Minimum Feedback for Collision-Free Scheduling in Massive Random Access

Justin Kang and Wei Yu
Department of Electrical and Computer Engineering
University of Toronto
js.kang@mail.utoronto.ca weiyu@ece.utoronto.ca

***Abstract*—This paper considers a massive random access scenario where a small random set of k active users out of a larger number of n total potential users seek to transmit data to a base station. Specifically, we examine an approach in which the base station first determines the set of active users based on an uplink pilot phase, then broadcasts a common feedback message to all the active users for the scheduling of their subsequent data transmissions. Our main question is: What is the minimum amount of common feedback needed to schedule k users in k slots while completely avoiding collisions? Instead of a naive scheme of using $k\log(n)$ feedback bits, this paper presents upper and lower bounds to show that the minimum number of required common feedback bits scales linearly in k, plus an additive term that scales only as $\Theta(\log\log(n))$. The achievability proof is based on a random coding argument. We further connect the problem of constructing a minimal length feedback code to that of finding a minimal set of complete k-partite subgraphs that form an edge covering of a k-uniform complete hypergraph with n vertices. Moreover, the problem is also equivalent to that of finding a minimal perfect hashing family, thus allowing leveraging the explicit perfect hashing code constructions for achieving collision-free massive random access.**

I. INTRODUCTION

Massive connectivity is vital for future wireless networks. As the number of connected devices grows, the challenges in providing connectivity also grow commensurately. The distinguishing features of machine-type networks, also known as the Internet of Things (IoT), include massive numbers of devices (in the order of $10^5 \sim 10^6$ per base-station (BS)) and sporadic traffic, making both the identification of active devices and the subsequent scheduling of their data transmissions challenging tasks [1]–[5].

This paper considers a massive random access model with n users in a cell, of which a random set of $k \ll n$ users seek to send a small payload data to the BS in the uplink [1]. We propose the following random access scheme involving three phases with limited feedback. In the first phase, k active users transmit pre-assigned uniquely identifying pilot sequences over the multiple access channel to indicate their activities. The BS uses a multiuser detection algorithm, typically involving compressed sensing [6]–[9] to determine the active user set. In the second phase, the BS transmits common feedback bits to all active users over a noiseless broadcast channel; these feedback bits specify a schedule for the subsequent data transmissions of k active users over k orthogonal slots. In the third and final phase, the users transmit their data over the scheduled slots based on the feedback.

In this paper, we assume that in the first phase the user activities are all detected correctly, and focus on the user scheduling problem in the second phase. A main issue with this *scheduled* approach to random access is the potentially large amount of feedback required to ensure no collision. A trivial feedback code is for the BS to index each of the n users and list off all the active user indices in the order which they should transmit. This requires a feedback of $k\log(n)$ bits. When the number of potential users is large, the $\log(n)$ factor can be significant. The main result of this paper is that it is possible to reduce the feedback overhead to $k\log(e) + \Theta(\log\log(n))$ bits, while maintaining zero collision in scheduling the k active users in k slots. This is a significant reduction from $k\log(n)$. The above scheduled approach to random access can be compared to the *contention* based schemes such as slotted ALOHA [10], which, due to collision and retransmission, has an overhead of roughly $Bk\left(\frac{1}{\eta} - 1\right)$ bits, where B is the payload size and η is the efficiency of the chosen ALOHA variant, which varies from $\eta = \frac{1}{e} \approx 0.37$ for classic slotted ALOHA, to $\eta \approx 0.8$ in irregular repetition slotted ALOHA [11]. The proposed scheduled approach can also be compared to *unsourced* multiple access in which the users are not required to transmit identification information [5]. But in both schemes, the user identification generally needs to be embedded in the payload. When the number of potential users is large (e.g., up to $n = 10^6$), the cost of identification can be significant (e.g., up to 20 bits), especially when the payload is small. This is in contrast to the much smaller $\log\log(n)$ overhead of the scheme proposed in this paper. This $\log\log(n)$ factor is reminiscent of the *identification capacity* [12]; but the use of identification code for scheduling would have required $k \cdot \Theta(\log\log(n))$ bits, while the scheme in this paper is more efficient in requiring only $k\log(e) + \Theta(\log\log(n))$ bits.

