Performance Isolation for 5G RAN Slices Across Multiple Interfering Cells

Muhammad Taimoor Tariq
University of Illinois, Urbana-Champaign (UIUC), USA

Yongzhou Chen
University of Illinois, Urbana-Champaign (UIUC), USA

Haitham Hassanieh
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Radhika Mittal
University of Illinois, Urbana-Champaign (UIUC), USA

Abstract

Radio Access Network (RAN) slicing, a key 5G feature, enables different slices (i.e. tenants or applications) to share the same physical network infrastructure while pursuing diverse objectives such as fairness, prioritization, or maximizing throughput. Each slice is allocated a share of radio resource blocks (RBs), which it further schedules among its users as per its own performance objective. In this paper, we identify the unique challenges that arise when performing RAN slicing in today's multi-cell deployments that require a mechanism for managing interference among cells. We highlight how interference management decisions, that can be easily made in the absence of slicing (where all users share a common objective set by the network operator), become challenging with 5G slicing where we must respect the individual objectives of multiple slices, while retaining performance isolation across slices. We present a system, RadioNinja, that tackles this challenge through a unique decision-making framework that allows different slices to independently contribute towards interference management decisions. RadioNinja further employs a series of techniques to make such decisions within tight RAN scheduling budget of hundreds of microseconds. Trace-driven simulations with real-world channel measurements show that RadioNinja improves slice-level objectives (e.g., throughput, fairness, flow completion times) by 20–60% over state-of-the-art baselines, while consistently meeting sub-millisecond decision deadlines.

2012 ACM Subject Classification Networks → Wireless access points, base stations and infrastructure

Keywords and phrases Cellular Networks, Resource Management, RAN Slicing, Interference Management

Digital Object Identifier 10.4230/OASIcs.NINeS.2026.2

1 Introduction

Network slicing [6, 49, 44, 22, 26] is a key architectural feature of 5G cellular networks. It enables cellular network operators to divide their network resources among different "groups of users" or services (referred to as *slices*) as per their respective SLAs (Service Level Agreements). Each slice then further sub-divides its share of resources among its own users as per its own policy or performance objective. In a way, 5G slicing crucially extends the notion of multi-tenancy to cellular networks, with each slice acting as a tenant. Slices can represent mobile virtual network operators (MVNOs), such as GoogleFi [27] or Cricket [14], that leverage the physical infrastructure of network operators such as AT&T and Verizon [25, 24]. They may also represent enterprise networks, application-driven slices (e.g., AR/VR, industrial control, or video analytics), campus deployments with heterogeneous workloads, or 3GPP-defined service categories such as enhanced Mobile Broadband (eMBB) and Ultra-Reliable Low-Latency Communications (URLLLC) [46, 18, 21, 55, 19]. These

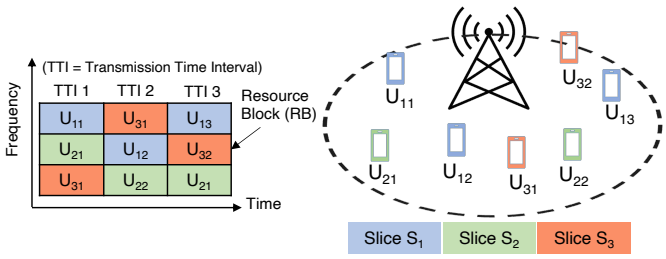


Figure 1 Radio resource blocks (RBs) at a base-station represented as a 2-D grid along time and frequency axes. The time slot spanning an RB is referred to as a TTI. The RBs are scheduled across users belonging to different slices (three slices depicted in this example). User U_{ij} belongs to slice S_i.

slices share the same physical infrastructure, while pursuing their own diverse performance objectives (e.g., fairness among all customers of an MVNO, maximizing throughput for a slice comprising a team of drones, an enterprise prioritizing certain users, etc). A key goal of slicing is to support such diverse performance objectives of each slice, while maintaining their SLAs, and ensuring performance isolation between slices.

Network slicing is particularly crucial at the Radio Access Network (RAN) where users across all slices share limited radio resources at a base-station. These radio resources are split along the time and frequency axes into units called *resource blocks (RBs)*, as shown in Fig. 1. A user's performance (i.e. its throughput or data rate) depends on the amount of RBs assigned to it, as well as the quality of the wireless channel associated with those RBs. This channel quality (that determines the data rate per RB) depends on various factors including the user's distance from the base-station, obstacles, and potential interference from neighboring base-stations transmitting on the same frequency (i.e. on the same RB). A slice's SLA with the network operator governs its resource *quota*, i.e. number of RBs allocated to the slice at a base-station [12, 34]. The slice then schedules its allocated RBs across its users as per its own objective, taking the associated channel qualities into account.

RAN slicing has primarily been explored in the context of a single base-station (or cell) [24, 35, 12]. But today we are seeing a sharp increase in *multi-cell deployments* [58, 50], where the capacity and coverage of a single base-station (macro-cell) is scaled by deploying multiple *small* cells in its vicinity [47, 42, 5, 8, 9], as shown in Fig. 2. Such multi-cell deployments are crucial for keeping up with increasing demands for cellular capacity. However, when two neighboring cells simultaneously transmit on the same frequency, the interference between them can severely degrade the channel quality (and, thereby, the throughput) of users in overlapping coverage regions. This necessitates some mechanism for interference management.

Prior work has explored RAN slicing and interference management in isolation but today's scale and applications demands require deploying both of them in tandem. In this paper, we identify and tackle the unique challenge that arises in such a setting – when we perform RAN slicing in conjunction with interference management in multi-cell deployments.

To understand the challenge, let us first consider how interference can be managed without any slicing, where all users share a single global objective set by the network operator (typically proportional fairness among all users). Interference management in such a "no slicing" setting has been explored by prior work [62, 54, 51]. A common strategy is to mute the transmission of certain cells on specific RBs [4, 28, 48]. As depicted in Fig. 2, muting the transmission of an RB (say the second RB at macro-cell C_1 in our example) ensures

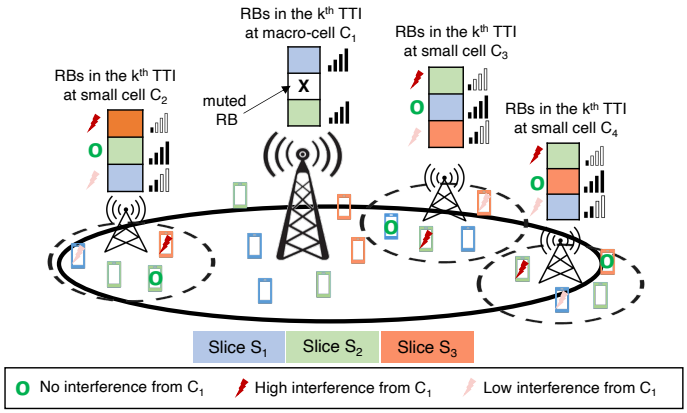


Figure 2 Typical multi-cell deployment with a macro-cell (C_1) and small cells (C_2, C_3, and C_4). The figure depicts how RBs at a given TTI are scheduled across users belonging to three different slices at each cell. Muting the transmission of an RB at a cell (say, the second RB at C_1, as shown) ensures that users scheduled on that RB at neighboring cells do not experience interference from C_1. The amount of interference from C_1 experienced by a user in a neighboring cell depends on the user's proximity to C_1, and impacts that user's *channel quality* (illustrated by signal-strength bars).

that users at other cells scheduled on that RB do not experience any interference from C_1. Deciding whether or not to mute an RB r on a cell C requires weighing the cost of muting incurred by users at cell C (due to unavailability of muted RBs) versus its benefit enjoyed by users at cells neighboring C (due to boosted channel quality caused by reduced interference). It is relatively straightforward to reason about such a trade-off without slicing. The benefit vs. cost of a muting decision can be analyzed based on whether or not it improves the network operator's global objective.

However, with 5G slicing, cellular users do not share a single global objective. Instead, each slice optimizes for its own objective, which is incomparable with other slices' objectives. The decision to mute RB r on a cell C might seemingly benefit a user of slice S_i at a neighboring cell, but hurt a user of another slice S_j (which would have otherwise been scheduled on RB r at cell C) hence violating the performance isolation promised by slicing. So then how do we reason about such a muting decision while supporting diverse slice objectives, and without violating performance isolation across slices?

Our system, RadioNinja, addresses this challenge by introducing a unique form of cost-benefit analysis that allows slices to *independently contribute* towards a muting decision based on their own objectives, without hurting the performance of any other slice. RadioNinja restricts the penalty of muting an RB at a cell C to the K slices that benefit from the muting decision, by reducing each of their quotas at cell C by $1/K$ RBs. This enables us to assess the cost (reduced quota at cell C) versus the benefit (reduced interference at nearby cells) independently for each of the K slices as per the slice's objective. The slice pays the penalty only if the muting decision improves its objective. RadioNinja mutes this RB at cell C only if one or more slices are willing to pay the penalty.

RadioNinja must tackle multiple challenges to realize this approach. The first challenge is identifying the set of K slices for which the benefits of muting the RB outweigh the costs. The cost incurred by a slice ($1/K$ RBs) depends on the number of benefiting slices K itself, creating a chicken-and-egg problem. RadioNinja tackles this through an iterative approach – starting with an initial set of slices that presumably benefit from the muting decision, and repeatedly updating the set after conducting the cost-benefit analysis for each such slice. The

second challenge stems from the fact that muting decisions are inter-twined with scheduling decisions – whether or not an RB at a cell is muted influences the channel quality (and therefore the scheduling decision) on that RB at neighboring cells. The changes in slice quotas caused by a muting/scheduling decision for an RB further has a ripple effect on how the *subsequent* RBs at each cell are scheduled. Therefore, in order to accurately assess the cost and benefit of muting an RB, RadioNinja must ideally re-run the scheduler for all the remaining RBs at each cell. However, doing so is prohibitively expensive – the joint muting and scheduling decisions must be made within tight time slots (i.e. within a TTI, which ranges from 250μs to 1ms in 5G). RadioNinja handles this challenge by employing a series of techniques that effectively approximate the benefit and cost of a muting decision without re-running the scheduler, as detailed in §4. Our approximation techniques also help tackle a third challenge – computing the penalty of reducing a slice's quota by a fraction of an RB (despite the fact that RBs are allocated to users as a whole and cannot be split).

We implement RadioNinja in an open-source RAN simulator [45] and collect real-world traces to perform trace-driven evaluations in §6. Our evaluation, across a variety of realistic scenarios spanning hundreds of users split across multiple slices with diverse objectives, shows how RadioNinja results in 20-60% improvement in the performance of individual slices (on their desired performance objective) when compared to several baselines. Our comparative baselines include (i) a RAN slicing system that independently schedules RBs at each cell without any interference management or muting [12], (ii) a baseline that makes its muting decision based on a single shared objective [4, 28, 48], ignoring the individual objectives of each slice, and (iii) a strawman approach to enable interference management with slicing [17] that we detail in §2. Our evaluation further shows how the approximation techniques employed by RadioNinja reduce the computation overhead by orders of magnitude, enabling it to make its muting decisions within stringent scheduling time budgets of ≤ 1 ms, without compromising on their correctness.

2 Background and Related Work

2.1 Radio Access Network (RAN)

Radio Access Network (RAN) refers to the wireless last-mile of cellular networks, connecting the base-station to the end-devices (referred as user equipments or UEs). We explain some relevant RAN concepts below.

Resource Block

RAN resources are divided along the frequency and time axes. Specifically, the frequency bandwidth of the radio spectrum is divided into multiple sub-carrier frequencies, and time is divided into equal slots called TTIs (Transmission Time Intervals). In 5G, a TTI slot can range from 250μs-1ms [23]. A resource block (RB) is formed of 12 frequency sub-carriers and 1 TTI slot, and is the smallest resource unit that can be allocated to a user. We can thus visualize the RBs as being organized into a 2D grid as shown in Fig. 1. For simplicity, the figure depicts RBs as the scheduling granularity across users. In practice, network operators schedule radio resources in the granularity of resource block groups (RBGs) to reduce control overhead. Each RBG contains a fixed number of consecutive RBs ranging from 2 to 16 [52, 23]. Henceforth, we will use the term RBG to refer to the scheduling granularity of radio resources across users.

