Taylor & Francis
Taylor & Francis Group

IETE Journal of Research

ISSN: 0377-2063 (Print) 0974-780X (Online) Journal homepage: http://www.tandfonline.com/loi/tijr20

APRA: Affinity Propagation-Based Resource Allocation Scheme in M2M for System Capacity Maximization

Hamayoun Shahwani, Phuc Chau, Jaehoon (Paul) Jeong & Jitae Shin

IETE JOURNAL OF RESEARCH, 2018
VOL. 64, NO. 1, 36–50
https://doi.org/10.1080/03772063.2017.1342570

APRA: Affinity Propagation-Based Resource Allocation Scheme in M2M for System Capacity Maximization

Hamayoun Shahwani [1], Phuc Chau [1], Jaehoon (Paul) Jeong [2] and Jitae Shin [1]

[1]School of Electronic and Electrical Engineering, Sungkyunkwan University, Suwon, Republic of Korea; [2]Department of Interaction Science, Sungkyunkwan University, Suwon, Republic of Korea

ABSTRACT
In this paper, we propose an enhanced affinity propagation (AP)-based resource allocation scheme (APRA) to overcome major issues in machine-to-machine (M2M), such as delay, complexity, throughput, and system capacity. There would be rapid increase of added devices, such as cellular and machine-type devices. It would be difficult for Evolved Node B (eNB) to control all of them. Considering this problem, we propose an AP-based group formation method in which machines make groups with other similar type of machines. After making groups, group members in each group can communicate directly with each other by getting a channel from eNB via their group head. A resource allocation method is proposed for different groups that can use the same channel at the same time. Considering energy constraints, we also propose different methods to rotate the role of a group head among group members, through the modification of AP or the application of Markov chain model. As expected, the group head will drain energy at a higher rate than the group members. Thus, the rotation of the group head will increase the overall performance. Simulation results show that the proposed method can minimize both data delivery delay and operation complexity while increasing the throughput, system capacity, and energy efficiency through the rotation of the group head.

KEYWORDS
Affinity propagation; energy efficiency; M2M; resource allocation

1. INTRODUCTION

Recently the Internet of things (IoT) has been spotlighted as a value-producing area related to humans, devices, and machines. Direct communication for devices and machines is defined to the following two types: (1) device-to-device (D2D) and (2) machine-to-machine (M2M). In D2D, mobile phones communicate with each other without involving Evolved Node B (eNB), but with human intervention. Direct communication is done between devices. In M2M, there are three types of M2M communication. First, the machines of the same or different machine types can communicate with each other through eNB, but without human intervention. Second, the machines of different machine types communicate with each other without human intervention and without eNB involvement. Third, the machines of the same machine type communicate without human intervention and without eNB involvement. Sensors, smart grids, and vehicular telematics may belong to one of these types. We will focus on the M2M communication of the third type in this paper. The detailed benefits of M2M in various domains (*e.g.*, agriculture, daily life, and health) have been discussed in [1].

There are two kinds of devices, such as cellular user equipments (CUEs) and machine-type communication devices (MTCDs) in one cell of long-term evolution-advanced (LTE-A). This paper considers MTCDs for M2M devices. There are two spectrum access modes. The first is the overlay mode in which CUEs and MTCDs are given separate bands, and the second is the underlay mode in which CUEs and MTCDs share the same band. Operators prefer the latter one (i.e., underlay mode) in terms of bandwidth utilization. In spite of having interference, the underlay mode can accommodate the maximum number of users [2]. The key technologies enabling the M2M service platform (e.g., addressing, naming, the identification of M2M devices, communication and networking protocols, peer-to-peer (P2P) communication, and the management of devices and networks) are described in [3]. The number of M2M devices is growing day by day, as 50 million devices in 2008 increased up to 200 million devices in 2014, which is expected to be 50 billion devices by 2020 [4,5].

As the number of MTCDs in a cell is increasing, the prevalence of making connections among them is also increasing. If a number of MTCDs simultaneously attempt to access the network, this leads to a low resource allocation (RA) success rate and also a high network congestion in physical random access channel (PRACH).

This may cause unexpected long delay, high packet loss rate, the waste of radio resource, *etc.* [6]. Figure 1(a) shows the conventional M2M architecture, and at any time, a machine asks an eNB to download some data from another machine. The eNB searches for another machine relevant to the requested data, checks proximity, and allocates a channel to allow the machines to communicate with each other directly. It would be difficult for the eNB to search for the desired contents of each machine if a number of machines start downloading simultaneously. This will exhaust the eNB, so it cannot perform its primary tasks of accommodating CUEs. A solid line in Figure 1(a) shows a direct link for traffic signalling and dotted lines show links for control signalling.

By considering network congestion due to an excessive number of MTCDs accessing the network, long delay due to a low RA success rate, throughput due to high packet loss rate, and low system capacity due to the waste of radio resources, we propose an affinity propagation (AP)-based RA scheme (APRA) for M2M, based on AP algorithm. AP can be considered as an efficient clustering algorithm as compared to other clustering algorithms such as *k*-means. Frey *et al.* [7] have performed several experiments to compare the results of AP and *k*-mean clustering algorithms. The advantages of AP over *k*-means are mentioned in [7] as follows:

- In AP, the number of clusters is not per-specified before running the algorithm; instead, AP takes similarities between members as an input and lets them exchange messages to select exemplar. On the other hand, in *k*-means, the number of clusters is fixed before running the algorithm.
- A preference value is selected to adjust the number of clusters in AP. On the other hand, in *k*-means, *k* centroids are selected at the beginning of the algorithm, and the members are assigned to each closest centroid. After that, the mean of all member's distance is calculated to select a new centroid. This is repeated until the best result is obtained.
- AP gives a better performance than *k*-means by selecting the best exemplar after executing only one run while *k*-means runs 10,000 times.

Therefore, it can be concluded that AP is an efficient clustering algorithm as compared to other clustering algorithms such as *k*-means. Like AP, APRA does not require to determine the number of devices in a group before running the algorithm as the other grouping algorithms do. This would be beneficial to make groups without knowing the exact number of devices in each group. The APRA constructs clusters/groups of MTCDs on the basis of machine type and distance. After the formation of the group, APRA allows them to communicate with each other. APRA focuses on the issues of delay and system capacity that are not addressed in the previous work. APRA also considers energy efficiency that is a hot issue for MTCDs. By using different techniques, the energy efficiency of our scheme is shown to be good. Our contributions are as follows:

- An AP-based group formation and communication of MTCDs. The group formation would be performed on the basis of distance and similarity.
- A RA procedure for different groups by using the same frequency band. The location and all basic information of groups are sent to an eNB after the group formation by a group head.
- An energy-efficient method for grouped MTCDs using Markov chain. The group head is rotated among members to increase the lifetime of the group.

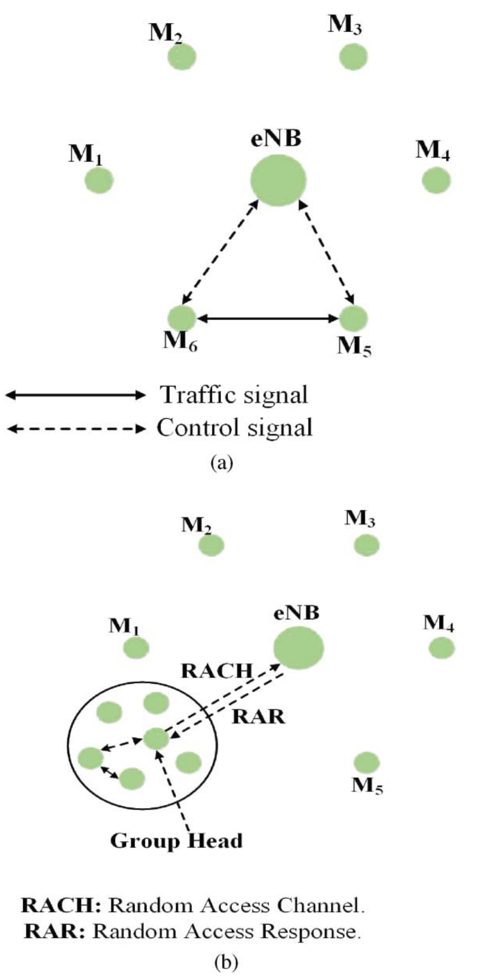

Figure 1: Flow diagram comparison. (a) Conventional M2M architecture; (b) group-based M2M communication

The rest of this paper is organized as follows. Section 2 summarizes and analyzes related work. Section 3 describes the detail about the grouping of machines by using modified version of AP. Section 4 explains a proposed scheme (called APRA) in detail. Section 5 shows the performance evaluation of APRA compared with a social ties-based cooperative video multi-cast (SoCast). In Section 6, the paper is concluded along with future work.

