Conv 및 FC 레이어의 활성화 함수로 채택한다. 제안된 DNN 구조의 규격은 표. 1 에 정리되어 있다.

Ⅴ. 시뮬레이션 결과

이 장에서는 컴퓨터 시뮬레이션을 통한 학습 기반 사용자 접속 및 캐시 교체 전략의 성능 평가를 보여준다. 별도로 명시되지 않은 한 시뮬레이션 환경 구성을 표. 2에 명시된 시스템 매개 변수로 사용한다. 각 SBS의 메모리는 콘텐츠 요청확률의 사전 정보가 가정되지 않기 때문에 학습 초기 단계에서 메모리는 무작위 M 청크로 점유되어 있다고 가정한다. 기존 기법과의 성능 비교를 위해 다음과 같은 캐시 교체 및 사용자 접속 전략을 고려한다.

- No-cache : 모든 K SBS에 캐싱 기능이 없다고 가정한다. 사용자는 가장 가까운 SBS와 접속되고 관련 SBS는 요청된 콘텐츠의 모든 청크를 CU로부터 수신하여 사용자에게 전송한다.
- Distance-based user association and least frequently used replacement (DUA-LFU) : 사용자는 가장 가까운 SBS와 접속하고 연결된 SBS는 CU로 부터 캐시 되지 않은 청크를 수신 받아 메모리가 오버플로 될 때마다 가장 자주 사용되지 않는 청크를 제거한다.
- Cache-based user association and least frequently used replacement

(CUA-LFU) : 사용자는 요청된 청크를 캐시한 SBS에 우선적으로 접속하고 SBS는 메모리가 오버플로될 때마다 가장 자주 사용되지 않는 청크를 제거한다. 요청된 청크가 어느 SBS에 캐시 되어 있지 않은 경우 DUA-LFU와 동일한 방식으로 동작한다.

또한 제안된 DNN 구조와 입력 피처 설계 효과를 확인하기 위해 다음과 같은 기본 DQN 알고리즘을 고려한다.

- DQN with fully connected network (DQN-FCN): 제안된 방식과 동일하게 알고리즘 1에 따라 학습 및 의사결정을 내린다. 그러나, 제안된 기법과 달리, 5개의 FC 계층을 갖는 FCN을 고려하며, 입력 피처는 다음과 같이 단순히 상태 정보의 나열로 정의 한다.

$$\mathrm{x}_{FCN} = [\mathbf{m}^{\mathrm{T}}, \mathbf{q}_j^{\mathrm{T}}, \mathbf{P}_{\mathrm{AC}}^{\mathrm{T}}]^{\mathrm{T}} \quad (15)$$

여기서 $\mathbf{m} \in \{0,1\}^{KLN_p \times 1}$ 는 M의 열들의 결합인 이진 캐시 상태를 나타낸다.

제안된 기법과 DQN-FCN의 훈련 프로세스는 표. 3에 설명된 하이퍼 매개 변수 (hyper-parameter) 를 사용한 adaptive moment estimation (Adam) 최적화 도구 [31]를 채택한다. 제안된 DNN 구조에서 필터의

표. 2 시스템 환경

파라미터	값
컨텐츠 개수, L	20
컨텐츠 당 청크, N_p	4
SBS 의 개수, K	4
SBS 당 메모리 용량, M	16
Access link 의 path loss 지수, α_{AC}	3.6
Zipf's 지수, s	0.8
Coverage, $(x_{\max}, y_{\max})$ [m]	(250, 250)
전송 SNR, ρ [dB]	90
전송 비, R [bits/s/Hz]	1
백홀 링크 지연, τ_c [ms]	5
액세스 링크 지연, τ_s [ms]	1
에피소드 당 컨텐츠 요청 수, T	1,000

표. 3 학습에서 사용된 하이퍼-파라미터

파라미터	값
Learning rate, η	0.001
Discount factor, γ	0.99
Maximum replay memory, $\mathcal{B}$	1,000,000
Mini-batch size	4,096

수는 C_1 = C_1 = 20이고 세 FC 레이어의 단위 수는 F_1 = 400, F_2 = 320, F_3 = 320이다. DQN-FCN의 DNN 구조에서, 5개의 FC 층의 단위 수는 F_1 = 400, F_2 = 400, F_3 = 400, F_4 = 320, F_5 = 320이다.

