Slice-Aware Resource Calendaring in Cloud-based Radio Access Networks

Zeinab Sasan  
Computer Engineering Department  
Amirkabir University of Technology  
Tehran, Iran  
z.sasan@aut.ac.ir

Siavash Khorsandi  
Computer Engineering Department  
Amirkabir University of Technology  
Tehran, Iran  
khorsandi@aut.ac.ir

Abstract—Network slicing has been introduced in 5G/6G networks to address the challenge of providing new services with different and sometimes conflicting requirements. With SDN and NFV technologies being used in the design of 5G and 6G wireless network slicing, as well as the centralization of control over these technologies, new services such as resource calendaring can also be used in wireless networks. In bandwidth calendaring, traffic with a low latency sensitivity and a high volume is shifted to later time slots so that applications with a high latency sensitivity can be served instead. We discuss how to calendar radio resources in the C-RAN architecture, which also makes use of network slicing. A model of the problem is developed as an ILP problem and two heuristic algorithms are proposed for solving it due to complexity of optimal solution. Observations have shown that when resources are shared between tenants, the number of accepted requests increases.

Index Terms—Network Slicing, 5G/6G, Resource Calendaring, Resource Allocation, C-RAN, Softwarized Networks

I. INTRODUCTION

One of the major challenges in the next generation of wireless networks is the emergence of new services with different requirements, so that sometimes these requirements are in conflict with each other [1]. The solutions to the this challenge in wireless networks is the introduction of network slicing technology.

The concept of network slicing enables the creation of multiple logically separate networks on top of a common physical infrastructure platform, allowing a flexible ecosystem to be created to facilitate technical and business innovation, utilizing physical and logical network resources to form a programmable and open multi-tenant network environment [2]. As a result, in architecture based on network slicing, any application can be served according to its requirements. Network slicing enabler technologies include SDN and NFV, which refer to centralized networking capabilities as well as the separation of software from hardware.

In addition, the Cloud-based RAN (C-RAN) architecture separates the Base Band Units (BBUs) from the Remote Radio Heads (RRHs) in base stations and pools them together in a centralized manner [3]. This Centralization allows mobile operators to effectively allocate resources and reduce operating and capital costs [4].

The move of the wireless networks towards softwarization and centralized management and configuration with SDN and NFV technologies has made it possible to introduce new services in wireless networks. One of these important services is bandwidth calendaring. Bandwidth calendaring refers to shifting some large data transfers with fewer real-time constraints to times when the network is less congested [5]. Applications updates are one example of bandwidth calendaring, such as updates that are not sensitive to latency but generate lots of traffic, and based on calendaring methods, these updates are made during the evening hours.

Bandwidth calendaring exploits the knowledge or estimation of future arrivals of requests and future demands in an optimal way in the network. Calendaring in wireless networks will optimize the use of resources and improve customer service. The network operator can also maximize revenue and profit by optimizing the use of resources.

In most of the research work done so far, the issue of bandwidth clearance in data center networks has been investigated. In recent years, several research projects have also focused on resource calendaring in wireless networks. But so far, the use of calendaring in next generation wireless networks based on network slicing has not been addressed. To the best of our knowledge, this paper is the first to study the Slice-Aware Resource Calendaring.

In this paper, we examine resource calendaring in wireless networks that use the C-RAN architecture and network slicing. In the defined problem, the network operator decides on the acceptance of user requests belonging to tenants and slices, as well as the scheduling and resource allocation to them. There are several tenants with eMBB and eMBBRLLC slices. In the eMBB slice a lot of resources are required, but delays are not a concern. The eMBBRLLC slice requires a lot of resources and is delay sensitive [6]. Each of the tenants has also reserved a portion of the operator’s resources at the time of contracting.

It is the network operator’s intention to serve tenants with different slices so that social welfare is maximized and the...
maximum number of requests is accepted. In addition, the operator should also try to consider resource constraints, slice and tenant level requirements. We model this scenario as an ILP problem, and due to the complexity of its optimal solution, two heuristic algorithms are proposed. According to numerical results, a heuristic method based on resource sharing results in a higher percentage of requests being accepted.

This paper is organized as follows. Section II discusses related works. Section III presents the system model. Section IV describes two efficient algorithms to compute sub-optimal yet good solutions for the slice-aware resource calendaring problem. Section V illustrates numerical results and performance evaluation. Finally, Section VI concludes the paper and briefly discusses the future research issues.