The optimal feedback coding scheme of this paper involves finding a minimal set of partitions over $\{1, \ldots, n\}$ such that no matter which user activity pattern occurs, there is always one partition for which each subset of the partition contains exactly one active user. This problem turns out to be equivalent to the hypergraph covering problem [13] and the perfect hashing family problem [14]. In fact, the main contribution of this paper can be viewed as the establishment of these connections, thereby allowing the leveraging of prior results in combinatorics to obtain upper and lower bounds and to construct explicit codes for optimal feedback.

The notations used in this papers are as follows. We use $[n]$ to denote $\{1, 2, \ldots, n\}$ and $\binom{[n]}{k}$ to denote the set of all k-element subsets of $[n]$. All other sets are typeset in upper case boldface. We use $\log(\cdot)$ to denote logarithm in base 2, and $\ln(\cdot)$ for natural logarithm in base e. We use $\mathbb{1}(\cdot)$ to denote the indicator function, which takes a value of 1 whenever the expression inside is true and 0 otherwise.

II. Feedback for Collision-Free Scheduling

A. Encoding and Decoding Functions

Assuming successful detection of k active users among n potential users in the first phase, the problem of designing a collision-free feedback code for the second and the third phases is that of constructing an encoding function at the BS that maps all possible occurrence of k-tuples out of n users to an index set $\mathbf{W}$ of rate R

$$f : \binom{[n]}{k} \to \left\{1, 2, \ldots, 2^R\right\} \triangleq \mathbf{W} \tag{1}$$

and decoding functions g_i at user i that specify each user's scheduled slot, i.e.,

$$g_i : \mathbf{W} \to [k]\,, \quad i \in [n] \tag{2}$$

such that the subsequent transmissions by the k active users over the k orthogonal slots can take place in a collision-free manner. More specifically, define an "activity pattern" to be some element $\mathbf{A} \in \binom{[n]}{k}$, which is a set of indices of k active users. A feedback scheme for collision-free transmission requires

$$\forall \mathbf{A} \in \binom{[n]}{k}, \ \ \forall i, j \in \mathbf{A}, \ \ g_i\left(f\left(\mathbf{A}\right)\right) \neq g_j\left(f\left(\mathbf{A}\right)\right) \tag{3}$$

as collision occurs whenever the decoding functions of two active users within the same activity pattern produce the same output. Another way to view the collision-free condition is:

$$\forall \mathbf{A} \in \binom{[n]}{k}, \ \ \exists w \in \mathbf{W} \ \text{ s.t. } \ \forall i, j \in \mathbf{A} \ \ g_i(w) \neq g_j(w) \tag{4}$$

where $w = f(\mathbf{A})$. An optimal collision-free feedback code is a code with minimum rate R that satisfies the collision-free condition above.

Note that in the definition above, we have assumed fixed n and k, and restricted the range of g_i to be $[k]$. Relaxing the range of g_i to be $[b]$ with $b > k$ can potentially significantly save feedback rate in the second phase, at a cost of a larger number of scheduling slots in the third phase. Investigating this tradeoff is a subject of future work. Further, the broadcast phase is assumed to be noiseless, so that all active users receive the same common feedback without error.

The interpretation of the collision-free condition (4) turns out to be equivalent to the separation condition considered by Fredman and Komlós [14] in the context of perfect hashing families and to the hypergraph covering problem considered by Snir [13], Radhakrishnan [15] and others. These relationships are investigated later in this paper.

B. Set Partitioning Scheme for Collision-Free Feedback

The goal is to find efficient encoding and decoding rules such that for any user activity pattern $\mathbf{A}$, we can ensure that the active users can be scheduled without collision. A simple way to do this is to assign a unique index to each of n users, then the feedback code simply consists of listing the k active users in the order at which they should transmit. Each user finds its index in the list, waits for its turn, then transmits at its scheduled slot. Thus, a feedback rate of $R = k\lceil\log(n)\rceil$ is achievable for collision-free scheduling.

The main point of this paper is that it is possible to do significantly better than the $O(\log(n))$ scaling in the feedback rate above. The key observation is that the above simple scheme specifies a precise order at which k users should transmit, but there are $k!$ collision-free schedules over the k users. It is possible to use the flexibility of only having to specify *one* of the $k!$ schedules to significantly reduce the feedback rate. Further, the above simple scheme reveals the identities of all the active users and their scheduled slots to everyone. This is clearly extraneous information, as each user only needs to know which slot it should transmit and does not care about the schedules of the other users.

