The Effect of Additive and Multiplicative Scheduler Weight Adjustments on 5G Slicing Dynamics

Matthew Andrews  Sem Borst  Siegfried Klein  Hans Kroener  Silvio Mandelli
Nokia

Abstract—We consider the problem of provisioning slices in 5G cellular networks and study two MAC scheduling methods for achieving slice rate and resource share constraints. The first uses an additive adjustment to the standard Proportional Fair scheduler weight and optimizes utility subject to the slice constraints. The second uses a multiplicative adjustment and achieves fairness between the users within a slice in addition to satisfying the slice constraints. We provide a comprehensive analysis of the tradeoffs between utility and fairness for the simple case of constant channel rates and use simulations to illustrate that similar effects hold for typical 3GPP channel models.

I. INTRODUCTION

We compare methods for achieving slice guarantees in cellular networks by adjusting the MAC scheduling weights. A network slice is a portion of a network that is carved out for a particular set of users or a particular type of traffic. For example, one slice provided by a national operator could be used to support a Mobile Virtual Network Operator (MVNO), another could support Ultra-Reliable Low Latency Communication for vehicular applications, and a third could support an enterprise application such as an advanced wireless-enabled factory. Slicing support is expected to be one of main differentiators for 5G wireless in comparison to LTE.

We refer to the mobile operator that owns the network as the slice host and the entities that purchase capacity on the network as slice tenants. Typically the host operator will provision capacity for a number of different tenants and so each tenant will want assurances about the levels of performance their traffic will receive. Hence a Service Level Agreement (SLA) will be negotiated between the two parties that specifies the level of capacity and delay performance that will be provided.

In general, the provision of slicing SLAs involves the solution of two problems. The end-to-end slicing problem takes the overall negotiated SLA and converts it into requirements for individual cells. The per-cell slicing problem takes the cell requirement calculated in the end-to-end problem and determines how to satisfy it at the cell. In this work we will focus on solving the per-cell slicing problem using MAC scheduling. The end-to-end problem is challenging because the users for a slice may be distributed non-uniformly among cells, but will not be addressed here. A game-theoretic treatment of the end-to-end problem is covered in [10].

A. The problem

We now describe our problem at a high level. A formal mathematical description is given in Section I-C. We consider a single 5G basestation operating in slotted time and serving a set $I = \{1, \ldots, M\}$ of end users.$^1$ Let $R_i$ and $X_i$ represent, respectively, the long-term service rate and the long-term share of resources achieved by user $i$. The key problem we consider is providing guarantees to a set of slices, $J$. Each slice consists of a (possibly weighted) subset of the users and has requirements on the aggregate service rate or aggregate share of resources provided to the slice. In addition, we may wish to impose a (weighted) intra-slice resource fairness constraint which specifies that the share of resources given to each user within a slice should be proportional to a given set of weights.

A typical objective in MAC scheduling is to maximize the aggregate throughput utility $\sum_{i \in I} U(R_i)$ for some (concave) utility function $U(\cdot)$. This can be achieved by a scheduling algorithm which allocates resources to users at each time slot so as to maximize $\sum_{i \in I} U'(R_i) S_i$, where $S_i$ denotes the total service rate received by user $i$ during the time slot. The term $U'(R_i)$ is referred to as the scheduling weight of user $i$, while $U'(R_i) S_i$ is called the scheduling metric. A key special case is the well-known Proportional Fair (PF) algorithm which corresponds to a utility function $U(\cdot) = \log(\cdot)$.

The above-described algorithm is known to optimize the aggregate utility [3], [12]. However, it does not account for the slicing constraints and hence may not satisfy them. In [11], it is shown that by introducing additive adjustments to the default scheduling metrics, we can optimize utility subject to slice service rate targets and slice resource share constraints.

