IEICE TRANSACTIONS

on Communications

DOI:10.1587/transcom.2018NVP0003

Publicized:2018/09/20

A PUBLICATION OF THE COMMUNICATIONS SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

RAN Slicing to Realize Resource Isolation Utilizing Ordinary Radio Resource Management for Network Slicing

Daisuke NOJIMA[†], ***Nonmember***, **Yuki KATSUMATA**[†], **Yoshifumi MORIHIRO**[†], **Takahiro ASAI**[†], **Akira YAMADA**[†], ***and*** **Shigeru IWASHINA**[†], ***Members***

SUMMARY In the context of resource isolation for network slicing, this paper introduces two resource allocation methods especially for the radio access network (RAN) part. Both methods can be implemented by slight modification of the ordinary packet scheduling algorithm such as the proportional fairness algorithm, and guarantee resource isolation by limiting the maximum number of resource blocks (RBs) allocated to each slice. Moreover, since both methods flexibly allocate RBs to the entire system bandwidth, there are cases in which the throughput performance is improved compared to when the system bandwidth is divided in a static manner, especially in a frequency selective channel environment. Numerical results show the superiority of these methods to dividing simply the system bandwidth in a static manner, and show the difference between the features of the methods in terms of the throughput performance of each slice.

key words: *5G, Network Slicing, Resource Isolation, Radio Resource Management*

1. Introduction

In order to launch a service of the fifth-generation (5G) mobile communication systems, the standardization work on New Radio (NR) has been completed in the 3rd Generation Mobile Partnership Project (3GPP), i.e., the specification of Release 15 [1]. The major scope of NR in Rel.15 is the enhanced mobile broadband (eMBB) some features of ultra-reliable low latency communication (URLLC) such as the maximum bit rate of 20 Gbps and the user plane latency of less than 0.5 msec in the radio access network (RAN) have been indicated as the final target values for 5G. 5G is anticipated to support a wide range of demands from application scenarios, e.g., mobile broadband, massive Internet of things (IoT), and mission-critical IoT, which require different types of features and networks in terms of mobility, charging, security, policy control, latency, reliability etc.

To address the various application demands, the next generation mobile networks (NGMN) recommends the concept of network slicing which establishes a service-based end-to-end dedicated virtual network by using two techniques, namely, slice leveraging the network functions virtualization (NFV) and software-defined networking (SDN) [2]–[5]. This network slice concept can be one of the key features of a 5G network; the resources of functional entities allocated to each slice are exclusive and isolated. In the Next Generation-Radio Access Network (NG-RAN) for NR connected to the 5G Core Network (5GC) in Rel.15 [1], network slicing is defined as follows: "*Network Slicing is a concept to allow differentiated treatment depending on each customer requirements. With slicing, it is possible for Mobile Network Operators (MNOs) to consider customers as belonging to different tenant types with each having different service requirements that govern in terms of what slice types each tenant is eligible to use based on Service Level Agreement (SLA) and subscriptions.*" In addition, for support of network slicing, it is also shown in [1] that the following key principles are applied in the NG-RAN.

- *RAN awareness of slices*
- *Selection of RAN part of the network slice*
- *Resource management between slices*
- *Support of Quality of Service (QoS)*
- *RAN selection of core network (CN) entity*
- *Resource isolation between slices*
- *Slice availability*
- *Support for UE associating with multiple network slices simultaneously*

As for resource isolation between slices, it is also described that resource isolation may be achieved by means of radio resource management (RRM) policies and protection mechanisms that should avoid a shortage of shared resources in one slice breaks the SLA for another slice. Also as described in the document, although a network slice consists of a RAN part and a core network part, the network in the RAN part can support differentiated handling of each slice through packet scheduling and different L1/L2 configurations. In this paper, we focus on packet scheduling, i.e., resource management in the RAN part, for the actualization of resource isolation.

As for resource management in the RAN part, several scheduling methods have been studied in [6]–[14]. In [6], the resource isolation between slices is guaranteed by allocating a certain minimum number of resource blocks (RBs) to each slice. More specifically, a mixed binary integer nonlinear programming problem for maximizing the sum rate subject to the constraints of the total power, the minimum number of RBs allocation, and proportional fairness is formulated, in which the iterative coordinate search and the suboptimal solution are shown. In this method, since RB allocation is conducted based on a service contract vector that characterizes resource isolation between slices, the normalized sum rate of each slice is almost proportional to the service

[†]Research Laboratories, NTT DOCOMO, INC. 3–6, Hikarinooka, Yokosuka-shi, Kanagawa, 239–8536 Japan.

contract vector. However, strict resource isolation cannot be guaranteed because the sum rate depends on the channel conditions of the users. In addition, the sum rate of a certain slice probably affects the sum rate of other slices. In [7], a throughput-maximum resource provisioning scheme that takes the average resource provision into account is proposed wherein the Lyapunov optimization is employed to tackle the difficulties of guaranteeing the average performance without the knowledge of future traffic arrival and wireless channel information. Furthermore, in [9], the resource allocation for a virtualized orthogonal frequency-division multiple access (OFDMA) uplink that optimizes the user transmission rates and quantization bit allocation for the compressed I/Q baseband signals is studied under the constraint of front-haul capacity and cloud computing resources. In [8], the optimization problem of multi-operator resource allocation is formulated, and a distributed semi-online algorithm for the problem is shown, since the problem is a nonlinear integer programming problem, leading to an NP-hard problem. In [11], traffic forecasting, admission control and scheduling for network slicing are discussed. In [12], a resource allocation based on the generalized mean of the system throughput is proposed that controls the tradeoff between the fairness and the spectrum efficiency. Most of the optimization problems formulated in the above-mentioned studies for which the sum rate is maximized under some constraints are hard to solve or are NP-hard, and therefore, the algorithms for which the problem is relaxed are discussed from the viewpoint of feasibility.

Meanwhile, hierarchical resource allocation techniques are proposed in [13], [14]. Specifically, in [13], a certain number of RBs is reserved for each slice in order to guarantee inter-slice isolation, while the remaining RBs are dynamically shared by all slices based on a hierarchical combinatorial auction model. In this method, in order to simplify the winner determination problem which is NP-hard, the channel gain of different RBs is assumed to be the same, and therefore, it is difficult for it to adapt to a frequency selective fading environment. In [14], a hierarchical resource allocation framework is proposed for mitigating inter-cell interference in extremely dense small cell networks. More specifically, the framework divides the problem into four steps, i.e., clustering, intra-cluster RB allocation, inter-cluster interference resolution, and power adjustment, in order to reduce the network complexity compared to that for centralized resource assignment approaches with an exhaustive search, whereas resource management for network slicing such as resource isolation is not considered.

On the other hand, in this paper, we introduce two resource allocation methods that can guarantee resource isolation and be easily implemented based on the conventional ordinary packet scheduling algorithm with a slight modification, thereby there is no need for a suboptimal algorithm for such as an integer programming problem which is NP-hard. Especially with respect to resource isolation, according to the viewpoint in [1] indicating that the shortage of shared resources in one slice should not break the SLA for another slice, we use the maximum number of allocated RBs in each slice as the constraint of the resource isolation instead of the typical key performance indicator (KPI) of the SLA such as throughput and latency. This is because the throughput and latency performance in a radio link heavily depend on the traffic load and channel conditions, i.e., time-varying fast fading channel, and therefore, the guarantee of throughput or latency performance in a practical manner is difficult to implement.