그림. 3은 에피소드가 진행됨에 따라 단일 콘텐츠를 전송하는 평균 지연 시간을 보여준다. No-cache와 LFU 기반 기준 기법 (DUA-LFU 및

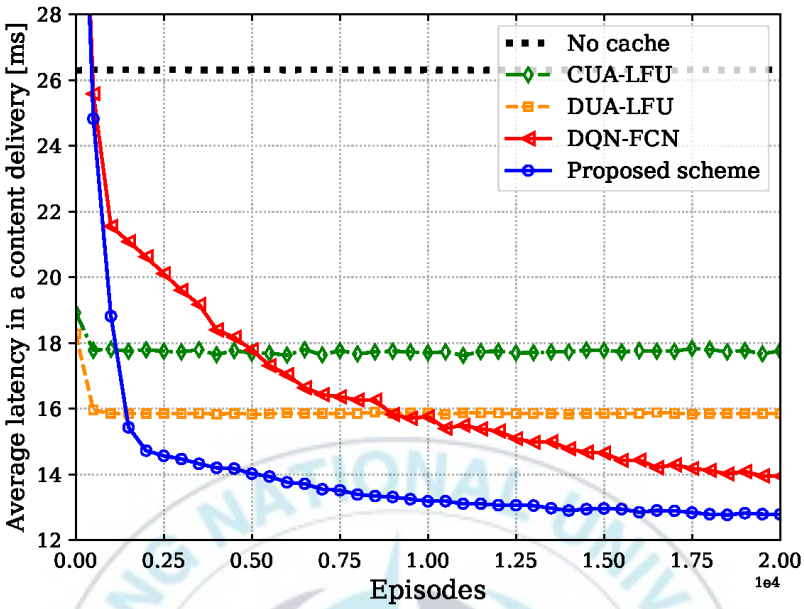

그림. 3 에피소드에 대한 평균 지연 시간.

CUA-LFU) 사이의 큰 성능 차이는 캐시 기능이 평균 지연 시간 측면에서 상당한 성능 향상을 제공할 수 있음을 보여준다. 학습이 없는 세 가지 기본 기법과 달리 DQN 기반 기법, 즉 DQN-FCN과 제안된 기법은 에피소드 경험을 통해 지속적으로 전략을 개선한다. 결과적으로 DQN 기반 기법은 충분히 상태를 경험한 후 기존 기법을 능가하는 것으로 나타난다. 특히, 제안된 기법은 DQN-FCN보다 수렴 속도 및 평균 지연 시간 측면에서 높은 성능을 달성하며, 이는 제안된 입력 피처 설계와 DNN 구조의 효과를 검증한다.

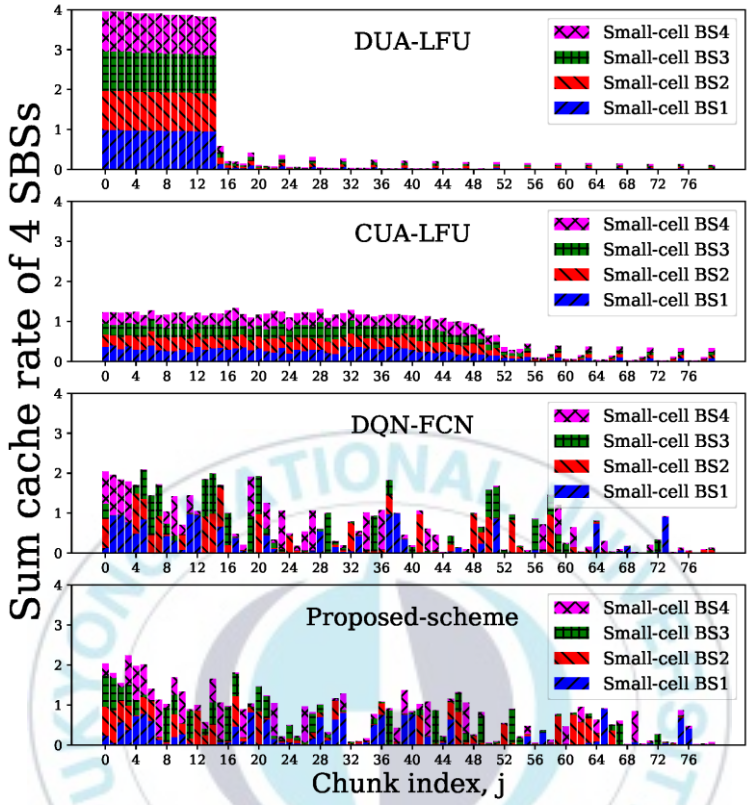

그림. 4 청크 캐시 수에 대해 정규화된 히스토그램.

