(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0072783
(43) 공개일자 2025년05월26일

(51) 국제특허분류(Int. Cl.)
H04L 41/0893 (2022.01) *H04L 41/0826* (2022.01)
H04W 88/08 (2009.01)
(52) CPC특허분류
H04L 41/0893 (2022.05)
H04L 41/0826 (2022.05)
(21) 출원번호 10-2023-0159794
(22) 출원일자 2023년11월17일
심사청구일자 2023년11월17일

(71) 출원인
세종대학교산학협력단
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
(72) 발명자
송형규
경기도 성남시 분당구 중앙공원로 17, 320-303(서현동, 한양아파트)
유영환
서울특별시 강남구 선릉로72길 26-16, 401호
(뒷면에 계속)
(74) 대리인
이강민, 남승희

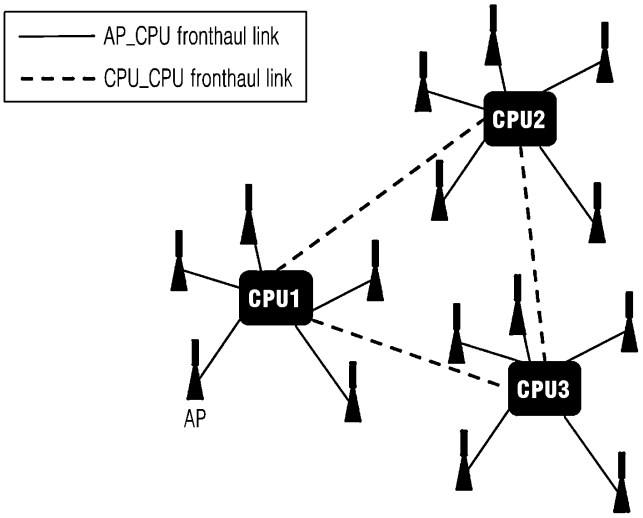
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 **셀프리 다중 중앙처리장치 시스템에서 중앙처리장치간 프론트 홀 부하를 줄이기 위한 클러스터링 알고리즘**

(57) 요 약

본 발명은 셀프리 다중 프로세서(Cell-Free Multiple CPUs) 시스템에서 CPU간 프론트홀 부하(fronthaul load)를 줄일 수 있는 군집화 알고리즘(clustering algorithm)에 관한 것으로서, 군집화(clustering)를 진행하는 AP(Access Point)의 CPU와 후보 UE(User Equipment)를 지원하는 CPU를 동일하게 설정하는 방식을 통해서 프론트홀 부하(fronthaul load)를 낮춤으로써, 프론트홀 부하(fronthaul load)를 낮추는 군집화 방법을 제공한다.

대 표 도 - 도1



(52) CPC특허분류
H04W 88/085 (2013.01)
(72) 발명자
신경호
서울특별시 광진구 광나루로19길 52, 201호
박상욱
경기도 군포시 수리산로203번안길 41, 101호
김진우
경기도 동두천시 장고갯로281번길 10, 402호
김형도
서울특별시 강북구 오현로31길 89, 103-1303

이 발명을 지원한 국가연구개발사업
과제고유번호 1345366486
과제번호 2020R1A6A1A03038540
부처명 교육부
과제관리(전문)기관명 한국연구재단
연구사업명 이공학학술연구기반구축
연구과제명 자율지능무인비행체연구소
기 여 율 33.4/100
과제수행기관명 세종대학교
연구기간 2023.03.01 ~ 2024.02.29
이 발명을 지원한 국가연구개발사업
과제고유번호 1711198096
과제번호 00254529
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성
연구과제명 메타버스 융합대학원(세종대학교)
기 여 율 33.3/100
과제수행기관명 세종대학교 산학협력단
연구기간 2023.07.01 ~ 2023.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711193909
과제번호 2021-0-01816-003
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성
연구과제명 메타버스 자율트윈 핵심기술 연구
기 여 율 33.3/100
과제수행기관명 세종대학교 산학협력단
연구기간 2023.01.01 ~ 2023.12.31

명 세 서

청구범위

청구항 1

셀프리 다중 프로세서(Cell-Free Multiple CPUs) 시스템 기반에 CPU간 프론트홀 부하(fronthaul load)를 줄이기 위한 군집화(clustering) 방법에 있어서,

각 UE(User Equipment)별로 마스터 AP(Master Access Point)와 마스터 AP에 연결된 CPU 목록을 획득하는 단계;

각 AP(Access Point)별로 동일한 파일럿(pilot)을 가지는 후보 UE(User Equipment) 목록을 획득하는 단계;

각 AP(Access Point)가 상기 획득한 후보 UE(User Equipment) 중 지원받는 CPU가 동일한 UE(User Equipment)를 군집화(clustering)하는 단계;

를 포함하는 프론트홀 부하 절감 군집화 방법.