Aside from this, since both of the introduced resource allocation methods dynamically allocate RBs throughout the entire system bandwidth according to the channel conditions of the users in each slice, a diversity effect can be expected in a frequency selective channel. Furthermore, the first method takes into consideration priority among slices in an example scenario in which there are several services with different requirements such as eMBB, URLLC, and mMTC, whereas fairness among slices is taken into account in the second method in an example scenario in which there are several slices with the same priority.

The rest of the paper is organized as follows. Section 2 outlines the scheduling methods that guarantee the resource isolation. Section 3 describes the evaluation assumptions such as the traffic model, followed by numerical results. Finally, we conclude this paper in Section 4.

2. Packet Scheduling with Resource Isolation

In this section, the conventional packet scheduling algorithm based on proportional fair (PF) criteria, and the modified methods that guarantee the resource isolation are described. Resource isolation in the modified methods is guaranteed by limiting the maximum number of RBs that are predefined according to a condition such as the required SLA and QoS.

2.1 Conventional Method

Let us consider the system model where the set of total users is defined as $\mathcal{K} = \{0 \leqq k \leqq K-1\}$ and the set of RBs used in a evolved NodeB (eNB) is defined as $\mathcal{F} = \{0 \leqq f \leqq F-1\}$. When the f^{th} RB index is assigned to the k^{th} user, the PF metric, $p_{k,f}(t)$, is derived as below [15], where $r_{k,f}(t)$ is the instantaneous throughput and $\bar{r}_{k,f}(t)$ is the average throughput.

$$p_{k,f}(t) = \frac{r_{k,f}(t)}{\bar{r}_k(t)} \tag{1}$$

$$\bar{r}_k(t) = \left[1 - \frac{1}{T_{avg}}\right] \bar{r}_k(t-1) + \frac{1}{T_{avg}} b_{k,f}(t) r_{k,f}(t-1) \tag{2}$$

Term T_{avg} is the throughput averaging time and $b_{k,f}(t)$ is a variable indicating whether or not the k^{th} user is assigned the f^{th} RB index at time t.

$$b_{k,f}(t) = \begin{cases} 1, & \text{if the } f^{\text{th}} \text{ RB index is assigned to} \\ & \text{the } k^{\text{th}} \text{ user at time } t \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The conventional algorithm itself cannot guarantee resource isolation, since the RBs are allocated only based on the channel conditions of each user.

2.2 Static Allocation

In this allocation, the RB index that can be allocated to each slice is preliminarily determined to guarantee the resource isolation, and then, the ordinary packet scheduling algorithm such as PF, Max C/I, or round robin (RR) is employed in each slice among the preliminarily determined RBs. Fig.1 shows an allocation example of contiguous RBs for three slices. In this allocation, since RBs are allocated in a fixed manner regardless of the channel conditions of the users in each slice, the diversity effect owing to frequency selective fading might be decreased due to the fact that the bandwidth available for each slice is equivalently reduced as compared to the total system bandwidth, as shown in Fig.1.

Note that, even in this case, by allocating a RB index in a scattered manner, i.e., non-contiguous RB allocation, some diversity effect can be obtained. This method is easy to implement and achieves firm resource isolation even when there is a heavy traffic demand concentrated on a certain slice. However, when traffic at a certain slice is sporadic, spectrum efficiency is degraded because the maximum number of RBs allocated to each slice is fixed.

The algorithm of the static allocation is summarized in Algorithm 1. Here, we define the set of RBs used in slice m as $\mathcal{F}_m = \{0 \leqq f_m \leqq F_m - 1\}$, and the set of users belonging to slice m as $\mathcal{K}_m = \{0 \leqq k_m \leqq K_m - 1\}$. Since there are assumed to be M slices, the set of total users can be expressed as $\{\mathcal{K}_0, \mathcal{K}_1, \ldots \mathcal{K}_m, \ldots, \mathcal{K}_{M-1}\} \in \mathcal{K}$ and the set of RBs used in the eNB can be expressed as $\{\mathcal{F}_0, \mathcal{F}_1, \ldots \mathcal{F}_m, \ldots, \mathcal{F}_{M-1}\} \in \mathcal{F}$.

Fig. 1: Static allocation (3 slices)

Algorithm 1 Static Allocation

```
Set the available RB group for each slice, F_m
for m = 0 to M − 1 do
    for all f_m = 0 to F_m − 1 do
        for all k_m = 0 to K_m − 1 do
            Calculate p_{k_m,f_m}(t)
        end for
        k_m* = arg max (p_{k_m,f_m}(t))
        Assign the RB index f_m to user k_m*
        K_m ← K_m − k_m*
        F_m ← F_m − f_m
    end for
end for
```

2.3 Allocation to Ordered Slices

In this allocation, RB allocation to each slice is performed successively throughout the entire system bandwidth according to the channel conditions of the users in each slice. Fig.2 shows an example of this allocation with three slices.

At first, RBs are allocated to all users of Slice #0 according to the scheduling metric such as the PF index in the entire system bandwidth under the constraint of the maximum number of RBs for Slice #0. Then, among the remaining RBs, RB allocation to Slice #1 is executed under the constraint of the maximum number of RBs for Slice #1. RB allocation to Slice #2 is conducted in the same manner within the remaining RBs.

In this method, Slice #0 is given the highest priority. In other words, the ordering of the priority descends according to the slice index. Therefore, this RB allocation method is appropriate for a scenario in which a relatively different degree of priority is provided to each slice. The algorithm of this allocation is summarized in Algorithm 2.

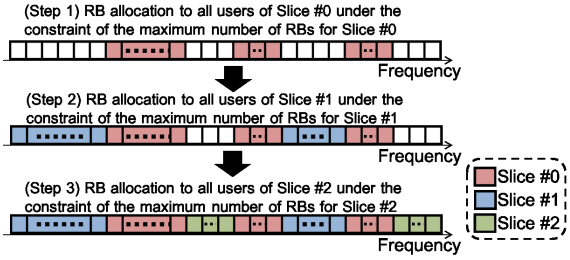


Fig. 2: Allocation to ordered slices (3 slices)

Algorithm 2 Allocation to Ordered Slices

```
Set the maximum number of RBs for each slice, F_m
initialize F_m ← 0
for all m = 0 to M − 1 do
    for F do
        for all k_m = 0 to K_m − 1 do
            Calculate p_{k_m,f}(t)
        end for
        k_m* = arg max (p_{k_m,f}(t))
        Assign the RB index f to user k_m*
        K_m ← K_m − k_m*
        F ← F − f ; F_m ← F_m + f
        if |F_m| = F_m then
            break
        end if
    end for
end for
```

2.4 Impartial Allocation to Slices

Fig. 3 shows an example of this allocation to three slices, in which RBs are allocated on a user basis that has a higher scheduling metric for each slice, according to the channel conditions. More specifically, in the first step, a certain number of RBs are allocated to a user who has the largest scheduling metric such as the PF index in Slice #0. In the

second step, among the remaining RBs, a certain number of RBs are allocated to a user who has the largest scheduling metric in Slice #1. Similarly in the third step, RBs are allocated to a user who has the largest scheduling metric in Slice #2 among the remaining RBs. Then, in the next step, RBs are allocated to a user who has the second largest scheduling metric in Slice #0. This allocation is repeated until the maximum number of allocated RBs is reached for each slice. With regard to the number of RBs to be allocated in each step, by allocating a single RB in each step, the finest RB allocation according to the channel conditions can be conducted. On the other hand, by allocating several contiguous RBs such as a RB group in LTE [16], the scheduling complexity is mitigated. In this method, RB allocation to each slice is fairly conducted to some degree. Therefore, this method suits a scenario in which fairness in terms of resource allocation to each slice should be taken into account. The algorithm in which a single RB is allocated in each step is summarized in Algorithm 3.

