

모바일 애드-혹 무선 통신망의 스트리밍 서비스를 위한 부분적 캐싱

박 경 모^{1*} · 김 백 현²^{1*}가톨릭대학교 컴퓨터정보공학부 교수²인천대학교 기초교육원 교수

Partial Caching Scheme for Streaming Service of Mobile Ad-hoc Wireless Networks

Kyeongmo Park^{1*} · Back-Hyun Kim²^{1*}Professor, Computer Science & Information Engineering School, The Catholic University of Korea, 14662, Korea²Professor, Department Faculty of Liberal Education, Incheon National University, Incheon, 22012, Korea

[요 약]

모바일 애드-혹 무선 네트워크는 자유롭게 움직이는 노드들로만 구성된 네트워크이다. 이 네트워크에서는 노드들의 자유로운 이동으로 인하여 네트워크 위상은 시시각각 변화한다. 본 논문에서는 전송 중인 콘텐츠는 경로를 구성하는 노드 및 요청 노드에서 캐싱 된다. 지연 없는 스트리밍 서비스를 위하여 요청 노드는 콘텐츠의 현재 프레임에서 향후 재생에 필요한 연속된 프레임을 윈도우 크기만큼 이웃 노드에 요청하고 캐싱한다. 성능 평가에서는 캐시 적중률 및 전송 채널 비율의 관점에서 수신되는 데이터를 조건 없이 저장하는 적극적 캐싱 기법과 비교 분석하였다. 시뮬레이션 결과는 부분적 캐싱 기법이 높은 캐시 적중률과 낮은 전송 채널 수를 보여준다. 제안된 기법은 네트워크에서 노드의 가용 에너지 및 전송 효율을 증가시키기 위한 대안 기법이 될 수 있을 것이다.

[Abstract]

Mobile ad-hoc wireless networks are the networks comprised of freely moving nodes only. Due to the free movement of nodes in mobile ad-hoc networks, the network topology changes every moment. In this paper, the content being transmitted is cached at the nodes constituting the route and the requesting node. For the delayless streaming service, the requesting node requests and caches consecutive frames necessary for future playback from the current frame of the content to the neighboring node as much as the window size. The performance evaluation was compared and analyzed with the aggressive caching scheme that unconditionally stores received data in terms of cache hit ratio and transmission channel ratio. Simulation results show that our partial caching has a high cache hit ratio and a low number of transmission channels. Our proposed scheme may be an alternative technique to increase the available energy and transmission efficiency of nodes in mobile networks.

색인어 : 캐싱, 하이브리드 모바일 애드-혹 네트워크, 멀티미디어, 윈도우, 스트리밍 서비스**Keyword** : Caching, Hybrid Mobile Ad-hoc Networks, Multimedia, Window, Streaming Service<http://dx.doi.org/10.9728/dcs.2022.23.3.489>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 06 January 2022; Revised 21 February 2022

Accepted 08 March 2022

***Corresponding Author, Kyeongmo Park**

Tel: +82-2-2164-4365

E-mail: kpark@catholic.ac.kr

I. 서론

통신 기술의 발달은 대용량 고품질의 멀티미디어 데이터의 광범위한 사용을 가능하게 하였다. 이를 위하여 대용량의 데이터가 장시간 동안 신뢰성 있는 경로를 따라 전송될 필요가 있다. 5G 이동 통신 네트워크는 기가(Giga) bps 이상의 전송 속도를 지원한다[1]. 이 네트워크는 통신 인프라의 구축이 필요하며 서비스가 제공되기 위해서는 많은 시간과 비용이 필요하다. 또한 통신 인프라가 구축되지 않거나 장애가 발생하면 서비스가 제공되지 못한다는 문제점이 있다. 애드-혹 네트워크는 통신 단말기들만으로 구성된 네트워크이다[2], [3]. 통신 인프라가 필요 없으므로 단말기들은 통신에 관련된 모든 작업을 스스로 수행하여야 한다. 통신 단말기의 이동성을 보장하는 모바일 애드-혹 무선 네트워크는 제한된 가용 에너지, 노드의 이동으로 인한 빈번한 전송 경로 변경, 낮은 전송 대역폭 등의 특성으로 인하여 유선 네트워크에서 사용되는 기법을 적용하기 힘들다. 그러나 대용량 고품질의 멀티미디어 콘텐츠에 대한 소비자들의 요구는 증가하고 있다. 멀티미디어 콘텐츠는 서버와 요청 노드 사이의 경로에 대하여 장시간 동안 전송 대역폭을 잠식한다[4]. 서비스 요청률이 증가할수록 요구되는 전송 대역폭이 증가하며 전송 경로를 구성하는 노드들의 에너지 사용량 또한 증가한다. 이러한 특성은 멀티미디어 콘텐츠 서비스가 지연되거나 단절되는 현상을 초래할 수 있다.

