Low-Overhead Joint Beam-Selection and Random-Access Schemes for Massive Internet-of-Things with Non-Uniform Channel and Load

Yihan Zou[†], Kwang Taik Kim[†], Xiaojun Lin[†], Mung Chiang[†], Zhi Ding[§], Risto Wichman[‡] and Jyri Hämäläinen[‡]
[†]School of ECE, Purdue University [§]ECE, UC Davis [‡]Electrical Eng., Aalto University
Email: {zou59, kimkt, linx, chiang}@purdue.edu, zding@ucdavis.edu, {risto.wichman, jyri.hamalainen}@aalto.fi

***Abstract*—We study low-overhead uplink multi-access algorithms for massive Internet-of-Things (IoT) that can exploit the MIMO performance gain. Although MIMO improves system capacity, it usually requires high overhead due to Channel State Information (CSI) feedback, which is unsuitable for IoT. Recently, a Pseudo-Random Beam-Forming (PRBF) scheme was proposed to exploit the MIMO performance gain for uplink IoT access with uniform channel and load, without collecting CSI at the BS. For non-uniform channel and load, new adaptive beam-selection and random-access algorithms are needed to efficiently utilize the system capacity with low overhead. Most existing algorithms for a related multi-channel scheduling problem require each node to at least know some information of the queue length of all contending nodes. In contrast, we propose a new Low-overhead Multi-Channel Joint Channel-Assignment and Random-Access (L-MC-JCARA) algorithm that reduces the overhead to be independent of the number of interfering nodes. A key novelty is to let the BS estimate the total backlog in each contention group by only observing the random-access events, so that no queue-length feedback is needed from IoT devices. We prove that L-MC-JCARA can achieve at least 0.24 of the capacity region of the optimal centralized scheduler for the corresponding multi-channel system.**

***Index Terms*—machine-type communication, low overhead, provable stability, Lyapunov analysis.**

I. Introduction

Internet of Things (IoT) has been envisioned as a key application scenario in the upcoming 5G wireless network, which aims to interconnect a massive number of devices to support emerging applications such as e-health, smart home, and industrial internet [1]. However, IoT also poses new challenges to the network and communication protocols due to its unique features. First, unlike traditional data communication systems where most data transmission happens in the downlink (DL), in IoT a significant portion of the communication occurs in the uplink (UL). As a result, the massive number of devices poses an enormous challenge on how to coordinate UL data transmissions with a limited amount of spectral resources. Second, in most IoT applications, each device generates intermittent data with very short message payload. Thus, the traditional data communication protocols, e.g., 4G-LTE, are not suitable for IoT traffic due to expensive signaling overhead before data transmission. Therefore, it remains an open challenge to develop IoT UL communication protocols that are low-overhead and highly spectrum-efficient.

This work was supported in part by NSF grants CNS-1703014 and CNS-1702752, Defense Advanced Research Projects Agency (DARPA) under contract No. HR001117C0048, and Academy of Finland under grant 311752.

MIMO (Multi-Input Multi-Output) technology has been essential to achieve high spectrum efficiency in 4G cellular networks [2]. Unfortunately, centralized MIMO schemes incur high overhead due to the need of collecting Channel State Information (CSI) from all users [2–4]. Recently, we proposed in [5] an UL random-access protocol using Pseudo-Random Beamforming (PRBF) [6] to exploit the performance gain of MIMO with limited centralized control. The idea is to let BS use multiple receiving beams at each time following a pseudo-random sequence. Devices also know this pseudo-random sequence (by sharing a common seed to the random number generator). Thus, assuming that the device knows its own CSI (which incurs low overhead when the channel is static or changes very slowly), the device will be able to know the effective channel quality of each beam at each time, without further information exchange with the BS [6]. Then, each device uses a channel-aware random access protocol, which attempts transmission only when its effective channel on the intended beam is strong whereas its interference to other beams is weak. For an infinite-backlog system and under the assumption of uniform load and channel conditions, we showed that such a PRBF-based random access protocol with low signaling overhead can match the throughput of a centralized scheme in the order sense [5].

However, the design of the transmission scheme in [5] uses non-adaptive parameters that must be chosen beforehand based on the assumption of uniform load and channel statistics. In practice, both the network load and the channel statistics vary due to heterogeneous traffic patterns and non-uniform spatial distribution of the devices. As a result, the statistics of the effective channels and the level of contention seen by different groups of devices can be highly non-uniform. This setting then leads to an interesting joint beam-selection and random access problem. Ideally, we want each device to *adaptively* select beams and transmission probabilities based on its own load and channel conditions, as well as that of others. In reality, with the large number of devices and the lack of global knowledge, it becomes extremely difficult to design efficient algorithms with low overhead. In the literature, most of the adaptive scheduling algorithms that can be shown to achieve a provable fraction of the optimal capacity region (such as the Max-Weight policy [7–10] and other distributed approximations [11–14]) require each device to at least know some information about the queue-length of its interferers. When the number of interferers is large (as in massive IoT), even collecting this level of queue-length information would

have introduced significant overhead.

In this paper, we address this question by proposing a new beam-selection and random access scheme with low overhead that is independent of the number of interferering nodes, and show that it can still achieve a provable fraction of the optimal capacity region. We first map the multi-beam (MIMO) system to a (virtual) multi-channel system with non-uniform load and channel quality (see Section II-B), This mapping allows us to borrow ideas from the distributed channel-assignment and link scheduling algorithm of [14] (see Section III). However, as we mentioned earlier, the channel assignment component of the algorithm in [14] requires each device to know the sum of the queue length at its interfering nodes. Further, its link scheduling component requires computing a Maximal Schedule [15] at each time, whose complexity may also grow with the number of links. As a result, both components incur high overhead when the number of interfering nodes is large. Instead, our proposed algorithm in Section IV reduces both types of overhead to be independent of the number of interferers. Our key idea is to let the BS estimate, and then broadcast to devices, an approximate sum of the total backlog in each contention group. By replacing the Maximal Scheduling component of [14] with a random access scheme whose attempt probability depends on this estimate, the BS can then update this estimate by simply observing the idle, success, and collision events of each contention group. In this way, no explicit exchange of queue-length information is needed. We rigorously show that our proposed algorithm can achieve at least 0.24 fraction of the optimal capacity region of the (virtual) multi-channel system.

We note that this idea of estimating the system backlog based on random access events has been used for stabilizing ALOHA in [16] and [17]. However, the work there assumes either a single channel or a multi-channel system with homogeneous channels (and thus a uniformly-random channel-selection policy suffices). In contrast, our setting has heterogeneous channels, and therefore the random access component must be integrated with an adaptive beam/channel-selection component. Thus, the analysis of the joint control algorithm becomes much more difficult, and requires a new Lyapunov drift analysis (see Section IV). To the best of our knowledge, our work is the first to utilize such backlog estimation as a component in a larger joint control algorithm to achieve a provable fraction of the optimal capacity with low overhead.

Our work is related to the large literature of distributed scheduling algorithms for ad hoc wireless networks (e.g, [11], [13], [14], [18]). Among these algorithms, CSMA is shown to achieve full capacity and requires each link to only use its own queue length to decide the attempt probability [18]. However, due to its large mixing time, CSMA may suffer large delay that grows with the size of the network, which would be unsuitable for IoT with a massive number of devices. For most other algorithms that require the queue length information of at least the interfering nodes, one common way to further reduce the overhead is to perform "lazy update," i.e., to exchange queue length information infrequently [19]. However, in order to attain the same level of low overhead as our proposed algorithm, the update frequency would have to be inversely proportional to the number of interfering nodes, which will also likely lead to large delay. Our work uses a different way to reduce the overhead, i.e., via backlog estimates based on random access events. Thus, the techniques that we developed may be of independent interest to other related problems facing high overhead. Finally, although Sparse-Code Multiple Access (SCMA) [20] and other related random access algorithms have been studied for IoT [21–23], they do not exploit the MIMO gain, and thus are orthogonal to our work.

