

포그 컴퓨팅 네트워크에서 작업 하역에 관한 연구 동향

우 미 애*

Research Trends on Task Offloading in Fog Computing Networks

Miae Woo*

요 약

포그 컴퓨팅 개념은 IoT 장비가 다양한 분야에서 생성하는 많은 데이터를 효율적으로 처리하기 위하여 기존의 클라우드 컴퓨팅을 보완하기 위하여 도입되었다. 포그 컴퓨팅 네트워크에서 중요한 연구 과제 중 하나는 작업 하역이다. 작업 하역은 IoT 장비가 생성한 데이터를 지연이나 에너지 소모 최소화와 같은 사용자가 요구하는 여러 가지 조건을 만족시킬 수 있는 최적의 장소로 보내서 작업을 처리하게 하는 것이다. 본 논문에서는 작업 하역 문제에 대한 연구 동향을 포그 컴퓨팅 네트워크를 구성하는 포그 노드의 특성과 작업 하역 판단 주체에 따른 분류로 나누어 알아보고, 각 방안의 메시지 양을 최악의 경우에 대하여 분석하였다.

Key Words : Task offloading, Optimization, Fog networks, Fog computing, IoT

ABSTRACT

Nowadays, IoT devices produce large amount of data in diverse areas. To handle these data efficiently, the concept of fog computing is introduced to supplement cloud computing. One of important research topics in fog computing network is task offloading. Task offloading deals with selection of optimal place to transfer data produced at the IoT devices as a form of task and to process the task to satisfy various user requirements such as minimizing delay and energy consumption. In this paper, we survey up-to-date research trends on task offloading in fog computing networks by classifying fog computing networks based on the characteristic of fog nodes and who is decision maker for the task offloading. Also, worst case analysis on message load for each study is provided.

I. 서 론

전 세계적으로 IoT 장비가 빠른 속도로 증가하고 있어 2022년에는 500억 개 정도가 될 것으로 추정한다^[1]. IoT 장비 숫자가 증가하고 이들을 이용한 응용 프로그램이 증가함에 따라 다양한 데이터가 많이 생성되고, 생성된 데이터의 종류에 따라 사용자의 요구 조건도 다양해져서 새로운 컴퓨팅 모델 필요성이 대두되었다. 주된 사용자 요구조건에는 지연의 최소화, 네트워크 대역폭 절약, 보안, 안정적인 운영, 데이터의

안전한 보관, 데이터 처리의 최적화, 에너지 소비의 최소화 등이 있다^[2]. 이러한 요구조건을 충족시키기 위하여 2012년에 시스코가 포그 컴퓨팅 개념을 도입하였다. 포그 컴퓨팅은 클라우드 컴퓨팅을 확장하여 네트워크의 가장자리에서 시간에 민감한 데이터를 처리하는 서비스를 제공함으로써, 클라우드 컴퓨팅을 대체하기 보다는 보완하는 개념이다.

포그 컴퓨팅 구조는 일반적으로 그림 1과 같이 IoT/단말 장치 계층, 포그 계층, 클라우드 계층의 3단계 구조로 구성된다. 1단계인 IoT/단말 장비 계층은

* First Author : Sejong University, Department of Electrical Engineering, mawoo@sejong.ac.kr, 정교수, 종신회원,
논문번호 : 202007-170-B-RN, Received July 28, 2020; Revised September 2, 2020; Accepted September 3, 2020

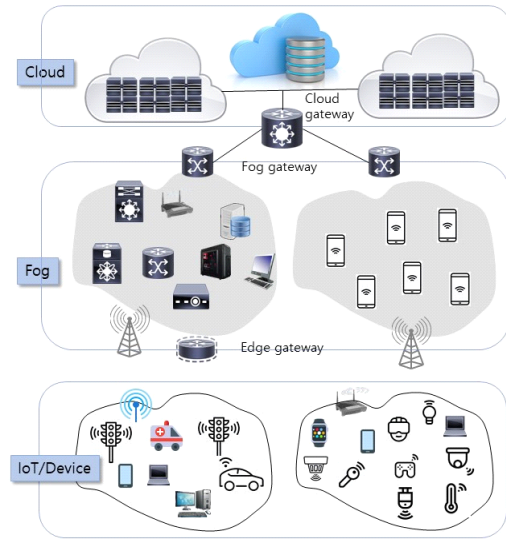


그림 1. 포그 컴퓨팅 네트워크 구조
Fig. 1. Fog computing network architecture

IoT 기능이 있는 장비로 구성된다. IoT 장비는 데이터를 수집하여 포그 계층에 전달하는 역할을 담당한다. 2단계인 포그 계층은 연산 자원, 통신 모듈, 데이터 저장 장치 등을 탑재한 포그 노드(fog node: FN)들로 구성된다. 포그 노드는 임시로 데이터를 저장하고 지역 데이터를 분석하는 등 IoT 장비와 클라우드 계층에 있는 데이터 센터의 부하를 덜어주는 역할을 한다. 포그 컴퓨팅에서는 시간에 민감한 데이터는 포그 노드에 저장, 처리, 분석하고, 영구적인 저장이 필요하거나 글로벌 분석이 필요한 데이터는 포그 노드가 클라우드에게 주기적으로 데이터 요약물을 전송한다. 3단계인 클라우드 계층은 데이터 센터, 클라우드 저장장치, 전통적인 클라우드 서버로 구성된다. 클라우드 계층은 충분한 저장 공간과 컴퓨팅 자원을 가지고 있어서, 연산과 저장 플랫폼을 통합하여 글로벌 관점에서 다양한 IoT 응용을 제공한다.