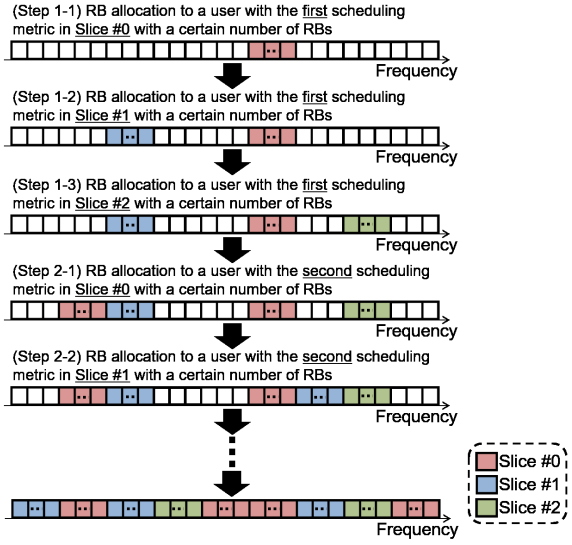


Fig. 3: Impartial allocation to slices (3 slices)

Algorithm 3 Impartial Allocation to Slices

Set the maximum number of RBs for each slice, F_m
initialize $\mathcal{F}_m \leftarrow 0$
for $\mathcal{F}$ **do**
  for all $m = 0$ to $M - 1$ **do**
    if $|\mathcal{F}_m| = F_m$ **then**
      break
    end if
    for $\mathcal{K}_m$ **do**
      Calculate $p_{k_m,f}(t)$
    end for
    ${k_m}^* = \arg\max\left(p_{k_m,f}(t)\right)$
    Assign the RB index f to user ${k_m}^*$
    $\mathcal{K}_m \leftarrow \mathcal{K}_m - {k_m}^*$
    $\mathcal{F} \leftarrow \mathcal{F} - f$; $\mathcal{F}_m \leftarrow \mathcal{F}_m + f$
  end for
end for

3. Evaluation of System Performance

In this section, evaluation assumptions are first described, and then, the throughput performance in the *Static Allocation* with the number of slices as a parameter is described as a preliminary evaluation for a full buffer traffic model. Also, comparison among the three RB allocation methods with 5 slices is performed in terms of throughput performance. After that, the performance levels of resource isolation, system throughput, and the flow time for various application traffic models are described. Note that, the flow time is defined as the time between traffic arrival at the eNB and the receiving completion at the UE in the downlink.

3.1 Evaluation of Full Buffer Traffic Model

Tables 1 and 2 give the major simulation parameters for full buffer traffic in the system level simulation. The parameters are based on the simulation model in the LTE [17], which corresponds to 100 RBs in total.

The number of users per sector is 60, and the number of users in each slice is assumed to be the same, i.e., 60 / (the number of slices). Furthermore, the maximum number of RBs in each slice for resource isolation is assumed to be the same, i.e., 100 / (the number of slices). With regard to the number of RBs allocated in each step, it is assumed that a single RB is allocated in each step for *Impartial Allocation to Slices* . Similarly, for *Allocation to Ordered Slices*, a single RB is assumed for RB allocation as the minimum unit for RB allocation. On the other hand, as for feedback information, a wideband precoding matrix indicator (PMI) and subband channel quality indicator (CQI) with the size of 2 RBs are assumed.

Table 1: Parameters for system level simulation

System bandwidth	20 MHz (100 RBs)
Cellular layout	Hexagonal grid, 7 sites, 3 sectors per site
Inter-site distance	500 m
eNB antenna pattern	Horizontal: 70-degree beam width Vertical: 10-degree beam width/ 15-degree down-tilt
Antenna configuration	2 × 2 MIMO
Hybrid Automatic Repeat Request (HARQ)	Chase combining (8 ms Round Trip Delay)
Retransmission limit	4 (HARQ)
Modulation	QPSK, 16QAM, 64QAM
Multipath delay profile	6 ray typical urban
UE velocity	3 km/h
Shadowing	Lognormal shadowing with standard deviation of 8 dB and inter-site correlation of 0.5

Table 2: Simulation parameters for full buffer traffic

Parameters	Value
Traffic model	Full buffer
Number of slices	2, 3, 4, 5
Number of users	60 / (number of slices)
Maximum number of RBs for each slice [RBs]	100 / (number of slices)