청구항 2

제 1항에 있어서,

각 UE(User Equipment)별 master AP(Access Point)와 master AP(Access Point)에 연결된 CPU 목록을 획득하는 단계는,

수식 1을 바탕으로 LSFC가 가장 큰 AP(Access Point)를 마스터 AP로 선택하고 군집화(clustering)하는 단계를 포함하는 프론트홀 부하 절감 군집화 방법.

(수식 1)

$$C_k = \{\arg\max_l \beta_{lk}\}$$

(C_k는 UE(User Equipment) k를 지원하는 AP(Access Point)집합이며, β_{lk}는 AP(Access Point) l과 UE(User Equipment) k의 LSFC)

청구항 3

제2항에 있어서,

각 UE(User Equipment)별 master AP(Access Point)와 master AP(Access Point)에 연결된 CPU 목록을 획득하는 단계는,

수식 2를 바탕으로 마스터 AP에 연결된 CPU 목록을 획득하는 단계;를 추가로 포함하는 프론트홀 부하 절감 군집화 방법.

(수식 2)

$$q_k = \{q : C_k \cap B\}$$

(여기서 q_k는 UE(User Equipment) k를 지원하는 CPU 집합, $B = \{B_1, B_2, \ldots, B_Q\}$는 네트워크에서 고려하는 AP(Access Point)집합이며, 각 원소 집합은 해당 CPU에 연결된 AP(Access Point) 집합)

청구항 4

제 1항에 있어서,

각 AP(Access Point)별로 동일한 파일럿(pilot)을 가지는 후보 UE(User Equipment) 획득하는 단계는,

수식 3로부터 파일럿(pilot) 별로 후보 UE(User Equipment)를 선정하는 프론트홀 부하 절감 군집화 방법.

(수식 3)

$$\mathcal{K}_c = \left\{ k: \max_{j \in \mathcal{K}_{\varphi_i}} \beta_{lj} - \beta_{l\mathcal{K}_{\varphi_i}} < \gamma \right\}$$

($\mathcal{K}_c$는 후보 UE(User Equipment) 집합, $\mathcal{K}_{\varphi_i}$는 파일럿(pilot) i를 갖는 UE(User Equipment) 집합이며 $\beta_{l\mathcal{K}_{\varphi_i}}$는 $\mathcal{K}_{\varphi_i}$의 모든 UE(User Equipment)의 LSFC이다. γ는 후보 UE(User Equipment)를 선정하는 기준인 소정의 임계값)

청구항 5

제 1항에 있어서,

각 AP(Access Point)가 상기 획득한 후보 UE(User Equipment) 중 지원받는 CPU가 동일한 UE(User Equipment)를 군집화(clustering)하는 단계는,

군집화(clustering)를 진행하는 AP(Access Point)에 연결된 CPU와 후보 UE(User Equipment)를 지원하는 CPU가 동일한 UE(User Equipment)중 LSFC가 가장 큰 UE(User Equipment)를 군집에 포함시키는, 프론트홀 부하 절감 군집화 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 셀프리 다중 프로세서(Cell-Free Multiple CPUs)시스템에서 CPU간 프론트홀 부하(fronthaul load)를 줄이기 위한 AP(Access Point)의 UE(User Equipment) 군집화 알고리즘(clustering algorithm)에 관한 것이다.

배 경 기 술

[0003] 셀프리(Cell-Free) 시스템은 UE(User Equipment)가 하나의 Cell의 AP(Access Point)에 연결되어 있는 기존의 셀룰러 시스템(Cellular System)에서 벗어나 AP(Access Point)가 분산되어 있는 환경에서 다중의 AP(Access Point)가 공동으로 UE(User Equipment)를 지원하는 시스템이다. 각 AP(Access Point)는 CPU에 연결되어 있으며 각 CPU는 연결된 AP(Access Point)들의 프리코딩(precoding)과 인코딩(encoding)을 중앙집중적으로 수행하는 프로세서이다. Cell-Free는 광범위한 지역에 AP(Access Point)가 분산되어 있어 기존의 셀룰러 시스템(Cellular System)에서의 cell edge UE(User Equipment)의 신호 품질 저하 문제와 인접 AP(Access Point)들의 간섭 영향을 해결할 수 있는 유망 기술로 주목받고 있다. 하지만 분산된 AP(Access Point)들에 의해서 제어(control) 정보와 데이터(data) 정보가 교환되는 프론트홀 신호화 요구사항(Fronthaul signaling requirements)이 높아진다.

[0004] 각 UE(User Equipment)는 기존의 셀룰러 시스템(Cellular System)과 동일하게 LSFC(Large Scale Fading Coefficient)가 가장 큰 AP(Access Point)와 연결한다. 이렇게 연결된 AP(Access Point)들은 각 UE(User Equipment)의 master AP(Access Point)라고 한다.

[0005] 기존 Cell-Free Network에서 군집화 알고리즘(clustering algorithm)은 각 UE(User Equipment)의 LSFC(Large Scale Fading Coefficient)와 파일럿(pilot) 자원을 기반으로 군집화(clustering)를 수행했다. 이 방식은 파일럿(pilot) 자원이 겹치는 UE(User Equipment) 중 채널게인(Channel Gain)이 가장 큰 UE(User Equipment)들을 군집화(clustering)하는 방식이다.

