



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0143729
(43) 공개일자 2023년10월13일

(51) 국제특허분류(Int. Cl.) H04L 67/563 (2022.01) H04L 67/1012 (2022.01) H04L 67/288 (2022.01) H04L 67/61 (2022.01)	(71) 출원인 한림대학교 산학협력단 강원도 춘천시 한림대학길 1, 한림대학교(옥천동)
(52) CPC특허분류 H04L 67/563 (2022.05) H04L 67/1012 (2022.05)	(72) 발명자 김태운 강원도 춘천시 백령로 215, 108동 602호
(21) 출원번호 10-2022-0042713	(74) 대리인 특허법인본
(22) 출원일자 2022년04월06일 심사청구일자 2022년04월06일	

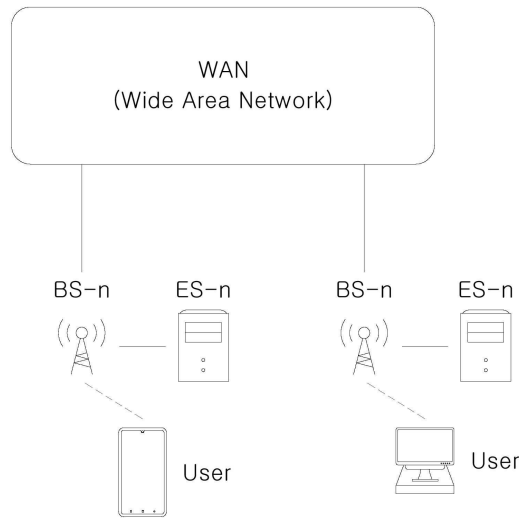
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 엣지 컴퓨팅에서 서비스 프로비저닝을 위한 시스템, 방법, 및 프로그램

(57) 요약

복수의 ES(Edge Server)로 구성된 시스템에 포함된 적어도 하나의 ESM(ES Manager)의 제어 방법이 개시된다. 본 제어 방법은, 적어도 하나의 사용자 단말로부터 일 서비스 유형에 대한 요청을 수신한 적어도 하나의 ES가 해당 요청에 매칭되는 서비스 유형의 실행 이미지를 저장하고 있지 않음에 따라 발생하는, 적어도 하나의 누락된 요청을 식별하는 단계, 누락된 요청이 누락된 요청과 매칭되는 서비스 유형의 실행 이미지를 저장하고 있는 적어도 하나의 다른 ES로 리디렉션 되도록 지원하는 단계를 포함한다.

대표도 - 도1



(52) CPC특허분류

H04L 67/288 (2022.05)

H04L 67/61 (2022.05)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711157335
과제번호	2021R1F1A1059109
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	기본연구
연구과제명	공공안전 및 재난 통신을 위한 지능형 무선 네트워크 최적화 기술 연구
기 여 율	1/1
과제수행기관명	한림대학교 산학협력단
연구기간	2021.06.01 ~ 2024.02.29

명세서

청구범위

청구항 1

복수의 ES(Edge Server)로 구성된 시스템에 포함된 적어도 하나의 ESM(ES Manager)의 제어 방법에 있어서,

적어도 하나의 사용자 단말이 상기 복수의 ES 중 적어도 하나에 요청하는 서비스 유형, 상기 복수의 ES 각각에 저장된 실행 이미지에 매칭되는 서비스 유형, 및 상기 복수의 ES 각각의 가용 용량을 실시간으로 모니터링하는 단계;

적어도 하나의 사용자 단말로부터 일 서비스 유형에 대한 요청을 수신한 적어도 하나의 ES가 상기 요청에 매칭되는 서비스 유형의 실행 이미지를 저장하고 있지 않음에 따라 발생하는, 적어도 하나의 누락된 요청을 식별하는 단계; 및

상기 누락된 요청이 상기 누락된 요청과 매칭되는 서비스 유형의 실행 이미지를 저장하고 있는 적어도 하나의 다른 ES로 리디렉션 되도록 지원하는 단계;를 포함하고,

상기 지원하는 단계는,

서비스 유형 각각의 우선 순위 및 복수의 ES 간의 전송 비용을 바탕으로, 상기 누락된 요청과 매칭되는 실행 이미지를 저장하고 있는 적어도 하나의 다른 ES와 관련된 엣지 가중치(edge weight)를 산출하고,

상기 산출된 엣지 가중치를 바탕으로, 상기 누락된 요청이 리디렉션 될 서버 ES를 선택하는, ESM의 제어 방법.

청구항 2

제1항에 있어서,

상기 누락된 요청을 식별하는 단계는,

누락된 요청이 발생한 ES의 식별 정보, 누락된 요청에 매칭되는 서비스 유형, 및 누락된 요청 각각을 처리하기 위해 필요한 자원 단위의 수를 식별하고,

누락된 요청과 매칭되는 서비스 유형의 실행 이미지를 저장하고 있는 적어도 하나의 ES의 식별 정보 및 여유 자원 단위의 수를 식별하는, ESM의 제어 방법.

청구항 3

제1항에 있어서,

상기 지원하는 단계는,

이하 수학식에 따라 누락된 요청의 항목인 i 및 ES인 k 간의 엣지 가중치(v_{ik})를 산출하고,

$$v_{ik} = f_{pri}(pri(i)) \cdot w_1 + f_{trans}(trans(i, k)) \cdot w_2$$

상기 $f_{pri}(pri(i))$ 는 상기 누락된 요청에 매칭되는 서비스 유형의 우선 순위($pri(i)$)를 이하 수학식에 따라 정규화한 것이고,

$$(pri(i) - pri_{min}) / (pri_{max} - pri_{min})$$

상기 $f_{trans}(trans(i, k))$ 는 상기 누락된 요청이 발생한 누락 ES와 상기 k 에 해당하는 ES 간의 전송 비용($trans(i, k)$)을 이하 수학식에 따라 정규화한 것이고,

상기 w_1 및 상기 w_2 는 각각 우선 순위 및 전송 비용과 관련된 가중치 비율을 나타내는 파라미터인, ESM의 제어 방법.

청구항 4

제3항에 있어서,

상기 지원하는 단계는,

이하 수학식에 따라 모든 누락된 요청의 인스턴스의 총 엣지 가중치를 최대화하도록 누락된 요청 각각의 서빙 ES를 선택하고,

$$\max_{\mathbf{x}} \sum_{i \in I} \sum_{k \in K} x_{ik} \times v_{ik}$$

x_{ik} 는 누락된 요청의 항목인 i 의 인스턴스 중 ES인 k 에 전달되는 인스턴스의 수이고, v_{ik} 는 누락된 요청의 항목인 i 및 ES인 k 간의 엣지 가중치이고,

상기 지원하는 단계는,

이하 수학식들에 따른 제약 조건에 따라 상기 수학식에 매칭되는 서빙 ES를 선택하고,

$$\begin{aligned} \forall k \in K: \sum_{i \in I} x_{ik} \times r_i &\leq \pi_k, \\ \forall i \in I: \sum_{k \in K} x_{ik} &\leq s_i, \\ \forall i \in I, \forall k \in K: x_{ik} &\geq 0, \text{ and integer.} \end{aligned}$$

r_i 는 누락된 요청의 항목인 i 의 각 인스턴스를 처리하는 데에 필요한 자원 단위의 수이고, s_i 는 누락된 요청의 항목인 i 의 전체 인스턴스의 수인, ESM의 제어 방법.

청구항 5

제3항에 있어서,

상기 지원하는 단계는,

상기 복수의 ES가 구분된 복수의 그룹 각각을 관리하는 ESM이, 각 그룹 내에서, 누락된 요청의 항목인 i 에 있어서 각 ES로 전달될 인스턴스의 수(x)를 선택하고,

$$L_{\eta(h)}(\mathbf{x}_{k \in \eta(h)}; \lambda) = \sum_{k \in \eta(h)} \sum_i (v_{ik} - \lambda_i) x_{ik}.$$

$\eta(h)$ 는 ESM인 h 의 제어 하에 있는 ES들의 인덱스 세트이고,

각 그룹 내에서, 이하 수학식들에 따라 ES인 k 의 λ 를 업데이트하고,

$$\begin{aligned} \mathbf{x}_k^*(\lambda) &= \arg \max_{\mathbf{x}_k} L_k(\mathbf{x}_k; \lambda) \\ &\sum_i x_{ik} \times r_i \leq \pi_k \\ \forall i: x_{ik} &\geq 0, \text{ integer.} \end{aligned}$$

$$\begin{aligned} \min_{\lambda} g(\lambda) &= \sum_k g_k(\lambda) + \sum_i \lambda_i s_i \\ \text{subject to:} \\ \forall i: \lambda_i &\geq 0, \end{aligned}$$

$g_k(\lambda) = L_k(\mathbf{x}_k^*, \lambda)$ 인, ESM의 제어 방법.

청구항 6

전자 장치에 있어서,

적어도 하나의 인스트럭션이 저장된 메모리;

복수의 ES(Edge Server)와 통신을 수행하기 위한 통신부; 및

상기 인스트럭션을 실행하여 제1항의 제어 방법을 수행하는 프로세서;를 포함하는, 전자 장치.

청구항 7

컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램에 있어서,

전자 장치의 프로세서에 의해 실행되어 상기 전자 장치로 하여금 제1항의 제어 방법을 수행하도록 하는, 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램.

발명의 설명

기술 분야

[0001] 본 개시는 엣지 컴퓨팅의 네트워크 운영 방식에 관한 것으로, 보다 상세하게는, 적어도 하나의 서비스 유형에 대한 요청이 누락된 경우 그에 적합한 ES(Edge Server)를 선택하여 누락된 요청을 할당(리디렉션)하는 기술에 관한 것이다.

배경 기술

[0002] EDGE 컴퓨팅(EC)은 유망한 컴퓨팅 기술로, 컴퓨팅 및 스토리지 리소스가 최종 사용자와 가까운 네트워크 에지에 배치되는 패러다임이다. 중앙 집중식 클라우드 컴퓨팅(CC)에 비해 엣지 컴퓨팅은 응답 시간 감소, 사용자 장치의 배터리 절약, 대역폭 비용 절감 등과 같은 추가 이점이 있다.

[0003] 오늘날, 엣지 컴퓨팅은 하나 이상의 기지국(BS) 또는 무선 액세스 네트워크와 함께 배치된 엣지 서버(ES)를 가짐으로써 다중 액세스 엣지 컴퓨팅(MEC)이라고 하는 모바일 셀룰러 네트워크(예: 5G)에 통합되었다. 이러한 배치를 통해 상당한 양의 데이터를 저장할 수 있고 모바일 사용자 근처에서 계산량이 많은 작업을 처리할 수 있다. 또한 ETSI(European Telecommunications Standards Institute)에서는 MEC ISG(Industry Specification Group)를 출범시켜 MEC에 대한 표준화를 시작했다.

[0004] MEC로 모바일 데이터 서비스나 오프로딩 서비스를 제공할 때, 각 MEC 서버(ES라 함)는 일반적으로 연산, 통신, 저장 용량에 제한이 있다. 따라서 각 ES에서 서로 다른 서비스 유형에 해당하는 서비스 실행 이미지를 모두 배포하는 것은 불가능하며, 따라서 MEC에 대한 서비스 이미지 배포 전략이 광범위하게 연구되어 왔다. 또한, 사용자 이동성(user mobility)은 모바일 네트워크의 엣지 컴퓨팅에서 또 다른 도전적인 문제이다. 모바일 사용자의 위치에 관계없이 사용자와 연결된 BS와 같은 위치에 있는 ES는 모바일 사용자의 서비스 요청을 처리할 책임이 있다. ES에 모든 서비스 실행 이미지를 설치하지 않았기 때문에 담당 ES가 모바일 사용자로부터 수신된 요청에 원하는 서비스를 제공할 수 없을 때 MEC service missing(MSM) 문제가 발생할 수 있다. 이러한 상황에서 담당 ES는 누락된 ES가 되고 원하는 서비스 요청은 '누락된 요청'이 될 수 있다.

[0005] 일반적으로 MSM 문제에 대한 두 가지 가능한 솔루션이 있다. 하나는 누락된 요청이 누락된 ES에서 누락된 요청에 대한 서비스 실행 이미지가 설치된 다른 서비스 ES로 전달되는 서비스 요청 전달이다. 다른 하나는 서비스 실행 이미지가 누락된 ES로 마이그레이션되는 서비스 이미지 마이그레이션이다(T. Kim et al). 일반적으로 마이그레이션 접근 방식에는 WAN을 통한 엄청난 양의 데이터 전송이 포함된다. 예를 들어 얼굴 인식 서비스를 위한 서비스 이미지 마이그레이션은 5Mbps WAN에서 247초가 걸린다(L. Ma et al).

[0006] MSM 문제에 대한 상당한 양의 연구가 있었다.

[0007] T.-D. Nguyen et al 에서 저자는 콘텐츠 중심 네트워킹 기반 MEC 서비스 배포/검색 프로토콜을 제안했다. 제안하는 프로토콜의 동작은 서비스 초기화와 서비스 발견의 두 단계로 이루어진다. 서비스 시작 단계에서 각 ES는 서비스 실행 이미지를 포함하지 않는다. 서비스는 해당 서비스 제공자(SP)에게 요청하기 위해 CC 사이트에 의해 ES에 배치된다. 일정 시간이 지나면 서비스 실행 이미지가 하나 이상의 ES에 배포된다. 그 후, 모바일 사용자가 ES의 커버리지 영역을 방문하는 경우, 각각의 서비스 요청은 로컬 ES, 동일한 지역의 원격 ES 또는 원격지일 수 있는 적절한 ES를 결정하기 위해 탐색 절차에 의해 사전 처리된다. 다른 인접 지역의 ES에서는 ES, 지역 클라우드(RC) 노드 및 코어 클라우드 사이트를 포함하는 3계층 계층 토폴로지에서 MEC 기반 네트워크 모델을 제안한다. ES는 연결된 BS를 통해 모바일 사용자와 통신한다. 2개 이상의 ES가 동일한 지역에 위치하는 경우 공통 RC 노드에 의해 관리된다. 모든 RC 노드는 코어 클라우드 사이트에서 관리한다. 탐색 절차의 ES 결정은 측정된 왕복 시간(RTT)과 요청된 서비스 지연 임계값을 비교하여 QoS 요구 사항을 고려한다. QoS 요구 사항을 충족할 수 있는 ES가 없으면 서비스 요청이 서비스 초기화로 CC 사이트로 전달된다. ES에서 서비스 요청을 실행하기 위해 가상화된 컨테이너는 실행 리소스 단위로 작동하며, 다른 서비스 요청은 서로 다른 실행 리소스 요구 사항(즉, 컨테이너 리소스 크기)을 갖는다. RTT 외에도 실행 자원의 가용성은 QoS 요구 사항을 충족하는 데 중요하지만 여기서는 논의되지 않았다.