2. RELATED WORK

Lien et al. proposed a group-based radio resource management in [8] and declared the group communication of MTCDs as an effective scheme to support large number of devices with small data transmission. MTCDs are grouped on the basis of quality-of-service (QoS) characteristics and requirements such as packet arrival rate and the maximum tolerable jitter. A cluster with a larger packet arrival rate has a higher priority. Wang *et al.* [9] gave the idea of developing Stackelberg game framework. A CUE and device-to-device users equipment (D2DUE) are grouped to form a leader–follower pair, and the leader owns the channel resource and charges a D2DUE some fee for using the channels. The CUE has an incentive to share the channel with the D2DUE if it is profitable, and the leader has the right to decide the price, and the D2DUE under the charging price can choose the optimal power to maximize its pay-off. In this way, an equilibrium can be reached. Cao *et al.* proposed a group formation based on social trust or social reciprocity for downloading missing video packets [10]. Some clients first determine how to obtain missing packets from other clients based on social ties, and then send D2D communication requests to an eNB. The eNB then allocates the channel based on the radio RA policy for D2D communication. Glorezaci et al. proposed a UE-based cache scheme such that users (smart-phones or tablets) cache popular video files and upon receiving request from other users, they serve them via D2D transmission [11]. For this, the eNB must be aware of the stored files, channel state information of the users, and control the D2D communication. You *et al.* [12] explain multiple-device-to-multiple-device (MD2MD) communication and assume that there are multiple downloaders wishing to get a popular content. All users create metadata to share their own multimedia contents and publish them on a content store in an eNB. As mentioned in the papers above, the devices will share their data with the eNB. On the request from other devices, the eNB will check the data and allocate them direct links (D2D/M2M), if they are in proximity. It means that the eNB should have the information about the data of all devices and their locations in addition to their basic information like naming, addressing, etc. Processing all these may overwhelm the eNB to perform all the tasks. Therefore, we propose a group-based MTCD communication model to release the burden for the eNB. Figure 1(b) illustrates group-based M2M communication in which a request for asking a channel will be informed to the group head instead of sending the request directly to the eNB. The group head then sends a random access channel (RACH) to eNB for channel allocation. The eNB will as a response send a RAR message. In the next section, we will explain our modified AP for grouping.

3. GROUPING BY USING MODIFIED AFFINITY PROPAGATION (AP)

AP was first proposed by Frey and Dueck [7] as a grouping algorithm in which groups are formed by passing messages between data points. The magnitude of each message between two data points reflects the current affinity (*i.e.*, close resemblance level) which one data point has for selecting another data point as its exemplar. First, real-time similarities are checked between data points by using Euclidean distance. There are two kinds of messages, such as responsibility message and availability message; note that the definitions of those messages will be explained later, and they are sent from data point i to exemplar k and from exemplar k to data point i, respectively, as shown in Figure 2. Through the iterative exchange of these messages, a set of exemplars as group heads and the corresponding groups emerge.

Definition 3.1 (Similarity by Euclidean distance): Algorithm 1 starts with the similarity measure between two data points, given by $s(k, i)$, to show the suitability of data point k as an exemplar for data point i. A data point is a group member having sensing data and an exemplar

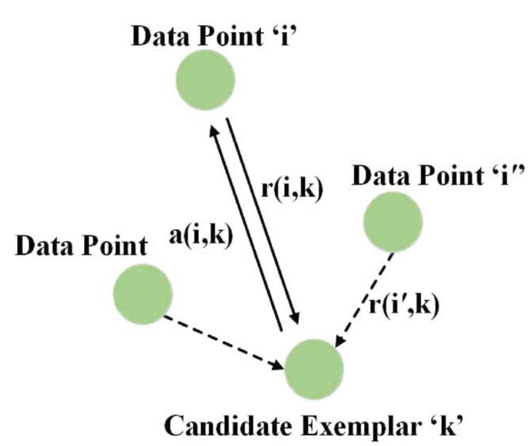

Figure 2: Affinity propagation for group formation

Algorithm 1 Modified Affinity Propagation Algorithm

```
1: function AP -ALGORITHM()
2:     for k=1:M do
3:         for i=1:M do
4:             a(k,i) ← 0 , a'(k,i) ← 0                ▷ Initialization.
5:             if k ≠ i then
6:                 s(k,i) = −( ||x_k − x_i||^2 + ||type_k − type_i||^2), i ≠ k
7:             end if
8:         end for
9:     end for
10:    p = median s(k,i)                     ▷ p is a priori preference.
11:    for k=1:M do
12:        for i=1:M do
13:            if k = i then
14:                s(k,i) = s(k,k) ← p
15:            end if
16:        end for
17:    end for
18:    while sim = 0 do
19:        for k=1:M do
20:            for i=1:M do
21:                r(k,i) ← s(k,i) − max_{k'∈1,...,M,k'≠k}
                       {a(k',i) + s(k',i)}
                                             ▷ Responsibility message.
22:            end for
23:        end for
24:        for k=1:M do
25:            for i=1:M do
26:                a(k,i) ← min {0, r(k,k)+
                       Σ_{i'∈1,...,M,i'≠i,k} max{0, r(i',k)}}
                                             ▷ Availability message.
27:                if k = i then
28:                    a(k,k) ← Σ_{i'∈1,...,M,i'≠k} max{0, r(k,i')}
                                             ▷ self availability
29:                end if
30:            end for
31:        end for
32:        if a'(k,i) = a(k,i) then
33:            sim = 1
34:        end if
35:        a'(k,i) ← a(k,i)
36:    end while
37:    for k=1:M do
38:        for i=1:M do
39:            sum ← a(k,i) + r(k,i)
40:        end for
41:    end for
42:    if {a(k,k) + r(k,k) > 0} or {a(k,k) + r(k,k) > a(k,k') + r(k,k')} then
43:        GH ← k
44:    end if
45:    return GH
46: end function
```

is a group head as a centre for a group having the representative sensing data for the group. Since the goal for the grouping is to minimize the sum of squared errors, $s(k, i)$ is set to negative squared error (Euclidean distance).

$$s(k,i) = -||x_k - x_i||^2 \text{ for } i \neq k. \tag{1}$$

In this paper, we assume that the location should be known before the execution of AP algorithm. Thus, each node must have the knowledge of its location. The localization for the location of each node can be performed by using different types of localization algorithms explained in [13] or by global positioning system (GPS). A threshold value for the distance between two data points is declared. If the distance between two M2M devices is less than or equal to the threshold value, they will be in one group. Note that similarities are not only squared errors, but also are derived for pairs of images, pairs of micro array measurements, pairs of English sentences, and pairs of cities.