Channel Quality and User Performance

The performance of a user (i.e. how much throughput or data rate they achieve) depends on the number of RBGs assigned to them and the *channel quality* associated with those RBGs (higher channel quality results in a higher data rate per RBG). Typically, the closer the user is to a base-station, the higher is its channel quality. However, obstacles or interference from neighboring base-stations can introduce noise in the signal and degrade the channel quality. Different users (in different locations) can thus experience drastically different channel quality for the same RBG. Moreover, a given user can also see a high variation in channel quality across RBGs, even within the same TTI, due to a phenomenon called frequency selective fading. The combination of these factors motivates the need for incorporating channel awareness when scheduling radio resources across users [11, 12]. The user equipment (UE) periodically reports its channel quality to the base station to enable channel-aware scheduling. We provide a brief primer on channel-aware scheduling mechanisms in §3.1.

2.2 RAN Slicing

As mentioned in §1, RAN slicing is a key 5G feature that brings the notion of multi-tenancy to cellular networks. With RAN slicing, the network operator (e.g. AT&T, Verizon, etc) divides the radio resources among heterogeneous slices (MVNOs, enterprises, applications, etc) as per their SLAs [25]. The SLA of a slice translates to a minimum *quota* of RBGs that the network operator must allocate to it at a base-station [12, 34]. Each slice then schedules its quota of radio resources across its own users as per its own performance objective (e.g. fairness, prioritization, throughput maximization, etc). The performance objective typically varies across slices [18, 21, 46]. A slicing system must support such diverse slice objectives, and ensure each slice gets its quota of resources while retaining performance isolation across slices.

Prior work on RAN slicing has largely been restricted to scheduling radio resources at a single base-station (e.g. [24, 34, 30, 12]). Other works [40, 10] look at the problem of mapping network-wide service-level agreements (SLAs) to per-cell resource quotas for a given slice but ignore potential interference across cells. We leverage these prior works for determining per-cell slice quotas and doing channel-aware slicing and scheduling [12] (as detailed in §3.1). The primary focus of our work, however, is to address the challenges that arise from the interplay of RAN slicing with interference management across multiple cells.

In terms of its goal, RAN slicing bears similarity to the classical concept of hierarchical fair scheduling, where we want weighted fairness across slices, and fairness (or other objectives such as prioritization or throughput maximization) within each slice. However, classical hierarchical scheduling assumes that the top-level allocator can reason about resource allocation independently of how those resources are used internally. In a wireless RAN, this assumption is already challenged even in a single-cell setting because user performance depends not only on how many radio resources (i.e., number of RBGs) are allocated, but also on the quality of the wireless channel associated with those RBGs. Moreover, channel quality varies across users for a given RBG, and across RBGs for a given user. This channel quality variation must be taken into account when assigning RBGs to different slices (and to different users within a slice). As a result, achieving hierarchical objectives efficiently requires channel-aware assignment of RBGs across slices (and across users within each slice), rather than treating RBGs as interchangeable units. §2.5 exemplifies how such channel-aware RAN slicing helps improve overall system throughput.

Our prior work, RadioSaber [12] proposed a system for channel-aware scheduling of radio

resources at a given base-station across slices, and across users within each slice. However, RadioSaber makes a crucial simplifying assumption: the utility of a radio resource is local to a cell and does not depend on scheduling decisions made at other cells. This locality assumption holds in single-cell settings, but is violated in multi-cell deployments where wireless interference impacts scheduling decisions across cells.

In this work we focus on the effects of this wireless interference in multi-cell settings, that introduces new and unique challenges to the problem of hierarchical scheduling across slices. With interference, the quality associated with an RBG R_i and cell C_j depends not only on which slice (and which specific user within the slice) is scheduled on R_i at C_j, but also on whether any user (belonging to any slice) is scheduled at R_i at neighboring cells. In other words, the effective value of a radio resource is no longer local to a single cell. Because of this cross-cell scheduling dependency, we cannot perform channel-aware RAN slicing for each cell in isolation as was done by RadioSaber [12]. Instead interference management decisions must be made jointly with channel-aware scheduling for each RBG across all cells and across all slices. This coupling therefore implies that interference management decisions cannot be cleanly layered above slicing: whether muting a resource is beneficial or not depends on slice-specific objectives and intra-slice scheduling decisions. Consequently, this cross-cell dependency can easily undermine performance isolation across slices – how slice S_k is scheduled at one cell impacts the channel quality (and hence scheduling outcomes) of another slice at a different cell. Our work deals with this challenge: how can we retain performance isolation while doing channel-aware scheduling across slices and interference management across cells.

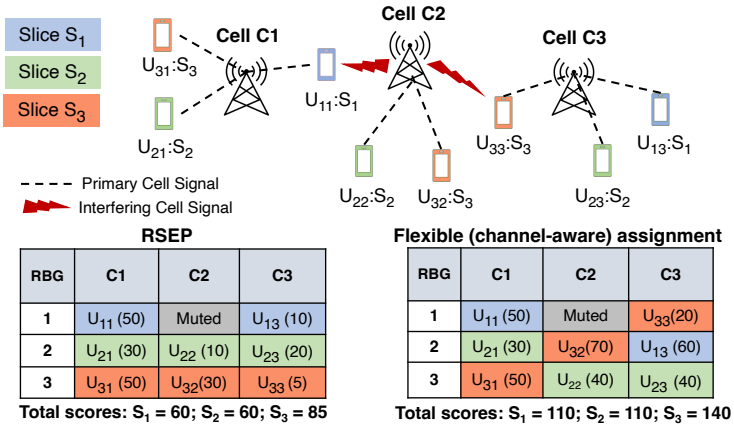
2.3 Multi-Cell Deployments

As depicted in Fig. 2, network operators commonly scale cellular capacity by augmenting coverage regions of macro base stations with lower powered "small cells" [47, 9, 5]. These small cells can be deployed in locations where the macro cell provides poor coverage due to increased distance or obstacles such as high buildings. A macro-cell region (with coverage radius of 1-25Kms) can be expected to support 1-10 small cells [2] (each with coverage radius spanning 0.1-1Kms) [42, 7, 41, 13]. Recent years have witnessed a sharp increase in such multi-cell deployments. At the end of 2022, there were 142K macro-cell towers across the US, and a total of 452K outdoor small cell nodes [58]. It is projected that by 2027, there will be 13 million outdoor 5G small cell deployments at a global level [50].

One way to deploy multiple cells is to partition the frequency bandwidth among cells, such that they never interfere on the same band. However, such static partitioning of frequency can result in severe under-utilization of the network and reduced user throughput (as highlighted in Appendix A). Therefore, an increasingly common mode of deploying multiple cells is for all the cells to share the same frequency band, which avoids partitioning the bandwidth, thereby increasing the overall network capacity [20, 16]. This, however, results in interference, and necessitates a mechanism for managing it.

2.4 Interference Management

Interference management has been studied in 4G context (without slicing). CoMP (Coordinated Multi-Point) [1, 56, 39, 31, 38, 42] encompasses a suite of techniques for managing interference that vary in their deployment complexity, including techniques such as coordinated beam-forming, coordinated muting, etc. In this paper, we focus on coordinated muting, which enables *muting* selected RBGs on selected cells during each TTI to manage interference.



RSEP

RBG	C1	C2	C3
1	U_{11} (50)	Muted	U_{13} (10)
2	U_{21} (30)	U_{22} (10)	U_{23} (20)
3	U_{31} (50)	U_{32}(30)	U_{33} (5)

Total scores: $S_1 = 60$; $S_2 = 60$; $S_3 = 85$

Flexible (channel-aware) assignment

RBG	C1	C2	C3
1	U_{11} (50)	Muted	U_{33}(20)
2	U_{21} (30)	U_{32}(70)	U_{13} (60)
3	U_{31} (50)	U_{22} (40)	U_{23} (40)

Total scores: $S_1 = 110$; $S_2 = 110$; $S_3 = 140$

Figure 3 Three RBGs at three cells assigned across users split among three slices. The numbers in parenthesis denote the data rate achieved by the scheduled user. RSEP (left) restricts slices to use the same RBGs across cells. A more flexible channel-aware assignment (right) results in higher performance across all slices.

When compared to other CoMP strategies, muting is far more feasible to implement. It requires the cells to synchronize and coordinate with one another at per-TTI timescales of $\approx 100s\ \mu s$. Prior work demonstrated the feasibility of deploying muting [31, 32], and of achieving sub-ms coordination between cells [42]. More recently, the move towards virtualized RAN, where RAN processing is deployed in servers hosted in the edge datacenters (away from, but still in the geographical vicinity of, the base station) further provides a ready-made platform where such coordinated decisions across multiple cells can take place [37, 61], that can be leveraged by our system.

However, interference management (and muting) has only been explored in the context of optimizing a single objective (i.e. proportional fairness) across all users [4, 28, 48] – we provide a primer on how muting works in this traditional single objective scenario in §3.2. The focus of our work is to manage interference via muting in the presence of RAN slicing, with each slice optimizing its own diverse objective.

2.5 Interference Management with Slicing

Summarizing the context so far, prior work on RAN slicing has not considered the problem of interference management in multi-cell scenarios. On the other hand, the prior work on interference management has not considered the new regime of network slicing where users are grouped into slices with diverse objectives.

One exception, that considers both interference management and slicing, is RSEP [17]. RSEP proposes an extreme solution of assigning a given slice exactly the same set of resource block groups (RBGs) at each cell. We depict this in Fig. 3 with the table on the left – slice S_1 is assigned the first RBG at each of the three cells (C_1, C_2 and C_3), while slices S_2 and S_3 are assigned the second and third RBG respectively at each cell. Such an assignment ensures that any muting decision for an RBG only affects users from the same slice at the neighboring cells. It thereby reduces the problem of muting to the no slicing (single objective) setting. However, such a constrained assignment inherently ignores channel diversity, where the channel quality on a given RBG varies drastically depending on which user is scheduled on it [12, 11, 59] (as explained in §2.1). This results in inefficient usage of the radio spectrum and reduced network throughput (up to 40% lower than channel-aware slicing [12]).

Fig.3 illustrates this through a simple example. The table on the left shows the schedule with RSEP, where each slice is restricted to use the same RBG at each cell. Slice S_1 prefers to mute its assigned RBG at cell C_2 to boost the channel quality of its user U_{11} at the neighboring cell C_1. Slice S_1's user at C_3 (U_{13}) is not significantly impacted by interference from C_2. The numbers in parenthesis denote the data rate (throughput) achieved by the scheduled user. The total data rate experienced by Slices S_1, S_2, and S_3 (summed across their scheduled users) is 60, 60, and 85 respectively.

The table on the right shows a more "flexible" assignment of RBGs across slices that can leverage channel diversity. Such an assignment can achieve superior performance by allowing different slices to use different RBGs across different cells, as per their respective channel qualities (e.g. RBG 2 is assigned to slice S_2 at cell C_1 and to slice S_3 at cell C_2). The unrestricted assignment further allows a given muting decision to simultaneously benefit multiple slices thus amortizing the cost of muting (for e.g. muting the transmission of RBG 1 at cell C_2 boosts the channel qualities of users U_{11} and U_{33} in Slice S_1 and Slice S_3 respectively). In this case, the total data rate, summed across all scheduled users for Slices S_1, S_2, and S_3 is 110, 110, and 140 respectively; which is higher across the board when compared to RSEP.

While this is a toy example, our evaluation with real-world traces in §6 reinforces the performance benefits of such "flexible" channel-aware assignment of RBGs across slices and cells, when compared to RSEP's restricted assignment.

