Vehicular Massive Multiway Relay Networks Applying Graph-Based Random Access

Ardimas Andi Purwita and Khoirul Anwar

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

***Abstract*—Recent vehicular communications problems entail various aspects, such as rapidly changing topologies and large number of users. We consider massive uncoordinated multiway relay networks (mu-mRN) that can serve a massive number of users expecting to fully exchange information among them via a common multiway relay. Relay vehicular networks is one of potential applications of the mu-mRN. In this paper, we are interested in improving normalized throughput T of the network using multiuser detection (MUD) capability of $K > 1$ based on the graph-based random access to flexibly adapt the topology changes. First, we present a network capacity bound of the mu-mRN with general K to investigate the theoretical limit of the networks. Then, we search for many optimal and practical degree distributions for each theoretical bound. Second, we aim to improve the normalized throughput by $10\times$ from the maximum normalized throughput of conventional systems. To achieve the goal, we propose the mu-mRN applying doubly irregular coded slotted ALOHA. We also propose an optimal practical encoding of the mu-mRN to closely approach the target with finite number of users.**

I. Introduction

In the near future, there will be billions of connected devices driven by the internet-of-things supported by machine-to-machine communication services. A forecast indicates that the number will reach 50 billion in 2020 [1]. A common framework to model various communication systems having a massive number (more than a hundred) of users is multiway relay networks (mRN) [2]. The potential applications of the mRN are, for example, intelligent networks for transportation systems, vehicle-to-vehicle communications, ad-hoc network for devastated areas, and satellite communication systems.

Fig. 1 shows $(M+1)$ 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 indicated by dash-dotted and dotted lines in Fig. 1.

If massive number of users are allowed to transmit during the MAC phase at one time slot, a very low-rate code is required, which also requires a high computational complexity of decoding. As an alternative, a *coordinated* scheduling technique can be used; however, it is also unpreferable due to the nature of massive number of users. Uncoordinated transmissions are preferable to avoid difficulties in transmission scheduling. 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 notion of the uncoordinated transmission for mRN applying graph-based random access (RA) is briefly introduced in [3], [4].

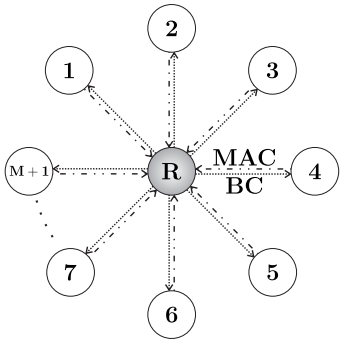


Fig. 1. Multiway relay networks (mRN) with $(M+1)$ users.

In [3], [4], they define a *pair-of-time-slot* (PTS) consisting of MAC phase followed by BC phase. Then, each user randomly transmits packet[1] at a given PTS based on a degree distribution. At the same PTS, the relay always amplifies and forwards its received packet using amplify-and-forward (AF) protocol. These ways enable adoption of irregular repetition slotted ALOHA (IRSA) [5] and coded slotted ALOHA (CSA) [6] 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 as *graph-based RAs*.

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); IRSA

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.

uses repetition codes, and CSA uses maximum distance separable (MDS) codes, e.g., Reed-Solomon (RS) code. IRSA's pointer shows to which PTS *replicas of a packet* are sent within a frame. CSA's pointer indicates to which PTS *encoded packets* are sent within a frame.

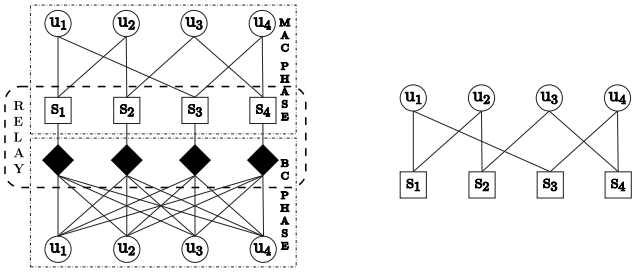
By carefully choosing the degree distribution, IRSA can asymptotically achieve a normalized throughput T (probability of successful packet times offered traffic) of 0.97 packets/slot (p/s). A well-known benchmark for the normalized throughput T is the maximum normalized throughput of the conventional slotted ALOHA (SA) 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). In this paper, we mainly focus on improving T multiple times using MUD technique to jointly decode colliding packets [7].

Our contributions are summarized as follows. **(i)** To the best of our knowledge, an mu-mRN applying 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 is presented in this paper. **(ii)** We present the optimal degree distributions of the mu-mRN applying graph-based RA, which are not discussed in [3], [4], [7]. **(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). **(iv)** For $K > 2$, iterative spatial demapping (IDM) [8] used in [4] becomes very complex since it considers all possible symbols for K users. Alternatively, we can adopt the code-division-multiple-access (CDMA). However, interleave-division-multiple-access (IDMA) [9] is a strong candidate to replace the CDMA due to its better performances. Therefore, instead of IDM and CDMA, we use the IDMA, which is further combined with the AF protocol in the mu-mRN. Optimality in physical encoding is also investigated in this paper to approach multiway relay channel (mRC) capacity [2] based on extrinsic information transfer (EXIT) chart [10].