IoT 응용은 다양한 서비스 품질을 요구하는 데, 요구조건의 중요도는 응용 프로그램마다 다르다. 모바일 장비에서 동작하는 일반적인 IoT 응용은 에너지 소비에 대한 조건이 까다롭고, 온라인 게임은 낮은 지연과 높은 에너지 효율을 요구한다. 지능적 생산시설, 촉각 인터넷, 차량 네트워크의 경우에는 에너지 소비보다는 평균 지연과 지연의 변화 등 지연에 대한 성능을 보장하는 것이 좀 더 중요한 요소다. 서비스 지연과 에너지 소비 간에 균형을 잡는 것은 사용자의 체감 품질에 직접 영향을 미치는 중요한 사안이나 쉽게 해결할 수

없는 문제다³⁾.

IoT 응용의 요구조건을 만족시키기 위해서는 충분한 연산 자원이 필요하나, IoT 장비는 일반적으로 제한된 연산 능력만을 갖고 있기 때문에 연산 작업을 포그 노드나 원격에 있는 클라우드 서버로 보내서 처리하게 하여 IoT 장비의 연산 능력 부족을 보완할 수 있다^{4,5)}. 포그 컴퓨팅 네트워크에서 지연에 민감한 데이터를 분석하기 위해서 해당 데이터 분석 작업을 어디서 할지 결정해야 한다. 즉, 포그 컴퓨팅 네트워크에 존재하는 컴퓨팅 능력을 갖고 있는 여러 노드들 중 어느 노드를 선택할 것이냐는 문제다. 일반적으로 작업을 수행할 노드를 선택할 때, 노드의 작업 수행 능력뿐만 아니라 선택하는 노드에서 해당 작업을 얼마나 빨리 처리할 수 있는지도 고려 대상이다. 따라서 작업을 하역할 최적의 노드 선택을 위한 방법론이 필요하다⁶⁾. 본 논문에서는 포그 컴퓨팅 네트워크에서의 작업 하역에 대한 연구 동향을 포그 계층을 구성하는 포그 노드들의 특성과 작업 하역을 누가 결정하느냐에 따른 분류에 따라 알아본다. 또한 각 연구 방안에 대하여 해당 방안의 동작에 필요한 메시지와 데이터 송수신 양을 IoT와 포그 노드에서 최악의 경우에 대한 분석을 하였다.

II. 작업 하역 관점에서의 포그 컴퓨팅 네트워크 분류

2.1 포그 노드 구성에 따른 네트워크 분류

포그 컴퓨팅 네트워크는 포그 계층을 구성하는 포그 노드의 동질성에 따라 이기종 네트워크와 동기종 네트워크로 나눌 수 있다. 이기종 포그 컴퓨팅 네트워크는 라우터, 스위치, 게이트웨이, 베이스 스테이션, 로컬 서버 등 다양한 종류의 포그 노드가 존재하고, 각각의 포그 노드의 컴퓨팅 능력이나 저장 능력이 다른 경우를 의미한다. 이에 반해 동기종 포그 컴퓨팅 네트워크는 모두 동일한 기종의 포그 노드로 포그 계층이 구성된다. 스마트폰과 같은 동일한 기종이 실생활에서 많이 사용되고 있고, 5G 네트워크를 사용하면 네트워크 연결 기술이 단일화되고 고속 데이터 전송이 가능하기 때문에 최근 동기종 포그 컴퓨팅 네트워크에서의 연구도 진행되고 있다.

2.2 작업 하역 판단 주체에 따른 분류

포그 서비스를 제공하기 위해 필요한 작업 하역 기능을 누가 어디서 제공하고 판단을 하느냐에 따라 논리적인 분류를 할 수 있다. 작업 하역 판단을 하나의

노드가 주도적으로 하는 중앙 집중적인 경우와 각각의 노드들이 독립적으로 하는 경우로 나눌 수 있다.

중앙 집중적인 경우, 특정 노드가 판단 주도권을 갖는다. 주도권을 갖는 노드는 일반적으로 사용자 요구 조건을 만족시킬 수 있는 포그 노드들이나 클라우드에게 작업을 하역한다. 이러한 주도권을 갖는 노드는 IoT 장비, 포그 노드, 클라우드 계층에 있는 서버가 될 수 있다. 그 중 작업 하역을 결정하는 포그 노드는 포그 제어 노드 (fog controller: FC)로 지칭한다. 이 경우 개념적으로 포그 제어 노드가 다른 일반 포그 노드 위에 위치하는 다단계 구성으로 볼 수 있다. IoT 장비가 처리해야 할 작업을 포그 제어 노드에 전달하는 방식은 포그 제어 노드에게 직접 하는 구조와 포그 접근 노드라고 불리는 포그 계층 접근을 지원하는 노드를 통해서 요청을 전달하는 구조로 나눌 수 있다.

제어 노드를 특별하게 지정하지 않는 경우, 작업 하역에 대한 결정은 관련된 객체가 자율적으로 한다. 진행된 작업 하역 연구들에서는 IoT 장비가 하는 경우와 IoT 장비로부터 작업을 의뢰받은 포그 노드가 하는 경우가 있다. 일반적으로 IoT 장비로부터 작업을 의뢰받은 포그 노드는 작업 노드로 부르고, 작업 노드가 작업 하역에 활용하는 노드는 협업 노드로 부른다.

