The 38th Symposium on Information Theory
and its Applications (SITA2015)
Kojima, Okayama, Japan, Nov. 24–27, 2015

Doubly Irregular Coded Slotted ALOHA for Massive Uncoordinated Multiway Relay Networks

Ardimas Andi Purwita * Khoirul Anwar *

Abstract— Massive uncoordinated multiway relay networks (mu-mRN) is an mRN that can serve a massive number of users expecting to fully exchange information among them via a common relay. In this paper, we aim to improve normalized throughput of the mu-mRN using multiuser detection (MUD) technique with capability of $K > 1$. First, we present a network capacity bound of the mu-mRN with general K to investigate the theoretical limit of the network. Then, we search for many optimal degree distributions for the MUD-based mu-mRN. Second, we aim to improve the normalized throughput by 10× from the maximum normalized throughput of conventional systems. To achieve the goal, we propose the mu-mRN applying doubly irregular coded slotted ALOHA.

Keywords— Multiway relay networks, random access, codes-on-graphs, EXIT chart, optimization.

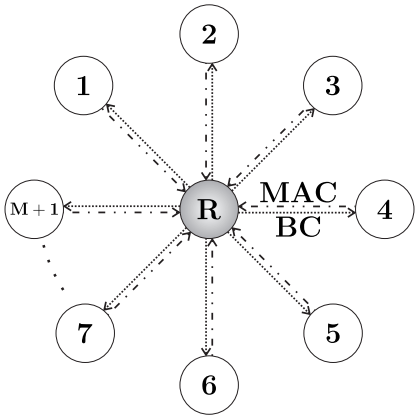


Figure 1. Multiway relay networks (mRN) with $(M + 1)$ users.

1 Introduction

In the near future, there will be billions of connected devices. Multiway relay networks (mRN) [1] is a common framework to model various communication systems having a massive number (more than a hundred) of users. Figure 1 shows the mRN with $(M + 1)$ users expecting to fully exchange data via a common relay. In the mRN, there are two different phases of transmission, i.e., multiple access (MAC) phase (where each user transmits data to the relay) and broadcast (BC) phase (where the relay broadcasts data to all users) as shown in Figure 1. This model is sufficient to describe various communication applications, e.g., satellite systems, ad-hoc disaster recovery networks, or sensor networks.

Let the massive number of users transmit data during the MAC phase at the same time slot. It requires a very low-rate code [2] having a high computational complexity. As an alternative, a *coordinated* scheduling technique can be used; however, it is also unpreferable due to its scheduling complexity [3]. In this paper, we discuss an *uncoordinated* transmission for the mRN serving the massive number of users, called massive uncoordinated mRN (mu-mRN). Recently, the uncoordinated transmission for mRN applying graph-based random access (RA) is briefly introduced in [4, 5].

The underlying notions of [4, 5] are that: (i) they define a *pair-of-time-slot* (PTS) consisting of an MAC phase followed by a BC phase, (ii) each user randomly transmits packet[1] at a given PTS based on a degree distribution, and (iii) at the same PTS, the relay always amplifies and forwards its received packet using amplify-and-forward (AF) protocol. The ideas enable adoption of irregular repetition slotted ALOHA (IRSA) [6] and coded slotted ALOHA (CSA) [7] into the mu-mRN. IRSA and CSA are an RA benefiting from successive interference cancelation (SIC) to resolve colliding packets. The SIC process can be represented by a bipartite graph; hence, we refer IRSA or CSA *graph-based RA*.

The representation of the SIC process assumes that each user or the relay has prior knowledge to which PTS each packet is sent. Practically, this can be done by inserting a pointer to show the position of each packet. The difference between IRSA and CSA lies on the type of *network encoding* (which is defined as packet-oriented linear block code) that is used; IRSA uses repetition codes, and CSA uses maximum distance separable (MDS) codes. IRSA's pointer shows to which PTS *replicas of a packet* are sent within a frame, and CSA's pointer indicates to which PTS *encoded packets* are sent within a frame.

By carefully choosing the degree distribution, IRSA can asymptotically achieve normalized throughput T (probability of successful packet times offered traffic) of 0.97 packets/slot (p/s). A well-known benchmark for the normalized throughput T is the maximum normalized throughput of the conventional slotted ALOHA (SA) [8] T_{SA}, where $T_{SA} = 1/\mathrm{e} \approx 0.37$ p/s. Therefore, the fact that the IRSA's normalized throughput T can approach one p/s is very encouraging. There are a lot of works have been devoted towards the graph-based technique since then, e.g., the CSA (which is the generalization of IRSA). Other works have been presented in [9, 10]. In this paper, we mainly focus on improving T multiple times using MUD technique to jointly decode colliding packets [11].