To illustrate how to do significantly better than $O(\log(n))$, consider an example of $k = 2$ and $n \gg k$, i.e., the task of scheduling two randomly chosen active users among a larger number of potential users. Index each of the n users with $\lceil\log(n)\rceil$ bits, using a binary representation of its index. Since the binary representations of any two distinct non-negative integers must differ in at least one position, we can use a feedback scheme that specifies the location where the indices of the two users differ. Each user would examine the bit value of its own index at that location. The user with bit value 0 would transmit first, and the user with bit value 1 would transmit second, thus avoiding collision. Since specifying a location in the index of length $\lceil\log(n)\rceil$ requires $R = \lceil\log\lceil\log(n)\rceil\rceil$ bits, we achieve $O(\log(\log(n)))$ scaling for collision-free feedback! The key to achieving such saving is in assigning multiple "compatible" activity patterns to the same feedback output, then defining decoding rules that result in zero collision for all "compatible" activity patterns.

This idea of defining "compatible" activity pattern can be generalized to arbitrary (n, k) and made rigorous using the following definition of an encoder and decoders. Define a k-partition of a set $[n]$ to be a tuple of subsets $\bar{\mathbf{X}} = (\mathbf{X}_1, \ldots, \mathbf{X}_k)$ such that $\mathbf{X}_i \bigcap \mathbf{X}_j = \emptyset, \forall i, j$, and $\bigcup_{i=1}^{k} \mathbf{X}_i = [n]$. Next define the following set of size-k subsets of $[n]$ as:

$$\mathbf{C}\left(\bar{\mathbf{X}}\right) = \left\{\{x_1, \ldots, x_k\} \mid x_i \in \mathbf{X}_i, i = 1, \ldots, k\right\}. \tag{5}$$

Intuitively, these size-k subsets of $[n]$ correspond to the activity patterns for which each active user belongs to a distinct subset in the partition. The idea is that by specifying a k-partition $\bar{\mathbf{X}}$, for all activity patterns in $\mathbf{C}\left(\bar{\mathbf{X}}\right)$, each active user can simply look at which subset it belongs to in the partition, then schedule itself in the slot corresponding to the index of the subset in a collision-free manner.

To make sure that all activity patterns are covered, we construct T partitions $\bar{\mathbf{X}}^{(1)}, \ldots, \bar{\mathbf{X}}^{(T)}$ such that

$$\bigcup_{t=1}^{T} \mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right) = \binom{[n]}{k}. \tag{6}$$

Then, whenever an activity pattern occurs, the BS only needs to specify a partition in which the activity pattern is covered.

More formally, define the encoding function as a mapping from the activity pattern to $[T]$ and the decoding functions as mappings from $[T]$ to the scheduling slots such that

$$f(\mathbf{A}) = t \ \text{ s.t. } \ \mathbf{A} \in \mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right) \tag{7}$$

$$g_i\left(t\right) = j \ \text{ if } \ i \in \mathbf{X}_j^{(t)} \tag{8}$$

where $\bar{\mathbf{X}}^{(t)} = (\mathbf{X}_1^{(t)}, ..., \mathbf{X}_k^{(t)})$. By (6), for any arbitrary activity pattern $\mathbf{A}$, one can always find t to satisfy the condition in (7). If more than one such t exists, an arbitrary choice is made. Since exactly one user is in each subset of $\bar{\mathbf{X}}^{(t)}$, (8) guarantees that the schedule is collision-free. We define the rate of this set-partition based feedback scheme as $R \triangleq \log(T)$.

The above set-partition view of scheduling with limited feedback is completely general in the sense that any choice of deterministic decoders $g_i'(t)$ that achieve no collision for every activity pattern at a feedback rate R can be written in this set-partition framework with 2^R partitions. Given the decoding functions $g_i' : \left[2^R\right] \to [k]$, we can define 2^R partitions $\bar{\mathbf{X}}^{(t)} = \left(\mathbf{X}_1^{(t)}, \ldots, \mathbf{X}_k^{(t)}\right)$, $t \in \left[2^R\right]$, where

$$\mathbf{X}_j^{(t)} = \left\{i \mid g_i'\left(t\right) = j, i \in [n]\right\}. \tag{9}$$

Using this construction, partition t covers precisely the activity patterns for which the feedback symbol t results in no collision. Since the set of functions g_i' needs to result in no collision for every activity pattern $\mathbf{A}$, this means that (6) must be satisfied. Thus, we can now restrict attention to this set-partition strategy without loss of generality for finding the minimum feedback rate for scheduling k out of n users in a collision-free manner. In other words, the problem now reduces to finding the minimum T needed to satisfy (6).

