Adaptive Resource Discovery in Mobile Cloud Computing

Wei Liu[a,*], Takayuki Nishio[a], Ryoichi Shinkuma[a], Tatsuro Takahashi[a]

[a]*Graduate school of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan*

Abstract

Mobile Cloud Computing (MCC) is aimed at integrating mobile devices with cloud computing. It is one of the most important concepts that have emerged in the last few years. Mobile devices, in the traditional agent-client architecture of MCC, only utilize resources in the cloud to enhance their functionalities. However, modern mobile devices have many more resources than before. As a result, researchers have begun to consider the possibility of mobile devices themselves sharing resources. This is called the cooperation-based architecture of MCC. Resource discovery is one of the most important issues that need to be solved to achieve this goal. Most of the existing work on resource discovery has adopted a fixed choice of centralized or flooding strategies. Many improved versions of energy-efficient methods based on both strategies have been proposed by researchers due to the limited battery life of mobile devices. This paper proposes a novel adaptive method of resource discovery from a different point of view to distinguish it from existing work. The proposed method automatically transforms between centralized and flooding strategies to save energy according to different network environments. Theoretical models of both energy consumption and the quality of response information are presented in this paper. A heuristic algorithm was also designed to implement the new adaptive method of resource discovery. The results from simulations demonstrated the effectiveness of the strategy and the efficiency of the proposed heuristic method.

Keywords: adaptive resource discovery, energy-efficient, heterogeneous wireless networks, mobile cloud computing

1. Introduction

Cloud computing, which is aimed at providing infrastructures, platforms and software as services has been introduced and implemented in the last few years. It is widely recognized as the next generation of computing architecture. Wireless communication technologies have simultaneously been extensively developed. Different kinds of wireless networks like the third generation of mobile telecommunications technology (3G), Bluetooth, wireless local area networks (WLANs) and worldwide interoperability for microwave access (WIMAX) have become available in our daily lives. Users can choose different networks according to different requirements. The network connectivity and data throughput of mobile devices have been greatly improved. Therefore, the integration of mobile devices with cloud computing has attracted a great deal of attention from both industrial and academic communities because of its potential value. Mobile Cloud Computing (MCC) has been widely accepted as one of the most important solutions to this issue.

MCC can be roughly divided into two different architectures: agent-client based and cooperation-based [1]. The cloud (data center), in the agent-client based architecture, provides overall resource management for mobile devices. Mobile devices use resources in the cloud to enhance their functionalities and improve their processing abilities (e.g., data storage and processing speed). However, along with the development of hardware and software technologies, modern mobile devices like smart phones and tablets have many more resources than before, e.g., computing, communication, sensor and software-application resources [2]. As a result, two shortcomings in the agent-client based architecture have emerged: (1) available resources in the mobile devices themselves are not utilized efficiently and (2) long delays are caused by the long distance between the cloud and mobile devices. To solve these problems, cooperation-based MCC views the mobile devices and other fixed wireless devices (e.g., wireless routers and sensors) as part of the cloud. Available resources in these devices could be shared among themselves through wireless communication. Delays could also be reduced by devices benefiting from high

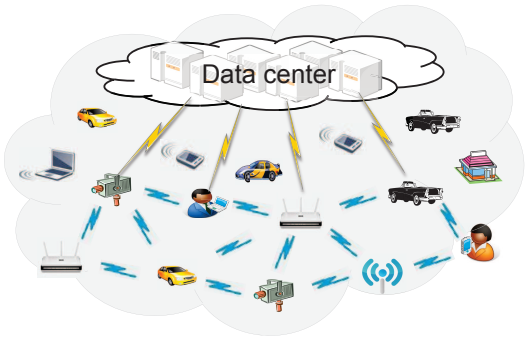


Fig. 1: Architecture for cooperation-based MCC.

*corresponding author. Tel., +81-75-753-4830
Email address: liu@cube.kuee.kyoto-u.ac.jp (Wei Liu)

throughput short-range communication and location proximity. The traditional cloud (data center) in the cooperation-based architecture plays a role as a scheduler in the collaboration by wireless devices. Of course, according to different contexts, the data center can also provide resources to mobile devices as it does in the agent-client based architecture. Due to its huge potential benefits, cooperation-based architecture has become the most interesting research area in MCC [3][4]. This new architecture of MCC has also been defined as "Fog Computing" in [3]. However, to achieve cooperative resource sharing among wireless devices, it is quite important to find how available resources in nearby devices are discovered. This paper introduces an energy-efficient method of adaptive resource discovery to solve this problem.

Much research on resource discovery has been published [5–15]. However, most of the existing work has adopted a fixed strategy for resource discovery and failed to adapt this strategy based on different network resource statuses, e.g., the degree of resource scarcity and the pattern of resource requirements (to differentiate it from "Network Status", which mainly refers to network traffic and bandwidth, Network Resource Status (NRS) is used in this paper to represent the characteristics of resource distribution and usage in the network). Apart from that, more resources consume more energy. Battery capacity also becomes a bottleneck in wireless applications. Consequently, more and more research [9, 10, 14, 15] has aimed at providing energy-efficient solutions to resource discovery. However, there are two main problems in their research: (1) most of them have saved energy through sacrificing other important quality metrics like the accuracy and coverage of response information without formal quantitative analysis and (2) they have only taken into consideration resource discovery and energy consumption in homogeneous networks like 3G cellular or ad hoc WLANs alone. Obviously, energy consumption in heterogeneous networks is more realistic and more important for modern society.

This paper proposes an energy-efficient method of resource discovery that automatically transforms between centralized and flooding strategies according to different NRSs. The three main contributions of this paper are: (1) According to the best of our knowledge, this is the first proposal that has introduced an adaptive solution to resource discovery based on strategy transformations and the first work that has taken into consideration resource discovery in heterogeneous wireless networks. (2) We also established theoretical models of energy efficiency and quality of response information. (3) A heuristic algorithm was designed to implement the proposal and it was proved to be energy-efficient through extensive simulations.

In the rest of this paper, Section 2 introduces our system model. Our analysis of the proposed method of adaptive resource discovery is presented in Section 3. The heuristic algorithm is introduced in Section 4. An extension of the proposed method is presented in Section 5. Section 6 explains how we evaluated the adaptive method through extensive simulations. Related work is discussed in Section 7. Conclusions are drawn and future work is discussed in the last section.

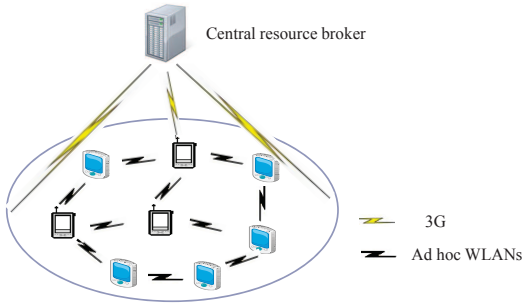


Fig. 2: System architecture.