II. RELATED WORKS

During the past, bandwidth calendaring was widely used to move large amounts of data from data centers to WANs, and SDN made it easier. As a result, many of the work on this topic has been carried out using SDN technology in data centers and wide area networks.

The paper [5] describes how a carrier network operator must, at minimum cost, accommodate the demands for predetermined, but time-varying, bandwidth. A few of the demands may be flexible, i.e., can be scheduled during a certain time period. Based on column generation, the authors propose a salable problem decomposition.

An online version of the bandwidth calendaring is presented in the paper [7]. At the moment of decision making, the amount of resources requested is unknown and information such as offline problems is not fully provided to the operator. Online optimization methods are used to solve this problem.

Calendaring of radio resources in C-RAN is considered in the papers [4, 8]. The objective of the network operator is to maximize social welfare and ensure resource requirements and delays per request. It should be noted that in this study, neither network slicing nor tenant and slice concepts were examined and our work is the first to consider calendaring in softwarized wireless networks.

The paper [9] provide an optimization framework that considers several key aspects of the resource allocation problem with cooperating Mobile Edge Computing nodes. Proposed model jointly optimizes (1) the user requests admission decision (2) their scheduling, also called calendaring (3) and routing as well as (4) the decision of which nodes will serve such user requests and (5) the amount of processing and storage capacity reserved on the chosen nodes.

III. SYSTEM MODEL

In this paper, we consider an operator that manages users requests in various slices of the C-RAN. Each slice and user belongs to a tenant. When concluding a contract with the operator, tenants agreed to the number of resources they reserved. So that the tenant’s reserved resources equal $R^k$. Additionally, each tenant has two types of slices of eMBB and eMBBRLLC. First slice requires a lot of resources, but is not sensitive to latency, the second slice requires a lot of resources and is sensitive to latency. In general, eMBB slice requests can be shifted to later time slots. $N$ is equal to the decision interval and depending on the scenario of 5G/6G networks, this timing window can be in the range of milliseconds.

The five fields of a request are as follows:
- $t^k_n$: The arrival time of request $k$.
- $r^k$: The number of resources request $k$ requires.
- $d^k$: Duration of request $k$.
- $u^k_n$: Request $k$ Utility Function in time slot $n$.
- $k^t$: A binary number that indicates that request $k$ belongs to tenant $t$

According to the arrival time of each request, a utility function is defined for that request. The function is non-ascending and depends on the type of request. An eMBB request has a utility function that is defined from the time of arrival until the end of its deadline and equals 1. The request must be serviced immediately if it is of the eMBBRLLC type, and the utility function is 1 only when the request arrives.

Table I summarizes the parameters and decision variables.

We use the following sets to model and solve this problem:
- $R = \{1, \ldots, R\}$: Set of the resources.
- $K = \{1, \ldots, K\}$: Set of the requests.
- $N = \{1, \ldots, N\}$: Set of the time slots.

| Table I | PARAMETERS AND VARIABLES DEFINITION |
|---------|-------------------------------------|
| $K$     | General                             |
| $R$     | Total number of requests            |
| $N$     | Total number of available resources |
| $t$     | Total number of tenants             |
| $U^t_n$ | Utility function of tenant $t$      |
| $R^t$   | Number of resource reserved by tenant $t$ |
| $t^k_n$ | Time slot in which the request $k$ arrives |
| $r^k$   | Number of resource requested by $k$ |
| $d^k$   | Duration of request $k$             |
| $u^k_n$ | Utility function of request $k$     |
| $k^t$   | Binary parameter that tells if the $k$-th request is belong to tenant $t$ |

This paper is based on the following assumptions:
- As soon as a request is initiated, the execution process cannot be interrupted.
- Over time, the amount of resources a request requires remains the same.
The decision-making process occurs after all requests have been received and the problem is offline (through forecasting method or previous knowledge).

Using Integer Linear Programming (ILP), we formulate the optimal calendaring problem. A C-RAN operator allocates resources with the objective of maximizing social welfare while considering resource constraints, delays, and the resources reserved for each tenant. In the case of the request $k$ being accepted at time slot $n$, the decision variable $x_n^k$ is 1, otherwise 0. The decision variable $r_{n,k}$ will be 1 if resource $j$ is assigned to request $k$ at time slot $n$, otherwise it will be 0.