캐싱은 전송 대역폭 및 노드 에너지의 가용성 문제를 해결할 수 있는 기법이다[5]. 캐싱 기법은 동일 콘텐츠를 서버만이 아니라 복수의 노드에 저장시킨다. 서비스 요청이 발생하면 해당 콘텐츠는 이를 저장하고 있는 노드 중에서 요청 노드와 가장 가까운 거리에 위치하는 노드에 의해 제공된다. 그러므로 콘텐츠가 서버로부터 전송되는 경우와 비교하여 전송 경로 수가 감소하기 때문에 네트워크 대역폭 및 노드 에너지 사용량이 감소한다. 노드의 저장 용량은 서버에서 제공하는 콘텐츠의 총량과 비교하여 상대적으로 매우 작다. 캐시 메모리가 가득 찬 경우 이미 저장된 콘텐츠를 삭제하여 새로운 저장 공간을 확보하여야 한다. 따라서 어떤 콘텐츠가 저장되고 삭제되어야 하는 것은 캐싱 성능을 좌우하는 중요한 문제이다. 모바일 애드-혹 네트워크에서 노드는 지속해서 이동하기 때문에 노드들 사이의 전송 경로는 시간에 대해 가변적이다. 다른 형태의 콘텐츠와 비교하여 상대적으로 큰 용량을 가지는 멀티미디어 콘텐츠는 전송하는 데 많은 시간이 소요된다. 콘텐츠의 중복적 저장을 감소시키기 위하여 일반적으로 노드를 지정하여 캐싱을 수행하도록 한다. 그러나 이러한 방식은 노드 이동성으로 인하여 캐싱 노드와 요청 노드 사이의 거리가 서버로부터 요청 노드 사이의 거리보다 길어질 수 있으므로 비효율적인 방식이 될 수 있다. 또한 캐싱 노드에 요청이 집중되는 경우 전송 대역폭 사용량 및 노드의 부하가 증가하는 문제점이 발생할 수 있다.

본 논문은 모바일 애드-혹 네트워크에서 멀티미디어 콘텐츠

를 위한 협업적 캐싱 기법을 제안한다. 캐싱은 지정된 특정 노드가 아니라 전송 경로를 구성하는 노드들에서 수행된다. 이 특성은 노드의 이동으로 인하여 캐싱 노드와 요청 노드 사이의 전송 길이가 길어지는 문제를 감소시킬 수 있다. 콘텐츠에 대한 서비스 요청이 특정 노드로 집중되지 않기 때문에 노드 부하는 감소할 수 있다. 캐싱 노드를 찾는 과정은 네트워크에 요청 패킷을 플래딩하지 않고 콘텐츠 서버로의 전송 경로 상에서만 수행된다. 요청 노드는 자신과 직접적으로 인접한 노드들로부터 요청된 멀티미디어 콘텐츠의 추가적인 프레임 요청을 받고 받는다. 이것은 전송 경로 및 블랙홀 문제 등으로 서비스가 지연 또는 단절로 인한 영향을 감소시킬 수 있다. 본 논문의 구성을 다음과 같다. 2장은 관련 연구에 관하여 기술한다. 3장에서는 본 논문에서 제안하는 윈도우 기반 캐싱 기법에 관해 기술한다. 4장에서는 제안된 기법을 요청률과 노드 최대 이동 속도에 대한 캐시 적중률 측면에서의 성능 평가에 대해 논한다. 마지막으로 5장에서는 결론을 기술한다.

II. 관련 연구

모바일 애드-혹 네트워크에서 노드는 지속적인 이동 특성을 갖도록 정의된다. 이동성 모델은 속도 v 와 방향 θ 을 시간 t 에 의존하여 다양하게 제안되었다[6], [7]. 노드가 정의된 목적지에 도달하면 새로운 목적지를 선택한다. 새롭게 선택되는 속도와 방향이 이전 속도와 방향과의 관계 여부에 따라 무작위 모델과 시간적 모델로 구분되어 진다. 무작위 모델은 새로운 목적지를 무작위로 선택하며, 이전 이동성과의 상관관계는 존재하지 않는다. 이로 인하여 목적지에 도달할 때까지 이동 특성은 같게 유지되며, 목적지에 도달하면 노드의 속도나 이동 방향의 급격한 변화가 발생할 수 있다. 시간적 모델은 무작위 모델과는 다르게 시간을 기준으로 새로운 이동성을 정의한다. 새롭게 정의되는 이동성은 이전 시간까지의 이동성에 영향을 받으며 매 시각 속도와 방향에 변화가 발생할 수 있다. 이동성 모델에 따라 특정 지역이 상대적으로 높은 노드 밀집도를 가질 수 있으며 다른 노드와 연결이 끊기는 블랙홀 현상이 발생하기도 한다. 이것은 전송 경로 수 및 서비스 지속성에 영향을 줄 수 있다.