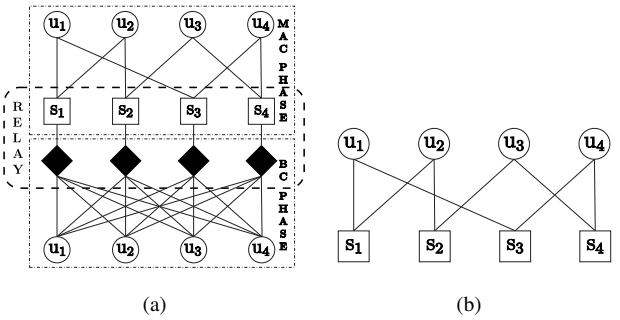
Our contributions are summarized as follows.

i. To the best of our knowledge, an mu-mRN applying

* School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahidai, Nomi, Ishikawa, 923-1292 Japan. Email : {ardimasp, anwar-k}@jaist.ac.jp.

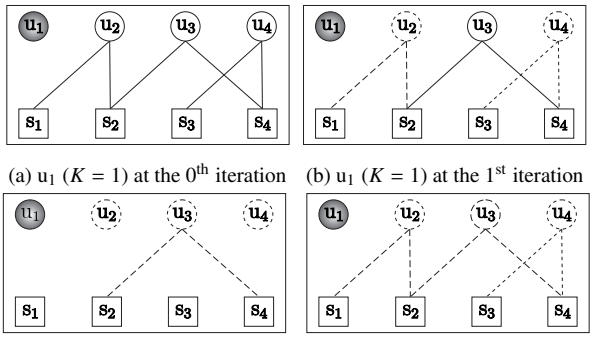
This work was supported by the Japan Society for the Promotion of Science (JSPS), Scientific Research KIBAN KENKYU (B) No. 25289113.

[1] Instead of data, we refer to packet in the context of the RA.



(a) (b)

Figure 2. A toy example of the mu-mRN. (a) A circle (◯), a square (□), and a black diamond (◆) describe a user, a PTS, and an amplifying factor, respectively. (b) A bipartite graph representation shown in Figure 2(a) at each user.



(a) u_1 ($K = 1$) at the 0^{th} iteration (b) u_1 ($K = 1$) at the 1^{st} iteration

(c) u_1 ($K = 1$) at the 2^{nd} iteration (d) u_1 ($K = 2$) at the 0^{th} iteration

Figure 3. SIC process with $K = 1$ and $K = 2$ at u_1 of the example in Figure 2 with an assumption that *replicas of a packet* are sent by each user.

graph-based RA with $K > 2$ has not yet been addressed in any literature. In this paper, we present a network capacity bound for the mu-mRN with general K.

ii. We present the optimal degree distributions of the mu-mRN applying graph-based RA, which are not discussed in [4, 5, 11].

iii. We aim to achieve multiple times improvement, e.g., $T = 10 \times T_{SA} = 3.7$ p/s. This goal can be intuitively achieved with $K = 4$. However, we find that as K increases, performances of the mu-mRN applying IRSA and CSA and their achievable bounds widen. Accordingly, the mu-mRN applying *optimal* IRSA or CSA cannot achieve the goal. To achieve the goal, we introduce the mu-mRN applying doubly irregular CSA (dir-CSA).

2 Graphical Representation of the mu-mRN

Each frame comprises a group of PTSs, and each PTS consists of an MAC phase and a BC phase, where every slot and frame transmissions are synchronized. We assume each packet is transmitted over erasure channel (error only happens because of colliding packets). This type of error may also happen if the transmission power is high enough to combat noise such that there is no bit-level error occurred.

Each frame can be represented by a bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{S}, \mathcal{E})$ consisting of a set $\mathcal{U}$ of $(M+1)$ user nodes (UNs) representing users, a set $\mathcal{S}$ of N slot nodes (SNs) representing PTSs, and a set $\mathcal{E}$ of edges. An edge connecting i-th UN and j-th SN represents an encoded packet transmission from i-th user at j-th PTS.

The mu-mRN using graph-based RA works as follows. At each frame during MAC phase, each user (u_m, $1 \le m \le (M+1)$) randomly transmits *encoded packets*[2] to a common relay based on a degree distribution $\mathbf{\Lambda} = \{\Lambda_h\}_{h=1}^{C}$ at given PTS (denoted as s_n, where $1 \le n \le N$). In the relay, the received packets are always amplified by factor A and forwarded to all users during BC phase. Figure 2(a) illustrates the toy example with $M = 3$ and $N = 4$.

[2] Note that the encoded packets are the generalization of replicas of packets.

Figure 2(b) shows the bipartite graph shown in Figure 2(a) at each user *before* subtracting its own packets from its received packets. The bipartite graph shown in Figure 2(a) can be simplified to the graph shown in Figure 2(b) since: (i) the AF protocol is used, (ii) the received packets at the relay are broadcast to all users, and (iii) a PTS comprises an MAC phase followed by a BC phase.

After subtracting its own packets, each user has different bipartite graph. The bipartite graph of user u_1 is depicted in Figure 3(a). Suppose IRSA is used so that *replicas of a packet* are sent instead of the *encoded packets*. We also assume $K = 1$ (without MUD) in this example. The SIC process in u_1 works as follows.

- In Figure 3(a), packets sent at s_1 and s_3 do not collide other packets.
- Thus, packets from u_2 and u_4 can be resolved during the 1^{st} iteration as shown in Figure 3(b).
- Since packets from u_2 and u_4 are known at the 2^{nd} iteration, there are no longer colliding packets at s_2 and s_4. Hence, a packet from u_3 can be successfully resolved as shown in Figure 3(c).