\[
\begin{align*}
\text{Maximize} & \sum_{t \in T, k \in K, n \in N : n \geq t_0^k} k_t x_n^k u_n^k \\
\text{subject to} & \sum_{n = t_0^k}^{N - d_t^k - 1} x_n^k \leq 1 \quad \forall k \in K \\
\sum_{k \in K, n \geq t_0^k} r_{n,k}^j \leq 1 \quad \forall n \in N, j \in R
\end{align*}
\]

A. Dedicated Resource Allocation (DRA)

According to this solution, first of all, the reserved resources are assigned to each tenant separately. Reserved resources are then assigned to each of the tenant’s requests. First, all requests that are not accepted are sorted by priority. A request with a closer deadline and more requested resources has a higher priority. If the resources required for the request are available during the $d_t^k$ period after the start of the time slot $n$, this request will be accepted and otherwise rejected. Algorithm 1 shows the algorithm for this solution.

```
Algorithm 1: Dedicated Resource Allocation (DRA)

for All Tenants $T$ do
    Allocate Dedicated Resources to Tenant $t$;
    Find the Requests of Tenant $t$;
    for All Time Slots $N$ do
        Compute Priority for Not-Accepted Requests of Tenant $t$ in Time Slot $n$;
        Sort Priority-List of Tenant $t$ ;
        for All Requests in Priority-List do
            if Tenant $t$ Reserved Resources $R_t$ is Insufficient for Acceptance of Request $k$ then
                Reject Request $k$;
            end
            Find Free Resource for Request $k$ in range $n:n+d_t^k$;
            if # of Candidate Resources is Sufficient for Acceptance of Request $k$ then
                Request $k$ Accepted and $x_n^k=1$;
                for $r$ in range(0,$r_t^k$) do
                    for $n'$ in range($n$, $n+d_t^k$) do
                        $r_{n'}^k = 1$;
                    end
                end
                if Request $k$ Accepted then
                    Tenant $t$ Resource-Usage ++;
                end
            end
        end
    end
end
```

B. Sharing-based Resource Allocation (SRA)

This method does not allocate resources precisely to each tenant. The unaccepted requests in each time slot are sorted by the priority. A request with a closer deadline and more
each tenant’s reserved resources. To find a compromise between sharing resources and isolating for each tenant, and this is a negative point. It is necessary for resource sharing to violate the reserved resource constraints with others who need them. It is important to remember that the SRA method, tenants with unused resources share them with others who need them. This request can be accepted if the other tenant has sufficient and free resources. The algorithm for this solution is shown in Algorithm 2.

V. PERFORMANCE EVALUATION

In this section, the proposed solutions are evaluated for their efficiency. As mentioned in the previous section, the main approach in the proposed solution is based on resource sharing. Dedicated Resource Allocation (DRA) is also presented and their performance are compared with each other. On the Google Colab platform, these algorithms have been implemented and evaluated. In this implementation, the requests are distributed among the 3 tenants uniformly, and each tenant’s requests are of the eMBB type with 0.5 probability and eMBBRLLC type with 0.5 probability. Table II presents the other parameters used in the simulation.

The number of requests was changed first. As shown in Figure [1] The number of requests accepted by the SRA solution is greater than the number of requests accepted by the DRA solution since the sharing method utilizes the resources available to other tenants.

It is important to note in this section that in general, the sharing method is more efficient than the deductible approach. Nevertheless, as the number of requests increases, the sharing method becomes less efficient and the results of the two algorithms become more similar.

In the following, the number of requests is considered constant and the number of resources available in each time slot is changed. According to Figure [2] as the number of resources increases, the number of requests accepted increases as well. Due to resource sharing between tenants, the SRA algorithm still has a higher number of accepted requests than the DRA algorithm.

In Figure [3] it is shown how the SRA method increases the number of resources used by each tenant compared to the DRA method, given a certain number of requests and resources. In the SRA method, tenants with unused resources share them with others who need them. It is important to remember that resource sharing can violate the reserved resource constraints for each tenant, and this is a negative point. It is necessary to find a compromise between sharing resources and isolating each tenant’s reserved resources.