- [0008] Y. Sun et al에서 저자들은 MEC 기반 서비스 모델을 위한 두 가지 서비스 제공 방법이 있다고 주장했다(로컬 프로세싱과 MEC 프로세싱). 전자의 경우 데이터는 사용자 장치에서 로컬로 처리되고 처리된 데이터는 ES를 통해 클라우드 사이트의 애플리케이션 서버에 업로드된다. 후자는 처리를 위해 무선 채널을 통해 원시(멀티미디어) 데이터를 ES로 전송한다. 저자는 전체 시스템의 에너지 소비를 최소화하기 위해 멀티미디어 데이터 처리 작업을 할당하고 두 가지 처리 방법 간에 무선 및 계산 리소스를 조정하는 방법에 대한 최적화 문제를 제안했다. 이때, 확률적 최적화 기반 솔루션을 제안했다. 잘 알려진 Lyapunov 최적화 기술을 사용하여 저자는 원래 문제를 작업 할당 결정, 로컬 리소스 할당, 무선 리소스 할당 및 MEC 계산 리소스 할당의 4가지 하위 문제로 분해했다. 이러한 하위 문제는 현재 시스템 정보와 낮은 계산 복잡성으로 해결할 수 있다. 그러나, 본 논문은, 멀티미디어 서비스만을 고려하였고, 시스템 모델에서는 하나의 BS-ES 사이트만 고려되었다. 또한, ES의 제한된 자원으로 인해 오버플로우 문제가 발생할 수 있으며 이는 여기서 다루지지 않았다.
- [0009] J. Zhang et al에서 저자는 지연에 민감한 계산 작업의 요구 사항을 충족하기 위해 계산 오프로딩 및 콘텐츠 캐싱, 스펙트럼 및 계산 리소스 할당을 공동으로 고려했다. 그들의 작업에서 콘텐츠 캐싱은 요청된 콘텐츠가 인터넷에서 ES로 캐싱 될 수 있다는 것이다. 이 경우, 동일한 콘텐츠의 중복 전송을 방지할 수 있다. 본 논문의 목적은 네트워크에서 모든 계산 작업의 총 대기 시간을 최소화하는 것이다. 이때, 결합 문제에 대한 연구는 혼합 정수 비선형 계획법(MINLP) 문제로 공식화된다. 저자는 비대칭 검색 트리를 제안한 다음 분기 및 바인딩 방법을 수정하여 오프로딩 및 캐싱 결정과 리소스 할당 전략 세트를 얻었다. 상술한 Y. Sun et al에서와 같이 최적화 문제는 단일 ES 시스템 모드에서 논의되었다. 콘텐츠 캐싱은 그들의 작업에서 연구되지만 콘텐츠 교체는 고려되지 않았다. 모바일 환경에서 모바일 사용자들이 어떤 서비스 콘텐츠를 요청할지 미리 알기 어렵다. 또한 ES에는 제한된 리소스가 있었다.
- [0010] 모바일 MEC 네트워크에서 사용자 요청은 무작위로 자주 ES에 도착할 수 있다. 따라서 ES 간에는 부하 불균형 문제가 불가피하다. ES의 경우 들어오는 사용자 요청이 너무 많으면 리소스가 곧 소진되어 추가 요청을 처리할 수 없게 된다. 이러한 서비스 중단을 방지하려면 ES 간에 계산 부하의 균형을 맞추는 것이 중요하다. L. Liu et al에서 각 ES는 사용자의 서비스 요청을 저장할 버퍼가 있다고 가정한다. 또한 모든 ES는 부하 공유를 위해 클러스터링되어 하나의 ES가 클러스터 헤드 역할을 한다. 버퍼가 가득 찬 ES에 요청이 도착하면 요청은 클러스터의 특정 ES로 리디렉션된다. 여기서, 사용자 요청을 리디렉션할 ES를 선택하는 방법에는 두 가지가 있다. 하나는 무작위 선택이다. 선택한 ES에 서비스 요청을 저장할 수 있는 버퍼가 있는 경우 요청이 그곳에서 처리된다. 그렇지 않으면 요청이 거부된다. 다른 방법은 클러스터에서 큐 길이가 가장 짧은 ES로 요청을 리디렉션하는 것이다. 리디렉션은 클러스터 헤드에 조회 메시지를 전송하여 수행된다. 그런 다음 클러스터 헤드는 클러스터 구성원에게 요청하여 해당 대기열 상태의 응답을 기다린다. 마지막으로 큐 길이가 가장 짧은 ES가 재지정된 요청을 처리한다. 두 가지 부하 공유 방식의 성능은 평균 대기 시간 및 차단 확률에 대해 2차원 Markov 체인으로 평가되었다.
- [0011] N. Kherraf et al에서 저자들은 사물 인터넷(IoT) 서비스에 대한 MEC 리소스 프로비저닝 및 워크로드 할당(RPWA) 문제를 해결했다. RPWA 문제는 다음 세 가지 작업으로 나뉜다. T-1) MEC Dimensioning: ES의 수와 배치를 결정, T-2) IoT 애플리케이션 배치: ES에 배치할 다양한 유형의 애플리케이션 인스턴스의 수와 배치를 결정, T-3) 워크로드 할당: IoT 장치에서 생성된 서비스 요청을 필요한 응답 시간을 충족할 수 있는 각각의 적절한 ES에 할당한다. 저자는 RPWA 문제를 에지 서버 배포 비용을 최소화하는 혼합 정수 프로그램(RPWA-MIP)으로 공식화했다. 정수 문제는 일반적으로 다루기 어렵기 때문에 저자는 RPWA 문제를 지연 인식 부하 할당과 모바일 에지 서버 차원 문제로 분해할 것을 제안했다.
- [0012] Beraldi et al은 자원의 한계를 극복하기 위해 주변 ES 간의 협력 전략을 제안했다. 제안된 Cooload는 요청 차단(즉, 거부) 확률과 실행 지연을 모두 최소화하는 것을 목표로 하는 부하 공유 방식이다. 즉, 버퍼가 가득 찼을 때 서비스 요청을 받으면 ES는 요청을 가까운 ES로 전달한다. 저자는 네트워크를 유한한 준 출생 및 죽음 Markov 프로세스로 모델링한 다음 제안된 Cooload의 협력 금지 상태에 대한 이점을 연구했다. 그러나 저자는 2개의 ES만 있는 간단한 네트워크를 가정하므로 더 많은 ES가 있는 일반적인 시나리오에는 결과가 적용되지 않을 수 있다. 또한 저자는 두 ES 간의 전송 지연이 무시할 수 있다고 가정했는데 실제 네트워크에서는 그렇지 않을 수 있다.

발명의 내용

해결하려는 과제

[0013] 본 개시에서는 서비스 요청 포워딩 기법을 활용하여 MSM 문제에 대한 해결책으로 모바일 네트워크 사업자(MNO: Mobile Network Operator, 또는 무선 네트워크 제공자)를 위한 동적 MEC 자원 프로비저닝 기법을 제안한다. MNO는 일반적으로 고정된 수의 서비스 유형을 사용자 또는 가입자에게 제안한다. 또한 네트워크에서 서로 다른 ES 집합을 예약하는 소수의 ESM(ES Manager)을 위임하여 복잡성을 줄이기 위해 분산 MEC 관리 메커니즘을 제안한다.

[0014] 본 개시의 목적들은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 본 개시의 다른 목적 및 장점들은 하기의 설명에 의해서 이해될 수 있고, 본 개시의 실시 예에 의해 보다 분명하게 이해될 것이다. 또한, 본 개시의 목적 및 장점들은 특허 청구 범위에 나타난 수단 및 그 조합에 의해 실현될 수 있음을 쉽게 알 수 있을 것이다.

과제의 해결 수단

[0015] 본 개시의 일 실시 예에 따라 복수의 ES(Edge Server)로 구성된 시스템에 포함된 적어도 하나의 ESM(ES Manager)의 제어 방법은, 적어도 하나의 사용자 단말이 상기 복수의 ES 중 적어도 하나에 요청하는 서비스 유형, 상기 복수의 ES 각각에 저장된 실행 이미지에 매칭되는 서비스 유형, 및 상기 복수의 ES 각각의 가용 용량을 실시간으로 모니터링하는 단계, 적어도 하나의 사용자 단말로부터 일 서비스 유형에 대한 요청을 수신한 적어도 하나의 ES가 상기 요청에 매칭되는 서비스 유형의 실행 이미지를 저장하고 있지 않음에 따라 발생하는, 적어도 하나의 누락된 요청을 식별하는 단계, 상기 누락된 요청이 상기 누락된 요청과 매칭되는 서비스 유형의 실행 이미지를 저장하고 있는 적어도 하나의 다른 ES로 리디렉션 되도록 지원하는 단계를 포함한다. 상기 지원하는 단계는, 서비스 유형 각각의 우선 순위 및 복수의 ES 간의 전송 비용을 바탕으로, 상기 누락된 요청과 매칭되는 실행 이미지를 저장하고 있는 적어도 하나의 다른 ES와 관련된 엣지 가중치(edge weight)를 산출하고, 상기 산출된 엣지 가중치를 바탕으로, 상기 누락된 요청이 리디렉션 될 서빙 ES를 선택한다.

발명의 효과

[0016] 본 개시에 따른 제어 방법은, 사용자로부터 받은 요청에 대해 ES가 서비스를 제공할 수 없을 때 발생하는 MEC 서비스 누락 문제를 해결하기 위한 솔루션을 제공하였다.

[0017] 다양한 네트워크 크기와 다양한 요청 도착 시나리오에 대한 평가를 통해 제안된 중앙 집중식 및 분산형 Lagrange 이중 솔루션이 실행 시간이 짧고 보상이 높은 솔루션이 될 수 있음을 보여주었다.

도면의 간단한 설명

[0018] 도 1은 본 개시의 일 실시 예에 따른 시스템의 구성을 설명하기 위한 도면,
 도 2는 본 개시의 일 실시 예에 따라 다양한 서비스 유형의 요청이 각 ES에 수신되면서 누락 ES와 서빙 ES가 정해지는 상황을 설명하기 위한 도면,
 도 3은 본 개시의 일 실시 예에 따른 시스템의 VMK(Variant Multiple-Knapsack) 공식에 따른 수요/공급 정보를 설명하기 위한 도면,
 도 4는 본 개시의 일 실시 예에 따른 아이템-배낭 이분 그래프,
 도 5는 본 개시의 일 실시 예에 따른 아이템-배낭 이분 그래프 상 엣지 가중치 산출 결과를 도시한 도면,
 도 6은 본 개시의 일 실시 예에 따른 누락된 요청의 포워딩 결정 결과를 도시한 표,
 도 7은 본 개시의 일 실시 예에 따라 4개의 ES를 포함하는 소규모 네트워크를 도시한 도면,
 도 8은 본 개시의 일 실시 예에 따라 9개의 ES를 포함하는 소규모 네트워크를 도시한 도면,
 도 9는 본 개시의 일 실시 예에 따른 소규모 네트워크 상에서의 종래 기술과 본 개시에 따른 기술의 성능 비교 결과를 도시한 도면,
 도 10은 본 개시의 일 실시 예에 따라 28개의 ES를 포함하는 대규모 네트워크를 도시한 도면,
 도 11은 본 개시의 일 실시 예에 따른 대규모 네트워크 상에서의 종래 기술과 본 개시에 따른 기술의 성능 비교 결과를 도시한 도면,
 도 12는 본 개시의 일 실시 예에 따른 대규모 네트워크 상에서의 다양한 시나리오에 따른 목적 값의 수렴을 설

명하기 위한 도면, 그리고

도 13은 본 개시의 일 실시 예에 따른 ESM(Edge Server Management)의 구성을 설명하기 위한 블록도이다.

발명을 실시하기 위한 구체적인 내용

- [0019] 본 개시에 대하여 구체적으로 설명하기에 앞서, 본 명세서 및 도면의 기재 방법에 대하여 설명한다.
- [0020] 먼저, 본 명세서 및 청구범위에서 사용되는 용어는 본 개시의 다양한 실시 예들에서의 기능을 고려하여 일반적인 용어들을 선택하였다. 하지만, 이러한 용어들은 당해 기술 분야에 종사하는 기술자의 의도나 법률적 또는 기술적 해석 및 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한, 일부 용어는 출원인이 임의로 선정한 용어도 있다. 이러한 용어에 대해서는 본 명세서에서 정의된 의미로 해석될 수 있으며, 구체적인 용어 정의가 없으면 본 명세서의 전반적인 내용 및 당해 기술 분야의 통상적인 기술 상식을 토대로 해석될 수도 있다.
- [0021] 또한, 본 명세서에 첨부된 각 도면에 기재된 동일한 참조번호 또는 부호는 실질적으로 동일한 기능을 수행하는 부품 또는 구성요소를 나타낸다. 설명 및 이해의 편의를 위해서 서로 다른 실시 예들에서도 동일한 참조번호 또는 부호를 사용하여 설명한다. 즉, 복수의 도면에서 동일한 참조 번호를 가지는 구성요소를 모두 도시되어 있다고 하더라도, 복수의 도면들이 하나의 실시 예를 의미하는 것은 아니다.
- [0022] 또한, 본 명세서 및 청구범위에서는 구성요소들 간의 구별을 위하여 "제1", "제2" 등과 같이 서수를 포함하는 용어가 사용될 수 있다. 이러한 서수는 동일 또는 유사한 구성요소들을 서로 구별하기 위하여 사용하는 것이며 이러한 서수 사용으로 인하여 용어의 의미가 한정 해석되어서는 안 된다. 일 예로, 이러한 서수와 결합된 구성요소는 그 숫자에 의해 사용 순서나 배치 순서 등이 제한되어서는 안 된다. 필요에 따라서는, 각 서수들은 서로 교체되어 사용될 수도 있다.
- [0023] 본 명세서에서 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "구성되다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0024] 본 개시의 실시 예에서 "모듈", "유닛", "부(part)" 등과 같은 용어는 적어도 하나의 기능이나 동작을 수행하는 구성요소를 지칭하기 위한 용어이며, 이러한 구성요소는 하드웨어 또는 소프트웨어로 구현되거나 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다. 또한, 복수의 "모듈", "유닛", "부(part)" 등은 각각이 개별적인 특정한 하드웨어로 구현될 필요가 있는 경우를 제외하고는, 적어도 하나의 모듈이나 칩으로 일체화되어 적어도 하나의 프로세서로 구현될 수 있다.
- [0025] 또한, 본 개시의 실시 예에서, 어떤 부분이 다른 부분과 연결되어 있다고 할 때, 이는 직접적인 연결뿐 아니라, 다른 매체를 통한 간접적인 연결의 경우도 포함한다. 또한, 어떤 부분이 어떤 구성요소를 포함한다는 의미는, 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [0026] 일반적으로 각 ES는 하나 이상의 서비스 유형의 서비스 실행 이미지와 함께 설치되지만 제한된 컴퓨팅 및 스토리지 리소스로 인해 모두 그런 것은 아니다. 따라서, 본 개시에서 고려된 문제의 본질은 각 ES에서 수신된 누락된 요청을 다른 ES로 전달하여 사용자가 요청에 대한 서비스를 받을 수 있도록 하는 것이다. 본 개시에 따라 제안되는 솔루션을 보다 실용적으로 만들기 위해 각 요청의 우선 순위(또는 서비스 품질 요구 사항, QoS)와 네트워크를 통한 전송 지연이 모두 고려된다. 또한, 두 가지를 동시에 고려한 보상 기능을 도입함으로써 MNO가 자신의 선호도에 따라 쉽게 스케줄링 결정을 조정할 수 있다.
- [0027] 구체적으로, 각 ESM은 ES(수요라고 함)에서 수신한 누락된 요청과 ES(공급이라고 함)에서 사용 가능한 서비스 유형 및 리소스를 모두 수집하고 이러한 정보를 사용할 수 있도록 한다.
- [0028] 누락된 요청 할당/포워딩 문제를 최적으로 해결하기 위해 MSM 문제를 VMK(Variant Multiple-Knapsack) 문제로 변환할 수 있다. 배낭 문제(knapsack problem)는 잘 알려진 조합 최적화 문제인 반면 VMK 문제는 여러 배낭과 다른 용량을 가진 고전적인(단일) 배낭 문제의 확장이다. 그러면 누락된 요청의 전달 문제의 VMK 공식이 최적의 솔루션을 찾을 수 있는 잘 알려진 ILP(정수 선형 프로그램)로 변환될 수 있다. 그러나, 고전적인 배낭 문제의 계산 복잡도는 NP-hard이고 VMK 문제는 훨씬 더 어렵다. Relaxation, branch-and-bound/cut과 같은 발전된 형태의 기술이 ILP의 복잡성을 크게 줄일 수 있지만 ILP는 특히 문제 크기가 증가하는 경우 온라인 방식으로 실현

가능하지 않을 수 있다. 복잡성을 크게 줄이기 위한 본 개시의 일 실시 예에 따르면, Lagrangian 이완(LR) 방법과 분해 이론이 활용되어 누락된 요청의 전달 문제를 효율적으로 해결할 수 있다.