2. System model

We assumed that mobile nodes were in heterogeneous wireless networks including both 3G cellular and ad hoc WLANs in this paper. Fig. 2 outlines the scenario. There is a central base station in the 3G cellular network that is able to communicate with all nodes in the area. We called it the central resource broker (CRB) in the background of resource sharing. Apart from that, every node can communicate with nearby nodes through a ad hoc WLAN. The assumed communication abilities are common in current smart phones and other devices. Nodes are assumed to be uniformly distributed throughout the area ([16] provides some application scenarios using this assumption. The effect of node mobility, which may violate this assumption, has been left for future work). Nodes either maintain a resource directory in the CRB through a widely-covered 3G network (a centralized mode) or flood resource requests in the area through a short-range ad-hoc WLAN (a flooding mode) to discover resources.

Different resource discovery modes consume different amounts of energy. We tried to minimize energy consumption through transformations between the two modes according to different NRSs. Time is divided into consecutive time slots in our model. We define x_i as an indicator that specifies whether a centralized or flooding mode is selected in time slot i :

$$x_i = \begin{cases} 1 & \text{centralized mode is selected} \\ 0 & \text{flooding mode is selected .} \end{cases} \quad (1)$$

Accordingly, $E_i(x_i)$ is defined as the energy consumed in time slot i based on different values of x_i. Without loss of generality, the period from time slot 1 to time slot Q is considered. The optimization problem is defined as the selection of an Q-dimensional vector comprised of x_i that minimizes the energy consumed by all 2^Q candidates, while keeping the expected value of resource information availability (RIA is a quality metric of response information defined in Subsection 3.2) no less than a threshold R_{thresh}. R_{thresh} is a real value in [0,1].

$$\begin{aligned} &\text{objective:} && \min \textstyle\sum_{i=1}^{Q} E_i(x_i) \\ &\text{subject to:} \\ &\quad x_i && = \ 0 \text{ or } 1 \\ &\quad E[RIA] && \geq \ R_{thresh} \end{aligned}$$

3. Proposed method of adaptive resource discovery

3.1. General description

As described in Section 2, nodes can discover available resources through both 3G cellular and ad-hoc WLANs. There are two modes in the proposed method:

(1) Centralized mode: A resource directory is maintained in the CRB in this mode. If a node wants to allocate resources to its tasks, it first checks whether it has the resources itself. If not, it sends a resource request to the CRB through the 3G network. The CRB returns the identifications of nodes in which the required resources are available.

(2) Flooding mode: We adopted on-demand flooding [10, 17] for the flooding mode in this paper. No resource information is maintained in the CRB in this mode. If node U wants to allocate resources to its tasks, it first checks whether it has the resources itself. If not, it sends a resource request (with a unique sequence number) as a broadcast package through the ad hoc WLAN, which is received by all the nodes within the wireless transmission range of U. When another node receives a new resource request, if it has the required resources, it replies to the resource requester with its identification. Every receiving node decreases the time-to-live (TTL)[1] value of the resource request by one. The node propagates this resource request by transmitting it as a broadcast package (with the same sequence number) if the remaining TTL value is positive. However, if this node receives a duplicate resource request that has already been dealt with, it discards the request to avoid duplicate propagations.

Nodes, automatically transform between the centralized and flooding modes to save energy in the proposed adaptive method of resource discovery based on different NRSs. Four sequential steps are executed. (1) Time is divided into consecutive time slots, (2) nodes send statistics to the CRB at the end of each time slot according to their experiences in the last time slot, (3) the CRB estimates the energy consumed by both modes based on the collected statistics, and (4) the CRB chooses the most energy-efficient method and notifies each node to use it in the next time slot.

3.2. Resource information availability and maintenance

Available resources in a node might change at certain times. This is affected by factors like task processing, environment changes, and remaining battery power. We have only taken into consideration the factor of task processing that is generated from two sources in this paper.

(1) Nodes allocate resources to a task from themselves.

(2) Nodes allocate resources to a task from other nodes.

A task is defined in this paper as a job generated by users that consumes resources to finish it. Different tasks consume different kinds of resources, e.g., image processing consumes CPU resources (FLOPS) and data transmission consumes bandwidth resources (Mbps). Since the proposal in this paper is not constrained by specific types of tasks or resources, we can encapsulate these details with the concept of the abstract size of tasks and units of resources in the following descriptions. Task processing occupies resources while the end of task processing releases occupied resources.

[1]The time-to-live value is the number of hops that the package of a resource request can take through the network before it gets discarded.

Table 1: Parameters for RIA maintenance.

S	Expected task size of a request for resource A
R	Sum of resources A in all nodes
λ_A	Number of generated tasks for resource A
λ_{A-o}	Number of generated resource requests for resource A
T	Length of one time slot
T'	Processing time for one task with all resources A
N	Maximum number of tasks that can be processed
$F_{A-regist}$	Expected number of updates for resource A
F_{regist}	Expected number of updates for all resource types

Resource information availability (RIA), which reflects the quality of response information, is defined as: the possibility that the responses to a request will accurately include all available resource information. This includes two aspects: accuracy and coverage. Energy is consumed to maintain RIA.

3.2.1. RIA in centralized mode

Every node benefits from the wide coverage of 3G network and can register its resource information with the CRB in the centralized mode. The coverage aspect of RIA is completely maintained. However, when the amount of available resources changes, nodes should update resource information in the CRB to maintain the accuracy aspect of RIA.

The number of resource information updates to maintain RIA must be calculated to estimate the energy consumed by the centralized mode. First, only one type of resource called A is considered. All the model parameters listed in Table 1 are for one time slot.

The processing time for one task with all available resources A in the nodes is:

$$T' = \frac{S}{R} . \tag{2}$$

The capacity of all nodes that indicates the maximum number of tasks that can be processed in one time slot is:

$$N = \frac{T}{T'} . \tag{3}$$

Depending on the relationship between λ_A and N, there are two situations for the number of updates in one time slot:

(1) $\lambda_A > N$: the number of tasks that can be processed in one time slot is constrained by the capacity of all nodes N. A node needs to update resource information at both when resources are allocated and released. Consequently, $F_{A-regist}$ can be calculated as:

$$F_{A-regist} = 2 \times N = \frac{2 \times R \times T}{S} . \tag{4}$$

(2) $\lambda_A \leq N$: the number of tasks that can be processed in one time slot is determined by the number of generated tasks λ_A. $F_{A-regist}$ is:

$$F_{A-regist} = 2 \times \lambda_A . \tag{5}$$

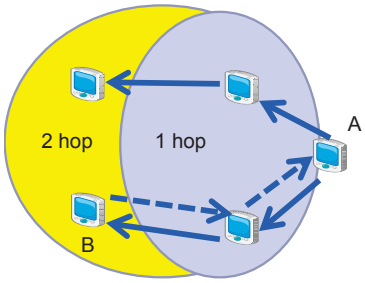


Fig. 3: RIA in flooding mode.