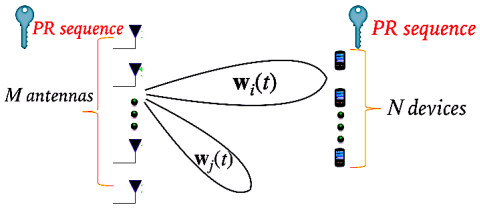

Fig. 1. The BS uses a PR sequence that is known at both the BS-side and the device-side to coordinate the beamforming vectors used at every time.

II. System Model

A. Pseudo-Random Beam-Forming (PRBF)

We consider a single-cell system (Fig. 1) where a base-station (BS) with M antennas serves N IoT devices, each of which has only a single antenna (in order to keep the cost of IoT devices low). We focus on the uplink, where the BS aims to decode the data transmissions from the IoT devices. Ideally, BS should be able to receive $\Theta(M)$ UL transmissions at the same time. The Pseudo-Random Beam-Forming (PRBF) scheme in [6] attains this goal without the need for the BS to acquire the UL Channel State Information (CSI). Assume that time is slotted. With PRBF, the BS uses a random beam pattern $\mathbf{W}(t) = [\vec{w}_1(t), ..., \vec{w}_M(t)]$ at time t. Each $\vec{w}_b(t) \in \mathcal{C}^M, b = 1, ..., M$, represents a receiving beamforming vector, and the BS uses M receive beamforming vectors simultaneously for decoding. Suppose that there are H possible beam patterns $\{\mathbf{W}^1, ..., \mathbf{W}^H\}$. At each time, the BS picks $\mathbf{W}(t)$ from one of the H beam patterns uniformly randomly, according to a pseudo random (PR) sequence. This PR sequence is also known to each device (by sharing a common seed to a random number generator) [6]. Thus, both the BS and the devices know all receive beamforming (BF) vectors used at all times, without further information exchange.

To use such a PRBF scheme, we assume that each device knows its own uplink CSI $\vec{g}_i \in \mathcal{C}^M$ to all M antennas, which can be acquired with low overhead if the CSI is symmetric between UL and DL, and is static or changes very slowly. Further, assume that the transmission power P_i of each device i is fixed. Suppose that the h-th beam pattern is chosen at time t, i.e., $\mathbf{W}(t) = \mathbf{W}^h = [\vec{w}_1^h, ..., \vec{w}_M^h]$. Each device i can then compute its effective channel gain g_{ib}^h to the BS on each beam $b = 1, ..., M$, i.e., $g_{ib}^h = |(\vec{g}_i)^T \vec{w}_b^h|^2$. Let B_i^h denote the beam with the highest effective channel gain g_{ib}^h for device i, among all beams b from the beam pattern $\mathbf{W}^h$. Intuitively,

the device i should use the best beam B_i^h for transmission, in the sense that it expects the BS to use the beam B_i^h to decode its transmission, and its interference signal that the BS receives on every other beam should be low. Correspondingly, we require that the device i may transmit under the beam pattern $\mathbf{W}^h$ only if its interference power to every beam other than B_i^h is below a threshold P_0. For convenience, in the rest of the paper we will say that the device transmits "using" beam B_i^h from the beam-pattern $\mathbf{W}^h$, even though it is actually the BS that uses the beam for decoding. For the beam-selection and random access scheme that we will develop in the rest of the paper, we expect that at each time the average number of transmitting devices using a particular beam to be around 1. Thus, assuming that the device i is the only device transmitting using beam B_i^h from beam-pattern $\mathbf{W}^h$, it can estimate its average transmission rate as Hr_i^h, with r_i^h given by

$$r_i^h = \log\left(1 + \frac{P_i g^h_{i,B_i^h}}{(M-1)P_0 + n_0}\right),$$

where n_0 is the background noise and a total bandwidth of H is assumed for simplicity. Note that at each time only one device should transmit using a given beam, because otherwise the BS will not be able to decode their signals. Further, note that the value of r_i^h only depends on h and i, and $r_i^h = 0$ if the device i is not allowed to transmit on beam pattern $\mathbf{W}^h$ (i.e., when its interference power to any beam other than B_i^h is above P_0).

For this paper, we assume that the set of beam-patterns and the transmission powers of all devices are given, and they are chosen in such a way that for each device there is at least one beam-pattern that it can transmit. We refer the readers to [5], [6], [24] for how such beam-patterns and power assignments can be chosen for different types of channel models.

B. Mapping to a Multi-Channel System

With the above system model, we can equivalently view the system as a (virtual) multi-channel system. There are H (virtual) channels. The h-th channel corresponds to beam pattern $\mathbf{W}^h$. At each time t, each channel h appears with probability $1/H$. For each beam b of beam-pattern $\mathbf{W}^h$, there may be many devices who can transmit using beam b. However, only one of them can transmit at a time. Thus, we can view this set of devices as a contention group in channel h. Note that by our setup, each device can only belong to one contention group in each channel h. Thus, we let $I^h(i)$ denote the contention group that device i belongs to in channel h, and $I^h(i) = \emptyset$ if device i is not allowed to transmit in channel h. Then, the transmission rate that a device i gets when channel h appears is simply Hr_i^h, assuming that no other devices in $I^h(i)$ transmit at the same time. Again, $r_i^h = 0$ if the device i is not allowed to transmit on channel h (i.e., when the beam-pattern is $\mathbf{W}^h$). By our definition, if $k \in I^h(i)$, we must also have $I^h(k) = I^h(i)$. We use $\mathcal{I}^h$ to denote all M contention groups in each channel h.

With the above mapping, the multi-channel system is completely specified by $I^h(i)$ and r_i^h for all i and h. Thus, in the rest of the paper, we will discuss the algorithm design based on this multi-channel model.

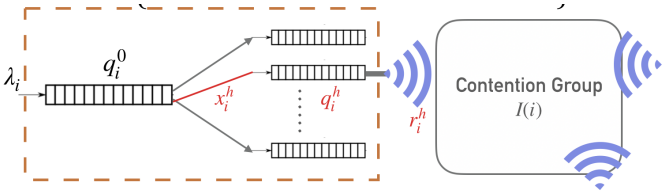

Fig. 2. Multi-channel virtual queues schematics.

C. Queue Dynamics and Capacity Region

Let $A_i(t)$ denote the number of packet arrivals at device i at time t. Throughout the paper, we assume that the arrivals are independent across different devices and *i.i.d.* in time. Denote the arrival rate vector $\vec{\lambda} = [\lambda_1, ..., \lambda_N]$ where $\lambda_i = \mathsf{E}[A_i(t)]$. Here, $\vec{\lambda}$ models the non-uniform loads of IoT devices. Suppose that the (virtual) channel h appears at time t. Let $D_i(t) = Hr_i^h$ if device i transmit successfully on channel h (i.e., $I^h(i) \neq \emptyset$ and no other devices in its contention group transmit at the same time), and $D_i(t) = 0$, otherwise. Let $Q_i(t)$ be the number of packets queued at device i at the beginning of time slot t. Then, the evolution of $Q_i(t)$ can be written as $Q_i(t+1) = [Q_i(t) + A_i(t) - D_i(t)]^+$, where $[\cdot]^+$ denote the projection function $\max(\cdot, 0)$. We say that the system is *stable* if the queue lengths at all devices remain finite [14], i.e.,

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\{\sum_{i=1}^N Q_i(t) > \eta\}} \to 0, \text{ a.s. as } \eta \to \infty.$$

We define the *capacity region* Ω under a particular policy as the set of $\vec{\lambda}$ such that the system remains stable. It is not difficult to show that, if a centralized scheduler knows the entire system state at all time and can schedule all transmissions without collision, its capacity region will be upper bounded by

$$\Omega_0 = \Big\{\vec{\lambda} \Big| \ \exists \lambda_i^h \text{ such that } \sum_{k\in I^h(i)} \frac{\lambda_k^h}{r_k^h} \leq 1 \text{ and } \sum_{h=1}^{H} \lambda_i^h = \lambda_i, \text{ for all } i, h\Big\}. \quad (1)$$

In this paper, we wish to design low-overhead control algorithms for massive IoT scenarios, and thus we cannot afford centralized control. The capacity region Ω that we can attain will then be a subset of Ω_0. An algorithm is said to be η-*optimal* if it can stabilize the system at any load $\vec{\lambda}$ that lies strictly inside $\eta\Omega_0$, where η is called the *efficiency ratio*. Our goal is then to design a low-overhead beam/channel-selection and multi-access algorithm with a provable efficiency ratio η.