III. Minimum Feedback Rate

Let the minimum common feedback rate required for collision-free scheduling of k out of n users be denoted as $R^*(k,n)$. This section first describes a random coding based argument that shows the existence of partitions of suitable size that satisfy (6), thereby giving an upper bound on $R^*(k,n)$ for arbitrary n and k. We also present two lower bounds. Together, we show that $R^*(k,n)$ has a linear scaling in k, plus an additive term which is only double logarithmic in n.

A. Achievable Rate via Random Coding

The main challenge is the explicit construction of a family of partitions that cover all activity patterns. In this section, we use a random coding argument to show the existence of a family of partitions with rate R that scales linearly in k plus an additive $O(\log\log(n))$ term.

Theorem 1. *Let T^* be the smallest integer such that there exist partitions $\bar{\mathbf{X}}^{(1)}, \ldots, \bar{\mathbf{X}}^{(T^*)}$ that satisfy condition (6). Let $R^*(k,n) \triangleq \log(T^*)$. Assuming $k|n$, we have*

$$R^*(k,n) \leq k\log(e) + \log\left(\ln\left(\frac{n}{k}\right) + 1\right) + \frac{1}{2}\log\left(\frac{k}{2\pi}\right). \tag{10}$$

Thus, for a massive random access network, there exists a feedback strategy for scheduling k out of n users with no collision at the above rate, which scales linearly in k plus an $O(\log(\log(n)))$ additive term.

Proof. Fix some $\mathbf{A} \in \binom{[n]}{k}$. Define a "balanced" k-partition of $[n]$ to be a partition $\bar{\mathbf{X}}$ for which $|\mathbf{X}_j| = \frac{n}{k}, \forall j$. If we choose a balanced partition at random, the probability that $\mathbf{A}$ is covered by $\mathbf{C}\left(\bar{\mathbf{X}}\right)$ can be written as:

$$\Pr\left(\mathbf{A} \in \mathbf{C}\left(\bar{\mathbf{X}}\right)\right) = \frac{\left(\frac{n}{k}\right)^k}{\binom{n}{k}} = \left(\frac{k!}{k^k}\right)\gamma(n,k) \tag{11}$$

where we have defined $\gamma(n,k) \triangleq \frac{n^k}{n(n-1)\ldots(n-k+1)}$. If T random balanced k-partitions are generated independently, the probability that none of the T partitions cover a given activity pattern is

$$\Pr\left(\mathbf{A} \notin \bigcup_{t=1}^{T} \mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right)\right) = \left(1 - \frac{k!}{k^k}\gamma(n,k)\right)^T. \tag{12}$$

We aim to use the above expression to establish that if T satisfies (10) then there must exist at least one family of T partitions $\bar{\mathbf{X}}^{(1)}, \cdots, \bar{\mathbf{X}}^{(T)}$ such that $\bigcup_{t=1}^{T} \mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right) = \binom{[n]}{k}$. To do this, we consider the difference between the number of elements in $\binom{[n]}{k}$ and the number of elements in $\bigcup_{t=1}^{T} \mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right)$, i.e., the total number of activity patterns which have not been covered by any of the T partitions. But computing this directly is difficult. Instead, we compute the expected value of this difference, where the expectation is taken over randomly and independently generated T balanced k-partitions, i.e.,

$$\mathbb{E}\left[\binom{n}{k} - \left|\bigcup_{t=1}^{T} \mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right)\right|\right] \triangleq D. \tag{13}$$

Note that both quantities in the difference are integers. Thus, if this expectation drops strictly below 1, then we can be assured that there is at least one family of T k-partitions that completely cover $\binom{[n]}{k}$. This is because if every such partitions has a positive gap from $\binom{n}{k}$, then the expected value of the difference would have to be greater than or equal to 1.