III. 작업 하역 방안들

작업 하역 목표는 총 실행 시간, 비용, 연산 자원 할당, 에너지 소모의 최소화 등 여러 가지가 있을 수 있다. 총 실행 시간의 최소화나 작업을 위한 메모리 할당을 최적화하는 문제는 NP-complete 문제이다^{7,8}. 그러므로 작업 하역의 최적에 근사한 해를 구하기 위한 여러 가지 방안들이 연구되고 있다. 본 절에서는 포그 컴퓨팅 네트워크의 구조와 작업 하역 판단 주제에 대한 분류별로 작업 하역 방안들에 대하여 알아본다. 소개하는 작업 하역 방안에서의 작업 하역 결정 객체와 작업 하역 대상 객체를 표 1에 정리하였다.

3.1 이기종 포그 컴퓨팅 네트워크에서 중앙 집중적 판단 방법

MTFCT(Mobile to Fog and Cloud Transfer) 알고리즘⁹은 실행시간, 비용, 에너지 소모를 고려하여 IoT 장비가 작업을 스스로 처리할 수 있는 지 결정한다. 스스로 처리할 수 없는 경우 포그 노드가 작업 처리 가능 여부를 결정하고, 불가능한 경우 클라우드로 하역한다. 포그 노드로의 하역이 결정되면 이를 위하여 포그 제어 노드가 포그 노드의 계산 능력과 임팩트 존에

표 1. 작업 하역 방안 별 작업 하역 결정자와 대상자
Table 1. Task offloading decision maker and candidates for each task offloading method

References	offloading decision maker	offloading candidates
[3]	FC	FN
[8]	cloud	FN, cloud
[9]	FC	FN, cloud
[10]	FC	FN, cloud
[11]	FC	FN, cloud
[12]	FN	itself, cloud
[13]	FN	itself, cloud
[14]	FN	FNs
[15]	FN	FN, cloud
[16]	FN	clouds
[17]	FN	IoT, FN, cloud
[18]	FN	itself, cloud
[19]	FN	FN
[20]	IoT	FN, cloud
[21]	IoT	FN, cloud
[22]	mobile FN	FN
[23]	IoT	FNs

기반하여 가장 적합한 포그 노드를 선정한다.

작업 하역 방침에 포그 노드의 제약사항에 대한 사용자의 선호도를 고려하는 방안¹⁰에서는 사용자가 포그 노드의 특징에 대한 개인적인 선호도를 서비스 요청 시 같이 전송한다는 가정을 하였다. 포그 노드는 IoT 장비와의 거리, 서비스 가격, 지연, 대역폭, 신뢰도와 같은 특징을 갖고 있고, 포그 제어 노드는 포그 노드의 순위를 매긴다. 포그 노드의 순위는 사용자의 선호도와 포그 노드의 기능을 언어적, 퍼지 수량화된 계획으로 정하였다. 비슷한 퍼지 수량화된 계획 값을 갖는 포그 노드들은 최소 만족 계획과 최대 만족 계획이라는 두 가지 파라미터를 사용하여 순위를 정하였다. 이 연구에서는 순위 기반 작업 스케줄링 방안을 제안하고, 메트릭으로는 사용자 만족도, 실행시간, 에너지 소비량을 사용하였다. 이 방안은 IoT 장비가 직접 포그 제어 노드로 작업 요청을 보내고, 포그 제어 노드가 최적의 포그 노드를 선정하여 작업을 하역한다.

ADGTS(Adaptive Double Fitness Genetic algorithm)에 기반한 작업 스케줄링 알고리즘¹¹은 스마트 도시에 산재한 IoT 장비의 연산 능력과 통신에 대한 요구조건을 고려하여 포그 컴퓨팅의 연산과 통신 문제를 모델링하고, 그에 따른 문제 해결방안을 제안하였다. 최적화 목표를 작업 총 실행시간과 포그 자원의 통신 비용을 최소화하는 것으로 설정하였다. 이 알고리즘은 유전적 진화, 돌연변이, 자연적 선택과 같

은 생물학적 현상에서 습득한 정보를 바탕으로 최선의 개인을 발견하고, 개인을 하나의 해결책으로 사용하여 개인의 메트릭을 평가하는 이중 적합성 함수를 사용하였다. 이를 통하여 포그 노드의 연산능력, 지연 조건, 통신 능력 등을 고려하여 포그 자원을 효율적으로 활용하고 협업할 수 있도록 하였다.

BLA(Bees Life Algorithm)^[8]은 바이오에 기반한 최적화 방안으로 지금까지 언급된 연구들과는 달리 제어 노드가 클라우드 계층에 있다고 가정하였다. 제어 노드는 IoT 사용자가 생성한 작업을 포그 노드를 통해서 받고, 하나의 작업으로 세부 작업으로 분할하여 포그 노드들에게 세부 작업을 하역한다. 포그 노드가 결과를 제어 노드에게 보내면 취합하여 서비스 응답을 IoT 사용자에게 보내는 식으로 동작하도록 하였다. BLA는 서비스 비용을 연산을 위하여 할당하는 메모리로 설정하고, CPU 실행 시간과 서비스 비용을 메트릭으로 사용하여 작업 스케줄링과 포그 노드로 작업의 최적화된 분배를 고려하였다.