II. Graphical Representation of the mu-mRN

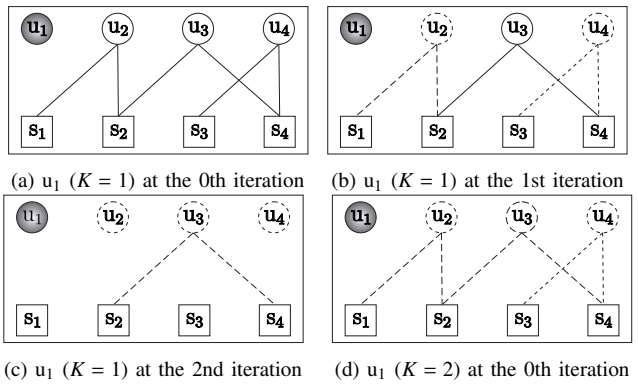
Each frame comprises a group of PTSs, and each PTS consists of an MAC phase and a BC phase, where every transmission is slot- and frame-synchronous. We assume each packet is transmitted over erasure channel (error only happens because of colliding packets). This type of error may also happen if the transmission 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 time slots, 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 time slot.



(a) (b)

Fig. 2. An example of the mu-mRN. (a) A circle (◯), a square (□), and a black diamond (◆) describe a user, a time slot, and an amplifying factor, respectively. (b) A bipartite graph representation shown in Fig. 2(a) at each user. A square (□) here represents a PTS instead of a single time slot.



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

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

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

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

Fig. 2(b) shows the bipartite graph shown in Fig. 2(a) at each user *before* subtracting its own packets from its received packets. The bipartite graph shown in in Fig. 2(a) can be simplified to the bipartite graph shown in Fig. 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 a time slot in MAC phase followed by a time slot in BC phase.

After subtracting its own packets, each user has different bipartite graph. The bipartite graph of user u_1 is depicted in Fig. 3(a). Suppose that *replicas of a packet* are sent instead of the *encoded packets*; in other words, IRSA is used. We also assume $K = 1$ (without MUD) in this example. The SIC process in u_1 works as follows. In Fig. 3(a), packets that are sent at s_1 and s_3 do not collide with other packets. Thus, packets from u_2 and u_4 can be resolved during the 1st iteration as shown in Fig. 3(b). Since packets from u_2 and u_4 are known

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

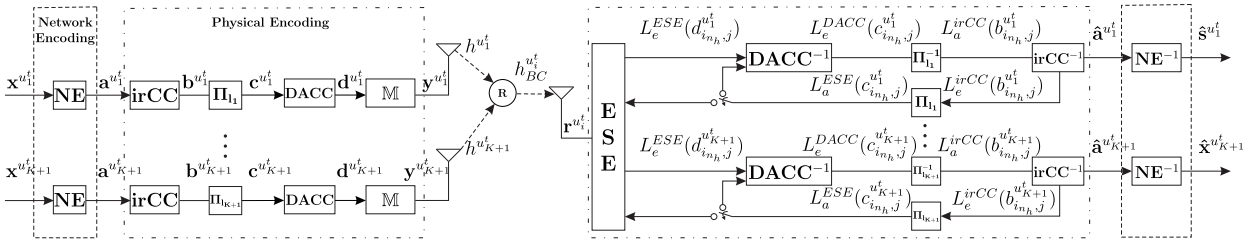


Fig. 4. Transmitter and receiver structures of the mu-mRN with $(K+1)$ users transmitting at a certain PTS.

at the 2nd 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 Fig. 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. In addition, Fig. 3(d) shows the SIC processes having $K = 2$

III. System Model

Fig. 4 describes the transmitter and receiver structures with MUD capability K, where $K << M$ and M denotes massive number of users. The left part of Fig. 4 depicts transmitter structure of the mu-mRN with $(K+1)$ users transmitting at the same PTS.[3] For *network encoder*, each user mutually independently picks 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, where n_h shows the number of output packets of the h-th codes, k_h shows the number of input packets of the h-th codes, and Λ_h shows a degree distribution of the h-th codes, where $\sum_{h=1}^{C} \Lambda_h = 1$ and $0 \le \Lambda_h \le 1$. Please note that the code $\{((n_h, k_h), \Lambda_h)\}_{h=1}^{C}$ is object of our optimization in designing the network encoding.

At a certain PTS t, $(K+1)$ users transmit their packets. Let define an ordered set $\mathcal{U}^t$ as a set of all users that transmit at the PTS t, and an element in the set $\mathcal{U}^t$ is denoted u_i^t, where $1 \le i \le |\mathcal{U}^t|$. Then, the code $(k_{h,u_i^t}, n_{h,u_i^t})$ denotes a code picked by the u_i^t-th user. For indexing purpose, we define variables $1 \le i_{k_{h,u_i^t}} \le k_{h,u_i^t}$ and $1 \le i_{n_{h,u_i^t}} \le n_{h,u_i^t}$. Input packet sequence for the u_i^t-th user is denoted as $\mathbf{x}^{u_i^t} = \{\bar{x}_1^{u_i^t}, \bar{x}_2^{u_i^t}, \ldots, \bar{x}_{k_{h,u_i^t}}^{u_i^t}\}$, where each packet $\bar{x}_{i_{k_{h,u_i^t}}}^{u_i^t}$ has block length L-bit. Either repetition code for IRSA or RS code for CSA is used to encode k_h packets into n_h packets. The n_h-packet form output packet sequence for the network encoder denoted as $\mathbf{a}^{u_i^t} = \{\bar{a}_1^{u_i^t}, \bar{a}_2^{u_i^t}, \ldots, \bar{a}_{n_{h,u_i^t}}^{u_i^t}\}$. Note that each encoded vector $\bar{a}_{i_{n_{h,u_i^t}}}^{u_i^t}$ has block length L-bit as well.