그림. 4는 10,000개의 연속 콘텐츠 요청을 처리하는 $K = 4$ SBS에서 청크 캐시 카운트를 정규화한 히스토그램을 보여준다. 예를 들어, 청크 j가 항상 모든 SBS에 캐시 되어 있는 경우 그림의 청크 j에 해당하는 스택 막대 길이는 4가 되고 4개의 동일한 크기 막대가 된다. DQN 기반 기법의 경우, 20,000 에피소드 학습 후 훈련된 DNN을 사용하여 시뮬레이션 결과를 얻는다. LFU 기반 기준 기법의 캐시 카운트는 요

청확률 기반 캐시 대체로 인해 DQN 기반 기법에 비해 일부 인기 콘텐츠에 상대적으로 더 집중되어 있는 것으로 나타난다. 또한 LFU 기반 기법에서 모든 SBS의 캐시 분포는 거의 동일한 것으로 나타난다. 각 SBS가 다른 SBS의 캐시 상태와 독립적으로 동일한 요청확률 기반 캐시 교체를 실시하고 있기 때문이다. 반면에 DQN 기반 기법의 캐시 카운트는 사용자 접속 및 캐시 교체를 함께 최적화하여 캐시 적중률을 개선하기 위해 협력 캐싱 개념을 추가로 고려하기 때문에 상대적으로 분산되어 있다. 특히 일부 인기 콘텐츠는 LFU 기반 방식과 유사하게 여러 SBS에 캐시 되지만, 비인기 콘텐츠는 캐시 적중률을 높이기 위해 단일 또는 소수의 SBS에만 캐시 된다. 그러나 일부 비인기 콘텐츠가 불필요하게 많은 SBS에서 캐시 되는 DQN-FCN의 관찰을 통해 DQN-FCN에서 학습한 전략이 완전하지 않는다는 것을 알 수 있다. 제안 기법은 비인기 콘텐츠의 협력 캐싱을 촉진하는 전략을 학습하는 것으로 나타나며, 이는 그림.3 에서 제안된 기법이 DQN-FCN보다 낮은 평균 지연 시간을 달성하는 이유 중 하나로 추론할 수 있다.

그림. 5−7은 시스템 매개변수에 대한 평균 지연시간의 영향을 보여준다. 이 모든 수치에서 DQN 기반 결과는 사전 20,000 에피소드 동안 훈련된 DNN을 사용하며, 다른 기법들과 성능 차이가 큰 No-cache의

결과는 생략한다.

그림. 5는 시스템 커버리지 크기 $x_{\max}$ 및 $y_{\max}$에 대해 단일 콘텐츠를 전송하는 평균 지연 시간을 보여준다. 시스템 커버리지는 통신 링크 거리와 직접적으로 관련이 있기 때문에, 넓은 시스템 커버리지는 식 (1)와 (2)와 같이 높은 전송 실패 확률로 이어진다. 결과적으로, 평균 지연 시간은 모든 기법에서 커버리지 크기에 따라 증가한다. 그러나 평균 지연 시간의 증가 속도는 각 기법마다 다른 것으로 나타난다. 전송 실패 확률이 매우 낮고 커버리지가 작을 때, 통신 신뢰성은 사용자 접속 및 캐시 교체의 최적화에서 무시될 수 있다. 통신 신뢰성을 고려하지 않고, 최적의 전략은 캐시 적중률을 최대화하여 CUA-LFU로 줄일 수 있도록 하는 데 초점을 맞춘다. 제안된 방식은 작은 커버리지 크기에서 CUA-LFU와 거의 동일한 평균 지연 시간을 달성하는 것으로 나타난다. 반면에, 넓은 커버리지 크기에서 전송 실패 확률이 높을 때, 통신 신뢰성은 지배적인 요소가 되어 최적화가 통신 신뢰도 향상에 초점을 맞출 필요가 있다. 즉, 최적의 전략은 대규모 커버리지 에서 DUA-LFU이 된다. 또한 커버리지가 증가함에 따라 제안된 기법의 평균 지연 시간이 DUA-LFU의 지연 시간에 가까워지는 것으로 나타난다.