3.2 이기종 포그 컴퓨팅 네트워크에서 자율적 판단 방안

사용자 인터페이스와 LAN 서버 시스템을 담당하는 프론트엔드 포털에서 사용자의 요청을 받아서 포그 노드로 전달하는 모델을 사용한 연구^[12]에서는 최적화 목적을 포그 노드와 클라우드 서버에서의 전력 소모를 최소화하면서 IoT 응용의 지연 제약조건을 만족시키는 것으로 설정하여 최적의 작업 부하 할당 문제로 공식화했다. 작업 부하 할당 문제를 포그 계층에서의 전력 소비와 계산 지연, 클라우드에서의 전력 소비와 계산 지연, 포그와 클라우드간의 통신 지연 문제로 분할하여 각각을 해결하는 방식을 사용하였다. 전력소비와 계산 지연 문제는 콘벡스 최적화 기법을 사용하고, 통신 지연 문제는 Hungarian 알고리즘으로 해결하였다.

작업 하역 전송 지연을 줄이고 제한된 라디오 자원의 제약조건을 없애기 위해서 비직교 다중 접속을 사용하여 여러 사용자가 같은 포그 노드에 동시에 데이터를 전송하는 시스템을 가정한 모델^[13]은 하역 결정, 사용자 스케줄링, 자원 할당 문제를 최적화 문제로 구성하여, 목표를 전체 시스템의 에너지 비용과 사용자의 지연을 최소화하는 것으로 설정하였다. 원래의 최적화 문제를 콘벡스 문제로 변형하여 분산처리가 가능하도록 변형하였다.

POST(parallel offloading of splittable tasks)^[14]은 포그 노드에서 분리 가능한 작업을 여러 개의 서버 작

업으로 나뉘서 각각의 서버 작업을 다른 포그 노드로 하역하여 병렬처리를 통해 서비스 지연을 최소화하는 문제를 연구하였다. 이를 위하여 게임 이론을 활용하여 일반화된 NE 문제(generalized Nash equilibrium problem: GNEP)로 공식화하고, 지역 정보와 Gauss-Seidel 형 방법을 사용하여 해결하였다. POST 알고리즘에서는 시스템 지연을 메트릭으로 사용하였다.

PORA(predictive offloading and resource allocation)^[15]은 작업 하역 문제를 트래픽을 예측하여 동적 하역과 자원 할당 문제를 세부 큐잉 모델을 사용하여 스토캐스틱 네트워크 최적화 문제로 구성하였다. 이 방안에서는 포그 계층을 에지 포그 단계와 중앙 포그 단계, 2단계로 구성하여, 필요 시 작업이 에지 포그 노드, 중앙 포그 노드, 클라우드로 하역되도록 하였다. 시스템의 모든 큐에 대하여 안정성을 보장하며 포그 노드의 시 평균 에너지 소모를 최소화 하는 것을 목표로 설정하고, Lyapunov 최적화 기술을 사용하여 해결하였다.

클라우드로 작업을 하역할 때 하나의 클라우드를 사용하는 일반적인 경우와는 달리, 하나의 포그 노드가 여러 개의 클라우드를 사용할 수 있는 포그 컴퓨팅 구조를 제안한 방안^[16]도 있다. 이 방안에서는 포그 노드의 자원 활용을 효율적으로 하고 서비스 지연을 줄이기 위해 자원 유닛으로 사용하는 컨테이너를 사용한 서비스 모델을 디자인하였다. 서비스 모델은 일시적인 서비스 모델, 장기적인 서비스 모델 두 가지를 사용하여, 일시적인 서비스 모델은 실시간 요청을 처리하고, 장기적인 서비스 모델은 포그 노드 자신 안의 서비스와 이웃 노드와 협업 서비스를 사용하도록 하였다. 이러한 서비스 모델을 사용하여 센서 네트워크의 수명을 연장하고 작업의 지연 제약 조건을 줄이기 위해 에너지 균형에 기반한 작업 스케줄링 알고리즘을 제안하였다. 단말 장치의 전송 에너지 소모를 제어하기 위해 에너지 균형 정책을 디자인하고, 실시간 요청에 대한 스케줄링을 위해 동적 임계치를 사용하여 포그 노드에서 동시 진행되는 작업의 숫자를 증가시켰다.

CAME(Cloud Assisted Mobile Edge computing) 프레임워크^[17]에서는 IoT 장비가 보내는 정보와 클라우드 정보를 포그 노드가 취합하여 IoT 장비로도 작업 하역을 할 수 있고, 포그 노드의 연산능력 외 추가적인 연산능력이 필요하면 클라우드 자원을 사용할 수 있게 했다. 연산지연과 통신지연을 포함하는 시스템 지연을 서비스 품질을 결정짓는 항목으로 설정하였다. 최적화 문제는 시스템 지연과 통신 오버헤드를

최소화하는 것으로 설정하고, Karush-Kuhn-Tucker 조건으로 풀 수 있는 컨벡스 문제임을 증명하였다. 문제 해결을 위하여 선형 복잡도를 갖는 알고리즘을 제안하였다.

스마트 공장에서 예외를 감지하거나 고장을 진단하는 데 사용되는 데이터의 실시간 분석과 대응하는 피드백 메커니즘은 주요 이슈이다. 스마트 제조 장비의 요청을 포그 노드나 클라우드로 전송하는 포그-클라우드 작업 스케줄링 알고리즘^[18]은 컨테이너에 기반한 포그 노드의 가상화 기술을 사용하였다. 이 방안에서는 제한된 지연 안에 완료할 수 있는 작업만 수락하고, 수락한 작업에 대하여 포그 노드에서 동시에 실행하는 작업 수를 최대화하는 것을 목표로 한다. 또한, 컨테이너의 특성에 따라 작업 지연을 줄이기 위한 해당 메커니즘을 제안하였다.