[0006] 그런데, 이와 같은 종래의 군집화 방식들은 CPU간 프론트홀 부하(fronthaul load)를 증가시키는 문제가 있다.

[0007] 관련된 선행기술로는 아래와 같은 것들이 있다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 특허문헌 1: 한국 등록특허 1877243 B(2018.07.05 등록)

(특허문헌 0002) 특허문헌 2: 한국 공개특허 2023-0085891A(2023. 6. 14)

발명의 내용

해결하려는 과제

[0009] 본 발명은 셀프리 다중 프로세서(Cell-Free Multiple CPUs) 시스템에서 기존 군집화(clustering) 방식의 SE(Spectral Efficiency) 성능은 보장하면서 CPU간 프론트홀 부하(fronthaul load)를 줄이는 군집화(clustering) 방식을 제공하고자 한다.

과제의 해결 수단

[0010] 셀프리 다중 프로세서(Cell-Free Multiple CPUs) 시스템에서 본 발명의 각 AP(Access Point)는 파일럿(pilot) 자원이 겹치는 UE(User Equipment)들 중 LSFC가 큰 유저의 후보를 선정한다. 후보 UE(User Equipment)의 cluster에 포함된 AP(Access Point)들이 연결된 CPU와 동일한 CPU의 지원받는 UE(User Equipment)를 군집화(clustering)하는 방법을 제공한다.

발명의 효과

[0011] 본 발명에 따르면, 셀프리 다중 프로세서(Cell-Free Multiple CPUs) 시스템에서 각 AP(Access Point)들은 최대한 연결되어 있는 CPU가 지원하고 있는 UE(User Equipment)를 군집화(clustering)하기 때문에 CPU에서 프리코더(precoder) 설계 시에 기존 군집화 알고리즘(clustering algorithm)에 비해 더 낮은 CPU간 프론트홀 부하(fronthaul load)를 발생하여 프론트홀 링크(fronthaul link)의 부담을 덜어줄 수 있다. 또한 동시에 기존의 군집화 알고리즘(clustering algorithm)과 비슷한 SE(Spectral Efficiency) 성능을 보장할 수 있다.

도면의 간단한 설명

[0013] [도 1]은 셀프리 다중 프로세서(Cell-Free Multiple CPUs) 시스템을 나타낸다.

[도 2]는 본 발명에서 제안한 군집화 알고리즘(clustering algorithm) 순서도를 나타낸다.

[도 3]은 본 발명에서 제안한 군집화 알고리즘(clustering algorithm)과 기존 군집화 알고리즘(clustering algorithm)의 CPU간 프론트홀 부하(fronthaul load)를 setup이 진행됨에 따른 누적함수로 나타낸 그래프이다.

[도 4]는 CPU간 UE(User Equipment)의 채널 정보를 교환하는 C-CPU(Cooperated-CPU)의 SE 성능을 나타낸 그래프이다.

[도 5]는 CPU간 UE(User Equipment)의 채널 정보를 교환하지 않는 D-CPU(Distributed-CPU)의 SE 성능을 나타낸 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0014] [도 1]은 본 발명에 따른 셀프리 다중 프로세서(Cell-Free Multiple CPUs) 시스템의 예시를 나타낸 것이다. 본 발명의 시스템에서 CPU는 Q개 각 CPU에 연결되어 있는 AP(Access Point)는 L개 각 AP(Access Point)는 M개의 안테나를 고려하며 UE(User Equipment)는 K개를 가정한다.

[0015] LSFC는 경로손실(pathloss)과 음영화(shadowing)를 반영하며 AP(Access Point) l과 UE(User Equipment) k의 LSFC는 β_{lk}로 표현한다. 또한 UE(User Equipment) k를 지원하는 AP(Access Point)집합을 C_k, CPU 집합을 q_k로 나타낸다. 표현상 편의를 위해서 AP(Access Point)가 어떤 UE(User Equipment)와 연결했는지에 대한 행렬을 $D \in \mathbb{C}^{L\times K}$로 표현하며 AP(Access Point) l과 UE(User Equipment) k가 연결되어 있으면 $D_{lk}=1$로 표현하며 연결되어 있지 않으면 $D_{lk}=0$으로 표현한다. 그리고 각 AP(Access Point)는 연결되어 있는 특정 CPU가 있고 각각

의 CPU는 point-to-point 연결을 가정했다.

[0016] 본 발명에 따른 군집화(clustering)를 수행하기 위해서는, AP(Access Point)들은 UE(User Equipment)의 LSFC, 파일럿(pilot)신호 및 UE(User Equipment)를 지원하는 CPU의 정보를 파악해야 한다. 각 AP(Access Point)들은 군집 내의 사용자들이 서로 다른 파일럿 신호를 사용하도록 조정하여 파일럿 오염(pilot contamination)을 최소화한다.