Next, we re-write the expectation as

$$D = \mathbb{E}\left[\sum_{l=1}^{\binom{n}{k}} \mathbb{1}\left(\mathbf{A}_l \notin \bigcup_{t=1}^{T} \mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right)\right)\right], \tag{14}$$

where $\mathbf{A}_l$, $l = 1, \cdots, \binom{n}{k}$ is an exhaustive list of all possible activity patterns of k out of n users. This holds because for each fixed partition, we now simply count how many activity patterns are not covered by the partition, then take expectation.

By linearity of the expectation, this is equivalent to:

$$D = \sum_{l=1}^{\binom{n}{k}} \mathbb{E}\left[\mathbb{1}\left(\mathbf{A}_l \notin \bigcup_{t=1}^{T} \mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right)\right)\right] \quad (15)$$

$$= \binom{n}{k}\mathbb{E}\left[\mathbb{1}\left(\mathbf{A} \notin \bigcup_{t=1}^{T} \mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right)\right)\right] \quad (16)$$

$$= \binom{n}{k}\left(1 - \frac{k!}{k^k}\gamma(n,k)\right)^T \quad (17)$$

where in the second last equality we utilize the fact that the quantity inside the summation (15) does not change if we re-label the entries $\mathbf{A}_l$, so each term in the summation must be equal, and in the last equality, we use (12).

As mentioned before, if $D < 1$, then there must exist at least one family of partitions that cover $\binom{[n]}{k}$. Using the fact that $(1-x) < e^{-x} \quad \forall x > 0$, this gives us the following sufficient condition on T that ensures (6):

$$\binom{n}{k} \exp\left(-\frac{k!}{k^k}\gamma(n,k)T\right) \leq 1. \quad (18)$$

Taking logarithm of both sides and simplifying yield

$$T \geq \ln\binom{n}{k}\left(\frac{k^k}{k!\gamma(n,k)}\right). \quad (19)$$

Now, let $R = \log(T)$ be the number of feedback bits required to avoid collision. The above calculation ensures that if

$$R \geq k\log(k) - \log(k!) + \log\left(\frac{\ln\frac{n^k}{\gamma(n,k)k!}}{\gamma(n,k)}\right), \quad (20)$$

then there must exist a collision-free feedback code for scheduling k out of n users.

Noting that $\gamma(n,k) > 1$, we have

$$\left(\frac{\ln\frac{n^k}{\gamma(n,k)k!}}{\gamma(n,k)}\right) < k\ln(n) - \ln(k!). \quad (21)$$

Using the fact that $k! > \sqrt{2\pi}k^{k+\frac{1}{2}}e^{-k}e^{\frac{1}{12k+1}}$, we arrive at the following sufficient condition on R for the existence of a collision-free feedback code:

$$R \geq k\log(e) + \log\left(\ln\left(\frac{n}{k}\right)+1\right) + \frac{1}{2}\log\left(\frac{k}{2\pi}\right). \quad (22)$$

The achievability at rate R means that the minimum rate $R^*(k,n)$ must be upper bounded by R, i.e.,

$$R^*(k,n) \leq k\log(e) + \log\left(\ln\left(\frac{n}{k}\right)+1\right) + \frac{1}{2}\log\left(\frac{k}{2\pi}\right). \quad (23)$$

This minimum rate scales linearly in k, plus an additive term that scales as $O(\log\log(n))$. □

As will be explained later in the paper, the minimum rate of the collision-free feedback code is closely related to the study of perfect hash families. The above argument is similar to the arguments used for proving bounds in the perfect hash function literature [14].

B. Lower Bounds on the Minimum Feedback Rate

We now present two converse results showing that the minimum feedback rate must have at least a linear scaling in k and double-log scaling in n. The first result is a simple volume bound; it is known in, e.g., [14].

Theorem 2. *The minimum number of partitions $\bar{\mathbf{X}}^{(1)}, \ldots, \bar{\mathbf{X}}^{(T^*)}$ that satisfy condition (6) must have its rate $R^*(k,n) \triangleq \log(T^*)$ bounded below by*

$$R^*(k,n) \geq k\log(e) - \log\left(\frac{n^k}{n(n-1)...(n-k+1)}\right) - \frac{1}{2}\log(2\pi k) - \frac{\log(e)}{12k}. \quad (24)$$

Thus, scheduling k out of a total of n users for massive random access with no collision requires a feedback rate that scales at least linearly in k when $k \ll n$.