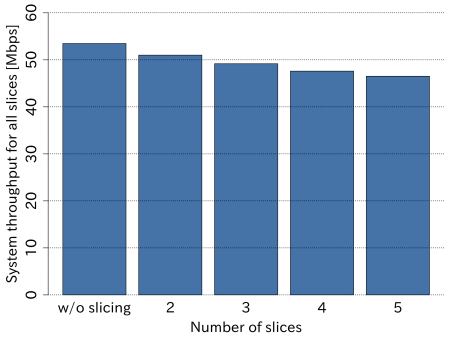


Fig. 4: System throughput for all slices in static allocation

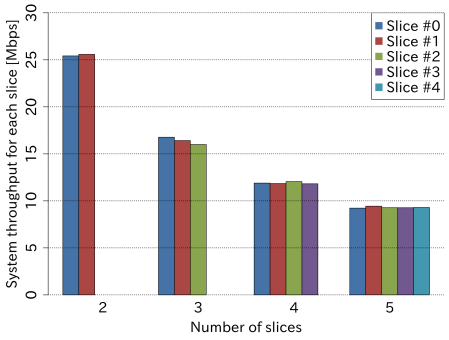


Fig. 5: System throughput for each slice in static allocation

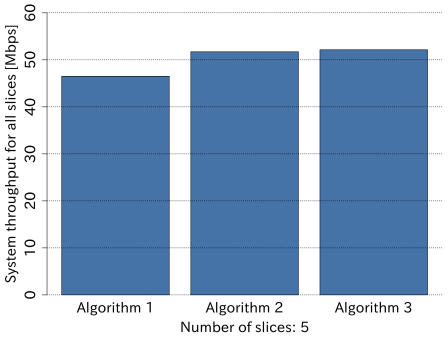


Fig. 6: Throughput comparison of all slices

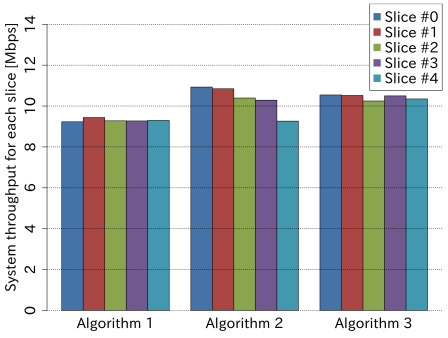


Fig. 7: Throughput comparison for each slice

3.1.1 Throughput in Static Allocation

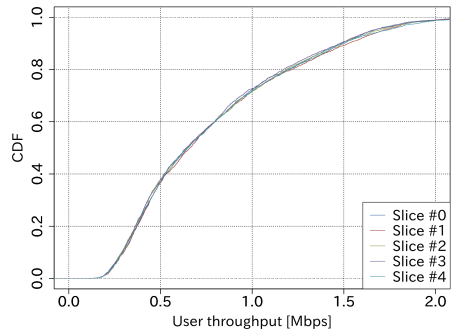
Fig. 4 shows the system throughput of all slices in the *Static Allocation*. The number of slices is 2, 3, 4, and 5, in which a contiguous bandwidth is allocated. More specifically, contiguous 50 RBs, 34 RBs (or 33 for Slice #1 and Slice #2), 25 RBs, and 20 RBs are allocated to each slice for the 2, 3, 4, and 5 slices, respectively. In the same manner, 30, 20, 15, and 12 users/sector are assumed to be allocated to each slice for the 2, 3, 4, and 5 slices, respectively. In addition, system throughput without slicing that utilizes all RBs with 60 users is also shown, for comparison. As shown in Fig. 4, compared to that without slicing, the system throughput of all slices is degraded as the number of slices increases. This is because the frequency diversity effect is decreased, since the available bandwidth of each slice is decreased with an increase in the number of slices. Another reason for this is that the multiuser diversity effect is decreased, since the number of users in each slice is decreased with an increase in the number of slices. Fig. 5 shows the system throughput of each slice. Since the total system bandwidth is equally divided and allocated to each slice, almost the same throughput is obtained for each slice despite the number of slices.

3.1.2 Comparisons Among RB Allocation Methods

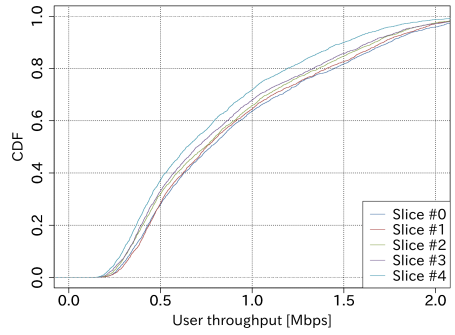
Fig. 6 shows the system throughput of all slices for the *Static Allocation* (Algorithm 1), the *Allocation to Ordered Slices* (Algorithm 2), and the *Impartial Allocation to Slices* (Algorithm 3). In this evaluation, the number of slices is also assumed to be 5. We can see that the system throughput of all slices in Algorithms 2 and 3 is improved, compared to that in Algorithm 1. This is because both methods allocate RBs throughout the entire system bandwidth, and therefore, a further frequency diversity effect can be obtained compared to that for Algorithm 1. In addition, the figure shows that the system throughput performance for Algorithm 3 is slightly improved compared to that for Algorithm 2. This is because, although both methods apply the same maximum number of RBs to each slice for resource isolation and have the same number of users in each slice, Algorithm 3 can attain a greater multiuser diversity effect due to the fact that RBs are allocated with priority to the users having a higher scheduling metric among the total 60 users, as compared to Algorithm 2 in which RBs are successively allocated to users that have a higher scheduling metric among the 12 users in each slice.

Figs. 7 and 8 show the system throughput performance and the cumulative distribution function (CDF) of the user throughput of each slice for the three methods, respectively. As shown in Fig. 7, since the RB allocation is successively executed in order of the slice index, the throughput in Algorithm 2 is decreased as the slice index increases, as mentioned in Section 2.3. However, in Algorithm 3, similar throughput performance levels are obtained despite the slice index, as discussed in Section 2.4. Similarly, as shown in Figs. 8 (b) and 8 (c), while the CDF of the user throughput in Algorithm 3 is almost at the same level despite the slice index, that in Algorithm 2 is degraded with an increase in the slice index.

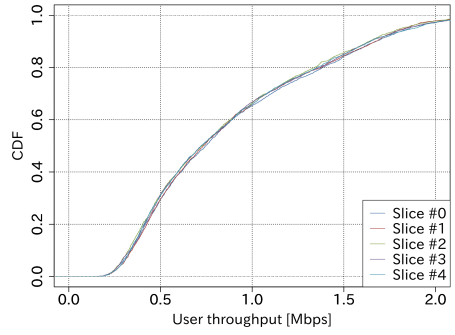
When comparing the slices for Algorithm 2 to those for



(a). *Static Allocation* (Algorithm 1)



(b). *Allocation to Ordered Slices* (Algorithm 2)



(c). *Impartial Allocation to Slices* (Algorithm 3)

Fig. 8: CDF of user throughput

Algorithm 1, we find that the throughput in the former is improved compared to that in the latter, since a greater frequency diversity effect can be obtained by enabling RB allocation to the entire system bandwidth. In the same manner, the CDF of the user throughput in Algorithm 3 is improved compared to that in Algorithm 1.

3.2 Evaluation for Various Application Traffic Models

Tables 1 and 3 give the major simulation parameters for various application traffic in the system level simulation. With regard to the number of users, we assume two scenarios that have different resource utilization, such as scenario (a) with 75 users per sector (25 users in Slice #0, 10 users in Slice #1, 40 users in Slice #2) and scenario (b) with 265 users per sector (90 users in Slice #0, 30 users in Slice #1, 145 users in Slice #2). Furthermore, the maximum number of RBs in each slice for resource isolation is assumed to be the 5, 80, and 15 RBs for Slice #0, Slice #1, and Slice #2, respectively. Table 3 gives the simulation parameters for the traffic model of each slice, in which URLLC, eMBB and mMTC services are assumed for Slice #0, Slice #1, and Slice #2, respectively. The parameters of those traffic models are decided by referring to [18]. In this evaluation, although generally speaking the priority of URLLC, eMBB, and MTC is different especially in terms of latency requirement, Algorithm 3 (*Impartial Allocation to Slices*) is also evaluated for performance comparison hereafter.

Table 3: Simulation parameters for various application traffic

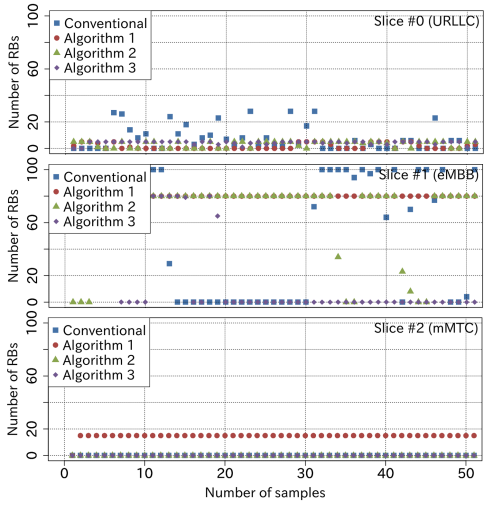
Parameters		URLLC (Slice #0)	eMBB (Slice #1)	mMTC (Slice #2)
Traffic model		Poisson		
Interval for traffic generation [s]		0.1	5	20
Size of generated traffic [kByte]		0.2	500	75
Number of users	Scenario (a)	25	10	40
	Scenario (b)	90	30	145
Maximum number of RBs [RBs]		5	80	15