스마트 공장에서 작업 하역 문제의 중점을 공장 장비의 에너지 소비 문제에 둔 ELBS(energy-aware load balancing and scheduling) 방안^[19]에서는 포그 노드에 다중 에이전트 시스템을 도입하여 스마트 공장의 자동화 능력을 향상시키는 것을 목적으로 한다. 다중 에이전트 시스템은 포그 노드를 사용하여 각 에이전트가 독립적, 자율적으로 주어진 작업의 다음 단계를 결정하는 전략을 선택하게 하고, 각 에이전트가 복잡한 문제를 병렬로 해결하기 위해서 서로 협상하며, 객체 지향적 방법을 사용하여 다단계, 복합적으로 구성되어 전체 생산 시스템의 복잡도를 줄일 수 있다. ELBS 방안에서는 포그 기반 에너지 인식 모델을 구성하여 장비의 에너지 소비와 작업 부하간의 관계를 계산하는 데 사용하고, 에너지 인식 부하 분배 최적화 함수를 구성하고, PSO(Particle swarm optimization) 알고리즘을 사용하여 최적해를 구했다.

포그 노드가 아닌 IoT 장비가 작업 하역을 결정하는 방안들도 있다. 전체 작업 응답시간을 최소화하는 작업 분배와 스케줄링하는 온라인 알고리즘 OnDisc(online job dispatching and scheduling algorithm)^[20]에서는 모바일 장비에서 작업이 전달되는 시간부터 해당 장비가 결과를 받는 데 걸리는 시간을 작업 응답시간으로 설정하여 모바일 장비와 서버간의 통신 지연, 서버에서의 대기 시간과 처리 시간을 포함하도록 하였다. 각 작업의 응답시간에 지연의 민감도를 나타내는 가중치 응답 시간을 설정하여 높은 가중치를 갖는 작업은 높은 우선순위로 처리될 수 있도록 하였다. 연구의 목적은 모든 작업의 전체 가중치 응답 시간을 최소화하는 것으로 두고, OnDisc를 통하여 IoT 장비가 작업을 스스로 처리할 지, 아니면 포그

노드나 클라우드로 하역할 지 결정하도록 하였다. 작업 하역은 그리디 방식으로 전체 가중치 응답 시간을 최소 증가시키는 포그 노드에 작업을 하역한다.

IoT 장비가 다중 라디오 접근 기술을 탑재하여 여러 포그 노드와 원격 클라우드 서버 연결이 가능하다는 전제 하에, 포그 노드와 원격에 있는 클라우드 서버의 자원을 IoT 사용자에게 하역하는 문제에 대한 연구^[21]도 있다. IoT 사용자들 간의 경쟁과 포그 노드들의 한정된 처리 능력 할당을 효율적으로 하는 문제를 계산 하역 게임으로 수식화하고, 지연과 에너지 소모로 서비스의 만족도를 측정하여, IoT 사용자는 자신의 경험 품질을 최대화하는 것을 게임 목적으로 설정하였다. Potential game 방안을 사용하여 NE가 있음을 증명하고, anarchy의 비용에 대한 상한치를 제공하였다. 사용자의 수가 증가함에 따라 균형값에 도달하는 시간 복잡도가 지수적으로 증가하므로, 최적에 근접한 자원 할당 방안을 제안하였다.

3.3 동기종 포그 컴퓨팅 네트워크에서 중앙 집중적 판단 방안

DEBTS(Delay energy balanced task scheduling)^[3]은 가상 현실, 모바일 단말에서의 온라인 3D 게임 등과 같은 에너지 소비와 지연이 중요한 응용 프로그램을 위하여, 서비스 지연과 에너지 소비 간의 균형을 고려하였다. DEBTS에서는 포그 제어 노드가 온라인 알고리즘을 디자인하여 실시간으로 작업 스케줄링을 동적으로 결정하고, 서비스 지연과 에너지 소비간의 상관관계를 특성화하기 위한 효과적인 제어 파라미터를 유출했다. 이러한 분석과 Lyapunov 최적화 기법을 결합하여 전체적인 에너지 소비를 최소화하며 평균 서비스 지연과 지연 지터를 줄이는 것을 목표로 설정하였다.

3.4 동기종 포그 컴퓨팅 네트워크에서 자율적 판단 방안

MEETS(a maximal energy-efficient task scheduling) 알고리즘^[22]은 스펙트럼과 시간 자원 모두 고려한 에너지 효율을 정확하게 평가하기 위하여 회로, 연산, 하역한 에너지 소비를 고려한 포괄적인 분석 모델을 사용하였다. 또한 모바일 작업 노드가 협업 노드로 작업 하역을 할 때 성능 이득과 에너지 비용 간의 트레이드오프를 탐구하여 에너지 효율 최적화 문제를 설정, 수학적으로 분석하였다. MEETS 알고리즘은 작업 스케줄링에서 에너지 효율을 최대화하는 것을 목표로 하고, 포그 노드에서의 연산 자원에 대한 실질적인 제

약과 이웃 환경에서 사용한 스펙트럼도 고려하였다.