However, enabling such flexible channel-aware assignment requires reasoning about the muting decision across multiple slices with diverse objectives: Is it okay to mute the first RBG at C_2 to benefit slices S_1 and S_3? Which slice incurs the cost of this muted RBG and how? We need a way to answer such questions in a manner that still retains performance isolation across slices. Our system, RadioNinja, deals with this challenge.

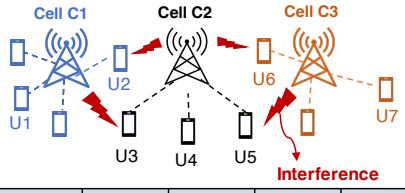
3 Primer on Relevant Mechanisms

Before diving into RadioNinja's design, we provide a brief primer on mechanisms for channel-aware slicing at a single cell (§3.1), and for interference mechanism via muting in a no slicing setting with a single global objective (§3.2).

3.1 Channel-aware Slicing and Scheduling

Channel-Aware Inter-Slice Scheduling

RadioNinja leverages RadioSaber [12] for channel-aware slicing at each cell, where an *inter-slice* scheduler assigns RBGs to slices and an *intra-slice* scheduler assigns RBGs to users within the slice. In each TTI, the *inter-slice* scheduler greedily assigns RBGs to slices, one RBG at a time as follows. It queries each *intra-slice* scheduler asking which user will be scheduled on the RBG if that RBG is assigned to that slice. Each slice responds to this query based on its own intra-slice scheduling policy (detailed next). The responses allow the *inter-slice* scheduler to determine the channel quality associated with that RBG for each slice (based on which user each slice picks). It then assigns the RBG to the slice with the highest channel quality. As mentioned in §2.2, each slice has a per-TTI quota of RBGs derived from its SLA. While greedily assigning RBGs to slices, the scheduler tracks their quota, and avoids assigning more RBGs to slices that have exhausted their quotas.



Muting Hypothesis	C1	C2	C3	Overall Score
H0: No muting	U1 (10)	U4 (15)	U7 (30)	55
H1: Mute C1	Muted	U3 (40)	U7 (30)	70
H2: Mute C2	U2 (50)	Muted	U6 (60)	110
H3: Mute C3	U1 (10)	U5 (20)	Muted	30

Figure 4 Example showing muting for a single objective (no slicing). The figure shows user locations with respect to cells. The table on the bottom shows the assignment of RBG R_i at each cell for the different muting hypotheses. The number in parentheses indicates the score for the corresponding user. Muting RBG R_i at cells C_1 or C_2 produces a net benefit, while muting it at C_3 produces a net loss.

Channel-Aware Intra-Slice Scheduling

For a given RBG R_i, the *intra-slice scheduler* assigns each of the slice's user u_k a score ($score(u_k, R_i)$) based on the desired performance objective of the slice. It then assigns R_i to the user with the highest score. For example, a slice wanting to maximize throughput would set $score(u_k, R_i) = inst\_rate(u_k, R_i)$ (i.e. the instantaneous data rate u_k would achieve if it is allocated R_i, as per its channel quality for R_i). A slice can optimize for proportional fairness (a popular objective in cellular networks [59, 29, 36]) by weighing the instantaneous rate with the average rate allocated to u_k so far, i.e. by setting $score(u_k, R_i) = \frac{inst\_rate(u_k, R_i)}{avg\_rate(u_k)}$. The scores can also be weighed as per user priorities. Such score-based scheduling of RBGs across users is also adopted in a no slicing setting, where all users share a common global objective [11].

3.2 Muting under a Single Objective

We next detail how muting decisions are made without slicing where all users share a single global objective. Past work adopts a greedy strategy for this [4, 28, 48]. In each TTI, we greedily pick RBG R_i, and analyze the cost-benefit trade-off of muting that RBG at each cell, one cell at a time (referred to as the muting hypothesis). It is beneficial to mute R_i at cell C_j if the boost in channel quality at the neighboring cells overpowers the penalty of losing out on that RBG at C_j. To assess this, we compute the system-wide total score under each muting hypothesis H_j (where R_i is muted at cell C_j) and under no muting (denoted as hypothesis H_0). The total score under hypothesis H_j is computed as: $\sum_{u_k \in U(R_i, H_j)} score(u_k, R_i)$. Here, $U(R_i, H_j)$ denotes the set of users that are scheduled on R_i at each cell under hypothesis H_j. The user scores are computed as described in §3.1, as per the shared global objective.

The difference between the total score of a muting hypothesis H_j and no muting hypothesis H_0 gives us the net benefit of muting RBG R_i at cell C_j. If the net benefit is negative for all muting hypothesis, R_i is not muted at any cell. Otherwise, we select the hypothesis with the largest net benefit and mute the RBG at that cell.

Fig. 4 presents an example for this, where we have 3 cells and want to determine whether or not to mute the transmission of a cell on a given RBG R_i. Hypothesis H_0 is when R_i is not muted at any cell (top row in Fig. 4). The three cells schedule UEs U_1, U_4 and U_7, that do

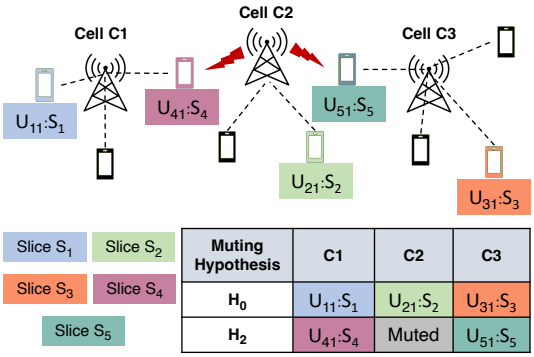


Figure 5 The example above shows the assignment of RBG R_i across slices under no muting hypothesis (H_0) and the hypothesis where the RBG is muted at cell $C2$ (H_2). The numbers in parenthesis indicate the user's slice association.

not experience significant interference from other cells. The total score for H_0 (last column) is computed by adding the scores of each of these three UEs (indicated in parentheses). The next hypothesis H_1 (second row) is where R_i is muted at C_1 – this impacts the scheduling decision at cell C_2, boosting channel quality of "edge" UE U_3. The set of scheduled UEs, based on whom H_1's total score is computed, therefore includes U_3 at C_2 and U_7 at C_3. Likewise, hypothesis H_2 (third row), where R_i is muted at C_2, boosts the channel quality of "edge" UEs U_2 and U_6, and they get scheduled on this RB at cells C_1 and C_3 instead. The overall score of H_2 is computed by adding their scores. We similarly consider the impact of muting R_i at C_3 and compute the overall score of H_3. Comparing scores of H_1, H_2 and H_3 with H_0, we can see how muting R_i at C_1 or C_2 produces a positive net benefit while muting it at C_3 incurs a loss. The benefit of muting at C_2 is higher, so the system will mute R_i at C_2. This freezes the muting and scheduling decision for that RBG, and we then move on to the next RBG and repeat the same steps.

A few points are worth noting:

- In the above example, we could have additional hypotheses where we mute R_i on more than one cell (resulting in 2^M hypothesis for each RBG, where M is the number of cells). We observed that muting multiple cells for the same RBG provides limited benefit since most interference occurs between the macro-cell and small cells, i.e., muting either the macro-cell or one of the small cells that interferes with the user being served by the macro-cell is sufficient. So we restrict the assessment to consider muting up to one cell per RBG (making the number of hypotheses linear in M).
- The muting hypothesis is evaluated without actually muting the cell – it uses knowledge about the channel quality each user will experience with and without interference from the neighboring cells. (The 3GPP standard specifies the use of specific "resource elements" to enable such interference measurement [3, 15], as detailed in Appendix B).

4 RadioNinja Design

RadioNinja uses the greedy muting approach described in §3.2, where in each TTI, it greedily decides whether or not to mute an RBG at any cell, one RBG at a time. However, rather than catering the muting decision towards a single shared objective across all users, RadioNinja makes these decisions in manner that respects individual slice objectives (as described in the remainder of this section). The greedy per-RBG muting decisions are made jointly with

greedy channel-aware scheduling decisions (i.e. which users, belonging to which slice, should be scheduled on the unmuted RBGs) – RadioNinja uses RadioSaber's scheduling logic for this, as described in §3.1. Jointly making muting decisions with channel-aware RAN slicing requires RadioNinja to address multiple challenges, that we detail in this section:

1. *How do we reason about a muting hypothesis while maintaining performance isolation between slices?* (§4.1)
2. *How do we efficiently compute the benefit of a muting decision for a given slice based on its own objective?* (§4.2)
3. *How do we efficiently compute the cost of a muting decision for a given slice based on its own objective?* (§4.3)
4. *How do we decide which muting hypothesis to implement (i.e. which cell to mute on an RBG) when different slices benefit from different decisions?* (§4.4)

4.1 Reasoning about a muting hypothesis

While it is straightforward to reason about muting decisions when all users share a single common objective (as described in §3.2), RadioNinja must deal with the challenge of handling diverse slice objectives in a manner that maintains performance isolation between slices. Fig. 5 illustrates the challenge by considering how an RBG R_i is assigned to users across slices under no muting and when it is muted at cell C_2. When R_i is not muted (hypothesis H_0, top row), users of slices S_1, S_2 and S_3 are scheduled on R_i at cells C_1, C_2 and C_3, respectively. Muting R_i at C_2 (hypothesis H_2, bottom row) changes the channel quality (and scores) of users in the neighboring cells C_1 and C_3, causing R_i to be assigned to a different set of slices (S_4 at C_1 and S_5 at C_3). It therefore appears as if the decision to mute RBG R_i at cell C_2 benefits slices S_4 and S_5, at the cost of slices S_1, S_2, and S_3. How do we make a systematic choice in such a scenario – is it okay to help some slices at the cost of others? Wouldn't that violate performance isolation across slices? RadioNinja effectively *avoids* making such choices using the following insight:

Resource attribution to retain isolation

Rather than considering a muted RBG at a cell C_j as a wasted resource that belongs to no one, we count it towards the quota of the K slices that benefit from muting it, i.e. we reduce the quota of those slices at C_j by $1/K$ RBGs. This restricts the cost of muting (i.e., reduction in available resources at C_j) to these K slices that benefit from boosted channel qualities at neighboring cells. This, in turn, enables independent cost-benefit analysis based on the specific objectives of these slices.

Revisiting the example in Fig. 5: given how we account for the muted resource, since the quotas of S_1, S_2, and S_3 remain intact, they are not really penalized by the decision to mute R_i at C_2 – each of these slices would simply be assigned a different RBG at cells C_1, C_2, and C_3 respectively. S_4 and S_5, in contrast, may lose some quota at C_2 if the RBG is muted – the cost of losing that quota at C_2 should be compared with the benefit that these slices experience at cells C_1 and C_3 due to muting. For instance, if serving S_4's user U_{41} at C_1, comes at the cost of penalizing a higher priority user at C_2 due to the reduced quota, S_4 might not want to proceed with the muting decision.

Challenge

How do we identify the set of K slices that benefit from muting, if weighing the benefit vs. cost of muting depends on K itself (as we reduce quota of benefiting slices by $1/K$ RBGs)?