모바일 애드-혹 네트워크는 노드의 이동성으로 인하여 노드 사이의 최적 연결성을 보장하기 위한 지속적인 경로 탐색, 설정 및 유지를 위한 과정이 필요하다. 이 과정은 서비스 요청 시 경로의 설정 여부에 따라 선행(proactive)방식과 반응(reactive)방식으로 구분된다[8]-[11]. 전자는 서비스 요청과는 상관없이 전체 네트워크에 존재하는 노드들 사이의 전송 경로를 주기적으로 미리 설정하는 방식이다. 이것은 서비스 발생 시 경로 탐색 과정이 필요 없으므로 지연 시간을 감소시킬 수 있다. 그러나 서비스 요청이 없는 경우에도 경로 설정 과정을 수행하기 때문에 네트워크 대역폭 및 노드 에너지의

불필요한 소비가 발생한다. 반응방식은 서비스 요청이 발생하는 경우에만 해당 노드로의 경로를 설정한다. 즉 경로 설정 과정이 필요한 경우에만 수행되기 때문에 전송 대역폭과 노드 에너지의 효율적인 사용을 가능하게 한다. 그러나 서비스 제공에 앞서 경로 설정 과정이 필요하므로 서비스 지연이 발생한다. 전송 경로가 실시간적으로 관리되면 최적의 효율성을 나타내지만, 에너지 소모가 증가하게 되어 노드 가용성에 문제가 발생한다. 비컨(beacon)처럼 일정 시간마다 경로 관리를 수행하면 노드 가용성 및 전송 대역폭 사용 효율은 향상되지만, 이동성에 따라 최신 경로 정보를 사용하지 못할 수도 있다.

캐싱은 요청된 콘텐츠를 서버로부터가 아니라 요청 노드와 가까이 위치한 노드로부터 제공함으로써 서비스 지연 및 서버 부하를 감소시키는 기법이다[12]-[16]. 통신 인프라와 노드들로 구성되는 네트워크는 노드와 연결되는 통신 인프라 지점에 캐싱 장치를 설치함으로써 이러한 특성을 구현할 수 있다. 그러나 지속해서 이동하는 노드들로만 구성되는 모바일 애드-혹 네트워크는 통신 인프라와 노드 사이의 경계 지점을 구분할 수 없다. 특정 노드를 지정하여 캐싱을 수행하면 캐싱 노드와 요청 노드 사이의 거리가 가변적이기 때문에 효율성을 기대하기 어렵다. 노드는 서버와 비교하여 저장 용량이나 처리 능력이 상대적으로 낮으므로 멀티미디어 콘텐츠를 수용하기에는 적합하지 않다. 네트워크를 클러스터와 같은 일정 영역으로 구분하여 영역마다 캐싱 노드를 지정하면 서비스 지연을 감소시킬 수 있다. 각 영역에서 지정된 캐싱 노드는 자신에게 집중되는 서비스 요청으로 인하여 가용 시간이 급격히 감소한다. 적극적 캐싱 기법은 경로 노드와 이웃 노드들 모두 자신이 수신하는 모든 데이터를 저장한다[17]. 멀티미디어 데이터는 시간 지역성(Temporal locality)보다는 공간 지역성(Spatial locality) 특성이 강하다. 노드의 이동성은 캐시 내의 공간 지역성 특성을 저하할 수 있다. 또한 적극적 기법 사용으로 인한 캐시 내의 갱신 빈도가 증가하여 부하가 증가하게 된다. 그러므로 멀티미디어 콘텐츠가 노드들 사이에 효율적으로 분산 저장되어 제공되게 함으로써 노드의 부하를 감소시키고 가용성을 증가시키는 기법이 필요하다.

III. 부분적 캐싱 기법

모바일 애드-혹 무선 네트워크에서 요청 콘텐츠의 전송 경로 수를 줄임으로써 노드 에너지의 가용성을 증가시키는 캐싱 기법을 제안한다. 이러한 네트워크는 노드들로만 구성된 네트워크이기 때문에 전송을 담당하는 별도의 통신 장비가 존재하지 않는다. 노드는 자신의 전송 안의 범위에 위치하는 다른 노드들과의 링크를 설정한다. 네트워크에 존재하는 모든 노드는 이러한 과정을 수행하며 인접 노드들과 링크 정보를 교환함으로써 다른 노드와의 멀티 홉 전송 경로를 설정할 수 있다. 요청된 콘텐츠는 서버로부터 전송 경로를 구성하는 노드들에서 순차적으로 저장 후 전송 방식을 사용하여 요청 노드로 전달된다.