VI. CONCLUSION

Slice-Aware calendaring of radio sources in C-RAN is examined in this paper. According to C-RAN’s mission, the operator intends to serve a range of slices with varying requirements. In addition, each tenant’s resource constraints, delays, and resources should be taken into account. The problem is modeled as an ILP problem, which is NP-Hard with a complex optimal solution. Therefore, two heuristic solutions have been proposed for this problem. In the resource sharing algorithm, the number of accepted requests increases. It should be noted that a compromise must be found between sharing resources and the number of resources reserved for each tenant. The online version of this problem can be reviewed and solved in future work so that all details about the problem aren’t

---

**Algorithm 2: Sharing-based Resource Allocation (SRA)**

```python
for All Time Slots N do
    Compute Priority for All of Not-Accepted Requests;
    for All Requests in Priority List do
        Find Tenant t of Request k;
        if $R^t + r^k > Tenant \text{ t} \text{ Reserved-Resources}$ then
            Find Tenant $t'$ with Not-Fulled Reserved-Resources;
            Find Free Resource for Request $k$ in range $n:n + d^k$;
            if # of Candidate Resources is sufficient for Request $k$ then
                Request $k$ Accepted and $x^k_n = 1$;
                for $r$ in range($0, r^k$) do
                    for $n'$ in range($n, n + d^k$) do
                        $r^k_{n'} = 1$;
                    end
                end
                if Request $k$ Accepted then
                    Tenant $t$ Resource-Usage ++;
                end
            end
            end
        end
        else
            Find free Resource for Request $k$ in Range $n:n + d^k$;
            if # of Candidate Resources is Sufficient for Request $k$ then
                Request $k$ Accepted and $x^k_n = 1$;
                for $r$ in range($0, r^k$) do
                    for $n'$ in range($n, n + d^k$) do
                        $r^k_{n'} = 1$;
                    end
                end
                if Request $k$ Accepted then
                    Tenant $t$ Resource-Usage ++;
                end
            end
        end
    end
end
```
TABLE II
SETTIGNGS AND PARAMETERS

| Parameter                      | (a)          | (b)          |
|--------------------------------|--------------|--------------|
| Number of time slots           | \( N = 10 \) | \( N = 10 \) |
| Number of tenants              | \( T = 3 \)  | \( T = 3 \)  |
| Number of resources available in a time slot | \( R = 20 \) | \( R = [0; 100] \) |
| Number of requests             | \( K = [0; 100] \) | \( K = 50 \) |
| Percentage of eMBB requests    | 50\%         | 50\%         |

**Per Request Parameters**

| Parameter                  | Value |
|----------------------------|-------|
| Arriving time \( t_{0} \) generated from | \( u.d \in [1:5] \) |
| Number of resources requested by request \( k \) generated from | \( u.d \in [1:5] \) |
| Request duration \( d_k \) generated from | \( u.d \in [1:5] \) |
| First Tenant Share of Resources | 0.2 |
| Second Tenant Share of Resources | 0.2 |
| Third Tenant Share of Resources | 0.6 |

available in each time slot.

**REFERENCES**

[1] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, G. S. Paschos, The algorithmic aspects of network slicing, IEEE Communications Magazine 55 (8) (2017) 112–119.

[2] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, H. Flinck, Network slicing and softwarization: A survey on principles, enabling technologies, and solutions, IEEE Communications Surveys & Tutorials 20 (3) (2018) 2429–2453.

[3] F. Z. Morais, C. A. da Costa, A. M. Alberti, C. B. Both, R. da Rosa Righi, When sdn meets c-ran: A survey exploring multi-point coordination, interference, and performance, Journal of Network and Computer Applications 162 (2020) 102655.

[4] J. Elias, F. Martignon, M. Morcos, L. Chen, T. Chahed, Radio resource calendaring in cloud-based radio access networks, in: 2018 Wireless Days (WD), IEEE, 2018, pp. 207–212.

[5] L. Gkatzikis, S. Paris, I. Steiakogiannakis, S. Chouvardas, Bandwidth calendaring: dynamic services scheduling over software defined networks, in: 2016 IEEE International Conference on Communications (ICC), IEEE, 2016, pp. 1–7.

[6] A. Yazar, S. Dogan-Tusha, H. Arslan, 6g vision: An ultra-flexible perspective, ITU Journal on Future and Evolving Technologies 1 (1) (2020) 121–140.

[7] M. Dufour, S. Paris, J. Leguay, M. Draief, Online bandwidth calendaring: On-the-fly admission, scheduling, and path computation, in: 2017 IEEE International Conference on Communications (ICC), IEEE, 2017, pp. 1–6.

[8] M. Morcos, J. Elias, F. Martignon, T. Chahed, L. Chen, On efficient radio resource calendaring in cloud radio access network, Computer Networks 162 (2019) 106862.

[9] B. Xiang, J. Elias, F. Martignon, E. Di Nitto, Resource calendaring for mobile edge computing: Centralized and decentralized optimization approaches, Computer Networks 199 (2021) 108426.