

Control Period Adaptation and Resource Allocation for Joint Uplink and Downlink in NB-IoT Networks

Ya-Ju Yu, You-Chiun Wang, and Chia-Hsin Fan

Abstract—Narrowband Internet of Things (NB-IoT) offers three coverage enhancement (CE) levels to serve massive machines in a large area. For each CE level, the base station configures control periods to determine the number of allocatable radio resources for signal and data transmissions. Both uplink (UL) and downlink (DL) communications use the same control periods, but UL and DL machines need different period lengths. In general, a control period suitable for the UL is longer than that for the DL. Then, the base station assigns UL and DL resources in control periods to each machine. To this end, we study how to choose a suitable length of control periods and allot radio resources to UL and DL machines to minimize resource consumption, thereby improving NB-IoT performance. Two efficient algorithms are thus proposed. Based on the CE level, the *control period adaptation algorithm* flexibly adjusts control periods using a scale factor. The *joint UL and DL resource allocation algorithm* distributes radio resources in each control period among machines to increase utilization. Simulation results demonstrate that our algorithms can efficiently decrease the consumption of UL and DL subframes, especially for machines with bad channel qualities.

Index Terms—control period, downlink, NB-IoT, resource allocation, uplink.

1 INTRODUCTION

THE 3rd generation partnership project (3GPP) regulates narrowband Internet of Things (NB-IoT) to support massive connections and large coverage [1]. It is considered one essential technology for cellular IoT connections [2]. NB-IoT machines employ radio resources reserved by cellular systems, which are sparse and valuable. Besides, many machines send or receive small data packets. Therefore, the issue of improving NB-IoT resource utilization is important.

NB-IoT supports three coverage enhancement (CE) levels to extend coverage areas, as shown in Fig. 1. A base station (BS) can configure control periods for each CE level, which are shared by uplink (UL) and downlink (DL) communications. One control period contains a narrowband physical DL control channel (NPDCCH) and a narrowband physical DL shared channel (NPDSCH) in the DL direction. Besides, one or more narrowband physical random access channels (NPRACHs) and a narrowband physical UL shared channel (NPUSCH) occupy UL resources in a control period. The control period determines the lengths of NPDCCH, NPDSCH, and NPUSCH.

How to adjust the length of a control period (also known as *control period adaptation*) and parcel out its radio resources to machines (called *resource allocation*) are two key problems to improving NB-IoT resource utilization. These two problems are challenging because we need to consider multiple factors for machines, such as resource types, modulation and coding schemes (MCSs), and repetitions. A too-long control period will cause a waste of resources, as some subframes may not be utilized. If the period length is set too short, some machines may need more control periods to meet data demands, thereby

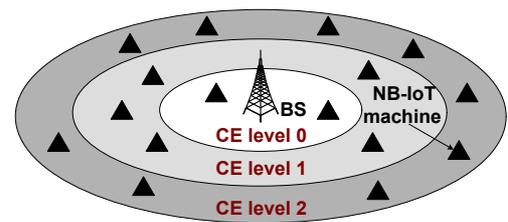


Fig. 1: Three CE levels in NB-IoT.

wasting energy. Even with proper control periods, improper resource allocation could lead to low resource utilization [3].

The existing solutions to the control period adaptation and resource allocation issues rest on the assumption that only UL or DL communications take place in a control period. In effect, NB-IoT allows UL and DL communications to share the same control periods. However, existing solutions cannot efficiently deal with this case. The reason is that four resource types are supported for UL communications, while only one resource type is given to DL communications in NB-IoT [4]. Moreover, the BS has larger transmitted power than machines [5]. Thus, a machine needs a longer NPUSCH in the UL direction than the NPDSCH in the DL direction. Unavoidably, suitable control period lengths for UL and DL will be different. However, for a CE level, merely one period length can be set for both UL and DL. As can be seen, it is necessary to design new methods for the case of mixed UL and DL communications.

Motivated by the above observation, this article formulates the problems of control period adaptation and resource allocation for joint UL and DL communications in NB-IoT. The objective is to use as few UL and DL subframes as possible to meet the data demands of all machines. In this way, we can increase overall resource utilization and improve NB-IoT performance. To do so, we propose a *control period adaptation algorithm* that uses a scale factor to flexibly adjust the length

Ya-Ju Yu is with the Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 81148, Taiwan (e-mail: yjyu@nuk.edu.tw).

You-Chiun Wang and Chia-Hsin Fan are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (e-mail: ycwang@cse.nsysu.edu.tw; fanchsin@gmail.com).

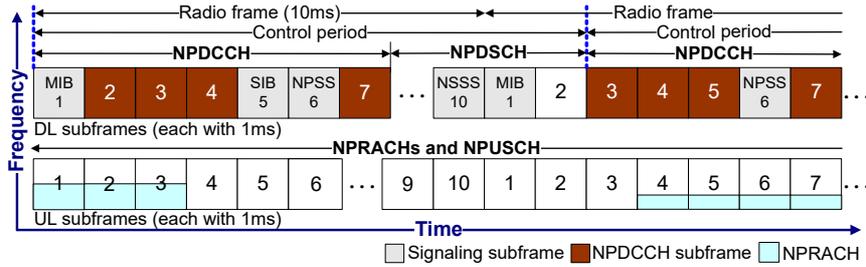


Fig. 2: NB-IoT frame structure (MIB: master information block; SIB: system information block; NPSS/NSSS: narrowband primary/secondary synchronization signal).

of a control period depending on the CE level. According to the determined control period, a *joint UL and DL resource allocation algorithm* is proposed to distribute radio resources among machines to better utilize resources.

Our contributions are twofold:

- Unlike previous studies that considered either UL or DL communications, we solve the control period adaptation and resource allocation problems in a more general case where UL and DL communications coexist in the same control periods.
- We implement four solutions developed by recent studies [6]–[9] for performance comparison. Simulation results reveal that our proposed algorithms can efficiently reduce UL and DL subframe consumption at different CE levels, especially for machines with bad channel qualities.

The rest of this article is organized as follows: Section 2 surveys related work. Section 3 gives the system model and problem formulation. Then, we detail the proposed algorithms and analyze their time complexities in Section 4. The performance evaluation is presented in Section 5. Finally, Section 6 contains concluding remarks and future work.

2 RELATED WORK

Many studies have discussed issues related to UL resource allocation in NB-IoT. Hsieh *et al.* [6] give options for a UL scheduler and propose a solution to allocate *DL control information (DCI)* and subcarriers. Liang *et al.* [10] discuss how to ensure reliable UL communications and minimize the energy consumption of NB-IoT machines. The study [11] develops two loop link adaptation methods for transmission reliability and throughput improvement. The inner loop link adaptation handles block error ratio variation, and the outer one selects MCSs and data repetitions. In [12], a joint power control and resource allocation method is proposed to maximize the energy efficiency of cluster-based NB-IoT networks. The study [13] analyzes relevant factors for NB-IoT UL resource scheduling and adjusts the selection of MCSs and repetitions to decrease activity time and resource consumption. Elgarhy *et al.* [14] investigate the tradeoff between rate and latency in resource allocation for NB-IoT networks. The work [7] takes account of NPRACHs in both link adaptation and resource allocation for UL communications to reduce subframe consumption. In [15], a UL resource allocation approach is proposed to exploit the NPUSCH subframes of the next control period to improve the utilization of radio resources. Kodheli *et al.* [16] consider low-Earth orbit satellites in UL resource allocation to maximize a profit function. The study [17] applies reinforcement learning to the control of UL transmissions.

On the other hand, regarding DL resource allocation, Reddy *et al.* [8] explain the rationale and search space allocation for NPDCCH and design NPDCCH scheduling algorithms. The work [9] deals with both period adaptation and DL scheduling for NPDCCH. In accordance with the constraints of NPDCCH, a resource allocation method is proposed in [18] to reduce power consumption and improve NPDCCH's utilization. Using non-orthogonal multiple access, the study [19] maximizes DL user connection density via a graph-matching method.

With the NPDCCH offset mechanism, the study [20] allots UL and DL resources to NB-IoT machines, whose objective is to minimize resource usage and also allow each machine to transmit or receive its data. To the best of our knowledge, the issue of finding suitable control period lengths for UL and DL communications at the same time has not been addressed yet. Compared to previous studies, our work copes with control period adaptation and resource allocation for joint UL and DL, which can efficiently reduce consumed subframes and thereby improve resource utilization in NB-IoT.

3 SYSTEM MODEL AND PROBLEM DEFINITION

3.1 System Model

In NB-IoT, a BS takes the frequency division duplex mode for the UL and DL. The channel bandwidth is 180 kHz. A DL subcarrier has a bandwidth of 15 kHz. For a UL subcarrier, the bandwidth can be 15 kHz or 3.75 kHz. Fig. 2 shows the frame structure. Each radio frame lasts 10 ms and has 10 subframes. In the DL direction, subframes are used for signals, NPDCCH, or NPDSCH. DCIs and DL data are sent to machines through NPDCCH and NPDSCH subframes, respectively. Regarding the UL direction, there are two channel types: NPRACHs for the random access procedure and NPUSCH for UL data.

3.1.1 Control Period

Let us consider a BS serving NB-IoT machines, as shown in Fig. 1. For each CE level, the BS can configure a combination of R_{\max} and G values to adjust control periods. Here, R_{\max} decides the number of subframes used for NPDCCH, and G is a system parameter. Except for NPDCCH and signaling subframes, other subframes in a control period can be used as NPDSCH to send DL data. In the UL direction, in addition to radio resources allocated to NPRACHs, other resources are used as NPUSCH for UL communications. Fig. 2 gives an example. By setting $R_{\max} = 4$ and $G = 3$, a control period has 12 subframes (*i.e.*, $R_{\max} \times G = 12$). Signals have the highest priority to use subframes, and NPDCCH is in front of NPDSCH. In the first control period, since subframes 1, 5, and 6 (of the first radio frame) have been occupied by signals,

NPDCCH uses subframes 2, 3, 4, and 7. Other subframes not used by signals in the control period are allocated to NPDSCH.

The control period length has a great impact on the utilization of radio resources. Both UL and DL follow the same control period length, but their appropriate control periods are different. The UL requires a longer control period than the DL due to two reasons. First, a machine has weaker signals than a BS. A longer control period improves the chance of successful UL transmissions. Second, UL communications can have four resource types, but only one resource type is given to DL communications. Thus, a suitable control period for the UL will be naturally longer to support multiple resource types. To this end, we investigate how to find a proper control period length for each CE level when both UL and DL are considered. This problem is called *control period adaptation*.

3.1.2 Resource Allocation

After the control period is settled, the BS allocates UL and DL resources in each control period and decides related parameters carried by a DCI. To do so, the BS shall decide 1) resource assignment, 2) MCS, 3) scheduling delay, 4) DCIs, and 5) resource type for each machine. This problem is referred to as the *resource allocation* problem.