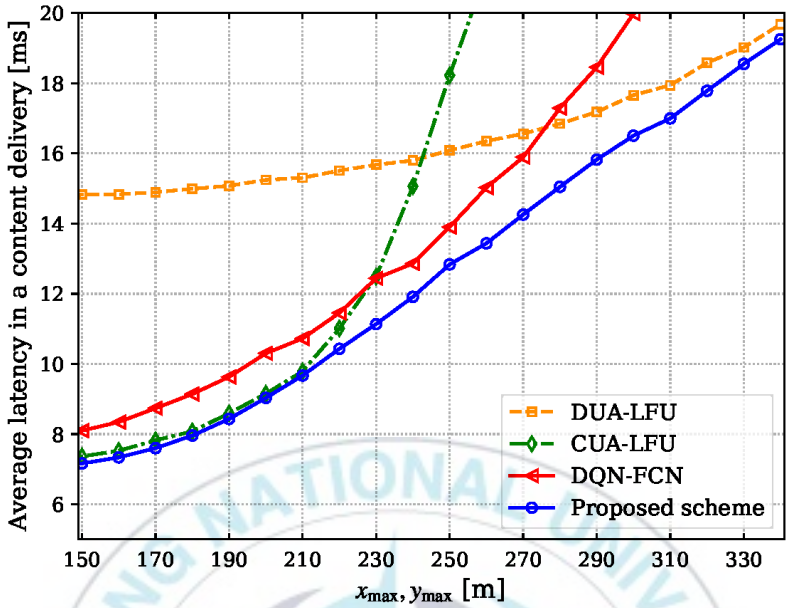

그림. 5 커버리지에 대한 평균 지연 시간, $x_{\max}$ 및 $y_{\max}$.

이러한 결과를 통해, 우리는 제안된 기법이 주어진 커버리지 크기에 적합한 효과적인 전략을 배울 수 있음을 알 수 있다.

그림. 6은 콘텐츠 요청확률에 대해 단일 콘텐츠를 전송하는 평균 지연 시간을 보여준다. 콘텐츠의 요청확률은 Zipf의 지수 s에 따라 변화하며, 몇몇 인기 콘텐츠의 중복 요청이 캐시 적중률을 높여 지연 시간을 줄일 수 있다. 그 결과, 모든 기법에서 평균 지연 시간이 s와 함께 감소한다. 지수 s가 낮아진다면 요청확률 분포가 분산되어, 제안 기법은 SBS 중복을 방지하여 청크를 캐시 할 수 있도록 협력 캐시 하므로 상당한 성능 향상을 제공한다. 반면에, 지수 s가 높아 요청확률 분포가

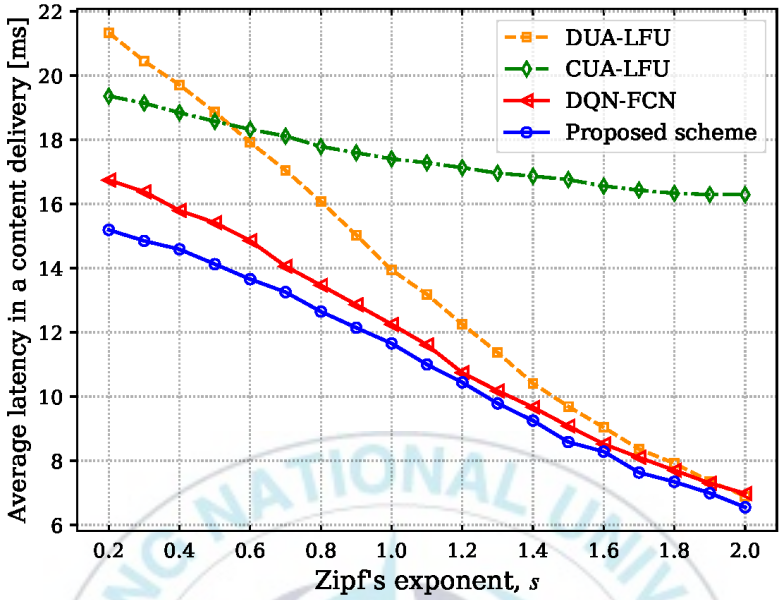

그림. 6 콘텐츠 요청확률에 대한 평균 지연 시간, s.