[0029] 본 개시의 주요 기여는 다음과 같이 요약된다.

[0030] * 모바일 MEC 네트워크에서 다양한 서비스 유형의 사용자 요청에 서비스를 제공하기 위한 MSM 문제를 연구하고 온라인에서 실행할 수 있는 실용적인 누락 요청 전달 솔루션이 제안된다. 연구된 문제는 다음과 같은 실제적인 가정을 전제로 한다.

[0031] - MNO는 고정된 수의 서비스 유형을 사용자/가입자에게 제공한다.

[0032] - 서비스 유형마다 QoS 수준 또는 우선 순위가 다르다. 각 서비스 유형의 실행 이미지는 하나 또는 일부 ES에 사전 배포된다.

[0033] - ES는 이질적이며 다른 리소스를 가지고 있다.

[0034] - 사용자는 언제 어디서나 연결된 ES에 다양한 유형의 서비스 요청을 보낼 수 있다. 단, ES 서버는 일부 서비스 유형의 실행 이미지를 포함하지 않을 수 있다.

[0035] * 누락된 요청 및 ES 리소스 정보를 수집하기 위한 분산 MEC 관리 메커니즘 및 메시지 구조가 제공된다.

[0036] * 본 개시의 일 실시 예에 따르면, MSM 문제에 대한 해결책으로 VMK 프레임워크를 사용하여 누락된 요청 전달 문제를 공식화한 다음 ILP로 변환한다. 목표는 앞서 전달된 누락된 요청의 우선 순위와 전송 지연을 공동으로 고려하도록 설계되었다. 조정 가능한 가중치 매개변수(weight parameters)를 사용하여 MNO는 QoS(우선순위) 또는 지연에 더 많은 가중치를 둘 수 있다.

[0037] * 계산 복잡도를 줄이고 제안된 솔루션을 온라인 처리에 적합하게 만들기 위해, 저복잡도 LR 기반 스케줄링 알고리즘이 제안된다.

[0038] * 제안된 솔루션의 효율성을 보여주기 위해 ILP의 최적 솔루션과 기존의 휴리스틱 알고리즘을 비교하였다.

[0039] * 제안된 접근 방식의 감소된 복잡성과 수렴성을 보여주기 위해 평가를 수행하고 결과를 분석하였다.

[0040] 이하 표 1은 종래 기술 및/또는 본 개시에서 주로 사용되는 용어/약어를 나타낸 것이다.

표 1

BS	Base Station
CC	Cloud Computing
ES	Edge Server (MEC Server)
ESM	Edge Server (MEC Server) Manager
MEC	Multi-access Edge Computing
QoS	Quality of Service
RAN	Radio Access Network

[0041] 먼저, 가정된 모바일 MEC 시스템의 시스템 모델이 제시된다. 그런 다음, 본 개시에서 고려되는 MSM 문제와 관련된 기존 연구를 선별하여 논의한다.

[0042] 도 1은 본 개시의 일 실시 예에 따른 시스템 모델을 도시한다. MNO는 고객(사용자)을 위해 넓은 지리적 영역에 걸쳐 다양한 서비스 유형의 데이터(오프로딩) 서비스를 제공할 수 있다. 서비스 중인 영역은 여러 무선 커버리지 영역으로 분해되며, 각 영역은 해당 영역을 방문하는 사용자에게 무선 연결을 제공하기 위해 BS와 함께 배포될 수 있다. BS는 또한 BS와 관련된 사용자의 서비스 요청을 처리하는 ES를 갖추고 있다. 하나의 ES에 여러 BS가 연결될 수 있지만, 편의상 BS와 ES 간의 일대일 관계를 가정한다. 기지국과 해당 기지국은 WAN(Wide Area Network)으로 연결될 수 있다.

[0043] 본 개시는 MSM 문제에 대한 동적 누락 요청 전달 솔루션을 제시한다. 제안된 방법은 서비스 리디렉션 기술을 기반으로 하며 총 리디렉션 비용을 최소화하면서 실제 서비스되는 누락 요청의 수를 최대화하는 것을 목표로 한다. 두 가지 목표를 함께 최적화하기 위해 두 요소를 반영하는 배치 값을 사용하여 변형 VMK(다중 배낭) 문제를 공식화한다. VMK 문제는 잘 알려진 배낭 문제를 NP-hard 시간 복잡도로 확장한 것이다. 높은 계산 오버헤드를 피하기 위해 Lagrangian Relaxation(LR) 방법을 활용하여 완화된 반복 알고리즘이 활용된다.

- [0045] 본 개시에 따라 제안하는 기법의 기본 개념을 설명하기 전에 먼저 다음과 같은 정의를 소개한다.
- [0046] 정의 1: 서비스 유형 y_i 및 해당 실행 이미지. 모바일 MEC 시스템은 모바일 사용자가 사용할 수 있는 고정된 수의 다양한 서비스 유형을 제공한다. 각 서비스 유형은 일부 ES에 미리 설치되어 있는 해당 실행 이미지로만 처리할 수 있다.
- [0047] 정의 2: 누락된 ES 및 서비스 유형 y_i 의 누락된 요청. 리소스 제한으로 인해 각 ES는 몇 개의 실행 이미지와 함께 설치된다. BS와 그 co-located ES의 커버리지 영역 내의 모바일 사용자가 y_i 의 서비스 요청을 전송할 때, ES는 서비스 유형의 실행 이미지를 갖지 않을 수 있다. 이러한 경우 ES를 Missing(누락) ES라 하고 서비스 요청을 y_i 에 대한 Missed Request(누락된 요청)라 한다.
- [0048] 정의 3: 서비스 유형 y_i 의 서비스 ES. ES의 경우 요청된 서비스 유형 y_i 의 실행 이미지가 있는 경우 y_i 에 대해 서빙 ES라고 칭한다. 누락된 서비스 수신 요청의 경우 y_i 의 누락된 요청은 누락된 ES에서 이러한 ES로 리디렉션될 수 있다.
- [0049] 정의 4: ES 관리자, ESM. ESM은 누락된 요청(수요라고 함)과 설치된 서비스 실행 이미지 및 ES의 가용 용량(공급이라고 함) 정보를 제안된 메시지 형식(참조: 도 3 참조)을 사용하여 일정 간격으로 수집하는 역할을 할 수 있다.
- [0050] 정의 5: MEC 서비스 누락(MSM) 문제. 누락된 EC는 해당 범위 내에서 모바일 사용자로부터 수신된 특정 요청에 서비스를 제공할 수 없다. 이를 MSM 문제라고 하며 사용자에게 서비스를 제공/보장하기 위해서는 이러한 요청에 대한 포워딩 전략이 필요하다.
- [0051] 정의 6: 가변 다중 배낭(VMK) 문제. 항목 세트와 하나 이상의 배낭을 가정한다. 아이템은 배낭에 넣을 수 있지만 배치 제약이 있을 수 있다. 목표는 배낭 내부에 배치된 항목 인스턴스의 총 가치를 최대화하는 것이다. VMK 문제는 잘 알려진 단일 배낭 문제의 확장으로 다음과 같은 특징을 갖는다.
- [0052] * 용량이 다른 여러 배낭이 있다.
- [0053] * 각 항목은 특정 수의 인스턴스 상에 존재한다. 일 항목에 있어서, 해당 항목의 인스턴스들은 서로 다른 배낭에 넣을 수 있다. 동일한 항목 유형에 속하는 인스턴스는 크기와 값이 동일하다.
- [0054] * 충분한 공간이 있더라도 일 항목의 인스턴스를 일부 배낭에 넣을 수 없는 것과 같은 배치 제약이 있을 수 있다.
- [0055] 도 2는 앞서 언급한 정의를 예시하기 위한 소규모 모바일 MEC 네트워크를 나타낸다. 도 2의 (a)와 같이 MEC 시스템은 m_1 부터 m_6 까지 6개의 ES로 구성되며 y_1 , y_2 , y_3 의 3가지 서비스 유형을 제공한다. m_1 , m_2 , m_3 가 ESM으로 지정되었다고 가정한다. 각 서비스 유형의 누락 ES 및 서빙 ES는 도 2의 (b)에 나열되어 있다. 서비스 유형 y_1 의 경우 m_1 , m_3 및 m_4 에서 누락된 요청이 발생했다. 이러한 ES는 y_1 에 대한 실행 이미지와 함께 사전 설치되지 않았기 때문이다. 도 2의 (b)에서 누락된 요청의 수와 사용 가능한 자원 단위의 수(즉, 각 요청의 크기)는 임의로 선택됨을 유의할 필요가 있다. m_2 , m_5 , m_6 는 y_1 에 대한 실행 이미지가 있으므로 y_1 의 서빙 ES 역할을 할 수 있다. 각 ESM은 ES 집합을 관리한다. 위의 누락된 요청 정보 및 사용 가능한 자원 단위 정보는 해당 ESM에서 수집되며, (missing ES ID, 누락된 요청 수, 각 요청을 처리하는 데 필요한 자원 단위 수) 및 (serving ES ID, 사용 가능한 리소스 단위 수), 등과 같은 폼으로 관리된다. 두 정보 집합은 각각 MSM 문제의 수요 및 공급 정보로 간주된다. ESM은 서비스 유형의 누락/제공 ES일 수도 있다. 예를 들어, m_1 은 y_1 의 ESM이고 y_1 의 누락된 ES이며 y_3 의 서빙 ES일 수 있다.
- [0056] 요컨대, 본 개시에 따라 제안된 아이디어의 전체적인 절차는 다음 5단계로 구성된다. 첫째, 각 ESM은 자신의 통제 하에 있는 ES로부터 수요 및 공급 정보를 수집한다(1단계). ESM은 제안된 품목 및 배낭 데이터 구조를 사용하여 수요 및 공급 정보의 일부 또는 전체를 서로 교환한다(2단계). 다음으로 VMK 구조를 사용하여 누락된 요청 전달 문제를 공식화한다(3단계). 그런 다음 VMK 문제는 ILP로 변환된다(4단계). 마지막으로 누락된 요청 전달 일정은 ILP를 직접 해결하거나 Lagrangian Relaxation(5단계)을 사용하여 찾을 수 있다.
- [0057] 이하에서는 본 개시에 따라 제안된 시스템의 세부적인 동작을 자세히 설명할 것이다. 제안된 방식의 운영을 지원하기 위해 제안된 4가지 모듈이 있다. 1) VMK 데이터 구조 형성, 2) MSM과 VMK 간의 문제 변환, 3) VMK의 ILP 모델 공식화, 4) 누락된 요청의 할당 또는 전달 스케줄링.
- [0058] 1) 데이터 구조 형성: 앞서 언급한 바와 같이 서비스 유형의 수요 및 공급 정보는 ESM에서 수집하는 각각의 서

빙 ES에서 누락된 요청 및 가용 자원이다. VMK의 경우 수요 및 공급 정보는 각각 항목에 대한 인스턴스 수(즉, 서비스 유형의 누락된 요청) 및 배낭 용량(즉, ES 제공)이 된다. 누락된 요청의 전달 문제를 VMK로 쉽게 변환하기 위해 수요 및 공급 정보를 통합하는 두 가지 데이터 구조가 각각 제안된다. Item(항목) 및 knapsack(배낭)은 각각 누락된 요청 및 ES를 나타낸다.

[0059] 항목 및 배낭에 대한 데이터 구조는 도 3의 (a)에 도시된다. ESM 중 하나는 ESM 리더로 지정될 수 있으며 ESM 간의 정보 수집 절차를 관리할 수 있다. 도 2의 (b)에 도시된 바와 같이, i1부터 i9까지 3개의 관리자 ES에 의해 수집된 9개의 수요 정보 레코드가 있다. 첫 번째 항목 i1은 서비스 유형 y1의 누락된 요청에 해당하고 이 항목의 해당 누락 ES는 m1이다. 인스턴스 수는 3이며, 이는 동일한 ES에서 동일한 유형의 누락된 요청이 3개 있음을 의미한다. 이 항목의 인스턴스 크기는 1이며 각 y1 유형 요청을 처리하는 데 필요한 리소스 단위의 수를 나타낸다.

[0060] 공급 정보 통합을 위해 하나의 배낭은 하나의 ES에 해당한다. 6개의 서버 ES가 있는 도 2의 경우, 3개의 관리자 ES에 의해 수집된 6개의 공급 정보 레코드가 있다. 따라서 6개의 배낭이 도 3의 (b)와 같이 배낭 자료 구조로 형성된다. 첫 번째 배낭은 서비스 유형 y1 및 y2의 누락된 요청에 서비스를 제공하기 위해 리소스를 사용할 수 있는 서버 ES m2에 해당한다. 이 배낭의 해당 서버 ES는 m2이다. 서비스 유형 프로비저닝(즉, 설치된 실행 이미지)은 {y1, y2}이다. m2에 5개의 사용 가능한 자원 단위가 있는 경우 용량 크기는 5이다. 도 3의 (b)에서 항목 및 배낭에 지정된 특정 숫자는 임의로 선택될 수 있다.

[0061] 2) VMK 문제 공식화: 위의 항목 및 배낭 데이터 구조는 MSM 문제를 해결하기 위해 VMK 문제에 대한 입력으로 사용할 수 있다. 이를 위해 다음 정의를 사용하여 VMK 문제에서 항목(누락된 요청)과 배낭(ES 제공) 간의 배치 관계를 나타내기 위해 BG라고 하는 항목 배낭 이분 그래프가 형성된다.

[0062] 정의 7: 이분 그래프 $BG = (I, K, E)$ 는 노드가 항목과 배낭을 각각 나타내는 두 개의 분리된 집합 I 및 K로 분할되는 그래프이다. 각 에지 $e \in E$ 는 다음 수집 가능 조건이 충족되면 노드 $i \in I$ 를 노드 $k \in K$ 에 연결한다.