서버는 N개의 콘텐츠를 제공하며, 콘텐츠 $n \in \{1, 2, \dots, N\}$ 의 재생 시간을 $T(n)$, 콘텐츠 서버를 S, 요청 노드를 R, 전송 경로 수를 H, 콘텐츠 C(n)의 전송 경로를 구성하는 노드를 $Ln(i), i = [1, 2, \dots, H]$,라고 하자. 시간 t에서 서버 S와 노드 R 사이의 전송 경로가 $\langle S - Ln(1) - Ln(2) - \dots - Ln(H-1) - R \rangle$ 로 설정되었다고 가정하자. 경로 노드 $Ln(i)$ 은 부모 노드인 $Ln(i-1)$ 으로부터 전송된 콘텐츠 C(n)를 자손 노드인 R로 보내기 위해 자식 노드 $Ln(i+1)$ 로 포워딩한다. 그러므로 서버로부터 전송된 콘텐츠 C(n)는 노드 R까지 H 번 반복적으로 전송된다. 경로 노드 $Ln(i)$ 는 요청 노드는 아니지만 전송되는 콘텐츠 C(n)를 수신하기 때문에 차후에 발생하는 요청을 위해 이를 저장한다. 노드의 지속적인 이동으로 인하여 시간 t에서의 전송 경로와 시간 $t + \Delta t$ 에서의 전송 경로는 다를 수 있다. 따라서 경로 노드 $Ln(i)$ 에서 저장되는 콘텐츠 C(n)는 전송 경로를 구성하는 기간에 해당하는 콘텐츠 n의 일부분만을 저장하게 되며 $C(n) \leq T(n)$ 이다. 이것은 지정된 노드에서만 캐싱이 수행되도록 하는 방식이 아니기 때문에 노드 이동으로 인한 캐싱 성능 저하 및 노드 부하를 감소시킬 수 있다.

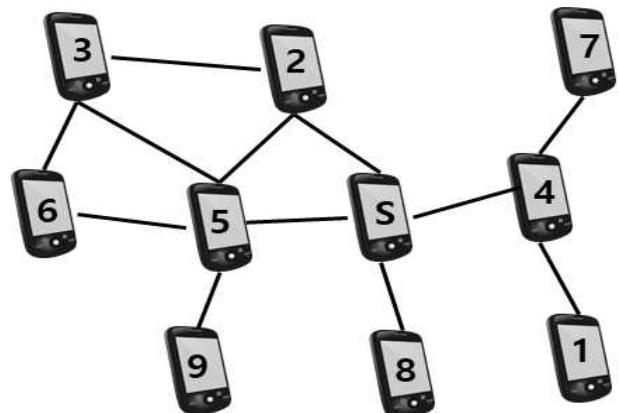


그림 1. 시간 t0에서 네트워크 토폴로지의 예
Fig. 1. An example of network topology at time t0

그림 1은 시간 t0에서의 네트워크 위상의 예를 보여주고 있다. S는 콘텐츠 서버를 나타내며 실선은 각 노드 사이의 링크를 보여준다. 시간 t0에서 노드 6이 콘텐츠 n에 대하여 최초로 요청을 한다고 가정하자. 따라서 콘텐츠 n은 이전에 전송되지 않았으며 다른 어떤 노드도 이를 저장하고 있지 않다. 요청 패킷은 경로를 구성하는 부모 노드 5를 거쳐 서버 노드에 전달된다. 서버 S는 요청 콘텐츠 n을 노드 6으로 보내기 위해 자식 노드 5로 전송한다. 노드 5는 요청 노드는 아니지만, 경로를 구성하는 노드이기 때문에 수신된 콘텐츠를 저장 후 자식 노드 6으로 전송한다. 이 과정에서 콘텐츠 n은 요청 노드와 전송 경로를 구성하는 노드들에 중복적으로 저장된다. 시간 $t_0 + \Delta t$ 에서의 네트워크 위상이 그림 1과 같으며 노드 9가 노드 6이 시간 t0에서 요청한 동일 콘텐츠 n을 요청한다고 가정하자. 노드 9의 요청은 부모 노드 5가 수신한다. 노드 5는 시간 t0에서 콘텐츠 n을 저장하였기 때문에 이 요청 패킷

을 서버 노드 S로 전송하지 않고 자신이 저장하고 있는 콘텐츠를 노드 9로 전송한다. 이 경우 콘텐츠 n의 전송은 {S, 5, 9}가 아니라 {5, 9} 사이에서 발생하기 때문에 전송을 위한 경로 수 및 대역폭이 감소하게 된다.