When considering this result, a natural question is whether this is the only type of scheduler metric adjustment that would achieve this result. In particular, since single-user rate guarantees have been obtained with multiplicative adjustments [4], we ask whether such schemes could produce the same performance for slicing as additive offsets. We also examine the effects of these options on fairness within a slice. Qualitatively, our results are:

- Additive adjustments satisfy slicing constraints whenever feasible to do so at all, and maximize utility subject to these constraints. However, in case of slice rate targets, this optimality generally comes at the expense of fairness among users within a slice, i.e., intra-slice fairness. (Due to space restrictions we omit some of the proofs.)
- Multiplicative adjustments maintain intra-slice fairness, and satisfy the slicing constraints whenever feasible to do so under that additional requirement. Moreover, they maximize utility subject to the slicing constraints and the intra-slice fairness condition. However, the intra-slice fairness requirement im-

$^1$In practical systems each User Equipment (UE) may have a number of distinct flows (that are sometimes called Data Radio Bearers in the 5G nomenclature). For simplicity we shall identify users with flows and so multiple users may correspond to the same physical UE.
pacts the feasibility of slice rate targets, and lowers the utility compared to additive offsets.

We establish these results as follows:

- We start with the special case of constant, frequency-independent channels which allow an explicit analysis of user service rates and resource shares. We confirm the utility-optimality of additive adjustments and demonstrate that multiplicative adjustments provide intra-slice fairness. If there are slice resource share constraints only (i.e. no slice rate constraints), then both types of adjustments achieve both utility optimality and intra-slice fairness. However, if slice rate constraints are present, then neither type of adjustment achieves both utility-optimality and intra-slice fairness. Lastly, if we wish to impose slice rate and resource constraints and also impose intra-slice fairness, then this shrinks the feasible rate region. In this scenario multiplicative adjustments can optimize utility over this restricted region.

- While the case of constant channels is restrictive, the main insights in fact carry over to the realistic case of time-varying channels. In particular, the utility-optimality of additive adjustments in [11] applies, and we will establish an approximate notion of intra-slice fairness for multiplicative adjustments. We support our observations via 5G-compliant simulations.

B. Related work

As slice support has become a desired feature of cellular networks, a number of studies have examined schemes for sharing radio resources among slices. For example, Ksentini & Nikaer [9] consider a two-level MAC scheduler that has separate mechanisms for inter-slice and intra-slice scheduling. Cabellero et al. [5] examine methods for providing weighted Proportional Fairness among slices. For the specific problem of slice-aware MAC scheduling Kasgari & Saad [7] utilize a Lyapunov drift-plus-penalty method and Khodapanah et al. [8] address slice aggregate rate targets via slice-dependent multipliers applied to the regular scheduling weights. The current work is a follow-up to [11] which showed how additive adjustments can be used to optimize utility subject to rate and resource constraints. Our goal in this follow-up is to examine in detail the implications of additive vs. multiplicative slice-aware scheduler adjustments.

C. Formal model and results

We consider a single basestation serving a set \( I = \{1, \ldots, M\} \) of \( M \) users on \( F \) frequencies. Time is divided into time slots or Transmission Time Intervals (TTIs). The job of the scheduler is to allocate time slots and frequencies among the \( M \) users. Let \( A(f, \tau) \) be the rate region in time slot \( \tau \), i.e. a vector \( (s_1, \ldots, s_M) \in A(f, \tau) \) if and only if we can simultaneously serve user \( i \) at rate \( s_i \) for all \( i \in I \). We assume that the rate region \( A(f, \tau) \) is available to the scheduler due to channel feedback. It also encompasses physical-layer features such as MIMO and beamforming. In case only one user can be served on frequency \( f \) in time slot \( \tau \), we use \( A_s(f, \tau) \) to denote the data rate that user \( i \) would receive if it is scheduled.

For a given schedule let \( S(f, \tau) \) denote the service rate achieved by user \( i \) on frequency \( f \) at time \( \tau \), and let \( Y(f, \tau) \) denote the corresponding fraction of the frequency allocated to user \( i \). We track the average rate \( R_i(\tau) \) and average fraction of resources \( X_i(\tau) \) obtained by user \( i \) according to, \( R_i(\tau) = (1-\delta)R_i(\tau-1) + \delta S_i(\tau) \), and \( X_i(\tau) = (1-\delta)X_i(\tau-1) + \delta Y_i(\tau) \), for some small parameter \( \delta \). Here \( S_i(\tau) = \sum_f S_i(f, \tau) \) and \( Y_i(\tau) = \sum_f Y_i(f, \tau) \).