Proof. The number of activity patterns covered by a partition is maximized when the sizes of the sub-partitions take integer values surrounding $\frac{n}{k}$. In particular, we can show that

$$\left|\mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right)\right| \leq \left\lceil\frac{n}{k}\right\rceil^{n \bmod k}\left\lfloor\frac{n}{k}\right\rfloor^{k-n \bmod k} \leq \left(\frac{n}{k}\right)^k. \quad (25)$$

Thus, in order to cover all the activity patterns, i.e., to satisfy condition (6), we must have $T^* \geq \frac{\binom{n}{k}}{\left(\frac{n}{k}\right)^k}$.

This bound is not necessarily tight, because the covering sets $\mathbf{C}\left(\bar{\mathbf{X}}^{(t)}\right)$ are not necessarily disjoint. But it already provides the desired linear scaling bound. If we use the upper bound $k! < \sqrt{2\pi}k^{k+\frac{1}{2}}e^{-k}e^{\frac{1}{12k}}$, we get (24).

The second term in (24) is close to zero in the regime of interest (i.e., $n \gg k$). Thus, the minimum feedback rate must scale at least linearly in k in this regime. □

Theorem 3. *The minimum number of partitions $\bar{\mathbf{X}}^{(1)}, \ldots, \bar{\mathbf{X}}^{(T^*)}$ that satisfy condition (6) must have its rate $R^*(k,n) \triangleq \log(T^*)$ bounded below by*

$$R^*(k,n) \geq \log\log\left(\frac{n}{k-1}\right) + \log(k). \quad (26)$$

Thus, scheduling k out of a total of n users for massive random access with no collision requires a feedback rate that scales at least double logarithmically in n.

Proof. Consider the first partition. We seek to bound the number of activity patterns that this first partition cannot cover by noting that

$$\mathbf{C}\left(\bar{\mathbf{X}}^{(1)}\right) \cap \binom{[n] - \mathbf{X}_j^{(1)}}{k} = \emptyset, \quad j = 1, \ldots, k, \quad (27)$$

i.e, $\bar{\mathbf{X}}^{(1)}$ cannot cover an activity pattern which has all its elements drawn from $[n] - \mathbf{X}_j^{(1)}$. Since the partition must have at least one subset of size at most $\left\lfloor\frac{n}{k}\right\rfloor$, it must be that $\bar{\mathbf{X}}^{(1)}$ cannot have covered any activity patterns whose elements are exclusively drawn from a set of indices of size $m_1(n,k)$, where

$$m_1(n,k) \geq n - \left\lfloor\frac{n}{k}\right\rfloor \geq n\left(1 - \frac{1}{k}\right). \quad (28)$$

Now take the second partition, and consider how many of the above activity patterns are still not covered by the second partition. By the same logic, since the second partition cannot cover any activity patterns drawn from an index set with indices from one of the subsets of the partition removed, and when restricted to the set of indices of size $m_1(n,k)$, there is at least one subset which overlaps with at most $\left(1-\frac{1}{k}\right)$ portion of $m_1(n,k)$ indices, we conclude that all the activity patterns whose elements are drawn from an index set of size $m_2(n,k)$, where

$$m_2(n,k) \geq n\left(1-\frac{1}{k}\right)^2, \tag{29}$$

cannot possibly be covered by either the first partition or the second partition. Continuing for T partitions, the only way that the remaining indices cannot support any activity patterns is for $m_T(n,k) \leq k-1$. This gives us the following necessary condition on T:

$$n\left(1-\frac{1}{k}\right)^T \leq k-1. \tag{30}$$

Since T^* must be greater than or equal to any T that satisfies the above, by taking the logarithm of the above, we have

$$T^* \geq \frac{\log(n)-\log(k-1)}{\log(k)-\log(k-1)}. \tag{31}$$

In terms of rate, by taking the logarithm again and by noting that $-\log\left(1-\frac{1}{k}\right) < \frac{1}{k}$ for $k>1$, we get our desired result,

$$R^*(k,n) \geq \log\log\left(\frac{n}{k-1}\right) + \log(k). \tag{26}$$

Thus for fixed k, the minimum feedback rate for zero collision must scale at least double logarithmically in n. □

Note that in the context of hypergraph coverings, which is described next, (31) can be interpreted as Snir's bound [13].