[0017] 본 발명에서 AP(Access Point)는 채널 추정단계(Channel Estimation Phase)에서, AP(Access Point)가 국부적(local)으로 추정한 UE(User Equipment)의 채널 정보를 CPU로 전달 하고, 하향링크 데이터 전송단계(downlink data transmission phase)에서 AP(Access Point)는 CPU에서 encoding하고 precoding한 유저 정보를 전송하는 구조를 가정했다.

[0018] (1) 각 UE(User Equipment)와 master AP(Access Point) 연결단계

[0019] 군집화(clustering)를 수행하기 이전에 각 UE(User Equipment)는 LSFC(Large Scale Fading Coefficient)가 가장 큰 AP(Access Point)들을 연결해 나가는 과정이다. 이렇게 연결된 AP(Access Point)를 마스터 AP(Master Access Point)라고 한다. 수학식 1은, LSFC가 가장 큰 AP 집합을 나타내며 이는 유저 k에게 최적의 통신성능을 제공하는 마스터 AP를 의미한다.

수학식 1

[0020] $$C_k = \{\arg\max_l \beta_{lk}\}$$

[0021] 여기에서, β_{lk}는 UE k와 AP l사이의 라지스케일 페이딩 코에피션트이며, 유저와 AP 사이의 거리에 의존하는 변수이다.

[0022] 수학식 1을 바탕으로 각 UE에 대해 LSFC가 가장 큰 AP를 선택하여 해당 AP에 연결된 CPU 목록을 획득하고, 이를 기반으로 클러스터를 형성하는 과정을 수행한다. 이렇게 함으로써 클러스터는 각 UE의 특성에 맞게 최적화되며, 전체 시스템의 성능을 향상시키게 된다.

[0023] 이때, UE k를 지원하고 선택된 마스터 AP에 연결된 CPU 목록은 수학식 2로 표현된다.

수학식 2

[0024] $$q_k = \{q : C_k \cap B_q \neq \emptyset\}$$

[0025] 여기서 C_k는 UE k를 지원하는 AP의 CPU 집합이며, $B = \{B_1, B_2, \ldots, B_Q\}$은 네트워크에서 고려하는 AP집합이며, 각 원소 집합은 해당 CPU에 연결된 AP 집합이며, Q는 시스템에서 고려하는 CPU의 개수이다.

[0026] 상기 과정들은, 각 UE 별로 마스터 AP(master AP)와 마스터 AP에 연결된 CPU 목록을 획득하고, 각 사용자 장비(UE)에 대해, 해당 UE를 지원하는 마스터 AP(Master Access Point)를 식별하고, 해당 마스터 AP에 연결된 CPU 목록을 획득하는 절차로 설명될 수 있다.

[0027] (2) 각 AP(Access Point)의 군집화 단계

[0028] 이는 각 AP별로 동일한 파일럿(pilot) 신호를 가지는 "후보 UE 목록"을 획득하는 절차로 설명된다.

[0029] AP(Access Point)에서 군집화(clustering)를 수행할 때 각 파일럿(pilot)별로 선정되는 후보 UE(User Equipment)는 수학식 3으로 표현될 수 있다.

수학식 3

[0030] $$\mathcal{K}_c = \left\{ k: \max_{j \in \mathcal{K}_{\varphi_i}} \beta_{lj} - \beta_{l\mathcal{K}_{\varphi_i}} < \gamma \right\}$$

[0031] 여기서 φ_i는 UE(User Equipment)에게 할당된 ^{i}번째 파일럿(pilot)신호이며 $\mathcal{K}_{\varphi_i}$는 φ_i가 할당된 UE(User Equipment)집합이다. $\mathcal{K}_c$는 $\mathcal{K}_{\varphi_i}$중 최대 LSFC를 갖는 유저와 LSFC 차이가 소정의 임계값 γ를 넘기지 않는 UE(User Equipment) 후보 집합이다.

[0032] 위의 수식으로 표현되는 동일 파일럿(pilot)을 갖는 후보 UE(User Equipment)들을 LSFC가 높은 순으로 정렬한다.

[0033] 그리고 군집화(clustering)를 진행하고 있는 AP(Access Point)에 연결된 CPU의 지원을 받고 있는 UE(User Equipment)가 있는 경우, 해당 UE를 해당 군집(cluster)에 포함시키고 그렇지 않으면 해당 파일럿(pilot)의 군집화(clustering)를 종료한다. 만약 모든 파일럿에 대하여, 연결된 CPU의 지원을 받는 후보 UE(User Equipment)가 없으면 LSFC가 가장 큰 UE(User Equipment)를 cluster에 포함한다. 이 과정은 [도 2]에 순서도로 나타나져 있다.

[0034] 이는 각 AP가 상기 획득한 후보 UE 중 CPU가 동일한 UE를 군집화하는 단계로 설명된다. 이 단계는 CPU 간 프론트홀 부하를 최소화하기 위해 동일한 CPU에 연결된 UE들을 같은 군집으로 배치한다.