3.2.1 Performance of Resource Isolation

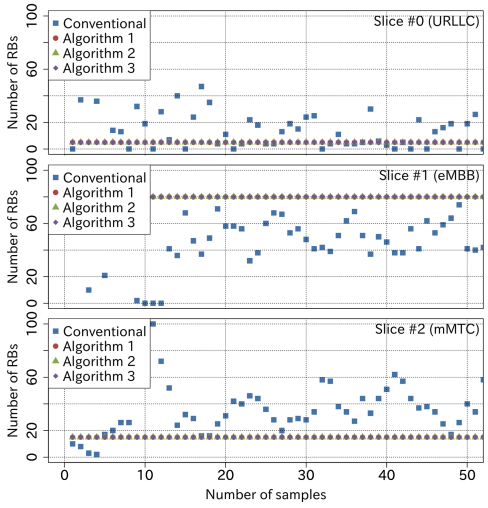
Table 4 shows the average number of allocated RBs in scenarios (a) and (b) of the conventional method described in Section 2.1, Algorithms 1, 2, and 3. It should be noted here that the average number of allocated RBs in Table 4 shows the number of allocated RBs in each slice averaged by all the transmission time intervals (TTIs) in the system level simulation even when there is no traffic to be transmitted.

As shown in Table 4, since the number of users assumed in scenario (a) is less than that in scenario (b), the average number of allocated RBs in scenario (a) is less than that in scenario (b) for all slices. In addition, the average number of allocated RBs in each slice does not exceed the predetermined maximum number of RBs in Table 3, i.e., 5 (Slice #0), 80 (Slice #1), and 15 (Slice #2), even in the conventional method for scenario (a), whereas that in Slices #0 and #2 exceeds the predetermined maximum number of RBs in the conventional method for scenario (b). Figs. 9 (a) and 9 (b) show an example of the allocated RBs in scenarios (a) and (b) for the conventional method, Algorithm 1, Algorithm 2, and Algorithm 3. Also as shown in the figure, the number of allocated RBs in the conventional method occasionally exceeds the predetermined maximum number of RBs for Slices #0 and #1 in scenario (a) and Slices #0 and #2 in scenario (b) in this example. Figs. 10 (a) and 10 (b) show the CDF of the number of allocated RBs in scenarios (a) and (b). Note that, Fig. 10 shows the CDF of the number of allocated RBs only when there is traffic to be transmitted in each slice. The figure shows that the number of allocated RBs does not exceed the predetermined maximum number of RBs for Algorithms 1, 2, and 3 in both scenarios (a) and (b). On the other hand, the number of allocated RBs in the conventional method exceeds the predetermined maximum number of RBs with more than 50 % (Slice #0), 90 % (Slice

#1), and 95 % (Slice #2) of the RB allocation, respectively in scenario (a). Similarly, that with more than 80 % (Slice #0), 35 % (Slice #1), and 70 % (Slice #2) exceed the predetermined maximum number of RBs, respectively, in scenario (b). From these results, we discern that the conventional method does not guarantee resource isolation in terms of the maximum number of RBs, but Algorithms 1, 2, and 3 guarantee the resource isolation between slices.



(a). Scenario (a)



(b). Scenario (b)

Fig. 9: Example of allocated RBs in the evaluation

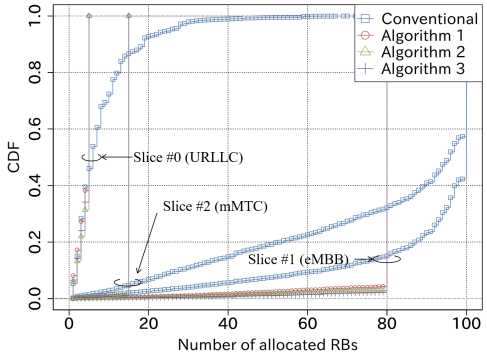
3.2.2 Comparison of Throughput Performance

Fig. 11 shows the system throughput in scenario (a) that corresponds to the total of 75 users per sector. Since the average resource utilization is approximately 32 %, this scenario is regarded as a relatively low resource utilization environment. As shown in Fig. 11, we see that the throughput performance levels especially of Algorithms 1 and 2 are slightly degraded

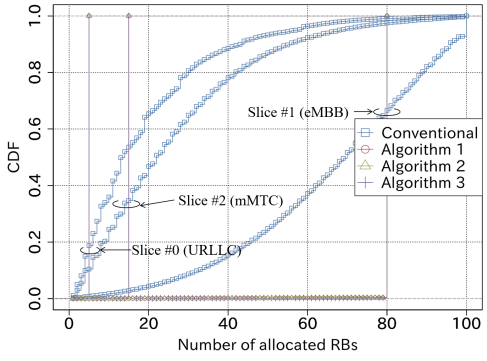
Table 4: Average number of allocated RBs

Scenario (a)	URLLC (Slice #0)	eMBB (Slice #1)	mMTC (Slice #2)
Conventional [RBs]	2.47	26.13	5.09
Algorithm 1 [RBs]	1.98	23.90	4.08
Algorithm 2 [RBs]	2.02	23.88	4.28
Algorithm 3 [RBs]	2.03	25.40	3.97

Scenario (b)	URLLC (Slice #0)	eMBB (Slice #1)	mMTC (Slice #2)
Conventional [RBs]	12.93	64.15	21.90
Algorithm 1 [RBs]	5.00	71.17	14.67
Algorithm 2 [RBs]	5.00	71.49	14.56
Algorithm 3 [RBs]	5.00	72.17	14.20



(a). Scenario (a)



(b). Scenario (b)

Fig. 10: CDF of number of allocated RBs

compared to that for the conventional method. These algorithms cannot allocate RBs to sporadically concentrated traffic by limiting the maximum number of RBs for resource isolation even with the resource utilization of 32 %. This fact can be also observed from the results in Table 4 in which the average number of allocated RBs of Slice #1 in the conventional method is slightly larger than that of Algorithms 1 and 2. In addition, Fig. 11 shows that the system throughput performance of Slice #1 in Algorithm 3 is at almost the same level compared to the conventional method. This is because Algorithm 3 can attain a greater multiuser diversity effect due to the fact that RBs are allocated with priority to the users having a higher scheduling metric among the total 75

users as compared to the Algorithms 1 and 2.

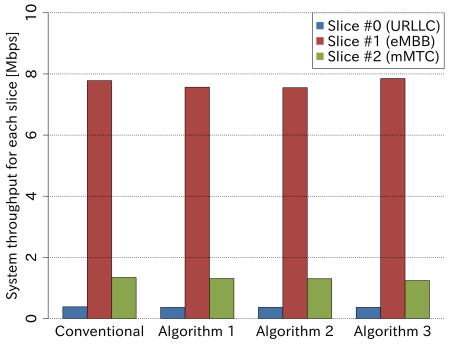


Fig. 11: System throughput for each slice in scenario (a)