III. A Joint Channel-Assignment and Scheduling Algorithm with Exact Congestion Information

The multi-channel model allows us to borrow ideas from the rich literature of scheduling algorithms for ad hoc wireless networks [8], [11], [13], [14]. In particular, although it is well-known that Maximum Weight Scheduling (MWS) can achieve

the full capacity region Ω_0, it requires centralized control and incurs high communication overhead due to the need to obtain a global "view" of the system at all times. As we discussed in the introduction, many distributed and lower-complexity scheduling algorithms have been developed to address the weakness of MWS. However, most of them require each node to at least know some information of the backlog of its interfering nodes, which also incur high overhead when the number of interfering nodes is large. Below, we will describe one particular algorithm that is of interest to us, which is the distributed channel-assignment and scheduling algorithm of [14]. It requires each node to know the sum of the backlog at interfereing nodes. We will then discuss the challenges to apply this algorithm to the massive IoT setting.

A. The Algorithm of [14]

For a multi-channel system with H channels, the algorithm of [14] lets each device maintain $H+1$ virtual queues. There is one virtual queue $q_i^h, h = 1, ..., H$ that corresponds to each of the H channels, plus another virtual queue q_i^0. As shown in Fig. 2, when incoming packets arrive at device i, they first enter virtual queue q_i^0. Then, a channel assignment algorithm determines how to route packets from q_i^0 to each per-channel virtual queue q_i^h. Once a packet is in the per-channel virtual queue q_i^h, it will only transmit on channel h. For each contention group I in channel h, define

$$N_I^h(t) = \sum_{k \in I} \frac{q_k^h(t)}{r_k^h}. \tag{2}$$

Thus, N_I^h can be viewed as the *congestion information*, which represents the current total backlog (normalized by each device's rate) in contention group I of channel h. Further, let α_i be an arbitrary positive constant chosen for device i. For our setting, [14] makes channel assignment and scheduling decisions as the MC-JCAS algorithm in Algorithm 1. For each device i, let $x_i^h(t)$ be the number of packets that are routed from q_i^0 to q_i^h at time t.

Algorithm 1: Multi-channel Joint Channel-Assignment and Scheduling (MC-JCAS)

1 For all channels h and for all devices i:
2 If $I^h(i) = \emptyset$, then $x_i^h(t) = 0$; Otherwise,
3 **if** $\frac{q_i^0(t) r_i^h}{\alpha_i} \geq N_{I^h(i)}^h \triangleq \sum_{k \in I^h(i)} \frac{q_k^h(t)}{r_k^h} \wedge q_i^0(t) \geq r_i^h$ **then**
4 $\quad x_i^h(t) = r_i^h$;
5 **else**
6 $\quad x_i^h(t) = 0$;
7 Suppose that channel $h(t)$ appears at time t. A maximal schedule [14] is computed for all $I \in \mathcal{I}^{h(t)}$.

Note that, in Line 2-6 of Algorithm 1, the decision is made by each device independently. Then, the virtual queue q_i^0 evolves as $q_i^0(t+1) = q_i^0(t) + A_i(t) - \sum_{h=1}^H x_i^h(t)$. In Line 7, a maximal schedule [14] is computed for all contention groups $I \in \mathcal{I}^{h(t)}$. In other words, for any contention group $I \in \mathcal{I}^{h(t)}$ such that at least one device in I has backlog $q_i^{h(t)}(t) \geq H r_i^{h(t)}$, exactly one of them will be scheduled to transmit. Suppose that this transmitting device in contention group I is i. Thus, we have $D_i^{h(t)}(t) = H r_i^{h(t)}$, and all other devices $i' \neq i$ in the same contention group I will have $D_{i'}^{h(t)}(t) = 0$. Then, all virtual queues q_i^h for channel $h(t)$ are updated by $q_i^{h(t)}(t+1) = q_i^{h(t)}(t) + x_i^{h(t)}(t) - D_i^{h(t)}(t)$, and all virtual queues for other channels $h'(t) \neq h(t)$ are updated by $q_i^{h'(t)}(t+1) = q_i^{h'(t)}(t) + x_i^{h'(t)}(t)$.

As in [14], the channel assignement decision in Line 2-6 of Algorithm 1 can be interpreted with the notion of "congestion costs." In particular, the quantity $\frac{q_i^0(t)}{\alpha_i}$ can be viewed as the *backlog cost* at device i. Meanwhile, the quantity $N_{I^h(i)}^h = \sum_{k \in I^h(i)} \frac{q_k^h(t)}{r_k^h}$ can be viewed as the congestion cost of channel h seen by device i, which is contributed by the entire contention group that device i belongs to. Therefore, device i will assign packets to q_i^h only if the backlog cost is higher than the congestion cost of channel h, normalized by the channel rate r_i^h. This normalization is the key: packets are more likely to be assigned to the per-channel queue q_i^h if the corresponding rate r_i^h is high. This design thus helps the devices to use channels that are good and that are less congested. Using the techniques of [14], it is not difficult to show that the above algorithm will achieve the full capacity region Ω_0 (i.e., with an efficiency ratio of 1).

B. The Challenge of Applying MC-JCAS to Massive IoT

Despite its distributed operation, MC-JCAS still requires each device to know the exact congestion information from its contention group at each time. To obtain this information, each device needs to report its current per-channel queues either to the BS or to each other. When the number of devices in a contention group is large, such reporting will incur high overhead. Further, distributed computation of the maximal schedule [15] may also incur overhead that grows with the number of interfering nodes. Therefore, this algorithm is not suitable for massive IoT scenarios where low overhead is required. As we discussed in the introduction, similar levels of overhead are also required by most other distributed scheduling algorithms with provable efficiency ratios. Standard approaches to address this issue include using CSMA [18] or "lazy update" [19], both of which tend to introduce large delay. In the next section, we will propose a new approach to address this overhead issue, which will reduce the overhead to be independent of the number of interfering nodes, and still attain a provable efficiency ratio.

IV. Low-Overhead Channel-Selection and Random Access Using Congestion Estimation

In this section, we propose a new channel/beam-selection and random access algorithm that eliminates both the need of reporting the queue length from the devices, and that of distributively computing the maximal schedules. As a result, our algorithm will reduce the overhead to be independent of the number of interfering nodes. Our algorithm uses two key

ideas. First, instead of having each device directly report its per-channel queues, we let the BS form an estimate of the congestion information $N_I^h(t)$ in each contention group I and each channel h, and update this estimate by observing the events that happen in the contention group. Let $S_I^h(t)$ denote this estimate for $N_I^h(t)$. Note that the BS can let all devices know these estimates via a simple broadcast. The devices can then make their channel-assignment decisions independently based on such estimates. Second, in order for the BS to update such estimates, it needs to have some observation as input. We thus let each device follows a random access policy whose attempt probability depends on the estimate $S_I^h(t)$. In this way, the channel events (i.e., idle, success, collision) become a function of both the true congestion information $N_I^h(t)$ and its estimate $S_I^h(t)$. The BS can then update the estimate $S_I^h(t)$ based *only* on observing the channel events, without additional overhead. Further, this random access policy eliminates the need to distributively compute a maximal schedule.