본 연구에서 전송되는 콘텐츠는 경로 노드와 요청 노드에 서만 캐싱 된다. 경로 노드가 될 확률을 PP, 요청 노드가 될 확률을 PR이라고 하자. 서버와 요청 노드 사이의 전송 경로 수가 H라면 전송되는 콘텐츠 n이 노드에 캐싱 될 확률 PC(n)은 식 (1)과 같다.

$$P_c(n) = \sum_{i=1}^{H-1} P_P + P_R \tag{1}$$

시간 t0에서 콘텐츠 n에 대한 노드 6의 요청이 최초로 발생한 것이 아니라고 가정하자. 이 경우 콘텐츠 n이 네트워크에 존재하는 다른 노드에 저장되어 있을 확률이 증가한다. 콘텐츠 n은 T(n)의 재생 시간을 가진다. 본 논문에서 콘텐츠는 M개의 프레임으로 구성되어 있다고 가정하며 각 노드는 프레임 단위로 캐싱을 수행한다. 콘텐츠 n의 m번째 프레임, nm이 노드에 캐싱 될 확률 PC(nm)은 식 (2)와 같다.

$$P_c(n_m) = \sum_{i=1}^{H-1} P_P(n_m) + P_R(n_m) \tag{2}$$

노드의 이동성은 서버와의 경로를 단절시키거나 캐싱 노드와의 거리를 가변적으로 만든다. 이것은 멀티미디어 콘텐츠의 실시간 서비스의 지연이나 단절을 초래할 수 있다. 이를 해결하기 위하여 수신 노드는 자신의 이웃 노드들로부터 현재 재생 중인 콘텐츠 프레임의 추가 프레임을 요청하고 수신한다. 추가로 요청되는 프레임의 크기는 윈도우 w로 정의된다. 그러므로 현재 재생 중인 프레임이 콘텐츠의 m번째 프레임이라면 이웃 노드들로부터 추가로 m+w 크기의 프레임을 요청하고 수신한다. 이 과정은 오직 요청 노드와 직접적으로 연결된 이웃 노드에 대해서만 수행된다. 노드가 시간 t에서 콘텐츠 n의 m번째 프레임이 캐싱 되어 있을 확률을 PC, t(nm), 삭제될 확률을 Pd, t(nm)이라고 하자. 윈도우 기법을 사용할 때 요청 노드가 콘텐츠 nm을 이미 캐싱하고 있을 확률 PL,t(nm)은 다음 수식 (3)과 같다.

$$P_{L,t}(n_m) = 1 - \left(\prod_{i=0}^t (1 - P_{C,i}(n_m)) + \sum_{j=1}^{t-1} P_{d,j}(n_m) \right) \tag{3}$$

노드가 요청한 콘텐츠는 요청 노드, 경로 노드, 이웃 노드에 의해서 제공될 수 있다. 요청 노드, 경로 노드, 이웃 노드에서 캐시 적중 확률, 이웃 노드의 수를 각각 PRequester, PPath, PSibling, Sm이라 하자. 콘텐츠가 서버가 아니라 노드에 의해 제공될 확률, 즉 캐시 적중 확률 Pcache-hit-ratio는 식 (4)와 같다.

$$\begin{aligned} P_{cache-hit-ratio} &= P_{Requester} + P_{Path} + P_{Sibling} \\ &= P_L(n_m) + \sum_{i=1}^{H-1} P_{P,i}(n_m) \\ &\quad + \left(1 - \left(\prod_{i=1}^{S_m} (1 - P_C(n_m)) \right) \right) \end{aligned} \tag{4}$$

캐시를 사용하지 않는 경우 콘텐츠 n은 서버로부터 요청 노드까지 홉 수 H만큼 반복 전송된다. 콘텐츠 nm이 전송될 때 캐시 적중 위치에 따라 전송 채널 수 TC(nm)는 요청 노드로부터 경로 노드까지의 거리가 h (h < H)일 때 식 (5)와 같다.

$$TC(n_m) = \begin{cases} 0 & , local\ cache\ hit \\ h & , path\ cache\ hit \\ h+1 & , adjacent\ cache\ hit \\ H & , cache\ miss \end{cases} \tag{5}$$

IV. 시뮬레이션 및 성능분석

본 절에서는 제안된 캐싱 기법의 성능을 분석한 실험 결과를 보여준다. 콘텐츠 서버는 1개만 존재한다고 가정한다. 멀티미디어 콘텐츠의 수 N은 100, 재생 시간 T(n)는 30초, 초당 재생률 SCBR은 1Mbps이다. 노드의 캐시 크기는 300SCBR 이다. 캐시 관리 알고리즘은 LRU(Least Recently Used)를 사용한다[18]. 150m의 전송 범위를 가지는 100개의 노드가 800m x 800m 크기의 네트워크에 위치한다. Zipf 분포를 사용하여 콘텐츠 인기도를 정의하며 Skew factor a는 0.85이다[15]. RWP(Random Way Point)를 사용하여 노드의 이동성을 정의하였으며 최대 속도는 8km/h, 평균속도는 4km/h이며 정지시간(pause time)은 5초이다 [6][7]. DSDV 라우팅 프로토콜을 사용하였으며 라우팅 수행 간격은 1초이다 [9][10]. 시뮬레이션이 수행된 시간은 20,000초이다.