[0035] (3) CPU에서 프리코더(precoder) 설계

[0036] 각 CPU마다 UE(User Equipment)를 지원하는 AP(Access Point)들이 지원하는 UE(User Equipment)들의 간섭을 제거하는 P-MMSE(Partial Minimum Mean Squared Error) 프리코더(precoder)를 설계하는 방식을 수학식 4, 5, 6을 들어 설명한다.

[0037] 수학식 4의 S_k는 UE(User Equipment) k를 지원하는 AP(Access Point)와 동일한 지원을 받는 UE(User Equipment)의 집합이며, 수학식 5는 CPU가 설계한 유저 k에 대한 프리코더이다.

수학식 4

[0038] $$S_k = \{ i: D_k D_i^T \neq 0_{L \times L} \}$$

[0039] 여기에서, D_k, D_i는 모든 AP와 각각 UE k와 UE i에 대한 연결여부를 나타내는 벡터이며, 행열곱 $D_k D_i^T$가 0행렬이 아닌 경우에만 i가 k를 지원하는 AP와 동일한 지원을 받는 것을 나타낸다. D_i^T는 유저 i와 연결여부를 나타내는 행렬의 전치행렬(transpose matrix)이다.

[0040] 수학식 4의 UE 집합에 대하여, CPU가 설계한 UE k에 대한 프리코더는 수학식 5로 주어진다.

수학식 5

[0041] $$w_k = p_k \left(\sum_{i \in S_k} p_i \left(D_{kM} \hat{h}_i \hat{h}_i^H D_{kM} + D_{kM} C_i D_{kM} \right) + \sigma^2 I_{LM} \right)^{-1} D_{kM} \hat{h}_k$$

[0042] 여기에서, p_k는 UE k에 할당된 전송 전력, $\hat{h}_i$는 채널의 측정값, C_i는 유저 i의 채널에러행렬, $\sigma^2 I_{LM}$는 노이즈를 나타낸다.

[0043] CPU에서 설계한 UE k의 프리코더는 UE k를 지원하는 AP가 생성하는 간섭과 잡음을 제거하기 위해 신호가 보내지

기 전에 곱해지는 행렬이다.

[0044] 수학식 5에서, D_{kM}은 유저 k에 대한, 안테나 개수가 M인 경우의 다중 안테나 연결 매트릭스로서, 아래 수학식 6으로 정의된다.

수학식 6

[0045] $$D_{kM} = diag(D_{k1} \otimes I_M, D_{k2} \otimes I_M, \dots, D_{kM} \otimes I_M) \in \mathbb{R}^{LM \times LM}$$

[0046] M은 안테나 개수, $D_{k1} \otimes I_M$ 은 유저 k와 AP(Access Point) L 이 연결되어 있으면 I_M, 연결되어 있지 않으면 0 행렬을 가진다.

[0047] 수학식 4, 5, 6에서, S_k는 UE(User Equipment) k를 지원하는 AP(Access Point)와 동일한 지원을 받는 UE(User Equipment)의 집합이다. w_k는 CPU에서 central하게 설계한 P-MMSE 프리코더(precoder)이며 p_k는 channel estimation phase에서 UE(User Equipment) k의 uplink power이다. $\hat{h}_i \in \mathbb{C}^{LM \times 1}$는 UE(User Equipment) i의 추정 채널, $C_i \in \mathbb{C}^{LM \times LM}$는 채널 추정 error, σ^2는 잡음 전력이다.

[0048] CPU에서 프리코더(precoder)를 설계할 때 AP(Access Point)로부터 S_k에 속하는 UE(User Equipment)의 채널 정보를 fronthaul을 통해서 수신하고 간섭을 제거한다. Multiple CPUs 시스템에서 S_k에 속하는 UE(User Equipment)범위를 만족하기 위해서 다른 CPU에 지원받는 UE(User Equipment)의 채널 정보가 필요하다. 따라서 CPU간의 프론트홀 부하(fronthaul load)가 발생하게 된다.

[0049] (4) 전력 할당 알고리즘

[0050] 각 UE(User Equipment)의 전송 전력은 LSFC와 프리코더(precoder)를 고려한 fractional power control algorithm을 적용한 것을 수학식 7 내지 9로 표현할 수 있다.

수학식 7

[0051] $$w'_k = \frac{w_k}{\|w_k\|^2}$$

수학식 8

[0052] $$\omega_k = \max_{l \in C_k} \mathbb{E}\{\|w'_{kl}\|^2\}$$

수학식 9

[0053] $$\rho_k = \rho_{max} \frac{\left(\sum_{l \in C_k} \beta_{kl}\right)^{\nu} \omega_k^{-\kappa}}{\max_{l \in C_k} \sum_{i \in D_l} \left(\sum_{l \in C_i} \beta_{il}\right)^{\nu} \omega_i^{1-\kappa}}$$