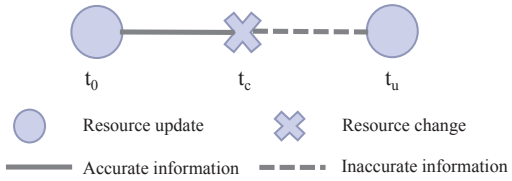


Fig. 4: Tradeoff in centralized mode.

Obviously, the number of updates for all types of resources in one time slot is:

$$F_{regist} = \sum F_{I-regist} \,, \tag{6}$$

where I stands for different types of resources.

3.2.2. RIA in flooding mode

Because on-demand flooding is adopted, nodes in the flooding mode fully know what their available resources are when they receive a request. The accuracy aspect of RIA is completely maintained. However, the requester has to set a large enough TTL value to ensure request packages reach every node in the area to maintain the coverage aspect of RIA.

To estimate the appropriate TTL value, we define P_i as the expected percentage of extra nodes that will receive the requests when their TTL value increases from $i-1$ to i, e.g., $P_1 = 40\%$ in Fig. 3, since 40% of new nodes will receive the requests when their TTL value increases from 0 to 1. We define $P_0 = 1/N_{node}$, which is the percentage occupied by the requester itself. N_{node} is the number of nodes in the area. As a result, TTL value k should satisfy the following equation asymptotically to provide complete coverage of nodes:

$$\sum_{i=0}^{k} P_i = 1 \,, \tag{7}$$

where k is the minimum integer value that satisfies the equation, e.g., k is 2 for the network topology depicted in Fig. 3, since $P_0 + P_1 + P_2 = 1$. The calculation of P_i depends on three parameters of (1) the wireless transmission range of the nodes, (2) the number of nodes in the area, and (3) the size of the area (length and width). The detailed method and equations have not been explained in this paper because they are beyond its scope. However, a detailed description is available in [18].

3.3. Tradeoff between RIA and energy consumption

We assumed the RIA was completely maintained in the previous model. However, nodes may also agree to a lower RIA to save energy. This subsection discusses our analysis of the tradeoff between the two metrics.

3.3.1. Tradeoff in centralized mode

A node updated resource information right after its available resources had changed in Subsection 3.2.1. Instead of that, it could wait for a period of time before the update. Because update frequency decreases, less energy is consumed with this strategy.

As Fig. 4 shows, t_0 is the time for the previous resource update. The t_c is the time when the available resources changed. The node did not update information in the CRB until t_u. Resource requests generated between t_c and t_u were responded to with inaccurate information by the CRB. If t_u is further from t_c, the expected RIA from t_0 to t_u decreases. If t_u is further from t_c, on the other hand, update frequency F_{regist} also decreases, and less energy is consumed to maintain the RIA.

Resource requests from other nodes are assumed to be a Poisson arriving process in the following analysis. Let H represent a random variable defined as:

$$H = \begin{cases} 1 & \text{response to a request is accurate} \\ 0 & \text{response to a request is inaccurate} \,. \end{cases} \tag{8}$$

We need to find expected value $E[H]$ for one request:

$$E[H] = \int_0^{T_{0-u}} f(t) \times E[H|T_{rc} = t]dt \,, \tag{9}$$

where T_{0-u} is the length of the period from t_0 to t_u, T_{rc} is a random variable of t_c, and $f(t)$ is the probability density function of T_{rc}. Assuming a Poission arriving process, the arrival of requests are uniformly distributed in T_{0-u} provided a request occurs within this period. Therefore, $f(t)$ is:

$$f(t) = \frac{1}{T_{0-u}} \,. \tag{10}$$

$E[H|T_{rc} = t_c]$ is the expected value of H provided that available resources changed at time t_c. Obviously, the requests generated between t_0 and t_c are responded to with accurate information. Based on a uniform distribution for the arrival time of requests, the expected value of H is calculated as:

$$E[H|T_{rc} = t_c] = \frac{t_c - t_0}{t_u - t_0} \,. \tag{11}$$

Nodes can keep $E[H|T_{rc} = t_c]$ constant through choosing t_u according to t_0 and t_c. Let $\alpha = (t_u - t_c)/(t_c - t_0)$. Based on Eqs. (9), (10), and (11), $E[H]$ can be computed as:

$$E[H] = E[H|T_{rc} = t_c] \int_0^{T_{0-u}} f(t)dt = \frac{t_c - t_0}{t_u - t_0} = \frac{1}{1+\alpha} \,. \tag{12}$$

To keep $E[H]$ no less than R_{thresh}, the choice of t_u should make α satisfy the inequality:

$$\alpha \leq \frac{1 - R_{thresh}}{R_{thresh}} . \quad (13)$$

The resulting expected number of updates in one time slot is reduced to:

$$F'_{regist} = F_{regist} \times \frac{t_c - t_0}{t_u - t_0} = \frac{F_{regist}}{1 + \alpha} , \quad (14)$$

since the update interval increases from $(t_c - t_0)$ to $(t_u - t_0)$.

3.3.2. Tradeoff in flooding mode

The TTL value of resource requests in Subsection 3.2.2 was assumed to be large enough to reach every node in the area. A larger TTL value means a larger probability of discovering the required resource. However, it also consumes more energy to propagate requests. If a lower RIA is acceptable, a smaller TTL value could be used to save energy.

The method of evaluation is nearly the same as that in Eq. (7). However, only an R_{thresh} fraction of nodes are assumed to be covered by the requests:

$$\sum_{i=0}^{k} P_i \geq R_{thresh} , \quad (15)$$

where k is the minimum integer value that satisfies the inequality, e.g., when R_{thresh} is 0.5, k is 1 for the network topology depicted in Fig. 3, since $P_0 + P_1 = 0.6$.

3.4. Energy consumption models

This subsection presents the energy consumption models for both modes. We define N_{resp} as the average number of responses to a resource request. λ_o is the number of resource requests for all types of resources in one time slot.