Definition 3.2 (Similarity by machine identities): We modify the AP algorithm by adding machine type identities in it. Now similarities are checked by machine types along with Euclidean distance. If the distance between i and k is less than or equal to the threshold value, then type equality of device $i's$ type is checked as follow:

$$t(k,i) = -||type_k - type_i||^2 \text{ for } i \neq k. \tag{2}$$

$$\begin{cases} Type_i \equiv Type_k, & \text{if } \quad t(k,i) = 0 \\ Type_i \not\equiv Type_k, & \text{otherwise} \end{cases}$$

Similarity matrix is modified as

$$s(k,i) = -\left(||x_k - x_i||^2 + ||type_k - type_i||^2\right) \text{ for } i \neq k. \tag{3}$$

If these two conditions (Euclidean distance and type) are satisfied, messages are exchanged to make groups and select group heads. In Algorithm 1, the number of groups is not prespecified. Algorithm 1 takes a real number $s(k, k)$ for each data point k as input. This value of $s(k, k)$ is defined as preference because a data point with a large value of $s(k, k)$ is selected as an exemplar. In line 4 of Algorithm 1, availabilities are initialized to zero; note that availability will be defined later. Input similarities along with machine type are defined in lines 5–7 by negative Euclidean distance. The value of $s(k, k)$ is set to an appropriate value as a priori preference according to the required number of groups. The data points having the same exemplar are grouped into the same group. Line 14 describes the self-similarity $s(k, k)$, given a priori preference value p, which is set to be the median of the input similarities in line 10, i.e., $s(k, i)$. This will be able to make a fair number of groups. On the other hand, if p is set to the minimum of input similarities, there will be a small number of groups or just one group. Line 21 computes the responsibility message $r(k, i)$, sent from data point i to exemplar k. The message $r(k, i)$ means how well-suited point k would be to serve data point i as an exemplar (i.e., representative), considering other potential exemplars for point i. In line 26,

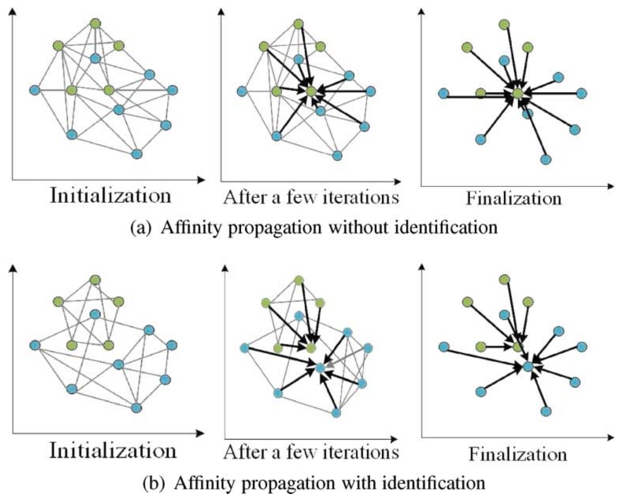

Figure 3: Affinity propagation with (a) and without (b) identification

availability message $a(k, i)$ is sent from exemplar k to data point i. The message $a(k, i)$ means how much appropriate it would be for data point i to choose point k as its exemplar, considering the support from other points that point k should be an exemplar. $a(k, i)$ is obtained by summing the self-responsibility $r(k, k)$ and the sum of responsibilities that exemplar k has for the other data points. A negative value of $r(k, k)$ shows that the data point k is not suitable for an exemplar as a group head, but it can join another group. Self-availability is defined in line 28. Availabilities and responsibilities are summed up, and then self-availabilities and self-responsibilities are compared to get a group head from lines 37 to 45. AP grouping algorithm in [7] only considers the similarity based on Euclidean distance. We add the machine types along with Euclidean distance because APRA deals with different types of M2M devices.

In M2M, every device has a type ID which depends on its type. Let us suppose that there are two different types of sensors. Five sensors have $Type_1$ and eight sensors have $Type_2$. They will make two groups by following the next steps. However, if they follow the conventional AP algorithm, they will make only one group. Figure 3 explains this fact in which green colour represents machine $Type_1$ and blue colour represents $Type_2$. Figure 3 (a) shows that the AP algorithm selects a head without type identification. On the other hand, Figure 3(b) shows that AP algorithm selects two heads with type identification to make two groups according to machine types.

4. PROPOSED SCHEME

The first step of APRA is to make a group by using the modified AP algorithm as explained in Section 3. Once the group is formed, the group head will send the necessary details of the group to the eNB. The eNB does not have any knowledge of other group members. Suppose that one *MTCD* of a group has some missing packets and there is a helper *MTCD* in the same group. The *MTCD* with missing packets will send a request to the group head. The head has the information about the data of MTCDs. It is assumed that all the MTCDs share their information with their group head. The head will send a RACH request to eNB. Now eNB will not search for data or other information. It just allocates a channel to the group for the group header's request. After the allocation of a channel from eNB, the helper *MTCD* will send the missing packets to the requesting *MTCD* via a direct link. We assume that members communicate with the group head on carrier-sense multiple access (CSMA) basis (Wi-Fi), which do not need the coordination of eNB. As explained in [10], users can interact with each other using the CSMA technique for getting the information of helpers.

As discussed earlier, eNB must have all the information in a general M2M case. eNB performs three tasks. First, eNB searches for the data which MTCD has requested. Second, eNB then searches for the location of the device which can share its data with the one which has requested the data. Third, eNB will allocate the resource of a channel to them and ask them to initiate a direct link.

The pseudo-code for overall procedure is given in Algorithm 2 (APRA Algorithm). The first portion explains the group formation via AP algorithm. The second portion demonstrates the channel allocation

Algorithm 2 APRA Algorithm

```
function APRA-ALGORITHM()
    /* Group formation */
    for each MTCD M do
        Process_AP_technique ()
        if r(k,k) + a(k,k) > a(k,k') + r(k,k') ← true then
            A group is formed such that k is the group head (GH)
        else
            Do the process for another GH
        end if
    end for
    /* Channel allocation */
    Process_Channel_Allocation ()
    /* Energy efficiency */
    Process_Energy_Efficiency ()
        ▷ Compare the energy levels and select the head with a higher energy level and a less distance from other members.
end function
```

procedure. The last portion explains the energy efficiency of APRA algorithm. For the detailed version of all the three parts, refer to the algorithms in the Appendix.

In Section 4.1, information sent by the group head to eNB is given. RA is explained in Section 4.2. In Section 4.3, we explain energy efficiency by applying Markov chain and also describe four cases of selecting a group head wisely.

4.1 Information Exchange Between eNB and Group Head

Once the groups are formed by using AP algorithm, all group heads will start transmitting the necessary information to the eNB for further process. Each group head has the responsibility to inform eNB of its identity, location, the number of group members, and the group diameter (*i.e.*, the size of the geographical region for the group). It will not send the complete information of all group members, as the intensive report may overwhelm the eNB. The message sent to eNB is represented by

$$M_{eNB} = \left(ID_H, L_H, D_g, m_g\right), \tag{4}$$

where M_{eNB} is the message that eNB receives from the group head, ID_H is the identity of the head, L_H is the location of the head, D_g is the group diameter of the gth group, and m_g is the number of machines in the gth group.

4.2 Resource Allocation

The RA procedure in LTE-A is classified into two types, such as contention-based RA and contention-free RA [6].

- Contention-based RA is done in four steps. In the first step, the UE sends a preamble message by selecting one among 64 orthogonal preambles via physical random access channel (PRACH). In the second step, the eNB sends a RAR that contains ID, uplink scheduling grant, uplink timing information, and a temporary identifier called cell radio network temporary identifier (CRNTI). In the third step, an international mobile subscriber identity (IMSI) is transmitted by the UE via physical uplink shared channel (PUSCH) for scheduling a radio resource request. In the last step, the eNB sends a contention resolution message via a physical down-link shared channel (PDSCH) containing the IMSI of the UE.
- Contention-free RA has three steps. In the first step, the eNB directly assigns a preamble to the UE, so the UE does not need to select a random preamble among 64 orthogonal preambles. In the second step, the UE sends a request using the assigned preamble. In the last step, the eNB sends an RAR to the UE. This generally happens in handover process.

The requirement for data connection will be higher than that for voice connection in CUEs. This may reduce the RA success rate. During any step of the RA procedure, a congestion may happen leading to high delay, high packet loss, or more energy consumption. To overcome these issues, we propose an efficient RA procedure by grouping MTCDs.