An example with $M = 999$ and $C = 2$ is illustrated in Fig. 5. There are only 2 set of codes ($C = 2$) with $\{\Lambda_1, \Lambda_2\} = \{0.5, 0.5\}$ and 1000 users in Fig. 5. The 1st, 2nd, ..., and 500th users $(u_1, u_2, \ldots, u_{500})$ are assumed to pick the *first* code which is $(n_1, k_1) = (3, 2)$, and the 501st, 502nd, ..., and 1000th

[3]An additional user is a remainder that each user can subtract its own signal from its received signal.

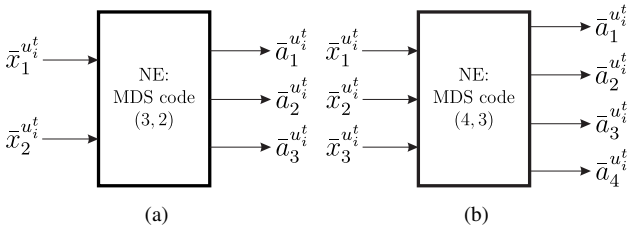


Fig. 5. The network encoder of the mu-mRN with $\{((n_h, k_h), \Lambda_h)\}_1^2 = \{((3,2), 0.5), ((4,3), 0.5)\}$, $M = 999$, and $C = 2$: (a) the *first* codes ($h = 1$), i.e., $(n_1, k_1, \Lambda_1) = (3, 2, 0.5)$ and (b) the *second* codes ($h = 2$), i.e., $((n_2, k_2), \Lambda_2) = ((4, 3), 0.5)$.

users $(u_{501}, u_{502}, \ldots, u_{1000})$ are assumed to pick the *second* code, which is $(n_2, k_2) = (4, 3)$. Then at a certain PTS, for example $t = 1$, the 1st, 5th, 7th, 100th and 547th users transmit their packets. In this case $\mathcal{U}^1 = \{u_1, u_5, u_7, u_{100}, u_{547}\}$, where $u_1^1 = u_1$, $u_2^1 = u_5$, $u_3^1 = u_7$, $u_4^1 = u_{100}$, and $u_5^1 = u_{547}$. Since u_1 picks the first code, its packet sequences are $\mathbf{x}^{u_1} = \{\bar{x}_1^{u_1}, \bar{x}_2^{u_1}\}$ and $\mathbf{a}^{u_1} = \{\bar{a}_1^{u_1}, \bar{a}_2^{u_1}, \bar{a}_3^{u_1}\}$ as shown in Fig. 5(a). Meanwhile, the packet sequences for u_{547} are $\mathbf{x}^{u_{547}} = \{\bar{x}_1^{u_{547}}, \bar{x}_2^{u_{547}}, \bar{x}_3^{u_{547}}\}$ and $\mathbf{a}^{u_{547}} = \{\bar{a}_1^{u_{547}}, \bar{a}_2^{u_{547}}, \bar{a}_3^{u_{547}}, \bar{a}_4^{u_{547}}\}$ as shown in Fig. 5(b).

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 has better performances than those of IRSA and CSA.

In *physical encoder*, each vector $\bar{a}_{i_{n_{h,u_i^t}}}^{u_i^t}$, is then encoded by an irregular convolutional code (irCC) [11] into J-tuple $\{b_{i_{n_h}, j}^{u_i^t}\}$, where $1 \le j \le J$ and J is the block length.[4] IrCC comprises several subcodes, and each subcode in irCC has different rate R_f with $1 \le f \le F$ and F is the number of subcodes forming the irCC. Each subcode, then, encodes L-bit sequence into $(a_f \times J)$-bit encoded sequence, where a_f denotes percentage of the f-th subcode in the J-bit sequence, $\sum_{f \in F} a_f = 1$ s.t. $0 \le a_f \le 1$ for $\forall f$. The set of pair $\{(R_f, a_f)\}_{f=1}^{F}$ is object of our optimization in the physical encoding.

At each user, the bit sequence $\mathbf{b}^{u_i^t}$ is then interleaved by interleaver $\Pi_{u_i^t}$ to obtain bit sequence $\mathbf{c}^{u_i^t}$. Since we apply IDMA, the interleaver $\Pi_{u_i^t}$ must be unique for every user. We assume that all interleavers are both independent and random.

[4]Note that without loss of generality, for the physical encoder we use notation $\{b_{i_{n_h}, j}^{u_i^t}\}$ instead of $\{b_{i_{n_{h,u_i^t}}, j}^{u_i^t}\}$.

Doped-unity-rate convolutional code (DACC) encodes a bit sequence $\mathbf{c}^{u_i^t}$ into bit sequence $\mathbf{d}^{u_i^t}$. DACC is a systematic and recursive convolutional code with generator polynomial (GP) $([3,2]_8)$ [8]; however, every Q-th bit output is replaced by its accumulated bit. Because the DACC's rate is unity, rate for physical encoder is $R_I = \sum_{f\in F} a_f R_f$.