FTO(Fair task offloading) 방안²³⁾은 배터리로 동작하는 포그 노드를 고려하여 작업 지연을 낮추면서 다수의 포그 노드들에게 공평하게 작업 하역을 하는 방안이다. FTO는 작업 지연을 낮추고 대응하는 에너지 소비를 줄이는 것을 기준으로 작업을 하역할 포그 노드를 선택하였다. 이를 위해 작업 지연과 그에 해당하는 에너지 소비를 공식화하고, 각 포그 노드에 공평한 스케줄링 메트릭을 도입하였다. 먼저 공평함을 평가하는 메트릭에 따라 하역할 포그 노드 군을 선택하고, 선택된 노드들에게 작업 지연을 최소화하는 방침에 따라 작업을 서브 작업으로 나눠서 하역한다.

IV. 작업 하역 방안 별 메시지 송수신 부하 분석

앞 절에서 소개된 작업 하역 방안들은 네트워크 구조, 최적화를 판단하는 메트릭, 네트워크 구성 요소인 IoT 장비, 포그 노드 등의 특성에 대한 가정이 다르기 때문에 각 방안들의 성능 비교를 직접적으로 하기는 어렵다. 또한 대부분의 방안들은 해당 방안이 설정한 최적화 목표를 달성하기 위한 분석만을 기반으로 성능 분석을 하였으나, 제안하는 방식이 동작하기 위하여 필요한 정보는 작업 하역을 판단하는 주체가 이미 가지고 있다는 전제하여 설명하였다. 이에, 본 논문에서는 각 방안들이 동작하는 데 필요한 제어 메시지와 작업 하역 및 결과 전달을 위하여 필요한 메시지를 성능분석의 비용으로 산정하고, 최악의 경우에 대하여 성능을 분석한다. 필요한 제어 메시지, 작업의뢰 데이터와 처리 결과로 유발되는 전달 메시지들은 네트워크 트래픽, 전송비용을 유발하며, 해당 메시지를 송수신하는 과정에서 작업 하역에 연관된 장비들에게 메모리를 요구하고, 전송 시간은 지연 요인으로 산정되므로 각 방안의 성능에 영향을 미치는 요소가 된다. 분석은 작업 하역에 관여하는 IoT, 포그 노드, 포그 제어 노드에 대하여 중점적으로 한다. 클라우드의 경우에는 IoT나 포그 계층의 장비에 비하여 자원 제약에서 자유로우므로 작업 하역 판단을 클라우드에서 하는 BLA¹⁸⁾을 제외한 다른 방안들에 대해서는 분석하지 않는다.

포그 컴퓨팅 네트워크 구성에 따라 IoT, 포그 노드, 포그 제어 노드 간의 연결이 다양하게 있을 수 있다. IoT가 무선으로 신호 환경 내 여러 포그 노드와 연결할 수 있는 경우, 해당 환경을 IoT_{Zone} 으로 설정한다. 또한 포그 노드에 연결할 수 있는 IoT들을 포함하는

공간을 FN_{Zone} 으로 설정한다. 최악의 경우에 대한 분석을 위하여 시스템 내의 IoT 장비들이 모두 작업 하역 요청을 하고, 이러한 작업 하역 요청 마다 작업 하역 결정을 위한 정보 요청과 취합을 한다고 가정한다.

표 2에 해당 분석에 사용하는 기호를 정리하였고, 각 방안에 대하여 작업 하역 관련 대상 노드별로 송수신 하는 메시지 양을 분석한 결과를 표 3에 정리하였다.

분석한 내용은 각 작업 하역 방안이 가정한 포그 컴퓨팅 네트워크 구조와 밀접한 연관이 있다. IoT와 포그 노드의 연결 방식, 포그 노드들 간의 연결 방식, 포그 노드 혹은 포그 제어 노드와 클라우드 간의 연결 방식에 따라 필요한 제어 메시지와 작업 하역 데이터와 결과가 전달되는 경로가 지정된다. 따라서 각 객체들 간의 연결 방식이 단순하면 발생하는 메시지의 경로도 단순해진다. 그러나 네트워크를 구성하는 다른 객체들과의 협업을 함으로써 작업 하역의 성능을 높이려고 한 연구 방안들은 작업 데이터에 비하여 크기는 작지만 다른 객체의 상태를 알기 위한 제어 메시지가 많이 발생하게 된다.

BLA¹⁸⁾의 경우 작업 하역 결정이 클라우드에서 이루어지고, 작업 데이터와 결과도 클라우드에서 작업 하역 포그 노드로 전달하고 취합하기 때문에 전반적으로 전송지연이 크고, 포그 노드의 메시지 부하가 다른 방안보다 높다. CAME¹⁷⁾에서는 작업 하역이 IoT로도 이루어 질 수 있기 때문에 IoT의 메시지 부하가 높아서 자원이나 연산 능력이 낮은 일반적인 IoT 응

표 2. 표기법
Table 2. Notation

Symbol	Definition
N_{IoT}	No. of IoTs in the system
$N_{IoT_{Zone}}$	No. of IoTs in a fog node's zone
N_{FN}	No. of fog nodes in the system
$N_{FN_{IoT_{Zone}}}$	No. of fog nodes in a IoT's zone
$N_{FC_{IoT_{Zone}}}$	No. of fog control nodes in IoT's zone
S_T	Size of task
S_R	Size of task result
S_{CR}	Size of control information request
S_{CI}	Size of control information
S_O	Size of task offloading request
S_{OM}	Size of task offloading management message

표 3. 작업 하역 관련 구성 요소 별 송수신 메시지 분석
Table 3. Analysis of sending and receiving messages for entities which are participating in task offloading