Due to the limitation of resources, all groups may not be given resources at the same time. But according to our proposed RA scheme, the same channel can be allocated to different groups at the same time by the geographical locality, as shown in Figure 4(a). A central solid-line circle represents the group diameter (*i.e.*, D_g) and a dashed

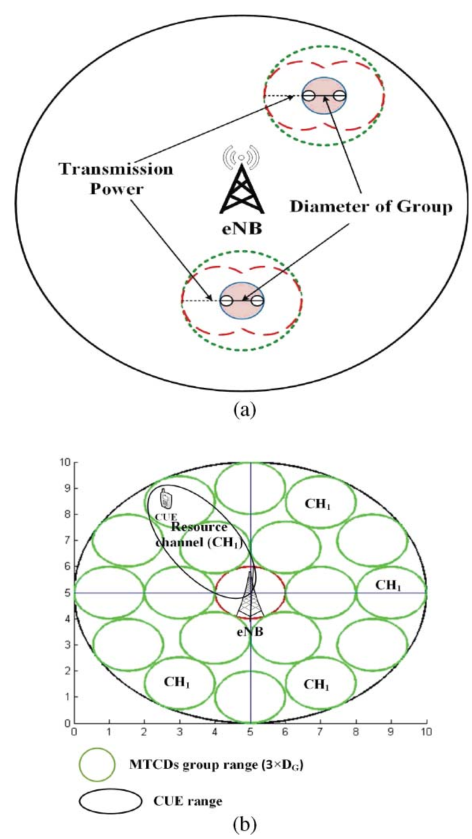

Figure 4: Resource allocation procedure for MTCDs in underlay mode. (a) Resource allocation criteria; (b) maximum number of groups that can use the same resource channels

line (union of two circles) shows the transmission range of the MTCDs on edges. Assuming that the transmitter and receiver have omni-directional antennas and are at the edges of the group (circle), in this case, the transmission range can cover the whole diameter of the group. An upper dotted-line circle is the area where the eNB will not allocate the same channel to any other user in the adjacent areas belonging to other groups (*i.e.*, $3D_g$). In this way, the two groups can use the same channel without interference. The diameter of a group is equal to D_g. The RA diameter will be equal to three times D_g.

We consider a single-cell scenario that has M MTCDs and C CUEs, operating in underlay mode. The M MTCDs will make G groups having the different number of members in each group, depending on their distance and type according to the AP algorithm, where G = 1, ..., M. There are N_{CH} orthogonal channels that are used by CUEs and some of them may be shared with MTCDs. One channel can be used by one CUE and one or more MTCD groups, depending on their location and distance from each other. The minimum transmission data rate for CUE and MTCD can be represented by SINRs of γ_C and γ_M, respectively. We assume that the maximum transmission data rate is determined on the basis of modulation and coding scheme (MCS), denoted by SINR of γ_m. According to Shannon capacity formula in [14], the data rate of CUE, when its frequency resource is not reused by MTCDs, is given by R_{CUE}:

$$R_{CUE} = B\log_2\left(1 + \frac{P_c G_c PL_c}{N_0}\right), \tag{5}$$

where B is the bandwidth, P_c, G_c and PL_c denote the transmit power, channel gain and the path-loss between the c^{th} CUE and eNB, respectively, and N_0 is the noise variance. The data rate for MTCD groups when they reuse the frequency resource of the c^{th} CUE is given by R_{MTCD}:

$$R_{MTCD} = \sum_{m=1}^{G} B\log_2\left(1 + \frac{P_m G_m PL_m}{I_c + N_0}\right), \tag{6}$$

where P_m, G_m, PL_m, and I_c denote the transmit power, channel gain and the path-loss between MTCD pairs and the interference from c^{th} CUE. The total throughput of CUE and MTCDs that use the same resource channel is given by

$$R_{(CUE_c,MTCD)} = B\log_2\left(1 + \frac{P_c G_c PL_c}{I_m + N_0}\right) + \sum_{m=1}^{G} B\log_2\left(1 + \frac{P_m G_m PL_m}{I_c + N_0}\right), \tag{7}$$

where I_m is the interference from mth MTCD pair using the same resource of CUE. CUEs are given priority, while MTCDs will reuse the resources of CUEs, which can be given as

$$K = \arg\max_{J} \left\{R_{(CUE_c,MTCD)} - R_{CUE}\right\}, \quad \text{for } J \in \{\gamma_C \le \eta_C \le \gamma_m, \gamma_M \le \eta_M \le \gamma_m\}, \tag{8}$$

where $\eta_C = \frac{P_c \cdot G_c \cdot PL_c}{I_m + N_0}$ and $\eta_M = \frac{P_m \cdot G_m \cdot PL_m}{I_c + N_0}$. Path-loss for CUEs is expressed by $66.5 + 35 * \log(d)$, using Xia model [15] and for MTCDs is $38.4 + 20 * \log(l)$, using free space model, where d is the link distance between CUE and MTCDs and l is the distance between MTCDs. According to [16], CUEs listen to the signals periodically on a common control channel (CCCH) to check the proximity of D2D users. CUEs then send the position information for themselves and neighbouring D2D users to the eNB on the CCCH. In APRA, group information, as explained in Section 4.1, is sent to the eNB just after group formation, so there is no need to send position information periodically. The interference in APRA is minimized by intelligent RA by the eNB, on the basis of the information provided by the group heads. The equations in [17] are used for the data rate of cellular users and D2D users when they share the frequency resources.

An optimum RA method can cause the maximum throughput of CUEs and MTCDs communication, represented as

$$R_{max} = \max\left(R_{(CUE_c,MTCD)}\right), \tag{9}$$

such that

$$CH_c \le 1, \tag{10}$$

$$A_c \cap 3D_g = \varnothing \text{ and } 3D_g \cap 3D_{g'} = \varnothing, \tag{11}$$

$$CH_m \le X, \tag{12}$$

where A_c means the coverage area of cth CUE and $3D_g$ means three times the diameter of gth MTCD group. Constraint (10) shows that at least one cellular user has occupied a channel denoted by CH_c. Constraint (11) shows that there should be nothing in common between the CUE coverage area and group coverage area of MTCDs and also among other group coverage areas of MTCDs (i.e., $3D_{g'}$) to use the same channel of CUE. This will give a higher throughput and a minimum interference. Based on constraints (10) and (11), MTCD channels denoted by CH_m in constraint (12) show that the resource channel of one CUE can be used by X number of MTCD groups, as shown in Figure 4(b), and the cell is covered with equal-sized MTCD groups [18]. According to Figure 4(b), the CUE in the first quadrant uses the resource channel CH_1, which can be used by other groups on the basis of constraint (11). From Figure 4(b), we can see that the one resource channel of CUE can be used by other groups.

4.3 Energy Efficiency

As already discussed, M2M is defined as the direct communication between users without involving the core network or eNB. Researchers show that this direct communication increases the energy efficiency of the devices. Furthermore, this energy efficiency is more increased if the devices communicate in the form of a group. The members of the group can be more energy efficient than an individual device communicating with the network. However, the problem with grouping is that the head of a group will drain its energy more drastically. In this paper, we propose two methods to rotate the group head among the members so that the life of the group can be increased. This is done by either applying Markov chain or modifying the AP algorithm.

4.3.1 Applying Markov Chain

Markov states a random process that undergoes a transition from one state to another state. In our case, the transition is triggered on energy basis. An MTCD will remain a group head until 20% of its energy is drained. After that, another MTCD will play a role of the head. Once all the members have drained 20% of their energy, the head selection is started again. Let us assume that three MTCDs (m_1, m_2, and m_3) construct a group according to AP algorithm. Where m_1 is selected as a head, it will remain as a head until 20% of its energy is drained. Then, there will be a transition from m_1 to m_2 or m_1 to m_3. $P_{m_1m_1}$, $P_{m_2m_2}$, and $P_{m_3m_3}$ represent the probabilities that the state remains unchanged, and $P_{m_1m_2}$, $P_{m_2m_3}$, and $P_{m_3m_1}$,..., show the probabilities of state transition from m_1 to m_2, m_2 to m_3, and m_3 to m_1,..., respectively, and vice versa.

$$P = \begin{matrix} m_1 \\ m_2 \\ m_3 \\ . \\ . \\ . \end{matrix} \begin{pmatrix} m_1 & P_{m_1m_1} & P_{m_1m_2} & P_{m_1m_3} & . & . & . \\ m_2 & P_{m_2m_1} & P_{m_2m_2} & P_{m_2m_3} & . & . & . \\ m_3 & P_{m_3m_1} & P_{m_3m_2} & P_{m_3m_3} & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \end{pmatrix}$$

The above transition matrix represents the probabilities of the transition from one state to another. Row index is the current state and the column index is the next state. We denote the probability of transition by $P_{m_im_j}$ in Equation (13). Probability here is a conditional probability, which depends upon the energy and proximity preference value Y of each MTCD. If a Markov chain has n states, the equation is given by

$$P_{m_im_j} = Y\left(\sum_{k=1}^{M} P_{m_im_k}P_{m_km_j}\right), \tag{13}$$

where $Y = a, b, c, ...$ The values of $a, b, c, ...$ will be different for each case as follows:

Case 1 (Conventional method). In conventional method, only one head is selected for the group until it exhausts its energy. The energy of the head will drain drastically than the other members. The group will remain alive up to 50 hours if 20% of the remaining energy is drained every 5 hours.

