

기회적 포그 컴퓨팅 환경을 고려한 IoT 테스트의 지연된 오프로딩 제공 방안

경연웅*

한신대학교 컴퓨터공학부 교수

Delayed offloading scheme for IoT tasks considering opportunistic fog computing environment

Yeunwoong Kyung*

Professor, Division of Computer Engineering, Hanshin University

요약 다양한 IoT(Internet of Things) 서비스들이 등장하면서 IoT 기기의 테스트를 오프로딩 시키는 연구가 진행되었다. 기존에는 클라우드 컴퓨팅을 통한 오프로딩이 고려되었지만 서비스 응답 지연 및 코어 네트워크의 부하 등의 이슈로 인해 IoT 기기 근처에서 오프로딩을 지원하는 포그 컴퓨팅 개념이 도입되었다. 하지만 포그 컴퓨팅 환경에서도 서비스 대상 IoT 기기가 증가하게 되면 클라우드 환경과 마찬가지로 부하 집중 문제로 인해 서비스 응답 지연이 발생할 수 있다. 이를 해결하기 위하여 자동차, 드론 등 IoT 기기 근처에 존재하는 컴퓨팅 가능 노드들을 통해 오프로딩을 수행하는 개념인 기회적 포그 컴퓨팅이 등장하였다. 기존의 포그 및 기회적 포그 컴퓨팅 노드들을 활용한 오프로딩 연구들은 서비스의 요청이 있을 때 가능한 노드를 통해 오프로딩을 수행한다. 기존의 연구 방법대로 오프로딩을 수행한다면 기회적 포그 컴퓨팅 노드가 가용할 때에 발생된 요청들만 해당 노드들로 오프로딩이 가능하다. 하지만 서비스의 응답 지연 요구사항만 만족시킨다면 즉시적으로 요청을 처리할 필요가 없고 최대한 많은 테스트를 기회적 포그 컴퓨팅 노드로 오프로딩 시키는 것이 부하 분산에 용이하다. 그러므로 본 논문에서는 오프로딩 타이머를 기반으로 서비스 응답 지연 요구사항을 만족시키면서 최대한 기회적 포그 컴퓨팅 노드들을 통해 오프로딩 시킬 수 있는 지연된 오프로딩 방법을 제안하고자 한다.

주제어 : 포그 컴퓨팅, 기회적 포그 컴퓨팅, 지연된 오프로딩

Abstract According to the various IoT(Internet of Things) services, there have been lots of task offloading researches for IoT devices. Since there are service response delay and core network load issues in conventional cloud computing based offloadings, fog computing based offloading has been focused whose location is close to the IoT devices. However, even in the fog computing architecture, the load can be concentrated on the fog computing node when the number of requests increase. To solve this problem, the opportunistic fog computing concept which offloads task to available computing resources such as cars and drones is introduced. In previous fog and opportunistic fog node researches, the offloading is performed immediately whenever the service request occurs. This means that the service requests can be offloaded to the opportunistic fog nodes only while they are available. However, if the service response delay requirement is satisfied, there is no need to offload the request immediately. In addition, the load can be distributed by making the best use of the opportunistic fog nodes. Therefore, this paper proposes a delayed offloading scheme to satisfy the response delay requirements and offload the request to the opportunistic fog nodes as efficiently as possible.

Key Words : fog computing, opportunistic fog computing, delayed offloading

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2020R1G1A1100493).

*교신저자 : 경연웅(ywkyung@hs.ac.kr)

접수일 2020년 11월 14일 수정일 2020년 12월 14일 심사완료일 2020년 12월 18일

1. 서론

4차 산업 혁명의 등장과 함께 우리 사회는 IoT(Internet of Things) 서비스의 급격한 발전에 영향을 받고 있다. 스마트 헬스, 스마트 팩토리, 스마트 빌딩 등 IoT 서비스는 점차적으로 그 영역을 확대해가고 있다 [1-3]. 이렇게 IoT 기반 서비스가 다양한 분야에서 활용되면서 방대한 양의 데이터를 빨리 처리할 수 있는 능력이 요구되었는데 IoT 기기들은 컴퓨팅 및 배터리 자원이 제약적이기 때문에 이러한 요구사항을 만족시킬 수 없다. 이러한 한계를 해결하기 위해 클라우드 컴퓨팅을 통해 IoT 기기의 서비스 요청을 처리하는 오프로딩 방법들이 제공되었다. 하지만 클라우드 컴퓨팅을 통한 오프로딩은 서비스 응답 지연 및 코어 네트워크 부하 증가 등의 이슈가 발생하였고 이러한 문제들을 해결하기 위해 IoT 기기 근처에서 오프로딩을 지원하는 포그 컴퓨팅 개념이 등장하였다. 즉 포그 컴퓨팅 기반의 오프로딩을 통해 IoT 서비스의 지연 및 고성능 지원이 가능해졌다[4].