[0063] 정의 8: i와 k 사이의 수집 가능한 조건. BG에서 i와 k 사이에 에지가 존재하는 경우 해당 항목 i는 처리를 위해 서버 ES인 k로 전달된다. 이러한 경우 i는 k의 수집 가능한 항목(collectable item)이라고도 지칭할 수 있다.

[0064] 정의 9: 에지 $(i, k) \in BG$ 에 대해 연관된 엣지 가중치는 i와 k 사이의 배치 값 v_{ik} 를 나타내며 다음과 같이 결정될 수 있다.

수학적식 1

$$v_{ik} = f_{pri}(pri(i)) \cdot w_1 + f_{trans}(trans(i, k)) \cdot w_2, \quad (1)$$

[0066] 여기서 f_{pri} 및 f_{trans} 는 정규화 함수이고 각각은 가중치 w_1 및 w_2 와 연관된다. 함수 $f_{pri}(\cdot)$ 는 i의 관련 서비스 유형의 우선 순위 $pri(i)$ 를 $(pri(i) - pri_{min}) / (pri_{max} - pri_{min})$ 으로 정규화한다. 여기서 pri_{max} 및 pri_{min} 은 모든 서비스 유형 중 최대 및 최소 우선 순위 값에 해당한다. $pri(i)$ 는 배치 값에 긍정적인 기여를 하며 높은 우선 순위는 i의 연관된 서비스 유형이 더 중요함을 나타낸다. 유사하게, $f_{trans}(\cdot)$ 는 i의 대응하는 누락 ES와 대응하는 서버 ES인 k 사이의 전송 비용 $trans(i, k)$ 를 $(trans_{max} - trans(i, k)) / (trans_{max} - trans_{min})$ 로 정규화할 수 있다. 여기서 $trans_{max}$ 및 $trans_{min}$ 은 각각 두 ES 간의 최대 및 최소 전송 비용에 해당한다. 전송 비용은 배치 값에 부정적인 기여를 할 수 있다. 예를 들어 전송 비용이 높으면 배치 값이 낮아진다. 따라서 $trans_{max} - trans(i, k)$ 를 사용하여 배치 값에 대한 전송 비용의 기여도를 나타낼 수 있다. 일반적으로 전송 비용은 전송 지연으로 간주된다. $w_1 + w_2 = 1$ 인 매개변수 $w_1, w_2 \in [0, 1]$ 은 각각 우선 순위 및 전송 비용과 관련된 기여 비율 또는 가중치에 해당한다.

[0067] 각 엣지 가중치(edge weight) v_{ik} 는 k가 i의 서버 ES이면 $[0, 1]$ 값을 취하거나 그렇지 않으면 N/A를 취한다. 이하 평가에서는 edge weight 값을 다음과 같이 수정하고 0이 되는 것을 방지하기 위해 유효한 배치 값에 ϵ 를 더한다.

수학식 2

$$v_{ik} = \begin{cases} -\infty, & \text{if } v_{ik} \text{ is N/A,} \\ v_{ik} + \epsilon, & \text{otherwise,} \end{cases} \quad (2)$$

여기서 ϵ 는 이 작업에서 1로 설정되지만 강력한 양수일 수 있다.

도 3의 (b)의 주어진 항목과 배낭 정보 및 위의 정의를 사용하여 항목-배낭 이분 그래프는 도 4와 같이 형성될 수 있다. 여기서, 항목 i_1 은 배낭 k_1, k_2 및 k_3 와 수집 가능한 관계를 갖는다. 도 3의 (b)에서 볼 수 있듯이 세 배낭의 해당 ES(m_2, m_5, m_6)는 i_1 의 연관된 서비스 유형(y_1)의 서비스 이미지를 포함한다. 따라서 i_1 은 각각 k_1, k_2 및 k_3 에 대해 3개의 간선을 갖는다. 계산된 에지 가중치는 도 2에 표시된 소형 MEC 시스템에서 추출한 매개변수 값을 사용하여 도 5에 제공될 수 있다. ϵ 는 상술한 수학식에 따라 N/A가 아닌 값에 추가된다. 최소값 및 최대값은 우선순위 및 전송 비용 매개변수를 정규화 하는 데에 사용될 수 있다. 정규화된 값은 다른 비율로 통합되어 항목-배낭 쌍의 배치 값(엣지 가중치)이 된다. 도 5에서는 전송 비용보다 우선순위(QoS)가 더 중요하도록 $w_1 = 0.9, w_2 = 0.1$ 로 설정하였다. w_1 과 w_2 도 다르게 설정할 수 있지만 그 합은 1이어야 한다. (i_1, k_1) 및 (i_4, k_4)의 엣지 가중치 값은 각각 다음과 같다.

$$0.9 \times \frac{1-1}{3-1} + 0.1 \times \frac{5-5}{5-3} = 0 \text{ and } 0.9 \times \frac{2-1}{3-1} + 0.1 \times \frac{5-4}{5-3} = 0.5$$

항목이 배낭의 수집 가능한 항목이 아닌 경우 해당 가장자리 가중치는 N/A로 설정될 수 있다(예: (i_1, k_4)).

3) ILP 공식화: ILP 프레임워크(A. Schrijver, K. Mehlhorn et al)를 사용하여 VMK의 최적 솔루션을 얻을 수 있다. 고전적인(단일) 배낭 문제의 ILP 모델은 종래의 문헌들(ex. S. Dasgupta et al)에 나와 있다. 단일 배낭 문제와 달리 VMK 문제에는 여러 배낭, 각 항목당 하나 이상의 인스턴스 종류, 배치 제약이라는 세 가지 별개의 지점이 있다. 단일 배낭 문제의 ILP 모델을 다음과 같이 VMK 문제 해결에 적용할 수 있도록 확장할 수 있다(P. 3이라고 함).

수학식 3

$$\max_{\mathbf{x}} \sum_{i \in I} \sum_{k \in K} x_{ik} \times v_{ik} \quad (3a)$$

subject to :

$$\forall k \in K : \sum_{i \in I} x_{ik} \times r_i \leq \pi_k, \quad (3b)$$

$$\forall i \in I : \sum_{k \in K} x_{ik} \leq s_i, \quad (3c)$$

$$\forall i \in I, \forall k \in K : x_{ik} \geq 0, \text{ and integer.} \quad (3d)$$

목표에 해당하는 Eq. 3a는 포워딩 할 모든 아이템 인스턴스의 총 배치 값(즉, edge weight)을 최대화하는 것으로 이를 reward라고 한다. 각 항목에는 하나 이상의 인스턴스가 있다. 결정 변수 $\mathbf{x} \in \mathbf{Z}^{|I| \times |K|}$ 는 x_{ik} 에 의해 배낭 k 에 넣을(즉, ES k 로 전달됨) 항목 i 의 인스턴스 수를 결정한다. 0에서 s_i 항목의 총 인스턴스 수인 s_i 까지의 값을 취할 수 있다(Eq. 3d 참조). v_{ik} 는 Eq 1과 같이 (i, k) 쌍을 허용하기 위한 배치 값(또는 보상)에 해당한다. 수학식 3b 및 수학식 3c는 품목 및 배낭 제약 조건이다. 각 항목 i 에 대해 배치된 인스턴스의 수는 총 인스턴스 수 s_i 보다 클 수 없다. 항목의 인스턴스를 다른 배낭에 넣을 수 있다. 배낭 예산 제약(Eq. 3b)은 과밀 상황을 방지하고 배낭 k 에 배치되는 항목 인스턴스의 총량은 배낭 용량 π_k 를 초과할 수 없다. r_i 는 항목 i 의 인스턴스를 처리하는 데 필요한 자원 단위의 수이다.

ILP 모델 P.3을 사용하여 도 4의 bipartite item-knapsack 그래프의 최적 솔루션은 $I = \{1, 2, \dots, 9\}$, $K = \{1, 2, \dots, 6\}$, $s = \{s_1, \dots, s_9\} = \{3, 4, 4, 3, 1, 5, 4, 5, 2\}$, $\pi = \{\pi_1, \dots, \pi_6\} = \{5, 7, 1, 13, 9, 6\}$ 이고 edge weight는 도 5의 (b)로 설정되도록 얻어질 수 있다. 결정 변수인 $\mathbf{x}$ 는 9개 항목 및 6개의 배낭 사이의 최적 배치를 결정할 수 있다. 결과는 다음과 같다. $x_{12} = 2, x_{13} = 1, x_{22} = 4, x_{31} = 4, x_{45} = 3, x_{55} = 1, x_{64} = 1, x_{76} = 1, x_{84} = 5, x_{96} = 2$, 나머지는 $x_{ij} = 0$ 에 해당한다. 항목 1의 경우 2개는 배낭 k_2 , 1개

는 k3에 입력되었다. 배낭에 넣을 수 있는 아이템 인스턴스의 총 개수는 24개이다. 아이템 6과 7에는 놓지 말아야 할 인스턴스가 각각 4개와 3개가 있다. 모든 배낭의 총 용량은 41개이다. 그러나 모든 항목 인스턴스에 필요한 총 크기는 51단위에 해당한다. 배낭 1, 2, 4, 5는 모두 배치 후 1 단위가 남아 있지만 두 배낭을 항목 6 및 7의 인스턴스를 수용하는 데 사용할 수 없다. 이는 항목 6 또는 7의 인스턴스 크기가 2단위이기 때문이다(도 3 참조).

[0077] 4) 솔루션: 누락된 요청 전달 결정: ILP 공식에서 결정 변수 $x_{ik} \in x$ 의 값은 VMK 문제에 대한 최적의 항목 배치를 결정할 수 있다. 앞서 언급한 바와 같이 item 및 knapsack 데이터 구조는 이미 항목과 누락된 ES 간의 관계는 물론 배낭과 서빙 ES 간의 관계를 유지하고 있다(도 3 참조). 따라서 ILP의 최적의 배낭 배치는 도 6과 같이 누락된 요청의 전달 결정으로 쉽게 변환될 수 있다. 예를 들어, 도 3에서 항목 $i=1((i1))$ 은 누락된 ES $m=1(m1)$ 에 해당하고 배낭 $k=2(k2)$ 는 서빙 ES인 $m=2(m2)$ 에 해당한다. 그러면 $x_{11} = q$ 가, 누락된 ES인 $m = 1$ 의 q 개의 누락된 요청이 서빙 ES인 $m = 2$ 로 전달되어야 함을 나타냄을 쉽게 확인할 수 있다. 도 6과 같이 MSM 솔루션은 누락된 요청에 대한 전달 결정을 성공적으로 수행한다.

[0078] 5) Lagrangian relaxation and decomposition: P.3에서 제안한 ILP는 일반적으로 난해하며 네트워크 크기 및 누락된 요청의 수에 따라 복잡도가 급격히 증가한다. CPU 시간을 줄이고 솔루션을 실시간으로 제공하기 위해, 본 개시는 Lagrangian 이완 및 분해 이론을 사용하여 원래 ILP P. 3을 낮은 복잡한 하위 문제로 분해할 것을 제안한다. 첫 번째 단계는 P. 3을 병렬로 해결할 수 있는 낮은 복잡한 ES별 문제로 분해하는 것이다. 그런 다음 더 적은 수의 ESM별 문제로 그룹화된다. 그러나, Eq.3c의 결합 제약으로 인해 P. 3을 ES/배낭별 하위 문제로 분해할 수 없다. 따라서, 해당 제약 조건을 완화하면 다음과 같이 Lagrangian을 형성할 수 있다.

수학식 4

$$\max_{\mathbf{x}} \sum_i \sum_k x_{ik} \times v_{ik} - \sum_i \lambda_i (\sum_k x_{ik} - s_i) \quad (4a)$$

subject to :

Eq. 3b and 3d,

$$\forall i: \lambda_i \geq 0, \quad (4b)$$

[0079]

여기서, $\lambda \in \mathbf{R}_+^{|I|}$ 이며, 상술한 Eq.4a를 통해 이하 수학식이 획득될 수 있다.

수학식 5

$$L(\mathbf{x}; \lambda) = \sum_k \sum_i (v_{ik} - \lambda_i) x_{ik} + \sum_i \lambda_i s_i. \quad (5a)$$

[0081]

또한, 이하 수학식이 획득될 수 있다.

수학식 6

$$L_k(\mathbf{x}_k; \lambda) = \sum_i (v_{ik} - \lambda_i) x_{ik}, \quad (6)$$

[0083]

여기서, $\mathbf{x}_k \in \mathbf{Z}^{|I|}$ 이다.

[0084]

[0085] 상술한 수학식 6과 같이 각 배낭 또는 ESk에 대해, v_{ik} 에서 λ_i 가 차감된 결과가 i 를 취급하는 가격으로 해석될 수 있다. λ_i 의 값은 아래와 같이 반복적으로 업데이트되며, 이를 통해 이 분산 솔루션은 효과적인 차선책 솔루션을 생성할 수 있다.

[0086] 커플링 제약이 완화되어 P.4의 전체 문제는 두 가지 수준의 문제로 분해될 수 있다. 바로, 하나의 마스터 문제(즉, 상위 수준 문제)와 k 개의 하위 문제(즉, 하위 수준 문제)이다. 낮은 수준에서 각 배낭 k 는 주어진 λ 에 대해 다음 ILP를 해결한다(P. 7이라고 함).

수학식 7

$$\begin{aligned} \text{For knapsack } k: \\ \mathbf{x}_k^*(\lambda) &= \arg \max_{\mathbf{x}_k} L_k(\mathbf{x}_k; \lambda) \quad (7a) \\ \sum_i x_{ik} \times r_i &\leq \pi_k \quad (7b) \\ \forall i: x_{ik} &\geq 0, \text{ integer.} \quad (7c) \end{aligned}$$

그런 다음 마스터 이중 문제는 다음을 해결한다(P. 8이라고 함).

수학식 8

$$\min_{\lambda} g(\lambda) = \sum_k g_k(\lambda) + \sum_i \lambda_i s_i \quad (8a)$$

$$\text{subject to:} \\ \forall i: \lambda_i \geq 0. \quad (8b)$$

여기서, $g_k(\lambda) = L_k(\mathbf{x}_k^*, \lambda)$.

하위 문제 P. 7은 전통적인 단일 배낭 문제이며 잘 알려진 동적 프로그래밍 방식은 $O(S \times T)$ 에서 최적의 솔루션을 효율적으로 찾을 수 있다. 여기서 S는 요소의 수이고 T는 배낭의 용량에 해당한다. 마스터 문제는 λ 에 대한 다음 업데이트 규칙을 사용하여 하위 그래디언트 방법으로 효율적으로 해결할 수 있다.

수학식 9

$$\lambda_i^{t+1} = [\lambda_i^t - \alpha^t (s_i - \sum_k x_{ik}^*(\lambda^t))]^+, \quad (9)$$

여기서 t는 반복 카운터, α 는 단계 크기, $[\cdot]^+ = \max(0, \cdot)$ 이다. 예를 들어, $\alpha^t = \frac{1+m}{t+m}$ for $m \in \mathbb{R}_+$ (D. P. Palomar et al)와 같이 단계 크기 α 를 업데이트하는 다양한 방법이 있지만, 실제로 효과가 입증된 다음 규칙(: M. L. Fisher)이 이 작업에서 사용된다.