The energy consumed by the centralized mode in one time slot can be calculated as:

$$\begin{aligned} E_{central} &= (\lambda_o \times E_{3G-trans}) + (N_{resp} \times \lambda_o \times E_{3G-recv}) \\ &+ (F_{regist} \times E_{3G-trans}) , \end{aligned} \quad (16)$$

where $E_{3G-trans}$ and $E_{3G-recv}$ are the energy consumed by transmission and reception through the 3G interface. The total consumption of $E_{central}$ includes three parts: sending resource requests to the CRB, $\lambda_0 \times E_{3G-trans}$, receiving resource responses from the CRB, $N_{resp} \times \lambda_0 \times E_{3G-recv}$, and sending updates to the CRB to maintain the RIA, $F_{regist} \times E_{3G-trans}$.

In the flooding mode, the expected number of neighboring nodes that are within the wireless transmission range of a node, N_{neig}, is equal to the expected number of extra nodes that will receive the resource requests when their TTL value increases from 0 to 1:

$$N_{neig} = P_1 \times N_{node} . \quad (17)$$

The average distance HC (hop count) from the resource requester to the providers is:

$$HC = \sum_{i=0}^{k} P_i \times i . \quad (18)$$

W_{trans} and W_{recv} are the energy consumed by transmission and reception through the ad-hoc WLAN interface. Only P_i percentage of nodes that receive a resource request for the first time at hop i will propagate the request when i is not the last hop. The energy consumed by propagating the request is $(\sum_{i=0}^{k-1} P_i \times N_{node}) \times W_{trans}$. All the neighbors of propagating nodes will receive the request. The energy consumed by receiving the request is $(\sum_{i=0}^{k-1} P_i \times N_{node} \times N_{neig}) \times W_{recv}$. Only unicasting is needed when returning response messages. The energy consumed by relaying and receiving responses is $N_{resp} \times (W_{trans} + W_{recv}) \times HC$, where HC is derived from Eq. (18). The energy consumed by one request in the flooding mode is the sum of the three previous parts:

$$\begin{aligned} E'_{flooding} = &\sum_{i=0}^{k-1} P_i \times N_{node} \times W_{trans} + \sum_{i=0}^{k-1} P_i \times N_{node} \\ &\times N_{neig} \times W_{recv} + N_{resp} \times (W_{trans} + W_{recv}) \times HC . \end{aligned} \quad (19)$$

Therefore, the energy consumed by all requests in one time slot is:

$$E_{flooding} = \lambda_o \times E'_{flooding} . \quad (20)$$

4. Proposed heuristic algorithm

This subsection explains how we implemented the proposed adaptive method through a heuristic algorithm. According to the energy consumption models in the last section, three statistics are needed to estimate the energy consumed by two modes in a time slot. These are:

(1) The number of generated resource requests λ_o ,

(2) The number of generated resource updates F_{regist} , and

(3) The average number of responses for each request N_{resp} .

Distributed nodes send the previous three statistics to the CRB at the end of each time slot according to their experiences in that time slot. The CRB processes raw data (e.g., sums up λ_o from all the nodes) and stores the records of previous N_{slot} time slots in a list. After initialization, the *Check* part tests whether the NRS is too dynamic to be predicted. If there are more than N_{trans} transformations of the discovery mode in the retained records, the NRS is assumed to be unpredictable. The centralized mode is used until the NRS becomes relatively regular. If the energy consumption ratio of a better mode to a worse mode is less than C_{thresh} in the *Large_diff* part, the better mode is directly chosen. If none of the previous conditions are satisfied, the CRB uses the average value of previous N_{trend} records to predict the NRS of the next time slot in the *Prediction* part. The CRB chooses an energy-efficient mode and sends the decision to every node. If a transformation is from the flooding

Heuristic_algorithm ($N_{slot}, N_{trans}, C_{thresh}, N_{trend}, P_{thresh}$)

Initialize :
preprocess raw data and create a new record r.
insert r into the list and delete outdated records.

Check :
if (there are N_{trans} or more transformations in the record list)
choose centralized mode.
goto *Make_choice*.

Large_diff :
if (better consumption / worse consumption $\leq C_{thresh}$)
choose better mode.
goto *Make_choice*.

Prediction :
//X_{i-j} means the statistic X_j in the i-th record
$\lambda_o = \frac{\sum_{i=1}^{N_{trend}} \lambda_{i-o}}{N_{trend}}$, $F_{regist} = \frac{\sum_{i=1}^{N_{trend}} F_{i-regist}}{N_{trend}}$,

$N_{resp} = \frac{\sum_{i=1}^{N_{trend}} N_{i-resp}}{N_{trend}}$.

use λ_o , F_{regist}, N_{resp} to estimate energy consumption.
if (better consumption / worse consumption $\geq P_{thresh}$)
remain the current mode.
else
choose the better mode.

Make_choice :
send decision to every node.

mode to the centralized mode, nodes should update their resource information in the CRB first to ensure RIA. The pseudocode describes the algorithm.

The N_{slot} in the heuristic algorithm defines how long the records of passed time slots are kept. The ratio of N_{trans} to N_{slot} indicates the degree of dynamicism of the NRS. This is used to prevent meaningless transformations when the network is too dynamic to be predicted. C_{thresh} is just a threshold value. If the discrepancy between two modes is quite large, it is highly probable that the better mode will be chosen in the next time slot except for unpredictably heavy changes in status. N_{trend} indicates the smoothness of NRS changes. Because transformation also consumes energy, P_{thresh} prevents transformation when the difference between the two modes is small.

5. Extension of flooding method

The proposed adaptive method in the previous sections only included basic centralized and flooding methods. However, it can also be provided with improved versions of both methods. A simple extended flooding method is introduced in this section as an optional choice to illustrate this.

Intuitively, if more nodes have the required resources, there is a larger chance of finding available providers of resources with a smaller TTL value. Moreover, if fewer tasks are generated by nodes, it is also easier to find available providers of resources with a smaller TTL value. This is due to the fact that most of the resources are not used. Consequently, nodes can send related statistics to the CRB. The CRB calculates a reasonable TTL value for resource requests based on the collected statistics.

Table 2: Parameters for method of the extended flooding.

RES_{A-dens}	Percentage of nodes that has resources A (resource density)
RES_{A-con}	Expected percentage of resources A occupied by tasks (consumption ratio)
$RES_{A-con-A}$	Average number of resources A occupied by each task
RES_{A-avai}	Expected percentage of available resources A
R_{A-avai}	Expected percentage of nodes having available resources A
N_{A-res}	Average number of resources A each node owns
λ_A	Number of generated tasks for resource A
k_{ex}	TTL value used by extended flooding method
k	TTL value used by basic flooding method

First, the method of calculating the TTL value with the extended flooding method is analyzed. One type of resource called A is considered. Related parameters are summarized in Table 2. The resource consumption ratio RES_{A-con} is the ratio of occupied resources A to the number of resources A in the area:

$$RES_{A-con} = \frac{\lambda_A \times RES_{A-con-A}}{RES_{A-dens} \times N_{node} \times N_{A-res}} , \quad (21)$$

where N_{node} is the number of nodes in the area.