Slicing Constraints: We assume a set of slices \( J \) each of which is defined by a set of non-negative coefficients \( (\alpha_{i,j})_{i \in I} \). The slice rate and resource share constraints for slice \( j \) are given by,

\[
\begin{align*}
\beta_j^\text{min} & \leq \sum_{i \in I} \alpha_{i,j} R_i(\tau) \leq \beta_j^\text{max} \quad (1) \\
\zeta_j^\text{min} & \leq \sum_{i \in I} \alpha_{i,j} X_i(\tau) \leq \zeta_j^\text{max} \quad (2)
\end{align*}
\]

for some specified bounds \( \beta_j^\text{min}, \beta_j^\text{max}, \zeta_j^\text{min}, \zeta_j^\text{max} \) (some of which may be 0 or \( \infty \)). As described earlier, these bounds would be derived from the slice SLAs as part of the end-to-end slicing problem. Since the end-to-end problem is not the focus of this work, we assume that these bounds are given as input. Note that a natural special case is one where \( \alpha_{i,j} \in \{0,1\} \) for each \( i,j \) and so each slice corresponds to a subset of the users. If in addition each user is a member of at most one slice then we say that the slices are non-overlapping.

Intra-Slice Fairness: In addition to the rate and resource share constraints described above, operators may also wish to provide (Weighted) Intra-Slice Resource Fairness (ISRF) for the case of non-overlapping slices (which implies \( \alpha_{i,j} \in \{0,1\} \)). This condition holds when the long-term resource shares for the users in the same slice are proportional to a set of pre-specified per-user weight factors \( \nu_i \), and in particular equal in case of identical per-user weights. That is, for each slice \( j \) there is a constant \( \chi_j \) such that \( \lim_{T \to \infty} \frac{1}{T} \sum_{\tau=1}^T \alpha_{i,j} Y_i(\tau) = \chi_j \alpha_{i,j} \nu_i \) for all \( i \).

ISRF concerns the long-term statistics of the resource allocation decisions, and is commonly viewed as a favorable feature of the PF scheduler. Tenant operators may therefore wish to provide this same property within the resources that they receive from the host. While the PF scheduler indeed yields ISRF for constant channel rates, this generally only holds in an approximate sense for variable channel rates, even in the absence of any slicing constraints.

A concept that is related to ISRF is Nondiscriminatory Intra-Slice Treatment (NIST) in which the priority order among users in the same slice is equivalent to the priority order provided by the scheduling metric \( \nu_i U(\rho_s(\tau)) S_i(\tau) \) (which is not slice aware). NIST pertains to the criterion driving the slot-by-slot resource allocation decisions, and is commonly viewed as a favorable proxy for ISRF accordingly.

Problem Statement: Once we introduce slices our problem becomes maximizing \( \sum_i U(R_i(\tau)) \) subject to the constraints (1) and (2). We consider the problem both with and without the additional requirement of ISRF. In [11] it was shown that if ISRF is not imposed, the following Scheduling Metrics for Slicing (SMS) algorithm achieves the optimal solution. In particular, (1) and (2) are maintained using token
To distinguish this algorithm from alternative approaches that adjust the scheduling metric in a multiplicative manner, we note that SMSm achieves both utility-optimality and ISRF while maximizing the utility subject to that additional requirement. We also establish that for non-overlapping slices, SMSm provides a loose notion of intra-slice fairness in the form of NIST. We support these claims via 3GPP-compliant simulations.