Case 2 (Round robin scheduling). In a round robin scheduling, each MTCD will get a chance to be the group head. The most appropriate one according to proximity preference (PP) is selected as a head in the beginning of the algorithm, then the second and then the third. In this case, $a = PP_{m_1} = 100\%$, $b = PP_{m_2} = 90\%$, and $c = PP_{m_3} = 80\%$.

Case 3 (Energy -based scheduling). In case 3, the role of head is to rotate on the basis of energy level. The values of $a, b, c, ...$ are calculated by

$$a = W_E \times E_{m_1},\ b = W_E \times E_{m_2},\ c = W_E \times E_{m_3}, \ldots$$

where W_E is the weight of energy (e.g., 40%) and E_m is the energy level of mth MTCD. If two or more have the same energy, the selection will be on proximity preference base.

Case 4 (Proximity-and-energy-aware scheduling). In this case, AP algorithm selects m_1 as a group head on the basis of proximity and energy in proportion of 60% and 40%, respectively. Every iteration both proximity preference values and energy levels are checked and calculated. The one having the highest probability will be selected as a head. The values of $a, b, c, ...$ are calculated by

$$a = W_E \times E_{m_1} + W_P \times PP_{m_1}, \text{in the same way for } (b, c, \ldots),$$

where W_E and E_{m_1} are the weight of energy and energy level of m_1 and W_P and PP_{m_1} are the weight of proximity and proximity preference value of A, respectively.

4.3.2 Modifying AP Algorithm

As mentioned earlier, the AP algorithm is used to make groups and select a head for each group on the basis of distance and machine type. Now for energy efficiency, energy comparison is added to the distance and machine type-based similarities of the algorithm. Suppose that k is the suitable candidate to be selected as a group head, and its power is more or equal to the others. In this case, k is selected as a group head. If k's energy is already 20% less than the other members, the AP algorithm will select another suitable head. Once the group head is selected, the communication process will be started. If 20% of the energy of group head is drained, the AP algorithm will start from the beginning and select another group head. In this way, the overall energy of the group will be balanced by rotating the role of head among the members.

4.3.3 Optimization of System Performance

The optimization of system performance can maximize the overall lifetime of the system. System performance depends on the selection of a group head and is expressed as *SP*:

$$SP_{\max} = \max\left(P_{m_i m_j}\right), \tag{14}$$

where $W_P > W_E$, which is used for proper accuracy of our algorithm. The equation shows that by selecting a group head with the highest sum value of proximity and energy level, we can maximize the system performance.

5. PERFORMANCE EVALUATION

We evaluate the performance by comparing the SoCast [10] with APRA, a new RA method, and simulation results.

5.1 Comparison of APRA with SoCast

In general scenario as shown in Figure 5(a), eNB performs all the tasks, such as checks the availability of data, then looks for the proximity of devices and at last allocates resource channels to the requesting device. In SoCast model, groups are made on social ties basis and then data is downloaded. Figure 5(b) explains the flow diagram of SoCast. All the devices first broadcast their missing packet IDs, and then collect the data by feedback of other devices. All devices maintain information tables that contain information about their helpers. They will make social ties-based groups and ask eNB for RA. Every time they need to download something, they should follow the procedure in Figure 5(b). Therefore, computational complexity for SoCast is $O(N^2)$, where N is the number of devices. In SoCast, authors evaluate the results using video frames. But we are using data frames for comparison.

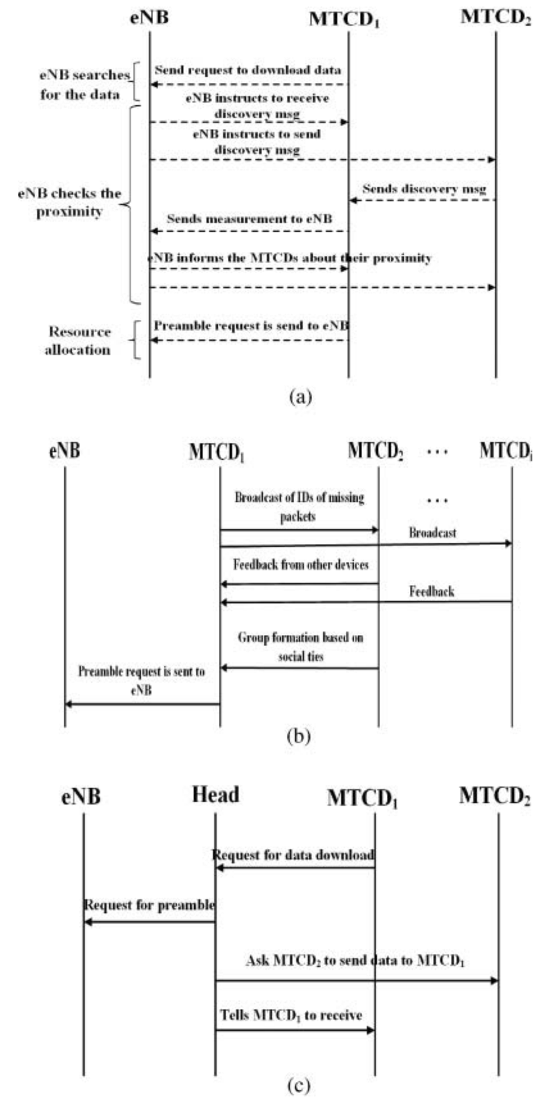

Figure 5: Flow diagrams. (a) Flow diagram for a general scenario; (b) flow diagram for SoCast; (c) flow diagram for the proposed model

In APRA, groups are made and group heads are selected only once in the beginning because we are dealing with similar machines. Figure 5(c) elaborates the flow diagram, in which a device sends a request to a group head for downloading a data. The head will check for the data and ask eNB for RA. The important thing here is that eNB does not know anything about the members of the groups. eNB has the information of only group heads and allocates a channel on a group head's request. In this way, we can overcome the burden on eNB and can also increase the throughput and minimize the delay. The computational complexity for APRA once the groups are formed is $O(m)$, where m is the number of members in a group that can retrieve data, depending on available frequency channels.

5.2 Simulation Results

We consider a single-cell scenario in which the eNB is in the centre of the cell. The radius of the cell is 500 meters. We evaluate the performance for (1) the different number of machines in one group by keeping packet size constant and (2) the different packet size by keeping the number of MTCDs constant. Having all the information for intelligent RA, more than one MTCD groups can use the same channel. eNB allocates the channels on the basis of information provided by group heads. The diameter of groups depends on the preference value p. Transmission power of MTCDs varies from 2 to 23 dBm as it depends on the diameter of group. The iteration of simulations for every number of users is 200. γ_C, γ_M, and γ_m are assumed to be 0, -10, and 20 dB. Other parameters are given in Table 1. Figure 6 illustrates the average system throughput with respect to average transmission power. Transmission power is related to the diameters of groups; as the diameter of the group is increased, the number of groups in a cell decreases so that the transmission power also increases. If the number of groups with interference is much more than that of without interference, the throughput of that with interference is slightly higher as compared to that without interference. Figure 7 shows that when the number of groups is increased, the throughput also increases. If the number of groups with interference is much more than that of without interference, then the throughput of that with interference is slightly higher, but if the difference is little, then that without interference will be much better in throughput. We randomly distribute the MTCDs in the cell of radius 500 m and apply AP algorithm to make groups. We check the results for different preference values which gives different number of groups by changing its value. Figure 8 shows the cumulative distribution function (CDF) of the system throughput of only CUE, CUE with one MTCD pair, and CUE with two MTCD pairs. The more MTCD pairs

Table 1: Simulation parameters.

