

분산 Near-RT RIC 을 이용한 초저지연 오픈랜 구현

김건, 이성진, 홍인기
경희대학교

gun@khu.ac.kr, ssjj3552@khu.ac.kr, ekhong@khu.ac.kr

Implementation of Ultra-Low Latency Open RAN Using Distributed Near-RT RIC

Geon Kim, Seong Jin Lee, Een Kee Hong
Kyunghee Univ.

요약

본 논문은 Near-RT RIC의 구성 요소를 클라우드, 엣지 컴퓨터에 걸쳐 분산시키는 오픈랜 구조와 이를 최적화하는 RIC Orchestrator를 소개한다. Near-RT RIC을 분산시킨 오픈랜 구조는 xApp의 엄격한 지연시간을 만족하는 동시에 유연한 네트워크 구성을 가능하게 한다. 기존 Orchestrator에서 Near-RT RIC의 구성 요소 배치를 최적화하기 위해 사용되는 휴리스틱 알고리즘의 성능을 분석하고 Tabu Search 알고리즘을 이용한 새로운 최적화 방법을 제안한다.

I. 서론

6G 시대에 접어들면서, 네트워크의 유연성과 성능 향상을 위한 기술적 혁신이 더욱 중요해지고 있다. 이러한 흐름에서 오픈랜은 네트워크 혁신의 중심축으로 자리잡고 있으며 [1], 분산 컴퓨팅 기술의 발전에 의해 더욱 가속화되고 있다 [2]. 본 연구는 6G 환경에서 오픈랜 구조를 최적화하는 방법에 중점을 두고 있으며, 이를 이용해 네트워크 관리의 효율성과 속도를 대폭 개선할 수 있는 방안을 제시한다.

본 연구는 오픈랜 구조 내에서 Near-RT RIC(RAN Intelligent Controller)의 구성 요소를 클라우드 엣지 컴퓨팅 환경에 분산시키는 방법을 탐구한다. 기존에 진행된 선행연구 [3]에서는 Near-RT RIC을 중앙집중적 구조와 분산 구조를 이용해 구현하는 새로운 연구 방향을 제시하였다. 선행 연구에서 분산 구현된 Near-RT RIC 오픈랜의 xApp이 엄격한 지연 시간 요구사항을 만족시키면서 네트워크 구성의 유연성을 제공할 수 있음을 제기하였다. 더 나아가 Near-RT RIC의 분산 구현은 사용자에게 더 가까운 위치에서 데이터 처리와 의사 결정을 수행함으로써 통신 거리를 줄이고 지연 시간을 감소시킨다.

또한 본 연구는 선행연구 [4]에서 사용된 Greedy 알고리즘과는 다른 접근 방식을 채택하여, Near-RT RIC 구성 요소의 최적 배치를 위한 알고리즘으로 Tabu Search 알고리즘을 제안한다. 이 알고리즘은 네트워크 자원의 효율적 사용과 비용 효율성을 동시에 달성하기 위한 것으로, 6G 환경에서의 오픈랜 구조 최적화에 중요한 기여를 할 것으로 기대된다.

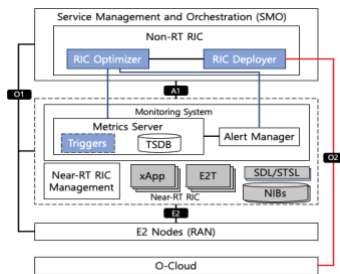


Fig. 1. RIC-Orchestrator 구조

II. 본론

1. 시스템 모델

본 연구는 분산 Near-RT RIC 구조에서 RIC-Orchestrator가 구현하는 기능을 체계적으로 모델링하고, xApp의 지연 시간 요구사항을 만족시키는 동시에 네트워크 비용을 최소화하는 최적화 문제를 해결하는 것을 목표로 한다. Fig. 2에 도시된 E2 노드와 CN(Computing Node)들 사이의 오버레이 통신 네트워크 구조에 기반하며, 각 노드 간의 효율적인 데이터 흐름과 Near-RT RIC 구성요소 분배를 최적화함으로써 문제를 해결할 수

있다. 논문에서 목표로 하는 최적화 문제의 종류는 Mixed-Integer Nonlinear Programming(MINLP) 문제로 정의되며, 이는 NP-Hard 문제임이 이미 선행 연구에서 증명되었다 [4].

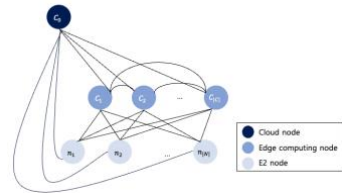


Fig. 2. E2 노드와 CN들 사이 오버레이 통신 네트워크

2. Tabu Search

MINLP 문제를 보다 효율적으로 해결하기 위해, 본 논문에서는 Tabu Search 알고리즘을 제안한다. Tabu Search 알고리즘은 Greedy 알고리즘과 다르게 지역 최적해에서 벗어나 더 넓은 탐색 공간을 확인함으로써 전체적인 비용 최소화를 실현한다.