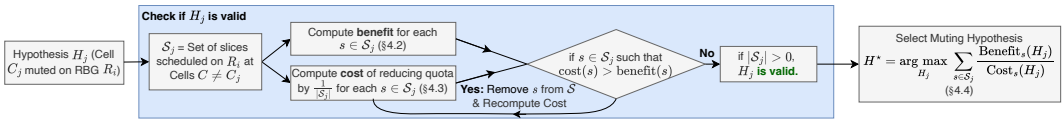


Figure 6 RadioNinja's Workflow

Our approach

We use an iterative algorithm (summarized in Fig. 6). Consider the hypothesis H_j of muting R_i at C_j:
(i) We start with assuming that each slice that uses R_i at C_j's neighboring cells (i.e. at $C_n \neq C_j$) under H_j is benefited by the muting. We refer to this set of slices as $\mathcal{S}$. We accordingly reduce the quota of each slice in $\mathcal{S}$ by $\frac{1}{|\mathcal{S}|}$ RBGs at cell C_j.
(ii) We then individually assess the benefit (§4.2) and cost (§4.3) of muting for each slice in $\mathcal{S}$. If the assessment reveals a slice S' in $\mathcal{S}$ is hurt by muting (i.e. the cost it pays due to the quota reduction at C_j is higher than the benefit it gets at the neighboring cells), then we exclude it from participating in the muting decision. Specifically, we undo the reduction in quota of slice S' at C_j, and remove it from the set $\mathcal{S}$. This reduces the size of set $\mathcal{S}$, thereby increasing the share of quota reduction at C_j for the remaining slices in $\mathcal{S}$.
(iii) We accordingly update the cost of muting for the remaining slices in $\mathcal{S}$, and re-assess the benefit vs cost of muting for each of these slices as per step (ii).
(iv) We repeat steps (ii) and (iii) in a loop until the set $\mathcal{S}$ stops changing, giving us the converged set of $K = |\mathcal{S}|$ slices that benefit from the muting hypothesis. (The number of iterations is bounded by the number of cells.)

So for our example in Fig. 5, we will start by assuming both S_4 and S_5 benefit from muting hypothesis H_2, and evaluate the cost of reducing 0.5×RBG from the quota of both of these slices at C_2. If the cost of muting for S_4 turns out to be higher than the benefit it sees, S_4 will get back its quota of 0.5×RBG at C_2 and it will be removed from $\mathcal{S}$. The muting hypothesis H_2 will then be considered valid if S_5 sees sufficiently high benefit over the cost of 1×RBG quota reduction at C_2.

The following is worth noting.

- **Slice Eligibility for Muting:** A slice in $\mathcal{S}$ can participate in the muting decision only if it has sufficient quota (at least $\frac{1}{|\mathcal{S}|}$ RBGs) remaining at C_j to pay for the cost of muting. If a slice has exhausted all of its quota at C_j (from the previously allocated RBGs in that TTI), then it cannot induce muting at C_j and cannot be added to the set $\mathcal{S}$.
- **Valid Muting Hypothesis:** We consider H_j to be a valid hypothesis if the resulting set $\mathcal{S}$ is non-empty (i.e. there are non-zero number of slices that benefit from the muting decision after paying the shared cost of muting). We similarly check the validity of other muting hypothesis (i.e. of muting R_i at other cells). If none of the muting hypotheses for R_i are valid, we do not mute it at any cell. If there are multiple valid muting hypotheses, we select one (as detailed in §4.4) and proceed with that decision.

4.2 Computing Benefit of Muting

We now explain how RadioNinja computes the benefit of a given muting hypothesis for a given slice in $\mathcal{S}$. Let us refer again to the example in Fig. 5, and consider how to compute the benefit of muting hypothesis H_2 (i.e muting of RBG R_i at C_2) that slice S_4 enjoys at the neighboring cell C_1. At a high-level, this can be computed by comparing the score of S_4 at cell C_1 with and without muting R_i at C_2 (with scores defined based on the slice's

objective, as detailed in §3.1). However, naively comparing the assignment of R_i under H_2 with the no muting hypothesis H_0, it would seem that the performance of S_4 (which was not even assigned R_i at any cell under H_0) is far better under H_2 (where it is assigned R_i at C_1). This is a misleading assessment that overestimates the benefit of muting. This is because, by getting scheduled on R_i at C_1 under hypothesis H_2, S_4 uses up a quota of 1 RBG at C_1. It will therefore be scheduled on 1 less RBG among the remaining $N-1$ RBGs in that TTI under H_2, when compared to the remaining $N-1$ RBGs under H_0.

In order to truly assess the benefit of muting, we need to compare how the RBG quota used by S_4 to schedule user U_{41} on R_i under H_2 compares to the use of that RBG quota under H_0. This ideally requires comparing the total score of S_4 across hypotheses H_0 and H_2, not just for RBG R_i, but for all of the remaining $N-1$ RBGs in that TTI, such that we can account for how that RBG quota is subsequently used under H_0. This implies that we need to run the scheduler to compute the hypothetical assignment of UEs on these remaining $N-1$ RBGs for each muting hypothesis that we wish to greedily assess for each RBG. As we show in §6, the $O(N^2)$ complexity induced by this is prohibitively expensive, given the tight TTI timescales at which scheduling and muting decisions must be made. So how do we work around this?

Our approximation technique

We use a simple heuristic to *approximate* the benefit of muting without re-running the scheduler for the remaining RBGs. Referring to our example in Fig. 5, in order to assess the benefit to S_4 at neighboring cell C_1 under hypothesis H_2, we compute which S_4 user would have hypothetically been allocated R_i at C_1 under no muting, if that RBG was restricted to being allocated exclusively to S_4. We then subtract S_4's score at C_1 obtained from this hypothetical assignment of R_i under no muting from its score at C_1 under hypothesis H_2. We use the outcome as our proxy for the benefit of muting. This essentially captures the impact of reducing interference from C_2 for slice S_4's users at C_1. However, it masks the impact of channel diversity in our assessment (since in reality, S_4 is not allocated R_i under no muting, but would instead be allocated a different RBG). Nonetheless, we find this approximation works well in practice because in situations where the benefit substantially outweighs the cost of muting, the effect of interference mitigation overpowers the effects of channel diversity. When the benefit is low (interference effects are small), the slice is anyway removed from set $\mathcal{S}$, and the precise value of the benefit does not make a difference.

If a given slice S_k has users scheduled in R_i at multiple neighboring cells under hypothesis H_j, we approximate the benefit seen by S_k at each of these cells and sum them up to get the overall benefit to S_k for hypothesis H_j.

4.3 Computing Cost of Muting

As mentioned before, we account for the cost of muting RBG R_i at C_j by subtracting $\frac{1}{|\mathcal{S}|}$RBGs at cell C_j from the quota of each slice in set $\mathcal{S}$ (that benefit from the muting). Assessing this cost for a given slice S_k requires comparing how S_k is scheduled on remaining RBGs at C_j with and without muting (i.e. with and without the quota reduction). This causes two hurdles. First, as discussed in §4.2, re-running the schedulers across all remaining RBGs for each muting hypothesis is prohibitively expensive. Second, how do we even assess the impact of fractional quota reductions? RBGs must be wholly allocated to a single user, and cannot be split across multiple users. When multiple slices have non-integral quotas, the scheduler allocates the last RBG in the TTI (that must ideally have been split between these

slices) to one of these slices picked randomly. It then keeps track of the (fractional) surplus or deficit in quota allocation, that rolls over to the next TTI. Rolling over of fractional offsets ensures that a slice pays its due cost of muting over time. But the impact of that might not be seen in that TTI, and re-running the scheduler over multiple TTIs would increase the computational complexity further.

Our approximation technique

At the start of each TTI, we run the scheduler once to determine an initial assignment of each RBG under the hypothetical scenario where no RBG is muted at any cell. We use this as a reference schedule to assess how much an RBG is worth to a given slice at a given cell and accordingly compute the cost of quota reduction, when assessing individual muting hypothesis for each RBG (without re-running the scheduler).

The worth of an RBG for S_k at cell C_j increases as we keep reducing the slice's quota at that cell. This is because when a slice's quota is reduced, it will kick out (i.e. avoid scheduling) users on RBGs with the lowest score. Further quota reduction incurs a higher cost, by kicking out users with higher scores. We account for this by clustering RBGs based on their scores and using the average score within each cluster as the worth of the RBG in that cluster. (We use a cheap clustering logic based on standard deviations from the average score.) When computing the cost of quota reduction, we first consider RBGs in the lowest score cluster. If the average score of that cluster is X, then a quota reduction of 0.5 has a cost of $0.5X$. If the total quota reduction for a slice (summed across muting decisions over multiple RBGs) exceeds the number of RBGs in the lowest score cluster, we move over to the next lowest score cluster, and so on.

The reference schedule does not account for potential muting at other cells which can increase the worth of RBGs. Therefore, the cluster average scores taken directly from the reference schedule as described above can underestimate the cost that slice S_k incurs for muting at C_j. To compensate for that, we update S_k's average score for each RBG cluster at C_j by multiplying it with an error ratio β. We compute β from past observation – by how much did S_k's estimated score at C_j in the previous TTI t differed from the actual score it achieved at C_j in that TTI. Specifically, $\beta = \frac{A_{j,k,t}}{E_{j,k,t}}$, where $A_{j,k,t}$ is the (actual) average score across all RBGs actually assigned to S_k at C_j in TTI t and $E_{j,k,t}$ is the (estimated) average score across all RBGs assigned to S_k at C_j in the reference schedule during the last TTI t.

4.4 Comparing Muting Hypotheses

We use the benefit and cost computed as described above to assess whether the muting hypothesis is valid as outlined in §4.1. We empirically observed that in most cases, at most one muting hypothesis is valid for a given RBG (for the reason explained in §3.2). When there are multiple valid muting hypothesis H_j, we pick one as follows. For each valid muting hypothesis H_j, we compute the total benefit seen by each slice in set $\mathcal{S}_j$ (where set $\mathcal{S}_j$ is the converged set of benefiting slices in H_j) and divide it by the cost incurred by that slice. We then sum this benefit to cost ratio across all slices in set $\mathcal{S}_j$, and favor the muting hypothesis that has the highest value for this sum.

5 Implementation

Customizing Slice Objectives

We leverage the interface provided by RadioSaber [12] to enable slice operators to customize their objectives using certain parameters. In particular, for a given slice S_k, we assign the following score to user u_m in the slice for RBG R_i: $score(u_m, R_i) = w_m \times \frac{inst\_rate(u_m, R_i)}{(avg\_rate(u_m))^\alpha}$, where w_m is the weight of user u_m (set based on its relative priority) and $\alpha \in \mathbb{R}_{\geq 0}$ controls the degree of fairness. This directly maps to optimizing a well-known parameterized objective known as weighted alpha-fairness [33] for that slice. Each slice can configure w_m and α differently to capture different objectives. With equal weights across all users, setting $\alpha = 0$ maximizes throughput (MT). Setting $\alpha = 1$ achieves proportional fairness (PF), where the α-fair objective is reduced to maximizing $\sum_{u_m \in U_k} w_m log(x_m)$ (where x_m is the throughput achieved by u_m). Increasing α further increases the degree of fairness among users of the slice in terms achieved throughput.

Implementation in Simulator

Due to the absence of required features in current versions of open-source RAN testbeds (notably, per-RBG muting capability), we evaluate our system using trace-driven simulations that further allows us to scale to hundreds of users. We implement RadioNinja in an open-source cellular simulator [45, 12]. Our simulator configuration for 5G small cell deployment (detailed in §6) adheres to the 3GPP-specified parameters [2].

Collecting user traces

We use NG-Scope [60] deployed on USRP X310 to collect real-world channel quality data from cellular base-stations across different frequency bands and operators. The tool allows us to extract fine-grained metrics that capture channel quality of the UE across different RBGs over time. We move around with this setup to collect channel quality traces at different locations relative to the transmitting cell, mapping different chunks of the resulting trace to different users in our simulations.

Software Implementation

We also implement RadioNinja as a separate module in software on a 12th Gen Intel Core i9-12900F machine in order to evaluate its computational overhead. Specifically, we assess the time it takes for making its joint scheduling and muting decisions across all RBGs in a TTI for different 5G configurations (that vary in TTI duration and the number of RBGs per TTI). Our software implementation aligns with the move towards virtualized RAN, where RAN processing is done in software [30, 61].