According to Eq. (21), it is obvious that $0 \leq RES_{A-con} \leq 1$. The percentage of available resources A that has not been used is:

$$RES_{A-avai} = 1 - RES_{A-con} . \quad (22)$$

As a result, the expected percentage of nodes that still owns available resources A satisfies the following inequality:

$$R_{A-avai} \geq RES_{A-dens} \times RES_{A-avai} . \quad (23)$$

To find at least one available provider of resources A, the TTL value should be large enough to cover more than $1-R_{A-avai}$ percentage of nodes:

$$\sum_{i=0}^{k_{ex}} P_i > 1 - Min(R_{A-avai}) = 1 - RES_{A-dens} \times RES_{A-avai} , \quad (24)$$

where k_{ex} is the smallest integer value that satisfies this inequality.

According to the described model, the CRB notifies nodes of TTL value k_{ex} for resources A. If a node failed to discover any provider with k_{ex}, it again tries to discover resources A with the basic flooding TTL value, k. The energy consumption model of the extended flooding method is nearly the same as that of the basic flooding method except that two different TTL values are used: k_{ex} and k. The energy consumption for resources A is:

$$E_{A-flooding} = E_{A-flooding}(k_{ex}) + E_{A-flooding}(k) . \quad (25)$$

The energy consumption for all kinds of resources is:

$$E_{flooding} = \sum E_{I-flooding} , \quad (26)$$

where I stands for different types of resources.

The heuristic algorithm has to be slightly changed to enable the extended flooding method to be integrated into the proposed adaptive method of resource discovery. Because the number of resource requests with TTL value k is unavailable in the centralized mode[2], only k_{ex} is used to estimate the energy consumption of the flooding mode. Both k_{ex} and k are used for estimation in the flooding mode. Although the methods of estimating energy are slightly different for the two modes, because of the conservative estimates in Eqs. (23) and (24), the proposed method is rarely affected. This is proved by the simulation results presented in Section 6. Because different types of resources are associated with different values of k_{ex}, the statistics (λ_o, F_{regist}, and N_{resp}) should also be sent separately for each type of resource.

It should be noted that the extended flooding method still reflects the tradeoff between RIA and energy consumption. Although it is energy-efficient, it only intends to discover at least one available provider of resources rather than all potential providers in the area.

6. Simulation evaluation

We verified the proposed adaptive method and heuristic algorithm through simulations. We compared the energy consumed by the proposed method with that consumed by the centralized and flooding methods. The proposed method was further divided into two sub-categories: (1) the basic adaptive method (adaptive-b), which was provided with the basic flooding method and (2) the extended adaptive method (adaptive-ex), which was provided with the extended flooding method. The extended flooding method relied on the CRB to calculate the TTL value. Consequently, the performance of the extended flooding method alone was meaningless. Therefore, this was not evaluated in the simulations. The energy consumption caused by different TTL values (k and k_{ex}) in the adaptive-ex method was not further distinguished. Requests with the basic TTL value k in all the following simulations, actually only occupied less than 5% of the overall energy consumption caused by the extended flooding method. Since the effectiveness of the proposed method mainly resulted from transforming discovery modes, we did not focus on improving the centralized or flooding method itself. However, the proposed method also benefitted from the integration of improved components. This was proved by the superior performance of the adaptive-ex method in the simulations that followed.

6.1. Parameters, definitions and assumptions

The nodes in the simulations were uniformly distributed in a rectangular area. The nodes could discover available resources through the 3G network in the centralized mode or the ad hoc WLAN in the flooding mode. We assumed one unit of resources could process one task within a time slot (this means the 'time slot length' parameter in Table 3 was not sensitive in the simulation. We chose 30 s just because of the tradeoff between the duration of simulation and machine processing capabilities). Every node generated tasks with an equal probability.

Table 3: Basic simulation parameters.

Rectangular area	$1000 \times 1000\ m$
Number of nodes	100
Resource density, RES_{dens}	100%
Ad hoc WLAN range	$250\ m$
3G transmit, $E_{3G-trans}$	20
3G receive, $E_{3G-recv}$	10
WLAN transmit, W_{trans}	1
WLAN receive, W_{recv}	0.5
Size of resource requests	1 KB
Size of response identifications	0.1 KB
Size of statistics messages	0.1 KB
Size of decision notification messages	0.1 KB
Time slot length	30s

The size of both the statistics message and the decision notification message were set to 0.1 KB. This was a conservative approximation. Indeed, only three integer (or float) variables of the payload were needed for the statistics messages. Only one bit of payload was needed for the decision notification messages. Since only a comparison between different methods was concerned, we adopted the results in [19] where the ratio of energy consumption between 3G and ad hoc WLANs was 20/1 without a specific unit.

The proposed adaptive methods chose the centralized mode when they started up. The expected RIA was 0.95 for both modes. The heuristic algorithm was initialized with $< N_{slot} = 5, N_{trans} = 3, C_{thresh} = 0.5, N_{trend} = 2, P_{thresh} = 0.9 >$. The parameters for the algorithm are further discussed in Subsection 6.4.3. Other basic simulation parameters are listed in Table 3.

The flooding mode consumed a great deal of bandwidth when there were many resource requests (λ_o). However, when λ_o was large, in the proposed adaptive solution to resource discovery, the energy consumed by the flooding mode was much greater than that consumed by the centralized mode. The proposed methods chose the centralized mode automatically to prevent "flooding storm". Because of this, channel collisions were rare when the messages were short like those exchanged in the resource discovery process. Therefore, we assumed an idealized communication channel in the simulations that followed.

Fig. 5 plots the energy consumed in one time slot for different modes with the basic parameters listed in Table 3. Each node had five units of resources in this simulation (flooding: the basic flooding method and flooding-ex: the extended flooding method). The discontinuity of energy consumed by the extended flooding method was due to the discontinuity of calculated TTL value k_{ex} for different numbers of generated tasks.

We divided the NRS into three regions according to the energy consumed by the centralized and basic flooding methods:

[2] Since the extended flooding method is not executed, the number of resource requests that failed to discover any provider with TTL value k_{ex} is unknown.

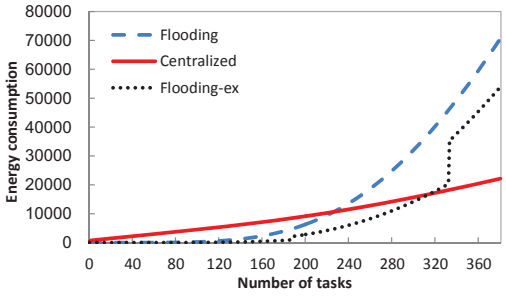


Fig. 5: Energy consumed by 5 units of resources.