수학식 10

$$\alpha^t = \frac{\beta^t (g(\lambda^t) - f^*)}{\sum_i \|e_i\|^2}, \quad (10)$$

여기서 $\beta^t \in (0, 2]$ 이고, f^* 는 원래 문제의 가장 잘 알려진 목적 값이며, $e_i = s_i - \sum_k x_{ik}^*(\lambda^t)$ 이다. 일반적으로 β^t 는 처음에는 2로 설정하고, 몇 번 반복해도 $g(\lambda^t)$ 가 변하지 않으면 2로 나눈다. 업데이트 규칙에 해당하는 수학식 10의 수렴에 대한 연구는 M. Held et al의 논문에 개시되어 있다.

공식 P. 7 및 P. 8은 네트워크의 모든 ES 또는 배낭이 자체 솔루션을 계산하기 때문에 완전 분산(FD라고 함) 접근 방식을 나타낸다. 그러나, 이는 다음 두 가지 이유로 바람직한 접근 방식이 아니다. 첫째, FD는 λ 를 업데이트하기 위해 많은 메시지 교환을 발생시킨다. 둘째, FD는 로컬 탐욕 알고리즘(local greedy algorithm)과 동일하며 솔루션을 계산할 때 다른 ES를 고려하지 않기 때문에 불가능하거나(예: 실제로 있는 것보다 더 많은 항목을 전달하려고 시도하는) 비효율적인 솔루션을 산출할 가능성이 있다.

앞서 언급했듯이, 본 개시에 따라 가정된 네트워크에는 ESM이 있으며 각 ESM이 전체 솔루션의 하위 집합을 분산 병렬 방식으로 해결하는 방식으로 문제를 분할할 것이 제안된다. H를 $h \in H$ 및 $|H| \leq |K|$ 인 네트워크에서 ESM의 인덱스 집합이라고 가정한다. 여기서 $|K|$ 는 네트워크에 있는 ES의 수이다. $\eta(h)$ 를 ESM h의 제어 하에 있는

ES의 인덱스 세트라고 할 수 있다. FD를 ESM별 문제로 다시 공식화하기 위해 수학적 6을 다음과 같이 다시 작성할 수 있다.

수학적 11

$$L_{\eta(h)}(\mathbf{x}_{k \in \eta(h)}; \lambda) = \sum_{k \in \eta(h)} \sum_i (v_{ik} - \lambda_i) x_{ik}. \quad (11)$$

[0099]

[0100]

마찬가지로, P. 7과 P. 8은, ES별 문제 그룹이 하나로 결합되어 |H|개의 낮은 문제가 도출되도록 다시 작성될 수 있다.

[0101]

Lagrangian 완화 기반 접근 방식(이하 LR이라고 함)은 최적성 격차가 0인 일부 경우가 있지만 차선택 솔루션을 생성한다.

[0102]

C. 매니저 선정

[0103]

앞서 언급했듯이 ESM은 자신이 통제하는 ES 집합에서 수요 및 공급 정보를 모두 수집하는 역할을 한다. ESM은 임의로 선택할 수 있지만 선택된 ES에서 나머지 ES로의 평균 전송 지연이 최소화되는 ES를 선택할 필요가 있다. 다시, |H|는 MNO에 의해 결정된 ESM의 수이고 |K|는 |H| ≤ |K|인 ES의 수이다. 일단 ESM이 선택되면 ESM의 제어 하에 있는 ES의 집합은 ES와 ESM 간의 가장 짧은 전송 지연에 의해 결정될 수 있다.

[0104]

D = [d_{ij}]를 인접 ES 간의 1홉 전송 지연 행렬이라고 하고, 여기서 두 인접 ES i 및 j 간의 지연은 d_{ij}로 표시된다. Dijkstra의 최단 경로 알고리즘을 사용하여 소스 ES인 i에서 대상 ES인 j까지의 최단 경로 지연을 나타내는 행렬 C = [c_{ij}]를 계산할 수 있다. i번째 행 c_{i*}는 ES i에서 네트워크의 나머지 ES까지의 지연을 나타내며 c_{ii}는 0이다. ES i에서 나머지까지의 평균 지연은 다음과 같다. $\bar{c}_i := \sum_j c_{ij} / (|K| - 1)$

[0105]

마지막으로 제안되는 방법은 c_i, ∀i 중 1번째, 2번째, . . . , |H|번째로 작은 값을 선택하여 해당 ES가 ESM이 되도록 하는 것이다. 관리자의 수가 1인 경우 LR은 LR(c) 또는 중앙 집중식 LR이라고 하는 ILP의 Lagrange dual이 된다. 반면에 관리자가 두 명 이상인 경우 이를 LR(d) 또는 분산 LR이라고 한다. 평가에서는 인덱스가 가장 작은 ESM이 마스터 이중 문제, 즉 λ 업데이트를 담당한다고 가정했다. 관리자가 한 명 뿐인 경우 하위 및 마스터 문제를 모두 처리한다.

[0106]

제안한 방법의 성능을 평가하기 위해 Matlab[26], CVX[27], MOSEK[28]를 사용하여 시뮬레이션 환경과 제안한 방법을 구현하였다. Matlab은 네트워크를 구성 및 에뮬레이트하고 사용자로부터 서비스 요청을 생성하는 데 사용된다. CVX는 제안된 최적화 문제를 모델링/공식화하는 데 사용되며 MOSEK 솔버는 최적 솔루션을 계산한다. MOSEK는 혼합 정수 프로그램을 포함한 다양한 최적화 문제를 해결한다. Intel(R) Core(TM) i9-10940X CPU 및 128GB RAM이 장착된 고성능 워크스테이션은 시뮬레이션 및 성능 평가에 사용되었다.

[0107]

A. 구성

[0108]

교차점에 위치한 R × R 라우터로 구성된 R-by-R 그리드 네트워크를 가정한다. 두 개의 인접한 라우터를 연결하는 각 링크는 Uniform[1, 2]의 단위 없는 임의의 지연을 가진다. 물리적 매체의 특성을 반영하거나 실제 네트워크에서 직접 획득한 지연은 수정 없이 사용될 수 있다. ES의 수 |K| 및 이들의 네트워크상의 위치는 무작위로 선택된다. ES가 하나의 라우터에 연결되어 있고 연결된 라우터에 대한 지연이 0에 가깝다고 가정한다. 네트워크에서 지원되는 서비스 유형 {1, 2, . . . , Qtype} 목록은 모든 ES와 사용자에게 알려져 있으며, 각 ES에는 무작위로 선택된 서비스 실행 이미지 세트가 설치된다. 각 사용자는 하나의 BS/ES와 연결되고 무작위로 선택된 서비스 유형과 연결된 임의의 수의 요청을 보낸다. 사용자 요청의 도착은 각 사용자에게 대한 Poisson 분포 Poi(Uniform[0, Qrate])를 따른다. 각 서비스 유형(QoS)의 우선 순위는 {1, 2, . . . , Qpri}에서 무작위로 선택된다.

[0109]

각 서비스 유형의 요청 인스턴스 크기는 {1, 2, . . . , Qsz}에서 무작위로 선택될 수 있다. 각 ES의 사용 가능한 용량은 {1, 2, . . . , Qcapa}에서 무작위로 선택된다. 다음 하위 섹션에서는 다양한 네트워크 시나리오를 고려하며 각 시나리오에 사용할 구성 값은 각 하위 섹션의 시작 부분에 지정된다.

[0110]

구현해야 할 접근 방식은 6가지가 있으며 다음과 같이 서로 비교될 수 있다. ILP(c)와 LR(c,d)는 본 개시에 따라 제안된 것이고 나머지는 성능 비교를 위한 것이다. (c)와 (d)는 각각 중앙 집중식과 분산형을 나타낸다. 중

양 집중화는 글로벌 정보를 가진 단일 엔티티가 글로벌 결정을 내리는 경우이고 분산은 제한된 정보를 가진 소수의 엔티티가 동시에 개별적으로 로컬 결정을 내리는 경우이다. 분명히 중앙 집중식 접근 방식은 분산 방식에 비해 더 나은 솔루션을 제공한다.

[0111] 그러나 최적화 문제를 해결해야 하는 ILP(c)와 LR(c) 모두 계산 복잡도가 높을 수 있다. 또한 중앙 집중식 접근 방식은 설치된 서비스 이미지와 모든 ES의 가용 용량을 포함한 글로벌 지식을 필요로 하므로 최적화 문제를 해결할 준비를 하기 위해 대량의 빈번한 데이터 전송이 발생할 수 있다. 반면에 LR(d)는 ESM에 의해 병렬로 문제의 파티션을 해결할 수 있다. 감소된 문제 크기와 병렬 처리는 계산 시간을 줄이는 데 도움이 된다. 각 반복 후에 LR(d)는 하위 문제와 λ 의 최적 솔루션을 모두 교환해야 하지만 크기는 작다.

[0112] 1. ILP(c): 본 개시에 따라 제안되는 정수 프로그래밍 기반의 최적 방법

[0113] 2. LR(c), LR(d): Lagrangian 완화 및 문제 분해를 사용하여 본 개시에 따라 제안된 차선책 솔루션. 이중 변수 λ 는 처음에 Uniform[0, 1]로 무작위로 초기화됨. LR은 이중 변수를 반복적으로 업데이트하며 제한된 반복 횟수 동안 실행되도록 구성되어 있음.

[0114] 3. RND(c), RND(d): 요청을 처리할 수 있는 경우 선택한 누락된 요청을 무작위로 선택한 ES에 할당하는 무작위 접근 방식(알고리즘 1).

[0115] 4. GRD(c), GRD(d): 가장 높은 배치 값을 산출하는 누락된 요청을 먼저 처리하는 탐욕적(Greedy) 접근(알고리즘 2).

[0116] ILP, LR, GRD는 우선 순위와 전송 비용의 균형을 유지하기 위해 가중치를 $w1 = 0.5$, $w2 = 0.5$ 로 설정하고 RND는 작동 중 가중치와 관련이 없다. 단, 총 배치 금액(보상)을 비교할 때 RND는 동일한 가중치를 적용한다. 다음 내용에서 각 하위 섹션은 ES의 특정 세트 및 배치가 있는 특정 네트워크 레이아웃을 가정한다. 그런 다음, 10가지 다른 시나리오, 즉 사용자로부터 10가지 다른 요청 도착으로 10번 평가가 수행될 수 있다.

Algorithm 1 RND algorithm description

```

1:  $X \leftarrow \mathbf{0}^{|I| \times |K|}$ 
2:  $\pi \leftarrow$  available capacity of ESs
3:  $l \leftarrow$  list of ESs
4: for each randomly chosen missed request  $i$  do
5:    $y_i \leftarrow$  type of  $i$ 
6:    $s_i \leftarrow$  instance count of  $i$ 
7:    $r_i \leftarrow$  size of each instance of  $i$ 
8:   for each randomly chosen ES  $k$  in  $l$  do
9:     if  $k$  can process  $y_i$  and  $\pi_k \geq r_i$  then
10:       $n_{assign} \leftarrow \min\{s_i, \lfloor \frac{\pi_k}{r_i} \rfloor\}$ 
11:      if  $n_{assign} > 0$  then
12:         $\pi_k \leftarrow \pi_k - n_{assign} \times r_i$ 
13:         $s_i \leftarrow s_i - n_{assign}$ 
14:         $X(i, k) \leftarrow X(i, k) + n_{assign}$ 
15:        if  $s_i = 0$  then
16:          break  $\triangleright$  Done processing missed
      end if
    end if
  end for
  end for
17: end for

```

Algorithm 2 GRD algorithm description

```

1:  $X \leftarrow \mathbf{0}^{|I| \times |K|}$ 
2:  $\pi \leftarrow$  available capacity of ESs
3:  $V \leftarrow$  placement value,  $\mathbb{R}^{|I| \times |K|}$ 
4:  $s \leftarrow$  instance counts of missed requests
5: while  $\text{sum}(s) > 0$  do
6:    $v, i, k \leftarrow \max(V)$   $\triangleright$  the max value  $v$ 
     appears at  $i$ -th row and  $k$ -th column in  $V$ , where  $i$  and  $k$ 
     indicates the missed request and ES index, respectively
7:   if  $v \leq 0$  then
8:     break
9:   end if
10:  if  $s_i \leq 0$  then
11:     $V(i, k) \leftarrow 0$  and then continue
12:  end if
13:   $y_i \leftarrow$  service type of  $i$ 
14:   $r_i \leftarrow$  size of each instance of  $i$ 
15:  if ES  $k$  can handle  $y_i$  then
16:    if  $\pi_k \geq r_i$  then
17:       $n_{\text{assign}} \leftarrow \min\{s_i, \lfloor \frac{\pi_k}{r_i} \rfloor\}$ 
18:       $\pi_k \leftarrow \pi_k - n_{\text{assign}} \times r_i$ 
19:       $s_i \leftarrow s_i - n_{\text{assign}}$ 
20:       $X(i, k) \leftarrow X(i, k) + n_{\text{assign}}$ 
21:      if  $\pi_k \leq 0$  or  $s_i \leq 0$  then
22:         $V(i, k) \leftarrow 0$ 
23:      end if
24:    else
25:       $V(i, k) \leftarrow 0$ 
26:    end if
27:  end if
28: end while

```

B. Sub-Optimality Analysis on a Small-Scale Network

먼저 도 7과 같이 소규모 네트워크를 구성하고 ILP와 비교하여 LR(c,d)의 차선책을 연구하기 위해 시뮬레이션이 수행되었다. 네트워크에는 2, 6, 15, 16으로 표시된 4개의 ES가 있다. 이는 각 ES가 직접 연결된 라우터의 ID이지만 편의상 ES의 ID로도 사용된다. LR(d)의 경우 모든 ES가 관리자라 구성될 수 있다. 이는 극단적이고 완전히 분산된 경우이지만 네트워크에 존재하는 ES의 수가 적기 때문에 아래와 같이 최적성 격차가 크게 증가하지 않는다. 또한, Qtype, Qrate, Qpri, Qsz, Qcapa는 각각 5, 3, 3, 3, 15로 설정된다.

표 2

V				no. of instances
$-\infty$	$-\infty$	$-\infty$	1.4500	2
$-\infty$	1.9923	1.9182	$-\infty$	3
1.0923	$-\infty$	1.0460	1.0279	1
$-\infty$	$-\infty$	1.9460	$-\infty$	1
$-\infty$	$-\infty$	$-\infty$	1.4779	5
$-\infty$	1.0460	$-\infty$	$-\infty$	2
$-\infty$	$-\infty$	$-\infty$	1.55	5
$-\infty$	$-\infty$	2.0000	$-\infty$	3
$-\infty$	1.9279	2.0000	$-\infty$	3

표 3

	3		
1			
		1	
			4
			5
		3	
	2	1	

표 2의 다음 행렬 $V \in \mathbb{R}^{|I| \times |K|}$ 을 고려할 수 있다. 여기서, 9개의 행과 4개의 열은 9개의 누락된 요청과 4개의 ES(즉, 각각 열 인덱스 1, 2, 3, 4에 해당하는 2, 6, 15 및 16)를 나타낸다. 예를 들어, $V(1, 4)$ 는 누락된 요청 1이 처리를 위해 ES 16으로 전달될 때의 배치 값이다. 반면에 누락된 요청 2는 ES 2로 전달될 수 없으며 해

당 서비스 이미지가 ES에 설치되어 있지 않기 때문에 V에서 $-\infty$ 가 된다. ILP(c)와 LR(c,d) 모두에서 공식화된 이 문제의 단순성으로 인해 둘 다 최적의 솔루션을 계산하는 데 약간의 시간이 걸렸다. 이 특정 문제 설정에서 ILP(c) 및 LR(c,d)는 34.5324의 동일한 목표 값과 표 3에 표시된 것과 동일한 최적의 결정을 초래한다. 특히, LR(c,d)는 차선택이므로 그 해는 일반적으로 ILP(c)의 해만큼 좋지 않다. 그러나 이 특정 시나리오에서 최적성 격차는 0이었다. 그 이유는 대부분의 높은 보상 누락 요청이 ES 간에 겹치지 않았기 때문이다. 예를 들어, V의 네 번째 열에 해당하는 ES 16의 경우 가장 높은 보상을 받지 못한 요청은 V의 7번째 행에 있는 요청이며 나머지 ES에는 해당되지 않는다. 따라서 모든 인스턴스를 처리하기로 결정했다. 두 번째로 높은 보상을 받지 못한 요청은 V의 5번째 행에 있는 요청이며, 이는 나머지 ES에도 해당되지 않는다. 그래서 최대한 처리하기로 결정되었다.