The SIC processes for other users work similarly. For CSA, *encoded packets* (which are output packets of a network encoder) are sent within a frame. For example, if the *encoded packets* are output packets of a network encoder using (3, 2) MDS code, it means that 2 original packets are encoded into 3 encoded packets. Then, to be able to recover the original packet, there must be at least 2 non-colliding packets.

Figure 3(d) shows the SIC processes having $K = 2$. In this case, we assume that each user can decode at most 2 colliding packets at a PTS. Therefore, at the 0^{th} iteration, all packets can be successfully decoded.

3 Network Encoding

3.1 System Model

In *network encoder* (NE), a code from a set of C codes denoted as $\{(n_h, k_h, \Lambda_h)\}_{h=1}^{C}$ with minimum distance at least 2 are mutually independently picked by all users, where

- n_h shows how many output packets of the h-th codes,
- k_h shows how many input packets of the h-th codes, and

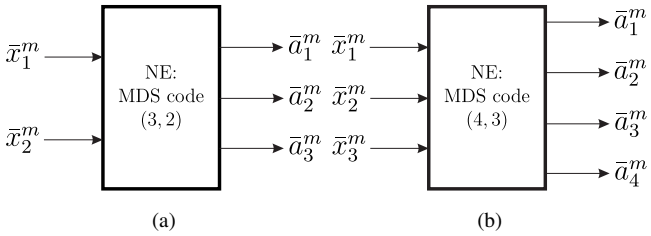


Figure 4. The network encoder of the mu-mRN with $\{(n_h, k_h, \Lambda_h)\}_1^2 = \{(3, 2, 0.5), (4, 3, 0.5)\}$, $M = 999$, and $C = 2$: (a) user $m \in \{1, 2, \ldots, 500\}$ picks the *first* codes ($h = 1$), i.e., $(n_1, k_1, \Lambda_1) = (3, 2, 0.5)$ and (b) user $m \in \{501, 502, \ldots, (M + 1)\}$ picks the *second* codes ($h = 2$), i.e., $(n_2, k_2, \Lambda_2) = (4, 3, 0.5)$.

- Λ_h shows a degree distribution of the h-th codes, where $\sum_{h=1}^{C} \Lambda_h = 1$ and $0 \leq \Lambda_h \leq 1, \forall \Lambda_h$.

Please note that the code $\{(n_h, k_h, \Lambda_h)\}_{h=1}^{C}$ is object of our optimization in designing the network encoding.

A toy example with the massive number of users $M = 999$ and $C = 2$ is illustrated in Figure 4, where $\bar{x}_l^m$ and $\bar{a}_l^m$ are the l-th packet of user m. In this example, there are only 2 set of codes ($C = 2$) with $\{\Lambda_1, \Lambda_2\} = \{0.5, 0.5\}$ and 1000 users[3] in Figure 4. The $1^{st}, 2^{nd}, \ldots,$ and 500^{th} users are assumed to pick the *first* codes which is $(n_1, k_1) = (3, 2)$, and the $501^{st}, 502^{nd}, \ldots,$ and 1000^{th} users are assumed to pick the *second* codes which is $(n_2, k_2) = (4, 3)$.

It is worth noting that for IRSA, $k_h = 1, \forall h$, since repetition codes are used. For CSA, the number of input packets of the network encoder is constant, i.e., $k_h = k, \forall h$, where $k \in \mathbb{Z}^+$ and $k > 1$. Here, we briefly introduce dir-CSA, where we let k_h be irregular. This additional irregularity is important since it makes the dir-CSA has wider class of codes. Consequently, the dir-CSA is shown later to have better performances than those of IRSA and CSA.

3.2 Asymptotic Analysis

Asymptotic performances are evaluated by setting $M \to \infty$, $N \to \infty$, and keeping the normalized offered traffic channel per user G constant, where

$$G = \frac{M \sum_{h=1}^{C} \Lambda_h k_h}{N}. \tag{1}$$

The asymptotic performance of IRSA, CSA, or dir-CSA depends on their degree distributions given K being constant. The degree distribution, then, can be used to see evolution of the SIC process in IRSA, CSA, or dir-CSA under asymptotic assumption using an EXIT chart. The EXIT chart displays evolution of average erasure probabilities emanating from both SNs and UNs, denoted as p and q respectively.

Based on [11] and [7]

$$p = 1 - e^{q\frac{G}{R_n}} \sum_{j=0}^{K-1} \frac{(q\frac{G}{R_n})^j}{j!} = f_s(q), \tag{2}$$

where network rate per user for each frame $R_n = \bar{k}/\bar{n}$, $\bar{k} = \sum_{h=1}^{C} \Lambda_h k_h$, and $\bar{n} = \sum_{h=1}^{C} \Lambda_h n_h$. The network rate R_n intuitively expresses how much total power required to transmit all packets in a frame. For example, IRSA with the network rate per user R_n of 0.2 means that a packet is retransmitted 5× in a frame by each user in average. Therefore, the lower the network rate, the higher the total transmit power required. We aim to achieve our target of $0.4 < R_n < 0.6$.