IV. Hypergraphs and Perfect Hashing Families

The minimum set-partition problem turns out to be closely connected to the problem of finding an edge covering of a complete k-uniform hypergraph with a set of complete k-partite subgraphs, and also the problem of finding a family of perfect minimal hashing functions. These connections allow us to leverage existing results in combinatorics for even tighter bounds and for possible explicit feedback code constructions.

Consider a k-uniform complete hypergraph $\mathcal{A} = (\mathbf{V}, \mathbf{E})$ with $\mathbf{V} = [n]$ and $\mathbf{E} = \binom{[n]}{k}$. We can interpret the partition defined in Section III-A as a k-partite complete subgraph of this hypergraph with edge set $\mathbf{C}\left(\bar{\mathbf{X}}\right)$. Then, the question of whether every edge of a hypergraph $\mathcal{A}$ is covered by a set of T complete k-partite subgraphs can be interpreted as precisely the condition (6). Thus, finding a set of complete k-partite subgraphs of $\mathcal{A}$ which cover $\mathcal{A}$ is equivalent to the minimum set-partition problem described earlier. The concept of graph entropy has been used to establish lower bounds on the minimum T^* required for edge covering [15].

The perfect hashing family problem is introduced by Fredman and Komlós [14]. An (n,b,k)-family of perfect hash functions is a family of functions from $[n] \to [b]$ for $n \geq b \geq k$ such that for every $\mathbf{A} \subset [n], |\mathbf{A}| = k$, there exists a function in the family that is injective on $\mathbf{A}$. An (n,k)-family of minimal perfect hash functions is an (n,b,k)-family of perfect hash functions where $b=k$. We can view the decoding functions (2) as a family of $T = |\mathbf{W}|$ functions from $[n] \to [k]$, if we swap the argument and the subscript. With this interpretation, we can see that our decoding functions are nothing more than an (n,k)-family of minimal perfect hash functions.

Theorem 4 (Fredman and Komlós [14], Körner and Marton [16] [17])**.** *The minimal number of functions T^* in an (n,b,k)-family of perfect hash functions is given by:*

$$\frac{\log n}{\min_{1\leq s\leq k-1} g(b,s)\log\frac{b-s+1}{k-s}} \lesssim T^* \lesssim \frac{(k-1)\log n}{\log\frac{1}{1-g(b,k)}} \tag{32}$$

where $g(b,k) = \prod_{j=0}^{k-1}\left(1-\frac{j}{b}\right)$.

The notation $P(n) \lesssim Q(n)$ means $P(n) \leq (1+o(1))Q(n)$, where the $o(1)$ term tends to zero as n tends to infinity for fixed b,k. The upper bound in Theorem 4 is derived using a similar argument as in Theorem 1. The proof of the lower bound draws heavily from the theory of the entropy of graphs introduced by Körner in earlier works. For the case of $b=k>3$, we can simplify the lower bound by setting $s=k-1$ in the minimization, (which corresponds to the bound in Fredman and Komlós [14]). The simplified lower bound can be shown to have an $\Omega\left(\log\log\left(n\right)\right)$ scaling term as well as a linear term in k when expressed in term of rate, thus it essentially combines the more elementary results on $R^*(k,n)$ from Theorems 2 and 3 into a single bound. For fixed k,n, the upper bound in Theorem 4 is a strictly decreasing function in b. In the massive access problem, b can be interpreted as the number of orthogonal slots available for transmission in the third phase. Intuitively, it takes less information to schedule users with no collision if more slots are available. As a final remark, note that perfect hashing families are also connected to a related, albeit different, random-access problem [18]–[20].

V. Concluding Remarks

This paper considers the problem of finding the minimum-rate feedback strategy for scheduling k out of n users in a massive random access scenario. Our main contributions are the formulation of this problem as a set-partition problem, and in showing the connection of this problem to the hypergraph covering problem and the minimum perfect hash function problem. In doing so, we provide elementary proofs that the optimal feedback strategy must have a rate that scales linearly in k but double logarithmically in n, i.e., $\Theta(\log\log(n))$. This is a significant feedback reduction as compared to the naive feedback strategy that uses $k\log(n)$ bits for scheduling or contention based strategies that implicitly require $\log(n)$ identification bits per user.

The connections to hypergraph covering and perfect hash function open the possibilities for leveraging results in combinatorics (e.g., [21]–[24]) for the explicit construction of practical feedback codes; this will be subject of future studies.