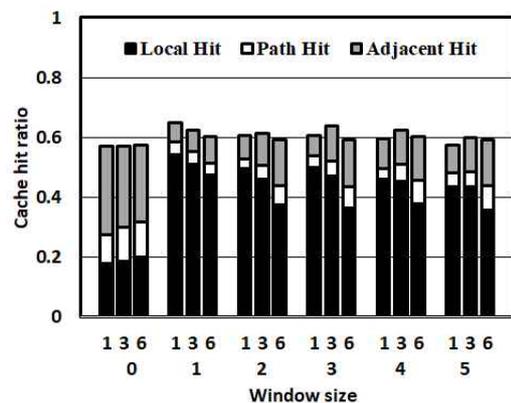


그림 2. 적극적 캐싱에서 윈도우 크기와 요청률에 따른 캐시 적중률

Fig. 2. Cache Hit Rate depending on Window Size and Request Rate in Aggressive Caching

초기 상태에서 노드들은 네트워크의 임의의 좌표에 위치한다. 그림 2는 분당 요청률 $\lambda \in \{1, 3, 6\}$ 일 때 요청 노드, 경로 노드, 인접 노드에서 적극적 캐싱을 수행하며, 윈도우 크기 $w = [0, 5]$ 일 때 캐시 적중률을 보여주고 있다. Local hit, Path hit, Adjacent hit는 각각 요청 노드, 경로 노드, 이웃 노드에서의 캐시 적중률이다. 시뮬레이션 결과는 윈도우 크기와 전체 캐싱 적중률과는 상관관계가 적음을 보여준다. 그러나 윈도우 기법을 사용하는 경우 local hit은 증가함을 보여준다. 제안된 기법은 블랙홀 현상과 전송 지연 등의 영향을 감소시키기 위하여 콘텐츠의 현재 재생 프레임보다 앞선 프레임들을 이웃 노드들로부터 윈도우 크기만큼 미리 요청하여 저장한다. 이로 인하여 결과적으로 local hit이 상대적으로 증가한다. 인기도에 기반한 콘텐츠 요청은 노드의 적은 캐시 용량으로 인한 캐시 미스의 영향을 줄인다. 같은 윈도우 크기에서 요청률이 증가하면 local hit과 전체 캐시 적중률이 감소한다. 이것은 요청률이 증가할수록 캐시에 저장된 프레임들의 갱신 빈도가 증가하기 때문이다. 그림 3은 경로 노드와 요청 노드만이 캐싱을 수행할 때 윈도우 크기와 요청률에 따른 캐시 적중률을 보여준다. 제안된 기법은 적극적 캐싱과 비교하여 전체 캐시 적중률에서 우수한 성능을 보인다.

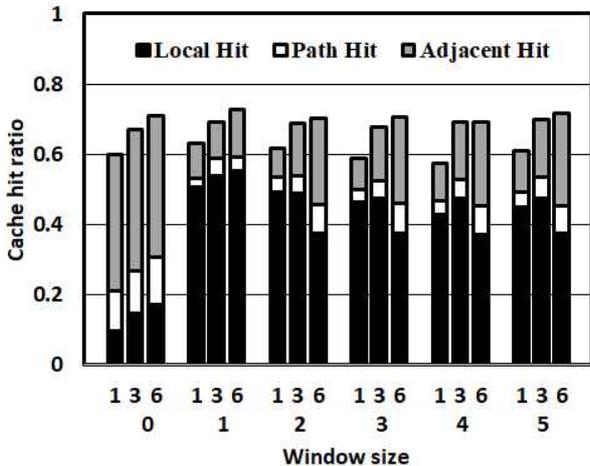


그림 3. 부분적 캐싱의 윈도우 크기와 요청률에 따른 캐시 적중률
Fig. 3. Cache Hit Rate depending on Window Size and Request Rate in Partial Caching

그림 2와 그림 3의 결과로부터 윈도우 크기가 클수록 캐시 적중률이 감소함을 알 수 있다. 이것은 요청률의 증가할수록 캐시 갱신 빈도 또한 증가하기 때문이다. 그러므로 전체 캐시 적중률이나 local hit 측면에서 보았을 때 윈도우의 크기가 작을수록 우수한 성능을 나타냄을 확인할 수 있다.

그림 4는 λ 가 6일 때 매 시각 평균 전송 채널 수 TC를 보여준다. NC(λ)는 caching을 사용하지 않을 때, A(λ)는 적극적 캐싱, P(λ)은 제안된 캐싱 기법을 사용할 때의 전송 채널 수이다. 시뮬레이션 결과로부터 전송 채널 수가 P(6)일 때 약

30%, A(6)는 약 38% 정도의 전송 채널 수가 필요함을 알 수 있다. T(n)의 크기를 가지는 멀티미디어 콘텐츠는 항상 첫 프레임부터 재생이 되어야 한다. LRU는 캐시에 최신 정보를 기억한다. 그러므로 캐시 갱신 빈도가 증가하면 요청된 콘텐츠의 첫 번째 프레임이 아니라 최근에 전송된 프레임이 저장되어 있을 확률이 증가하기 때문에 캐시 미스 확률이 증가한다. 윈도우 기법은 이러한 캐시 미스 영향을 감소시키는 역할을 한다.