We note that the idea of stabilizing ALOHA based on backlog estimates have been proposed in [16] for a single-channel system. More recently, the work in [17] extends the idea to multi-channel systems where all channels are homogeneous. In contrast, our work is the first to use this idea in a setting where the channels are heterogeneous (as is the case with PRBF). For homogeneous channels, a simple uniformly-random channel assignment scheme suffices. Thus, one only needs to estimate one piece of congestion information for all channels together. In contrast, for heterogeneous channels, each channel (and each contention group) corresponds to one piece of congestion information. Further, the channel assignment algorithm also depends on the congestion estimates, which creates a complex coupling between the random access component and the channel assignment component. As a result, the analysis of the system dynamics becomes much more complicated than that of [16], [17]. In Section IV-D, we will develop a new Lyapunov drift analysis to address this difficulty. Next, we first present the random access policy, followed by the proposed algorithm and its analysis.

A. Low-overhead Queue-based Random Access Policy Using Congestion Estimate

In the L-QRA policy presented in Algorithm 2, each device i uses the congestion estimate $S^h_{I^h(i)}$ and its own backlog $q_i^h(t)$ to determine the transmission probability.

Consider one contention group I in channel h. Suppose that the true congestion information is N_I^h and its estimate is S_I^h. Let $\theta = N_I^h/S_I^h$. Under L-QRA, the idle probability that no devices transmit in contention-group I is

$$\Pr\{\mathbf{idle}(I,h)\} = \prod_{k\in I} p^h_{k,0} = e^{-N_I^h/S_I^h} = e^{-\theta}. \tag{3}$$

The success probability that exactly one device transmits a data message is

$$\Pr\{\mathbf{success}(I,h)\} = \sum_{k\in I} p^h_{k,1} \prod_{j\neq k} p^h_{j,0} = \frac{N_I^h}{S_I^h} e^{-\frac{N_I^h}{S_I^h}} = \theta e^{-\theta}, \tag{4}$$

Algorithm 2: Low-overhead Queue-based RA (L-QRA)

Suppose that channel h appears at time t. The BS broadcasts S_I^h for each contention group $I \in \mathcal{I}^h$;
For each device i:
if $q_i^h(t) < Hr_i^h$ *or* $I^h(i) = \emptyset$ **then**
  Stay silence;
else
  Set $n_i^h = q_i^h(t)/r_i^h$;
  switch $z \sim \mathcal{U}[0,1]$ **do**
    when $z \in [0, p^h_{i,0})$ where $p^h_{i,0} = e^{-n_i^h/S^h_{I^h(i)}}$, stay silence;
    when $z \in [p^h_{i,0}, p^h_{i,0} + p^h_{i,1})$ where $p^h_{i,1} = (n_i^h/S^h_{I^h(i)})e^{-n_i^h/S^h_{I^h(i)}}$, transmit Hr_i^h packets from the virtual queue $q_i^h(t)$;
    otherwise transmit a dummy collision signal with probability $p^h_{i,2} = 1 - p^h_{i,0} - p^h_{i,1}$;
  end
end

and the rest $1 - e^{-\theta} - \theta e^{-\theta}$ becomes the "collision" probability $\Pr\{\mathbf{collision}(I,h)\}$. Readers may notice that, with probability $p^h_{i,2}$, device i transmits a collision signal. Thus, even if this device is the only transmitter, the channel will be considered to have experienced a collision (and thus the event $\mathbf{collision}(I,h)$ occurs). This assumption simplifies our analysis since the above probabilities depend only on $\theta = N_I^h/S_I^h$, and not on the exact backlog of each device. On the other hand, this design will lead to some waste in the system capacity. When $\frac{n_i^h}{S^h_{I^h(i)}}$ is small, we can verify that $p^h_{i,1} = \Theta(n_i^h/S^h_{I^h(i)})$ and $p^h_{i,2} = O(n_i^h/S^h_{I^h(i)})$. Thus, we expect that the fake collision will play a small role when the number of interfering nodes is large, which will be confirmed by our simulation results in Section V.

If the congestion estimate is accurate, i.e., $\theta = N_I^h/S_I^h = 1$, then the expected number of transmissions in each contention group is $\sum_{k\in I}(1 - p^h_{i,0}) \leq \sum_{k\in I} n_i^h/S_I^h = 1$. Further, the success probability is $1/e$, which is also the optimal throughput for ALOHA [25]. Compared to maximal scheduling that schedules exactly one successful transmission in the contention group, we thus expect that the above random access policy will lead to a $1/e$ reduction in the capacity region. However, the congestion estimate is not always accurate. Therefore, we have to carefully account for the impact of the estimate errors on the system performance.

B. Updating the Congestion Estimates

We now describe how the BS updates the congestion estimates. Similar to [16], we assume that the BS can observe the channel events in each contention group I and each channel h. Indeed, for each contention group I in channel h, **idle** means that the BS does not detect any signal power using the corresponding beamforming vector; **success** means that the BS can successful decode a data message using the

corresponding beamforming vector; and **collision** means that, using the corresponding beamforming vector, the BS either can detect a signal but cannot decode it, or can decode a message but it contains a collision signal. Then, the BS updates the congestion estimate $S_I^h(t)$ by

$$S_I^h(t+1) = \max\{1, S_I^h(t) + \Delta S_I^h(t)\}, \tag{5}$$

$$\begin{aligned}\Delta S_I^h(t) &= a\mathbb{1}_{\mathbf{idle}(I,h)}(t) + b\mathbb{1}_{\mathbf{success}(I,h)}(t) \\ &\quad + c\mathbb{1}_{\mathbf{collision}(I,h)}(t).\end{aligned} \tag{6}$$

Note that *no additional overhead* to report the per-channel queues is needed. Intuitively, the parameter a should be negative so that $\Delta S_I^h(t) < 0$ whenever N_I^h is much smaller than S_I^h (and thus the channel is idle most of the time), and the parameter c should be negative so that $\Delta S_I^h(t) > 0$ whenever N_I^h is much larger than S_I^h (and thus the channel experiences collisions most of the time). We will give specific values for a, b and c below when we prove the efficiency ratio of the overall algorithm.

C. The Complete Algorithm and Main Result

Algorithm 3 describes the our proposed low-overhead joint channel/beam-selection and random access algorithm (L-MC-JCARA) with congestion estimation. Similar to MC-JCAS, here α_i is an arbitrary positive constant chosen for device i.

Algorithm 3: (L-MC-JCARA)

1 Suppose that channel (i.e., beam pattern) $h(t)$ appears at time t. The BS broadcasts $S_I^{h(t)}$ for all $I \in \mathcal{I}^{h(t)}$;
2 For all channels h and each device i, determine channel-assignment $x_i^h(t)$ for all h as follows. If $I^h(i) = \emptyset$, then $x_i^h(t) = 0$. Otherwise,
3 **if** $\frac{q_i^0(t) r_i^h}{\alpha_i} \geq 1.2 S^h_{I^h(i)}$ *and* $q_i^0(t) \geq r_i^h$ **then**
4 $\quad x_i^h(t) = r_i^h$;
5 **else**
6 $\quad x_i^h(t) = 0$;
7 **end**
8 For the current channel $h(t)$, access the channel according to L-QRA in Algorithm 2;
9 BS updates $S_I^{h(t)}$ from channel outcomes using (6).

Note that L-MC-JCARA differs from MC-JCAS in using the contention estimates and in using L-QRA instead of maximal scheduling. Further, there is an additional factor of 1.2 in the comparison step in line 3, which is needed for our analytical results. Our main result is as follows.

Theorem 1. *Suppose that BS chooses the parameters $a = lH(1-e)$ and $b = c = lH$ in* (6)*, where $l = 22K_{\max}$ and $K_{\max}$ is the maximum number of devices in any contention group I. Then, our proposed L-MC-JCARA scheme can stabilize any arrival rate vector $\vec{\lambda}$ that is strictly inside $\eta\Omega_0$, where $\eta = 0.24$.*

Theorem 1 shows that our proposed algorithm achieves an efficiency ratio of at least $\eta = 0.24$. The assumption of the theorem requires the parameters a, b, and c to grow linearly with the number of interfering nodes. While we need this assumption for our analysis, we have found in our simulation results that using values independent of the number of interfering nodes tends to produce an even smaller queue backlog. We thus conjecture that this assumption may be removed via finer analysis, which we leave for future work.