Tabu Search는 문제 공간을 더 넓게 탐색하면서 공간에서의 이동 경로를 추적하게 된다. 이 알고리즘은 지역 최적해에 갇히지 않도록 임시적으로 금지된 해를 관리하고, 이를 피해 전역 최적해를 찾기 위한 탐색을 하게 된다. 즉, 여러 해에 대한 탐색을 동시에 진행하며 현재 탐색 중인 해의 이동 경로를 기록하게 된다. 이를 통해 이미 방문한 해나 지역 최적해로 되돌아가는 것을 방지하며, 탐색 과정을 다양하게 유지하게 된다. 또한, 임시적으로 타부 리스트에 방문한 해를 기록해 관리함으로써 지역 최적해에 갇히는 것을 방지하게 된다.

- 초기화 단계: 탐색을 시작할 초기 해로 랜덤으로 하나의 configuration을 설정하고, 탐색할 이웃 구조를 정의한다. 또한, 탐색 과정에서 반복하지 않도록 금지된 이동을 기록할 타부 리스트를 초기화한다.
- 최적 노드 선택: 이전에 선택하지 않은 엣지 컴퓨팅 노드들 중에서, 고정비용이 가장 낮은 엣지 컴퓨팅 노드를 고른다.
- 타부 리스트 기록: 선택된 엣지 컴퓨팅 노드들을 타부 리스트에 기록하여, 다음 번에는 다른 엣지 컴퓨팅 노드들을 고려하도록 한다.
- 총 비용 계산: 각 E2 노드에 선택된 엣지 컴퓨팅 노드의 비용을 합산하여 총 비용을 계산한다

3. 최적화 문제 정의

아래의 표는 최적화 수식에 쓰이는 parameter들을 정리한 것이다.

TABLE I. PARAMETERS

기호	의미
ϕ_{Fix}	고정 네트워크 운영 비용
ϕ_{Var}	가변 네트워크 운영 비용
P	Configuration의 집합
x_{pi}	E2 노드 i 에 대해 configuration p 가 선택되었는지를 나타내는 이진 변수
n_i	E2 노드
N	모든 E2 노드들의 집합
$L_i^r, L_i^{r,CA}$ 등	Latency 관련 파라미터들
A_{ch}	xApp 채널의 집합
ρ_a	xApp a 에 대한 최대 허용 latency
$R_{cm}, T_{cm}, D_{cm}, S_{cm}$	컴퓨팅 노드 m 에서 특정 리소스가 실행되고 있는지 나타내는 이진 변수
$c_{cm}^{Proc}, c_{cm}^{Mem}, c_{cm}^{Sto}$	컴퓨팅 노드 m 의 처리, 메모리 및 저장 용량에 대한 파라미터들
A_{cm}^a	xApp a 가 컴퓨팅 노드 m 에서 실행되는지를 나타내는 이진 변수

본 논문에서 풀고자 한 MINLP 최적화 문제의 목적 함수와 제약 조건들은 다음과 같이 기술될 수 있다.

$$\text{minimize } (\phi_{Fix} + \phi_{Var}) \quad (1)$$

$$\sum_{p \in P} x_i^p = 1, \forall n_i \in N \quad (2)$$

$$\mathcal{L}_i^{Cr} + \mathcal{L}_i^{Cr,CA} + \mathcal{L}_i^{CA,CS,CD} + \sum_{a, a' \in A_{ch}} (\mathcal{L}_i^{CA, a'} + \mathcal{L}_i^{a, CA'}) \leq \rho_a, \quad \forall n_i \in N, \forall a \in A \quad (3)$$

$$R_{cm} R^{Proc} + T_{cm} T^{Proc} + S_{cm} S^{Proc} + D_{cm} D^{Proc} + \sum_{a \in A_{cm} \subseteq A} A_{cm}^a A^{a, Proc} \leq c_{cm}^{Proc}, \quad \forall c_m \in \mathcal{V}_C \quad (4)$$

$$R_{cm} R^{Mem} + T_{cm} T^{Mem} + S_{cm} S^{Mem} + D_{cm} D^{Mem} + \sum_{a \in A_{cm} \subseteq A} A_{cm}^a A^{a, Mem} \leq c_{cm}^{Mem}, \quad \forall c_m \in \mathcal{V}_C \quad (5)$$

$$R_{cm} R^{Sto} + T_{cm} T^{Sto} + S_{cm} S^{Sto} + D_{cm} D^{Sto} + \sum_{a \in A_{cm} \subseteq A} A_{cm}^a A^{a, Sto} \leq c_{cm}^{Sto}, \quad \forall c_m \in \mathcal{V}_C \quad (6)$$

(2)식은 네트워크 운영 중 선택해야 하는 configuration 집합 내에서 특정 configuration 채택해야 한다는 제약을 나타낸다. (3)식은 xApp 실행되는 동안 발생하는 지연 시간이 사전에 정의된 특정 임계값을 넘지 않아야 한다는 조건을 제시한다. 이는 네트워크 내에서 xApp의 성능을 보장하기 위한 중요한 제약 조건으로, 사용자 경험 및 서비스 품질을 유지하는데 필수적이다. 마지막으로 (4), (5), (6)식은 각 컴퓨팅 노드에서 실행되는 프로그램들이 해당 노드의 처리 용량을 초과하지 않도록 제한한다. 이러한 제약조건은 네트워크 노드의 자원 관리와 부하 분산을 조절하기 위한 것으로, 각 노드가 과도한 부하로 인한 성능 저하를 방지하고 안정적인 운영을 유지하기 위해 필수적인 조건이다.