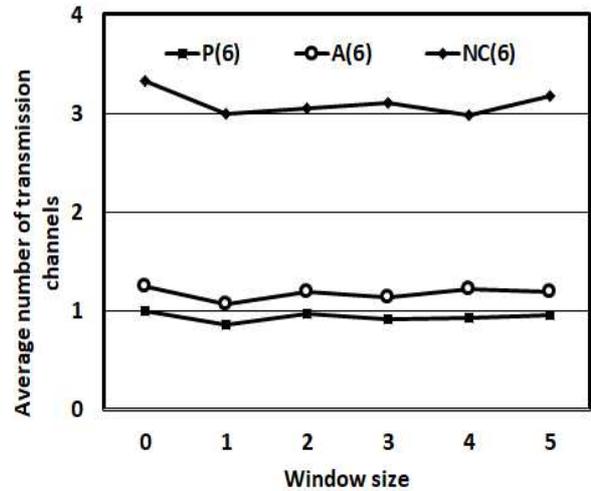


그림 4. 전송 채널의 평균 개수
Fig. 4. Average Number of Transmission Channels

V. 결론

본 연구에서는 모바일 애드-혹 무선 네트워크에서 멀티미디어 콘텐츠의 스트리밍을 위한 협업적 부분적 캐싱 기법을 제안한다. 이 기법에서 전송되는 콘텐츠는 경로 노드와 요청 노드에서 캐싱하여 사용된다. 제안된 기법은 콘텐츠 요청률, 윈도우 크기의 관점에서 캐시 적중률, 전송 채널 수와 비율을 평가하였다. 요청률이 증가할수록 캐시 갱신 빈도는 증가한다. 이것은 캐시 적중률 및 전송 채널의 사용 효율을 저하한다. 이를 개선하기 위하여 요청 노드는 차후 재생에 필요한 콘텐츠 프레임을 설정된 프레임 크기만큼 이웃 노드들에 요청하여 저장한다. 제안된 기법의 성능을 평가하기 위하여 수신되는 데이터를 무조건 저장하는 적극적 캐싱 기법과 비교 분석하였다. 시뮬레이션 결과로부터 제안된 윈도우 기법은 높은 캐시 적중률을 보이며 특히 로컬 캐시 적중률이 증가함을 보인다. 이로 인하여 전송에 필요한 채널 숫자를 감소시킬 수 있다. 그러므로 제안된 캐싱 기법은 노드 에너지 및 전송 대역폭의 효율적 사용을 가능하게 하는 대안적 기법이 될 수 있을 것이다. 제안하는 기법의 캐시 미스 발생 원인, 갱신 오버헤드, 캐시 크기에 따른 성능 평가 등에 관한 연구를 진행할 예정이다.