Parameter	Values
Cell radius	500 m
MTCD group diameter	Depends on p
Bandwidth	10 MHz.
CUE transmit power	46 dBm
MTCD transmit power	2–23 dBm
CUE antenna gain	9 dBi
MTCD antenna gain	5 dBi
Noise spectral density	-174 dBm/Hz

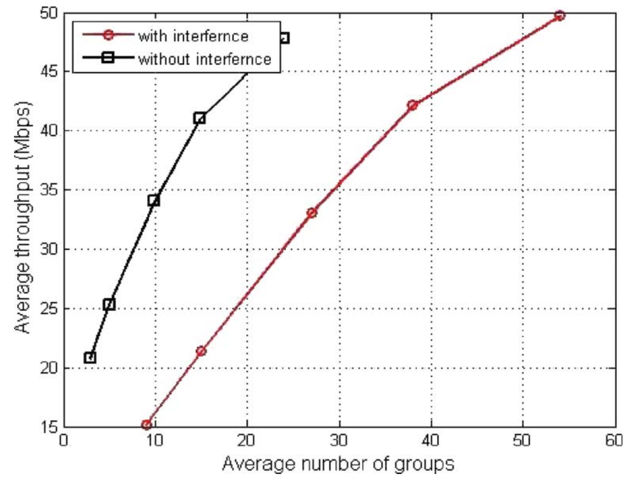

Figure 7: Throughput increases as the number of groups increases

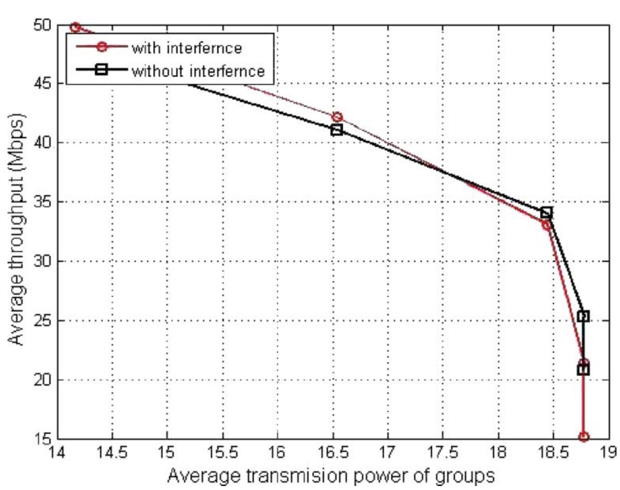

Figure 6: Average system throughput when a number of groups use the same resource of CUE with and without interference

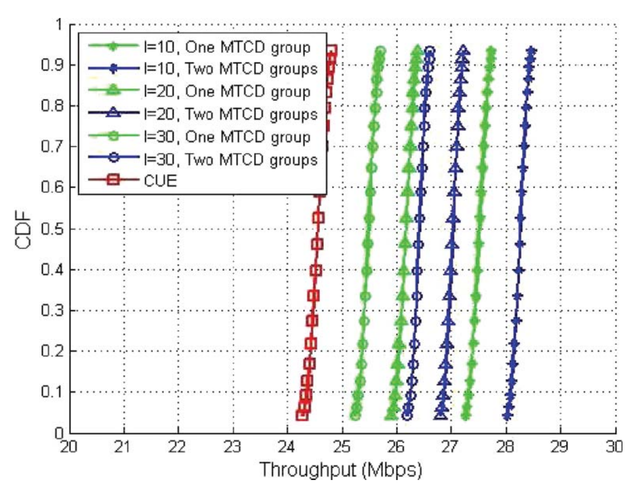

Figure 8: CDF of system throughput with different number of MTCD groups

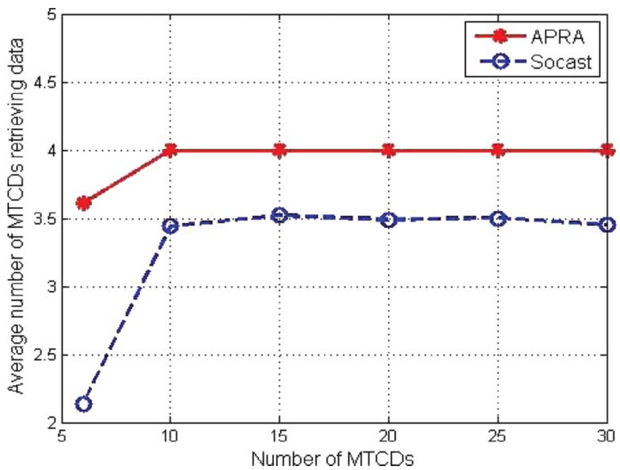

Figure 9: The maximum number of MTCDs retrieving data when the number of packets is the same

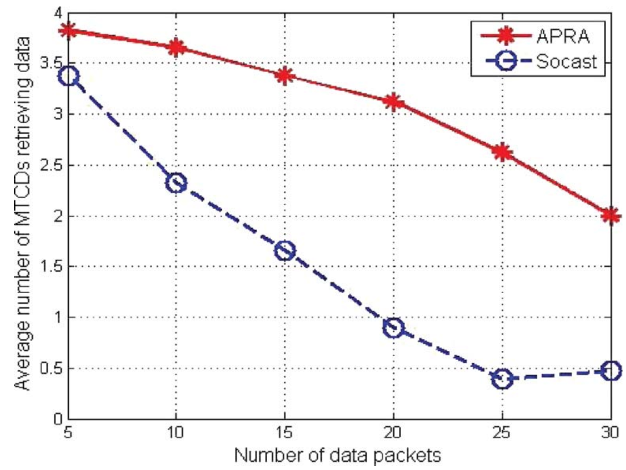

Figure 10: Maximum number of MTCDs retrieving data when the number of data packets is increased

use the resource channel of a CUE, the higher will be the throughput of the system. d is 400 m while l is 10, 20, and 30.

Our proposed scheme APRA is compared with SoCast by using the two parameters, such as "Retrieving Maximum MTCDs Data" and "Delay or Computational Complexity". We evaluate APRA for energy efficiency.

5.2.1 Retrieving Maximum MTCDs Data

We evaluate the performance in terms of the average number of MTCDs retrieving data. It is supposed that the number of channels is fixed and 4 channels are used for delivering data in the cell. MTCDs can get their missing packets from other group members on direct link. Figure 9 shows the retrieved data comparison. A dotted line with circles represents the results of SoCast while a solid line with asterisk is for APRA. As the number of MTCDs increases, the number of helpers also increases. We can see that if the number of MTCDs is less, the probability of making a group and retrieving data is less, and also that the probability increases with the increasing number of MTCDs. APRA gives a better result as compared to SoCast. When there are a total of six MTCDs, the number of helpers for each MTCD is five. The probability of retrieving data is less. But as the number of MTCDs increases, the number of helpers also increases. Figure 10 demonstrates that when the number of packets increases, the probability of missing or lost packets increases. Hence, the probability of retrieving data from other MTCDs decreases in the case where the number of MTCDs is kept fixed.

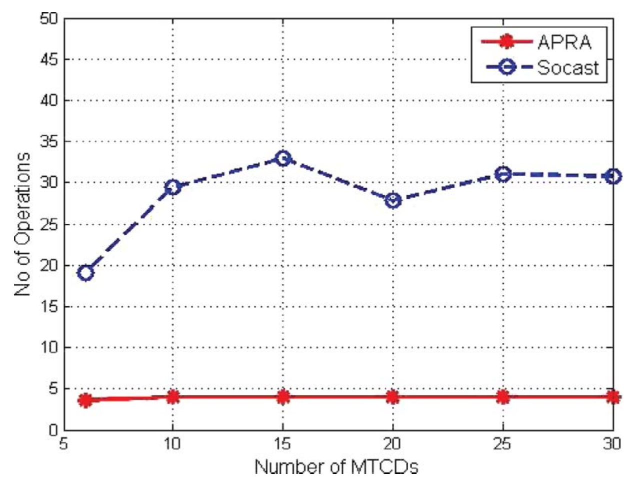

Figure 11: Computational complexity when the number of MTCDs is increased

5.2.2 Delay or Computational Complexity

Every time in SoCast, MTCDs have to make groups to retrieve data. Therefore, it will be more complex and takes more time as compared to APRA which in fact will take less time. In Figure 11, we can see that SoCast will have more number of operations as the number of MTCDs increases. But APRA gives a constant result, because when an MTCD wants data, it asks its head for the data and the head directs a helper to let it give the MTCD the data. Figure 12 explains that the number of operations will be increased by increasing the number of packets for SoCast. This is because it will take more time for an MTCD (having more missing packets) to make a reciprocity group and find a helper. On the other hand, in APRA, the greater number of data packets leads to the decreased number of operations.