II. CONSTANT CHANNELS

By constant channels we mean that there is a set of constants $C_{i,f}$ (i.e. not dependent on time) such that the achieved rate $S_i(f, \tau) = C_{i,f} Y_i(f, \tau)$. (Recall that $Y_i(f, \tau)$ is the fraction of frequency $f$ allocated to user $i$ at time $\tau$.) The resource constraint is given by $\sum_i Y_i(f, \tau) \leq k$ for some parameter $k$. (A natural case is when each frequency is allocated to exactly one user in which case $k = 1$. However, if multi-user MIMO is deployed then we might have $k > 1$.)

Let $X_{i,f}$ be the average share of frequency $f$ allocated to user $i$ over time. The problem becomes:

$$
\max \sum_i U(\sum_f C_{i,f} X_{i,f})
$$

subject to:

$$
\begin{align*}
\sum_i X_{i,f} &\leq k \quad \forall f \\
\beta_j^{\text{min}} &\leq \sum_i \alpha_{i,j} \sum_f C_{i,f} X_{i,f} \leq \beta_j^{\text{max}} \\
\varsigma_j^{\text{min}} &\leq \sum_i \alpha_{i,j} \sum_f X_{i,f} \leq \varsigma_j^{\text{max}}
\end{align*}
$$

By the theory of Lagrange multipliers, $X_{i,f}$ is a solution to the above problem if and only if there are constants $Q_j$ and $Z_j$, such that $X_{i,f}$ maximizes:

$$
\begin{align*}
&\sum_i U'(\sum_f C_{i,f} X_{i,f}) C_{i,f} \\
&\quad + \sum_j Q_j \sum_{i,f} \alpha_{i,j} C_{i,f} X_{i,f} + \sum_j Z_j \sum_{i,f} \alpha_{i,j} X_{i,f},
\end{align*}
$$

where,

$$
\begin{align*}
\sum_{i,f} \alpha_{i,j} C_{i,f} X_{i,f} \in \begin{cases} [\beta_j^{\text{min}}, \beta_j^{\text{max}}] & Q_j > 0 \\ [\beta_j^{\text{min}}, \beta_j^{\text{max}}] & Q_j = 0 \\ [\beta_j^{\text{min}}, \beta_j^{\text{max}}] & Q_j < 0
\end{cases}
\end{align*}
$$

and,

$$
\begin{align*}
\sum_{i,f} \alpha_{i,j} X_{i,f} \in \begin{cases} [\varsigma_j^{\text{min}}, \varsigma_j^{\text{max}}] & Z_j > 0 \\ [\varsigma_j^{\text{min}}, \varsigma_j^{\text{max}}] & Z_j = 0 \\ [\varsigma_j^{\text{min}}, \varsigma_j^{\text{max}}] & Z_j < 0
\end{cases}
\end{align*}
$$

The metric (5) employed by SMSa ensures that the expression (8) is maximized. Moreover, the token update rules (3) and (4) ensure that the complementary slackness conditions (9) and (10) are satisfied. This implies that SMSa does indeed maximize the aggregate utility subject to the constraints on slice rate and slice resource allocation.

We now turn our attention to fairness between flows and make the further assumption that the channel rates are frequency-independent, i.e. there is a constant $C_i$ so that $S_i(f, \tau) = C_i Y_i(f, \tau)$. We also consider the special case in which $U(r) = \log(r)$ and so $U'(r) = 1/r$. 

counter $Q_j(\tau)$ and $Z_j(\tau)$ that are updated according to:

$$
Q_j(\tau + 1) = \begin{cases} Q_j(\tau) + \beta_j^{\text{min}} - \sum_{i,f} \alpha_{i,j} S_i(\tau) & \text{if } Q_j(\tau) \geq 0 \\
Q_j(\tau) + \beta_j^{\text{max}} - \sum_{i,f} \alpha_{i,j} S_i(\tau) & \text{if } Q_j(\tau) < 0
\end{cases}
$$

$$
Z_j(\tau + 1) = \begin{cases} Z_j(\tau) + \varsigma_j^{\text{min}} - \sum_{i,f} \alpha_{i,j} Y_i(\tau) & \text{if } Z_j(\tau) \geq 0 \\
Z_j(\tau) + \varsigma_j^{\text{max}} - \sum_{i,f} \alpha_{i,j} Y_i(\tau) & \text{if } Z_j(\tau) < 0
\end{cases}
$$