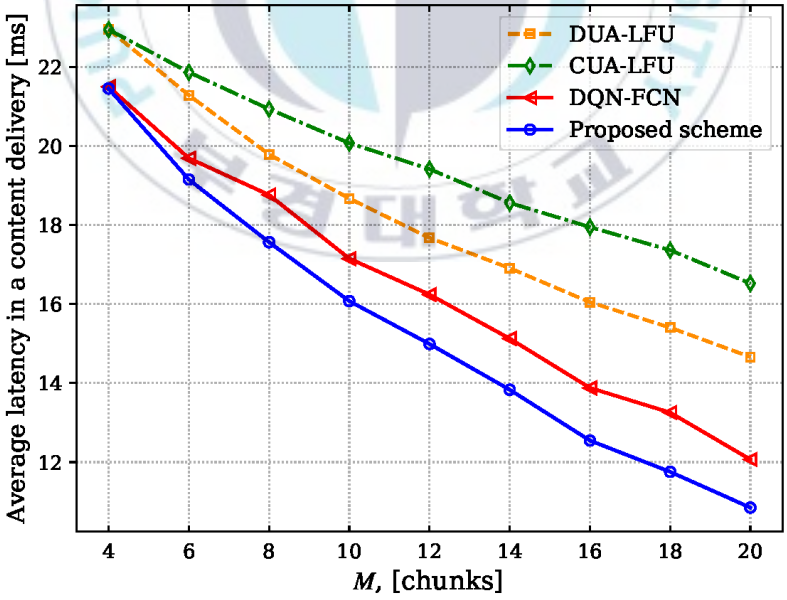

그림. 7 SBS, M의 메모리 용량에 대한 평균 지연 시간.

크게 편향된 경우, 제안된 계획은 DUA-LFU와의 작은 성능 차이를 보여준다. 높은 s 에서는 캐시 적중률을 높이기 위해 모든 SBS 가 가장 인기 있는 콘텐츠를 캐시 해야 하므로 s가 증가할수록 최적의 전략은 DUA-LFU에 근사한다.

그림. 7은 SBS, M의 메모리 용량에 대해 단일 콘텐츠를 전송하는 평균 지연 시간을 보여준다. 모든 기법에서 M이 있을 때 평균 지연 시간이 감소하고, 제안된 기법은 M과 관계없이 다른 모든 기본 기법들보다 성능이 우수하다. 제안된 기법과 다른 기법 사이 성능 격차가 M과 함께 커진다는 결과로부터, 메모리 기술의 급속한 발전과 함께 사용자 접속과 캐시 교체의 최적화가 점점 더 중요해 질 것임을 기대 할 수 있다.

그림. 8에서는 시간이 지남에 따라 콘텐츠의 요청확률이 변하는 non-stationary 시나리오를 고려한다. 내용의 요청 확률은 500회마다 5위씩 순환적으로 이동한다고 가정한다. 예를 들어 콘텐츠 $l \in \{1, 2, \dots, L\}$의 요청 확률 q_r은 500회 후 $r > 5$이면 q_{r-5}, $r \leq 5$이면 q_{L+r-5}로 변경된다. 시뮬레이션 결과는 콘텐츠 요청확률의 변화가 모든 기법에서 평균 지연 시간을 즉시 증가시키지만 어느정도 에피소드가 경과하면 다시 감소한다는 것을 보여준다. DUA-LFU는 콘텐츠 요청확률이 변화할 때