[0124] 반면에 V의 두 번째 열과 세 번째 열에 각각 해당하는 ES 6과 15의 경우는 상당히 복잡하다. ES 6의 경우 가장 큰 보상을 받지 못하는 누락된 요청은 V의 두 번째 행에 있는 요청이므로 모든 인스턴스를 처리하기로 결정되었다. 나머지 용량에 대해서는 마지막으로 누락된 요청을 최대한 처리하기로 결정하여 두 개의 인스턴스를 가져갔다. 그러나, 마지막 누락된 요청은 ES 15에 대한 두 가지 가장 중요한 누락된 요청 중 하나이다. 마지막 두 개의 누락된 요청은 ES 15의 모든 것 중 가장 높은 자리 값인 2를 갖는다. 그러나, λ 업데이트 덕분에 마지막 누락된 요청은 마지막 누락된 요청에서 두 번째보다 더 많은 패널티를 받는다. 따라서 ES 15에서는 마지막에서 두 번째로 놓친 요청이 가장 높은 보상을 받았다. 이런 식으로 λ 를 업데이트함으로써 하나의 ES에 대한 가장 높은 보상의 놓친 요청이 나머지 ES에 대한 가치가 떨어진다. 따라서 각 ES의 개별 의사 결정 프로세스도 ILP(c)와 동일한 전역 최적 솔루션으로 귀결되었다. 실험에서 LR(d)와 LR(c)는 모두 빠르게 수렴했다. 앞서 언급한 바와 같이 λ 값은 무작위로 초기화되며 평균적으로 LR은 고려되는 시나리오에 대해 다섯 번째 반복에서 수렴된다. 각 반복에서 LR(d)는 먼저 ESM별 최적 솔루션과 ESM별 목표값을 계산하여 이중 목표값, 즉 모든 목표값의 합을 산출한다. 그런 다음 LR에서 집계된 최적 솔루션을 사용하여 ILP의 목적 값을 계산한다. 그러나, 일반적으로 LR 기반 접근 방식은 ILP(c)에 비해 차선택이 된다. 동일한 네트워크 구성에서 10개의 다른 요청 도달 시나리오에 대한 시뮬레이션이 수행되었으며 LR(d)의 평균 최적성 격차는 0.86이었고 표준 편차는 0.80이며 이는 ILP의 평균 최적 목표 값의 3.31%에 해당한다. 그러나 LR(c)와 ILP(c) 사이의 최적성 격차는 고려된 모든 시나리오에서 0에 가까웠다.

[0125] C. Performance Comparison on Small-Scale Networks

[0126] 이하에서는 도 8과 같이 네트워크 크기와 ES의 수를 모두 약간 늘린 시뮬레이션이 진행되었다. ESM의 수는 앞서 언급된 규칙에 따라 3개(7, 8, 18)로 설정되었다. 다른 네트워크 매개변수, 즉 Qtype, Qrate, Qpri, Qsz 및 Qcapa는 위와 동일하게 유지되며 LR(c,d)에 대한 반복 제한은 10이다. 10개의 서로 다른 시나리오(즉, 다른 요청 도착)이 도 9에 묘사되어 있다.

[0127] ILP(c)를 제외하고 LR 기반 접근 방식은 나머지를 능가했으며 LR(c)는 최적의 솔루션을 산출했다. LR(d)의 최적성 격차는 평균 0.68로 ILP(c)보다 1.74%만 낮다. 이러한 작은 성능 저하가 본 개시에 따라 제안된 LR(d)의 효율성을 증명한다. 게다가, 고려된 시나리오에서 GRD(c)는 평균적으로 최적의 솔루션과 비교하여 약간의 성능 저하를 보였다. 그러나 시나리오 #10과 같은 일부 시나리오에서는 차이가 컸다. 나머지 방법인 GRD(d) 및 RND(c,d)는 많은 양의 성능 저하를 기록하여 MSM 문제에 대한 솔루션으로 부적절함이 확인되었다.

[0128] Distribution Price(DP)를 중앙 집중식 접근 방식과 분산 방식 간의 평균 객관적 값의 차이라고 할 수 있다. 분명히 중앙 집중식 접근 방식은 많은 메시지 교환과 큰 계산 복잡성을 발생시키면서 글로벌 지식을 가지고 있기 때문에 항상 더 나은 성능을 보인다. LR의 DP는 0.68(1.74%)에 불과한 반면 GRD와 RND의 DP는 각각 4.81(13.06%) 및 4.57(13.83%)로 상대적으로 크다. 이는 전역 정보가 없음에도 불구하고 제안된 LR(d)가 LR(c)와 유사한 성능을 달성할 수 있음을 보여준다. 그러나 제한된 로컬 뷰는 RND 및 GRD의 성능 향상을 크게 제한한다.

표 4

Algo.	Priority	Delay	Fwd. MRs
ILP	36.50	3.05	23.80
LR(d)	36.00	3.10	23.40
LR(c)	36.50	3.05	23.80
RND(d)	27.50	4.44	18.80
RND(c)	32.60	4.56	21.60
GRD(d)	31.60	3.83	19.60
GRD(c)	34.70	3.00	22.20

[0129]

- [0130] 표 4는 다음 기준과 관련하여 고려된 알고리즘의 성능을 요약한 것이다.
- [0131] * 우선 순위: 10개 시나리오에 대해 평균 처리된/전달된 누락된 요청의 우선 순위 값의 합계이다.
- [0132] * 지연: 선택한 누락된 요청을 10개 시나리오에 걸쳐 평균화하여 전달하는 평균 지연이다.
- [0133] * 빠른 MR(Forwarded Missed Request): 10개 시나리오에 대한 평균 전달을 위해 선택된 누락된 요청의 총 수이다.
- [0134] ILP(d) 및 LR(c,d)에 대한 최적화 문제를 공식화하고 해결하는 데 중요한 역할을 하는 배치 값 행렬 V 를 계산할 때 우선 순위와 전송 지연이 모두 고려된다. 또한 전달된 누락된 요청의 수는 이 작업의 주요 목적 중 하나이다.
- [0135] 값이 클수록 Priority와 Fwd 모두에 더 좋다. MR은 낮을수록 지연에 더 좋다. ILP(c)와 LR(c) 모두 Delay를 제외한 모든 기준에서 동일하고 최적의 성능을 보여준다. GRD(c)가 가장 짧은 지연을 산출했지만 ILP(c) 및 LR(c,d)에 비해 개선이 크지 않다. 또한 다른 모든 기준에서 GRD(c)는 ILP(c) 및 LR(c,d)보다 성능이 우수하다. 무작위 접근 방식인 RND(c)와 RND(d)는 중앙 집중식 알고리즘과 분산 알고리즘 중 모든 기준에서 각각 가장 낮은 성능을 기록했다.
- [0136] ILP(c)가 가장 시간이 많이 소요되지만 문제의 크기가 상대적으로 작기 때문에 ILP(c,d)에 비해 시간의 차이는 크지 않았다. 그럼에도 불구하고 RND(c,d)와 GRD(c,d)는 해결하는 데 가장 짧은 시간이 걸렸다. 더 많은 ES가 있는 더 큰 네트워크 레이아웃을 고려하는 다음 하위 섹션에서 CPU 시간과 수렴에 대해 추가적으로 설명한다.
- [0137] D. Performance Comparison on Large-Scale Networks
- [0138] 이 섹션에서는 더 큰 네트워크가 고려되었다. 네트워크 크기는 10×10 그리드로 증가하였고, 무작위로 생성된 ES의 총 개수는 도 10과 같이 28개이다. ESM의 개수는 3, 즉 55, 56, 64이다. 앞서 ESM 선택 방법을 제안한 것처럼 ES의 인구가 많은 경우 네트워크의 중앙에 가깝게 배치된 ES가 관리자로 선택될 가능성이 높다. LR(c)의 반복 한계는 이전과 같이 10으로 유지되는 반면 LR(d)의 반복 한계는 3배(즉, 30)이다. 이러한 한계는 반복된 실증 연구에 의해 발견되며 문제 크기에 따라 다를 수 있다. Qtype은 두 배(즉, 10)인 반면 Qrate, Qpri, Qsz 및 Qcapa는 이전 섹션과 동일하게 유지되었다. 10가지 다른 시나리오에 걸쳐 계산된 목표 값은 도 11을 통해 제공된다.
- [0139] 소규모 네트워크의 경우와 유사하게 LR(c,d)는 ILP(c)를 제외하고 나머지를 능가한다. LR(c)가 이번에는 차선책을 산출했지만 최적성 격차는 사소한 정도이다(즉, 0.29%). 반면에 LR(d)의 차선책이 눈에 띄게 되었고 고려된 모든 시나리오에서 목적값의 저하가 관찰되었다. LR(d)를 사용하면 각 ESM이 제한된 정보로 전체 문제의 분할을 해결할 수 있었다. 따라서 특정 누락된 요청 i 에 대해 존재하는 것보다 더 많은 인스턴스가 전달되도록 예약될 수 있다. 이러한 경우 누락된 요청이 있는 ES는 추가 전달 일정 요청을 무시할 수 있다. 결과적으로 다른 누락된 요청을 처리하는 데 사용할 수 있는 일부 공간이 아무 것도 예약되지 않아 목표 값이 저하된다. 증가된 문제 크기는 솔루션의 이러한 비효율성을 악화시킬 수 있다.
- [0140] 흥미로운 사실은 이전의 소규모 네트워크 사례와 달리 GRD(c,d)의 목표 값이 RND(c,d)의 목표 값과 유사해졌다는 것이. 문제 크기가 작을 때 탐욕스러운 접근 방식은 놓친 요청을 큰 배치 값으로 먼저 할당하여 더 나은 솔루션을 얻을 수 있다. 그러나 ES의 수와 누락된 요청의 수가 모두 증가함에 따라 무작위 완전 배낭 채우기 방법으로 작동하는 RND가 GRD만큼 좋은 솔루션을 얻을 수 있었다.
- [0141] LR의 DP는 3.47%인 반면 GRD와 RND의 DP는 각각 23.75%와 17.19%이다. 고려된 모든 접근 방식에서 소규모 네트워크의 경우에 비해 DP가 증가했으며, 이는 글로벌 정보의 부재로 인해 분산 방식의 성능이 훨씬 더 크게 저하되었음을 의미한다.

표 5

Algo.	Priority	Delay	Fwd. MRs
ILP	310.50	4.65	174.90
LR(d)	290.40	4.32	169.10
LR(c)	310.20	4.75	174.60
RND(d)	218.00	9.06	121.50
RND(c)	253.30	8.86	148.40
GRD(d)	237.60	4.37	101.60
GRD(c)	284.40	4.13	127.20

[0142]

표 6

Algo.	Time (seconds)	
	Mean	Std. dev.
ILP	35.33	90.57
LR(d)	0.01	0.00
LR(c)	0.08	0.00
RND(d)	0.00	0.00
RND(c)	0.01	0.00
GRD(d)	0.00	0.00
GRD(c)	0.02	0.00

[0143]

[0144]

표 5는 총 우선 순위, 누락된 요청 전달 지연 및 전달된 누락된 요청 수와 관련하여 고려된 알고리즘의 성능을 요약한다. 도 11에서 볼 수 있듯이 LR(c)는 차선책을 제공하므로 우선 순위 및 전달된 누락 요청 모두에 대해 ILP(c)가 LR(c)보다 성능이 좋다. LR(c)가 ILP(c)에 비해 지연이 개선된 것으로 나타났지만 솔루션의 효율성을 평가하는 데에만 사용할 수는 없다. GRD와 RND의 다른 특성은 놓친 요청이 더 많은 대규모 네트워크에서 더 명확해졌다. GRD는 보상이 높은 누락된 요청을 먼저 할당하는 경향이 있으므로 우선 순위가 높고 지연이 낮은 누락된 요청 수가 더 적다. 반면에 RND는 누락된 요청을 무작위로 철저하게 할당했고, 결과적으로 지연이 더 커지고 더 많은 누락된 요청을 전달하면서 총 우선 순위가 더 작아졌다.

[0145]

표 6은 고려된 문제를 해결하기 위한 CPU 시간을 보여준다. 예상대로 ILP(c)는 최적화 문제를 해결하는 데 가장 높은 CPU 시간을 기록했다. 평균값은 35.33초이므로 온라인 처리에 사용되기 어렵다. 또한 런타임은 큰 표준 편차를 생성하는 문제 크기에 크게 의존하며, 나머지는 그렇지 않다. 반면에 GRD와 RND는 모든 시나리오에서 빠르게 종료되었다. 제안된 GRD 구현에서는 대부분의 양의 배치 값을 방문하므로 차이는 무시할 수 있지만 RND보다 시간이 오래 걸렸다. 더 나은 구현은 런타임을 더욱 줄일 수 있다. LR 접근 방식은 또한 짧은 시간에 솔루션을 찾을 수 있어 온라인 알고리즘으로 활용될 수 있는 가능성을 제시하였다. 그러나 앞서 언급한 바와 같이 중앙 집중식 접근 방식은 설치된 이미지와 모든 ES의 남은 용량을 포함하는 전역 정보를 필요로 하므로 이러한 정보의 교환은 상당한 오버헤드가 될 수 있다. 반면 LR(d)의 각 ESM은 정보의 일부를 수집하고 문제 크기가 줄어들어 솔루션을 반환하는 데 약간의 시간만 소요되었다.

[0146]

도 12는 LR(c,d)의 수렴 거동을 보여준다. 도 12의 (a)에서 볼 수 있듯이 LR(c)는 빠르게 수렴한다. 평균적으로 처음 5번의 반복 실행 내에 수렴에 도달했다. 반면, LR(d)는 도 12의 (b)와 같이 수렴하는 데 시간이 더 오래 걸리며 이는 중앙 집중식 방식에 비해 분산 방식의 또 다른 단점이다. 반복 제한이 작은 경우 솔루션을 곧 사용할 수 있게 되지만 얻을 수 있는 최선은 아닐 수 있다. 반면에 더 큰 반복 제한의 경우 더 나은 솔루션을 얻을 수 있지만 솔루션을 사용할 수 있게 될 때까지 시간이 더 오래 걸릴 수 있다. 균형을 맞추기 위해, 추가 반복이 더 나은 솔루션을 생성하지 않을 것으로 예상되는 경우 더 빨리 종료되는 향상된 접근 방식을 예로 사용할 수 있다. 이러한 특성과 문제 크기를 고려하여 반복 제한을 신중하게 결정해야 한다.