주로 포그 컴퓨팅 노드는 IoT 기기가 접속한 액세스 장비(e.g., BS(Base Station), AP(Access Point))에 연결된 컴퓨팅 자원을 의미하는데 이러한 포그 컴퓨팅에서도 IoT 기기들의 요청이 증가함에 따라 부하가 집중되는 문제가 발생할 수 있다. 이러한 문제를 해결하기 위하여 이웃한 포그 컴퓨팅 노드들과의 협력 [5,6] 및 포그 컴퓨팅 노드와 클라우드 컴퓨팅 노드의 협력 [7,8] 기반의 태스크 처리 방안들이 고려되었다. 하지만 해당 협력에 참여하는 포그 및 클라우드 컴퓨팅 노드들은 자신에게 할당된 태스크 요청들이 이미 존재하기 때문에 외부로부터 전달된 태스크 처리를 위한 협력에 한계가 존재한다.

이를 해결하기 위하여 자동차, 드론 등 IoT 기기 근처에 유동적으로 존재하는 컴퓨팅 가능 노드들을 통해 오프로딩을 수행하는 기회적 포그 컴퓨팅에 대한 개념이 등장하였다[9]. 이러한 기회적 포그 컴퓨팅 노드들은 이동성 등의 특성으로 인해 항상 가용한 컴퓨팅 자원은 아니지만 자동차, 드론 등 이동 노드들이 점차적으로 컴퓨팅 능력을 탑재함으로써 정적인 포그 컴퓨팅 노드의 부하를 분산시켜주는 방법으로서 많은 연구가 진행되고 있다[10-13].

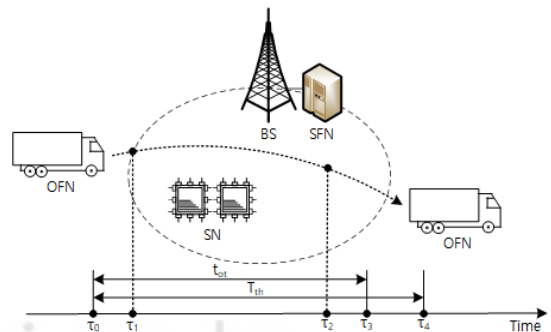
일반적으로 기존의 정적인 포그 컴퓨팅 노드들 (Static Fog Computing Nodes; SFN) 및 기회적 포그 컴퓨팅 노드들 (Opportunistic Fog Computing Nodes; OFN)을 이용한 오프로딩 연구들은 서비스의 요청이 발생하면 바로 해당 요청을 가용한 SFN 또는 OFN으로 오

프로딩을 시도한다. 즉 서비스의 요청이 발생하면 즉시적으로 가용한 자원으로 오프로딩 하는 방식이다. 하지만 이러한 방식은 다음의 두 가지를 고려하지 않았다. 먼저 IoT 서비스의 요청들은 다양한 지연 요구사항을 갖고 있기 때문에 모든 요청들에 대해서 반드시 즉시적으로 처리해야 할 필요는 없다[14]. 지연 요구사항을 만족시키면서 최대한 부하를 분배시켜서 자원 효율적으로 처리하는 것이 효과적일 수 있다. 그리고 OFN을 고려한 상황에서는 OFN으로 오프로딩 하는 것이 SFN의 부하를 분산시킬 수 있게 되는데, OFN은 항상 가용한 자원이 아니기 때문에 서비스 요청이 발생할 때 OFN이 불가한 상황이 존재한다. 즉, 일정 시간 뒤에 OFN이 가용함에도 불구하고 기존 연구들은 당장 가용한 SFN을 사용함으로써 SFN으로 부하가 집중될 수 있다.