Afterward, the bit sequence $\mathbf{d}^{u_i^t}$ is modulated by binary-phase-shift-keying (BPSK) mapper $\mathbb{M}$ into symbol sequence $\mathbf{y}^{u_i^t} \in \{1,-1\}$.[5] Each symbol sequence $\mathbf{y}^{u_i^t}$, where $\mathbf{E}[|\mathbf{y}^{u_i^t}|^2] = 1$, is then transmitted to the multiway relay R over a channel with gain $h^{u_i^t}$. We assume a single carrier transmission in our system.

The relay always amplifies its received packets during the MAC phase and forwards them to all users during the BC phase. The received signal at the relay is $r_R = \sum_{i=1}^{K+1} h^{u_i^t}\sqrt{P^{u_i^t}}\mathbf{y}^{u_i^t} + n_R$, where n_R is zero mean additive white Gaussian noise (AWGN) with σ_R^2 and $P^{u_i^t}$ being variance and transmit power of the u_i^t-th user, respectively. Amplifying factor A of the relay is expressed by $A = P_R/|r_R|^2 = P_R/(\sum_{i=1}^{K+1}(h^{u_i^t})^2 P^{u_i^t} + \sigma_R^2)$, where P_R is transmit power of the relay. The received packet in each user after subtracting its own packet from them is denoted as $r^{u_i^t} = h_{BC}^{u_i^t}\sqrt{A}\left(\sum_{t=1,t\neq i}^{K+1} h^{u_i^t}\sqrt{P^{u_i^t}}\mathbf{y}^{u_i^t}\right) + h_{BC}^{u_i^t}\sqrt{A}n_R + n^{u_i^t}$, where $h_{BC}^{u_i^t}$ where $h_{BC}^{u_i^t}$ is channel gain of vehicular channel from the relay to u_i^t-th user during BC phase, and $n^{u_i^t}$ is zero mean AWGN in u_i^t-th with variance $\sigma_{u_i^t}^2$. For the sake of simplicity, we assume all channel gains and all AWGN variances are unity, i.e., $h^{u_i^t} = h_{BC}^{u_i^t} = 1$ and $\sigma_R^2 = \sigma_{u_i^t}^2 = 1, \forall u_i^t$, and each transmit power is P.

The receiver consists of an elementary signal estimator (ESE) [9], interleavers, deinterleavers, and soft decoders for irCC and DACC. The ESE computes extrinsic log-likelihood ratio (LLR) for i^{n_h}-th packet and j-th bit of bit sequence $\mathbf{d}^{u_i^t}$, $d_{i^{n_h},j}^{u_i^t}$, denoted as $L_e^{ESE}(d_{i^{n_h},j}^{u_i^t})$. Then, an iterative process between the ESE and both soft decoders irCC and DACC is invoked.

At last, after each $\hat{\mathbf{a}}^{u_i^t}$ is estimated in physical decoding, each estimated user's packet $\hat{\mathbf{x}}^{u_i^t}$ is further decoded by network decoder (NE^{-1}), i.e., decoder for repetition code or decoder for RS code. Note that there is no iteration between the network and physical decoders.

IV. Physical Encoding

The irCC is formed by F subcodes with each subcode having rate R_f, $\forall f$. Each subcode also contributes to $(a_f \times J)$-bit encoded output of the irCC. Designing the irCC is a matter of how to pick the set of pair $\{(R_f, a_f)\}_{f=1}^F$ such that the EXIT [10] curve of irCC well matches the EXIT curve of DACC.

A mutual information between bit sequence x and its corresponding LLR $L(x)$ is denoted as $I(x;L(x))$. Useful notations at u_i^t-th user to draw the EXIT chart are expressed as[6] $\mathrm{I}_{\mathrm{A}}^{\mathrm{irCC}} \triangleq I(\mathbf{b}^{u_i^t}; L_a^{irCC}(\mathbf{b}^{u_i^t}))$, $\mathrm{I}_{\mathrm{E}}^{\mathrm{irCC}} \triangleq I(\mathbf{b}^{u_i^t}; L_e^{irCC}(\mathbf{b}^{u_i^t}))$, $\mathrm{I}_{\mathrm{A}}^{\mathrm{DACC}} \triangleq I(\mathbf{c}^{u_i^t}; L_a^{DACC}(\mathbf{c}^{u_i^t}))$, $\mathrm{I}_{\mathrm{E}}^{\mathrm{DACC}} \triangleq I(\mathbf{c}^{u_i^t}; L_e^{DACC}(\mathbf{c}^{u_i^t}))$. For $f-$th subcode with *apriori* LLR I_a and extrinsic LLR I_e, the EXIT chart can be drawn as a transfer function of $I_e = T_f(I_a)$. Transfer function for the irCC is then denoted as $T^{irCC}(I_a) = \sum_{f\in F} a_f T_f(I_a)$.

All EXIT chart subcodes are drawn from the same mother code by puncturing. In this paper, we use a mother code with GP$([57,65,71,73,75]_8)$ having free distance of 22 so that the code has a steep upward slope at early iterations ($0 \le \mathrm{I}_{\mathrm{E}}^{\mathrm{irCC}} \le 0.1$).