[0147]

이렇듯 본 개시는 사용자로부터 받은 요청에 대해 ES가 서비스를 제공할 수 없을 때 발생하는 MEC 서비스 누락 문제를 해결하기 위한 솔루션을 제공하였다. 이러한 요청을 해당 서비스 이미지와 함께 설치된 ES에 효과적으로 전달하기 위해 변형 다중 배낭 프레임워크를 사용하여 요청 전달 문제를 공식화했다. 결과적인 정수 선형 문제는 최적의 요청 전달 일정을 생성할 수 있지만 문제 크기가 증가함에 따라 높은 계산 복잡성을 겪을 수 있다. 다양한 네트워크 크기와 다양한 요청 도착 시나리오에 대한 평가를 통해 제안된 중앙 집중식 및 분산형 Lagrange 이중 솔루션이 실행 시간이 짧고 보상이 높은 솔루션이 될 수 있음을 보여주었다. 중앙 집중식 접근 방식은 몇 번의 반복만으로 거의 최적의 솔루션을 얻을 수 있지만 문제를 올바르게 해결하려면 전역 지식이 필

요하다. 네트워크 전체의 정보를 수집하는 데 필요한 많은 메시지 교환을 고려하면 전체 보상을 희생하면서 분해된 Lagrangian 이중 접근 방식을 사용할 수 있다. 또한 계산 시간을 더욱 단축할 수 있어 온라인 처리에 더 적합하다. 중앙 집중식 및 분산형 라그랑주 이중 접근 방식의 수렴에 대한 연구도 수행되었다. 중앙 집중식 접근 방식은 적은 수의 반복 내에서 수렴되었지만 분산된 접근 방식은 수렴하는 데 더 많은 반복이 필요했다. 최적에 가까운 포워딩 일정을 필요로 하는 소규모 네트워크 또는 이상적인 네트워크 조건(즉, 전송 지연이 적고 큰 대역폭)과 고성능 컴퓨팅 장치를 갖춘 대규모 네트워크는 제안된 중앙 집중식 접근 방식을 온라인 스케줄링 방법 상에 활용할 수 있다. 반면에 온라인 스케줄링이 필요한 자원이 제한된 ES를 가진 대규모 네트워크에서는 제안된 분산 솔루션을 활용할 수 있다.

- [0148] 도 13은 본 개시의 일 실시 예에 따른 ESM(Edge Server Management)의 구성을 설명하기 위한 블록도이다.
- [0149] 도 13을 참조하면, ESM(100)은 메모리(110), 통신부(120), 및 프로세서(130)를 포함하는 전자 장치 내지는 시스템으로 구현될 수 있다.
- [0150] 메모리(110)는 ESM(100)의 구성요소들의 전반적인 동작을 제어하기 위한 운영체제(OS: Operating System) 및 서버(100)의 구성요소와 관련된 적어도 하나의 인스트럭션 또는 데이터를 저장하기 위한 구성이다.
- [0151] 메모리(110)는 ROM, 플래시 메모리 등의 비휘발성 메모리를 포함할 수 있으며, DRAM 등으로 구성된 휘발성 메모리를 포함할 수 있다. 또한, 메모리(110)는 하드 디스크, SSD(Solid state drive) 등을 포함할 수도 있다.
- [0152] 통신부(120)는 다양한 유무선 통신방식으로 적어도 하나의 외부 장치와 통신을 수행하기 위한 회로, 모듈, 칩 등을 포함할 수 있다.
- [0153] 통신부(120)는 다양한 네트워크를 통해 외부 장치들(ex. 다른 ES, 다른 ESM, BS, 사용자 단말 등)과 연결될 수 있다.
- [0154] 프로세서(130)는 ESM(100)을 전반적으로 제어하기 위한 구성이다. 구체적으로, 프로세서(130)는 메모리(110)와 연결되는 한편 메모리(110)에 저장된 적어도 하나의 인스트럭션을 실행함으로써 본 개시의 다양한 실시 예들에 따른 동작을 수행할 수 있다.
- [0155] 프로세서(130)는 CPU, AP, DSP(Digital Signal Processor) 등과 같은 범용 프로세서, GPU, VPU(Vision Processing Unit) 등과 같은 그래픽 전용 프로세서 또는 NPU와 같은 인공지능 전용 프로세서 등을 포함할 수 있다. 인공지능 전용 프로세서는, 특정 인공지능 모델의 훈련 내지는 이용에 특화된 하드웨어 구조로 설계될 수 있다.
- [0156] 한편, 이상에서 설명된 다양한 실시 예들은 서로 저촉되거나 모순되지 않는 한 두 개 이상의 실시 예가 서로 결합되어 구현될 수 있다.
- [0157] 한편, 이상에서 설명된 다양한 실시 예들은 소프트웨어(software), 하드웨어(hardware) 또는 이들의 조합된 것을 이용하여 컴퓨터(computer) 또는 이와 유사한 장치로 읽을 수 있는 기록 매체 내에서 구현될 수 있다.
- [0158] 하드웨어적인 구현에 의하면, 본 개시에서 설명되는 실시 예들은 ASICs(Application Specific Integrated Circuits), DSPs(digital signal processors), DSPDs(digital signal processing devices), PLDs(programmable logic devices), FPGAs(field programmable gate arrays), 프로세서(processors), 제어기(controllers), 마이크로 컨트롤러(micro-controllers), 마이크로 프로세서(microprocessors), 기타 기능 수행을 위한 전기적인 유닛(unit) 중 적어도 하나를 이용하여 구현될 수 있다.
- [0159] 일부의 경우에 본 명세서에서 설명되는 실시 예들이 프로세서 자체로 구현될 수 있다. 소프트웨어적인 구현에 의하면, 본 명세서에서 설명되는 절차 및 기능과 같은 실시 예들은 별도의 소프트웨어 모듈들로 구현될 수 있다. 상술한 소프트웨어 모듈들 각각은 본 명세서에서 설명되는 하나 이상의 기능 및 작동을 수행할 수 있다.
- [0160] 한편, 상술한 본 개시의 다양한 실시 예들에 따른 전자 장치(100)에서의 처리동작을 수행하기 위한 컴퓨터 명령어(computer instructions) 또는 컴퓨터 프로그램은 비일시적 컴퓨터 판독 가능 매체(non-transitory computer-readable medium)에 저장될 수 있다. 이러한 비일시적 컴퓨터 판독 가능 매체에 저장된 컴퓨터 명령어 또는 컴퓨터 프로그램은 특정 기기의 프로세서에 의해 실행되었을 때 상술한 다양한 실시 예에 따른 ESM/전자 장치(100)에서의 처리 동작을 상술한 특정 기기가 수행하도록 한다.
- [0161] 비일시적 컴퓨터 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 비일시적

컴퓨터 판독 가능 매체의 구체적인 예로는, CD, DVD, 하드 디스크, 블루레이 디스크, USB, 메모리카드, ROM 등이 있을 수 있다.

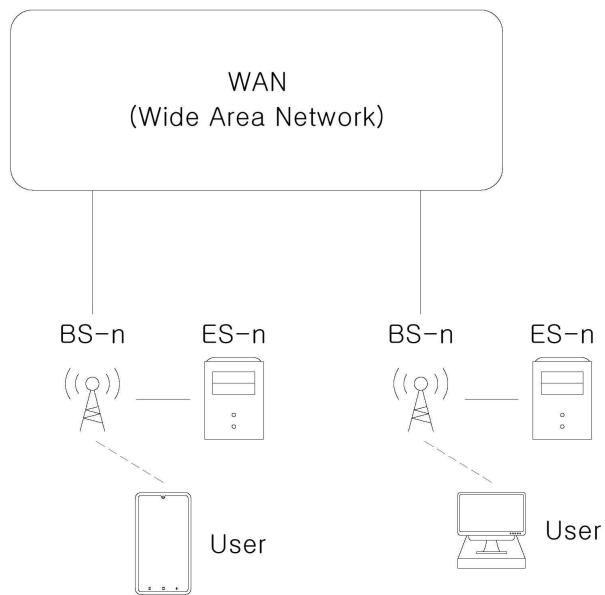
[0162] 이상에서는 본 개시의 바람직한 실시 예에 대하여 도시하고 설명하였지만, 본 개시는 상술한 특징의 실시 예에 한정되지 아니하며, 청구범위에서 청구하는 본 개시의 요지를 벗어남이 없이 당해 개시에 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변형실시가 가능한 것은 물론이고, 이러한 변형실시들은 본 개시의 기술적 사상이나 전망으로부터 개별적으로 이해되어서는 안될 것이다.

부호의 설명

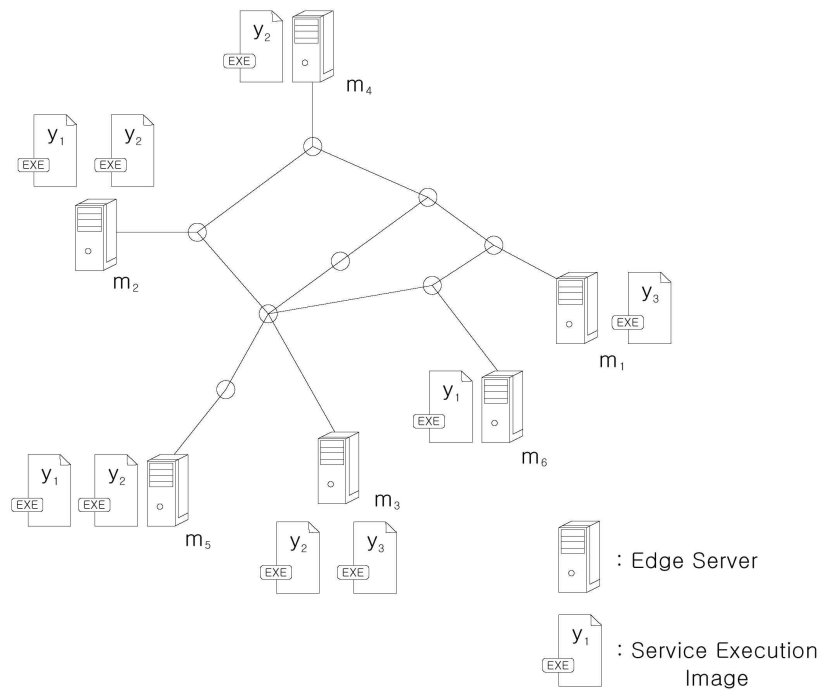
[0163] 100: ESM(Edge Server Manager)

도면

도면1



도면2



(a) A small-scale system model.

Service type		y_1	y_2	y_3
Size (no. of resource units of each request)		1	2	2
Missing ESs	ID	m_1, m_3, m_4	m_1, m_6	m_2, m_4 m_5, m_6
	Number of requests received	3, 4, 4	3, 1	5, 4, 5, 2
Serving ESs	ID	m_2, m_5, m_6	m_2, m_3 m_4, m_5	m_1, m_3
	Number of resource units available	5, 7, 1	5, 13, 9, 7	6, 13

(b) The missing and serving ESs in the system

도면3

Data structure for Item

Knapsack ID	Corresponding ES	Service-type provisioning set (i.e., installed imgs)	Capacity (no. available resource units)
-------------	------------------	--	---

Data structure for Knapsack

Item ID	Corresponding missing ES	Associated service type	Number of Instances	Size of each instance
---------	--------------------------	-------------------------	---------------------	-----------------------

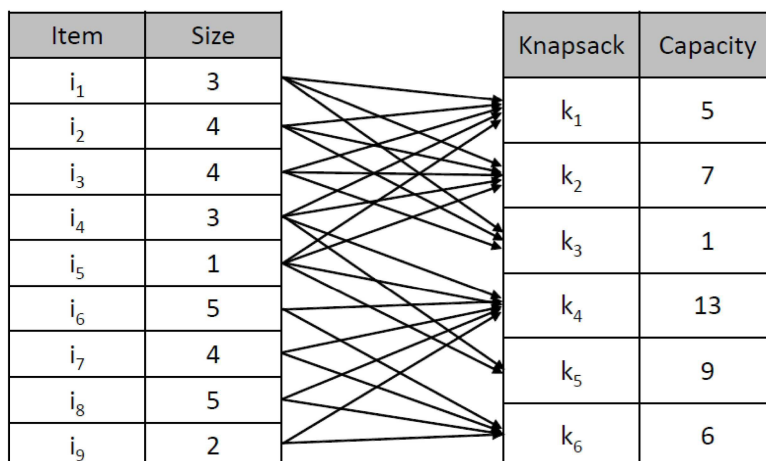
(a) Proposed data structure format.

Item ID	Corresponding missing ES	Associated service type	Number of instances	Size of each instance
i_1	m_1	y_1	3	1
i_2	m_3	y_1	4	1
i_3	m_4	y_1	4	1
i_4	m_1	y_2	3	2
i_5	m_6	y_2	1	2
i_6	m_2	y_3	5	2
i_7	m_4	y_3	4	2
i_8	m_5	y_3	5	2
i_9	m_6	y_3	2	2

Knapsack ID	Corresponding ES	Service-type provisioning set (installed imgs)	Capacity size (no. avail. Resource units)
k_1	m_2	$\{y_1, y_2\}$	5
k_2	m_5	$\{y_1, y_2\}$	7
k_3	m_6	$\{y_1\}$	1
k_4	m_3	$\{y_2, y_3\}$	13
k_5	m_4	$\{y_2\}$	9
k_6	m_1	$\{y_3\}$	6

(b) Example of information integration by using the proposed item and knapsack structures.

도면4



도면5

Service priority

Service type	1	2	3
Priority	1	2	3
Min. priority	1		
Max. priority	3		

Transmission costs (hop count)

ES	m_1	m_2	m_3	m_4	m_5	m_6
m_1	0	5	4	4	5	3
m_2	5	0	3	3	4	4
m_3	4	3	0	4	3	3
m_4	4	3	4	0	5	4
m_5	5	4	3	5	0	4
m_6	3	4	3	4	4	0
Min. cost	3					
Max. cost	5					

(a) The assumed priority parameter values.