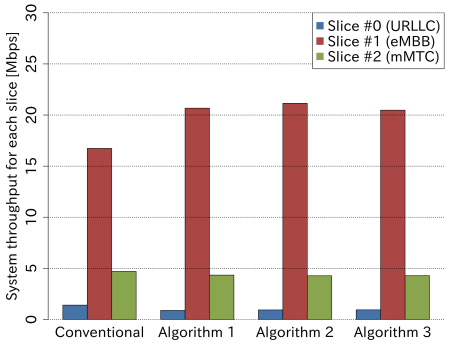
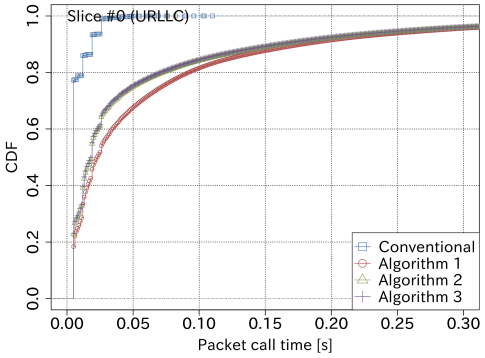
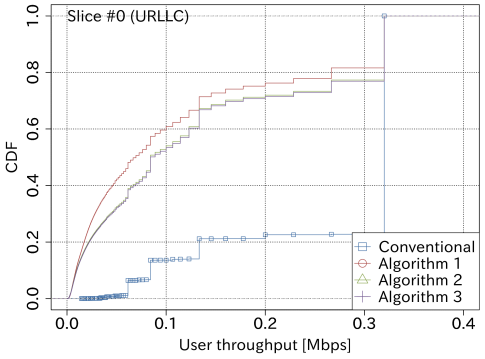


Fig. 12: System throughput for each slice in scenario (b)



(a). CDF of flow time



(b). CDF of user throughput

Fig. 13: CDF of flow time and user throughput for Slice #0 (URLLC) in scenario (b)

Fig. 12 and Table 4 show the system throughput in scenario (b) that corresponds to the total of 265 users per sector. The average resource utilization is approximately 95 %, and therefore, this scenario is regarded as a high resource utilization environment. As shown in Fig. 12 and Table 4, the system throughput performance levels of Algorithms 1, 2, and 3 are improved as compared to that for the conventional method. This is because Algorithms 1, 2, and 3 allocate more RBs to Slice #1 as indicated in Table 4 due to the fact that Algorithms 1, 2, and 3 limit the maximum number of RBs for each slice. On the other hand, the conventional method allocates more RBs to Slice #0 and Slice #2 simply using the PF metric.

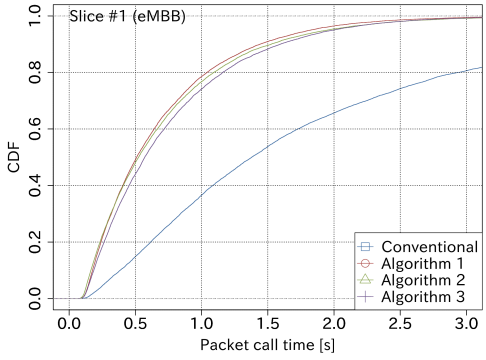
Furthermore, the system throughput of Algorithm 2 especially for Slice #1 is greater than that for Algorithm 1. This is because Algorithm 2 can obtain a greater diversity gain in a frequency selective channel by allocating RBs to the entire system bandwidth. However, since Algorithm 3 fairly allocates RBs to each slice, the system throughput of Algorithm 3 for Slice #1 is degraded. Figs. 13, 14, and 15 show the CDF of the flow time and user throughput for Slices #0, #1, and #2 in scenario (b), respectively. The results of Fig. 13 (a) and 13 (b), show that the flow time and user throughput of Slice #0 with Algorithms 2 and 3 are improved in comparison to those of Algorithm 1. This is because Algorithms 2 and 3 obtain a further frequency diversity gain due to the fact that both methods allocate RBs among the entire system bandwidth. It should be noted here that the flow time and user throughput of Algorithms 1, 2, and 3 are degraded compared to that for the conventional method. This is because the conventional method allocates more RBs than Algorithms 1, 2, and 3 that restricts the predetermined maximum number of RBs for resource isolation, as shown in Fig. 10 (b). In addition, since the average throughput of Slice #0 is relatively low compared to that for Slices #1 and #2, the PF metric of Slice #0 in Eq. (1) becomes relatively higher than that for Slices #1 and #2. As a result, more RBs are allocated to Slice #0 than Slices #1 and #2 in the conventional method. Consequently, the flow time and user throughput of Slice #0 are improved compared to those for Slices #1 and #2 in the conventional method, but resource isolation is not guaranteed.

From the results of Fig. 14, the flow time and user throughput of Algorithms 1, 2, and 3 for Slice #1 are improved compared to that for the conventional method due to the fact that more RBs are allocated to Slice #1 on average as indicated in Table 4 and Fig. 10 (b). This is because the PF metrics of Slices #0 and #2 are relatively higher than that for Slice #1 as described in the previous paragraph. As a result, the conventional method tends to allocate more RBs to Slices #0 and #2 than Slice #1 as indicated in Table 4. Contrarily, Algorithms 1, 2, and 3 can allocate more RBs to Slice #1 because those algorithms restrict the maximum number of allocated RBs to Slices #0 and #2. By comparing the results of Fig. 13 and 14, we discern that the flow time

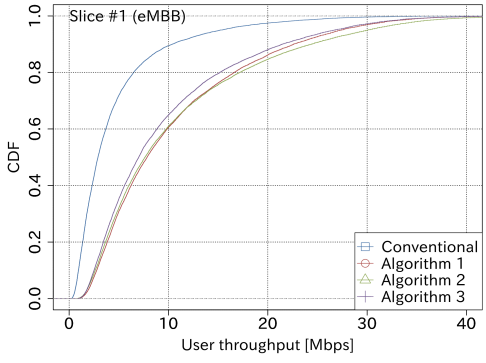
of Slice #0 is lower than that of Slice #1. Also from the results, we see that the proposed method is effective even for such a low latency slice, since the flow time is a critical performance metric for Slice #0.

With regard to Fig. 15, although the flow time and user throughput of Algorithms 1, 2, and 3 are degraded compared to that for the conventional method due to the same reasons for Fig. 13, the user throughput of Algorithm 3 is better than that for Algorithm 2, since the RB allocation to each slice is fairly conducted.

Finally, we conducted a performance comparison between the proposed method and the existing QoS-aware scheduling as described in the Appendix. The results show that the QoS-aware scheduling adjusts the typical KPI of the SLA such as the latency performance and throughput performance, but it might be difficult to guarantee resource isolation in a strict sense especially with regard to the number of allocated RBs.



(a). CDF of flow time

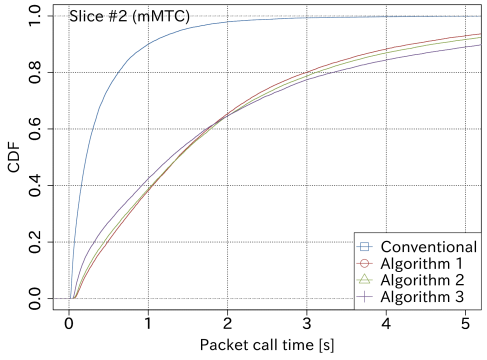


(b). CDF of user throughput

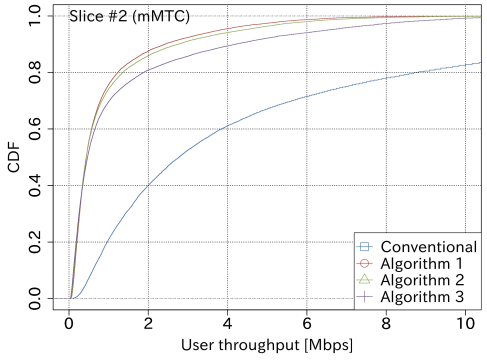
Fig. 14: CDF of flow time and user throughput for Slice #1 (eMBB) in scenario (b)

4. Conclusions

In this paper, we have presented resource isolation methods in the RAN part for network slicing. These methods are slight modifications of the ordinary packet scheduling algorithm, and guarantee resource isolation by limiting the



(a). CDF of flow time



(b). CDF of user throughput

Fig. 15: CDF of flow time and user throughput for Slice #2 (mMTC) in scenario (b)

maximum number of allocated RBs to each slice. In particular, the first method allocates RBs to each slice successively, while the second one allocates RBs to users having a higher scheduling metric for each slice. The system level simulation results confirmed that performance levels of the three methods, *Static Allocation*, *Allocation to Ordered Slices*, and *Impartial Allocation to Slices*, were improved especially in a high resource utilization environment. Furthermore, we observed that the *Allocation to Ordered Slices* obtained a greater diversity gain in a frequency selective channel.