6 Evaluation

Our evaluation is divided into three key components:

1. Comparison with Baselines

In §6.2, we use trace-driven simulations across scenarios spanning hundreds of users split across slices with diverse objectives to show how RadioNinja consistently outperforms the following baselines:

- **RadioSaber** [12]: that does channel-aware slicing at each cell without any interference management.
- **Single Objective Muting (SOM)** [4, 28, 48]): This baseline uses channel-aware RAN slicing, and schedules RBGs across users within each slice as per their individual objectives. However, it uses the single objective muting algorithm discussed in §3.2 to assess the scores of scheduled users and make its muting decision as per the specified global objective agnostic of individual slice objectives.
- **RSEP** [17]: that assigns a given slice the same set of RBGs at each cell, thereby ignoring channel diversity when scheduling RBGs. It then performs objective-based muting decisions for individual slices in isolation.

2. Comparison with Exhaustive Approach

In §6.3, we evaluate how RadioNinja's techniques to approximate the benefit and cost of muting closely match the outcome of the more rigorous (exhaustive) assessment that requires repeatedly re-running the scheduler. RadioNinja's logic comfortably fits within the TTI budget across different 5G configurations, while the exhaustive approach is prohibitively expensive.

3. Factor Analysis

We individually evaluate the impact of certain system design choices made by RadioNinja, as discussed in Section 4, in Appendix E.

6.1 Experiment Settings

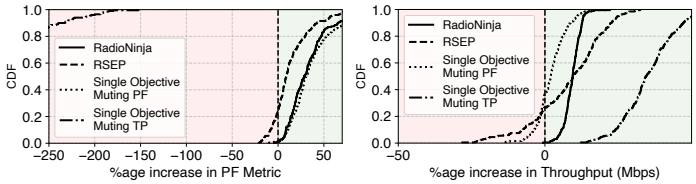
We simulate a multi-cell deployment comprising one macro cell and four small cells (similar to what is depicted in Fig. 2) [2, 42]. We configure each cell with a bandwidth of 100 MHz, split across 512 RBs per TTI, grouped into 64 RBGs [53]. The small cells are deployed uniformly within the macro-cell's coverage region at a randomly-chosen distance of at least 300m from the macro-cell and 500m from neighboring small cells. We simulate 160 users split across 8 slices [2]. We situate each user within the coverage region of a randomly selected small cell with a probability of 2/3, and outside of the coverage region of any small cell (thereby attached to the macro-cell) with probability 1/3. Of the total users, 60% are static while 40% are mobile (traveling at speeds ranging from 3 to 30 km/h). Once a user is generated, we apply the 3GPP urban area path loss model to calculate the user's wideband channel quality (aggregated across all RBGs) based on its distance from each cell. We then assign a real-world trace to each user that maps the computed wideband value for each cell (thereby extracting the effects of channel diversity from real-world conditions). We use the standard policy of assigning each user the serving cell from which it experiences highest wideband channel quality. We configure each slice to have the same quota of RBGs at each cell. We vary the slice objectives and user traffic patterns across different scenarios, and test the scenarios against different scheduling and muting schemes described above. For each setting, we run experiments with multiple random seeds that end up varying the specific user locations, and user distribution across cells and slices. We capture different slice objectives by varying the parameters of weighted α-fairness (as described in §5). We experiment with two categories of user traffic patterns: (i) backlogged (representing high bandwidth video streaming, gaming, etc that would saturate the datarate allocated to the user), and (ii) web flows (fixed sized flows with Poisson inter-arrival times and flow sizes drawn from a heavy-tailed distribution [43], generated at an average rate of 10Mbps).

6.2 Comparative Evaluation

We use RadioSaber without muting as the reference baseline and report the relative changes induced by RadioNinja, SOM, and RSEP over it for the given slice-objective.

Parameter	PF Slices	TP Slices
# Slices	4	4
Traffic Pattern	Backlogged	Backlogged
Objective	PF	Max Throughput
Metric	$\sum_i \log x_i$	avg tpt
Result	Fig. 7b	Fig. 7c

(a) Scenario 1 configuration



(b) Per slice performance of PF slices

(c) Per slice performance of MT slices

Figure 7 RadioNinja's, RSEP's and SOM-PF's and SOM-TP's impact on slice-level performance when compared to RadioSaber (no muting) for (b) PF slices and (c) Max TP slices under Scenario 1, aggregated across 50 experiment runs with different seeds.

Scenario 1: Proportional Fairness and Maximizing Throughput

In our first scenario (summarized in Table 7a), we configure four slices with proportional fair (PF) objective by setting $\alpha = 1$. We refer to these as "PF slices". The remaining four slices (referred to as MT slices) are configured with the objective of maximizing throughput (MT) by setting $\alpha = 0$. We accordingly experiment with two different variants of SOM baseline, one configured with PF as the global objective and another with MT as the global objective. Fig. 7 reports the results (aggregated over 50 runs): In Fig 7b we compute the percentage improvement in the PF metric for each of the PF slices, when compared to RadioSaber (no muting), and plot the resulting CDF. Fig 7c similarly plots the CDF of the percentage improvement in the total throughput over RadioSaber (no muting) for each of the MT slices.

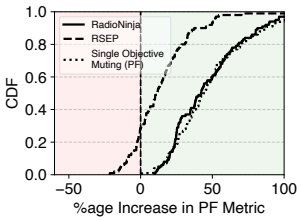
(i) RadioNinja vs RadioSaber without muting: RadioNinja's interference management benefits slices across the board when compared to no muting, resulting in an average of 33.5% increase in PF metric for the PF slices (Fig 7b) and 9.3% increase in throughput for the MT slices (Fig7c).

(ii) RadioNinja vs SOM: SOM's performance depends on how the global objective is configured. SOM-PF (configured with PF objective) results in an average of 37.4% increase in PF metric over RadioSaber without muting (similar to RadioNinja) for the PF slices. However, it comes at the cost of hurting a large proportion of MT slices (more than 37% slices, as shown in Fig. 7c) whose objective does not align with SOM-PF, hence violating performance isolation. In contrast, SOM-MT (configured with MT objective) substantially improves throughput for the MT slices (with an average increase of 35.2% over no muting), but it comes at the cost of significantly hurting the PF slices thereby violating performance isolation between slices (we cut off the x-axis at -200%, so the extent of this cost is not fully visible). This drastic penalty is caused by the gross misalignment between SOM's muting objective (MT) and the slice's scheduling objective (PF), where the PF slice often allocates RBGs to relatively poor users (that suffer from low channel quality) in order to achieve fairness, and SOM optimizing for throughput pointedly mutes those RBGs (with low MT scores), thereby degrading the PF metric.

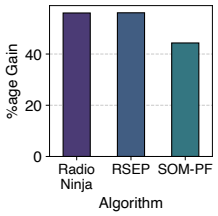
(iii) RadioNinja vs RSEP: RSEP's performance is inferior to RadioNinja's across all PF slices, with 18.4% lower PF metric than RadioNinja on an average. For the MT slices, RSEP fares slightly better than RadioNinja for half of the slices (which are incidentally allocated reasonably good RBGs and benefit from isolated muting decisions with RSEP). However, it has worse throughput than RadioNinja for the other half, of which 26.5% slices have even worse throughput than the baseline, caused by RSEP's channel-unaware scheduling.

Parameter	PF Slices	$\alpha = 3$ Fair Slices
# Slices	4	4
Traffic Pattern	Backlogged	Backlogged
Objective	PF ($\alpha = 1$)	Fairness with $\alpha = 3$
Metric	$\sum_i \log x_i$	10%ile, avg tpt
Result	Fig. 8b	Fig. 8c Fig. 8d

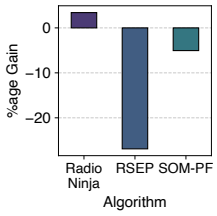
(a) Scenario 2 configuration



(b) Per slice performance of PF-slices



(c) % increase in 10^{th} percentile throughput across users in $\alpha = 3$



(d) % increase in avg throughput across users in $\alpha = 3$ Fair slices

Figure 8 RadioNinja's, RSEP's and SOM's impact on slice-level performance when compared to RadioSaber (no muting) for the (b) PF slices and (c, d) $\alpha = 3$ Fair slices under Scenario 2, aggregated across 50 experiment runs with different seeds.

Scenario 2: Varying Degree of Fairness

In this scenario (summarized in Table 8a), we configure four slices (referred as PF slices) with proportional fair (PF) objective by setting $\alpha = 1$. The remaining four slices were configured with an objective with higher degree of throughput fairness (α set to 3). We refer to these as "$\alpha = 3$ Fair slices". We configure the SOM baseline with a global objective that maps to PF. All users are configured with backlogged traffic pattern. We report the results (aggregated over 50 runs with different seeds) in Fig 8. Fig 8b plots the CDF of percentage improvement in the PF metric experienced by each of the PF slices, when compared to RadioSaber (no muting). Fig. 8c plots the percentage improvement in the throughput of the poorest (lowest 10%ile) users over RadioSaber (no muting), as a measure of fairness for $\alpha = 3$ Fair slices. Fig. 8d plots the corresponding percentage improvement in the average throughput across all users for $\alpha = 3$ Fair slices. We have the following key takeaways:

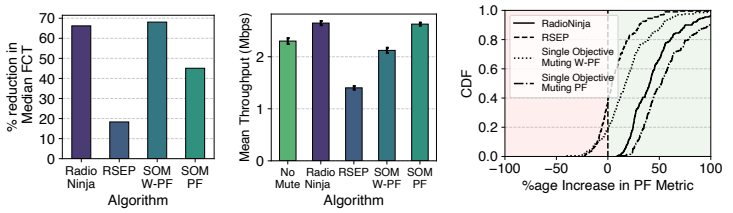
(i) RadioNinja vs RadioSaber without muting: RadioNinja's interference management benefits slices across the board. For the set of PF slices, the corresponding PF metric of each slice increases with muting (as shown in Fig 8b), resulting in 48.0% average improvement over RadioSaber without muting. For the set of $\alpha = 3$ Fair slices, the throughput of the poorest (lowest 10%ile users) increases by 56.1% with RadioNinja (Fig. 8c). The average throughput (not directly aligned with fairness objective) also improves by a marginal 3.4% (Fig. 8d).

(ii) RadioNinja vs SOM: We find that SOM, configured with global PF objective, performs as well as RadioNinja for the PF slices (Fig 8b). It also improves 10%ile throughput for the $\alpha = 3$ Fair slices by 44.3% over no muting – SOM optimizing for proportional fairness with $\alpha = 1$ helps these users, but not to the same extent as RadioNinja that can better tailor its muting decisions with the $\alpha = 3$ fair objective of these slices (Fig. 8c).

(iii) RadioNinja vs RSEP: RSEP's performance is inferior to RadioNinja's across all PF slices, with 33% lower PF metric than RadioNinja on an average (Fig 8b). When compared to RadioSaber no mute, RSEP is able to improve performance for 71% of slices due to muting, but 29% slices experience *worse* performance due to channel unawareness. For the $\alpha = 3$ Fair slices, we find that RSEP is able to achieve sufficiently good throughput (similar to RadioNinja) for the poorest 10%ile users that are more prone to the effects of interference than channel diversity (Fig. 8c). However, it induces a 26.9% reduction in the average throughput across all users in this category of slices due to channel-unaware scheduling, when compared to RadioSaber, resulting in an overall poorer outcome than RadioNinja (Fig. 8d).

Parameter	WPF Slices	PF Slices
# Slices	4	4
Traffic Pattern	• 40% Web Flow (W = 1000x) • 60% Backlogged	Backlogged
Objective	Weighted PF	PF
Metric	Median FCT	$\sum_i \log x_i$
Result	Fig. 9b Fig. 9c	Fig. 9d

(a) Scenario 3 configuration



(b) Improvement in Median FCT of prioritized web flows **(c)** Avg throughput of non prioritized flows **(d)** Per slice performance of PF slices

Figure 9 RadioNinja's, RSEP's and SOM W-PF's and SOM-PF's impact on slice-level performance when compared to RadioSaber (no muting) for (b,c) Weighted PF slices and (d) PF slices under Scenario 3, aggregated across 50 experiment runs with different seeds.