Resource assignment means the number of resource units given to a machine without repetition. We can use this number and an MCS index to look up the *transport block size (TBS)* table to know the number of data bits that a machine sends or receives. An MCS index is equal to a TBS index. When an MCS and a resource type are used for a machine, they need a *repetition number (RN)* to satisfy the transmission reliability of data according to a *signal-to-noise ratio (SNR)*. However, a combination of MCS and resource type may not be feasible for a machine if using the maximum RN still cannot meet the required transmission reliability.

Scheduling delays and DCIs control machines to receive or send data in NPDSCH or NPUSCH subframes, respectively. For the DL, two sets of eight scheduling delays are supported, depending on the R_{\max} value, and there is only one resource type. For the UL, four scheduling delays are supported. There are four resource types: 12 subcarriers with 1 ms, 6 subcarriers with 2 ms, 3 subcarriers with 4 ms, and 1 subcarrier with 8 ms. The BS selects one resource type for a resource unit. Table 1 summarizes the acronyms.

3.2 Problem Definition

In a channel, each subframe has F subcarriers. For the UL, the BS can support U resource types for a resource unit ($U = 4$). Resource type u requires f_u subcarriers in the frequency domain and t_u subframes in the time domain, where $1 \leq u \leq U$. Here, $f_u > f_{u+1}$ and $t_u < t_{u+1}$, for $u = 1, \dots, U - 1$. Regarding the DL, only one resource type is available (*i.e.*, $U = 1$). There is a set of allocable numbers for a resource unit, as denoted by $\hat{I}_{RU} = \{I_1, I_2, \dots, I_h, \dots, I_H\}$. Moreover, $I_{h,d}$ signifies the number of I_h resource units for machine d .

Let \hat{D}_{UL} and \hat{D}_{DL} be the sets of UL and DL machines served by the BS at a given CE level. Since each machine can only perform either UL or DL communications at a time, we have $\hat{D}_{UL} \cap \hat{D}_{DL} = \emptyset$. A machine d requests a data size of ψ_d . The SNR between the BS and machine d is $10 \log_{10}(\rho/\sigma)$, where ρ is the received power at machine d in the DL or at the BS in the UL. In addition to the resource type, the received power will

TABLE 1: Summary of acronyms.

Acronym	Full name
BS	Base station
CE	Coverage enhancement
CPARA	Control period adaptation and resource allocation
DCI	Downlink control information
DL/UL	Downlink/uplink
LAURA	Link adaptation and uplink resource allocation
MCS	Modulation and coding scheme
MRR	Max-Ri relaxed
NANIS	NPDCCH period adaptation and NB-IoT scheduling
NB-IoT	Narrowband Internet of Things
NCCE	Narrowband control channel element
NPDCCH	Narrowband physical downlink control channel
NPDSCH	Narrowband physical downlink shared channel
NPRACH	Narrowband physical random access channel
NPUSCH	Narrowband physical uplink shared channel
RN	Repetition number
SNR	Signal-to-noise ratio
SS	Start subframe
TBS	Transport block size
UUS	UE-specific uplink scheduler

be affected by the transmitted power, path loss, shadowing, and fast fading [21]. Besides, σ is the noise spectral density.

The BS offers M MCS indexes. With the SNR, machine d at least requires a data RN $\bar{N}_{d,u,m}^{UL}$ for the UL and $\bar{N}_{d,1,m}^{DL}$ for the DL to ensure transmission reliability when resource type u and MCS m are used. To decode a DCI, machine d requires a DCI RN no smaller than \bar{N}_d^{DCI} . The BS provides three sets of RNs \hat{I}_{Rep}^{UL} , \hat{I}_{Rep}^{DL} , and \hat{I}_{Rep}^{DCI} for UL, DL, and DCI, respectively. Based on the allocated resource unit number $I_{h,d}$ and MCS m , a data size of $\eta(I_{h,d}, m)$ can be provided to machine d .

The length of a control period is $R_{\max} \times G$ (in subframes), where R_{\max} decides the number of control subframes. We have a set \hat{V}_R of R_{\max} values and a set \hat{V}_G of G values. Given a DCI RN $R \in \hat{I}_{Rep}^{DCI}$, the number of allocable DCIs in the DL is $\xi = R_{\max}/R$. We consider that R is constant for each CE level. Each machine has to monitor its NPDCCH search space to blindly decode DCI for transmitting or receiving data. An NPDCCH subframe has two *narrowband control channel elements (NCCEs)*. There are two formats of DCI. Format N0 (for the UL) occupies one NCCE, and format N1 (for the DL) needs both NCCEs. Hence, the resource requirement of the DCI format for the UL is half that for the DL. In other words, the number of allocable DCIs is 2ξ in the UL.

Let us also define a function $\lambda(d, c, p)$. If DCI c in the p -th control period is assigned to machine d , we have $\lambda(d, c, p) = 0.5$ for the UL and $\lambda(d, c, p) = 1$ for the DL. Otherwise, we will have $\lambda(d, c, p) = 0$. Moreover, $S_{d,s,p}^{UL}$ and $S_{d,s,p}^{DL}$ indicate the number of UL and DL subcarriers at subframe s in the p -th control period used for machine d , respectively. The sets of scheduling delays for the UL and DL are represented as $\hat{K}_{UL} = \{k_1^{UL}, k_2^{UL}, \dots, k_i^{UL}, \dots\}$ and $\hat{K}_{DL} = \{k_1^{DL}, k_2^{DL}, \dots, k_i^{DL}, \dots\}$. When using DCI c with delay k_i^{UL} , the UL *start subframe (SS)* can be calculated by $b_{c,i}^{UL} = k_i^{UL} + \tau_p^c + 1$, where τ_p^c is the last NPDCCH subframe to send DCI c in the p -th control period, and the constant 1 is the switching time from NPDCCH to NPUSCH [7]. If DCI c with delay k_i^{DL} is used, the DL SS can be estimated by $b_{c,i}^{DL} = k_i^{DL} + \tau_p^c + 5$, where 5 is the constant for hardware and software processing delays [9].

According to the above specification of NB-IoT defined by 3GPP, our target problems (*i.e.*, control period adaptation and resource allocation) have the following constraints:

Repetition constraint: To ensure the communication relia-

TABLE 2: Summary of notations

Notations	Description
$\hat{\mathcal{D}}_{\text{UL}}, \hat{\mathcal{D}}_{\text{DL}}$	Sets of UL and DL NB-IoT machines
$\hat{\mathcal{I}}_{\text{RU}}$	Set of allocable resource unit numbers
$\hat{\mathcal{I}}_{\text{Rep}}^{\text{DCI}}, \hat{\mathcal{I}}_{\text{Rep}}^{\text{UL}}, \hat{\mathcal{I}}_{\text{Rep}}^{\text{DL}}$	Sets of RNs for DCI, UL, and DL
$\hat{\mathcal{V}}_{\text{R}}, \hat{\mathcal{V}}_{\text{G}}$	Sets of R_{max} and G values
$\hat{\mathcal{K}}_{\text{UL}}, \hat{\mathcal{K}}_{\text{DL}}$	Sets of scheduling delays for UL and DL
R_{max}	Number of control subframes in a control period
F	Number of subcarriers in a subframe
U, M	Numbers of resource types and MCSs
P	Number of consumed control periods
ξ	Number of allocable DCIs in the DL
$S_{d,s,p}^{\text{UL}}, S_{d,s,p}^{\text{DL}}$	Numbers of UL and DL subcarriers at subframe s in the p -th control period used for machine d
G	System parameter to decide a control period
R	DCI RN for transmissions
f_u, t_u	Subcarriers and subframes for resource type u
ψ_d	Data demand of machine d
N_d^{DCI}	DCI RN for machine d
$N_{d,u,m}^{\text{UL}}, N_{d,1,m}^{\text{DL}}$	UL and DL data RNs for machine d with resource type u and MCS m
$b_{c,i}^{\text{UL}}, b_{c,i}^{\text{DL}}$	UL and DL Ss using DCI c with delays k_i^{UL} and k_i^{DL}
$\eta(I_{h,d}, m)$	Achievable data size when $I_{h,d}$ resource units and MCS m are assigned to machine d
$\lambda(d, c, p)$	This function is 0.5 and 1 for UL and DL if DCI c of the p -th control period is given to machine d

bility of each machine for the UL, DL, and DCI, the BS needs to choose numbers $N_{d,u,m'}^{\text{UL}}$, $N_{d,1,m'}^{\text{DL}}$, and $R_{\text{max}'}$, subject to

$$N_{d,u,m}^{\text{UL}} \geq \bar{N}_{d,u,m}^{\text{UL}}, \quad N_{d,u,m}^{\text{UL}} \in \hat{\mathcal{I}}_{\text{Rep}}^{\text{UL}}, \quad (1)$$

$$N_{d,1,m}^{\text{DL}} \geq \bar{N}_{d,1,m}^{\text{DL}}, \quad N_{d,1,m}^{\text{DL}} \in \hat{\mathcal{I}}_{\text{Rep}}^{\text{DL}}, \quad (2)$$

$$N_d^{\text{DCI}} = R_{\text{max}}/\xi \geq \bar{N}_d^{\text{DCI}}, \quad R_{\text{max}} \in \hat{\mathcal{V}}_{\text{R}}. \quad (3)$$

Demand constraint: Let P denote the number of consumed control periods. Each machine d can send (*i.e.*, UL) or receive (*i.e.*, DL) the data size of at least ψ_d to satisfy its demand:

$$\sum_{p=1}^P \sum_{I_{h,d} \in \hat{\mathcal{I}}_{\text{RU}}} \eta(I_{h,d}, m) \geq \psi_d, \quad \forall d \in \hat{\mathcal{D}}_{\text{UL}} \cup \hat{\mathcal{D}}_{\text{DL}}. \quad (4)$$

Subcarrier constraint: The number of subcarriers used in each subframe cannot exceed its capacity (*i.e.*, F):

$$\sum_{d \in \hat{\mathcal{D}}_{\text{UL}}} S_{d,s,p}^{\text{UL}} \leq F \text{ and } \sum_{d \in \hat{\mathcal{D}}_{\text{DL}}} S_{d,s,p}^{\text{DL}} \leq F, \quad \forall s, p. \quad (5)$$

Signaling constraint: If machine d transmits data in UL subframes or receives data in DL subframes, it has to receive a DCI in a series of control subframes:

$$\sum_{s=1}^{R_{\text{max}} \times G} S_{d,s,p}^{\text{UL}} \geq 1, \quad \sum_{c=1}^{2\xi} \lambda(d, c, p) = 0.5, \quad \forall p, d \in \hat{\mathcal{D}}_{\text{UL}}, \quad (6)$$

$$\sum_{s=1}^{R_{\text{max}} \times G} S_{d,s,p}^{\text{DL}} \geq 1, \quad \sum_{c=1}^{\xi} \lambda(d, c, p) = 1, \quad \forall p, d \in \hat{\mathcal{D}}_{\text{DL}}. \quad (7)$$

Since the goal is to use the minimum number of subframes to fulfill the UL and DL demands of all NB-IoT machines at a CE level, the objective function can be expressed as follows:

$$\arg \min_{R_{\text{max}}, G, \lambda(d, c, p), S_{d,s,p}^{\text{UL}}, S_{d,s,p}^{\text{DL}}, I_{h,d}} P \times R_{\text{max}} \times G. \quad (8)$$

In Eq. (8), we determine parameters $R_{\text{max}'}$, G , $\lambda(d, c, p)$, $S_{d,s,p}^{\text{UL}}$, $S_{d,s,p}^{\text{DL}}$, and $I_{h,d}$ to minimize used subframes. Since P control periods are consumed, the total number of subframes is $P \times R_{\text{max}} \times G$. Table 2 summarizes the notations.