[0054] 전송 전력에 대한 설계는 fractional power control algorithm을 사용했다. w'_k는 UE(User Equipment) k의 프리코더(precoder)를 정규화된 프리코더(precoder) vector이다. ω_k는 가장 큰 정규화된 프리코더(precoder)의 전력이다. ρ_k는 CPU에서 설계한 UE(User Equipment) k의 전력이며 ρ_{max}는 AP(Access Point)에서 전송할 수 있는 최대 downlink 전력이다. ν는 $-1 < \nu < 1$, κ는 $0 < \kappa < 1$ 값을 가질 수 있는 scaling factor이다. ν값이

작을수록 UE(User Equipment)간 공정한 전력 분배를 하며, ^{v}값이 클수록 채널 상황이 좋은 UE(User Equipment)에 전력 할당을 많이 한다. $^{\kappa}$는 값이 클수록 UE의 프리코더의 전력이 비대칭일 경우 전력을 적게 할당한다. AP에서 신호를 전송 시에 각 UE에 분배된 전력으로 신호를 전송한다.

[0055] (5) SINR과 spectral efficiency

[0056] 군집화된 채널의 spectral efficiency는 수학식 10 내지 12로 주어진다.

수학식 10

[0057] $$\bar{w}_k = \sqrt{\rho_k}\frac{w_k}{\sqrt{\mathbb{E}\{\|w_k\|^2\}}}$$

수학식 11

[0058] $$SINR_k = \frac{|h_k^H D_k \bar{w}_k|^2}{\sum_{i=1}^{K}|h_k^H D_i \bar{w}_i|^2 - |h_k^H D_k \bar{w}_k|^2 + \sigma^2}$$

수학식 12

[0059] $$SE_k = \left(1 - \frac{\tau_p}{\tau_c}\right)\log_2(1 + SINR_k)$$

[0060] $\bar{w}_k$는 정규화된 프리코더(precoder)와 할당된 전력의 곱이다. $SINR_k$와 SE_k는 UE(User Equipment) k의 SINR과 spectral efficiency를 나타낸다. τ_c는 coherence bandwidth와 coherence time을 가정할 수 있는 coherence interval을 나타내며, τ_p는 파일럿(pilot) symbol 길이를 표현한다.

[0061] <실시 예>

[0062] [도 3]~[도 5]는 본 발명을 이용해 시뮬레이션 한 결과이다.

[0063] 본 발명에서 고려한 시뮬레이션 환경은 $1km \times 1km$ 환경에서 CPU 개수는 $Q = 9$, 각 CPU에 연결되어 있는 AP(Access Point) 개수는 $L = 5$, AP(Access Point)의 안테나 개수는 $M = 4$, UE(User Equipment) 개수는 $K = 100$을 가정했다. 또한 모든 AP(Access Point)와 UE(User Equipment)는 랜덤으로 분포되어 100번의 setup을 구성했다. 그리고 전송 symbol $\tau_c = 200$, 파일럿(pilot) symbol $\tau_p = 10$을 가정했다.

[0064] [도 3]에는 각 AP(Access Point)에서 군집화(clustering)를 수행할 때 threshold 10dB를 가정했을 때 CPU간 교환되는 UE(User Equipment)의 채널 정보의 complex scalar의 개수를 setup에 따라 누적되는 그래프를 나타냈다. 이 때 UE(User Equipment)당 연결된 CPU 개수는 기존 군집화 알고리즘(clustering algorithm)은 2.66개이지만 본 발명의 군집화 알고리즘(clustering algorithm)은 1.65개로 CPU간 UE(User Equipment) 채널 정보의 프론트홀 부하(fronthaul load)를 줄일 수 있음을 보여준다.

[0065] Pilot 전송 시 uplink 전력은 $p_k = 100mW$, data 전송 시 downlink 최대 전송 전력은 $\rho_{max} = 200mW$이다. 그리고 전력 할당 시에 scaling factor는 $v = -0.5$, $\kappa = 0.5$로 UE(User Equipment)간 공정하게 전력 분배를 했다.

[0066] [도 4]는 CPU간 UE(User Equipment)의 채널 정보를 교환하는 C-CPU(Cooperated-CPU)의 SE 성능을 나타낸다. 본 발명에서 제공하는 군집화 알고리즘(clustering algorithm)이 기존 군집화 알고리즘(clustering algorithm)비

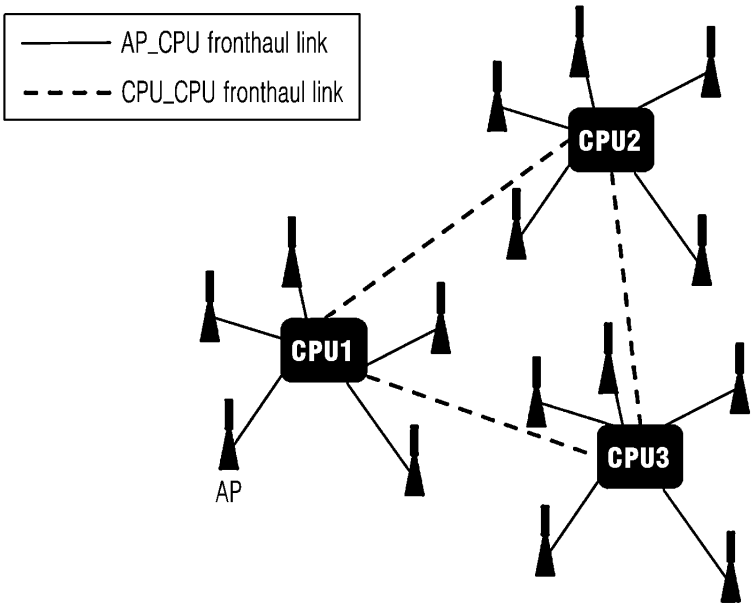
해 성능 열화가 굉장히 적다.