and $Q_j(\tau)$ drifts positive and so the users with positive $\alpha_{i,j}$ are more likely to be served. A similar effect happens with $Z_j(\tau)$ if $\sum_{i,f} \alpha_{i,j} X_{i,f}(\tau) < \varsigma_j^{\text{min}}$ (and so constraint (2) is violated). For the upper bounds, if $\sum_{i,f} \alpha_{i,j} X_{i,f}(\tau) > \beta_j^{\text{max}}$, then $Q_j(\tau)$ drifts negative and so the users with positive $\alpha_{i,j}$ are less likely to be served. Similary with $Z_j(\tau)$ if $\sum_{i,f} \alpha_{i,j} X_{i,f}(\tau) > \varsigma_j^{\text{max}}$.

Although the approach of [11] does indeed solve the problem of maximizing $\sum_i U(R_i(\tau))$ subject to the constraints (1) and (2), a number of questions remain.

- Is it important that the token counters $Q_j(\tau)$ and $Z_j(\tau)$ are used to adjust the scheduling metric in an additive manner? Another natural solution would be to apply the tokens in a multiplicative manner. Would that achieve the same result?
- Does the SMSa algorithm achieve ISRF for non-overlapping slices? If not, how can ISRF be achieved?

To answer these questions we compare the SMSa algorithm that operates by maximizing (5), with an alternative SMS-multilative (SMSm) algorithm that operates by maximizing:

$$
\sum_{i,f} U'(R_i(\tau)) S_i(\tau) g(\sum_{j,f} \alpha_{i,j} (Q_j(\tau) + Z_j(\tau))),
$$

for some increasing function $g(\cdot)$. For concreteness we shall follow [4] and typically use $g(x) = b^x$ for some constant $b$.

Now that we have set up our model more formally, we can describe our results in more detail.

- We start with the case of constant channel rates which allows a detailed characterization of the user service rates $R_i$ and resource shares $X_i$. We confirm that SMSa maximizes $\sum_{i,f} U(R_i(\tau))$ subject to the slicing constraints (1) and (2), and demonstrate that SMSm provides ISRF while maximizing the utility subject to that additional requirement. We also establish that for non-overlapping slices, SMSa as well as SMSm achieve both utility-optimality and ISRF if the only slice constraints are resource constraints. In contrast, in the presence of rate targets, utility-optimality and ISRF can generally not both be achieved at the same time. Lastly, if we wish to impose ISRF in addition to the slice constraints (1) and (2), then the feasible rate region shrinks but we show that SMSm optimizes utility over this restricted region.

- Although the case of constant channels is artificial, it provides valuable insights into the effect of the weight adjustments on performance which also extend to the case of time-varying channels. As proved in [11], SMSa maximizes $\sum_{i,f} U(R_i)$ subject to the slicing constraints in this case as well, and SMSm provides a loose notion of intra-slice fairness in the form of NIST. We support these claims via 3GPP-compliant simulations.

We start with the case of constant channels is artificial, it provides valuable insights into the effect of the weight adjustments on performance which also extend to the case of time-varying channels. As proved in [11], SMSa maximizes $\sum_{i,f} U(R_i)$ subject to the slicing constraints in this case as well, and SMSm provides a loose notion of intra-slice fairness in the form of NIST. We support these claims via 3GPP-compliant simulations.

![Image]

36
Lemma 1: Suppose that the scheduling metric for user $i$ on frequency $f$ is given by,

$$w_i C_i \frac{r_i}{R_i} + C_i q_i + z_i,$$

for fixed $w_i, q_i, z_i$. The long-term resource share of user $i$ on frequency $f$ is,

$$X_{i,f} = w_i/(M - z_i - C_i q_i),$$

and the long-term average rate received by user $i$ on frequency $f$ is,

$$R_{i,f} = C_i X_{i,f} = C_i w_i/(M - z_i - C_i q_i),$$

with $M$ the unique solution of the equation $\sum_i w_i/(M - z_i - C_i q_i) = k$.