Scenario 3: Varying Weights Across Users

In this scenario (summarized in Table 9a), we configure the first four slices (referred to as WPF slices) with weighted proportional fairness objective, setting $\alpha = 1$, with 40% of randomly selected users having a very high weight of 1000 and web flow traffic pattern, and the remaining 60% users have a weight of 1 and backlogged traffic pattern. The remaining four slices (referred to as PF slices) are configured with proportional fairness objective with equal weight and backlogged traffic pattern across all users. We try two variants of SOM here – SOM-WPF and SOM-PF. SOM-WPF is configured with weighted PF as the global objective (where the prioritized 40% users of WPF slices have a weight of 1000 and all other users have a weight of 1). SOM-PF is configured with PF objective. We report the results in Fig. 9. We consider two metrics for WPF slices: reduction in median flow completion time (FCT) of the prioritized 40% users, when compared to RadioSaber without muting (Fig. 9b), and the average throughput of the remaining 60% users (Fig. 9c). For the PF slices, we report improvement in PF metric over RadioSaber without muting (Fig. 9d), similar to other scenarios.

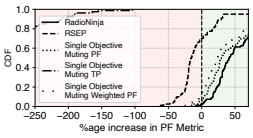
(i) RadioNinja vs RadioSaber without muting: RadioNinja results in 66.2% reduction in median FCT for the prioritized users in WPF slices over RadioSaber without muting (Fig. 9b), along with a 15.0% improvement in overall throughput of the non-prioritized users (Fig. 9c). Among the PF slices, RadioNinja increases the PF metric of all slices (Fig. 9d), with an average increase of 45.1% over no muting.

(ii) RadioNinja vs SOM: SOM's performance again depends on how the global objective aligns. SOM-WPF results in 68% reduction in median FCT of the prioritized users in the WPF slices over no muting. However, it has 25% lower PF metric than RadioNinja for the PF slices, with 19% of these slices having even worse performance than without muting. SOM-PF, in contrast, fares as well as RadioNinja for the PF slices, but has 21% higher median FCT (or lower reduction in FCT) than RadioNinja for the prioritized users in WPF slices.

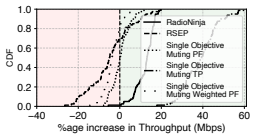
(iii) RadioNinja vs RSEP: RSEP underperforms RadioNinja across the board in this scenario due to its channel agnostic slicing. It has 48% higher median FCT for the prioritized users in WPF slices when compared to RadioNinja. It further achieves 39% lower throughput than RadioNinja for the non-prioritized users in the WPF slices. For the PF slices, RSEP has 37.0% lower PF metric compared to RadioNinja on an average, with almost 40% slices performing worse than RadioSaber no muting.

Parameter	PF Slices	TP Slices	WPF Slices
# Slices	4	4	4
Traffic Pattern	Backlogged	Backlogged	• 40% Web Flow (W = 1000x) • 60% Backlogged
Objective	PF	Max Throughput	Weighted PF
Metric	$\sum \log x_i$	avg tpt	Median FCT
Result	Fig. 10b	Fig. 10c	Fig. 10d

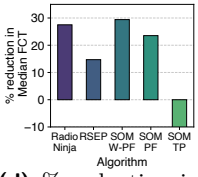
(a) Slice Level Metrics for Scenario 4



(b) Per slice performance for PF-slices



(c) Per slice performance for MT-slices



(d) % reduction in median FCT for web flows

Figure 10 RadioNinja's, RSEP's and SOM's impact on slice-level performance when compared to RadioSaber (no muting) for the (b) PF slices (b) MT slices and (c) Weighted PF Slices

Scenario 4: Varying Number of Slices

To test the scalability of RadioNinja across variable number of slices, in this scenario (summarized in Table 10a), we increase the number of slices to 12, consisting of three categories: PF slices (4 slices) optimizing for proportional fairness ($\alpha = 1$), MT slices (4 slices) optimizing for throughput, and WPF slices (4 slices) using a weighted proportional fairness objective. The WPF slices assign a high weight (1000) to 40% of randomly selected users following a web flow traffic pattern, while the remaining 60% of users have a weight of 1 and follow a backlogged traffic pattern.

(i) RadioNinja vs RadioSaber without muting: RadioNinja maintains its trend of ensuring slice-level isolation, providing targeted performance improvements across slices while incurring minimal loss (only a small subset of MT slices see a slight -5% drop in performance). For the PF slices, RadioNinja improves the PF metric by an average of 86%, while for the MT slices, it increases throughput by 12%. Among WPF slices, it reduces the median FCT of prioritized internet flows by 27%.

(ii) RadioNinja vs SOM: We evaluate all three SOM variants, each optimizing a different global objective (PF, MT, WPF). As seen in previous scenarios, SOM benefits slices aligned with its chosen metric but imposes severe penalties on others. SOM-PF improves the PF metric by 72% (14% lower than RadioNinja) on average while limiting performance degradation to only 5% of the slices. However, it performs poorly for MT slices, offering only a 0.3% throughput improvement (11.3% lower than RadioNinja) while degrading 49% of the slices. SOM-WPF reduces the median FCT of prioritized internet flows by approximately 30% but severely impacts MT slices, decreasing throughput for 80% of them by an average of -5%, with some slices experiencing drops as large as -19%. SOM-MT performs well for MT slices, improving throughput by 39%, but drastically degrades the PF slices, reducing their PF metric by an average of -200%, as seen in Fig. 10b. It also increases median FCT by up to 500%, though we cut off the y-axis at -10% to avoid distorting the figure.

(iii) RadioNinja vs RSEP: Due to its channel-unaware scheduling, RSEP consistently underperforms compared to RadioNinja. It provides only a 6% improvement for PF slices (80% lower than RadioNinja) while degrading 21% of slices. It further reduces the average throughput of MT slices by -5% and offers only a 13% median FCT improvement (14% lower than RadioNinja).

Scenario 5: Unequal Slice Weights Across Cells

We also evaluate the performance of RadioNinja under skewed user distribution patterns. The traffic parameters for this experiment remain the same as those described in Table 7a. The ***key difference*** *is that approximately 75% of the users in the Max Throughput (MT) slices are now attached to the Macro Cell, with only 25% attached to the Small Cells.* As a

(a) Per slice performance of PF slices

(b) Per slice performance of MT slices

Figure 11 RadioNinja 's, RSEP's and SOM-PF's and SOM-TP's impact on slice-level performance when compared to RadioSaber (no muting) for (a) PF slices and (b) Max TP slices under Scenario 1 (Table 7a) with unequal slice quotas and distribution across cells.

result, the quota of the Max Throughput slices gets increased proportionally at the Macro Cell and reduced at the Small Cells. Conversely, *for the Proportional Fairness (PF) slices, the distribution is reversed, with 75% of users attached to the Small Cells and the remaining 25% to the Macro Cell.* The results are shown in Figure 11.

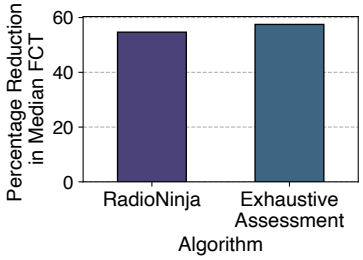
(i) RadioNinja vs RadioSaber without muting: RadioNinja continues to demonstrate its effectiveness, offering a mean improvement of 10.2% in the PF metric for PF slices, while maintaining slice-level isolation with near-zero loss across all slices (approximately 5% of slices experience a minor mean loss of 1.5%). For MT slices, RadioNinja provides a 12.8% improvement in throughput, again ensuring no performance degradation for any slice.

(ii) RadioNinja vs SOM: Under this unequal distribution, both variants of SOM—configured with either the PF or MT metric—fail to perform well even on their respective metrics, breaking slice-level isolation and degrading performance for most slices - 90% of all slices under SOM-PF and 50% of all slices under SOM-MT see a reduction in their slice level metric than compared to the no muting baseline. This failure stems from the skewed distribution of users, where there are more MT users attached to the macro-cell and more PF users attached to the small cells. So when SOM-PF attempts to maximize the PF metric globally across all cells, it inadvertently includes PF scores of macro-cell users (largely comprised of MT slices) in its global optimization decisions. Since these MT users inherently prioritize throughput over fairness, their inclusion distorts the PF metric, resulting in poorly informed decisions that harm slice-level objectives. Similarly, SOM-MT faces a similar issue: by trying to globally maximize throughput, it incorporates MT scores from the PF users at small cells, whose scheduling decisions focus on fairness rather than throughput. This mismatched metric aggregation leads to suboptimal global optimization and significantly impacts the individual objectives of the slices.

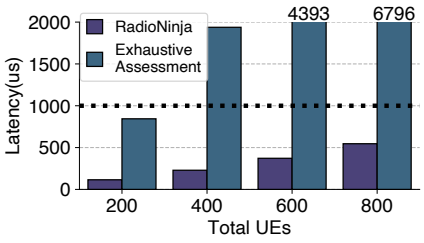
(iii) RadioNinja vs RSEP: The unequal distribution exacerbates mismatches between RBs across cells, breaking RSEP's RB-linkage rule. This prevents RSEP from effectively managing interference on those RBs, further compounded by its channel-unaware allocation mechanism. As a result, 90% of all PF slices and 76% of all MT slices experience degraded performance compared to the no muting baseline.

Key Takeaway

RadioNinja consistently improves slice-level performance over RadioSaber (channel-aware slicing without muting) for a substantial fraction of slices, while preserving performance isolation—no slice is penalized by interference management decisions. In contrast, RSEP has inferior performance, in some cases even relative to RadioSaber, due to its rigid assignment of RBGs across cells that ignores channel diversity. Single-objective muting (SOM) typically benefits slices whose objectives align with the chosen global objective, but systematically



(a) Reduction in Median FCT



(b) Runtime latency

Figure 12 Comparing RadioNinja with exhaustive assessment in terms of (a) performance and (b) computational overhead.

penalizes slices with misaligned objectives, thereby violating performance isolation. Even when performance objectives are aligned, single-objective muting breaks down under skewed user distributions, since the global objective no longer serves as a good approximation of slice-level objectives. We further stress-test RadioNinja across a wide range of system configurations – varying the number of users and cells (results are excluded for brevity but the same trends hold), and considering deployments with multiple categories of slice objectives. Across all these settings, the same qualitative trends hold: RadioNinja continues to improve slice-level objectives without violating slice-isolation.

6.3 Comparison with Exhaustive Assessment

In Fig. 12a we show how RadioNinja (with its heuristics to assess benefit and cost of muting without repeatedly re-running the scheduler) compares against a variant that re-runs the schedulers for a more rigorous assessment (we refer to this variant as exhaustive assessment). We find that RadioNinja's well-designed approximations produce no significant difference (+2.8%) in the outcome. For brevity, we only report a key result for scenario 3 – median FCT of prioritized users in WPF slices –(Fig. 12a), but trend holds more generally.

We use the software implementation (described in §5) on a single core to evaluate the runtime latency of both RadioNinja and the exhaustive assessment baseline. Fig. 12b shows the results with 5G numerology 1 configuration (500μs TTI and 32 RBGs), four small cells and five slices. We vary the total number of users from 200 to 800 (3GPP standards recommend 200-400 number of users for a cluster of 1 macro-cell and four small cells [2]). We find that RadioNinja's runtime latency comfortably fits within the TTI budget with up to 600 users. The exhaustive assessment approach, in contrast, exceeds the TTI budget even with 200 users. The exhaustive approach can potentially be parallelized to use more cores and reduce latency, but the increased cost and power might not be worth it given that RadioNinja's approximations work as well (as shown in Fig. 12a). We repeat these experiments for other 5G numerologies in Appendix D, confirming the same trends.