참고문헌

- [1] M. Agiwal, A. Roy, A. and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, Vol. 18, No. 3, pp. 1617-1655, Feb, 2016.
<https://doi.org/10.1109/COMST.2016.2532458>
- [2] M. M. Hamdi, L. Audah, S. A. Rashid, A. H. Mohammed, S. Alani, and A. S. Mustafa, "A review of applications, characteristics and challenges in vehicular ad hoc networks (VANETs)," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), IEEE*, pp. 1-7, Jun, 2020.
<https://doi.org/10.1109/HORA49412.2020.9152928>
- [3] M. Conti and S. Giordano, "Mobile ad hoc networking: milestones, challenges, and new research directions," *IEEE Communications Magazine*, Vol. 52, No. 1, pp. 85-96, Jan, 2014.
<https://doi.org/10.1109/MCOM.2014.6710069>
- [4] J. Yang, S. He, Y. Lin, and Z. Lv, "Multimedia cloud transmission and storage system based on internet of things," *Multimedia Tools and Applications*, Vol. 76, No. 17, pp. 17735-17750, Oct, 2015.
<https://doi.org/10.1007/s11042-015-2967-9>
- [5] D. Wu, B. Liu, Q. Yang, and R. Wang, "Social-aware cooperative caching mechanism in mobile social networks," *Journal of Network and Computer Applications*, Vol. 149, pp. 1-12, Jan, 2020.
<https://doi.org/10.1016/j.jnca.2019.102457>
- [6] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications and Mobile Computing (WCNC): Special issue on Mobile Ad Hoc Networking Research, Trends and Applications*, Vol. 2, No. 5, pp. 483-502, Sep, 2002. <https://doi.org/10.1002/wcm.72>
- [7] P. Nayak, and P. Sinha, "Analysis of random way point and random walk mobility model for reactive routing protocols for MANET using NetSim simulator," in *2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), IEEE*, pp. 427-432, Dec, 2015.
<https://doi.org/10.1109/AIMS.2015.87>
- [8] A. S. Mustafa, M. M. Al-Heeti, M. M. Hamdi, and A. M. Shantaf, "Performance analyzing the effect of network size on routing protocols in MANETs," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), IEEE*, pp. 1-5, Jun, 2020.
<https://doi.org/10.1109/HORA49412.2020.9152838>
- [9] G. Adam, C. Bouras, A. Gkamas, V. Kapoulas, G. Kioumourtzis, and N. Tavoularis, "Performance evaluation of routing protocols for multimedia transmission over mobile ad hoc networks," in *Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP, IEEE*, pp. 1-6, Oct, 2011.
<https://doi.org/10.1109/WMNC.2011.6097250>
- [10] S. Mitra, R. Islam, K. Mukherjee, A. Das, and S. Nandi, "A Modified Algorithmic Approach of DSDV Routing Protocol for Wireless Ad Hoc Network," *IOSR Journal of Computer Engineering (IOSR-JCE)*, Vol. 14, No. 4, pp. 49-54, Sep, 2013.
<https://doi.org/10.9790/0661-1444954>
- [11] N. Chaubey, A. Aggarwal, S. Gandhi, and K. A. Jani, "Performance analysis of TSDRP and AODV routing protocol under black hole attacks in manets by varying network size," in *Advanced Computing and Communication Technologies (ACCT), IEEE*, pp. 320-324, Feb, 2015.
<https://doi.org/10.1109/ACCT.2015.62>
- [12] P. Srivastava and R. Kumar, "An efficient proxy adaptive gateway discovery algorithm based on quality of service parameters," *International Journal of Communication Networks and Distributed Systems*, Vol. 16, No. 3, pp. 261-280, May, 2016.
<https://doi.org/10.1504/ijcnds.2016.076653>
- [13] G. Kioumourtzis, A. Gkamas, and C. Bouras, "Mobile Ad hoc networks (MANETs) for multimedia transmission," in *Encyclopedia of Information Science and Technology, Third edition*, pp. 6239-6248, 2015.
<https://doi.org/10.4018/978-1-4666-5888-2.ch614>
- [14] D. Prerna, R. Tekchandani, and N. Kumar, "Device-to-device content caching techniques in 5G: A taxonomy, solutions, and challenges. Computer Communications," Vol. 153, pp. 48-84, Mar, 2020.
<https://doi.org/10.1016/j.comcom.2020.01.057>
- [15] C. G. Krishnan, Y. H. Robinson, E. G. Julie, A. M. Bamini, R. Kumar, P. H. Thong, and L. H. Son, "Hybrid cache management in ad hoc networks," *Wireless Personal Communications*, Vol. 118, No. 4, pp. 2843-2865, Mar, 2021.
<https://doi.org/10.1007/s11277-021-08158-z>
- [16] R. A. Rehman and B. S. Kim, "LOMCF: Forwarding and Caching in Named Data Networking Based MANETs," *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 10, pp. 9350-9364, May, 2017. <https://doi.org/10.1109/TVT.2017.2700335>

- [17] B. Kim, and I. Kim, "Multimedia caching strategy based on popularity in Mobile Ad-Hoc Network," *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 10, No. 4, pp. 173-182, Apr, 2015. <https://doi.org/10.14257/ijmue.2015.10.4.17>
- [18] N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," *Performance Evaluation*, Vol. 63, No. 7, pp. 609-634, Jul, 2006. <https://doi.org/10.1016/j.peva.2005.05.003>



박경모(Kyeongmo Park)

1983년 : 서울대학교 대학원 전산학과 석사
1990년 : 미국 New Jersey Institute of Technology CIS학과 컴퓨터학 석사
1994년 : 미국 George Mason University IT&E공학부 공학박사

1983년~1985년: 삼성전자(주) 컴퓨터연구개발(R&D) 연구원
1990년~1995년: 조지메이슨대학교(GMU) 컴퓨터학과 강의/연구조교, 연구원
1995년~1996년: 한국전자통신연구원(ETRI) 컴퓨터시스템 연구원
1996년~현 재: 가톨릭대학교 컴퓨터정보공학부 교수
※관심분야: 컴퓨터시스템, 운영체제, 클라우드 컴퓨팅, 시뮬레이션, 성능평가, IT융합



김백현(Back-Hyun Kim)

2001년 : 인천대학교 정보통신공학 공학석사
2006년 : 인천대학교 정보통신공학 공학박사

1993년~1997년: 삼성전자(주) 반도체 연구원
2013년~2016년: 인천대학교 기초교육원 객원교수
2019년~현 재: 인천대학교 기초교육원 객원교수
※관심분야: 인공지능, 사물인터넷, 모바일 네트워크, 클라우드 컴퓨팅, 분산 컴퓨팅