The average erasure probability from an SN is expressed as

$$q = \sum_{h=1}^{C} \lambda_h f_u^{(n_h, k_h)}(p) = f_u(p), \tag{3}$$

where $\lambda_h = \Lambda_h n_h / \bar{n}$. This equation is derived with an assumption that every UN's local processor uses maximum *a posteriori* decoding. The term $f_u^{(n_h, k_h)}(p)$ is called the average EXIT function of a type-h UN, which is expressed as

$$f_u^{(n_h, k_h)}(p) = \sum_{l=0}^{k_h - 1} \binom{n_h - 1}{l} (1 - p)^l p^{n_h - l - 1}, \tag{4}$$

such that

- for IRSA, $n_h = h$ and $k_h, \forall h$ (a special case for IRSA, (3) can be simplified into $q = \sum_{h=2}^{C} \lambda_h p^{h-1}$ [4])[4],
- for CSA, $n_h = h + k$, and
- for dir-CSA, $n_h = h + k_h$.

An EXIT chart analysis is based on two curves from (2) and (3), i.e., $1 - f_s(q)$ vs. $1 - q$ and $1 - f_u(p)$ vs. $1 - p$. Since $f_s(q) = p$ and $f_u(p) = q$, both curves can be drawn altogether in one chart.

The asymptotic threshold G^* is defined as the maximum value of G such that if $G < G^*$, all colliding packets can be successfully resolved. In other words,

$$G^* \triangleq \sup\{G \geq 0 : p_i \to 0 \text{ as } i \to \infty, p_0 = 1\} \tag{5}$$

$$= \sup\{G \geq 0 : q_i \to 0 \text{ as } i \to \infty, q_0 = 1\}, \tag{6}$$

where index i shows the iteration index of the SIC. Furthermore, for all $G < G^*$, G^* is equal to the maximum normalized throughput T in the asymptotic setting [12]. *Our goal is to achieve the threshold* $G^* = 3.7$ *p/s in the asymptotic setting*.

Using the EXIT chart, G^* is defined as the maximum value of G such that the two EXIT curves do not intersect each other by keeping the tunnel between the two curves remains open.

A network capacity bound can also be derived using the *area theorem* of the EXIT chart. A necessary condition for successful decoding is that the tunnel in the EXIT chart must be kept open, and the areas under the curves should satisfy

$$A_u + A_s < 1, \tag{7}$$

where $A_u = \int_0^1 f_u(p)dp$ and $A_s = \int_0^1 f_s(q)dq$.[5] Then, we can obtain the bound of the mu-mRN with MUD capability

[3] Remember the $(M + 1)$ users.

[4] Note that it is necessary to define $\Lambda_1 = 0$ for IRSA.

[5] This inequality is a necessary but not a sufficient condition. In practice, we also need $q < f_u^{-1}(q), \forall p, q \in (0, 1]$.

K as

$$R_n + \left(K\frac{R_n}{G} + \sum_{j=1}^{K-1} \frac{K-j}{j!}\left(\frac{G}{R_n}\right)^{j-1}\right)e^{\frac{-G}{R_n}} - K\frac{R_n}{G} < 0, \qquad (8)$$

such that $G > 0$, $R_n > 0$, and $K > 1$. For $K = 1$, the term $\sum_{j=1}^{K-1} \frac{K-j}{j!}\left(\frac{G}{R_n}\right)^{j-1} = 0$; thus, the bound becomes $R_n + \frac{R_n}{G}e^{\frac{-G}{R_n}} - \frac{R_n}{G} < 0$ such that $G > 0$ and $R_n > 0$.

3.3 Optimization using Differential Evolution (DE)

In (2), variable p depends on channel (the normalized offered traffic G), the MUD capability K, and the network rate per user R_n. Variable q primarily depends on the set of codes $\{(n_h, k_h, \Lambda_h)\}_{h=1}^{C}$. Since the normalized offered traffic G and the network rate per user R_n are obtained by designing the set of codes $\{(n_h, k_h, \Lambda_h)\}_{h=1}^{C}$ first, the variable p depends also on the set of codes $\{(n_h, k_h, \Lambda_h)\}_{h=1}^{C}$. Thus, it is important to carefully pick a good set of codes $\{(n_h, k_h, \Lambda_h)\}_{h=1}^{C}$.

We search the optimal set of code $\{n_h, k_h, \Lambda_h\}_1^C$ leading to a high threshold G^*. The optimization problem given C number of codes is denoted as

$$\begin{aligned} &\text{maximize} && G^* \\ &\text{subject to} && q < f_u^{-1}(q), \forall q \in (0,1] \\ & && 0 \le \Lambda_h \le 1, \forall h, \text{and } \sum_h^C \Lambda_h = 1. \end{aligned}$$

The inequality $q < f_u^{-1}(q), \forall q \in (0,1]$ is to guarantee that the two EXIT curves do not intersect each other. This optimization is carried out using the so-called differential evolution (DE) [13]. We use *EXIT-chart-based DE* to pick a good set of codes $\{(n_h, k_h, \Lambda_h)\}_{h=1}^{C}$. The DE setting is describes as follows. We choose 100 initial populations uniformly that satisfy $0 \le \Lambda_h \le 1, \forall h$, and $\sum_h^C \Lambda_h = 1$. We conduct the DE with representation *"DE/best/1-with-jitter"* and crossover constant $CR = 0.8$.[6] In choosing a good code, we define the allowed number of digits after decimal point for the degree distribution $\mathbf{\Lambda}$ equals 2, i.e., fractional-part number $FP = 2$, because the higher the number, the more PTS M or number of users N is required. Moreover, we prefer to relax the constraint R_n since if we include the R_n in our constraints, the threshold G^* given the same C will be lesser than that if we exclude the R_n.