Proof: For SMS we only have a multiplicative adjustment $w_i$. The scheduling metric does not have any additive adjustments $q_i$ or $z_i$. Since the adjustment $w_i$ is the same for each user in a slice, Lemma 1 implies that the long-term resource share is the same for each user in the slice.

We now examine the consequences of a scheduler such as SMSm that achieves ISRF.

Lemma 2: Imposing ISRF (which SMSm does) in addition to the slicing constraints (1) and (2) can lower the value of the optimal utility. Moreover, imposing ISRF can change a feasible problem into an infeasible one.

Proof: Our example has one frequency and six users with channel rates $C_i = i, i = 1, \ldots, 6$. At most one user can be served at a time, i.e. $k = 1$. Slice 1 consists of users 1, 2, 3 (i.e. $\alpha_{i,j} = 1$ for $i \in \{1, 2, 3\}$ and slice 2 consists of users 4, 5. We have $\beta_{\min}^1 = 1.1$ and $C_{\min}^2 = 0.4$. All other slice constraints are nonbinding.

A numerical solution of (7) has value $\log(0.0122)$ and is given by $M = 10.04, q = 1.84, z = 5.05, X = [0.122, 0.157, 0.221, 0.2, 0.2, 0.1], R = [0.122, 0.314, 0.664, 0.8, 1.0, 0.6]$. Note that slice 2 achieves ISRF but slice 1 does not. If we also impose ISRF then the optimal solution is reduced to $\log(0.009)$ and is given by $X = [0.183, 0.183, 0.183, 0.2, 0.2, 0.051], R = [0.183, 0.366, 0.549, 0.8, 1.0, 0.306]$.

For the second part of the result the example is the same except we increase $C_{\min}^2$ to 0.5. In this case a feasible solution is given by $X = [0.1, 0.1, 0.267, 0.25, 0.25, 0.033], R = [0.1, 0.2, 0.8, 0.8, 1.0, 0.198]$ since the aggregate rate for slice 1 is 1.1. However, if we also impose ISRF then the maximum resource share for each user in slice 1 is 0.167 and so the maximum possible aggregate rate for the slice is 1.0 (which is less than the target).

A. Additional results for constant channels on a single frequency

The above example raises two natural questions:

- What if we wish to optimize utility subject to slicing constraints (1) and (2) and subject to ISRF? Can this be achieved with SMSa or SMSm?

We address these questions for the case of constant, frequency-independent channel rates and the Proportional Fair utility function $U(r) = \log(r)$. We also assume that slices are non-overlapping (i.e. $\alpha_{i,j} \in \{0, 1\}$ for all $i, j$ and $\sum_j \alpha_{i,j} \in \{0, 1\}$ for all $i$).

Lemma 3: For constant, frequency-independent channel rates and non-overlapping slices, if we only have resource share constraints then there is an optimal solution that satisfies ISRF. Moreover, both SMSa and SMSm achieve this solution.

Proof: In the absence of rate constraints, the optimality of SMSa and the update rule for $Q_j$ implies there is an optimal solution with $Q_j(\tau) = 0$ for all $j, \tau$. Moreover, Lemma 1 implies that SMSa achieves this solution while also satisfying ISRF since $z_i = Z_j$ for all users $i$ in slice $j$. It remains to show that we can achieve the solution using SMSm.