[0067] [도 5]는 CPU간 UE(User Equipment)의 채널 정보를 교환하지 않는 D-CPU(Distributed-CPU)의 SE 성능을 나타낸다. 본 발명에서 제공하는 군집화 알고리즘(clustering algorithm)이 UE(User Equipment)간 간섭을 기존 군집화 알고리즘(clustering algorithm)보다 잘 제거할 수 있기 때문에 향상된 성능을 보여준다.

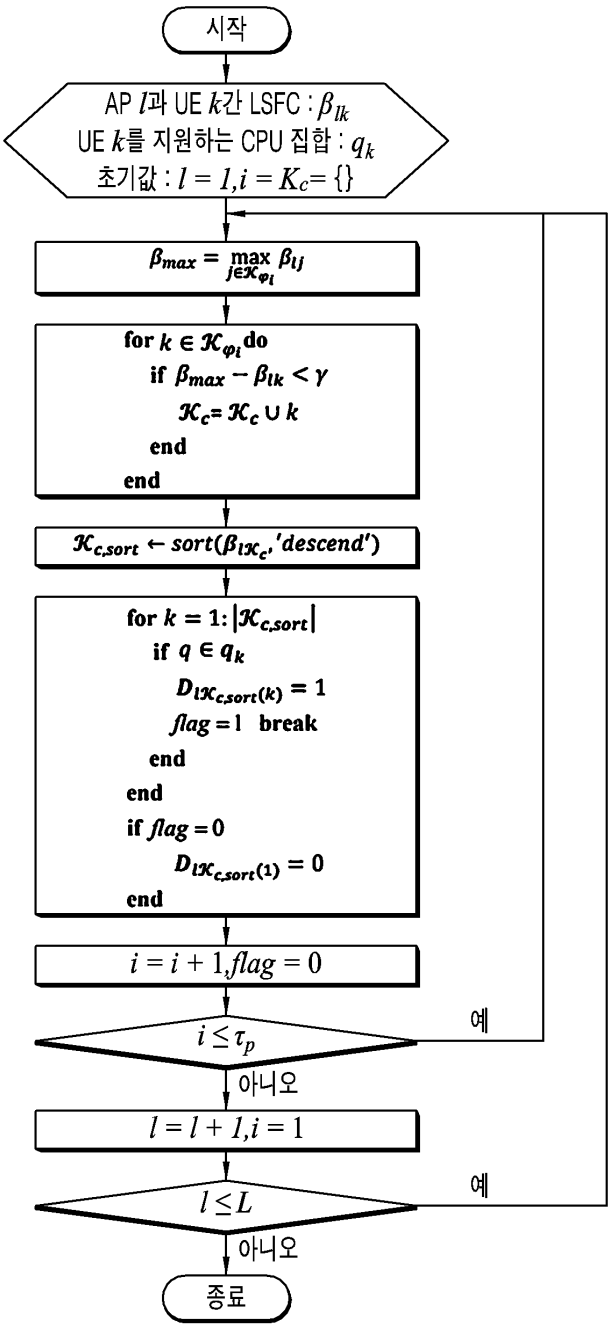
부호의 설명

도면

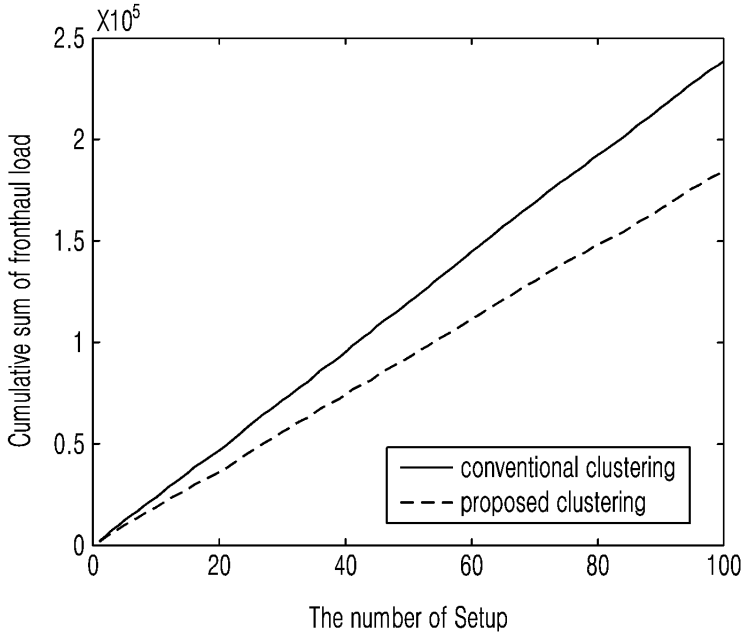
도면1



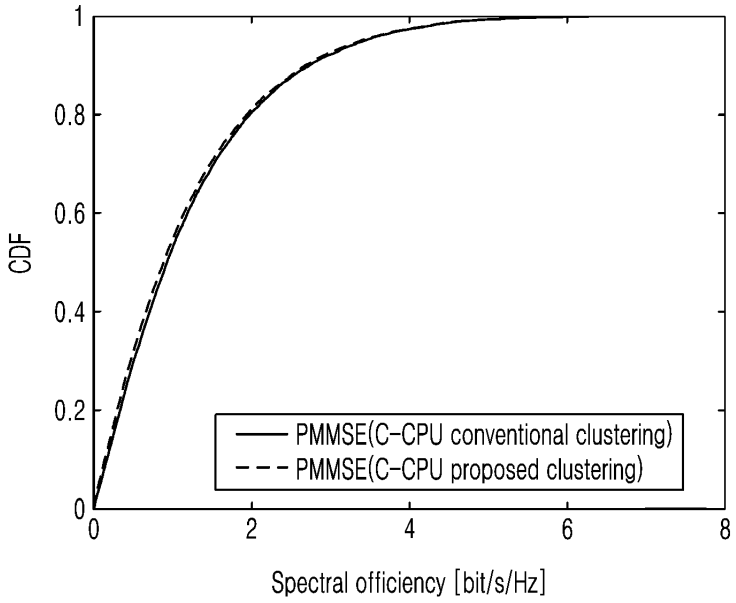
도면2


시작
AP l과 UE k간 LSFC : β_{lk}
UE k를 지원하는 CPU 집합 : q_k
초기값 : $l = 1, i = K_c = \{\}$
$\beta_{max} = \max_{j \in \mathcal{K}_{\varphi_l}} \beta_{lj}$
for $k \in \mathcal{K}_{\varphi_l}$ do
if $\beta_{max} - \beta_{lk} < \gamma$
$\mathcal{K}_c = \mathcal{K}_c \cup k$
end
end
$\mathcal{K}_{c,sort} \leftarrow sort(\beta_{l\mathcal{K}_c}, 'descend')$