4 Doubly Irregular CSA (dir-CSA)

First, we want to show an effect of increasing K as shown in Figure 5(a).[7] Note that this EXIT curve is drawn based on (2). As K increases the average erasure probability p is not purely exponential since there is the second level of iteration for SIC, i.e., the SIC works with iterated K. The second level of iteration is represented by sum equation on (2). It causes the EXIT curve for SN to be no longer convex. The consequence is that the optimal IRSA (with $K = 4$ and $C = 16$) cannot well match the curve as depicted in Figure 5(b). Even for the optimal CSA (with $K = 4$, $k = 2$, and

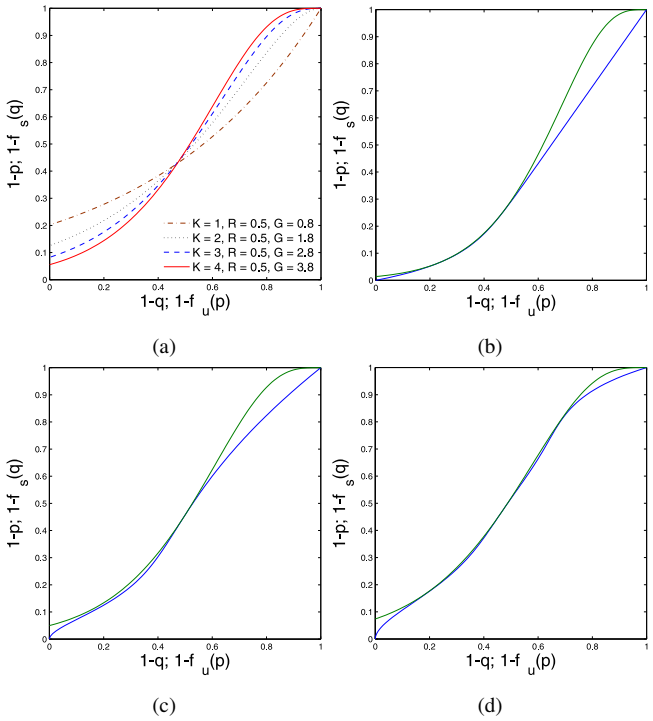


Figure 5. (a) SN EXIT curves showing that as K increases, the curve is no longer convex, (b) EXIT chart for IRSA with $K = 4$ and $C = 16$ ($\Lambda_{16}^{\mathrm{I},4}$), (c) EXIT chart for CSA with $K = 4$, $k = 2$ and $C = 16$ ($\Lambda_{2,16}^{\mathrm{CSA},4}$), and (d) EXIT chart for dir-CSA with $K = 4$ and $C = 8$ ($\Lambda_{8}^{\mathrm{DIC},4}$).

$C = 16$), which has wider class of codes than that of IRSA, the gaps are still quite wide, see Figure 5(c). By introducing more wider class of codes than those of IRSA and CSA, the optimal dir-CSA (with $K = 4$ and $C = 8$) can better match the SN EXIT curve as shown in Figure 5(d).

The aforementioned degree distributions in Figs. 5(b-d) are obtained by using the EXIT-chart-based DE explained in Section 3.3. In addition, we summarize many optimal set of codes $\{(n_h, k_h, \Lambda_h)\}_{h=1}^{C}$ for IRSA ($\Lambda_C^{\mathrm{I,K}}$), CSA ($\Lambda_{k,C}^{\mathrm{CSA,K}}$), and dir-CSA ($\Lambda_C^{\mathrm{DIC,K}}$) in Table 1.[8] For each set of code, the achievable threshold G^* and network rate per user R_n are given. Although we focus on the moderate R_n, $0.4 \le R_n \le 0.6$, several set of codes having $R_n < 0.4$ or $R_n > 0.6$ are also presented because we want to show a fact that as C increases, the achievable network rate per user R_n of a set of codes having C codes decreases; it means that a larger transmit power is required, while the increasing C shows relatively more complex network decoding.[9] The fact is important because it shows the superiority of dir-CSA than the others.

As seen in Table 1, the optimal IRSA with $K = 4$ and $C = 16$ ($\Lambda_{16}^{\mathrm{I},4}$) has the threshold G^* of 3.555 p/s, which is still below our target ($G^* = 3.7$ p/s). Even for CSA with $K = 4$, $k = 2$ and $C = 16$ ($\Lambda_{2,16}^{\mathrm{CSA},4}$), the achievable threshold G^* is 3.666 p/s. In fact, we have searched for several different C

[6] Readers who are interested in this configuration can refer to [13].