그러므로 본 연구에서는 오프로딩 타이머를 도입하여 서비스 요청이 발생했을 때 바로 처리하지 않고 서비스의 지연 요구사항을 만족시키면서 타이머가 만료되기 전에 OFN이 가용해지면 OFN을 활용할 수 있는 방법을 제안하고자 한다.

2. 제안하는 지연된 오프로딩 방법

본 연구에서 제안하는 방법의 절차는 [Fig. 1]과 같다. 스마트 헬스, 스마트 팩토리 등에서 사용되는 IoT 기기인 센서 노드(SN)가 존재하고, SN에서는 IoT 서비스의 태스크가 τ_0 에서 요청된다. 기존의 오프로딩 방식들은 τ_0 에서 바로 오프로딩 가용자원을 확인하고 오프로딩을 수행하는데, 이 때에 OFN이 불가하기 때문에 BS에 연결된 SFN에게 오프로딩을 수행한다. 하지만 제안하는 방법에서는 해당 서비스의 오프로딩 타이머(t_0)를 시작하고



[Fig. 1] Offloading timer procedure

타이머 내에 OFN이 나타나면(τ_1) 해당 OFN으로 테스크의 오프로딩을 수행한다. 타이머 만료 시간인 τ_3 전까지 OFN에서 해당 테스크를 처리하고 응답을 주면(τ_2) 해당 테스크의 OFN을 통한 처리가 완료되는 것이다. 만약 타이머가 만료될 때까지 OFN이 나타나지 않거나, OFN의 처리 응답이 수신되지 않으면 SN은 BS의 SFN에게 오프로딩을 수행하고 이 오프로딩은 반드시 서비스의 지연 요구사항인 $\tau_0 + T_{th}$, 즉 τ_4 까지 처리되어야 한다.

제안하는 지연된 오프로딩 방법을 수식적으로 정리하면 다음과 같다. 먼저 오프로딩 타이머는 서비스의 지연 요구사항을 만족시키기 위해 수식(1)을 따른다.

$$t_{ot} \leq T_{th} - T_{SFN} \quad (1)$$

수식(1)에서 T_{SFN} 은 SN에서 SFN으로 테스크 오프로딩 요청을 송신하고 응답을 수신하는 시간과 SFN에서의 테스크 컴퓨팅 시간을 포함한다. 즉 SN은 오프로딩 타이머의 시간이 만료되면 테스크를 바로 항상 가용한 자원인 SFN에게 오프로딩하고, 이에 대한 응답이 지연 요구사항 전에 수신되어야 서비스의 품질 저하가 발생하지 않는 것이다. 이러한 서비스 품질 저하를 outage 확률로 표현하면 수식(2)와 같다.

$$P_{out} = P[t_{ot} + T_{SFN} > T_{th}] \quad (2)$$

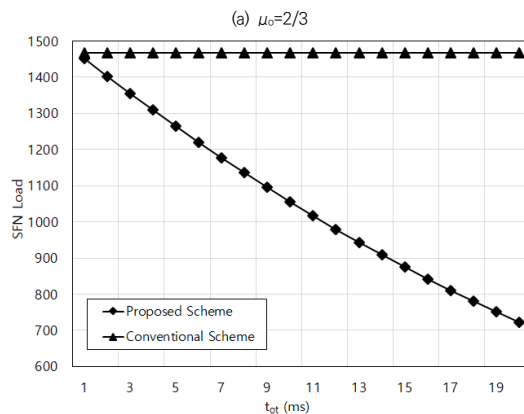
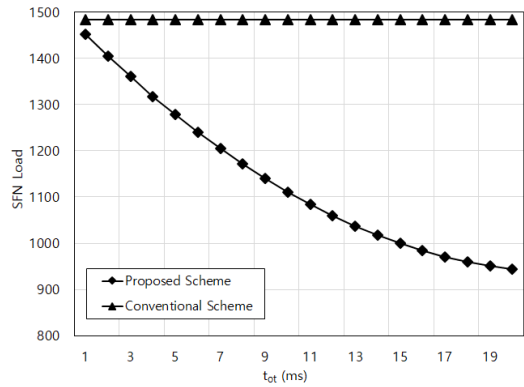
즉 제안하는 방법은 P_{out} 을 일정 임계치(Th) 이하로 디자인 할 수 있는 오프로딩 타이머를 적용해야 하고 이는 수식(3)으로 표현된다.