We are inspired by [12] to define several acceptable gaps ϵ between $\mathrm{I}_{\mathrm{A}}^{\mathrm{DACC}}$ and $\mathrm{I}_{\mathrm{E}}^{\mathrm{irCC}}$. Let define index w, where $w \in \{1,2,\ldots,W\}$, and $W \in \mathbb{Z}^+$. $\mathrm{I}_{\mathrm{E,w}}^{\mathrm{irCC}}$ and $\mathrm{I}_{\mathrm{A,w}}^{\mathrm{DACC}}$ denotes as the w-th sample of $\mathrm{I}_{\mathrm{E}}^{\mathrm{irCC}}$ and $\mathrm{I}_{\mathrm{A}}^{\mathrm{DACC}}$, respectively. Let define $\mathrm{I}_{\mathrm{E,w}}^{\mathrm{irCC}} = \sum_{f\in F} a_f \mathrm{I}_{\mathrm{E,w}}^{\mathrm{irCC,f}}$, where $\mathrm{I}_{\mathrm{E,w}}^{\mathrm{irCC,f}}$ is extrinsic LLR of f-th subcode of the irCC at w-th sample. Therefore, the optimization problem is defined as

$$\begin{aligned} \text{maximize} \quad & \sum_{f\in F} a_f R_f \\ \text{subject to} \quad & \left(\sum_{f\in F} a_f \mathrm{I}_{\mathrm{E,w}}^{\mathrm{irCC,f}}\right) - \mathrm{I}_{\mathrm{A,w}}^{\mathrm{DACC}} > \epsilon_w^{R_f}, \forall w \text{ and } f \\ & 0 < a_f \le 1, \forall f, \text{and } \sum_{f\in F} a_f = 1. \end{aligned}$$

This optimization is carried out using linear programming (LP). For the sake of simplicity, we define $\epsilon_w^{R_f} = \epsilon_w = 0.001, \forall f$. The value of $\mathrm{I}_{\mathrm{A,w}}^{\mathrm{irCC}}$, $\forall w$ and f, can be obtained by a curve matching technique. Furthermore, it is important to decide allowed number of digits after decimal point, i.e., fractional-part number (FP), of the coefficient a_f. In this paper, we define $FP = 2$.

For the LP, we define $W = 100$. Then, we get the optimal set of pair $\{(R_f, a_f)\}_{f=1}^5 = \{(\frac{1}{5}, 0.45), (\frac{2}{3}, 0.12), (\frac{2}{5}, 0.09), (\frac{1}{3}, 0.1), (\frac{3}{11}, 0.09), (\frac{1}{2}, 0.15)\}$. Its corresponding bit-error-rate (BER) is shown later in Section VII, which explains our performance results.

V. Network Encoding

A. Asymptotic Analysis

Asymptotic performances are evaluated by setting $M \to \infty$ and $N \to \infty$ and keeping the normalized offered traffic channel per user (G) constant, where $G = \frac{M\sum_{h=1}^{C}\Lambda_h k_h}{N}$.

The asymptotic performance of IRSA, CSA, or dir-CSA depends on their degree distributions given K being constant. Evolution of the SIC process in IRSA, CSA, or dir-CSA under asymptotic assumption for given degree distribution can be seen using an EXIT chart.[7] The EXIT chart displays evolution of average erasure probabilities emanating from both SNs and UNs, denoted as p and q, respectively.

Based on [6] and [7]

$$p_i = 1 - e^{q_{i-1}\frac{G}{R_n}} \sum_{j=0}^{K-1} \frac{(q_{i-1}\frac{G}{R_n})^j}{j!} = f_s(q_{i-1}), \tag{1}$$

[5] Extension to higher order modulation is straightforward.

[6] The EXIT chart for each user is identic.

[7] Note that there two different type of EXIT charts discussed in this paper, i.e., the one discussed in Section IV for physical encoding and the one discussed in this section for network encoding.

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 indirectly expresses how much total power required to transmit all packets in a frame. For example, IRSA with the network rate R_n 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 $0.4 \leq R_n \leq 0.6$.

The average erasure probability from a UN is expressed as

$$q_i = \sum_{h=1}^{C} \lambda_h f_u^{(n_h,k_h)}(p_{i-1}) = f_u(p_{i-1}), \tag{2}$$

where $\lambda_h = \Lambda_h n_h/\bar{n}$ and index i shows the iteration index of the SIC. This equation is derived with an assumption that the proposed physical encoder can decode each non-colliding packet successfully. The term $f_u^{(n_h,k_h)}(p)$ is called the average EXIT function of a type-h UN, which is expressed as

$$f_u^{(n_h,k_h)}(p) = \sum_{l=0}^{k_h-1} \binom{n_h-1}{l} (1-p)^l p^{n_h-l-1}, \tag{3}$$

such that

- for IRSA, $n_h = h$ and $k_h, \forall h$ (a special case for IRSA, (2) can be simplified into $q_i = \sum_{h=2}^{C} \lambda_h p^{h-1}$ [3])[8],
- for CSA, $n_h = h + k$, and
- for dir-CSA, $n_h = h + k_h$.

An EXIT chart displays two curves based on (1) and (2), 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 of the SIC process G^* is defined as the maximum value of G such that if $G < G^*$, all colliding packets can be successfully resolved. Furthermore, for all $G < G^*$, G^* is equal to the maximum normalized throughput T in the asymptotic setting. Therefore, *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. In other words, 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 $A_u + A_s < 1$, where $A_u = \int_0^1 f_u(p)dp$ and $A_s = \int_0^1 f_s(q)dq$.[9] Then, we can obtain the bound of the mu-mRN with MUD capability K as

$$R_n + \left(K\frac{R_n}{G} + \sum_{j=1}^{K-1} \frac{K-j}{j!} \left(\frac{G}{R_n}\right)^{j-1} \right) e^{\frac{-G}{R_n}} - K\frac{R_n}{G} < 0, \tag{4}$$

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 $R_n > 0$.