for $k = 1 : |\mathcal{K}_{c,sort}|$
if $q \in q_k$
$D_{l\mathcal{K}_{c,sort}(k)} = 1$
$flag = 1$ break
end
end
if $flag = 0$
$D_{l\mathcal{K}_{c,sort}(1)} = 0$
end
$i = i + 1, flag = 0$
$i \leq \tau_p$
예
아니오
$l = l + 1, i = 1$
$l \leq L$
예
아니오
종료

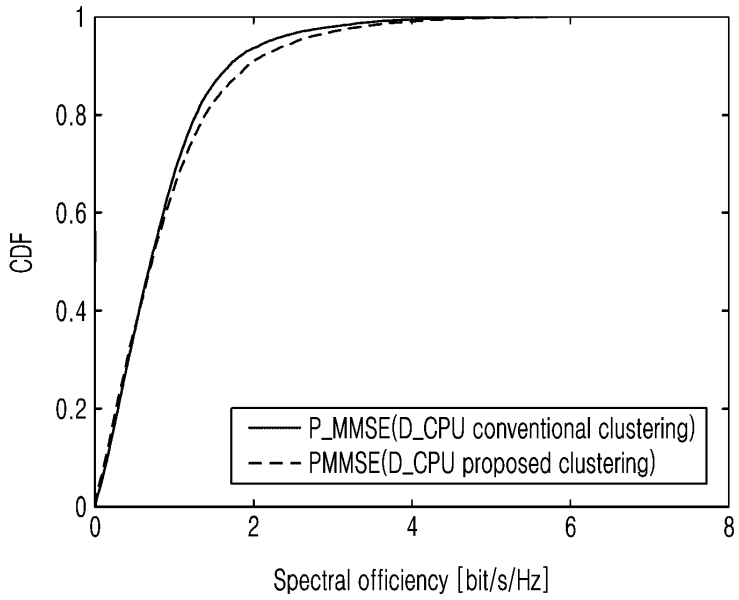
도면3


X10^5
2.5
2
1.5
1
0.5
0
Cumulative sum of fronthaul load
0
20
40
60
80
100
The number of Setup
conventional clustering
proposed clustering

도면4


1
0.8
0.6
0.4
0.2
0
CDF
0
2
4
6
8
Spectral officiency [bit/s/Hz]
PMMSE(C-CPU conventional clustering)
PMMSE(C-CPU proposed clustering)

도면5


1
0.8
0.6
0.4
0.2
0
CDF
0
2
4
6
8
Spectral officiency [bit/s/Hz]
P_MMSE(D_CPU conventional clustering)
PMMSE(D_CPU proposed clustering)