References	entity	Amounts of messages
[3]	IoT	$S_O + S_T + S_R$
	FN	$2N_{IoT_{FN_{ave}}} S_O + N_{IoT}(S_{CR} + S_{CI} + 2(S_T + S_R))$
	FC	$N_{IoT}(S_{CR} + S_{CI} + S_O + 2S_{OM})$
[8]	IoT	$S_O + S_T + S_R$
	FN	$2N_{IoT_{FN_{ave}}} S_O + N_{IoT}(S_{CR} + S_{CI} + S_{OM}) + 4N_{IoT}(S_T + S_R)$
	cloud	$N_{IoT}(S_O + S_{CR} + S_{CI} + S_{OM}) + 4N_{IoT}(S_T + S_R)$
[9]	IoT	$S_O + S_T + S_R$
	FN	$2N_{IoT_{FN_{ave}}} S_O + N_{IoT}(S_{CR} + S_{CI} + 2(S_T + S_R))$
	FC	$N_{IoT}(S_O + S_{CR} + S_{CI} + S_{OM} + 2(S_T + S_R))$
[10]	IoT	$N_{FN_{iZave}}(S_O + S_T) + S_R$
	FN	$2N_{IoT_{FN_{ave}}} S_O + N_{IoT}(S_{OM} + S_{CR} + S_{CI} + 2(S_T + S_R))$
	FC	$N_{IoT}(S_O + 2S_{OM} + S_{CR} + S_{CI} + 2(S_T + S_R))$
[11]	IoT	$S_O + S_T + S_R$
	FN	$2N_{IoT_{FN_{ave}}} S_O + N_{IoT}(S_{OM} + S_{CR} + S_{CI} + 2(S_T + S_R))$
	FC	$N_{IoT}(S_O + 2S_{OM} + S_{CR} + S_{CI} + 2(S_T + S_R))$
[12], [13], [16], [18]	IoT	$S_O + S_T + S_R$
	FN	$2N_{IoT_{FN_{ave}}}(S_O + S_T + S_R)$
[14], [19], [22]	IoT	$S_O + S_T + S_R$
	FN	$2N_{IoT_{FN_{ave}}} S_O + N_{IoT}(S_{OM} + S_{CR} + S_{CI} + 2(S_T + S_R))$
[15]	IoT	$S_O + S_T + S_R$
	edge FN	$2N_{IoT_{FN_{ave}}}(S_O + S_T + S_R)$
	central FN	$N_{IoT}(S_O + 2(S_T + S_R))$
[17]	IoT	$S_O + N_{IoT_{FN_{ave}}}(S_T + S_R) + (N_{IoT_{FN_{ave}}} - 1)(S_{CR} + S_{CI} + S_{OM})$
	FN	$N_{IoT_{FN_{ave}}}(S_O + S_{CR} + S_{CI} + S_{OM} + 2(S_T + S_R))$
[20]	IoT	$N_{FN_{iZave}}(S_{CR} + S_{CI}) + S_O + S_T + S_R$
	FN	$N_{IoT_{FN_{ave}}}(S_{CR} + S_{CI} + S_O + S_{OM} + 2(S_T + S_R))$
[21], [23]	IoT	$N_{FN_{iZave}}(S_{CR} + S_{CI} + S_O) + S_T + S_R$
	FN	$N_{IoT_{FN_{ave}}}(S_{CR} + S_{CI} + S_O + 2(S_T + S_R))$

용에는 적용하기 어려운 방법이다. 포그 제어 노드가 전권을 갖고 작업 하역을 하는 방안들^[3,9,10,11]은 포그 제어 노드에 장애가 발생하는 경우 전체 시스템이 동작하지 않는 단일점 장애의 문제가 있다. IoT 장비가 작업 하역을 결정하는 방안들^[20,21,23]은 IoT 장비가 자신의 연결 환경 안에 있는 포그 노드들의 상태를 파악하여 최적의 결과로 작업을 수행하는 포그 노드를 판

단해야 하므로 제어 메시지 송수신에 대한 부담과 작업 하역 판단을 위한 연산 능력이 필요하다.

V. 결 론

본 논문에서는 포그 컴퓨팅 네트워크에서 IoT 장비의 작업 요청을 처리할 최적의 장소를 결정하여 작업

을 보내는 작업 하역 문제를 해결하기 위한 다양한 방안들을 알아보았다. 최적의 목표는 여러 가지가 있을 수 있는데, 전체 작업의 총 실행 시간, 에너지 소모, 비용, 지연의 최소화 등 다양한 항목들로 구성된다. 어떤 목표를 설정하느냐는 실행하는 응용 프로그램의 특성에 따라 달라질 수 있다. 포그 계층을 구성하는 포그 노드의 특성과 작업 하역을 누가 결정하느냐에 따라 여러 방안들을 분류하여 알아보았다. 작업 하역 문제는 NP-complete 문제이므로 여러 접근방식으로 문제를 해결하려는 연구가 진행되었는데, 휴리스틱 방안, Lyapunov 최적화 기법을 사용한 방안, potential game으로 모델링한 방안, 바이오에 기반한 최적화 방안, 컨테이너에 기반한 가상화 기술을 사용하는 방안 등 다양한 방안들을 소개하였다. 또한 각 방안의 동작에 필요한 송수신 메시지를 성능분석의 비용으로 산정하고, 최악의 경우에 대하여 성능을 분석하였다.