D. Sketch of Proof

In the rest of the section, we will sketch the main steps of the proof for Theorem 1, which is also our main contribution. As we discussed at the beginning of this section, our analysis is more difficult than that of [16], [17] because we consider heterogeneous channels and have to account for the complex interaction between the random access component and the channel-assignment component, both of which are coupled by the congestion estimates. Due to this reason, we cannot use the Lyapunov drift analysis from [16], [17].

In the following, we will construct a new Lyapunov function for the multi-channel system with congestion estimation, and study its expected drift. Let $\vec{q}(t)$ denote the system state at time t, which collects all virtual queues $q_i^h(t)$ and all congestion estimates $S_I^h(t)$. Our new Lyapunov function for the entire system with congestion estimation consists of three parts, i.e.,

$$V_{\text{tot}}(\vec{q}(t)) = V_0(\vec{q}(t)) + V_N(\vec{q}(t)) + V_{N,S}(\vec{q}(t)). \tag{7}$$

The first two parts are given as follows, and are similar to those used in the proof in [14]:

$$V_0(\vec{q}(t)) = \sum_{i=1}^{N} \frac{(q_i^0(t))^2}{2\alpha_i}, \quad \text{and} \tag{8}$$

$$V_N(\vec{q}(t)) = \sum_{h=1}^{H}\sum_{i=1}^{N} \frac{q_i^h(t)}{2r_i^h} \sum_{k\in I^h(i)} \frac{q_k^h(t)}{r_k^h} = \sum_{h=1}^{H}\sum_{I\in\mathcal{I}^h} \frac{N_I^h(t)^2}{2}, \tag{9}$$

where $N_I^h(t)$ is the congestion information defined in (2). The second equality of (9) holds because, in each channel h, $I^h(i) = I^h(k)$ for any $i, k \in I$. The third part of (7) is

$$V_{N,S}(\vec{q}(t)) = \sum_{h=1}^{H}\sum_{I\in\mathcal{I}^h} \frac{1}{2}\left(N_I^h(t) - S_I^h(t)\right)^2, \tag{10}$$

which captures the gap between the exact N_I^h and its estimate S_I^h. Although inspired by [16], (10) is different from the Lyapunov function used there, and is essential for analyzing the coupling with the channel-assignment component of our algorithm. Let $\Delta V_0(\vec{q}(t)) = V_0(\vec{q}(t+1)) - V_0(\vec{q}(t))$ denote the drift of $V_0(\vec{q}(t))$ and define other drifts analogously. Similar to [14], the expected drifts of (8) and (9) can be bounded by,

$$\mathsf{E}[\Delta V_0(\vec{q}(t))|\vec{q}(t)] \leq \sum_{i=1}^{N} \frac{q_i^0(t)}{\alpha_i}\left(\lambda_i - \sum_{h=1}^{H} x_i^h(t)\right) + C_1, \tag{11}$$

$$\mathsf{E}[\Delta V_N(\vec{q}(t))|\vec{q}(t)] \leq \sum_{h=1}^{H}\sum_{I\in\mathcal{I}^h} N_I^h(t)\overline{\Delta N_I^h(t)} + C_2, \tag{12}$$

where C_1 and C_2 are constants and $\overline{\Delta N_I^h(t)} = \mathsf{E}[N_I^h(t+1) - N_I^h(t)|\vec{q}(t)]$. From (4), we can show that

$$\overline{\Delta N_I^h(t)} = \sum_{k\in I} \frac{x_k^h(t)}{r_k^h} - \theta_I^h(t)\exp(-\theta_I^h(t)),$$

where $\theta_I^h(t) = \frac{N_I^h(t)}{S_I^h(t)}$. For (10), we can show that its expected drift is bounded by, for some constant C_3,

$$\begin{aligned}&\mathsf{E}[\Delta V_{N,S}(\vec{q}(t))|\vec{q}(t)] \qquad (13)\\ &\le \sum_{h=1}^{H}\sum_{I\in\mathcal{I}^h}\left(N_I^h(t) - S_I^h(t)\right)\left(\overline{\Delta N_I^h(t)} - \overline{\Delta S_I^h(t)}\right) + C_3,\end{aligned}$$

where $\overline{\Delta S_I^h(t)} = \frac{1}{H}\mathsf{E}[\Delta S_I^h(t)]$ and $1/H$ corresponds to the probability that channel h appears at time t. By the definition of $\Delta S_I^h(t)$ in (6), the probabilities (3)-(4), and the choices of a, b and c in the theorem, we have,

$$\begin{aligned}\overline{\Delta S_I^h(t)} &= \frac{1}{H}\left[c + (a-c)e^{-\theta_I^h(t)} + (b-c)\theta_I^h(t)e^{-\theta_I^h(t)}\right]\\ &= l(1 - e^{1-\theta_I^h(t)}). \qquad (14)\end{aligned}$$

Suppose $(1+\epsilon)\vec{\lambda} \in \eta\Omega_0$. By definition, there exist $\tilde{x}_i^h$ such that

$$(1+\epsilon)\lambda_i \le \sum_{h=1}^{H}\tilde{x}_i^h, \text{ and } \sum_{k\in I^h(i)}\frac{\tilde{x}_k^h}{r_k^h} \le \eta, \text{ for all } i \text{ and } h. \qquad (15)$$

Here, $\tilde{x}_i^h$ can be viewed as a reference for the desirable average number of packets routed to virtual queue q_i^h in the long run. Therefore, the total Lyapunov drift for the entire system is

$$\begin{aligned}&\mathsf{E}[\Delta V_{\text{tot}}(\vec{q}(t))|\vec{q}(t)]\\ &\le \sum_{i=1}^{N}\frac{q_i^0(t)}{\alpha_i}\left(\lambda_i - \sum_{h=1}^{H}x_i^h(t)\right) + \sum_{h=1}^{H}\sum_{I\in\mathcal{I}^h}N_I^h(t)\overline{\Delta N_I^h(t)}\\ &\quad + \sum_{h=1}^{H}\sum_{I\in\mathcal{I}^h}\left(N_I^h(t) - S_I^h(t)\right)\left(\overline{\Delta N_I^h(t)} - \overline{\Delta S_I^h(t)}\right) + C_4,\\ &\le \sum_{i=1}^{N}\frac{q_i^0(t)}{\alpha_i}\Big(\lambda_i - \sum_{h=1}^{H}\tilde{x}_i^h\Big) + \sum_{i=1}^{N}\frac{q_i^0(t)}{\alpha_i}\Big(\sum_{h=1}^{H}\left[\tilde{x}_i^h - x_i^h(t)\right]\Big)\\ &\quad + \sum_{h=1}^{H}\sum_{I\in\mathcal{I}^h}N_I^h(t)\overline{\Delta N_I^h(t)}\\ &\quad + \sum_{h=1}^{H}\sum_{I\in\mathcal{I}^h}\left(N_I^h(t) - S_I^h(t)\right)\left(\overline{\Delta N_I^h(t)} - \overline{\Delta S_I^h(t)}\right) + C_4\\ &\le \sum_{h=1}^{H}\sum_{I\in\mathcal{I}^h}\Bigg[\sum_{k\in I}\frac{q_k^0(t)}{\alpha_k}\left(\tilde{x}_k^h - x_k^h(t)\right) + N_I^h(t)\overline{\Delta N_I^h(t)}\\ &\quad + \left(N_I^h(t) - S_I^h(t)\right)\left(\overline{\Delta N_I^h(t)} - \overline{\Delta S_I^h(t)}\right)\Bigg] - \epsilon\sum_{i=1}^{N}\frac{q_i^0(t)\lambda_i}{\alpha_i}\\ &\quad + C_4, \qquad (16)\end{aligned}$$

where $C_4 = C_1 + C_2 + C_3$. Denote the term in the $[\cdot]$ of (16) as ΔV_I^h. The last inequality of (16) thus successfully decouples the total drift of the system into the sum of single-channel drifts ΔV_I^h across all contention groups I and channels h. To show that the total Lyapunov drift is negative, it is then *sufficient* to show that ΔV_I^h is non-postive for all I and h.