III. 실험결과

컴퓨팅 노드의 수를 점진적으로 증가시키며 수행한 실험 결과를 통해 본 논문에서 제안한 Tabu Search 알고리즘은 기존 Greedy 알고리즘에 비해 비용을 더 효과적으로 절감함을 확인할 수 있었다. 특히 컴퓨팅 노드의 개수가 증가할수록 Tabu Search는 네트워크 운영 비용이 완만하게 증가하는 경향을 보였다. 이러한 결과를 통해 노드 수가 증가할 때 Tabu Search 알고리즘이 더 효과적으로 사용할 수 있다는 점을 확인할 수 있었다.

다만, Tabu Search 알고리즘을 적용할 경우, 노드의 수가 많아질수록 처리해야 할 데이터의 양과 복잡성이 증가함에 따라 계산 시간이 늘어나는 현상도 관찰되었다. 이는 네트워크 운영의 실시간성이 중요한 상황에서는 부담이 될 수 있다. 특히 대용량의 데이터를 빠르게 처리하고 분석해야 하는 상황에서는 Greedy 알고리즘보다 상대적으로 높은 지연 시간이 발생할 가능성이 있다. 따라서, 이러한 계산 시간 증가가 네트워크의 성능에 미치는 영향을 최소화하기 위한 방향이 필요하며, 이를 위해 알고리즘의 파라미터를 조정하거나, 계산 효율성을 개선할 수 있는 방법에 대한 추가적인 연구가 요구됨도 확인할 수 있었다. 예를 들어, 병렬 계산 방식의 도입이나 알고리즘의

휴리스틱을 개선함으로써 Tabu Search의 계산 효율성을 증가시킬 수 있다.

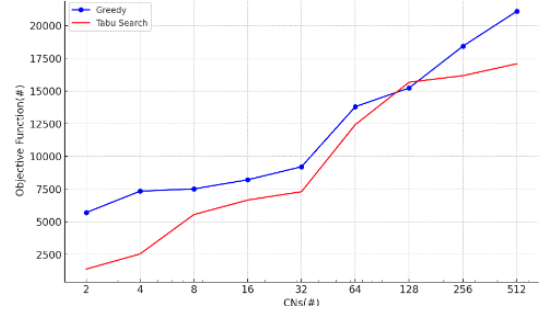


Fig. 3. 기존 알고리즘과 제안 알고리즘의 metric

IV. 결론

실험 결과, 기존의 Greedy 대신 Tabu Search 알고리즘을 적용한 Orchestrator가 현저하게 낮은 네트워크 운영 비용을 산출하였다. 더불어 연구 결과를 통해 Tabu Search 알고리즘이 컴퓨팅 노드의 개수가 증가함에 따라 더 효율적인 네트워크를 구성할 수 있는 가능성을 시사하였다. 그러나 노드 수가 증가함에 따라 Tabu Search 알고리즘이 Greedy 알고리즘에 비해 약간의 계산 시간 증가를 보였고, 이는 특정 상황에서 네트워크 운영에 부정적인 영향을 미칠 수 있음을 확인할 수 있었다. 이를 방지하기 위해 Tabu Search 알고리즘의 효율성을 높이기 위한 파라미터 조정에 대한 추가 연구가 필요하다.

ACKNOWLEDGMENT

“본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터육성지원사업의 연구결과로 수행되었음” (IITP-2021-0-02046*)

“이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 정보통신방송기술국제공동연구사업 지원을 받아 수행된 연구임” (No.RS-2022-00207387, 5G Open RAN 기반 지능형 네트워크 슬라이싱 기술 개발)

참고 문헌

- [1] A. Garcia-Saavedra and X. Costa-Pérez, “O-RAN: Disrupting the Virtualized RAN Ecosystem,” *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 96-103, 2021.
- [2] O-RAN Alliance, “O-RAN Alliance,” 2022. [Online]. Available: <https://www.o-ran.org/>
- [3] M. Dryjanski and A. Kliks, “The O-RAN Whitepaper 2022 RAN Intelligent Controller, xApps and rApps,” RIMEDO Labs, Tech. Rep., 2023.
- [4] G. M. Almeida et al., “RIC-O: Efficient placement of a disaggregated and distributed RAN Intelligent Controller with dynamic clustering of radio nodes,” in *IEEE Journal on Selected Areas in Communications*, doi: 10.1109/JSAC.2023.3336159.