In the absence of rate constraints, the SMSa metric becomes

$$C_i \frac{R_i}{R_i} + \delta \sum_{j \in J} \alpha_{i,j} Z_j.$$

Moreover, since ISRF holds at optimality, there exists a per-user resource share $\hat{X}_j$ for each slice $j$ such that $X_i = \sum_{j \in J} \alpha_{i,j} \hat{X}_j$. We can therefore write the SMSa metric as,

$$\frac{C_i}{R_i} \sum_{j \in J} \alpha_{i,j} \hat{X}_j + \delta \sum_{j \in J} \alpha_{i,j} Z_j$$

$$= \frac{1}{\sum_{j \in J} \alpha_{i,j} \hat{X}_j} + \delta \sum_{j \in J} \alpha_{i,j} Z_j$$

$$= \sum_{j \in J} \alpha_{i,j} \eta_j \frac{C_i}{R_i},$$

where $\eta_j = (1 + \frac{\delta Z_j}{\hat{X}_j}) = \log_b(1 + (\delta Z_j/\hat{X}_j))$. (Note that this rewriting depends critically on the fact that the slices are non-overlapping.) Let $Z_j = \log_b(1 + (\delta Z_j/\hat{X}_j))$. Note that $Z_j > 0$ if and only if $Z_j > 0, \hat{Z}_j = 0$ if and only if $Z_j = 0$ and $\hat{Z}_j < 0$ if and only if $Z_j < 0$. Moreover, the scheduler metric has the form $\sum_{i,j} \alpha_{i,j} Z_j C_i R_i$, which conforms to the specification of SMSm. Hence we can update $Z_j$ according to a rule similar to (4). If we apply SMSm with scheduler metric $b \sum_{i,j} \alpha_{i,j} \frac{Z_j C_i R_i}{R_i}$ then after convergence we can set $Z_j = \frac{b}{2} (\hat{Z}_j - 1)$ and obtain a solution that satisfies (8), (9), (10), thereby establishing optimality.

Lemma 4: For constant, frequency-independent channel rates and non-overlapping slices, we can maximize utility subject to rate and resource slice constraints, and subject to ISRF, using SMSm. We can also achieve this using a scheduling metric that has additive adjustments (but is not the same as SMSa).

Proof: Omitted due to space restrictions.
III. VARIABLE CHANNELS

In the case of variable channel rates, even formulating the problem of maximizing the aggregate throughput utility subject to slicing constraints (rate targets and resource constraints) as an explicit mathematical program like in (7) is already cumbersome. Consequently, it is difficult to state optimality properties of the SMSa and SMSm algorithms in such explicit terms, and it turns out to be more convenient to make optimality statements in implicit form with respect to the long-term capacity region. Also, in the case of variable channels, strict resource fairness does not even hold for the Proportional Fair algorithm in the absence of any slicing constraints, and strict ISRF needs to be relaxed to NIST, which provides a proxy for ISRF as mentioned earlier.

<table>
<thead>
<tr>
<th>SMSa</th>
<th>SMSm</th>
</tr>
</thead>
<tbody>
<tr>
<td>satisfies slicing constraints whenever feasible to do so at all</td>
<td>satisfies slicing constraints whenever feasible to do so while ensuring FA</td>
</tr>
<tr>
<td>achieves optimal utility subject to slicing constraints</td>
<td>achieves optimal utility subject to slicing constraints and FA in case of rate targets as well as in case of resource constraints for constant channels; may not be utility-optimal in case of resource constraints for arbitrary channel distributions</td>
</tr>
<tr>
<td>provides FA for resource constraints, but generally not for rate targets</td>
<td>provides FA always</td>
</tr>
</tbody>
</table>

TABLE I: A comparison of the properties of SMSa and SMSm.

With these two notions in place, the statements for variable channels mostly parallel those for constant channels, as summarized in Table I. The acronym FA in the table stands for “Fair Allocation”, which is to be interpreted as ISRF for constant channels and NIST for variable channels. The only minor distinction that arises is that for arbitrary channels SMSm may not produce a throughput vector that is utility-optimal over the restricted capacity region that is obtained when imposing NIST and slice resource constraints. This is a manifestation of the somewhat paradoxical fact that slice resource constraints can cause the utility-optimal throughput vector to lie in the interior of the full capacity region for extreme channel distributions.