4 PERIOD ADAPTATION AND RESOURCE ALLOCATION

This section details two algorithms: 1) the control period adaptation algorithm for a CE level, and 2) the joint UL and DL resource algorithm based on the control period. Then, we analyze their time complexities and discuss some issues.

4.1 Control Period Adaptation

Algorithm 1 presents the pseudocode of the control period adaptation algorithm. In NB-IoT, machines need a longer time for the UL and have to monitor the control subframes in a DL channel. To find a suitable control period, UL and DCI RN requirements (*i.e.*, $N_{d,u,m}^{\text{UL}}$ and N_d^{DCI}) should be considered first. More concretely, to obtain $\mathcal{L}_{\text{Data}}$, the estimated length of NPUSCH, in line 1, we compute the average data RN required by UL machines using each resource type and MCS¹. Since there are ξ allocable DCIs, $\mathcal{L}_{\text{Data}}$ is set to be the product of this average and ξ . In line 2, \mathcal{L}_{DCI} is the estimated length of NPDCCH. As each machine must monitor its search space, the machine needs enough DCI repetitions (no matter UL or DL communications). Hence, \mathcal{L}_{DCI} is set to be the product of the maximum DCI RN required by all machines and ξ .

In addition to $\mathcal{L}_{\text{Data}}$ and \mathcal{L}_{DCI} depicted for RN requirements of data and DCIs, we design a scale factor γ that considers resource types to flexibly adjust the control period for different CE levels. Specifically, the lengths of NPDSCH and NPUSCH are extended by γ times. In Algorithm 1, the code in lines 3–5 deals with the case where only DL machines are served (*i.e.*, $\hat{\mathcal{D}}_{\text{UL}} = \emptyset$). In this case, merely one resource type with 1 ms and 12 subcarriers (for the DL) is supported. There is no need to extend the control period length, so we set $\gamma = 1$. Moreover, we calculate the average data RN (*i.e.*, $N_{d,1,m}^{\text{DL}}$) of DL machines using each MCS and update $\mathcal{L}_{\text{Data}}$ as the product of this average and ξ . Then, the code in lines 6–11 handles cases where UL and DL machines coexist at three CE levels. Since the machines whose channel qualities are worse require larger RNs and consume more resources, we need a longer control period at a higher CE level. Therefore, the scale factor for CE levels 0, 1, and 2 is set to 1, 2 and 8, respectively. Remark 1 discusses why we set $\gamma \in \{1, 2, 8\}$.

In lines 12–16, we search all possible combinations of R_{max} and G values to find a suitable control period for the CE level. Two conditions have to be met. First, the R_{max} value (deciding the number of control subframes) should be no smaller than \mathcal{L}_{DCI} (*i.e.*, the estimated length of NPDCCH), as shown in line 14. Second, the value of $R_{\text{max}} \times (G - 1)$ affects the lengths of both NPDSCH and NPUSCH for data delivery. Apparently, it should be at least $\mathcal{L}_{\text{Data}} \times \gamma$, as indicated in line 15. If such a combination can be found, line 16 returns the R_{max} and G values. Otherwise, a null value is returned by line 17.

Remark 1 (Scale factor). For CE level 0, we set $\gamma = 1$, as each machine generally has a good channel quality and can use the resource type with 1 ms (*i.e.*, $t_u = 1$). Thus, machines just need a short length for NPUSCH. Since the resource type with 1 ms may not be available for UL communications at CE level 1, the resource type with 2 ms (*i.e.*, $t_u = 2$) will be more suitable for machines. Hence, the γ value is set to 2 to extend NPUSCH's length. Similarly,

1. Note that we exclude those resource types and MCSs of machines that cannot meet transmission reliability even using their maximum RNs.

Algorithm 1: Control Period Adaptation

Data: sets $\hat{\mathcal{D}}_{\text{UL}}, \hat{\mathcal{D}}_{\text{DL}}, \hat{\mathcal{V}}_{\text{R}}, \hat{\mathcal{V}}_{\text{G}}$; allocable DCIs ξ ; RNs N_d^{DCI} , $N_{d,u,m}^{\text{UL}}, N_{d,1,m}^{\text{DL}}$ for each machine

Result: parameters R_{max} and G for a control period

```

1  $\mathcal{L}_{\text{Data}} \leftarrow \frac{\sum_{d \in \hat{\mathcal{D}}_{\text{UL}}} \sum_{u=1}^U \sum_{m=1}^M N_{d,u,m}^{\text{UL}}}{|\hat{\mathcal{D}}_{\text{UL}}| \times U \times M} \times \xi$ ;
2  $\mathcal{L}_{\text{DCI}} \leftarrow \max_{d \in \hat{\mathcal{D}}_{\text{UL}} \cup \hat{\mathcal{D}}_{\text{DL}}} \{N_d^{\text{DCI}}\} \times \xi$ ;
3 if  $\hat{\mathcal{D}}_{\text{UL}} = \emptyset$  then
4    $\gamma \leftarrow 1$ ;
5    $\mathcal{L}_{\text{Data}} \leftarrow \frac{\sum_{d \in \hat{\mathcal{D}}_{\text{DL}}} \sum_{m=1}^M N_{d,1,m}^{\text{DL}}}{|\hat{\mathcal{D}}_{\text{DL}}| \times M} \times \xi$ ;
6 else if CE level is 0 then
7    $\gamma \leftarrow 1$ ;
8 else if CE level is 1 then
9    $\gamma \leftarrow 2$ ;
10 else
11    $\gamma \leftarrow 8$ ;
12 for  $R_{\text{max}} \in \hat{\mathcal{V}}_{\text{R}}$  do
13   for  $G \in \hat{\mathcal{V}}_{\text{G}}$  do
14     if  $R_{\text{max}} \geq \mathcal{L}_{\text{DCI}}$  then
15       if  $R_{\text{max}} \times (G - 1) \geq \mathcal{L}_{\text{Data}} \times \gamma$  then
16         return  $R_{\text{max}}$  and  $G$ ;
17 return null;

```

because machines at CE level 2 have worse channel qualities, the BS may generally select the resource type with 8 ms (*i.e.*, $t_u = 8$) for machines. We require further prolonging NPUSCH's length by setting the γ value to 8 at CE level 2.

4.2 Joint UL and DL Resource Allocation

After deciding parameters R_{max} and G for a control period, this algorithm distributes radio resources among NB-IoT machines to meet their demands (*i.e.*, ψ_d), whose detailed steps are given in Algorithm 2.

In lines 1 and 2, we find resource type u_d and MCS m_d to maximize transmission efficiency for each UL machine. Here, $\eta(I_1, m)$ is the achievable data size using one resource unit (*i.e.*, I_1) and MCS m , and $N_{d,u,m}^{\text{UL}}$ is machine d 's data RN for UL using resource type u and MCS m . We choose u_d and m_d such that the average number of bits sent per data repetition (*i.e.*, $\eta(I_1, m_d)/N_{d,u_d,m_d}^{\text{UL}}$) is the maximum. Similarly, the code in lines 3 and 4 picks the best MCS m_d for each DL machine to maximize transmission efficiency. Since only one resource type is supported for the DL, we do not consider the resource type (*i.e.*, u_d) in line 4.

Line 5 initializes P (*i.e.*, the number of consumed control periods) as zero. Then, the while-loop in lines 6–16 allots radio resources to machines period by period until their demands can be met. Since the UL requires a longer control period than the DL, we consider satisfying UL demands first. For convenience, we use variables $\hat{\mathcal{D}}, \hat{\mathcal{K}}, \hat{\mathcal{I}}_{\text{Rep}}, \tilde{\xi}$, and $S_{d,s,P}$ to record parameters for UL and DL machines. The codes in lines 8–9 and 10–11 are for UL and DL machines, respectively.

In a control period, we try to allocate $\tilde{\xi}$ DCIs, as shown in lines 12–16. To do so, four auxiliary procedures are used²:

2. Since variables $R_{\text{max}}, G, \hat{\mathcal{D}}$, and P are required by most procedures, we assume that they are global variables and omit these four variables from the input parameters of each procedure.