[7] All EXIT charts are drawn by setting $G = G^*$.

[8] Since our goal is $G^* = 3.7$ p/s, our search is only until $K = 4$.

[9] In this paper, we justify the complexity by only considering the number of network codes required. Practically, we also need to consider the complexity of each code.

Table 1. Optimal set of codes for IRSA ($\Lambda_C^{\mathrm{I,K}}$), CSA ($\Lambda_{k,C}^{\mathrm{CSA,K}}$), and dir-CSA ($\Lambda_C^{\mathrm{DIC,K}}$).

K	Label	$\{(\mathbf{n_h}, \mathbf{k_h}, \mathbf{\Lambda_h})\}_{\mathbf{h}}^{\mathbf{C}}$	$\mathbf{G}^*$	$\mathbf{R_n}$
4	$\Lambda_{22}^{\mathrm{DIC,4}}$	{(22, 21, 0.1), (22, 20, 0.16), (23, 19, 0.03), (22, 17, 0.02), (8, 2, 0.02), (30, 23, 0.01), (22, 13, 0.03), (29, 19, 0.01), (14, 3, 0.01), (35, 23, 0.04), (16, 2, 0.01), (31, 16, 0.02), (30, 13, 0.04), (29, 11, 0.03), (21, 2, 0.2), (22, 2, 0.17), (29, 8, 0.1)}	3.876	0.45722
	$\Lambda_{16}^{\mathrm{DIC,4}}$	{(18, 17, 0.24), (12, 8, 0.06), (22, 17, 0.03), (19, 13, 0.07), (11, 4, 0.04), (11, 3, 0.05), (16, 6, 0.05), (23, 12, 0.01), (14, 2, 0.09), (27, 14, 0.02), (23, 9, 0.06), (18, 2, 0.28)}	3.86	0.47887
	$\mathbf{\Lambda_8^{DIC,4}}$	**{(10, 9, 0.35), (5, 3, 0.01), (5, 2, 0.14), (10, 6, 0.02), (8, 2, 0.03), (16, 9, 0.09), (10, 2, 0.36)}**	**3.806**	**0.53135**
	$\mathbf{\Lambda_{2,16}^{CSA,4}}$	**{(3, 2, 0.88), (6, 2, 0.01), (9, 2, 0.01), (10, 2, 0.02), (11, 2, 0.01), (12, 2, 0.01), (15, 2, 0.01), (16, 2, 0.01), (17, 2, 0.01), (18, 2, 0.03)}**	**3.66**	**0.4717**
	$\Lambda_4^{\mathrm{DIC,4}}$	{(6, 5, 0.51), (4, 2, 0.01), (5, 2, 0.01), (6, 2, 0.47)}	3.647	0.59129
	$\Lambda_{4,16}^{\mathrm{CSA,4}}$	{(5, 4, 0.75), (6, 4, 0.01), (7, 4, 0.02), (8, 4, 0.03), (9, 4, 0.02), (10, 4, 0.04), (11, 4, 0.01), (12, 4, 0.02), (15, 4, 0.01), (19, 4, 0.01), (20, 4, 0.08)}	3.646	0.56657
	$\Lambda_{4,8}^{\mathrm{CSA,4}}$	{(5, 4, 0.81), (12, 4, 0.19)}	3.562	0.63191
	$\mathbf{\Lambda_{16}^{I,4}}$	**{(2, 1, 0.95), (16, 1, 0.05)}**	**3.555**	**0.37037**
	$\Lambda_{2,8}^{\mathrm{CSA,4}}$	{(3, 2, 0.85), (7, 2, 0.02), (8, 2, 0.02), (9, 2, 0.02), (10, 2, 0.09)}	3.54	0.50891
	$\Lambda_8^{\mathrm{I,4}}$	{(2, 1, 0.95), (8, 1, 0.05)}	3.438	0.43478
	$\Lambda_4^{\mathrm{I,4}}$	{(2, 1, 1)}	3.399	0.5
2	$\Lambda_{16}^{\mathrm{I,2}}$	{(2, 1, 0.86), (5, 1, 0.01), (7, 1, 0.03), (9, 1, 0.01), (10, 1, 0.01), (11, 1, 0.01), (12, 1, 0.01), (13, 1, 0.01), (15, 1, 0.05)}	1.875	0.30488
	$\Lambda_8^{\mathrm{DIC,2}}$	{(10, 9, 0.17), (4, 2, 0.09), (8, 5, 0.12), (9, 5, 0.13), (7, 2, 0.06), (8, 2, 0.04), (9, 2, 0.01), (10, 2, 0.38)}	1.872	0.44671
	$\Lambda_8^{\mathrm{I,2}}$	{(2, 1, 0.87), (8, 1, 0.13)}	1.858	0.35971
	$\Lambda_{2,8}^{\mathrm{CSA,2}}$	{(3, 2, 0.74), (4, 2, 0.02), (5, 2, 0.02), (8, 2, 0.01), (9, 2, 0.03), (10, 2, 0.18)}	1.839	0.43956
	$\Lambda_4^{\mathrm{DIC,2}}$	{(6, 5, 0.33), (4, 2, 0.07), (6, 2, 0.6)}	1.779	0.51024
	$\Lambda_4^{\mathrm{I,2}}$	{(2, 1, 0.81), (4, 1, 0.19)}	1.748	0.42017
1	$\Lambda_{16}^{\mathrm{I,1}}$	{(2, 1, 0.5), (3, 1, 0.11), (4, 1, 0.22), (5, 1, 0.01), (6, 1, 0.02), (10, 1, 0.01), (11, 1, 0.01), (12, 1, 0.01), (13, 1, 0.01), (14, 1, 0.01), (15, 1, 0.06), (16, 1, 0.03)}	0.949	0.22936
	$\Lambda_8^{\mathrm{I,1}}$	{(2, 1, 0.51), (3, 1, 0.26), (4, 1, 0.01), (7, 1, 0.01), (8, 1, 0.21)}	0.938	0.27855
	$\Lambda_{2,16}^{\mathrm{CSA,1}}$	{(3, 2, 0.34), (4, 2, 0.11), (5, 2, 0.26), (7, 2, 0.06), (8, 2, 0.05), (9, 2, 0.01), (12, 2, 0.01), (16, 2, 0.02), (17, 2, 0.01), (18, 2, 0.13)}	0.932	0.30211
	$\Lambda_{16}^{\mathrm{DIC,1}}$	{(6, 5, 0.15), (16, 13, 0.04), (12, 6, 0.01), (20, 13, 0.02), (22, 14, 0.02), (11, 2, 0.17), (18, 8, 0.04), (25, 14, 0.01), (20, 8, 0.06), (20, 7, 0.07), (16, 2, 0.08), (17, 2, 0.18), (18, 2, 0.15)}	0.93	0.29773
	$\Lambda_{2,8}^{\mathrm{CSA,1}}$	{(3, 2, 0.33), (4, 2, 0.34), (5, 2, 0.02), (9, 2, 0.01), (10, 2, 0.3)}	0.882	0.36101
	$\Lambda_8^{\mathrm{DIC,1}}$	{(5, 4, 0.13), (4, 2, 0.07), (10, 7, 0.08), (11, 7, 0.05), (10, 5, 0.04), (11, 5, 0.04), (9, 2, 0.18), (10, 2, 0.41)}	0.88	0.35633
	$\Lambda_4^{\mathrm{I,1}}$	{(2, 1, 0.5), (4, 1, 0.5)}	0.868	0.33333
	$\Lambda_{3,8}^{\mathrm{CSA,1}}$	{(4, 3, 0.26), (5, 3, 0.32), (6, 3, 0.01), (7, 3, 0.01), (8, 3, 0.02), (10, 3, 0.01), (11, 3, 0.37)}	0.833	0.42254
	$\Lambda_{2,4}^{\mathrm{CSA,1}}$	{(3, 2, 0.42), (5, 2, 0.01), (6, 2, 0.57)}	0.805	0.42283
	$\Lambda_{4,8}^{\mathrm{CSA,1}}$	{(5, 4, 0.21), (6, 4, 0.31), (7, 4, 0.02), (9, 4, 0.02), (11, 4, 0.03), (12, 4, 0.41)}	0.784	0.4717