$$\operatorname{argmax}_{t_{ot}} (P_{out} \leq Th) \quad (3)$$

수식(2)를 만족시킬 수 있는 t_{ot} 가 결정되면 OFN의 가용 시간을 계산할 수 있다. 본 논문에서는 OFN의 대표적인 예로 이동하는 차량을 가정하였고 OFN이 SN의 가용한 영역 내에 존재할 시간의 분포는 평균 $1/\mu_0$ 를 갖는 exponential 분포를 나타낸다[15,16]. 즉 t_{ot} 동안 가용한 OFN의 시간은 평균 t_{ot}/μ_0 을 갖는 exponential 분포를 나타내고, 이 시간동안 SN은 OFN으로의 오프로딩이 가능하다.

3. 성능 분석

제안된 방법의 효과를 입증하기 위하여 테스크의 요청이 발생하면 해당 상황에서의 가용한 자원으로 오프로딩을 수행하는 기존 방법[10,11]과 제안하는 방법의 성능 분석 비교를 수행하였다. SN에서의 서비스 테스크의 요청 분포는 Poisson 분포를 따르며[17] 평균을 50으로



(a) $\mu_0=2/3$
(b) $\mu_0=1/3$
[Fig. 2] SFN load according to the offloading timer

설정하였고, 지연 요구사항은 30ms로 설정하였다. 지연 요구사항의 경우 서비스의 특성에 따라 가변적으로 설정할 수 있다. [Fig. 2]를 보면 오프로딩 타이머가 길어질수록 제안하는 방법에서 SFN의 부하가 줄어드는 것을 볼 수 있다. 즉 오프로딩 타이머가 길어지면 테스크 요청이 발생하더라도 바로 오프로딩을 시키지 않고 OFN이 가용해질 때 OFN에게 오프로딩 시키기 때문에 SFN의 부하가 감소하게 된다. 반면 기존 방법은 SFN의 부하가 일정한 것을 볼 수 있는데 이는 테스크의 요청이 발생했을 때, OFN이 가용할 확률은 일정하기 때문에 부하가 일정한 값이 감소된 상태를 유지하는 것이다. 또한 (a)와 (b)를 비교해보면 μ_0 가 커짐에 따라 SFN의 부하가 더 줄어드는 것을 확인할 수 있는데, 이는 OFN의 가용시간이 증가함에 따라 OFN으로의 오프로딩 로드 증가하기 때문이다.

4. 결론 및 향후 연구 방안

본 논문은 정적인 포그 컴퓨팅 자원과 기회적 포그 컴퓨팅 자원이 공존하는 아키텍처에서 테스크 처리요청이 발생했을 때 바로 오프로딩 시키지 않고 오프로딩 타이머를 기반으로 기회적 포그 컴퓨팅 자원이 가용할 때 최대한 해당 자원을 이용하는 방법을 제안하였다. 성능 분석을 통해 제안하는 방법의 오프로딩 타이머가 커질수록 정적인 포그 컴퓨팅 자원의 부하를 분산시킬 수 있음을 보였다. 향후에는 두 가지 방향으로 본 연구를 확장시키고자 한다. 먼저 제안하는 방법에서 서비스의 지연 요구 사항에 따라 outage 확률을 최소화시키는 최적의 오프로딩 타이머 값을 분석할 예정이고, 두 번째로 이동성을 가진 IoT 기기의 시나리오에서 지연된 오프로딩 기법을 적용하여 모델링을 진행하고자 한다.