7 Concluding Remarks

In this paper, we present RadioNinja, the first system that performs channel-aware RAN slicing across multiple cells while managing interference, maintaining isolation between slices and supporting customizable and diverse objectives within each slice. While our work focuses on downlink transmissions, we believe similar insights can also be applied to uplink. Moreover, we believe RadioNinja's framework is not limited to muting alone, but can be applied more broadly to any coordination-based interference management scheme (we leave detailed exploration to future work).

References

1 3GPP. Coordinated multi-point operation for lte physical layer aspects (3gpp release 11 3gpp tr 36.819 v11.2.0 2013-09). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2498, 2013.

2 3GPP. Further advancements for e-utra physical layer aspects. Technical Report 36.814, 3rd Generation Partnership Project (3GPP), 2020. Version 9.0.0. URL: https://www.3gpp.org/ftp/specs/archive/36_series/36.814/.

3 3rd Generation Partnership Project (3GPP). Physical Layer Procedures for Data. Technical Report TS 138 214, ETSI, 2021. URL: https://www.etsi.org/deliver/etsi_ts/138200_138299/138214/15.03.00_60/ts_138214v150300p.pdf.

4 Rajeev Agrawal, Anand Bedekar, Suresh Kalyanasundaram, Naveen Arulselvan, Troels Kolding, and Hans Kroener. Centralized and decentralized coordinated scheduling with muting. In *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2014. doi:10.1109/VTCSpring.2014.7022856.

5 Jeffrey G. Andrews, Stefano Buzzi, Wan Choi, Stephen V. Hanly, Angel Lozano, Anthony C. K. Soong, and Jianzhong Charlie Zhang. What will 5g be? *IEEE Journal on Selected Areas in Communications*, 32(6):1065–1082, 2014. doi:10.1109/JSAC.2014.2328098.

6 5G Slicing Association. 5g network slicing self-management white paper. https://www-file.huawei.com/-/media/corporate/pdf/news/5g-network-slicing-self-management-white-paper.pdf?la=en-us, 2020.

7 Gunther Auer, Vito Giannini, Claude Desset, Istvan Godor, Per Skillermark, Magnus Olsson, Muhammad Ali Imran, Dario Sabella, Manuel J. Gonzalez, Oliver Blume, and Albrecht Fehske. How much energy is needed to run a wireless network? *IEEE Wireless Communications*, 18(5):40–49, 2011. doi:10.1109/MWC.2011.6056691.

8 Naga Bhushan, Junyi Li, Durga Malladi, Rob Gilmore, Dean Brenner, Aleksandar Damnjanovic, Ravi Teja Sukhavasi, Chirag Patel, and Stefan Geirhofer. Network densification: the dominant theme for wireless evolution into 5g. *IEEE Communications Magazine*, 52(2):82–89, 2014.

9 Federico Boccardi, Robert W. Heath, Angel Lozano, Thomas L. Marzetta, and Petar Popovski. Five disruptive technology directions for 5g. *IEEE Communications Magazine*, 52(2):74–80, 2014. doi:10.1109/MCOM.2014.6736746.

10 Pablo Caballero, Albert Banchs, Gustavo de Veciana, and Xavier Costa-Pérez. Multi-tenant radio access network slicing: Statistical multiplexing of spatial loads. *IEEE/ACM Transactions on Networking*, 25(5):3044–3058, 2017. doi:10.1109/TNET.2017.2720668.

11 F. Capozzi, G. Piro, L.A. Grieco, G. Boggia, and P. Camarda. Downlink packet scheduling in lte cellular networks: Key design issues and a survey. *IEEE Communications Surveys Tutorials*, 15(2):678–700, 2013.

12 Yongzhou Chen, Ruihao Yao, Haitham Hassanieh, and Radhika Mittal. Channel-Aware 5g RAN slicing with customizable schedulers. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 1767–1782, 2023.

13 Essentra Components. A guide to 5G small cells and macrocells. https://www.essentracomponents.com/en-us/news/industries/telecoms-data/a-guide-to-5g-small-cells-and-macrocells.

14 Cricket. Cricket Wireless. https://www.cricketwireless.com/.

15 Erik Dahlman, Stefan Parkvall, and Johan Sköld. Chapter 13 - multi-point coordination and transmission. In Erik Dahlman, Stefan Parkvall, and Johan Sköld, editors, *4G LTE-Advanced Pro and The Road to 5G (Third Edition)*, pages 331–345. Academic Press, third edition edition, 2016. URL: https://www.sciencedirect.com/science/article/pii/B9780128045756000133, doi:10.1016/B978-0-12-804575-6.00013-3.

16 Mariana Dirani, Zwi Altman, and Mikael Salaun. Chapter 7 - autonomics in radio access networks. In Nazim Agoulmine, editor, *Autonomic Network Management Principles*, pages 141–166. Academic Press, Oxford, 2011. URL: https://www.sciencedirect.com/science/article/pii/B9780123821904000073, doi:10.1016/B978-0-12-382190-4.00007-3.

17 Salvatore D'Oro, Francesco Restuccia, Alessandro Talamonti, and Tommaso Melodia. The slice is served: Enforcing radio access network slicing in virtualized 5g systems. In *IEEE INFOCOM*, page 442–450, 2019.

18 Ericsson. Ericsson launches 5g ran slicing to spur 5g business growth. https://www.ericsson.com/en/press-releases/2021/1/ericsson-launches-5g-ran-slicing-to-spur-5g-business-growth, Jan 2021. Accessed: 2026-01-08.

19 Ericsson. Ericsson, tim and comau test the factory of the future thanks to 5g network slicing. https://www.ericsson.com/en/press-releases/3/2022/ericsson-tim-and-comau-test-the-factory-of-the-future-thanks-to-5g-network-slicing, Jun 2022. Accessed: 2026-01-08.

20 Ericsson. Maximizing capacity in spectrum-limited networks. https://www.ericsson.com/en/reports-and-papers/microwave-outlook/articles/maximizing-capacity-in-spectrum-limited-networks, 2023.

21 Ericsson. Network slicing – enabling differentiated connectivity. https://www.ericsson.com/en/network-slicing, 2026. Accessed: 2026-01-08.

22 ETSI. 5G; 5G System; Network Slice Selection Services (3GPP TS 29.531 version 15.1.0 Release 15). https://www.etsi.org/deliver/etsi_ts/129500_129599/129531/15.01.00_60/ts_129531v150100p.pdf, 2018.

23 ETSI. Evolved universal terrestrial radio access (e-utra); physical layer procedures. https://www.etsi.org/deliver/etsi_ts/136200_136299/136213/15.10.00_60/ts_136213v151000p.pdf, 2020.

24 Xenofon Foukas, Mahesh K. Marina, and Kimon Kontovasilis. Orion: Ran slicing for a flexible and cost-effective multi-service mobile network architecture. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, 2017. doi:10.1145/3117811.3117831.

25 Xenofon Foukas, Navid Nikaein, Mohamed M. Kassem, Mahesh K. Marina, and Kimon Kontovasilis. Flexran: A flexible and programmable platform for software-defined radio access networks. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*. Association for Computing Machinery, 2016. doi:10.1145/2999572.2999599.

26 Xenofon Foukas, Georgios Patounas, Ahmed Elmokashfi, and Mahesh K. Marina. Network slicing in 5g: Survey and challenges. *IEEE Communications Magazine*, 2017. doi:10.1109/MCOM.2017.1600951.

27 Google. Google Fi Wireless. https://fi.google.com/about/.

28 Shalini Gulati, Suresh Kalyanasundaram, Prakhar Nashine, Balamurali Natarajan, Rajeev Agrawal, and Anand Bedekar. Performance analysis of distributed multi-cell coordinated scheduler. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pages 1–5, 2015. doi:10.1109/VTCFall.2015.7391069.

29 Yan Huang, Shaoran Li, Y. Thomas Hou, and Wenjing Lou. GPF: A GPU-Based Design to Achieve 100us Scheduling for 5G NR. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, 2018. doi:10.1145/3241539.3241552.

30 Intel. FlexRAN Reference Architecture for Wireless Access. https://www.intel.com/content/www/us/en/developer/topic-technology/edge-5g/tools/flexran.html.

31 Ralf Irmer, Heinz Droste, Patrick Marsch, Michael Grieger, Gerhard Fettweis, Stefan Brueck, Hans-Peter Mayer, Lars Thiele, and Volker Jungnickel. Coordinated multipoint: Concepts, performance, and field trial results. *IEEE Communications Magazine*, 49(2):102–111, 2011. doi:10.1109/MCOM.2011.5706317.

32 V. Jungnickel, L. Thiele, T. Wirth, T. Haustein, S. Schiffermuller, A. Forck, S. Wahls, S. Jaeckel, S. Schubert, H. Gabler, C. Juchems, F. Luhn, R. Zavrtak, H. Droste, G. Kadel, W. Kreher,

J. Mueller, W. Stoermer, and G. Wannemacher. Coordinated multipoint trials in the downlink. In *2009 IEEE Globecom Workshops*, pages 1–7, 2009. doi:10.1109/GLOCOMW.2009.5360722.

33 Frank P. Kelly, A. K. Maulloo, and David Kim Hong Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *J. Oper. Res. Soc.*, 49(3):237–252, 1998. doi:10.1057/palgrave.jors.2600523.

34 Ravi Kokku, Rajesh Mahindra, Honghai Zhang, and Sampath Rangarajan. Nvs: A virtualization substrate for wimax networks. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, 2010. doi:10.1145/1859995.1860023.

35 Ravi Kokku, Rajesh Mahindra, Honghai Zhang, and Sampath Rangarajan. Nvs: A substrate for virtualizing wireless resources in cellular networks. *IEEE/ACM Trans. Network.*, oct 2012. doi:10.1109/TNET.2011.2179063.

36 Raymond Kwan, Cyril Leung, and Jie Zhang. Proportional fair multiuser scheduling in lte. *IEEE Signal Processing Letters*, 2009. doi:10.1109/LSP.2009.2016449.

37 Nikita Lazarev, Tao Ji, Anuj Kalia, Daehyeok Kim, Ilias Marinos, Francis Y. Yan, Christina Delimitrou, Zhiru Zhang, and Aditya Akella. Resilient baseband processing in virtualized rans with slingshot. In *Proceedings of the ACM SIGCOMM 2023 Conference*, ACM SIGCOMM '23, page 654–667, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3603269.3604841.

38 Daewon Lee, Hanbyul Seo, Bruno Clerckx, Eric Hardouin, David Mazzarese, Satoshi Nagata, and Krishna Sayana. Coordinated multipoint transmission and reception in lte-advanced: deployment scenarios and operational challenges. *IEEE Communications Magazine*, 50(2):148–155, 2012. doi:10.1109/MCOM.2012.6146494.

39 George R. MacCartney and Theodore S. Rappaport. Millimeter-wave base station diversity for 5g coordinated multipoint (comp) applications. *IEEE Transactions on Wireless Communications*, 18(7):3395–3410, 2019. doi:10.1109/TWC.2019.2913414.

40 Rajesh Mahindra, Mohammad A. Khojastepour, Honghai Zhang, and Sampath Rangarajan. Radio access network sharing in cellular networks. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–10, 2013. doi:10.1109/ICNP.2013.6733595.

41 Gabriele Manganaro and Domine MW Leenaerts. *Advances in analog and RF IC design for wireless communication systems*. Academic Press, 2013.

42 Rakesh Misra, Aditya Gudipati, and Sachin Katti. Quickc: Practical sub-millisecond transport for small cells. In *Proc. ACM Mobicom*, page 109–121, 2016.

43 Radhika Mittal, Justine Sherry, Sylvia Ratnasamy, and Scott Shenker. Recursively Cautious Congestion Control. In *Proc. USENIX NSDI*, pages 373–385, 2014.