Algorithm 2: Joint UL and DL Resource Allocation

Data: control period parameters R_{max} and G ; sets $\hat{\mathcal{D}}_{\text{UL}}, \hat{\mathcal{D}}_{\text{DL}}, \hat{\mathcal{K}}_{\text{UL}}, \hat{\mathcal{K}}_{\text{DL}}, \hat{\mathcal{I}}_{\text{Rep}}, \hat{\mathcal{I}}_{\text{Rep}}^{\text{UL}}, \hat{\mathcal{I}}_{\text{Rep}}^{\text{DL}}$; allocable DCIs ξ

Result: number P of used control periods; resource allocation parameters $S_{d,s,p}^{\text{UL}}, S_{d,s,p}^{\text{DL}}, \lambda(d, c, p), I_{h,d}$ for machine d in each control period p

```

1 foreach  $d \in \hat{\mathcal{D}}_{\text{UL}}$  do
2    $(u_d, m_d) \leftarrow \arg \max_{u,m} \{\eta(I_1, m)/N_{d,u,m}^{\text{UL}}\}$ ;
3 foreach  $d \in \hat{\mathcal{D}}_{\text{DL}}$  do
4    $m_d \leftarrow \arg \max_m \{\eta(I_1, m)/N_{d,1,m}^{\text{DL}}\}$ ;
5  $P \leftarrow 0$ ;
6 while  $\psi_d > 0, \exists d \in \hat{\mathcal{D}}_{\text{UL}} \cup \hat{\mathcal{D}}_{\text{DL}}$  do
7    $P \leftarrow P + 1$ ;
8   if  $\psi_d > 0, \exists d \in \hat{\mathcal{D}}_{\text{UL}}$  then
9      $\hat{\mathcal{D}} \leftarrow \hat{\mathcal{D}}_{\text{UL}}, \hat{\mathcal{K}} \leftarrow \hat{\mathcal{K}}_{\text{UL}}, \hat{\mathcal{I}}_{\text{Rep}} \leftarrow \hat{\mathcal{I}}_{\text{Rep}}^{\text{UL}}, \tilde{\xi} \leftarrow 2\xi,$ 
10     $S_{d,s,P} \leftarrow S_{d,s,P}^{\text{UL}}$ ;
11   else
12      $\hat{\mathcal{D}} \leftarrow \hat{\mathcal{D}}_{\text{DL}}, \hat{\mathcal{K}} \leftarrow \hat{\mathcal{K}}_{\text{DL}}, \hat{\mathcal{I}}_{\text{Rep}} \leftarrow \hat{\mathcal{I}}_{\text{Rep}}^{\text{DL}}, \tilde{\xi} \leftarrow \xi,$ 
13     $S_{d,s,P} \leftarrow S_{d,s,P}^{\text{DL}}$ ;
14   for  $c = 1$  to  $\tilde{\xi}$  do
15      $(b_{\text{used}}, b_{\text{last}}, c_{\text{used}}) \leftarrow \text{DCI-Delay}(\hat{\mathcal{K}}, \tilde{\xi})$ ;
16      $E_d \leftarrow \text{Usage}(\hat{\mathcal{I}}_{\text{Rep}}, b_{\text{used}}, b_{\text{last}})$ ;
17      $\hat{\mathcal{D}}' \leftarrow \text{Machine}(b_{\text{used}}, b_{\text{last}}, E_d)$ ;
18      $(I_{h,d}, \lambda(d, c, P), S_{d,s,P}) \leftarrow \text{Allocation}(\hat{\mathcal{D}}',$ 
19      $\hat{\mathcal{I}}_{\text{Rep}}, b_{\text{used}}, c_{\text{used}}, E_d)$ ;
20 return  $P, S_{d,s,p}^{\text{UL}}, S_{d,s,p}^{\text{DL}}, \lambda(d, c, p), I_{h,d}$ ;

```

- **DCI-Delay()**: Based on scheduling delays (*i.e.*, $\hat{\mathcal{K}}$) and allocable DCIs (*i.e.*, $\tilde{\xi}$), this procedure returns the earliest SS b_{used} to be used, the last SS b_{last} to be reserved, and the DCI c_{used} to be used. Here, we reserve the last SS to avoid the subframes behind it being wasted.
- **Usage()**: Given set $\hat{\mathcal{I}}_{\text{Rep}}$ of data RNs and two SSs b_{used} and b_{last} , the Usage() procedure finds the number E_d of allocable subframes for each machine.
- **Machine()**: Taking $b_{\text{used}}, b_{\text{last}}$, and E_d as inputs, the Machine() procedure selects a subset $\hat{\mathcal{D}}'$ of machines from $\hat{\mathcal{D}}$ for serving, whose goal is to maximize resource utilization. Notice that at most two UL machines or one DL machine can be included in $\hat{\mathcal{D}}'$.
- **Allocation()**: With $\hat{\mathcal{I}}_{\text{Rep}}, b_{\text{used}}$, and E_d , it allots the DCI c_{used} and subframes to the selected machines in $\hat{\mathcal{D}}'$. This procedure returns the number $I_{h,d}$ of resource units, function $\lambda(d, c, P)$, and the number $S_{d,s,P}$ of subcarriers at subframe s for each machine $d \in \hat{\mathcal{D}}'$.

If resources are not enough, we go to the next control period (*i.e.*, increasing P by one in line 7). When the demands of all machines have been met, line 17 returns the resource allocation parameters. Then, we elaborate on each procedure.

4.2.1 DCI-Delay Procedure

This procedure picks the earliest SS and reserves the last SS. The for-loop in lines 2–8 searches for the earliest SS b_{used} from each DCI c and scheduling delay k_i , where b_{used} is initialized as ∞ by line 1. Then, the code in lines 4–5 copes with two special cases: 1) the SS is behind the control period (*i.e.*, $b_{c,i} > R_{\text{max}} \times$

```

Procedure DCI-Delay ( $\hat{\mathcal{K}}, \tilde{\xi}$ ):
1   $b_{\text{used}} \leftarrow \infty$ ;
2  for  $c = 1$  to  $\tilde{\xi}$  do
3    foreach  $k_i \in \hat{\mathcal{K}}$  do
4      if  $b_{c,i} > R_{\text{max}} \times G$  or  $(b_{c,i} \leq R_{\text{max}}$  and  $\hat{\mathcal{K}} = \hat{\mathcal{K}}_{\text{DL}}$ )
5        then
6           $\lfloor$  continue;
7      if  $\sum_{d \in \hat{\mathcal{D}}_{\text{UL}} \cup \hat{\mathcal{D}}_{\text{DL}}} \lambda(d, c, P) < 1$  and
8         $\sum_{d \in \hat{\mathcal{D}}} S_{d, b_{c,i}, P} < F$  then
9          if  $b_{c,i} < b_{\text{used}}$  then
10            $\lfloor$   $b_{\text{used}} \leftarrow b_{c,i}$  and  $c_{\text{used}} \leftarrow c$ ;
11
12   $b_{\text{last}} \leftarrow -\infty$ ;
13  for  $c = 1$  to  $\tilde{\xi}$  do
14    foreach  $k_i \in \hat{\mathcal{K}}$  do
15      if  $b_{c,i} > R_{\text{max}} \times G$  or  $(b_{c,i} \leq R_{\text{max}}$  and  $\hat{\mathcal{K}} = \hat{\mathcal{K}}_{\text{DL}}$ )
16        then
17           $\lfloor$  continue;
18      if  $\sum_{d \in \hat{\mathcal{D}}_{\text{UL}} \cup \hat{\mathcal{D}}_{\text{DL}}} \lambda(d, c, P) < 1$  and
19         $\sum_{d \in \hat{\mathcal{D}}} S_{d, b_{c,i}, P} < F$  then
20          if  $b_{c,i} > b_{\text{last}}$  and  $c \neq c_{\text{used}}$  then
21            $\lfloor$   $b_{\text{last}} \leftarrow b_{c,i}$ ;
22
23  return  $b_{\text{used}}, b_{\text{last}}, c_{\text{used}}$ ;

```

G) and 2) the SS is an NPDCCH subframe when considering DL (i.e., $b_{c,i} \leq R_{\text{max}}$ and $\hat{\mathcal{K}} = \hat{\mathcal{K}}_{\text{DL}}$). In both cases, we skip the SS. The if-condition in line 6 means that the current DCI c and SS $b_{c,i}$ are available. When this SS is earlier than the previous one (i.e., line 7), we replace the earliest SS b_{used} by $b_{c,i}$ and set the used DCI c_{used} to c , as shown in line 8.

In lines 9–16, we find the last SS b_{last} to be reserved from the useable SSs. The last SS b_{last} is initialized as $-\infty$ in line 9, and its setups are similar to finding the earliest SS in lines 2–8. The difference is line 15. When the SS is later than b_{last} (i.e., $b_{c,i} > b_{\text{last}}$) and the used DCI c_{used} is not DCI c (i.e., $c \neq c_{\text{used}}$), b_{last} is set to $b_{c,i}$ by line 16. Lastly, the procedure returns subframes b_{used} and b_{last} and the used DCI c_{used} .

4.2.2 Usage Procedure

This procedure estimates the number E_d of subframes that can be given to each machine $d \in \hat{\mathcal{D}}$ based on two SSs b_{used} and b_{last} . In line 2, we initialize three variables, where δ is the number of allocatable subframes, h is an index used in $\hat{\mathcal{I}}_{\text{RU}}^3$, and x points to the current subframe. Then, the while-loop in lines 3–13 tests to allocate subframes one by one to machine d using resource type u_d and MCS m_d until its demand is met (i.e., $\eta(I_h, m_d) \geq \psi_d$) or index h reaches the maximum value H . In the while-loop, when giving a subframe to machine d (i.e., subframe $b_{\text{used}} + x$), two cases must be excluded to ensure that this subframe is usable:

- Line 5: The subframe is behind the control period.
- Line 7: The subframe does not have enough subcarriers.

3. As mentioned in Section 3.2, there is a set of allocable numbers for a resource unit, as denoted by $\hat{\mathcal{I}}_{\text{RU}} = \{I_1, I_2, \dots, I_h, \dots, I_H\}$.

```

Procedure Usage ( $\hat{\mathcal{I}}_{\text{Rep}}, b_{\text{used}}, b_{\text{last}}$ ):
1  foreach  $d \in \hat{\mathcal{D}}$  do
2     $\delta \leftarrow 0, h \leftarrow 0, x \leftarrow -1$ ;
3    while  $\eta(I_h, m_d) < \psi_d$  and  $h < H$  do
4       $x \leftarrow x + 1$ ;
5      if  $b_{\text{used}} + x > R_{\text{max}} \times G$  then
6         $\lfloor$  break;
7      else if  $\sum_{d \in \hat{\mathcal{D}}} S_{d, (b_{\text{used}} + x), P} + f_{u_d} > F$  then
8         $\lfloor$  break;
9      if  $b_{\text{last}} = b_{\text{used}} + x$  and  $R_{\text{max}} \times G - b_{\text{last}} >$ 
10         $(I_H - I_h) \times N_{d, u_d, m_d}$  then
11         $\lfloor$  break;
12       $\delta \leftarrow \delta + 1$ ;
13      if  $\delta = I_{h+1} \times N_{d, u_d, m_d} \times t_{u_d}$  then
14         $\lfloor$   $h \leftarrow h + 1$ ;
15       $E_d \leftarrow I_h \times N_{d, u_d, m_d} \times t_{u_d}$ ;
16  return  $E_d$ ;

```

In the two cases, we break the while-loop (i.e., lines 6 and 8) to finish the estimation. As shown in line 14, the number of allocatable subframes for machine d is $I_h \times N_{d, u_d, m_d} \times t_{u_d}$.

When subframe $b_{\text{used}} + x$ is usable, we further check if the subframe should be reserved. Specifically, if this subframe is the last SS (i.e., $b_{\text{last}} = b_{\text{used}} + x$) and machine d 's potential need is less than the remaining subframes (i.e., $R_{\text{max}} \times G - b_{\text{last}} > (I_H - I_h) \times N_{d, u_d, m_d}$), the last SS is worth keeping. Hence, we break the while-loop to reserve subframe $b_{\text{used}} + x$. The code is presented in lines 9–10. Otherwise, machine d can use this subframe, so we add δ by one in line 11. If the number of allocated subframes (i.e., δ) meets the transmission reliability that uses N_{d, u_d, m_d} repetitions (with resource type u_d and MCS m_d) and I_{h+1} resource units, the index h is increased by one, as shown in lines 12–13.