and k; none of the results can achieve the threshold $G^* = 3.7$ p/s. Increasing irregularity of CSA (dir-CSA) can improve the performances of IRSA and CSA. With only $K = 4$ and $C = 8$, dir-CSA ($\Lambda_8^{\mathrm{DIC,4}}$) can achieve the threshold $G^* = 3.806$ p/s. This superiority becomes clearer since the dir-CSA $\Lambda_8^{\mathrm{DIC,4}}$ has the highest R_n and the smallest C of the IRSA $\Lambda_{16}^{\mathrm{I,4}}$ and the CSA $\Lambda_{2,16}^{\mathrm{CSA,4}}$, which means that the total transmit power of the dir-CSA is the lowest of the others and the network decoding of dir-CSA is relatively less complex than the others. In addition, the network rate per user R_n of dir-CSA $\Lambda_8^{\mathrm{DIC,4}}$ lies between 0.4 and 0.6, which fulfills our target.

Relative position of the optimal IRSA, CSA, and dir-CSA shown in Table 1 and their bounds obtained from (8) are depicted in Figure 6. The figure strengthens our analysis on the asymptotic analysis until $K = 4$ to achieve the threshold $G^* = 3.7$ p/s. For example, if we want to achieve $G^* = 2.9$ p/s, then it is easy for IRSA or CSA with $K = 4$. In this case, it is highly possible that we can find a code capable of achieving the threshold $G^* = 2.9$ p/s even without op-