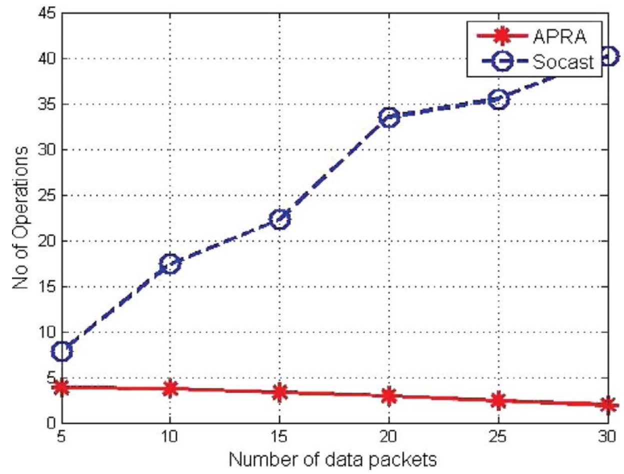

Figure 12: Computational complexity when the number of packets is increased

5.2.3 Energy Efficiency

In APRA, a group head will communicate with all the group members and with the eNB. Figure 13 shows the four cases of scheduling. The energy of the head will drain drastically as shown in Figure 13(a). According to the assumption, 20% of the energy is drained every 5 hours. Thus, it will take up to 50 hours to drain almost the whole energy. On the other hand, if we rotate the head among the members, the lifetime of a group will increase. In Figure 13(b), *i.e.*, Case 2, a head is rotated among three MTCDs, and it is clear that the overall lifetime of the group increases three times. In Case 2, we assume that only the group head consumes energy while the members are not consuming, and after 20% of energy consumption, the head will be changed to another one. In Case 3 (Figure 13(c)), the group head selection is based on energy level, such that one that has more energy will be selected as the group head. After 20% energy consumption, again compare and select the one having more energy as a head. In Case 2 and Case 3, we have rotated the head almost equally among the three members. For a real case, if we change the group head from one which is most appropriate to another, this will decrease the accuracy. Therefore, in Case 4, we have given preference to proximity (based on the design of AP algorithm) on energy. In Figure 13(d), the most appropriate group head is node *A*, which is selected as a head during most of the time. After many iterations, when node *A* has a deficiency of energy, the others are selected as a group head.

The average accuracy is shown in Figure 14. It can be clearly seen that Case 4 guarantees a higher average accuracy in consideration with proximity and energy while case 1 shows 100% accuracy for few transitions.

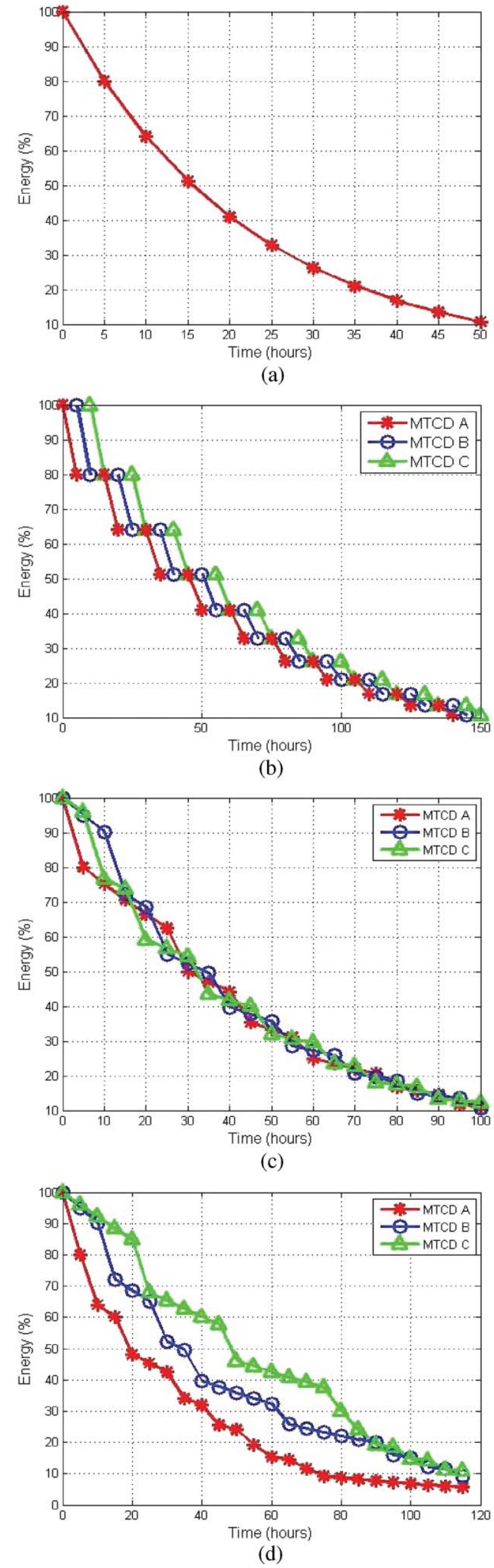

Figure 13: Scheduling. (a) Conventional method scheduling; (b) round robin scheduling; (c) energy-based scheduling; (d) proximity-and-energy-based scheduling

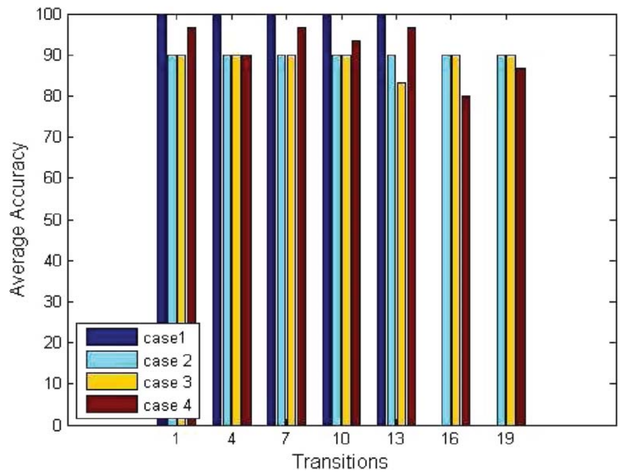

Figure 14: Average accuracy

Case 2 has a 90% accuracy until all the members are out of energy, because in this case, round robin scheduling is used. In Case 3, only energy is consider as a head selection parameter.

6. CONCLUSION

This paper proposed an AP-based RA scheme. We elaborated some major issues in M2M and via simulation results we showed that these issues can be overcome by APRA. We also proposed a RA method that different groups can use the same resources at the same time, leading to spectrum efficiency and system capacity maximization. We applied different methods to rotate group head to increase the overall energy efficiency of the system. As future work, APRA for mobile devices as MTCDs will be studied.

FUNDING

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (Research and Development of 5G Mobile Communications Technologies using CCN-based Multi-dimensional Scalability) [number 2013-0-00409], and also by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education [number 2017R1D1A1B03035885].