After we estimate the number of allocatable subframes for all machines in $\hat{\mathcal{D}}$, the procedure returns the E_d value for each machine, as indicated in line 15.

4.2.3 Machine Procedure

It chooses machines from $\hat{\mathcal{D}}$ to be served, and the selected machines are included in a set $\hat{\mathcal{D}}'$. Since the resource requirement of the DCI format for the UL is half that for the DL, two UL machines or only one DL machine can use the same NPDCCH subframes. Thus, $\hat{\mathcal{D}}'$ may contain one DL/UL machine or two UL machines. Then, we consider three cases to select machines, depending on the number of available DCIs in the current control period.

Case 1 (lines 2–12): Only one DCI remains. Since just one more machine can be scheduled, we should carefully select a machine to utilize the residual resources. To do so, we check if there is a machine d such that the number of its allocable subframes is exactly equal to the number of residual subframes (i.e., $E_d = R_{\text{max}} \times G - b_{\text{last}}$). If so, machine d can fully utilize the remaining shared subframes of the current control period in the time domain. The code is given in lines 3–6. However, if no such machine can be found (i.e., $\hat{\mathcal{D}}' = \emptyset$ in line 7), we sort machines in $\hat{\mathcal{D}}$ decreasingly based on their demands. Among machines whose demands have not been met (i.e., $\psi_d > 0$) and who are not scheduled yet (i.e., $\lambda(d, c, P) = 0, \forall c$), we

```

Procedure Machine ( $b_{\text{used}}, b_{\text{last}}, E_d$ ):
1   $\hat{\mathcal{D}}' \leftarrow \emptyset$ ;
2  if Only one DCI is left then
3    foreach  $d \in \hat{\mathcal{D}}$  do
4      if  $E_d = R_{\text{max}} \times G - b_{\text{last}}$  then
5         $\hat{\mathcal{D}}' \leftarrow \hat{\mathcal{D}}' \cup \{d\}$ ;
6        break;
7    if  $\hat{\mathcal{D}}' = \emptyset$  then
8      Sort machines in  $\hat{\mathcal{D}}$  by their demands  $\psi_d$  in
      descending order;
9      for  $d \in \hat{\mathcal{D}}$  do
10       if  $\psi_d > 0$  and  $\lambda(d, c, P) = 0, \forall c$  then
11          $\hat{\mathcal{D}}' \leftarrow \hat{\mathcal{D}}' \cup \{d\}$ ;
12         break;
13  else if two DCIs are left and  $\hat{\mathcal{D}} = \hat{\mathcal{D}}_{\text{UL}}$  then
14    Use Eq. (9) to pick machines  $x$  and  $y$  from  $\hat{\mathcal{D}}$ ;
15     $\hat{\mathcal{D}}' \leftarrow \hat{\mathcal{D}}' \cup \{x\} \cup \{y\}$ ;
16  else
17    for  $d \in \hat{\mathcal{D}}$  do
18      if  $\psi_d > 0$  and  $\lambda(d, c, P) = 0, \forall c$  then
19         $\hat{\mathcal{D}}' \leftarrow \hat{\mathcal{D}}' \cup \{d\}$ ;
20        break;
21  return  $\hat{\mathcal{D}}'$ ;

```

select the machine with the highest demand. Doing so helps improve resource utilization.

Case 2 (lines 13–15): Two DCIs are left, and we schedule UL machines (*i.e.*, $\hat{\mathcal{D}} = \hat{\mathcal{D}}_{\text{UL}}$). From $\hat{\mathcal{D}}$, two machines, x and y , are selected using the following equation:

$$\arg \max_{\forall x, y \in \hat{\mathcal{D}}, x \neq y} \left\{ \frac{E_x f_{u_x} + E_y f_{u_y}}{(R_{\text{max}} \times G - b_{\text{used}}) \times F} \right\}, \quad (9)$$

where machines x and y require E_x and E_y subframes in the time domain and f_{u_x} and f_{u_y} subcarriers in the frequency domain (due to using resource types u_x and u_y , respectively). The meaning behind Eq. (9) is to find machines x and y to maximize resource utilization in the residual subframes.

Case 3 (lines 16–20): In this case, there are adequate DCIs (*i.e.*, more than two DCIs for the UL or more than one DCI for the DL). As all machines have to be given DCIs to fulfill their demands (referring to the objective in Section 3.2), how to select a machine to use the DCI would have little impact on performance. Hence, we pick a machine that has not been scheduled yet and whose demand has not been satisfied. Then, this machine is included in $\hat{\mathcal{D}}'$.

4.2.4 Allocation Procedure

The Allocation () procedure allots DCI c_{used} and subframes to each machine in $\hat{\mathcal{D}}'$. If we allocate UL resources, the DCI format needs only half of the resource in an NPDCCH subframe, so we set $\lambda(d, c_{\text{used}}, P) = \lambda(d, c_{\text{used}}, P) + 0.5$, as shown in lines 2–3. Otherwise (*i.e.*, the DL case), we increase $\lambda(d, c_{\text{used}}, P)$ by one because the DCI format requires the whole resource of an NPDCCH subframe. The code is presented in lines 4–5.

In lines 6–7, we give E_d continuous subframes with f_{u_d} subcarriers to the machine using resource type u_d . The number

```

Procedure Allocation ( $\hat{\mathcal{D}}', \hat{\mathcal{I}}_{\text{Rep}}, b_{\text{used}}, c_{\text{used}}, E_d$ ):
1  foreach  $d \in \hat{\mathcal{D}}'$  do
2    if  $\hat{\mathcal{D}} = \hat{\mathcal{D}}_{\text{UL}}$  then
3       $\lambda(d, c_{\text{used}}, P) \leftarrow \lambda(d, c_{\text{used}}, P) + 0.5$ ;
4    else
5       $\lambda(d, c_{\text{used}}, P) \leftarrow \lambda(d, c_{\text{used}}, P) + 1$ ;
6    for  $z = 0$  to  $E_d$  do
7       $S_{d, (b_{\text{used}}+z), P} \leftarrow S_{d, (b_{\text{used}}+z), P} + f_{u_d}$ ;
8       $I_{h,d} \leftarrow E_d / (N_{d, u_d, m_d} \times t_{u_d})$ ;
9       $\psi_d \leftarrow \psi_d - \eta(I_{h,d}, m_d)$ ;
10   return  $I_{h,d}, \lambda(d, c, P), S_{d,s,P}$ ;

```

of resource units given to the machine (*i.e.*, $I_{h,d}$ in line 8) is E_d divided by $N_{d, u_d, m_d} \times t_{u_d}$, where E_d is the number of allocable subframes for the machine, N_{d, u_d, m_d} is the machine's data RN (using resource type u_d and MCS m_d), and t_{u_d} is the number of subframes required by resource type u_d . Since the machine can transmit or receive $\eta(I_{h,d}, m_d)$ data bits, we thus deduct $\eta(I_{h,d}, m_d)$ from its demand ψ_d in line 9. Finally, the procedure returns three parameters $I_{h,d}$, $\lambda(d, c, P)$, and $S_{d,s,P}$ used for resource allocation in line 10.

4.3 Time Complexity Analysis

Suppose that a BS has to serve ζ_{UL} UL machines and ζ_{DL} DL machines at a given CE level (*i.e.*, $|\hat{\mathcal{D}}_{\text{UL}}| = \zeta_{\text{UL}}$ and $|\hat{\mathcal{D}}_{\text{DL}}| = \zeta_{\text{DL}}$). Theorem 1 analyzes the time complexity of the control period adaptation in Algorithm 1.

Theorem 1. Given U resource types and M MCSs, the worst-case time complexity of Algorithm 1 is $O(M(\zeta_{\text{UL}} \times U + \zeta_{\text{DL}}) + |\hat{\mathcal{V}}_{\text{R}}| \times |\hat{\mathcal{V}}_{\text{G}}|)$, where $\hat{\mathcal{V}}_{\text{R}}$ and $\hat{\mathcal{V}}_{\text{G}}$ are the sets of R_{max} and G values, respectively.

Proof: Line 1 requires $O(\zeta_{\text{UL}} \times UM)$ time to calculate NPUSCH's length (*i.e.*, $\mathcal{L}_{\text{Data}}$) and line 2 takes $O(\zeta_{\text{UL}} + \zeta_{\text{DL}})$ time to estimate NPDCCH's length (*i.e.*, \mathcal{L}_{DCI}). In the if-else statement of lines 3–11, line 5 takes $O(\zeta_{\text{DL}} \times M)$ time to update $\mathcal{L}_{\text{Data}}$, and others spend a constant time. Then, the double for-loop in lines 12–16 selects a pair of R_{max} and G values from $(|\hat{\mathcal{V}}_{\text{R}}| \times |\hat{\mathcal{V}}_{\text{G}}|)$ combinations. Hence, the total time complexity is $O(\zeta_{\text{UL}} \times UM) + O(\zeta_{\text{UL}} + \zeta_{\text{DL}}) + O(\zeta_{\text{DL}} \times M) + O(|\hat{\mathcal{V}}_{\text{R}}| \times |\hat{\mathcal{V}}_{\text{G}}|) = O(M(\zeta_{\text{UL}} \times U + \zeta_{\text{DL}}) + |\hat{\mathcal{V}}_{\text{R}}| \times |\hat{\mathcal{V}}_{\text{G}}|)$. \square

Lemmas 1–4 show the time complexities of four procedures. We assume that $|\hat{\mathcal{D}}| = \zeta_D$, where $\hat{\mathcal{D}}$ is the set of machines to be scheduled (referring to lines 9 and 11 in Algorithm 2).

Lemma 1. Given set $\hat{\mathcal{K}}$ and variable $\tilde{\xi}$, the DCI-Delay () procedure requires a time of $O(\tilde{\xi}|\hat{\mathcal{K}}| \times (\zeta_{\text{UL}} + \zeta_{\text{DL}}))$.