B. Optimization using Differential Evolution (DE)

From (1) and (2), we can infer that both variable p and q depend 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

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

The inequality $q < f_u^{-1}(q), \forall q \in (0,1]$ is to guarantee that the two EXIT curves do not intersect with 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 described as follows. We choose 100 initial populations uniformly that satisfy $0 \leq \Lambda_h \leq 1, \forall h$, and $\sum_h^C \Lambda_h = 1$. We conduct the DE with representation *"DE/best/1-with-jitter"* and crossover constant $CR = 0.8$.[10] 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., $FP = 2$, 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.

VI. The Proposed Doubly Irregular CSA (dir-CSA)

First, we want to show an effect of increasing K as shown in Fig. 6(a).[11] Note that this EXIT curve is drawn based on (1). 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 (1). It causes the EXIT curve for SN to be no longer convex. Consequently, the optimal IRSA (with $K = 4$ and $C = 16$) cannot well match the curve as depicted in Fig. 6(b). Even for the optimal CSA (with $K = 4$, $k = 2$, and $C = 16$), which has wider class of codes than that of IRSA, the gaps are still quite wide, see Fig. 6(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 Fig. 6(d).

The aforementioned degree distributions in Figs. 6(b-d) are obtained by using the EXIT-chart-based DE explained in Section V-B. In addition, we summarize many optimal set of codes $\{((n_h, k_h), \Lambda_h)\}_{h=1}^{C}$ for IRSA ($\Lambda_C^{\text{I,K}}$), CSA ($\Lambda_{k,C}^{\text{CSA,K}}$), and dir-CSA ($\Lambda_C^{\text{DIC,K}}$) in Table I.[12] Although we focus on the

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

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

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

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

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

TABLE I. 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	$\{(\mathbf{n_h}, \mathbf{k_h}), \mathbf{\Lambda_h})\}_{\mathbf{h}}^{\mathbf{C}}$	$\mathbf{G}^*$ (p/s)	$\mathbf{R_n}$
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

moderate R_n, $0.4 \leq R_n \leq 0.6$, 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.[13] The fact is important to show the superiority of dir-CSA than the others.

As seen in Table I, 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 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.

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

VII. Performance Evaluation

A. Asymptotic Performances

Relative position of the optimal IRSA, CSA, and dir-CSA shown in Table I and their bounds obtained from (4) are depicted in Fig. 7. Using Fig. 7, the superiority of dir-CSA in achieving the threshold $G^* = 3.7$ p/s compared to the others becomes more apparent since only $\Lambda_{22}^{\mathrm{DIC,4}}$, $\Lambda_{16}^{\mathrm{DIC,4}}$, and $\Lambda_8^{\mathrm{DIC,4}}$ can achieve $G^* = 3.7$ p/s at moderate network rate per user R_n. In addition, with the same R_n, dir-CSA is asymptotically better than both IRSA and CSA for $1 \leq K \leq 4$.

B. Practical Finite-Length Performances

Finite-length simulations are carried out, first, by setting fixed number of PTSs per frame N. Both maximum number of SIC iteration and maximum number of iteration in physical

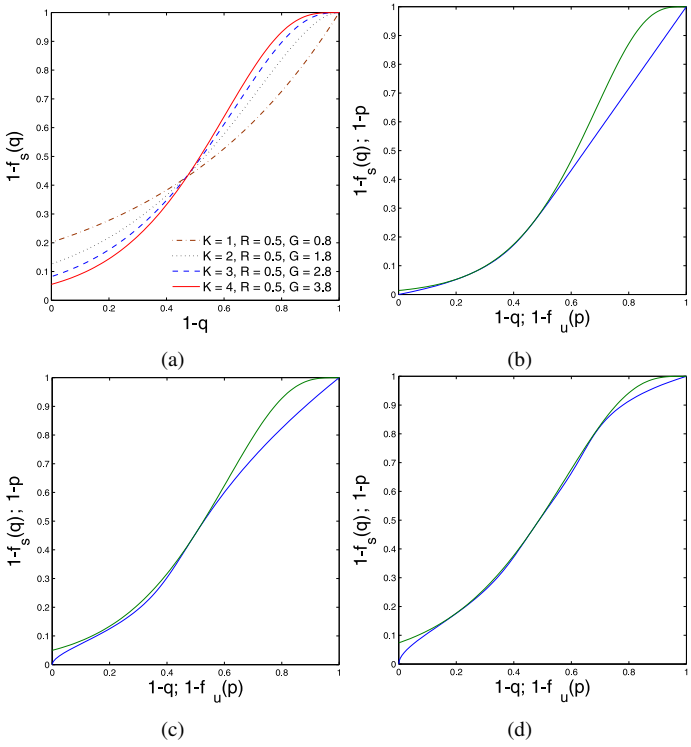


Fig. 6. (a) SN EXIT curves showing that as K increases, the curve is no longer convex, (b) EXIT chart for IRSA with $K = 4$ and $C = 16$ ($\Lambda_{16}^{\mathrm{I},4}$), (c) EXIT chart for CSA with $K = 4$, $k = 2$ and $C = 16$ ($\Lambda_{2,16}^{\mathrm{CSA},4}$), and (d) EXIT chart for dir-CSA with $K = 4$ and $C = 8$ ($\Lambda_{8}^{\mathrm{DIC},4}$).