ORCID

Hamayoun Shahwani http://orcid.org/0000-0003-2211-8360
Phuc Chau http://orcid.org/0000-0001-7510-3854
Jaehoon (Paul) Jeong http://orcid.org/0000-0001-8490-758X
Jitae Shin http://orcid.org/0000-0002-2599-3331

REFERENCES

1. E. Schneidar. (2013). "The future of M2M and IOT vision for 2020," *M2M J. 20*, pp. 5–18.
2. A. Asadi, Q. Wang and V. Mancuso. (2014). "A survey on device-to-device communication in cellular networks.," *IEEE Commun. Surv. Tut. 16 (4)*, pp. 1801–1819.
3. J. Kim, J. Lee, J. Kim and J. Yun. (2014). "M2M service plate forms: Surveys, issues and enabling technologies," *IEEE Commun. Surv. Tut. 16 (1)*, pp. 61–76.
4. B. Insight and G. Burg, *The Global Wireless MTC Market.*, 2nd ed., Gothenburg: Berg Insight, 2009.
5. Towards 50 billion connected devices, Ericsson. Available: http://www.ericsson.com/au/res/region_RASO/docs/2010/ericsson-50-billion-paper.pdf
6. M. Hasan, E. Hussain and D. Niyato. (2013). "Random access for machine-to-machine communication in LTE advance network: Issues and approaches," *IEEE Commun. Mag. 51*, pp. 86–93.
7. B. J. Frey and D. Dueck. (2007). "Clustering by passing messages between data points," *Sci. Mag. 315*, pp. 972–976.
8. S.-Y. Lien, K.-C. Chen and Y. Lin. (2011). "Towards ubiquitous massive access in 3GPP machine-to-machine communications," *IEEE Commun. Mag. 49*, pp. 66–74.
9. F. Wang, L. Song, Z. Han, Q. Zhao and X. Wang, "Joint scheduling and resource allocation for device-to-device underlay communications," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, 2013, pp. 134–139.
10. Y. Cao, X. Chen, T. Jiang and J. Zhang, "Socast: Social ties based cooperative video multicast," in *IEEE Conference on Computer Communications, INFOCOM, Toronto, Ontario*, 2014, pp. 415–423.
11. N. Glorezaci, A. F. Molisch and A. G. Dimakis. (2014). "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun. 13 (7)*, pp. 3665–3676.
12. D. You and D. H. Kim, "Multi-device-to-multi-device communication in cellular network for efficient contents distribution," in *IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2014, pp. 244–247.
13. A. Pal. (2010). "Localization algorithms in wireless sensor networks: Current approaches and future challenges," *Network Protocols Algorithms., 2 (1)*, pp. 45–73.
14. L. Song, Z. Han and C. Xu. *Resource management for device-to-device underlay communication*, New York, NY: SpringerBriefs, 2014.

15. X. Y. Zhu, S. Wen, et al. (2012). "A cross-layer study: Information correlation based scheduling scheme for device-to-device radio underlaying cellular networks," ICT, pp. 1–6.

16. S. Xu, H. Wang, T. Chen, Q. Huang and T. Peng, "Effective interference cancellation scheme for device-to-device communication underlaying cellular networks," in *Vehicular Technology Conference Fall, IEEE, 2010*, vol. 72, pp. 1–5.

17. S. Sun and Y. Shin, "Resource allocation for D2D communication using practical swarm optimization in LTE networks," in *Information and Communication Technology Convergence (ICTC), Oct. 2014*, pp. 371–376.

18. F. Fodor. (1999). "The densest packing of 19 congruent circles in a circle.," Geom. Dedic. *78 (2)*, pp. 139–145.

APPENDIX

We split Algorithm 2, that is APRA algorithm, into three parts. The first part is group formation that is shown via Algorithm 3. Input similarities are checked in line 4, and after similarity check, the machine types are checked for all MTCDs. MTCDs having similar machine types will make one group. From lines 14 to 24, the message-passing procedure is explained for selecting a group head. Responsibility and availability messages are exchanged to make groups. In the end of group formation, each groups head will send the necessary information to eNB regarding their groups. Algorithm 4 elaborates the channel allocation procedure. The same CUE channel will be allocated to X number of MTCD groups by following the constraints in 10, 11, and 12. Algorithm 5 explains the process of energy efficiency. We assume that once the group head is selected, it will remain group head until 20% of its energy is drained. After that we apply Markov chain to select another group head which would be the most appropriate head in terms of energy and proximity to maintain the accuracy.

Algorithm 3 Group Formation

Input: Position and type identities of machine.
Output: Group Head (GH).
1: **function** PROCESS AP TECHNIQUE
2: **for each** MTCD M **do**
3: /* Start similarity measures */
4: **if** $(s(i,k) = -||x_i - x_k||^2) \geq$ -TH **then**
5: /* Compute machine type check */
6: $Type_i \sim Type_k$
7: **if** $Type_i \sim Type_k \leftarrow TRUE$ **then**
8: $Type_i$ and $Type_k$ make one group
9: **else** $Type_i \sim Type_k \leftarrow FALSE$
10: $Type_i$ and $Type_k$ make separate group
11: **end if**
12: **end if**
13: **end for**
14: /* After similarity measures and machine type check, the responsibility and availability messages are exchanged between data point and GH */
15: **for each** MTCD M **do**
16: $s(k,k) \leftarrow p$ where p is a priori preference value
17: $a(i,k) \leftarrow 0$ (initialization)
18: /* Send responsibility and availability messages */
19: **if** $r(i,k)$ *and* $a(i,k) \leftarrow true$ **then**
20: Maximize $a(i,k) + r(i,k)$, that can identify point k as GH
21: **else**
22: Do the process for another GH
23: **end if**
24: **end for**
25: /* After completion of groups formation */
26: **for each** Group G **do**
27: GH sends "Group information" to eNB (containing ID_H, L_H, D_g, m_g)
28: **end for**
29: **end function**

Algorithm 4 Channel Allocation

function PROCESS CHANNEL ALLOCATION
/* For channel allocation GH_i sends $PRACH$ request to eNB */
for each CUE channel **do**
if $A_c \cap 3D_g = \varnothing$ *and* $3D_g \cap 3D_{g'} = \varnothing$ **then**
$CH_m \leq X$
eNB allocates the same channels to X MTCD groups if there is nothing common in their coverage area.
else
Discard request
end if
end for
end function

Algorithm 5 Energy Efficiency

function PROCESS ENERGY EFFICIENCY
for each Group member **do**
/*Check energy level*/
if GH_i *energy discharge* $\geq 20\%$ **then**
/*Apply Markov chain*/
if $GH_k(prox + enr) > GH_l(prox + enr)$ **then**
GH_k is selected as new GH
else
Check for another GH
end if
else
No transition
end if
end for
end function

Authors

Hamayoun Shahwani received the BS degree from Balochistan University of Information Technology, Engineering and Management Sciences in 2010. He is currently an integrated (MS leading to PhD) student in the Department of Electronic, Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University, Republic of Korea. His research interests include machine-to-machine communication, 5G communication systems, and vehicular ad hoc networks.

E-mail: hamayoun@skku.edu

Phuc Chau received the BS degree from Hochiminh University of Science, Vietnam in 2010. After working 2 years at the Hochiminh University of Science as teaching assistant, he is currently a PhD candidate in the Department of Electronic, Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University, Republic of Korea. His research interests include video signal processing and transmission over next generation Internet and wireless/mobile networks, 5G communication systems, and multimedia network control/ protocol issues.

E-mail: cmphuc@skku.edu

Jaehoon (Paul) Jeong is an assistant professor in the Department of Software at Sungkyunkwan University in Korea. He received his PhD degree from the Department of Computer Science and Engineering at the University of Minnesota in 2009. He received his BS degree from the Department of Information Engineering at Sungkyunkwan University and his MS degree from the School of Computer Science and Engineering at Seoul National University in Korea, in 1999 and 2001, respectively. His research areas are vehicular networks, cyber-physical systems, IoT, wireless sensor networks, mobile ad hoc networks, network softwarization, and network security. His two data forwarding schemes (called TBD and TSF) for vehicular networks were selected as spotlight papers in IEEE Transactions on Parallel and Distributed Systems in 2011 and in IEEE Transactions on Mobile Computing in 2012, respectively. Dr Jeong is a member of ACM, IEEE, and the IEEE Computer Society. Dr Jeong is the corresponding author.

E-mail: pauljeong@skku.edu

Jitae Shin received his BS degree from Seoul National University in 1986 and his MS degree from Korea Advanced Institute of Science and Technology (KAIST) in 1988. After working around eight years at Korea Electric Power Corp. and the Korea Atomic Energy Research Institute, he returned to study and received MS and PhD degrees in electrical engineering from the University of Southern California, Los Angeles, in 1998 and 2001, respectively. He is a professor in the School of Electronic and Electrical Engineering of Sungkyunkwan University, Suwon, Republic of Korea. His current research interests include image/video signal processing, video transmission over wireless/mobile communication systems, and multimedia network control/protocol issues.

E-mail: jtshin@skku.edu