Proof: The first double for-loop in lines 2–8 is repeated no more than $\tilde{\xi}|\hat{\mathcal{K}}|$ times. Computing $\sum_{d \in \hat{\mathcal{D}}_{\text{UL}} \cup \hat{\mathcal{D}}_{\text{DL}}} \lambda(d, c, P)$ and $\sum_{d \in \hat{\mathcal{D}}} S_{d, b_{c,i}, P}$ in line 6 takes $O(\zeta_{\text{UL}} + \zeta_{\text{DL}})$ and $O(\zeta_D)$ time, respectively. As other statements take $O(1)$ time, this double for-loop requires a time of $\tilde{\xi}|\hat{\mathcal{K}}| \times (O(\zeta_{\text{UL}} + \zeta_{\text{DL}}) + O(\zeta_D))$. The second double for-loop (*i.e.*, lines 10–16) is similar to the first one. So, the time complexity of the DCI-Delay () procedure is $2\tilde{\xi}|\hat{\mathcal{K}}| \times (O(\zeta_{\text{UL}} + \zeta_{\text{DL}}) + O(\zeta_D))$. Since $\hat{\mathcal{D}} \subseteq \hat{\mathcal{D}}_{\text{UL}} \cup \hat{\mathcal{D}}_{\text{DL}}$, we have $\zeta_D \leq \zeta_{\text{UL}} + \zeta_{\text{DL}}$. Thus, the time complexity is simplified to $O(\tilde{\xi}|\hat{\mathcal{K}}| \times (\zeta_{\text{UL}} + \zeta_{\text{DL}}))$. \square

Lemma 2. Given H allocable numbers for a resource unit in $\hat{\mathcal{I}}_{RU}$, the $\text{Usage}()$ procedure takes a time of $O(H\zeta_D^2)$.

Proof: This procedure only has a for-loop that is repeated ζ_D times. Both lines 2 and 14 take $O(1)$ time. The while-loop in lines 3–13 is repeated at most H times (due to the condition $h < H$). In the while-loop, computing $\sum_{d \in \hat{\mathcal{D}}} S_{d, (b_{\text{used}} + x), P} + f_{u_d}$ in line 7 takes $O(\zeta_D)$ time and other statements require a constant time. Hence, the time complexity of the $\text{Usage}()$ procedure is $\zeta_D \times H \times O(\zeta_D) = O(H\zeta_D^2)$. \square

Lemma 3. The time complexity of the $\text{Machine}()$ procedure is $O(\zeta_D^2)$.

Proof: The $\text{Machine}()$ procedure is composed of three mutually exclusive parts (*i.e.*, only one part can be conducted). In part 1 (lines 2–12), the for-loop in lines 3–6 takes $O(\zeta_D)$ time. Sorting machines in $\hat{\mathcal{D}}$ by line 8 requires $O(\zeta_D \log_2 \zeta_D)$ time. The for-loop in lines 9–12 spends $O(\zeta_D)$ time. Hence, part 1 consumes a time of $O(\zeta_D) + O(\zeta_D \log_2 \zeta_D) + O(\zeta_D) = O(\zeta_D \log_2 \zeta_D)$. In part 2 (lines 13–15), we use Eq. (9) to pick two machines x and y from $\hat{\mathcal{D}}$. Thus, part 2 takes a time of $O(\frac{\zeta_D!}{(\zeta_D-2)!}) = O(\zeta_D^2)$. In part 3 (lines 16–20), the only for-loop takes $O(\zeta_D)$ time. To sum up, the total time complexity is $\max\{O(\zeta_D \log_2 \zeta_D), O(\zeta_D^2), O(\zeta_D)\} = O(\zeta_D^2)$. \square

Lemma 4. The $\text{Allocation}()$ procedure takes $O(\zeta_D E_d^{\max})$ time, where E_d^{\max} is the maximum number of subframes required by a machine.

Proof: The outer for-loop in lines 1–9 is run $|\hat{\mathcal{D}}'|$ times. The inner for-loop in lines 6–7 is repeated at most E_d^{\max} times. Since other statements take $O(1)$ time and $\hat{\mathcal{D}}' \subseteq \hat{\mathcal{D}}$, the time complexity is $O(\zeta_D E_d^{\max})$. \square

Theorem 2 analyzes the time complexity of the joint UL and DL resource allocation in Algorithm 2. For simplification, we set $\zeta = \zeta_{UL} + \zeta_{DL}$ and assume that $UM < \zeta^2$, where U and M are the numbers of resource types and MCSs, respectively.

Theorem 2. In the worst case, Algorithm 2 has time complexity of $O(\zeta^2 \xi^2 |\hat{\mathcal{K}}| + H\zeta^3 \xi)$.

Proof: Two for-loops in lines 1–2 and 3–4 take $O(UM)$ and $O(M)$ time. Since there are ζ machines, the while-loop in lines 6–16 is repeated $O(\zeta)$ times. The code in lines 7–11 takes $O(1)$ time. The for-loop in lines 12–16 is repeated at most 2ξ times (due to $\hat{\xi} = 2\xi$ in the UL case). Based on Lemmas 1, 2, 3, and 4, the $\text{DCI-Delay}()$, $\text{Usage}()$, $\text{Machine}()$, and $\text{Allocation}()$ procedures in lines 13, 14, 15, and 16 take a time of $O(2\xi |\hat{\mathcal{K}}| \times \zeta)$, $O(H\zeta_D^2)$, $O(\zeta_D^2)$, and $O(\zeta_D E_d^{\max})$, respectively. Therefore, the total time complexity is $O(UM) + O(M) + O(\zeta) \times 2\xi \times (O(2\xi |\hat{\mathcal{K}}| \times \zeta) + O(H\zeta_D^2) + O(\zeta_D^2) + O(\zeta_D E_d^{\max}))$. Since $UM < \zeta^2$, $E_d^{\max} < H\zeta_D$, and $\zeta_D < \zeta$, the time complexity is simplified to $O(\zeta^2 \xi^2 |\hat{\mathcal{K}}| + H\zeta^3 \xi)$. \square

4.4 Discussion

IoT devices are usually battery-powered, so long battery life is one important feature for NB-IoT machines. Our proposed design can efficiently decrease the number of consumed subframes as well as increase resource utilization. This is especially helpful for saving the amount of time taken by machines to finish their data requests. In other words, Algorithms 1 and 2 take account of the feature of long battery life and are capable of providing energy efficiency for machines.

Fig. 3 illustrates how to implement the proposed algorithms in practical NB-IoT networks. More concretely, Algorithm 1 is implemented in the BS's *radio resource control* (RRC)

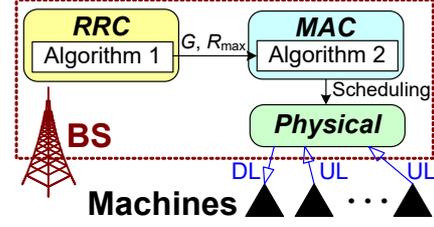


Fig. 3: Illustration for implementation of the proposed algorithms.

TABLE 3: Parameter settings in the simulation

Parameter	Value
Bandwidth	180 kHz
IoT machines	500, 1000, 1500, 2000, 2500, 3000
Data demand	1–200 bytes
Transmitted power	BS: 32 dBm, IoT machine: 23 dBm
Path loss	$120.9 + 30.76 \log_{10}(r)$ dB
Shadowing fading	Zero-mean log-normal distribution
Thermal noise density	-174 dBm/Hz
Resource units	$\hat{\mathcal{I}}_{RU}$: 1, 2, 3, 4, 5, 6, 8, 10
RNs	$\hat{\mathcal{I}}_{Rep}^{DCI}$: $R_{\max}/8, R_{\max}/4, R_{\max}/2$ $\hat{\mathcal{I}}_{Rep}^{UL}$: 1, 2, 4, 8, 16, 32, 64, 128 $\hat{\mathcal{I}}_{Rep}^{DL}$: 1, 2, 4, 8, 16, 32, 64, 128, 192, 256, 384, 512, 768, 1024, 1536, 2048
Scheduling delays	$\hat{\mathcal{K}}_{UL}$: 8, 16, 32, 64 $\hat{\mathcal{K}}_{DL}$ ($R_{\max} < 128$): 0, 4, 8, 12, 16, 32, 64, 128 $\hat{\mathcal{K}}_{DL}$ ($R_{\max} \geq 128$): 0, 16, 32, 64, 128, 256, 512, 1024
NPRACH subcarriers	CE 0: 48, CE 1: 24, CE 2: 12
NPRACH repetitions	CE 0: 1, CE 1: 16, CE 2: 64
NPRACH periodicity	CE 0: 320 ms, CE 1: 1280 ms, CE 2: 2560 ms

layer to determine control periods (*i.e.*, G and R_{\max}) when common search spaces are used [22]. The two parameters are sent to Algorithm 2 and delivered via the system information block messages to machines. Note that Algorithm 1 is implemented in the BS's *medium access control* (MAC) layer when specific search spaces are used. In this case, machines are notified of control period parameters via a random access procedure. On the other hand, Algorithm 2 performs in the MAC layer, and its outputs are passed to the physical layer.

5 PERFORMANCE EVALUATION

5.1 Simulation Setups

For performance evaluation, we develop a simulation using MATLAB. Table 3 summarizes the settings of our simulation parameters, which follow 3GPP specifications [22], [23]. Our simulation takes three CE levels into account. A BS supplies NB-IoT machines with UL or DL data requests at each CE level, whose bandwidth is 180 kHz. For each CE level, the number of machines increases from 500 to 3000. We consider three scenarios. In the *UL scenario*, all machines generate UL requests. For the *DL scenario*, machines produce DL requests. In the *hybrid scenario*, each machine randomly selects a UL or DL request (using a uniform distribution). The demand for each data request (*i.e.*, ψ_d) is set to [1, 200] bytes.

The transmitted power of the BS and a machine is set to 32 dBm and 23 dBm (around 1.58 W and 0.2 W, respectively). Regarding wireless transmissions, the amount of path loss (in dB) is calculated by $120.9 + 30.76 \log_{10}(r)$, where r denotes the distance between the BS and a machine in kilometers. To

TABLE 4: TBS table

m	I_h								
	1	2	3	4	5	6	8	10	
1	16	32	56	88	120	152	208	256	
2	24	56	88	144	176	208	256	344	
3	32	72	144	176	208	256	328	424	
4	40	104	176	208	256	328	440	568	
5	56	120	208	256	328	408	552	680	
6	72	144	224	328	424	504	680	872	
7	88	176	256	392	504	600	808	1032	
8	104	224	328	472	584	680	968	1224	
9	120	256	392	536	680	808	1096	1352	
10	136	296	456	616	776	936	1256	1544	
11	144	328	504	680	872	1032	1384	1736	
12	176	376	584	776	1000	1192	1608	2024	
13	208	440	680	1000	1128	1352	1800	2280	
14	224	488	744	1032	1256	1544	2024	2536	

model the shadowing fading, we use a zero-mean log-normal distribution with a standard deviation of 10 dBm. Besides, the spectral density of the thermal noise is -174 dBm/Hz.