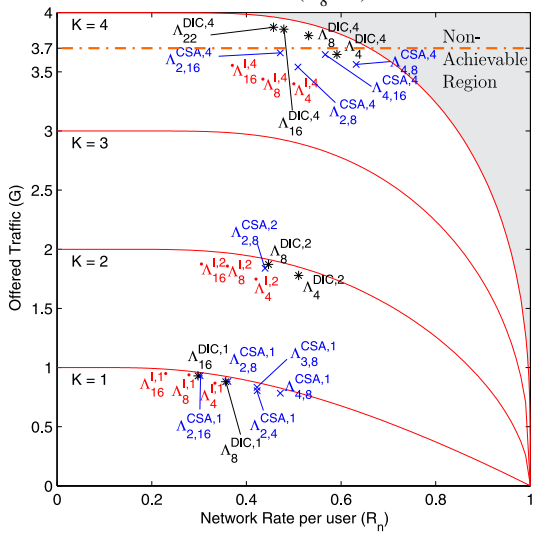


Fig. 7. Threshold G^* shown in Table I vs. network rate per user R_n for IRSA ($\Lambda_{C}^{\mathrm{I,K}}$), CSA ($\Lambda_{k,C}^{\mathrm{CSA,K}}$), and dir-CSA ($\Lambda_{C}^{\mathrm{DIC,K}}$) with $K \in \{1, 2, 3, 4\}$.

decoding is 100. For CSA and dir-CSA, we use RS code over Galois-Field (2^8). Several 8-bit sequences form a full packet. Each packet has length of $10,000$-bit ($L = 10,000$). Each user transmits its packet with transmit power P over AWGN channel with $\sigma_R^2 = \sigma_{l_i}^2 = 1, \forall l_i$.

We use IRSA with the set of codes of $\Lambda_{16}^{\mathrm{I},4}$, CSA with the set of codes of $\Lambda_{2,16}^{\mathrm{CSA},4}$, and dir-CSA with the set of codes of $\Lambda_{8}^{\mathrm{DIC},4}$. The $\Lambda_{16}^{\mathrm{I},4}$ IRSA and the $\Lambda_{2,16}^{\mathrm{CSA},4}$ CSA are considered good compared to the same type of RA with the same K in

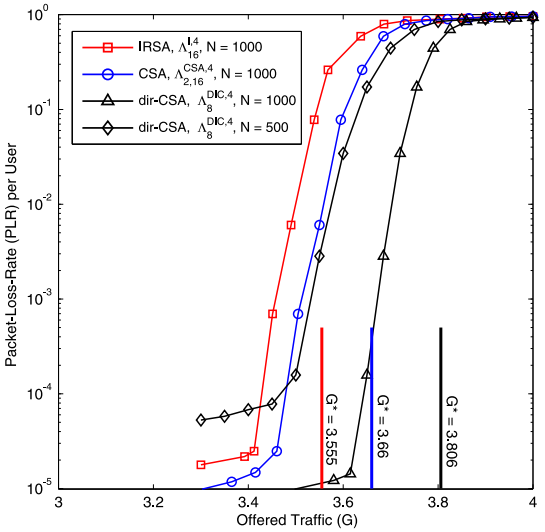


Fig. 8. Average packet-loss-rate (PLR) per user of IRSA ($\Lambda_{16}^{\mathrm{I},4}$), CSA ($\Lambda_{2,16}^{\mathrm{CSA},4}$), and dir-CSA ($\Lambda_{8}^{\mathrm{DIC},4}$) for $K = 4$ and $P = 4.3$ dB.

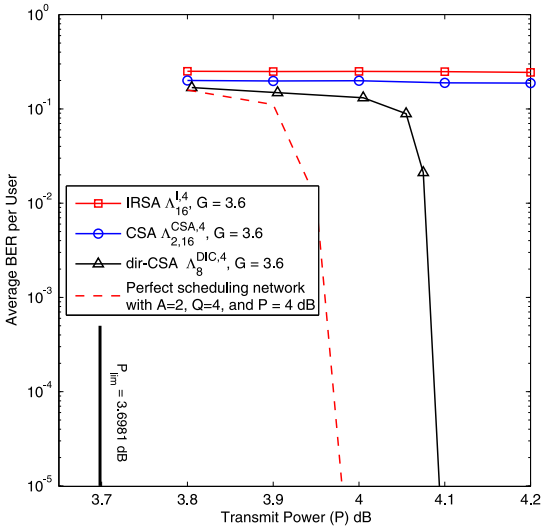


Fig. 9. Average BER per user over AWGN channel of IRSA ($\Lambda_{16}^{\mathrm{I},4}$), CSA ($\Lambda_{2,16}^{\mathrm{CSA},4}$), and dir-CSA ($\Lambda_{8}^{\mathrm{DIC},4}$) for $K = 4$ and $N = 1000$.

Table I.

Average packet-loss-rate (PLR) simulations in Fig. 8 are measured in UNs. This simulation is carried out with transmit power $P = 4.3$ dB. This value is set such that the BER for five-way relaying system is less than 10^{-5} with the fully coordinated mRN assumption (shown later).

The PLRs are lower than that predicted by previous asymptotic analysis G^*. It is reasonable as the results of practical N (finite N). The higher the number of PTSs N, the closer the average PLR threshold with the threshold G^*. In addition, even in this finite length performances, the dir-CSA, both with $N = 500$ and $N = 1000$, outperforms the IRSA and the CSA with $N = 1000$.