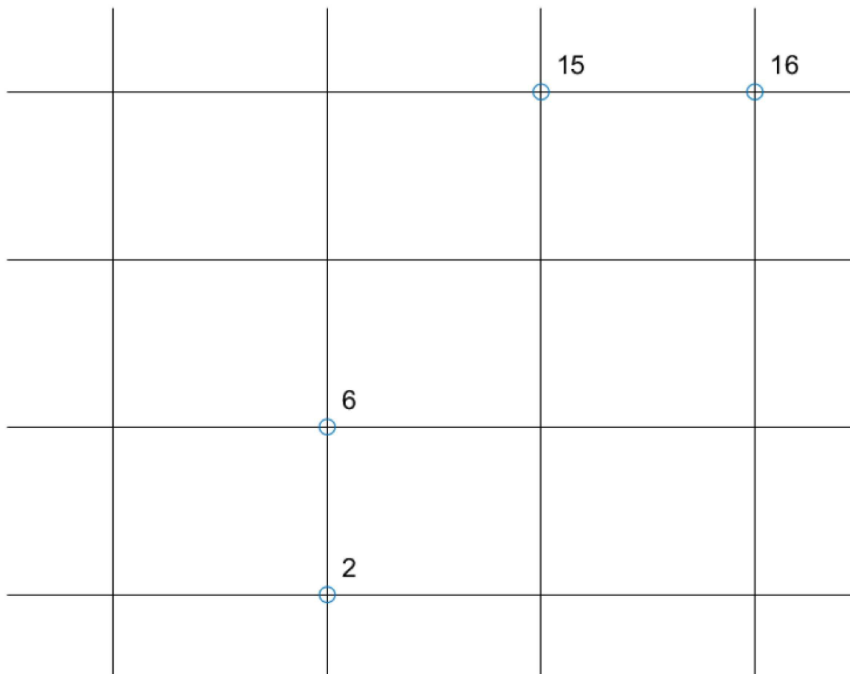
Knapsack ID /Item ID	$k_1(m_2)$	$k_2(m_5)$	$k_3(m_6)$	$k_4(m_3)$	$k_5(m_4)$	$k_6(m_1)$
i_1	0	0	0.1	N/A	N/A	N/A
i_2	0.1	0.1	0.1	N/A	N/A	N/A
i_3	0.1	0	0.05	N/A	N/A	N/A
i_4	0.45	0.45	N/A	0.5	0.5	N/A
i_5	0.5	0.5	N/A	0.55	0.5	N/A
i_6	N/A	N/A	N/A	1	N/A	0.9
i_7	N/A	N/A	N/A	0.95	N/A	0.95
i_8	N/A	N/A	N/A	1	N/A	0.9
i_9	N/A	N/A	N/A	1	N/A	1

(b) The resulting edge weight values.

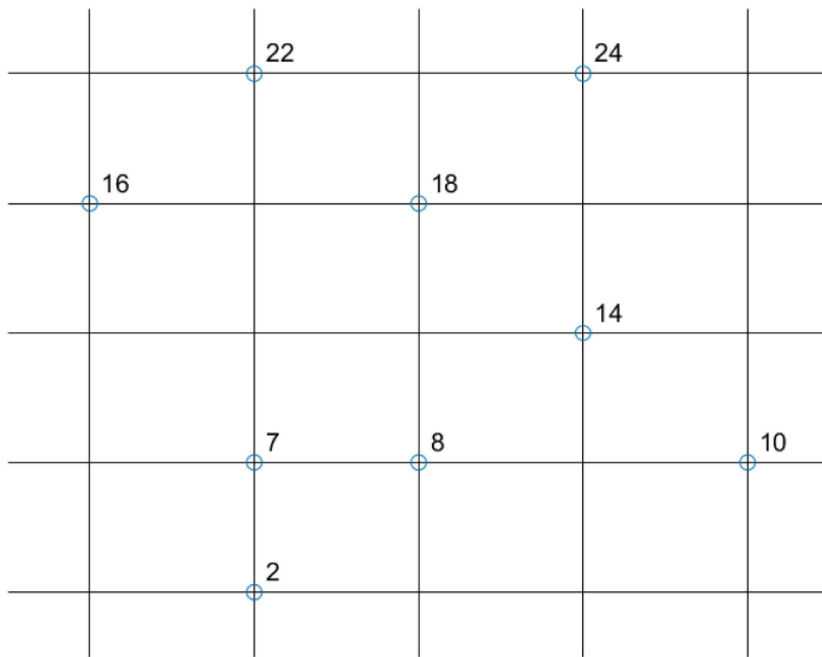
도면6

Decision variable	VMK solution			ILP solution		
	Item	Knapsack	Number of instances placed	Missing ES	Serving ES	No. of processed requests
$x_{12}=2$	1	2	2	m_1	m_5	2
$x_{13}=1$	1	3	1	m_1	m_6	1
$x_{22}=4$	2	2	4	m_3	m_5	4
$x_{31}=4$	3	1	4	m_4	m_2	4
$x_{45}=3$	4	5	3	m_1	m_4	3
$x_{55}=1$	5	5	1	m_6	m_4	1
$x_{64}=1$	6	4	1	m_2	m_3	1
$x_{76}=1$	7	6	1	m_4	m_1	1
$x_{84}=5$	8	4	5	m_5	m_3	5
$x_{96}=2$	9	6	2	m_6	m_1	2

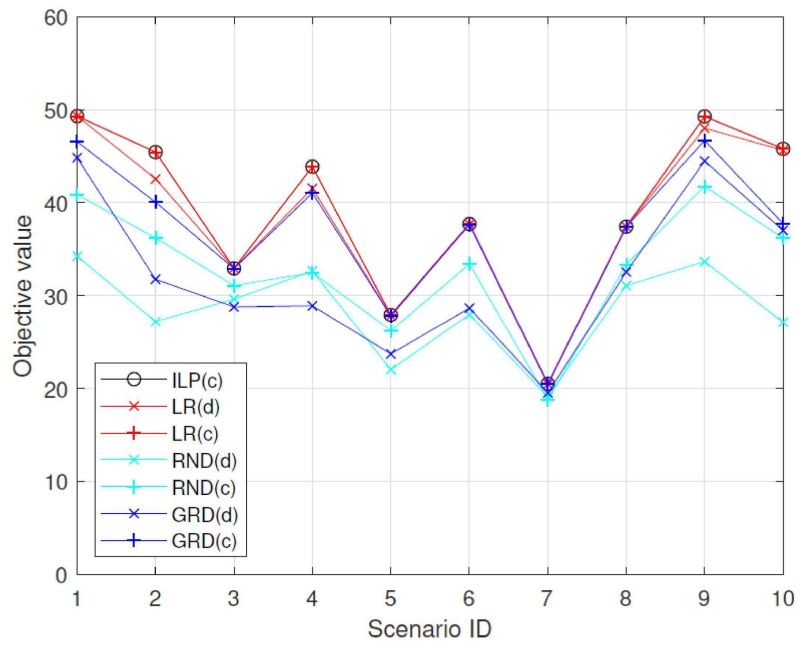
도면7



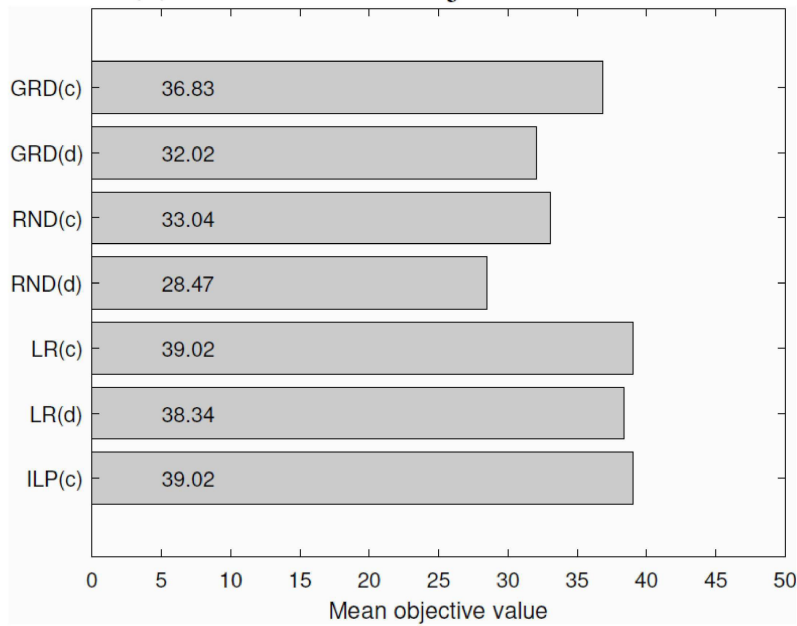
도면8



도면9

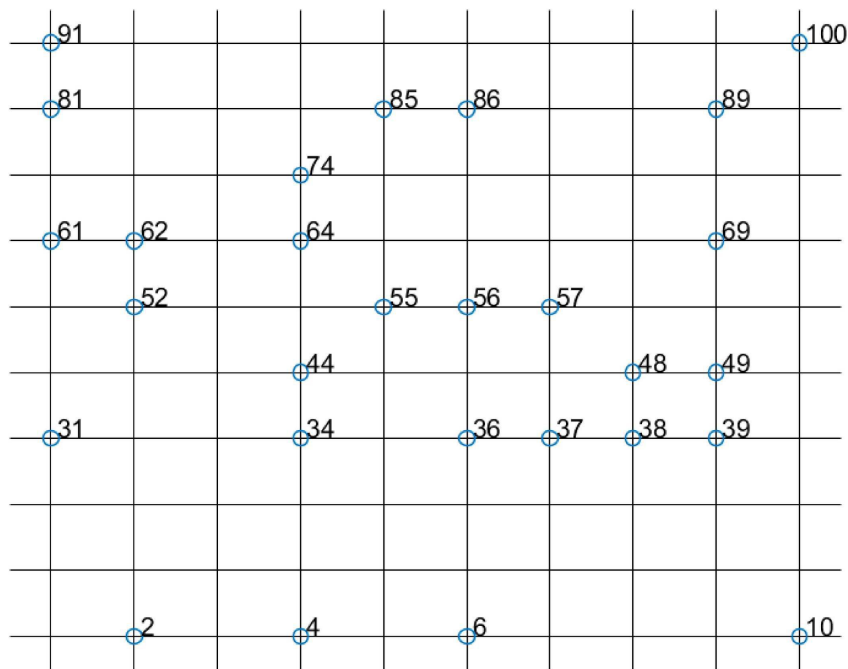


(a) Per-scenario objective values

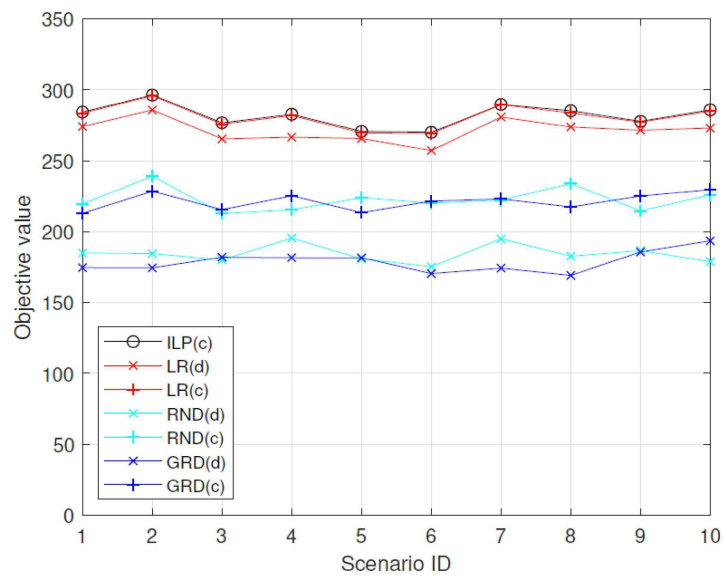


(b) Mean objective values

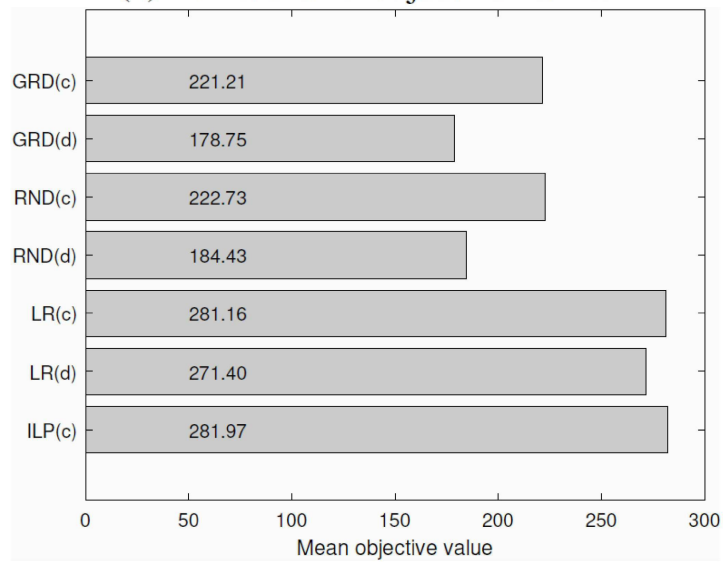
도면10



도면11

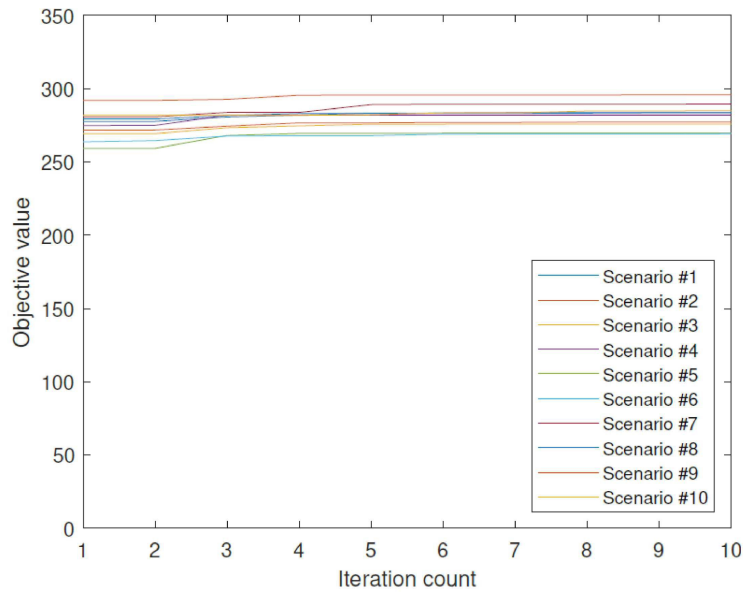


(a) Per-scenario objective values

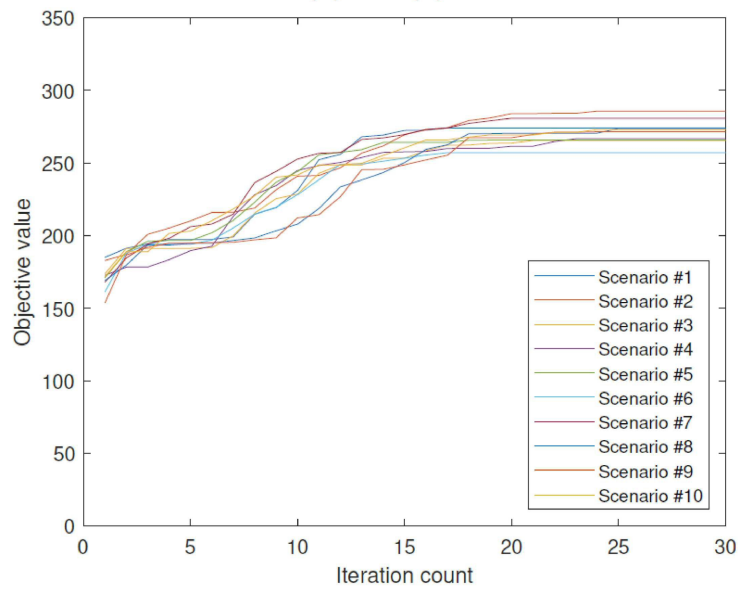


(b) Mean objective values

도면12



(a) LR(c)



(b) LR(d)

도면13