Per-channel One-contention-group Lyapunov Drift ΔV_I^h

For each channel h and each contention group $I \in \mathcal{I}^h$, we next show that the per-contention-group drift ΔV_I^h is non-positive. Note that the expected changes $\overline{\Delta N_I^h(t)}$ and $\overline{\Delta S_I^h(t)}$ depend on $\theta_I^h(t)$. We will divide into multiple cases with different ranges of $\theta_I^h(t)$. For ease of exposition, we will drop the subscript I and superscript h (i.e., use $N(t)$, $S(t)$ and $\theta(t)$ instead of $N_I^h(t)$, $S_I^h(t)$ and $\theta_I^h(t)$), whenever there is no source of confusion.

1) **When $0.9 \le \theta(t) \le 1.1$:** In this case, the values of $N(t)$ and $S(t)$ are close. Thus, we expect that (i) the success rate of the contention group will be close to $1/e$, and (ii) the system dynamics will be close to that of MC-JCAS (and that of [14]), which knows the true $N(t)$. we will make this intuition rigorous by considering the per-device drift under its different assignment decisions. Note that the drift ΔV_I^h in (16) can be written as

$$\begin{aligned}\Delta V_I^h &= \sum_{k\in I}\frac{q_k^0(t)}{\alpha_k}\left(\tilde{x}_k^h - x_k^h(t)\right) + N(t)\Delta N^*(t)\\ &\quad + \left(N(t) - S(t)\right)\left(\Delta N^*(t) - \overline{\Delta S(t)}\right) \qquad (17)\\ &\quad + \left(2N(t) - S(t)\right)\left(\overline{\Delta N(t)} - \Delta N^*(t)\right), \qquad (18)\end{aligned}$$

where we choose $\Delta N^*(t) = 0.9e^{-0.9} - \theta(t)e^{-\theta(t)}$, which is non-positive since $0.9 \le \theta(t) \le 1.1$. We thus have $N(t)\Delta N^*(t) \le 0$. Further, for (17), we have

$$\begin{aligned}\text{Eq. (17)} &= S(t)[\theta(t) - 1]\left[\frac{1}{e} - \theta(t)e^{-\theta(t)} - l(1 - e^{1-\theta(t)})\right]\\ &\quad + \left(N(t) - S(t)\right)\left(0.9e^{-0.9} - \frac{1}{e}\right)\\ &\le -\delta(N(t) - S(t)) \le \delta S(t), \qquad (19)\end{aligned}$$

where $\delta = 1/e - 0.9e^{-0.9} = 0.002$, and the inequality holds because for $l \ge 1$ we can verify that $1/e - \theta e^{-\theta} - l(1 - e^{1-\theta})$ is positive when $\theta < 1$, and is negative when $\theta > 1$. Define $\tilde{y}_k^h = 0.9e^{-0.9}r_k^h/|I|$. We then have, from (18),

$$\begin{aligned}\Delta V_I^h &\le \sum_{k\in I}\Bigg[\left(2N(t) - S(t)\right)\left(\frac{x_k^h(t)}{r_k^h} - \frac{\tilde{y}_k^h}{r_k^h}\right)\\ &\quad + \frac{q_k^0(t)}{\alpha_k}(\tilde{x}_k^h - x_k^h(t))\Bigg] + \delta S. \qquad (20)\end{aligned}$$

Denote by ΔL_k^h each term in the $[\cdot]$ in the last expression. Recall that our threshold-based channel assignment policy assigns $x_i^h(t) = r_k^h$ if $\frac{q_k^0(t)r_k^h}{\alpha_k} > 1.2S(t)$; otherwise, $x_k^h(t) = 0$. For each device $k \in I$, we now divide into two sub-cases.

Case 1: When $q_k^0(t)r_k^h/\alpha_k > 1.2S(t)$. In this case, we have $x_k^h(t) = r_k^h$. As $\tilde{x}_k^h \in [0, r_k^h]$ and $\tilde{y}_k^h \le \frac{1}{e}r_k^h$, we thus

have $x_k^h(t) - \tilde{y}_k^h \geq 0$ and $\tilde{x}_k^h - x_k^h(t) \leq 0$. Further, since $N(t) < 1.1S(t)$, we have $2N(t) - S(t) < 1.2S(t)$. Thus,

$$\begin{aligned}\Delta L_k^h &\leq 1.2S(t)\left(\frac{x_k^h(t)}{r_k^h} - \frac{\tilde{y}_k^h}{r_k^h}\right) + 1.2S(t)\frac{\tilde{x}_k^h - x_k^h(t)}{r_k^h} \\ &\leq 1.2S(t)\left(\frac{\tilde{x}_k^h}{r_k^h} - \frac{\tilde{y}_k^h}{r_k^h}\right).\end{aligned} \tag{21}$$

Case 2: When $\frac{q_k^0(t)r_k^h}{\alpha_k} \leq 1.2 \cdot S(t)$. In this case, we have $x_k^h(t) = 0$, and thus $\tilde{x}_k^h - x_k^h(t) \geq 0$ and $x_k^h(t) - \tilde{y}_k^h \leq 0$. Since $N \geq 0.9S$, we have $2N(t) - S(t) > 0.8S(t)$, and then

$$\begin{aligned}\Delta L_k^h &\leq 1.2S(t)\frac{\tilde{x}_k^h - x_k^h(t)}{r_k^h} + 0.8S(t)\left(\frac{x_k^h(t)}{r_k^h} - \frac{\tilde{y}_k^h}{r_k^h}\right) \\ &= 1.2S(t)\left(\frac{\tilde{x}_k^h}{r_k^h} - \frac{2}{3}\frac{\tilde{y}_k^h}{r_k^h}\right). \quad (\text{as } x_k^h(t) = 0)\end{aligned} \tag{22}$$

Combining the two cases and using (20), ΔV_I^h can be upper bounded by

$$\begin{aligned}\sum_{k\in I}\Delta L_k^h + \delta S(t) &\leq 1.2S(t)\sum_{k\in I}\left(\frac{\tilde{x}_k^h}{r_k^h} - \frac{2}{3}\frac{\tilde{y}_k^h}{r_k^h}\right) + \delta S(t) \\ &\leq 1.2S(t)\left(\eta - \frac{2}{3}0.9e^{-0.9}\right) + \delta S(t) \\ &\leq -\delta_1 S(t) \leq -\frac{\delta_1}{2}S(t) - \frac{\delta_1}{2}\frac{N(t)}{1.1},\end{aligned} \tag{23}$$

where $\delta_1 = 1.2(\frac{2}{3}0.9e^{-0.9} - \eta) - \delta > 0$, since $\eta = 0.24$.