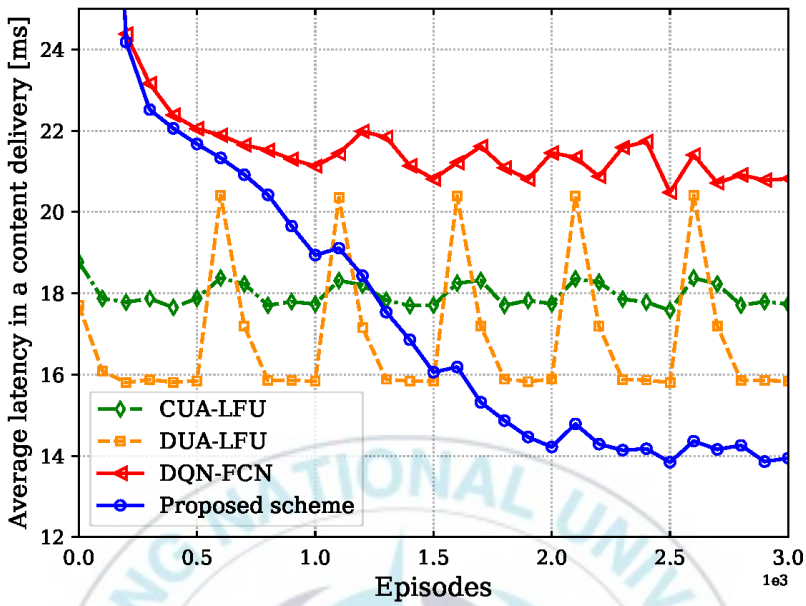

그림. 8 콘텐츠 요청확률 변동에 대한 평균 지연 시간.

다른 기법에 비해 상대적으로 높은 지연 시간 증가를 보여준다. 이러한 현상은 그림. 4와 같이 DUA-LFU의 캐시 분포가 소수의 인기 콘텐츠에 고도로 집중되어 있기 때문이다. 다시 말해 모든 SBS가 DUA-LFU에서 거의 동일한 청크를 캐시하기 때문에 콘텐츠 요청확률 변화에 따라 캐시 적중률이 크게 영향 받는다. 나아가 DQN-FCN은 학습 학습속도가 느려서 학습에 실패하지만, 제안된 기법은 훈련 과정이 가속화되어 훈련 과정 중 콘텐츠 요청확률 변화에도 불구하고 효과적인 전략을 성공적으로 학습할 수 있다는 점을 확인 할 수 있다.

Ⅵ. 결론

본 논문에서는 SBS가 캐시 기능을 가진 SCN에서 콘텐츠 제공 지연 시간을 최소화하기 위한 사용자 접속 및 캐시 교체 전략을 심층 강화 학습을 활용해 도출하는 기법을 제안했다. MDP와 순차적 의사결정 문제를 공식화하고 효과적인 전략 도출을 위해 DQN 알고리즘에서 문제 특성이 고려된 새로운 DNN 구조와 입력 피처 매트릭스를 설계했다. 다양한 상황에서 시뮬레이션 결과, 제안된 기법이 기존 사용자 접속 및 캐시 교체 전략 뿐만 아니라 단순한 DNN 구조를 가진 기존 DQN 알고리즘을 평균 대기 시간 측면에서 능가하는 것으로 나타난다. 제안된 기법과 기존 기법 사이의 성능 격차가 SBS의 메모리 용량에 따라 커진다는 관찰로부터, 메모리 기술의 급속한 발전과 함께 적절한 최적화된 전략이 점점 더 중요해 질 것임을 기대할 수 있다. 또한 학습 도중 콘텐츠 요청확률이 변하더라도 제안된 방식은 새로운 콘텐츠 요청 확률에 빠르게 적응하는 것을 확인했다.

References

[1] C. V. N. Index, "Cisco visual networking index: Forecast and methodology 2017-2022," *White paper, CISCO*, Feb. 2015.

[2] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, "Femtocells: Past, present, and future," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 497–508, 2012.

[3] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Base-station device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, 2014.

[4] X. Ge, H. Cheng, M. Guizani, and T. Han, "5G wireless backhaul networks: challenges and research advances," *IEEE Netw.*, vol. 28, no. 6, pp. 6–11, 2014.

[5] L. Liu, S. Bi, and R. Zhang, "Joint Power control and fronthaul rate allocation for throughput maximization in OFDMA-based cloud radio access network," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4097–4110, 2015.

[6] R. G. Stephen and R. Zhang, "Joint millimeter-wave fronthaul and OFDMA allocation in ultra-dense CRAN," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1411–1423, 2017.