For CE levels 0, 1, and 2, the distance of each machine from the BS (in meters) is between [1, 6000], [6001, 11000], and [11001, 14500], respectively. The SNR of a machine depends on the resource type, channel model, and distance r . With the SNR, we estimate the DCI and data RNs required by the machine. The set of allocable numbers for a resource unit is $\hat{I}_{RU} = \{1, 2, 3, 4, 5, 6, 8, 10\}$ [23]. Using the TBS table in Table 4, we get the number of data bits based on the assignment of resource units (*i.e.*, I_h) and MCS (*i.e.*, m). The RNs used to send a DCI at CE levels 0, 1, and 2 are set to $R_{max}/8$, $R_{max}/4$, and $R_{max}/2$, respectively. The other parameters regarding UL and DL RNs, scheduling delays, and NPRACH settings (*e.g.*, subcarriers, repetitions, and periodicity) are given in Table 3.

We name our proposed algorithms in Section 4 the *control period adaptation and resource allocation (CPARA)* scheme. As discussed in Section 2, existing solutions consider either UL or DL communications⁴. Therefore, regarding the UL scenario, we compare CPARA with two methods. The *UE-specific UL scheduler (UUS)* method [6] gives options to the scheduler for allotting DCIs and subcarriers. The *link adaptation and UL resource allocation (LAURA)* method [7] takes account of NPRACHs in link adaptation and resource allocation for UL communications to reduce subframe consumption. Since UUS does not consider control periods, the control periods used for USS are set based on [7]. For the DL scenario, two methods are compared. The *Max-Ri relaxed (MRR)* method [8] considers the rationale of search space allocation for NPDCCH when giving DL resources to machines. The *NPDCCH period adaptation and NB-IoT scheduling (NANIS)* method [9] copes with both period adaptation and DL scheduling for resource allocation. As for the hybrid scenario, we combine LAURA (in the UL) and NANIS (in the DL), called *LAURA plus NANIS*, and compare our CPARA scheme with this method.

5.2 Comparison of Consumed Subframes

Fig. 4 shows the number of subframes consumed using each method at CE level 0 with different numbers of machines. Recall that our goal in Section 3.2 is to use as few subframes as possible to fulfill the UL and DL demands of all machines.

4. The work [20] allots UL and DL resources to NB-IoT machines based on the NPDCCH offset mechanism. As the offset mechanism is not used in our work and other methods, we do not take the method in [20] for comparison.

Evidently, the number of consumed subframes increases as the number of machines grows because the BS needs to use more subframes to serve more data requests from machines.

In the UL scenario, as Fig. 4(a) shows, our CPARA scheme significantly reduces the number of subframes used to satisfy data requests compared to UUS and LAURA. The reason is that CPARA considers the control period adaptation, while the other two methods cannot flexibly adjust control periods based on the CE level. As can be seen, LAURA outperforms UUS, and CPARA further reduces subframe consumption by about 15% compared to LAURA.

Fig. 4(b) presents the experimental result in the DL scenario. Without adjusting control periods, MRR consumes far more subframes than NANIS and CPARA, indicating the importance of control period adaptation. Since machines have relatively high channel qualities, the control period adaptation in Algorithm 1 sets the scale factor to one for CE level 0. In this case, the control periods used in CPARA and NANIS are similar. That is why these two methods have similar performance.

Because LAURA outperforms UUS in the UL scenario and NANIS outperforms MRR in the DL scenario, we combine both LAURA and NANIS in the hybrid scenario. As shown in Fig. 4(c), when the BS simultaneously performs UL and DL communications, our proposed CPARA scheme can use fewer subframes than LAURA plus NANIS by approximately 7.5%.

Then, Fig. 5 reveals the impact of the number of machines on consumed subframes at CE level 1. In the UL scenario, as shown in Fig. 5(a), our CPARA scheme can save up to 49% of the subframes used for UL communications compared to LAURA. In NB-IoT, a BS provides four resource types for the UL. When channel qualities become worse, a machine requires a robust resource type with a longer time duration and a longer NPUSCH. Thus, the scale factor is set to two in Algorithm 1 (*i.e.*, line 9). Doing so helps reduce the number of subframes used by CPARA in the UL scenario.

Nevertheless, long control periods could degrade DL performance, as only one resource type with 12 subcarriers and one subframe is supported for the DL. In other words, there is a tradeoff between the needs of UL and DL in deciding control periods. As Fig. 5(b) shows, CPARA requires more subframes than NANIS in the DL scenario because NANIS is designed for the DL, while CPARA takes care of both UL and DL. In fact, the performance gap between CPARA and NANIS is not large. This verifies that our CPARA scheme can still perform efficiently at CE level 1 in the DL scenario.

Fig. 5(c) gives the number of subframes used in the hybrid scenario at CE level 1. Since CPARA flexibly adjusts control periods via Algorithm 1 and distributes UL and DL resources among machines using Algorithm 2, it economizes on subframes. Compared to LAURA plus NANIS, CPARA can save subframes to serve all machines by about 21%.

Fig. 6 presents subframe consumption at CE level 2. In the UL scenario, the trend in Fig. 6(a) is similar to that in Fig. 4(a) and Fig. 5(a). Regarding the DL scenario in Fig. 6(b), CPARA significantly outperforms NANIS. Since machines require high RNs to ensure transmission reliability at CE level 2, the DL needs long control periods. NANIS decides NPDSCH's length based on the maximum data RN of machines. Thus, the control periods found in NANIS may not be long enough, and many subframes in NPDSCH cannot be used. CPARA conquers this problem by adjusting control periods, so it reduces up to 50% of subframe consumption compared to NANIS. For the hybrid

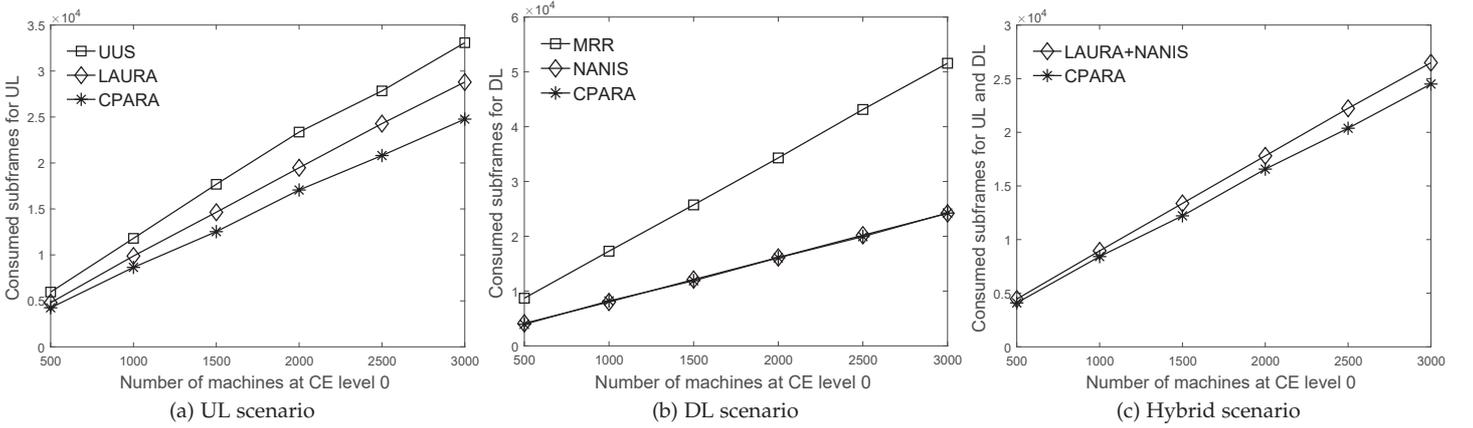


Fig. 4: Effect of the number of machines on the number of consumed subframes at CE level 0.

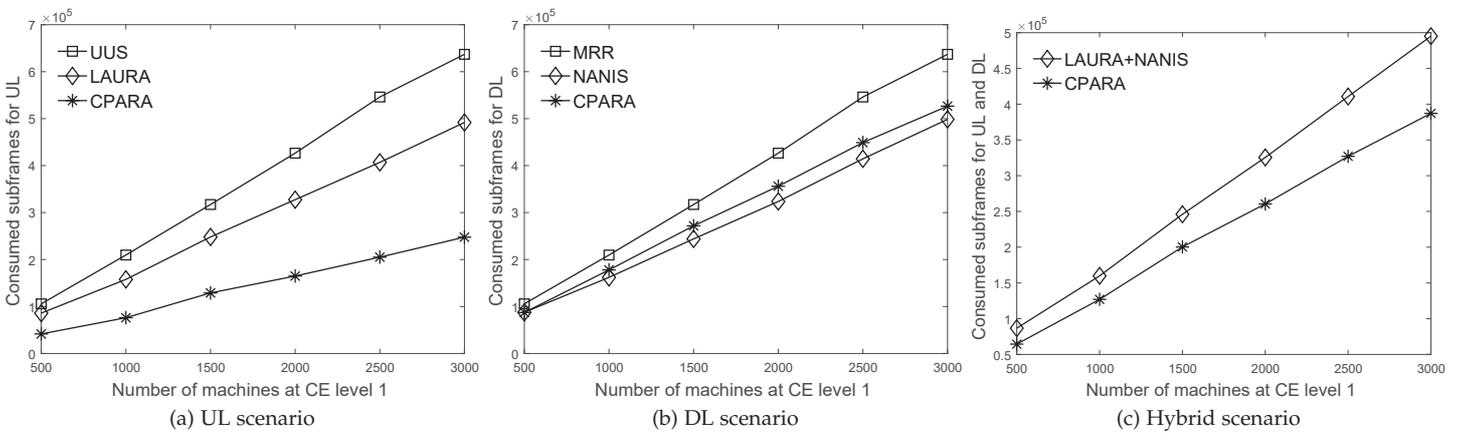


Fig. 5: Effect of the number of machines on the number of consumed subframes at CE level 1.