Appendix A: Performance Comparison with QoS-aware Scheduling

For the purpose of performance comparison between the proposed methods and existing QoS-aware scheduling, the following weighted PF metric is applied to the conventional method instead of Eq. (1), as a simple QoS-aware scheduling.

$$p_{k,f}(t) = \alpha_m \frac{r_{k,f}(t)}{\overline{r}_k(t)}$$
$$\text{where} \sum_{m=0}^{M-1} \alpha_m = 1, \qquad (\text{A}\cdot 1)$$

where α_m is a weight parameter for service (slice) #m. With regard to the QoS, taking into consideration the latency re-

quirement, the following four cases of parameters (Cases 1-4 in Table A· 1) are used in the evaluation for Slice #0 (URLLC), Slice #1 (eMBB), and Slice #2 (mMTC). By applying Case 1 or 2, the latency performance of Slice #0 is expected to improve since RBs tend to be allocated with higher priority to Slice #0 compared to Slices #1 and #2. On the other hand, by applying Case 3 or 4, the throughput performance of Slice #1 is expected to improve since RBs tend to be allocated with higher priority to Slice #1 compared to Slices #0 and #2. However, in any case, using only QoS-aware scheduling to achieve resource isolation might be difficult since a shortage in RBs for another slice occurs when more RBs are allocated to one slice by taking into consideration the QoS of only the corresponding slice.

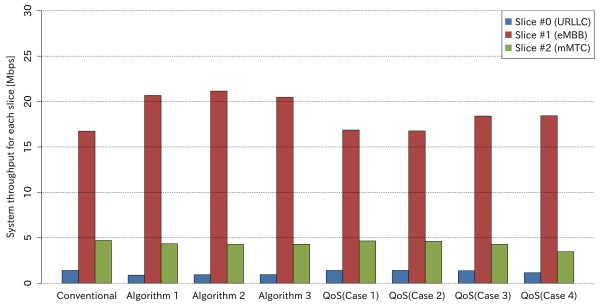


Fig. A· 1: System throughput of QoS-aware scheduling in scenario (b)

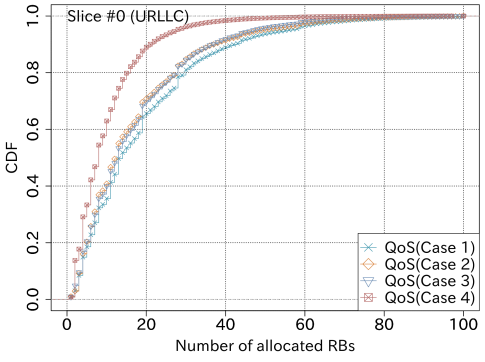
Table A· 1: Weight parameters for QoS-aware scheduling

	α_0 (Slice #0: URLLC)	α_1 (Slice #1: eMBB)	α_2 (Slice #2: mMTC)
Case 1	0.50	0.30	0.20
Case 2	0.70	0.20	0.10
Case 3	0.01	0.90	0.09
Case 4	0.001	0.99	0.009

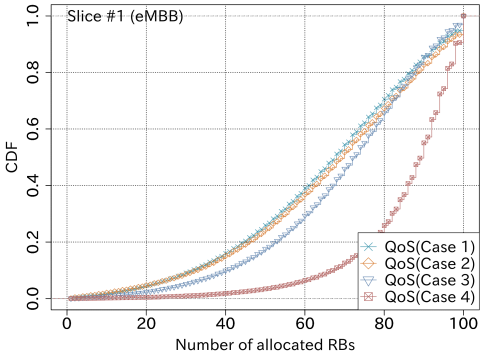
Table A· 2: Average number of allocated RBs of QoS-aware sheduling

Scenario (b)	URLLC (Slice #0)	eMBB (Slice #1)	mMTC (Slice #2)
Case 1 [RBs]	12.74	63.75	21.90
Case 2 [RBs]	12.89	63.70	22.05
Case 3 [RBs]	12.03	65.33	20.05
Case 4 [RBs]	7.81	76.61	13.49

Fig. A· 1 shows the system throughput for each slice in scenario (b) with the conventional scheduling method described in Section 2.1 (Conventional), QoS-aware scheduling method (QoS scheduling), and proposed scheduling methods (Algorithms 1, 2, and 3). Table A· 2 gives the average number of allocated RBs. Fig. A· 1, Table 4, and Table A· 2 show that the performance levels of QoS-aware scheduling for Cases 1 and 2 are almost the same as that for the conventional method. This is because, although the weighted PF metric based on Table A· 1 is used for considering the latency performance, more RBs tend to be allocated to Slice #0 in the original conventional method as discussed in Section 3.2.2, regardless of whether or not the weight parameter α_m is applied. The CDF performance levels of the number of allocated RBs for Slice #0 with Cases 1 and 2 in Fig. A· 2 are also almost the same as those in Fig. 10. In addition, the CDF performance levels of the flow time and user throughput with Cases 1 and 2 in Figs. A· 3 (a) and A· 3 (b) are similar to those in Figs. 13 (a) and 13 (b).



(a). Slice #0 (URLLC)

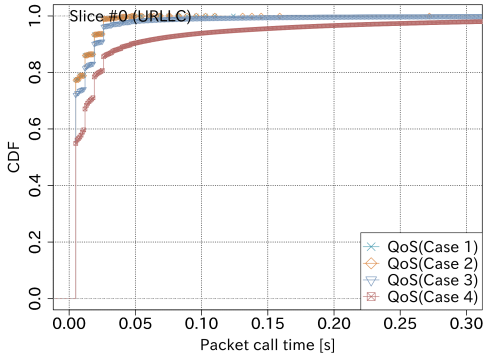


(b). Slice #1 (eMBB)

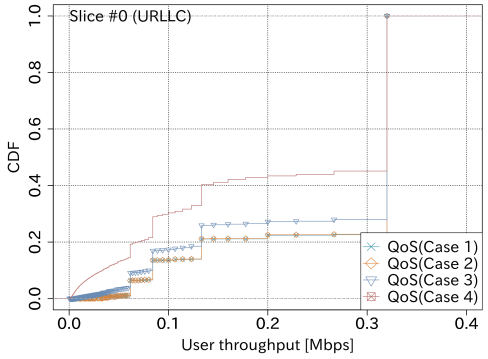
Fig. A· 2: CDF of number of allocated RBs of QoS-aware scheduling

With regard to Cases 3 and 4, the system throughput in Fig. A· 1 and the average number of allocated RBs in Table A· 2 for Slice #1 become larger than that for the conventional method in Fig. A· 1 and Table 4. This is because a larger weight parameter is applied to Slice #1 than Slices #0 and #2. Accordingly, the number of allocated RBs for Slice #1 with Cases 3 and 4 is increased compared to that with Cases 1 and 2 as shown in Fig. A· 2. Coupled with this, the flow time and user throughput with Cases 3 and 4 in Fig. A· 4 are improved compared to those with the conventional method in Fig. 14, at a sacrifice in performance for Slice #0 shown in Fig. A· 3.