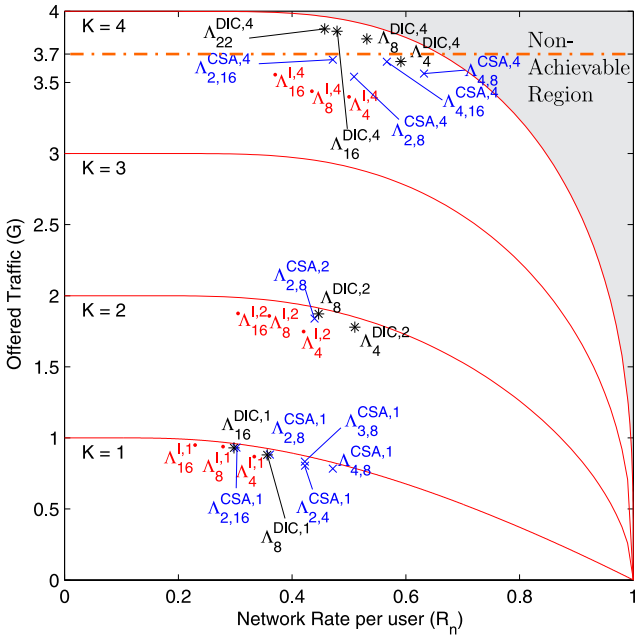


Figure 6. Threshold G^* shown in Table 1 vs. network rate per user R_n for IRSA ($\Lambda_C^{I,K}$), CSA ($\Lambda_{k,C}^{CSA,K}$), and **dir-CSA** ($\mathbf{\Lambda}_{\mathbf{C}}^{\mathbf{DIC,K}}$) with $K \in \{1, 2, 4\}$.

timizing the set of codes of IRSA or CSA with $K = 4$. However, there is the penalty of K since in fact we can achieve the threshold $G^* = 2.9$ p/s with $K = 3$ and carefully choosing a good set of codes.

As shown in Figure 6, the superiority of dir-CSA in achieving the threshold $G^* = 3.7$ p/s compared to the others becomes more apparent since only $\Lambda_{22}^{DIC,4}$, $\Lambda_{16}^{DIC,4}$, and $\Lambda_{8}^{DIC,4}$ can achieve $G^* = 3.7$ p/s at moderate network rate per user R_n. With the same R_n, dir-CSA is asymptotically better than both IRSA and CSA for $1 \leq K \leq 4$. This shows that dir-CSA has lower total power transmission while having better performance than the others.

5 Conclusions

This paper focussed on the application of graph-based RA, i.e., IRSA and CSA, to the mu-mRN. In this paper, we have derived the theoretical network capacity bound for the mu-mRN with general MUD capability K. We searched for many optimal degree distributions of IRSA and CSA with $K \in \{1, 2, 4\}$. We found that the gap between the threshold G^* of the optimal IRSA or CSA and their bounds widened as K increased. Correspondingly, the 10× improvement ($T = 3.7$ p/s) was not achievable by IRSA and CSA. Therefore, we introduced an improvement of CSA, namely dir-CSA having a wider class of codes than that of the CSA. The mu-mRN applying dir-CSA with $K = 4$ could achieve maximum normalized throughput $T = G^* = 3.806$ p/s while the optimal IRSA and CSA could not achieve it even with lower R_n and higher number of codes C (higher total transmit power and higher complexity). This result shows that our proposed system has better performances, lower complexity, and lower total power transmission than the conventional systems.

References

[1] D. Gunduz, A. Yener, A. Goldsmith, and H. Poor, "The multiway relay channel," *Information Theory, IEEE Trans. on*, vol. 59, no. 1, pp. 51–63, Jan 2013.

[2] A. Viterbi, "Very low rate convolution codes for maximum theoretical performance of spread-spectrum multiple-access channels," *Selected Areas in Comm., IEEE Journal on*, vol. 8, no. 4, pp. 641–649, May 1990.

[3] C. Zhai, W. Zhang, and G. Mao, "Uncoordinated cooperative communications with spatially random relays," *Wireless Comm., IEEE Trans. on*, vol. 11, no. 9, pp. 3126–3135, September 2012.

[4] K. Anwar and M. N. Hasan, "Uncoordinated transmissions in multi-way relaying systems," in *SCC 2015; 10th International ITG Conf. on Systems, Communications and Coding; Proceedings of*, Feb 2015, pp. 1–5.

[5] M. N. Hasan and K. Anwar, "Massive uncoordinated multiway relay networks with simultaneous detections," in *IEEE ICC 2015 - Workshop*, London, United Kingdom, Jun. 2015.

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

[7] E. Paolini, G. Liva, and M. Chiani, "Graph-based random access for the collision channel without feedback: Capacity bound," in *GLOBECOM, 2011 IEEE*, Dec 2011, pp. 1–5.

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

[9] G. Liva, E. Paolini, M. Lentmaier, and M. Chiani, "Spatially-coupled random access on graphs," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, July 2012, pp. 478–482.

[10] C. Stefanovic, P. Popovski, and D. Vukobratovic, "Frameless ALOHA protocol for wireless networks," *Comm. Letters, IEEE*, vol. 16, no. 12, pp. 2087–2090, December 2012.

[11] M. Ghanbarinejad and C. Schlegel, "Irregular repetition slotted ALOHA with multiuser detection," in *Wireless On-demand Network Systems and Services (WONS), 2013 10th Annual Conference on*, March 2013, pp. 201–205.

[12] E. Paolini, G. Liva, and M. Chiani, "High throughput random access via codes on graphs: Coded slotted ALOHA," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–6.

[13] R. Storn and K. Price, "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.