[7] U. Siddique, H. Tabassum, E. Hossain, and D. I. Kim, "Wireless backhauling of small cells: Challenges and solution approaches," *IEEE Wireless Commun.*, vol. 22, no. 5, pp. 22–31, 2015.

[8] J. Park, J.-P. Hong, and S. Beak, "Optimal beamforming with limited feedback for millimeter-wave in-band full-duplex mobile X-haul network," *IEEE Access*, vol. 6, pp. 51 038–51 048, Sep. 2018.

[9] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: The role of proactive caching in 5G,” *IEEE Commun. Mag*., vol. 52, no. 8, pp. 82–89, 2014.

[10] J.-P. Hong and W. Choi, “User prefix caching for average playback delay reduction in wireless video streaming,” *IEEE Trans. Wireless Commun*., vol. 15, no. 1, pp. 377–388, 2016.

[11] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, “Analyzing the video popularity characteristics of large-scale user generated content systems,” *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, 2009.

[12] X. Cheng, J. Liu, and C. Dale, “Understanding the characteristics of internet short video sharing: A youtube-based measurement study,” *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1184–1194, Aug. 2013.

[13] G. Ma, M. Zhang, J. Ye, M. Chen, and W. Zhu, “Understanding performance of edge content caching for mobile video streaming,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1076–1089, 2017.

[14] P. Blasco and D. G¨und¨uz, “Learning-based optimization of cache content in a small cell base station,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014.

[15] S. H. Chae and W. Choi, “Caching placement in stochastic wireless caching helper networks: Channel selection diversity via caching,” *IEEE Trans. Wireless Commun*., vol. 15, no. 10, pp. 6626–6637, 2016.

[16] S. H. Chae, T. Q. S. Quek, and W. Choi, “Content placement for wireless cooperative caching helpers: A tradeoff between cooperative gain and content diversity gain,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6795–6807, 2017.

[17] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: Wireless content delivery through distributed

caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[18] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1382–1393, 2017.

[19] J. Song, H. Song, and W. Choi, "Optimal content placement for wireless femto-caching network," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4433–4444, 2017.

[20] J. Song, M. Sheng, T. Q. Quek, C. Xu, and X. Wang, "Learning-based content caching and sharing for wireless networks," *IEEE Trans. Commun.*, vol. 65, no. 10, pp. 4309–4324, Jun. 2017.

[21] T. Zhang, S. Biswas, and T. Ratnarajah, "On the performance of cache-enabled hybrid wireless networks," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1818–1834, 2021.

[22] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-aho, "Content-aware user clustering and caching in wireless small cell networks," in *Proc. IEEE 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, 2014, pp. 945–949.

[23] B. Dai andW. Yu, "Joint user association and content placement for cache-enabled wireless access networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2016, pp. 3521–3525.

[24] W. Jing, X. Wen, Z. Lu, and H. Zhang, "User-centric delay-aware joint caching and user association optimization in cache-enabled wireless networks," *IEEE Access*, vol. 7, pp. 74 961–74 972, 2019.

[25] J. Sung, K. Kim, J. Kim, and J. K. Rhee, "Efficient content replacement in wireless content delivery network with cooperative caching," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2016, pp. 547–552.

[26] W. Wang, R. Lan, J. Gu, A. Huang, H. Shan, and Z. Zhang, “Edge caching at base stations with device-to-device offloading,” *IEEE Access*, vol. 5, pp. 6399–6410, Mar. 2017.

[27] P. Wu, J. Li, L. Shi, M. Ding, K. Cai, and F. Yang, “Dynamic content update for wireless edge caching via deep reinforcement learning,” *IEEE Commun. Lett*., vol. 23, no. 10, pp. 1773–1777, 2019.

[28] C. Zhong, M. C. Gursoy, and S. Velipasalar, “Deep reinforcement learning-based edge caching in wireless networks,” *IEEE Trans. Cogn. Commun. Netw*., vol. 6, no. 1, pp. 48–61, Mar. 2020.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[30] G. Cybenko, “Approximations by superpositions of a sigmoidal function,” *Math. Contr. Signals, Syst*., vol. 2, pp. 303–314, 1989.

[31] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, Dec. 2015.