REFERENCES

- [1] D.W.Lee, K.Cho, and S.H.Lee, "Analysis on Smart Factory in IoT Environment," *Journal of The Korea Internet of Things Society*, Vol.5, No.2, pp.1-5, 2019.
- [2] K.B.Jan.g, "A study on IoT platform for private electrical facilities management," *Journal of The Korea Internet of Things Society*, Vol.5, No.2, pp.103-110, 2019.
- [3] Y.W.Kyung and T.K.Kim, "Flow Handover Management Scheme based on QoS in SDN Considering IoT," *Journal of The Korea Internet of Things Society*, Vol.6, No.2, pp.45-50 2020
- [4] P.Mach and Z.Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys & Tutorials*, Vol.19, No.3, pp.1628-1656, 2017.
- [5] M.Mukherjee, S.Kumar, C.X.Mavromoustakis, G.Mastorakis, R.Matam, V.Kumar, and Q.Zhang, "Latency-driven Parallel Task Data Offloading in Fog Computing Networks for Industrial Applications," *IEEE Transactions on Industrial Informatics*, Vol.16, No.9, pp.6050-6058, 2020.
- [6] Y.Jiang and D.H.K.Tsang, "Delay-Aware Task Offloading in Shared Fog Networks," *IEEE Internet of Things Journal*, Vol.5, No.6, pp.4945-4956, 2018.
- [7] A.Yousefpour, G.Ishigaki, R.Gour, and J.P.Jue, "On Reducing IoT Service Delay via Fog Offloading," *IEEE Internet of Things Journal*, Vol.5, No.2, pp.998-1010, 2018.
- [8] J.Ren, G.Yu, Y.He, and G.Y.Li, "Collaborative Cloud and Edge Computing for Latency Minimization," *IEEE Transactions on Vehicular Technology*, Vol.68, No.5, pp.5031-5044, 2019.
- [9] N.Fernando, S.W.Loke, I.Avazpour, F.Chen, A.B.Abkenar, and A.Ibrahim, "Opportunistic Fog for IoT: Challenges and Opportunities," *IEEE Internet of Things Journal*, Vol.6, No.5, pp.8897-8910, 2019.
- [10] Y.Liu, S.Wang, Q.Zhao, S.Du, A.Zhou, X.Ma, and F.Yang, "Dependency-Aware Task Scheduling in Vehicular Edge Computing," *IEEE Internet of Things Journal*, Vol.7, No.6, pp.4961-4971, 2020.
- [11] Z.Ning, J.Juang, X.Wang, J.J.P.C.Rodrigues, and L.Guo, "Mobile Edge Computing-Enabled Internet of Vehicles: Toward Energy-Efficient Scheduling," *IEEE Network*, Vol.33, No.5, pp.198-205, 2019.
- [12] X.Wang, Z.Ning, and L.Wang, "Offloading in Internet of Vehicles: A Fog-enabled Real-time Traffic Management System," *IEEE Transactions on Industrial Informatics* Vol.14, No.10, pp.4568-4578, 2018.
- [13] Z.Ning, P.Dong, X.Wang, J.J.P.C.Rodrigues, and F.Xia, "Deep Reinforcement Learning for Vehicular Edge Computing: An Intelligent Offloading System," *ACM Transactions on Intelligent Systems and Technology*, Vol.10, No.6, pp.1-24, 2019.
- [14] M.Li, P.Si, and Y.Zhang, "Delay-Tolerant Data Traffic to Software-Defined Vehicular Networks with Mobile Edge Computing in Smart City," *IEEE Transactions on Vehicular Technology*, Vol.67, No.10, pp.9073-9086, 2018.
- [15] Y.Liu, W.Wang, Y.Ma, Z.Yang, and F.Yu, "Distributed Task Offloading in Heterogeneous Vehicular Crowd Sensing," *MDPI Sensors*, Vol.16, No.7, 2016.
- [16] J.Lee, G.Lee, and S.Pack, "Pseudonyms in IPv6 ITS Communications: Use of Pseudonyms, Performance Degradation, and Optimal Pseudonyms Change," *International Journal of Distributed Sensor Networks*, Vol.11, No.5, pp.1-7, 2015.
- [17] Q.Fan and N.Ansari, "Towards Workload Balancing in Fog Computing Empowered IoT," *IEEE Transactions on Network Service and Engineering*, Vol.7, No.1, pp.253-262, 2018.

경연웅(Yeun-Woong Kyung) [종신회원]



- 2011년 2월 : 고려대학교 전기전자전파공학부(공학사)
- 2016년 8월 : 고려대학교 전기전자전파공학부(공학박사)
- 2016년 9월 ~ 2020년 3월 : 삼성전자 무선사업부 책임연구원
- 2020년 3월 ~ 현재 : 한신대학교 컴퓨터공학부 교수

<관심분야>

사물인터넷(IoT), SDN, 5G/6G, 이동성, 모바일 서비스