44 Larry Peterson and Oguz Sunay. 5g mobile networks: A systems approach, 2020.

45 Giuseppe Piro, Luigi Alfredo Grieco, Gennaro Boggia, Francesco Capozzi, and Pietro Camarda. Simulating lte cellular systems: An open-source framework. *IEEE Transactions on Vehicular Technology*, 2011. doi:10.1109/TVT.2010.2091660.

46 Petar Popovski, Kasper Fløe Trillingsgaard, Osvaldo Simeone, and Giuseppe Durisi. 5g wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view. *IEEE Access*, 6:55765–55779, 2018. doi:10.1109/ACCESS.2018.2872781.

47 Raj Radjassamy. Think Small (Cell) to Go Big on 5G. RCR Wireless White Paper, 2021.

48 Oscar D Ramos-Cantor, Jakob Belschner, Ganapati Hegde, and Marius Pesavento. Centralized coordinated scheduling in lte-advanced networks. *EURASIP Journal on Wireless communications and Networking*, 2017(1):1–14, 2017.

49 Anil Rao. 5g network slicing: crossdomain orchestration and management will drive commercialization. https://www.cisco.com/c/dam/en/us/products/collateral/cloud-systems-management/network-services-orchestrator/white-paper-sp-5g-network-slicing.pdf, 2020.

50 ABI Research. 5G Data Traffic Explosion Will Drive 5G Small Cell Deployments to 13 million by 2027. https://www.abiresearch.com/press/5g-data-traffic-explosion-will-drive-5g-small-cell-deployments-to-13-million-by-2027, 2022.

51 Nazmus Saquib, Ekram Hossain, Long Bao Le, and Dong In Kim. Interference management in ofdma femtocell networks: issues and approaches. *IEEE Wireless Communications*, 19(3):86–95, 2012. doi:10.1109/MWC.2012.6231163.

52 sharetech. Resource allocation type. https://www.sharetechnote.com/html/Handbook_LTE_RAType.html, 2020.

53 ShareTechnote. 5g resource allocation types, 2024. URL: https://www.sharetechnote.com/html/5G/5G_ResourceAllocationType.html.

54 Maraj Uddin Ahmed Siddiqui, Faizan Qamar, Faisal Ahmed, Quang Ngoc Nguyen, and Rosilah Hassan. Interference management in 5g and beyond network: Requirements, challenges and future directions. *IEEE Access*, 9:68932–68965, 2021. doi:10.1109/ACCESS.2021.3073543.

55 STL Partners. 5g network slicing examples: Trials, proofs of concept and collaborations. https://stlpartners.com/articles/private-cellular/5g-network-slicing-examples/, 2023. Accessed: 2026-01-08.

56 Shaohui Sun, Qiubin Gao, Ying Peng, Yingmin Wang, and Lingyang Song. Interference management through comp in 3gpp lte-advanced networks. *IEEE Wireless Communications*, 20(1):59–66, 2013. doi:10.1109/MWC.2013.6472200.

57 Unknown. Moving 5g spectrum to a shared resource. https://www.rcrwireless.com/20170404/5g/20170404analyst-anglemoving-5g-spectrum-shared-resource, apr 2017. Accessed: 2024-04-28.

58 Unknown. US cell towers and small cells: By the numbers. https://www.lightreading.com/digital-transformation/us-cell-towers-and-small-cells-by-the-numbers, 2022.

59 C. Wengerter, J. Ohlhorst, and A.G.E. von Elbwart. Fairness and throughput analysis for generalized proportional fair frequency scheduling in ofdma. In *2005 IEEE 61st Vehicular Technology Conference*, pages 1903–1907 Vol. 3, 2005.

60 Yaxiong Xie and Kyle Jamieson. Ng-scope: Fine-grained telemetry for nextg cellular networks. *Proc. ACM Meas. Anal. Comput. Syst.*, 6(1), feb 2022. doi:10.1145/3508032.

61 Jiarong Xing, Junzhi Gong, Xenofon Foukas, Anuj Kalia, Daehyeok Kim, and Manikanta Kotaru. Enabling resilience in virtualized rans with atlas. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '23, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3570361.3613276.

62 Yiqing Zhou, Ling Liu, Hongyan Du, Lin Tian, Xiaodong Wang, and Jinglin Shi. An overview on intercell interference management in mobile cellular networks: From 2g to 5g. In *2014 IEEE International Conference on Communication Systems*, pages 217–221, 2014. doi:10.1109/ICCS.2014.7024797.

Appendix

A Importance of Small Cells and Frequency Sharing

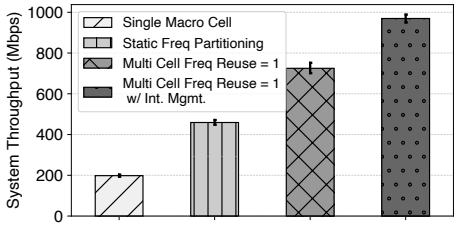


Figure 13 System throughput under different frequency allocation strategies

A key design decision in multi-cell deployments concerns how to partition the radio frequency among cells. One option is to partition the frequency statically, ensuring that neighboring cells never transmit at the same frequency and thus avoid interference. However, this approach leads to inefficient spectrum usage, reducing the usable bandwidth for each cell. An alternative is to allow neighboring cells to share the same frequency band (frequency reuse one). This introduces the challenge of managing interference that would arise when two cells transmit simultaneously on the same frequency, degrading the channel quality users in the overlapping coverage regions of both cells. Nonetheless, dynamically managing interference (depending on when which users suffer from it) still results in better resource utilization and overall system performance compared to statically partitioning radio frequency among cells. These performance benefits of frequency reuse are well-known [20, 16], making it a preferred deployment choice in many settings [57].

To understand the significance of these choices, we conduct a simple experiment, starting with 150 users, attached to a single macro-cell, randomly distributed in its range. We next introduce 4 small cells into the macro-cell's coverage region, but statically partition the frequency (reserving 50% of the bandwidth for the macro-cell, with the remaining 50% used by small cells). This results in 100% higher throughput compared to single cell setting. We next evaluate the multi-cell setup under frequency re-use one, where the entire frequency band is shared across cells. This improves throughput by 70% over the static partitioning setup. Dynamically conducting interference management using the muting strategy described in §3.2 (for throughput maximization objective) further improves throughput by 35%.

B Channel Quality Information for Muting

Assessment of muting decisions require knowing the channel quality of users with and without interference. 5G New Radio (NR) scheduling allows for the configuration of a set of Resource Elements (REs)—smaller individual subcarriers within an RBG—to be dedicated for the purpose of interference measurement, known as the Channel State Indicator Interference Measurement Resource (CSI-IM) [3]. Coordinating cells periodically co-schedule known reference symbols and zero-power symbols sequentially, based on the Channel Feedback interval (typically 40 ms), to achieve accurate per-cell interference values. For example, on Resource Element r_i, Cell C_j would transmit a known Reference Symbol while Cell C_{j+1} to C_n would schedule a Zero Power Reference Symbol (ZP-CSI-RS); here, Cell C_j is transmitting and creating interference, while all other cells are 'quiet' and only measuring

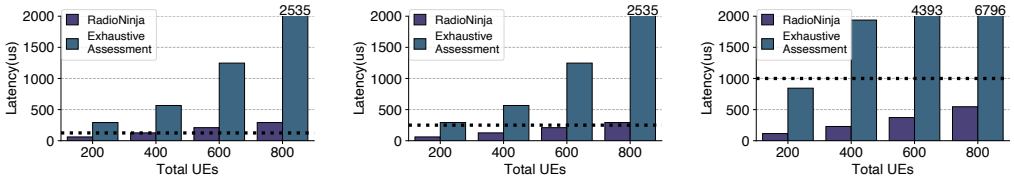
the interference created by Cell C_j [15]. This process is iteratively repeated by all the cells in the coordinating cluster—one cell transmitting and the rest remaining quiet on those REs and measuring the transmitter's interference—to achieve accurate interference values. This measurement is cost-effective since only a subset of an RBG (typically 2 out of 48 REs per RBG) are allocated for CSI-IM, allowing normal data transmission on all remaining REs.

C Details about RadioNinja's Cost Computation

We take the scores associated with each RBG allocated to a given slice at a given cell under the initial allocation (mentioned in §4.3), and group these RBGs into clusters. Our clustering logic is cheap and simple – we compute the average score (μ) and the standard deviation (σ) across all RBGs allocated to the slice, and cluster them into atmost five sets – RBGs with scores (i) $\leq (\mu - 2\sigma)$, (ii) $> (\mu - 2\sigma)$ and $\leq (\mu - \sigma)$, (iii) within $(\mu \pm \sigma)$, (iv) $\geq (\mu + \sigma)$ and $< (\mu + 2\sigma)$, and (v) $\geq (\mu + 2\sigma)$.

We then compute the average score within each cluster, and use them to assess the worth of an RBG. Suppose the cluster with lowest score for a given slice S_k at cell C_j has n RBGs and an average score of X. As long as the total quota reduced from slice S_k due to muting at C_j remains less than n RBGs, we associate a worth of X for each RBG (so a quota reduction of 0.5RBGs will incur a cost of $0.5X$). Once the quota reduction for S_k exceeds n RBGs (due to muting decisions made over multiple RBGs), we move over to considering the cluster with the next lowest average score, and so on. The clustering avoids excessive overfitting to the specific RBG scores in the initial hypothetical allocation.

D Runtime Overhead



(a) TTI = 125 **(b)** TTI = 250 **(c)** TTI = 1000

Figure 14 The runtime latency of RadioNinja and the exhaustive assessment approach with 5G numerology 0, 2, and 3. The corresponding TTIs are 125us, 250us, and 1000us.

Now we discuss how the numerology, the number of small cells, and the number of slices affect the runtime latency of both RadioNinja and the exhaustive assessment baseline. In the first experiment, we have five slices and five cells(same configuration in §6.3), and evaluate both systems with 5G numerology 0(32RBGs, 1000us TTI), 2(16RBGs, 250us TTI), and 3(16RBGs, 125us TTI). Figure14 shows the runtime latency of both systems with different number of users. We can see the runtime latency of the exhaustive baseline is well above one TTI in all numerology configurations with only 400 users. Nevertheless, RadioNinja can support up to 600 users in numerology 0, 2, and up to 400 users in numerology 3. This concludes that RadioNinja can be practical with most 5G numerology configurations.

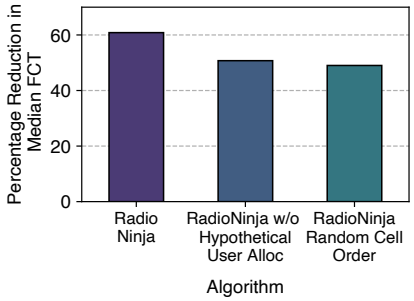


Figure 15 Impact of RadioNinja system optimizations on Median FCT

E Factor Analysis

To evaluate the impact of our system optimizations discussed in Section §6, we measure system performance in their absence. We use the same underlying traffic distribution as in Table 9a and record the reduction in median FCT for the prioritized web flows. The results are reported in Fig. 15.

Impact of hypothetical baseline when computing benefit

RadioNinja computes the benefit of muting by comparing the user score a slice achieves at a neighboring cell with a hypothetical allocation where we compute the score that the same slice would have achieved on the same RBG without muting. We try a variant where we directly use the score achieved by a slice at a neighboring cell without subtracting the score from this hypothetical baseline. We find that it produces notably smaller reduction in median FCT of prioritized users (10% lower than original RadioNinja) when compared to our original design (Fig. 15 first bar from the left).

Impact of cell selection heuristic

With multiple valid muting hypothesis across cells, RadioNinja selects one based on benefit-cost ratio across the set of benefitting slices and side-effect on other slices. We implement a variant where we randomly select a hypothesis (which cell to mute) among multiple valid ones. This simplified variant again produces a notably smaller reduction in median FCT (11% lower than our original design as shown in Fig. 15), justifying the usefulness of our optimizations in this context.