BER performances in Fig. 9 are measured in UNs. These simulations are carried out with $G = 3.6$ p/s to show that dir-CSA significantly outperforms the others. With $G = 3.6$ p/s, the bipartite graph for IRSA is broken because $G = 3.6$ p/s is

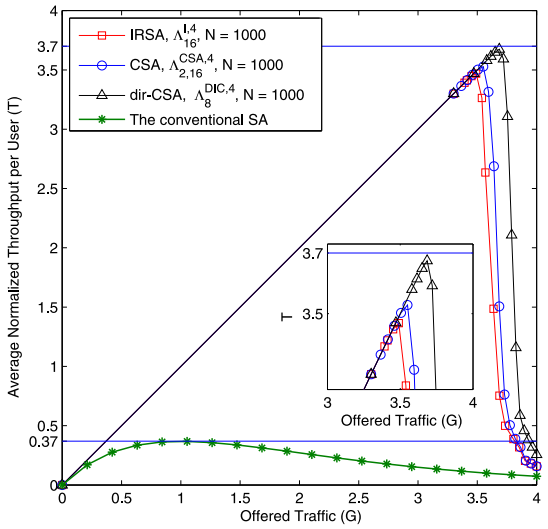


Fig. 10. Average normalized throughput per user over AWGN channel of IRSA ($\Lambda_{16}^{I,4}$), CSA ($\Lambda_{2,16}^{CSA,4}$), and dir-CSA ($\Lambda_{8}^{DIC,4}$) for $K = 4$ and $N = 1000$.

still greater than the threshold G^* of the IRSA $\Lambda_{16}^{I,4}$ ($G^* = 3.555$ p/s). Meanwhile for the CSA $\Lambda_{2,16}^{CSA,4}$ the PLR is still high, see Fig. 8 at $G = 3.6$ p/s. High PLR indicates the fact that there are still many unsuccessfully decoded packets. If we want to have the turbo cliff for the IRSA $\Lambda_{16}^{I,4}$ and CSA $\Lambda_{2,16}^{CSA,4}$, then we need to lower their offered traffic G, e.g., $G = 3.35$ p/s for the IRSA and $G = 3.45$ p/s for the CSA. Lowering the offered traffic G equivalently means decreasing the performances.

BER curve of perfect scheduling network shown in Fig. 9 is measured from a five-way relaying system by assuming that there exist such a smart way to manage scheduling for massive users (hundred or thousand) so that at every time-slot there will always be five users are transmitting at the same time. Therefore, the five-way relaying system can be used as a BER limit for the mu-mRN since it can be considered as the fully coordinated mRN, where the transmission is no longer random but fixed such that all colliding packets can be perfectly decoded in all frames. In adiition, based on [8] and [2], the capacity limit for each user in the mu-mRN is $R_I = \frac{1}{K} C\left(\frac{PP_R}{(K+1)P+P_R+1}\right)$, and $C(P_{lim}) = \log_2(1+P_{lim})$. P_{lim} is the limit power for the five-way relaying system with $K = 4$.

The gap between the limit and the achievable BER is 0.21 dB and deserves further remark. This gap is reasonable as the penalty of multiway relay systems serving hundreds of users.

The average normalized throughput T in Fig. 10 are obtained from the PLR simulations in Fig. 8. The normalized throughput T for the conventional SA is Ge^{-G}. For dir-CSA $\Lambda_{8}^{DIC,4}$, the improvement from the conventional SA is only 9.93×. These results show that our proposed system can approach the target, whereas the others cannot.

VIII. 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 $K \in \{1, 2, 4\}$ working for practical PTSs, such as $N = 500$ and $N = 1000$. 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. 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). For finite-length performances, we optimized physical encoding using SCC comprising irCC and DACC. As a result, 9.93× improvement was achieved by the mu-mRN applying dir-CSA; this result shows that the proposed system significantly outperforms the mu-mRN applying IRSA and CSA. Furthermore, we also conclude that the proposed techniques are efficient as indicated by the BER performance of the mu-mRN which is only 0.21 dB away from the BER limit (obtained from the perfect scheduling network) and 0.4 dB away from the mrC capacity in AWGN channel.

References

[1] Ericsson, "More than 50 billion connected devices - taking connected devices to mass market and profitability," *White Paper*, 2011.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] K. Anwar and T. Matsumoto, "Three-way relaying systems using iterative spatial demapping," in *Turbo Codes and Iterative Information Processing (ISTC), 2012 7th International Symposium on*, Aug 2012, pp. 96–100.

[9] L. Ping, L. Liu, K. Wu, and W. Leung, "Interleave division multiple-access," *Wireless Comm., IEEE Trans. on*, vol. 5, no. 4, pp. 938–947, April 2006.

[10] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *Comm., IEEE Trans. on*, vol. 49, no. 10, pp. 1727–1737, Oct 2001.

[11] M. Tüchler, "Design of serially concatenated systems depending on the block length," *Communications, IEEE Trans. on*, vol. 52, no. 2, pp. 209–218, Feb 2004.

[12] K. Fukawa, S. Ormsub, A. Tölli, K. Anwar, and T. Matsumoto, "EXIT-constrained BICM-ID design using extended mapping," *EURASIP Journal on Wireless Comm. and Networking*, vol. 2012, no. 1, 2012.

[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.