Based on these results, we can see that QoS-aware

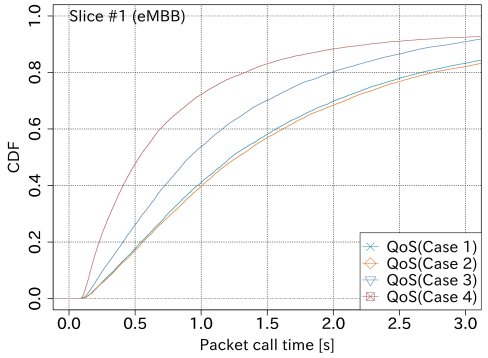


(a). CDF of flow time

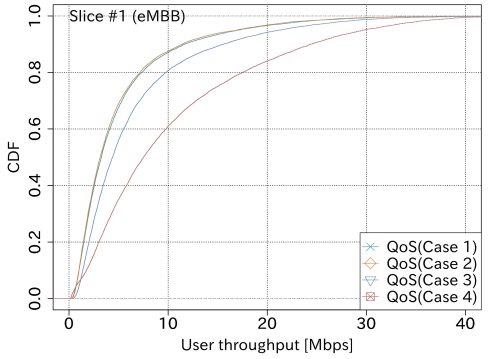


(b). CDF of user throughput

Fig. A· 3: CDF of flow time and user throughput for Slice #0 (URLLC) in scenario (b)



(a). CDF of flow time



(b). CDF of user throughput

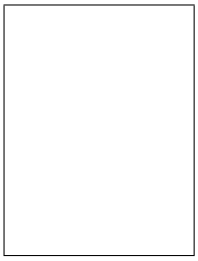
Fig. A· 4: CDF of flow time and user throughput for Slice #1 (eMBB) in scenario (b)

scheduling can adjust the typical KPI of the SLA such as the throughput performance, but the results of Fig. A· 2 show that it is difficult for QoS-aware scheduling to enforce a strict limitation on the number of allocated RBs for each slice. Consequently, it might be difficult for simple QoS-aware scheduling to guarantee resource isolation in a strict sense.

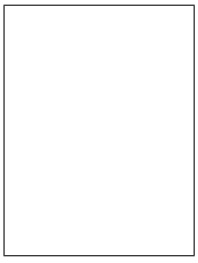
References

[1] 3GPP TS 38.300 V15.2.0, "NR; NR and NG-RAN overall description; Stage 2 (Release 15)," June 2018.
[2] NGMN Alliance, "NGMN 5G WHITE PAPER," Feb. 2015.
[3] NGMN Alliance, "Description of Network Slicing Concept," Jan. 2016.
[4] ETSI GS NFV 002 (V1.2.1), "Network Functions Virtualisation (NFV); Architectural Framework," Dec. 2014.
[5] N. Mckeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, "Openflow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69-74, Apr. 2008.
[6] M. I. Kamel, L. B. Le, and A. Girard, "LTE wireless network virtualization: Dynamic slicing via flexible scheduling," Proc. of IEEE 80th Vehicular Technology Conference (VTC2014-Fall), Vancouver, Canada, Sept. 2014.
[7] L. Yin, L. Qiu, and Z. Chen, "Throughput-maximum resource provision in the OFDMA-based wireless virtual network," Proc. of IEEE 85th Vehicular Technology Conference (VTC2017-Spring), Sydney, Australia, May 2017.
[8] P. Caballero, A. Banchs, G. Verciana, and X. Perez, "Multi-tenant radio access network slicing: Statistical multiplexing of spatial loads," IEEE/ACM Trans. on Networking, vol. 25, no. 5, pp. 3044-3058, Oct. 2017.
[9] V. Ha and L. Le, "End-to-end network slicing in virtualized OFDMA-based cloud radio access networks," IEEE Access, vol. 5, pp. 18675-18691, Sept. 2017.
[10] X. Li, M. Samaka, H. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5G: Challenges and opportunities," IEEE Internet Computing, vol. 21, no. 5, pp. 20-27, Sept. 2017.
[11] V. Sciancalepore, K. Samdanis, X. Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," Proc. of IEEE International Conference on Computer Communications (INFOCOM 2017), Atlanta, Georgia, May 2017.
[12] S. Mizuno, D. Muramatsu, Y. Yuda and K. Higuchi, "Investigation on optimum frequency bandwidth allocation method among service channels for system throughput maximization," Proc. of IEEE 23rd Asia-Pacific Conference on Communications (APCC2017), Perth, Australia, Dec. 2017.
[13] K. Zhu and E. Hossain, "Virtualization of 5G Cellular Networks as a Hierarchical Combinatorial Auction," IEEE Trans. on Mobile Computing, vol. 15, no. 10, pp. 2640-2654, Oct. 2016.
[14] J. Qiu, G. Ding, Q. Wu, Z. Qian, T. A. Tsiftsis, Z. Du, and Y. Sun, "Hierarchical Resource Allocation Framework for Hyper-Dense Small Cell Networks," IEEE Access, vol. 4, pp. 8657-8669, Nov. 2016.
[15] A. Jalali, R. Padovani and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," Proc. of IEEE 51st Vehicular Technology Conference (VTC2000-Spring), Tokyo, Japan, May 2000.
[16] 3GPP TS 36.213 V14.4.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 14)," Sept. 2017.

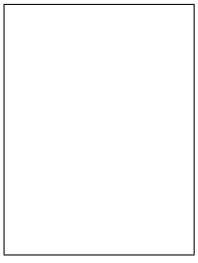
[17] 3GPP TR 36.814 V9.2.0, “Further advancements for E-UTRA physical layer aspects (Release 9),” Mar. 2017.
[18] 3GPP TR 38.802 V14.2.0, “Study on New Radio Access Technology Physical Layer Aspects (Release 14),” Sept. 2017.



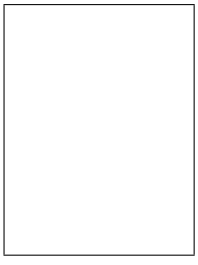
Daisuke NOJIMA



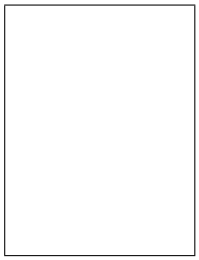
Yuki KATSUMATA



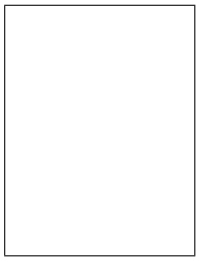
Yoshifumi MORIHIRO



Takahiro ASAI



Akira YAMADA



Shigeru IWASHINA