REFERENCES

[1] X. Chen, T. Chen, and D. Guo, "Capacity of Gaussian many-access channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3516–3539, June 2017.

[2] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? A survey of alternatives," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 4–16, 2014.

[3] L. Liu, E. G. Larsson, W. Yu, P. Popovski, C. Stefanovic, and E. De Carvalho, "Sparse signal processing for grant-free massive connectivity: a future paradigm for random access protocols in the internet of things," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 88–99, 2018.

[4] W. Yu, "On the fundamental limits of massive connectivity," in *Inf. Theory Appl. Work. (ITA)*, Feb 2017, pp. 1–6.

[5] Y. Polyanskiy, "A perspective on massive random-access," in *IEEE Int. Symp. Inf. Theory. (ISIT)*, 2017, pp. 2523–2527.

[6] G. Wunder, H. Boche, T. Strohmer, and P. Jung, "Sparse signal processing concepts for efficient 5G system design," *IEEE Access*, vol. 3, pp. 195–208, 2015.

[7] Z. Chen, F. Sohrabi, and W. Yu, "Sparse activity detection for massive connectivity," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1890–1904, 2018.

[8] A. Fengler, S. Haghighatshoar, P. Jung, and G. Caire, "Non-Bayesian activity detection, large-scale fading coefficient estimation, and unsourced random access with a massive MIMO receiver," 2019. [Online]. Available: https://arxiv.org/abs/1910.11266

[9] V. K. Amalladinne, J.-F. Chamberland, and K. R. Narayanan, "A coded compressed sensing scheme for uncoordinated multiple access," 2018. [Online]. Available: https://arxiv.org/abs/1809.04745

[10] L. G. Roberts, "ALOHA packet system with and without slots and capture," *SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, p. 28–42, Apr. 1975.

[11] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, 2 2011.

[12] R. Ahlswede and G. Dueck, "Identification via channels," *IEEE Trans. Inf. Theory*, vol. 35, no. 1, pp. 15–29, Jan 1989.

[13] M. Snir, "The covering problem of complete uniform hypergraphs," *Discrete Math.*, vol. 27, no. 1, pp. 103–105, 1979.

[14] M. L. Fredman and J. Komlós, "On the size of separating systems and families of perfect hash functions," *SIAM J. Algebr. Discret. Methods*, vol. 5, no. 1, pp. 61–68, 3 1984.

[15] J. Radhakrishnan, "Improved bounds for covering complete uniform hypergraphs," *Inf. Process. Lett.*, vol. 41, pp. 203–207, 1992.

[16] J. Körner, "Fredman–Komlós bounds and information theory," *SIAM J. Algebr. Discret. Methods*, vol. 7, no. 4, pp. 560–570, 1986.

[17] J. Körner and K. Marton, "New bounds for perfect hashing via information theory," *Eur. J. Comb.*, vol. 9, no. 6, pp. 523–530, 1988.

[18] T. Berger, "The poisson multiple-access conflict resolution problem," in *Multi-User Communication Systems*, G. Longo, Ed. Vienna, Austria: Springer, 1981, ch. 1, pp. 1–27.

[19] B. Hajek, "A conjectured generalized permanent inequality and a multiaccess problem," in *Open Problems in Communication and Computation*, T. M. Cover and B. Gopinath, Eds. New York, New York: Springer, 1987, ch. 4.11, pp. 127–129.

[20] J. Korner and K. Marton, "Random access communication and graph entropy," *IEEE Transactions on Information Theory*, vol. 34, no. 2, pp. 312–314, March 1988.

[21] N. Alon, "Explicit construction of exponential sized families of k-independent sets," *Discrete Math.*, vol. 58, no. 2, pp. 191–193, 1986.

[22] M. Atici, S. S. Magliveras, D. R. Stinson, and W. D. Wei, "Some recursive constructions for perfect hash families," *J. Comb. Des.*, vol. 4, no. 5, pp. 353–363, 1996.

[23] H. Wang and C. Xing, "Explicit constructions of perfect hash families from algebraic curves over finite fields," *J. Comb. Theory*, 2001.

[24] M. Atici, D. Stinson, and R. Wei, "A new practical algorithm for the construction of a perfect hash function," *J. Comb. Math. Comb. Comput.*, vol. 35, pp. 127–146, 2000.