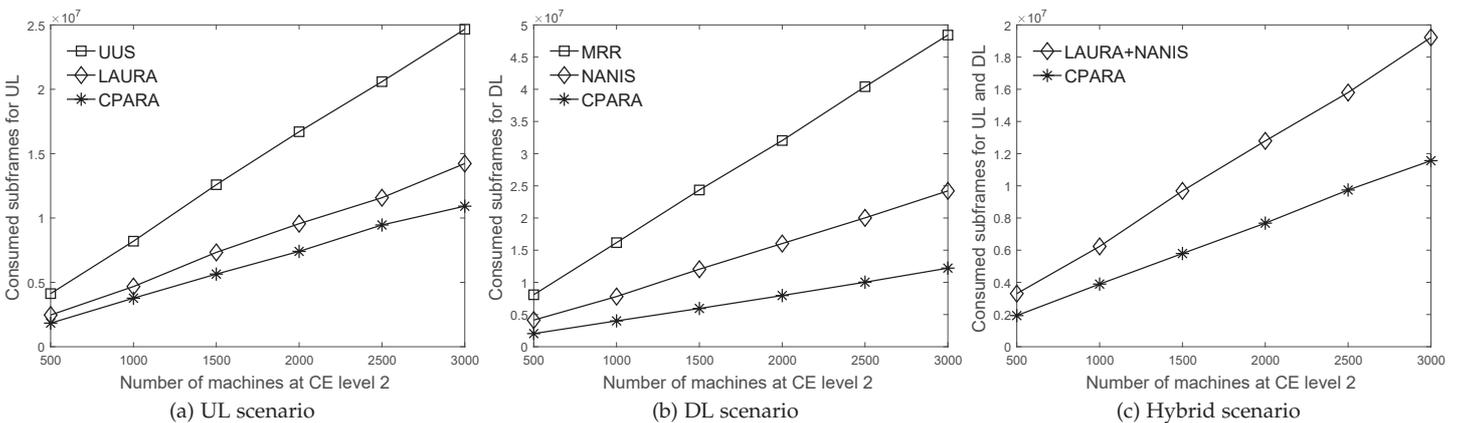


Fig. 6: Effect of the number of machines on the number of consumed subframes at CE level 2.

scenario in Fig. 6(c), CPARA decreases subframe consumption by about 40% compared to LAURA plus NANIS. The above results demonstrate the effectiveness of our CPARA scheme in terms of saving subframes.

5.3 Comparison of Resource Utilization

Fig. 7 compares the resource utilization of different channels (*i.e.*, NPDCCH, NPUSCH, and NPDSCH) using each method at different CE levels. For UL communications in Fig. 7(a), NPDCCH's utilization in UUS is 100% at each CE level, since

UUS attempts to maximize machines scheduled in NPDCCH. However, without appropriately setting the G parameter, UUS reduces NPUSCH's utilization as machines incur bad channel qualities at high CE levels. With the control period adaptation by Algorithm 1, our CPARA scheme can improve NPUSCH's utilization, especially at CE levels 1 and 2.

For DL communications in Fig. 7(b), MRR generally has the highest NPDCCH's utilization but the lowest NPDSCH's utilization because MRR is designed for increasing the control channel's utilization without considering NPDSCH's utiliza-

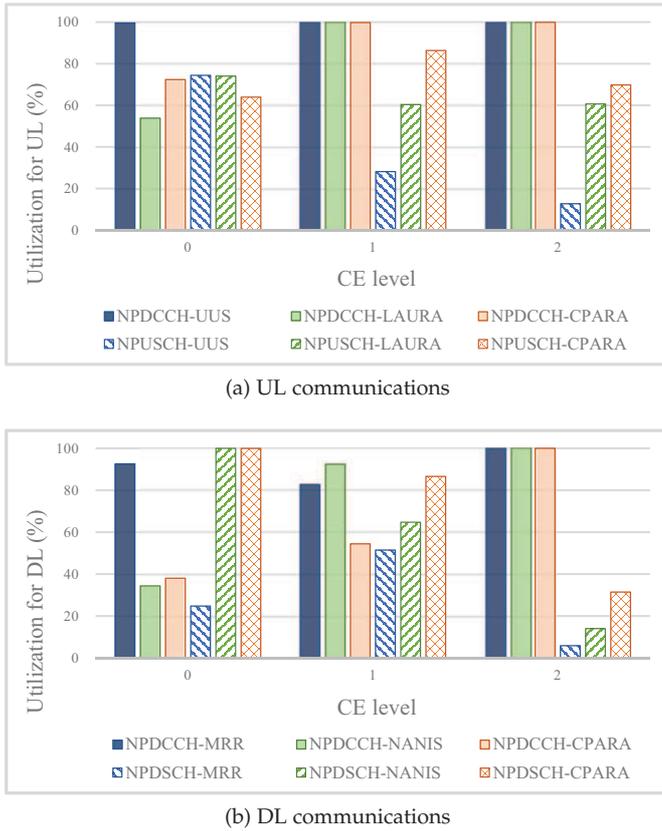


Fig. 7: Resource utilization of different channels at each CE level.

tion. As mentioned in Section 5.2, NANIS may waste many subframes in NPDSCH at CE level 2. That is why NPDSCH's utilization at CE level 2 is pretty low in NANIS. Compared to MRR and NANIS, our CPARA scheme can always maintain high NPDSCH's utilization.

6 CONCLUSION AND FUTURE WORK

In this article, we investigate the control period adaptation and resource allocation problems that consider joint UL and DL communications in NB-IoT networks, where the objective is to use the minimum number of subframes to serve requests from machines. We point out that UL and DL require different control period lengths to optimize performance, but they have to share the same control periods. To address this dilemma, we propose the control period adaptation algorithm to flexibly adjust control periods using a scale factor according to the CE level. Then, the joint UL and DL resource allocation algorithm properly distributes radio resources among machines. Simulation results reveal that our proposed algorithms not only reduce subframe consumption but also improve resource utilization in different scenarios compared to the UUS, LAURA, MRR, NANIS, and LAURA plus NANIS methods.

Though the demands of all machines can be satisfied, some machines may have significantly longer data latencies than others. Hence, we will further consider the transmission fairness among NB-IoT machines as future work. More concretely, let us consider that machines belong to different groups (*e.g.*, owned by different companies). To achieve fairness, we should minimize the difference between the average data latency of machines in each group. Doing so may lead to long control periods and degrade performance. Therefore, it is a challenge to strike a balance between performance and fairness.

REFERENCES

- [1] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio access capabilities," TS 36.306 V17.4.0, Mar. 2023.
- [2] Y.L. Lee, D. Qin, L.C. Wang, and G.H. Sim, "6G massive radio access networks: key applications, requirements and challenges," *IEEE Open J. Vehicular Technology*, vol. 2, pp. 54–66, 2021.
- [3] C.W. Huang, S.C. Tseng, P. Lin, and Y. Kawamoto, "Radio resource scheduling for narrowband Internet of Things systems: a performance study," *IEEE Network*, vol. 33, no. 3, pp. 108–115, 2019.
- [4] M. Kanj, V. Savaux, and M.L. Guen, "A tutorial on NB-IoT physical layer design," *IEEE Comm. Surveys & Tutorials*, vol. 22, no. 4, pp. 2408–2446, 2020.
- [5] Y.C. Wang and C.W. Chou, "Efficient coordination of radio frames to mitigate cross-link interference in 5G D-TDD systems," *Computer Networks*, vol. 232, pp. 1–13, 2023.
- [6] B.Z. Hsieh, Y.H. Chao, R.G. Cheng, and N. Nikaein, "Design of a UE-specific uplink scheduler for narrowband Internet-of-Things (NB-IoT) systems," *Proc. Int'l Conf. Intelligent Green Building and Smart Grid*, 2018, pp. 1–5.
- [7] Y.J. Yu and J.K. Wang, "NPRACH-aware link adaptation and uplink resource allocation in NB-IoT cellular networks," *IEEE Trans. Vehicular Technology*, vol. 70, no. 5, pp. 4894–4906, 2021.
- [8] M.P. Reddy, G. Santosh, A. Kumar, and K. Kuchi, "Downlink control channel scheduling for 3GPP narrowband-IoT," *Proc. IEEE Annual Int'l Symp. Personal, Indoor and Mobile Radio Communications*, 2018, pp. 1–7.
- [9] Y.J. Yu, "NPDCCH period adaptation and downlink scheduling for NB-IoT networks," *IEEE Internet of Things J.*, vol. 8, no. 2, pp. 962–975, 2021.
- [10] J.M. Liang, K.R. Wu, J.J. Chen, P.Y. Liu, and Y.C. Tseng, "Energy-efficient uplink resource units scheduling for ultra-reliable communications in NB-IoT networks," *Wireless Comm. and Mobile Computing*, vol. 2018, pp. 1–17, 2018.
- [11] C. Yu, L. Yu, Y. Wu, Y. He, and Q. Lu, "Uplink scheduling and link adaptation for narrowband Internet of Things systems," *IEEE Access*, vol. 5, pp. 1724–1734, 2017.
- [12] S. Zhu, W. Wu, L. Feng, P. Zhao, F. Zhou, P. Yu, and W. Li, "Energy-efficient joint power control and resource allocation for cluster-based NB-IoT cellular networks," *Trans. Emerging Telecomm. Technologies*, vol. 30, no. 4, pp. 1–16, 2019.
- [13] X. Wang, Z. Li, X. Jin, H. Yu, and C. Liu, "An uplink resource scheduling scheme for narrowband Internet of Things in smart grid," *Proc. IEEE Int'l Conf. Electronics and Comm. Engineering*, 2020, pp. 56–60.
- [14] O. Elgarhy, L. Reggiani, H. Malik, M.M. Alam, and M.A. Imran, "Rate-latency optimization for NB-IoT with adaptive resource unit configuration in uplink transmission," *IEEE Systems J.*, vol. 15, no. 1, pp. 265–276, 2021.
- [15] Y.J. Yu, Y.H. Huang, and Y.Y. Shih, "Cross-cycled uplink resource allocation over NB-IoT," *Sensors*, vol. 21, no. 23, pp. 1–16, 2021.
- [16] O. Kodheli, N. Maturo, S. Chatzinotas, S. Andrenacci, and F. Zimmer, "NB-IoT via LEO satellites: an efficient resource allocation strategy for uplink data transmission," *IEEE Internet of Things J.*, vol. 9, no. 7, pp. 5094–5107, 2022.
- [17] J.J. Alcaraz, F. Losilla, and F.J. Gonzalez-Castano, "Transmission control in NB-IoT with model-based reinforcement learning," *IEEE Access*, vol. 11, pp. 57991–58005, 2023.
- [18] P.R. Manne, S. Ganji, A. Kumar, and K. Kuchi, "Scheduling and decoding of downlink control channel in 3GPP narrowband-IoT," *IEEE Access*, vol. 8, pp. 175612–175624, 2020.
- [19] S. Mishra, L. Salaun, J.M. Gorce, and C.S. Chen, "Maximizing downlink user connection density in NOMA-aided NB-IoT networks through a graph matching approach," *Proc. IEEE Vehicular Technology Conf.*, 2022, pp. 1–7.
- [20] Y.J. Yu and L.X. Li, "Offset-aware resource allocation in NB-IoT networks," *IEEE Internet of Things J.*, vol. 9, no. 23, pp. 23967–23980, 2022.
- [21] Y.C. Wang and C.A. Chuang, "Efficient eNB deployment strategy for heterogeneous cells in 4G LTE systems," *Computer Networks*, vol. 79, pp. 297–312, 2015.
- [22] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," TS 36.331 V17.6.0, Sept. 2023.
- [23] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," TS 36.213 V17.5.0, Mar. 2023.