2) **When** $\theta(t) > 1.1$**:** In this case, the exact $N(t)$ may be much larger than its estimate $S(t)$. Thus, the first two parts of (7) may not produce a negative drift. Intuitively, we need the third part of (7) to provide a strong enough negative drift for the entire Lyapunov function. The following derivations make this intuition rigorous. Let $K = |I|$ denote the size of the contention group I. From (16), the drift ΔV_I^h can be written as

$$\begin{aligned}&\Delta V_I^h = \\ &\sum_{k\in I}\Bigg[\overbrace{\frac{q_k^0(t)}{\alpha_k}(\tilde{x}_k^h - x_k^h(t))}^{\Delta V_{0,k}} + \overbrace{N(t)\left(\frac{x_k^h(t)}{r_k^h} - \frac{\theta(t)e^{-\theta(t)}}{K}\right)}^{\Delta V_{N,k}} \\ &\quad \underbrace{+ (N(t) - S(t))\left(\frac{x_k^h(t)}{r_k^h} - \frac{\theta(t)e^{-\theta(t)}}{K} - \frac{\overline{\Delta S_I^h(t)}}{K}\right)}_{\Delta V_{N-S,k}}\Bigg].\end{aligned} \tag{24}$$

Let ΔL_k^h denote each term in the $[\cdot]$ of the above expression. Notice that $\overline{\Delta S_I^h(t)}$ is increasing and has one root at $\theta(t) = 1$. Thus, $\overline{\Delta S_I^h(t)} > 0$ for $\theta(t) > 1.1$. We now divide into two sub-cases.

Case 1: When $\frac{q_k^0(t)r_k^h}{\alpha_k} \geq 1.2S(t)$**, in which case** $x_k^h(t) = r_k^h$**.**

Note that this is the more difficult case because the positive drift of the first two terms of ΔL_k^h can be quite large. To see this, consider the scenario where both $q_k^0(t)$ and $S(t)$ are much smaller than $N(t)$. If the device knew the true $N(t)$, it should have not routed packets to q_k^h. Now that $x_k^h(t) = r_k^h$, $N(t)$ instead increases further, which adds a large increment to the Lyapunov drift. Thus, the only hope to have a negative drift is to reduce $(N(t) - S(t))^2$, i.e., by increasing $S(t)$ sufficiently fast. Specifically, since $\frac{q_k^0(t)r_i^h}{\alpha_k} \geq 1.2S(t)$ and $x_k^h(t) = r_k^h$, we have $\frac{q_k^0(t)}{\alpha_k}(\tilde{x}_k^h - x_k^h(t)) \leq 1.2S(t)(\frac{\tilde{x}_k^h}{r_k^h} - 1)$. Thus, we can bound ΔL_k^h as

$$\begin{aligned}\Delta L_k^h &\leq 1.2S(t)(\frac{\tilde{x}_k^h}{r_k^h} - 1) + S(t)\left(1 - \frac{\theta(t)e^{-\theta(t)}}{K}\right) \\ &\quad \underbrace{+ (N(t) - S(t))\left(2 - 2\frac{\theta(t)e^{-\theta(t)}}{K} + \frac{l(e^{1-\theta(t)} - 1)}{K}\right)}_{\text{negative when } l \geq 22K} \\ &\leq 1.2S(t)(\frac{\tilde{x}_k^h}{r_k^h} - 1) + S(t)\left(1 - \frac{\theta(t)e^{-\theta(t)}}{K}\right) - \delta_2 N(t) \\ &\leq S(t)\left(1.2\frac{\tilde{x}_k^h}{r_k^h} - 0.2\right) - \delta_2 N(t),\end{aligned} \tag{25}$$

where $\delta_2 > 0$ and in the second inequality we have used $N(t) - S(t) \geq 0.1/1.1N(t) > 0$ and $2K - 2\theta e^{-\theta} + l(e^{1-\theta} - 1) \leq 2K - l(e^{1-\theta} - 1) < 0$ for $l \geq 22K$ and $\theta > 1.1$.

Remark 1. *Note that the choice of $l \geq 22K$ is only needed to cover this case. Intuitively, the worst scenario is that all devices in the contention group fall into this case. As a result, $N(t)$ increase by $\Theta(K)$, and we need $S(t)$ to increase at the same magnitude to obtain a total negative drift, which leads to the choice of $l = \Theta(K)$. However, in reality this worst scenario may be very unlikely to occur. This explains why in our numerical results, a much smaller value of l suffices to stabilize the system. See details in Section V.*

Case 2: When $\frac{q_k^0(t)r_k^h}{\alpha_k} < 1.2S(t)$**, in which case** $x_k^h(t) = 0$

Compared to Case 1, this is the easier case because, even if the device knew the true $N(t)$, it may very well use the same decision $x_k^h(t) = 0$. Thus, we expect that the negative drift will be easier to establish. We can show that, for $\delta_3 > 0$,

$$\Delta L_k^h \leq S(t)\left(1.2\frac{\tilde{x}_k^h}{r_k^h} - \frac{0.449}{K}\right) - \delta_3 N(t), \quad \text{for } l \geq 1. \tag{26}$$

The detailed proof is similar to Case 1, and is omitted due to space limits. Combining Case 1 and Case 2, we have

$$\begin{aligned}\Delta V_I^h = \sum_{k\in I}\Delta L_k^h &\leq \sum_{k\in I} S(t)(1.2\frac{\tilde{x}_k^h}{r_k^h} - \frac{0.449}{K}) - \delta_4 N(t) \\ &\leq S(t)(1.2\eta - 0.449) - \delta_4 N(t) \\ &\leq -\delta_5 S(t) - \delta_4 N(t),\end{aligned} \tag{27}$$

where $\delta_5 = 0.449 - 1.2\eta > 0$, since $\eta = 0.24$, and $\delta_4 = \min\{\delta_2, \delta_3\}$.

For the last case of $0 < \theta(t) < 0.9$, we can show $\Delta V_I^h < -\delta_6 S(t) - \delta_7 N(t)$ using similar arguments. Details are omitted due to space constraints. Combining all three cases of $\theta(t)$ and using (16), we conclude that $\mathsf{E}[\Delta V_{\text{tot}}(\vec{q}(t))|\vec{q}(t)] < 0$ whenever $\vec{q}(t)$ is large. The result of the theorem then follows.

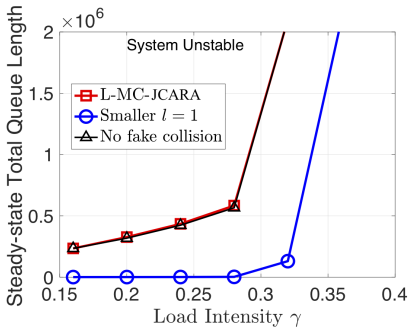

Fig. 3. Comparison of steady-state total queue length at different load intensity for L-MC-JCAS and its variants.

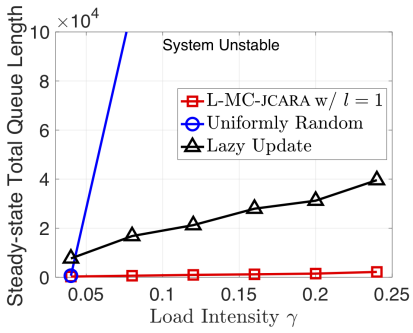

Fig. 4. Comparison of L-MC-JCAS and (i) uniformly random channel assignment; (ii) lazy update.

V. Simulation Results

In this section, we verify the performance of our proposed L-MC-JCARA scheme in Algorithm 3 through simulation. We simulate a cellular UL IoT system where the BS can form $M = 6$ beams at each time to serve $N = 300$ IoT devices. Although the system scale is relatively small for IoT, it demonstrates the increased capacity region of our proposed scheme over other schemes with comparable overhead. In certain sense, a larger number of devices can be more favorable for our proposed scheme, as each device only has a smaller share of the load. More extensive numerical experiments will be performed for the future work.

At each time, the BS can form one out of $H = 2$ beam patterns according to PRBF. At the BS, we first set $l = 22K$ in Theorem 1. At the device side, we assume that the users are distributed evenly so that $K = 50$ random devices belong to each beam (contention group). As beam pattern varies, each device will see two channels for transmission: one good channel with expected rate 10 kbps, and a bad channel with expected rate 2 kbps. At the beginning, each device i has an initial amount of data in the 0-th virtual queue, i.e., $q_i^0(0) \in \mathcal{U}(0, B_0]$. In the simulation, we vary the load intensity γ fed to the devices, so that the sum offered load to each contention group is 10γ kbps.