(1) Flooding region (F-R): the energy consumed by the basic flooding method was less than that by the centralized method, e.g., the number of generated tasks was within the range of [0 , 222) in Fig. 5.

(2) Centralized region (C-R): the energy consumed by the centralized method was less than that by the basic flooding method, e.g., the number of generated tasks was within the range of [245 , $+\infty$) in Fig. 5.

(3) Cross region (CR-R): the energy consumed by the two methods was nearly the same. This was defined as the number of generated tasks within a shift range of 5% based on the crossing point, e.g., the number of generated tasks was within the range of [222 , 245) in Fig. 5.

Due to the large differences in energy consumption in different simulation scenarios, it was difficult to unify the y-axes (energy consumption) in all graphs. Since only comparisons in the same situation were concerned, this did not affect the simulation results. All the results in this section were an average value obtained from 100 simulation trials.

6.2. Performance in single region of NRS

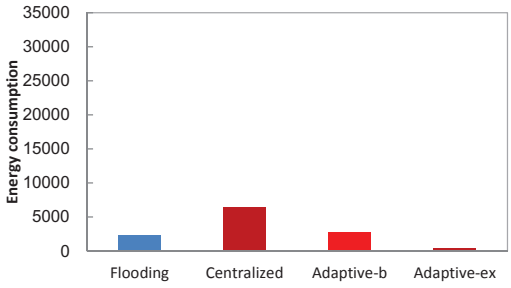
We verified the energy consumed by the proposed methods in different regions of NRS separately. The energy consumed by the proposed methods should ideally satisfy three characteristics:

(1) If the NRS is in F-R, the energy consumption should be near that consumed by the flooding method and less than that by the centralized method.

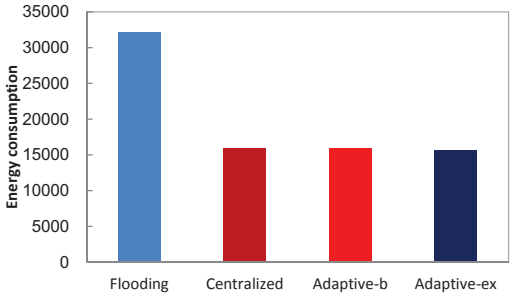
(2) If the NRS is in C-R, the energy consumption should be near that consumed by the centralized method and less than that by the flooding method.

(3) If the NRS is in CR-R, the energy consumption should be near that consumed by both methods. This should prevent meaningless transformations, which only waste energy .

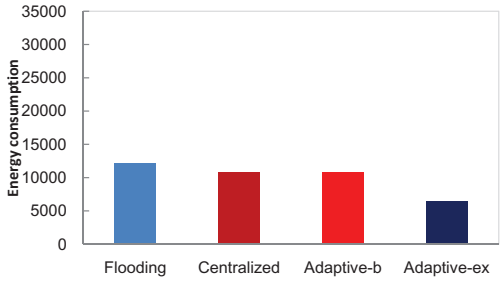
Every node had five units of resources in this simulation. The other parameters were the basic parameters summarized in Table 3. We chose 160 tasks to represent F-R, 300 tasks to represent C-R, and 240 tasks to represent CR-R (Fig. 5 shows the reasons for the choices). The average energy consumed by the different methods in one time slot is shown in Fig. 6. As the figures indicate, the proposed adaptive methods satisfied the previous three characteristics. When the flooding mode was selected, the adaptive-ex method consumed less energy than the



(a) Energy consumed in F-R



(b) Energy consumed in C-R



(c) Energy consumed in CR-R

Fig. 6: Energy consumed in single region of NRS.

adaptive-b method. This is due to the smaller TTL value chosen by the extended flooding method.

6.3. Performance in composite NRSs

The performance of the proposed methods was separately verified in different regions of the NRS, which was explained in Subsection 6.2. However, in reality, the NRS often exhibited various kinds of periodicities, e.g., in class and after class, on days and nights, and on weekdays and weekends. If the NRS had a period that included both F-R and C-R, the proposed methods automatically transformed between different modes to save energy. Two examples are given in this subsection to provide details on the energy consumed by the proposed methods in composite NRSs.

In example 1, the NRS transitions were caused due to the number of tasks generated by nodes. Every node had six units of resources in this example. In state S_1, 180 tasks were generated in a time slot. In state S_2, 360 tasks were generated in a time slot. As plotted in Fig. 9(c), S_1 belonged to F-R while S_2 belonged to C-R. The network stayed in S_1 for 10 time slots and

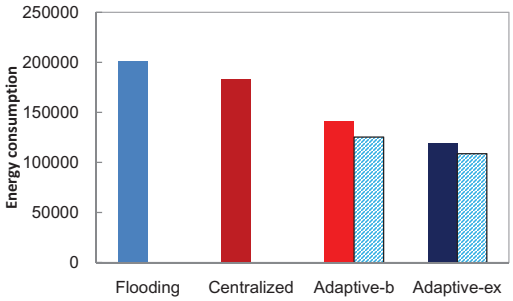


Fig. 7: Energy consumed in example 1.

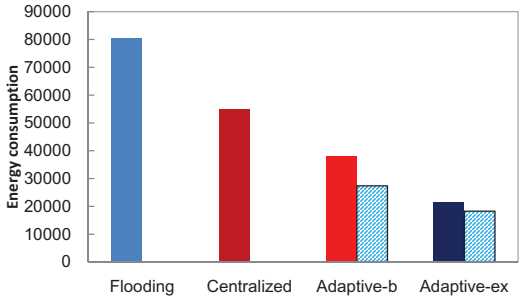


Fig. 8: Energy consumed in example 2.

then transformed to state S_2 through an intermediate state with 270 tasks generated in a time slot (270 is the median of 180 and 360). Then, it stayed in S_2 for another five time slots before returning to S_1 in reverse. This process went on infinitely. The process had a period of 17 time slots. Fig. 7 shows the energy consumed by different methods in one period. The bars at left for both adaptive methods indicate the energy consumed by the proposed methods. The bars at right for these two methods indicate the energy consumed in an ideal situation. Nodes in the ideal situation were assumed to be able to predict future NRSs without any mistakes and make the right choices in advance. Of course, this is not realistic.

As we can see in Fig. 7, both adaptive methods consumed less energy than the centralized and the basic flooding methods due to their adaptivity. The adaptive-b method consumed 77.06% of the energy of the centralized method and 70.36% of that of the flooding method. By benefiting from the extended flooding method, the adaptive-ex method performed better. It only consumed 64.98% of the energy of the centralized method and 59.33% of that of the flooding method. The energy consumed by both adaptive methods is close to the idealized consumption, i.e., adaptive-b (112.85%) and adaptive-ex (109.74%). The differences were due to prediction errors and the initial choice.