IV. SIMULATION RESULTS

We now present our simulation results. The experiments are performed in a downlink system-level simulator which is 3GPP calibrated [2] and abstracts the physical-layer effects through a link to system-level interface. The interface employs an equivalent Signal-to-Interference-and-Noise Ratio (SINR), computed given the cell/user topology and active transmissions, with a vertically polarized antenna configuration.

We assume a single slice (that we call Mobile Group 1 - MG1). The users in this slice are dropped randomly in the scenario and then associated with the cell that has the best channel. The average number of MG1 users per cell is 6 but due to the random placement the actual number may change. We assume the Proportional Fair utility function but we will plot the Geometric Mean of Throughput (which equals \(\exp(utility)\)). Each cell has an aggregate rate target for MG1 users that is the same across all cells (but varies in different experiments). All other users are best effort users that we refer to as Mobile Group 2 - MG2. (There are on average 8 such users per cell.) These and other relevant simulation parameters are summarized in Table II. We do not impose any resource constraints since we wish to focus on the fact that rate constraints create a tradeoff between utility and fairness.

We run the scenario assuming MG1 has rate targets ranging from 1 Mbps to 20 Mbps. In Figure 1 we show the Geometric Mean of Throughput achieved by SMSa and SMSm for each target. In particular, in Figure 2 we show the distribution of rates achieved by MG1 (over all cells and all user drops) when the slice rate requirement is 6 Mbps and 15 Mbps respectively. Already with 15 Mbps, there are some cells/drops where the slice requirement cannot be satisfied due to poor channel conditions and this happens more for SMSm than for SMSa. In Figure 3 we show MG1’s average rate and resource share across all cells. Note that both SMSa and SMSm produce a straight rate line for slice 1, until a point where the scheduling algorithm cannot always guarantee the target. This indicates that both algorithms achieve on average the slice rate constraint, until a breakpoint that is lower for SMSm. Considering the allocated resources, we note that SMSm produces a straight line because it achieves ISRF and so the rate achieved is approximately proportional to the resources assigned. In contrast, SMSa adjusts the resources provided to MG1 in order to achieve a better geometric mean of throughput. For example, for high slice rate targets it uses

![Fig. 1: Geometric mean of throughput obtained by SMSa and SMSm for the full set of slice rate targets.](image_url)
fewer resources overall because it assigns more resources to the high-rate users within the slice (which illustrates that SMSa is not enforcing ISRF). The difference in ISRF enforcement between SMSa and SMSm is illustrated further in Figure 4. Given that ISRF means equality of resources within the slice, we define the Resource Unfairness Index (RUI) as

$$\text{RUI} = E_{\text{cell, scenario}} \left[ \text{abs} \left( \left( \prod_{i \in \text{MG1}} x_i \right)^{\frac{1}{|\text{MG1}|}} \sum_{i \in \text{MG1}} x_i - 1 \right) \right],$$

where $|\text{MG1}|$ is the number of MG1 users in the cell. Note that in (11) we compare the geometric mean of average resources $x_i$ experienced by $i$-th user with $|\text{MG1}|^{-1} \sum_{i \in \text{MG1}} x_i$, that is the geometric mean that would be obtained if all users get the resources of assigned to MG1 equally split. We see that SMSm maintains consistent low RUI (and hence high fairness) whereas SMSa has higher unfairness for both low and high slice rate targets. The contrast between Figure 1 (Geometric Mean of Throughput) and Figure 4 (Fairness) illustrates the basic tradeoff between these two quantities that is the main emphasis of this paper.

V. CONCLUSIONS

In this work we have studied additive and multiplicative adjustments to MAC scheduler weights that aim to satisfy slice rate and resource requirements. We have shown that additive adjustments maximize utility subject to these constraints but do not provide intra-slice fairness. Imposing intra-slice fairness shrinks the feasible capacity region and the resulting problem can be solved using multiplicative adjustments.

We have focused on a full-buffer model for which the most natural constraints are on slice rate and slice resource share. In future work we shall incorporate delay constraints for slices that have many finite-buffer users, possibly corresponding to Ultra-Reliable Low Latency Communication (URLLC) traffic.

REFERENCES