References

- [1] *IoT connections to grow 140% to hit 50 billion by 2022, as edge computing accelerates roi, 2018*, <https://www.juniperresearch.com/press/press-releases/iot-connections-to-grow-140pc-to-50-billion-2022>.
- [2] *Fog computing and the internet of things: Extend the cloud to where the things are, 2015*, https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf.
- [3] Y. Yang, S. Zhao, W. Zhang, Y. Chen, X. Luo, and J. Wang, "DEBTS: Delay energy balanced task scheduling in homogeneous fog networks," *IEEE Internet of Things J.*, vol. 5, no. 3, pp. 2094-2106, Jun. 2018.
- [4] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog," *IEEE J. Sel. Areas Commun.*, vol. 4, no. 5, pp. 1185-1192, Oct. 2017.
- [5] V. Gazis, "A survey of standards for machine-to-machine and the internet of things," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 482-511, 1st Quart, 2017.
- [6] S. Park, J. Lee, and S. Park, "A study on the problems in fog computing design and implementation," in *Proc. Symp. KICS*, pp. 972-973, Jan. 2019.
- [7] X.-Q. Pham and E.-N. Huh, "Towards task scheduling in a cloud-fog computing system," in *Proc. APNOMS 2016*, pp. 1-4, Kanazawa, Oct. 2016.
- [8] S. Bitam, S. Zeadally, and A. Mellouk, "Fog computing job scheduling optimization based on bees swarm," *Enterp. Inf. Syst.*, vol. 12, no. 4, pp. 373-397, 2018.
- [9] R. Jindal, N. Kumar, and H. Nirwan, "MTFCT: A task offloading approach for fog computing and cloud computing," in *Proc. 2020 10th Int. Conf. Cloud Computing, Data Sci. & Eng. (Confluence)*, pp. 145-149, Noida, India, 2020.
- [10] M. A. Benblidia, B. Brik, L. Merghem-Boulahia, and M. Esseghir, "Ranking fog nodes for tasks scheduling in fog-cloud environments: A fuzzy logic approach," in *Proc. IWCMC*, pp. 1451-1457, Tangier, Morocco, 2019.
- [11] Q. Liu, Y. Wei, S. Leng, and Y. Chen, "Task scheduling in fog enabled internet of things for smart cities," in *Proc. IEEE ICCT 2017*, pp. 975-980, Oct. 2017.
- [12] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet of Things J.*, vol. 3, no. 6, pp. 1171-1181, Dec. 2016.
- [13] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. Leung, "Hybrid computation offloading in fog and cloud networks with non-orthogonal multiple access," in *Proc. IEEE INFOCOM Wkshps*, pp. 154-159, 2018.
- [14] Z. Liu, Y. Yang, K. Wang, Z. Shao, and J. Zhang, "POST: Parallel offloading of splittable tasks in heterogeneous fog networks," *IEEE Internet of Things J.*, vol. 7, no. 4, pp. 3170-3183, Apr. 2020.
- [15] X. Gao, X. Huang, S. Bian, Z. Shao, and Y. Yang, "PORA: Predictive offloading and resource allocation in dynamic fog computing systems," *IEEE Internet of Things J.*, vol. 7, no. 1, pp. 72-87, Jan. 2020.

- [16] J. Luo, L. Yin, J. Hu, C. Wang, X. Liu, X. Fan, and H. Luo, "Container-based fog computing architecture and energy-balancing scheduling algorithm for energy IoT," *Future Generat. Comput. Syst.*, vol. 97, pp. 50-60, 2019.
- [17] X. Ma, S. Zhang, W. Li, P. Zhang, C. Lin, and X. Shen, "Cost-efficient workload scheduling in cloud assisted mobile edge computing," in *Proc. IEEE/ACM IWQoS*, pp. 1-10, Vilanova i la Geltru, 2017.
- [18] L. Yin, J. Luo, and H. Luo, "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacture," *IEEE Trans. Ind. Informatics*, vol. 14, no. 10, pp. 4712-4721, Oct. 2018.
- [19] J. Wan, B. Chen, S. Wang, M. Xia, D. Li, and C. Liu, "Fog computing for energy-aware load balancing and scheduling in smart factory," *IEEE Trans. Ind. Informatics*, vol. 14, no. 10, pp. 4548-4556, Oct. 2018.
- [20] H. Tan, Z. Han, X.-Y. Li, and F. C. M. Lau, "Online job dispatching and scheduling in edge-clouds," in *Proc. IEEE INFOCOM*, pp. 1-9, May 2017.
- [21] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: A computation offloading game," *IEEE Internet of Things J.*, vol. 5, no. 4, pp. 3246-3257, Aug. 2018.
- [22] Y. Yang, K. Wang, G. Zhang, X. Chen, X. Luo, and M. Zhou, "MEETs: Maximal energy efficient task scheduling in homogeneous fog networks," *IEEE Internet of Things J.*, vol. 5, no. 5, pp. 4076-4087, Oct. 2018.
- [23] G. Zhang, F. Shen, Y. Yang, H. Qian, and W. Yao, "Fair task offloading among fog nodes in fog computing networks," in *Proc. IEEE ICC*, pp. 1-6, May 2018.

우 미 애 (Miae Woo)



1985년 2월 : 연세대학교 전자공학과 졸업

1991년 12월 : 미국 Purdue 대학교 전기컴퓨터공학과 석사

1995년 12월 : 미국 Purdue 대학교 전기컴퓨터공학과 박사

1985년~1989년 : DACOM 연구원

1996년~1998년 : 삼성전자(주) 수석연구원

1998년~현재 : 세종대학교 전자정보통신공학과 교수
<관심분야> 네트워크 구조 및 프로토콜, 이동 네트워크

[ORCID:0000-0002-1704-6430]