Two reasons caused the NRS transitions in example 2: (1) the number of tasks generated by nodes, and (2) different resource densities. There were two kinds of resources, A and B, in this example. Every node had five units of resource A. Only 60% of nodes had five units of resource B. In reality, resource A represented common resources like 3G. Resource B represented less popular resources like GPS or software applications. In state S_3, 100 tasks were generated for resource A while no tasks were generated for resource B. In another state, S_4, 50 tasks were generated for resource B while no tasks were generated for resource A. The network stayed in S_3 for 10 time slots. Then, it changed to state S_4 for another five time slots. This process went on infinitely. Fig. 8 shows the energy consumed by the different methods in one period.

Again, both the adaptive-b method (68.89% of the centralized method and 47.08% of the flooding method) and the adaptive-ex method (39.33% of the centralized method and 26.88% of the flooding method) consumed less energy than the centralized and basic flooding methods. The energy consumed by the adaptive-b method was 138.07% of that in the ideal situation. The energy consumed by the adaptive-ex method was 117.87% of that in the ideal situation. The differences were slightly larger than those in example 1 because there was no intermediate status in this example. Due to benefits from a smaller TTL value chosen in the flooding mode, the penalties for incorrect predictions were less expensive in the adaptive-ex method. Consequently, the energy consumed by the adaptive-ex method was closer to that of the ideal situation compared with the adaptive-b method.

The results in this subsection indicate that the proposed methods could automatically choose an energy-efficient discovery mode according to different NRSs.

6.4. Scalability

The basic parameters listed in Table 3 were used in the previous simulations. The scalabilities of the proposed methods were verified under different parameter settings and are discussed in this subsection.

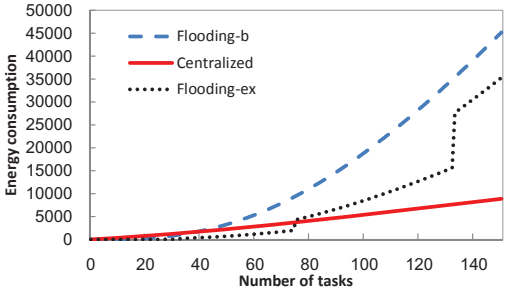
6.4.1. Relationship between number of resources and divisions in regions

The relationship between the number of resources each node had and the divisions in regions is first explained. If there were more resources in each node, with a fixed number of generated tasks, there was intuitively less chance for it to generate a resource request for available resources in other nodes. This meant little energy would be consumed in the flooding mode. However, the number of available resources in a node still changed even if the task was processed by the node itself. Energy was still consumed to maintain the resource directory accurate in the centralized mode. As the number of resources increased, the flooding mode was more preferable because there was no need to maintain a resource directory in it. This is proved in Fig. 9, where the range of F-R increases with the increasing number of resources in each node.

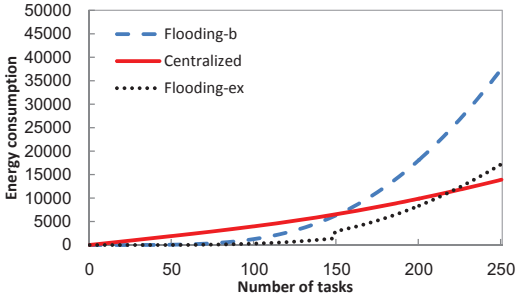
6.4.2. Node & resource densities

The performance of the proposed methods was verified under different node and resource densities instead of the default parameters in Table 3.

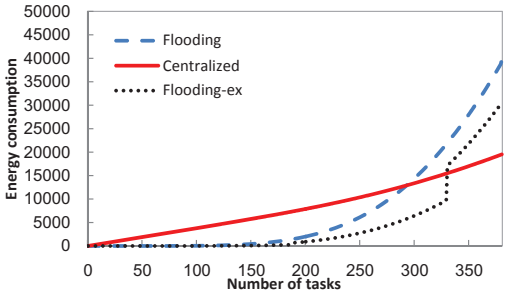
Fig. 10 plots the energy consumed by different methods under six different node densities (because the ranges of energy



(a) 2 units of resources per node



(b) 4 units of resources per node



(c) 6 units of resources per node

Fig. 9: Relationship between number of resources and divisions in regions.

consumed by different methods was too large, a 10 based log function was used as the y-axis). Every node in this simulation had five units of resources. A total of 250 tasks was generated by nodes in the area in each time slot. As the figure shows, the adaptive-b method always performed closest to the best choice for the centralized and basic flooding methods for different node densities. The adaptive-ex method consumed less energy than the others if the flooding mode was chosen because it benefitted from the extended flooding method. We can see that both adaptive methods chose the flooding mode when the number of nodes increased. This was due to the assumption that a fixed number of tasks (250) was generated in one time slot. The average number of tasks generated by each node decreased when the number of nodes increased under this assumption. Consequently, there was less chance for each node to ask the other nodes for resources. Therefore, the flooding mode was preferred.

The performance of the proposed methods with different resource densities (percentage of nodes that had the required re-

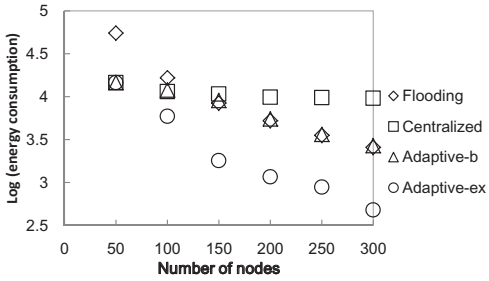


Fig. 10: Energy consumed under different node densities.

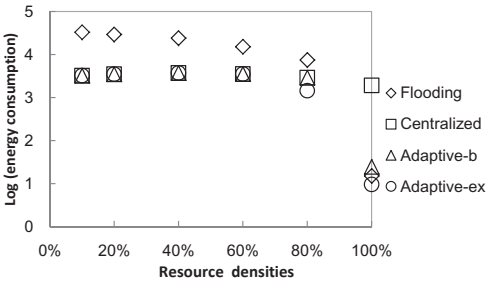


Fig. 11: Energy consumed under different resource densities.

sources) is plotted in Fig. 11 (a 10 based log function of energy consumption was used). A total of 500 units of resources was equally distributed in nodes that had the required resources in this simulation. 50 tasks were generated by nodes in the area in one time slot. As the figure shows, the adaptive-b method always performed closest to the best choice for the centralized and basic flooding methods for different resource densities. We can see that when the resource density was low, both adaptive-b and adaptive-ex methods chose the centralized mode. This is because many resource requests were generated by nodes without any resources. The centralized mode was more energy-efficient than the flooding mode in this situation. The emergence of this situation was due to our assumption that every node generated tasks with an equal probability.