We first verify the stability of our proposed L-MC-JCARA. We simulate the system for $T = 10^6$ time slots with $B_0 = 2500$kb. In Fig 3, we plot the *average steady-state total queue length*, $Q_{\text{total}}^{\text{SS}}$, versus different load intensity γ for our proposed L-MC-JCAS and two variants: (i) using small value of $l = 1$ as we mentioned in Section IV-C, and (ii) treating the fake collision as data transmissions. First, we observe that, in all cases, $Q_{\text{total}}^{\text{SS}}$ in the system increases as the load intensity increases, up to the point when the system becomes unstable. Our proposed L-MC-JCARA and the variant without fake collision have similar performance on $Q_{\text{total}}^{\text{SS}}$ and can maintain system stability up to $\gamma = 0.28$, indicating the impact of fake collision is small. We verify that the optimal centralized scheduler cannot stabilize offer load of 10kbps. Thus the achieved efficiency ratio at $\gamma = 0.28$ is at least 0.28. The variant of L-MC-JCARA with $l = 1$ stabilizes the system up to $\gamma = 0.32$ and obtains an even lower $Q_{\text{total}}^{\text{SS}}$. This is consistent of our previous conjecture that the large l in Theorem 1 is for analysis purpose, and a smaller l performs well in practice, especially when $K = |I|$ is large. Thus, in the rest of the section, we will use $l = 1$ when we simulate L-MC-JCARA.

Next, we demonstrate the advantages of L-MC-JCARA over two other low-overhead multi-channel random access schemes: (i) the uniformly random channel assignment (UR) used in [17], which is originally proposed for homogeneous multi-channel system; (ii) the L-QRA combined with "lazy" update of congestion information $N(t)$, i.e., the devices will report their queue length to the BS once every 300 time slots. We simulate for $T = 5 \cdot 10^5$ time slots with $B_0 = 1000$kb, and plot $Q_{\text{total}}^{\text{SS}}$ against different load intensity for the above-mentioned three schemes in Fig. 4. We observe that, the system $Q_{\text{total}}^{\text{SS}}$ under UR becomes unstable for $\gamma \geq 0.08$. This poor performance of UR is because uniform assignment does not account for heterogeneous channel qualities, i.e., a significant amount traffic is assigned to low-rate channels, which exceeds the channel capacity. Compared to L-MC-JCARA with $l = 1$, the scheme with lazy update can stabilize the system up to load intensity $\gamma = 0.24$, but with significantly higher $Q_{\text{total}}^{\text{SS}}$. Moreover, the value of $Q_{\text{total}}^{\text{SS}}$ under lazy update is oscillating due to the increasing inaccuracy of congestion information over time between two updates. In summary, our proposed L-MC-JCARA attains lower $Q_{\text{total}}^{\text{SS}}$ than both other schemes.

VI. Conclusion

In this work, we propose a L-MC-JCARA scheme for UL massive IoT system under non-uniform channel qualities and loads, and show that it can stabilize any offer load vector that is strictly inside $0.24\Omega_0$. A key novelty is to let the BS update an estimate of the congestion information in each contention group by observing only the channel events. As a result, our work is the first in the literature to guarantee stability in such a non-uniform multi-channel random access system *without any queue-length feedback from devices*. In the future, we will study the stability guarantee when a constant value is used for l, and perform larger-scale simulation for massive IoT systems. Moreover, we will study how to combine our schedule with SCMA/NOMA (Non-orthogonal Multiple Access) [20] to further lower the overhead.

REFERENCES

[1] 5G Americas White Paper, "5G: The Future of IoT," Tech. Rep., 07 2019.

[2] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wirel. Pers. Commun.*, vol. 6, no. 3, pp. 311–335, 1998.

[3] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J.Sel. A. Commun.*, vol. 16, no. 8, pp. 1451–1458, Sep. 2006.

[4] P. Viswanath, D. N. C. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1277–1294, June 2002.

[5] Y. Zou, K. T. Kim, X. Lin, M. Chiang, Z. Ding, R. Wichman, and J. Hämäläinen, "Low-overhead multi-antenna-enabled uplink random access for massive machine-type communications with low mobility," in *2019 IEEE Global Communications Conference (GLOBECOM),*, Waikoloa, HI, December 2019.

[6] A. A. Dowhuszko, G. Corral-Briones, J. Hämäläinen, and R. Wichman, "Performance of quantized random beamforming in delay-tolerant machine-type communication," *IEEE Transactions on Wireless Communications*, vol. 15, no. 8, pp. 5664–5680, Aug 2016.

[7] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Trans. on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, December 1992.

[8] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. on Networking*, vol. 13, no. 2, pp. 411–424, April 2005.

[9] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *2012 Proceedings IEEE INFOCOM*, March 2012.

[10] B. Li, B. Ji, and J. Liu, "Efficient and low-overhead uplink scheduling for large-scale wireless internet-of-things," in *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2018, pp. 1–8.

[11] C. Joo, X. Lin, and N. B. Shroff, "Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multi-hop Wireless Networks," in *IEEE INFOCOM*, 2008.

[12] A. L. Stolyar, "Dynamic distributed scheduling in random access networks," *Journal of Applied Probability*, vol. 45, no. 2, pp. 297–313, 2008.

[13] X. Lin and S. Rasool, "Constant-Time Distributed Scheduling Policies for Ad Hoc Wireless Networks," *IEEE Trans. on Automatic Control*, vol. 54, no. 2, pp. 231–242, February 2009.

[14] X. Lin and S. B. Rasool, "Distributed and provably efficient algorithms for joint channel-assignment, scheduling, and routing in multichannel ad hoc wireless networks," *IEEE/ACM Trans. on Network.*, vol. 17, no. 6, pp. 1874–1887, Dec 2009.

[15] A. Israel and A. Itai, "A fast and simple randomized parallel algorithm for maximal matching," *Inf. Process. Lett.*, vol. 22, no. 2, pp. 77–80, Feb. 1986.

[16] V. A. Milkhailov, "Geometrical analysis of the stability of markov chains in rn+ and its application to throughput evaluation of the adaptive random multiple access algorithm," *Probl. Inform. Transm.*, pp. 47–56, 1988.

[17] O. Galinina, A. Turlikov, S. Andreev, and Y. Koucheryavy, "Stabilizing multi-channel slotted aloha for machine-type communications," in *2013 IEEE International Symposium on Information Theory*, July 2013, pp. 2119–2123.

[18] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 960–972, Jun. 2010.

[19] L. Ying and S. Shakkottai, "Scheduling in mobile ad hoc networks with topology and channel-state uncertainty," in *IEEE INFOCOM 2009*, April 2009, pp. 2347–2355.

[20] K. Au, L. Zhang, H. Nikopour, E. Yi, A. Bayesteh, U. Vilaipornsawai, J. Ma, and P. Zhu, "Uplink contention based scma for 5g radio access," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 900–905.

[21] N. Jiang, Y. Deng, A. Nallanathan, X. Kang, and T. Q. S. Quek, "Analyzing random access collisions in massive iot networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6853–6870, Oct 2018.

[22] N. Jiang, Y. Deng, X. Kang, and A. Nallanathan, "Random access analysis for massive iot networks under a new spatio-temporal model: A stochastic geometry approach," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5788–5803, Nov 2018.

[23] C. Wei, R. Cheng, and S. Tsao, "Modeling and estimation of one-shot random access for finite-user multichannel slotted aloha systems," *IEEE Communications Letters*, vol. 16, no. 8, pp. 1196–1199, August 2012.

[24] Y. Zou, K. T. Kim, X. Lin, M. Chiang, Z. Ding, R. Wichman, and J. Hämäläinen, "Low-overhead multi-antenna-enabled uplink random access for massive machine-type communications with low mobility," https://engineering.purdue.edu/%7elinx/papers.html, Tech. Rep., 2019.

[25] N. Abramson, "The aloha system: Another alternative for computer communications," in *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, ser. AFIPS '70 (Fall), 1970, pp. 281–285.