6.4.3. Parameters for proposed heuristic algorithm

Finally, parameter settings for the proposed heuristic algorithm are discussed. The scenario for example 1 described in Subsection 6.3 was used for the simulations that are explained in this subsection.

Fig. 12 plots the energy consumed by different methods with different values of C_{thresh}. Energy consumed by the centralized and basic flooding methods has been presented for the sake of convenience, although it was not influenced by the parameters presented in this part. Both adaptive-b and adaptive-ex methods performed best when the C_{thresh} value was near 1. This was due to the fact that the centralized mode consumed about 65% – 85% of the energy consumed by the flooding mode in the first time slot of C-R in the simulation scenario[3]. As a result,

[3]The flooding mode consumed about 5% – 10% of the energy consumed by

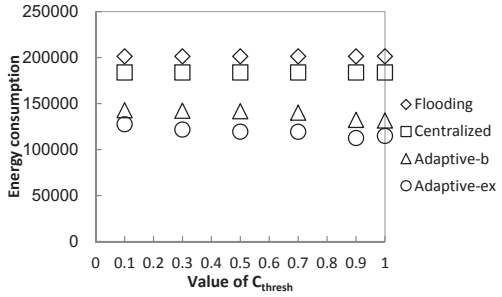


Fig. 12: Energy consumed with different values of C_{thresh}.

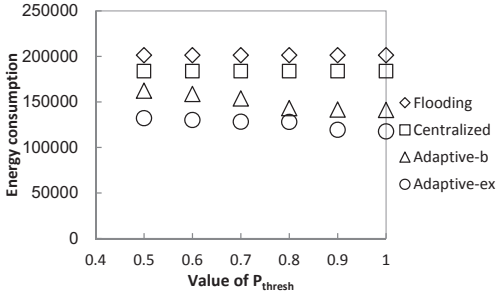


Fig. 13: Energy consumed with different values of P_{thresh}.

if the value of C_{thresh} was larger than 0.85, the proposed methods transformed to the centralized mode quickly to adapt to the transition of NRS. Otherwise, the transformation was slightly delayed by using the average value of the previous N_{trend} (2 by default) time slots. Therefore, the value of C_{thresh} should be a clear signal that the NRS stably entered into the region of F-R (or C-R). If the duration of NRS transitions is short, the value of N_{trend} should be kept small to adapt to transitions quickly. If the duration is long, the value of N_{trend} should become larger to filter out possible fluctuations during the process. The value of N_{trend} was assigned to 2 by default because the duration of transitions in the previous simulation scenarios were short (with one or no intermediate states).

The energy consumed by different methods with different P_{thresh} values is plotted in Fig. 13. Because the priority of P_{thresh} is lower than that of C_{thresh} (0.5 by default) according to the algorithm, values larger than 0.5 were considered. The energy consumption decreased along with the increasing value of P_{thresh}, since a small value of P_{thresh} prevented transformations even if substantial amounts of energy were saved. Consequently, the value of P_{thresh} should be near 1 for this kind of scenarios in which the transitions of NRS are regular. A smaller value of P_{thresh} combined with the ratio of N_{trans} to N_{slot} are helpful in preventing transformations for some extreme scenarios, e.g., when the NRS is in CR-R, the preferred mode changes frequently and irregularly due to the stochastic uncertainty of nodes that generate tasks.

the centralized mode in the first time slot of F-R. Consequently, it was rarely affected by different values of C_{thresh}.

As we can see, there is redundancy in the proposed algorithm, viz., (1) both N_{trend} and C_{thresh} control the sensitivity of transformations and (2) both the ratio of N_{trans} to N_{slot} and P_{thresh} prevent meaningless transformations. Redundancy is mainly retained for two reasons: (1) it keeps the proposed algorithm robust in extreme scenarios and (2) it reserves mechanisms that automatically optimize parameters for the proposed algorithm in future work.

Generally speaking, parameter settings for the algorithm are not universal and depend on different characteristics of networks. However, the effectiveness of the proposed method mainly resulted from transformations of discovery modes according to different NRSs. The values of parameters just control the degree of sensitivity and robustness for transformations, while they have limited impact on overall performance except for extremely impractical settings. This is the reason that we chose a common parameter setting rather than optimize parameters separately for different scenarios in this paper.

According to all the simulation results in this section, it can be concluded that: (1) When the NRSs only included a single region, the energy consumed by the adaptive-b method was always closest to the best choice for the centralized and basic flooding methods because of its adaptivity. (2) When the NRSs included both F-R and C-R, the adaptive-b method consumed less energy than either the centralized or flooding methods because of its adaptivity. Although the degree of reduced consumption depended on different NRS parameters, the energy consumed by the adaptive-b method was close to the ideal situation. (3) When the proposed method was provided with an improved version of components (adaptive-ex), energy consumption was further reduced while maintaining the previous advantages. (4) The previous conclusions were not sensitive to different NRSs. As a result, the proposed adaptive solution to resource discovery was energy-efficient in different network environments due to its adaptivity.

7. Related work

There is a great deal of related work that exists due to the importance of resource discovery. This work can be roughly divided into two categories: directory-aided and directory-less. Resource directories were used to facilitate resource discovery in directory-aided strategies [5–8, 11–13]. Two modes were defined for resource discovery in [5], i.e., service searching and service browsing. Service searching allowed a client to formulate a query containing the required attributes of the service. A client in the service browsing mode could send a generic query and obtain a list of all the services of a specific provider. However, it only supported one-hop discovery due to the limits of Bluetooth. A Chord based resource discovery method was introduced against the background of wireless mesh networks (WMNs) in [8]. It used location-awareness ID assignment and a cross-layer strategy to facilitate resource discovery. A centralized and homogeneous naming mechanism of global resource discovery for the Internet of Things (IoT) was presented in [11]. A context-aware service discovery framework based on

the virtual personal space (VPS) was introduced in [12]. A personal operating middleware (POM) in the framework was responsible for providing personalized response information. A framework of semantic service discovery for ubiquitous computing was proposed in [13]. Although different issues in service discovery and reciprocal work with current ontology languages were discussed, no theoretical models or prototype systems were mentioned by the authors. The [6] was the only work that took into consideration the combination of different discovery modes. However, their method always chose the centralized mode when it was available and did not take into account adaptivity to different network environments.

Flooding based methods were used in the directory-less strategies. However, the energy consumed by flooding exponentially increased along with the increasing number of resource requests. Several improved strategies like probability based [9] and location based [15] resource discovery methods have been proposed to solve this problem. A node that received a resource request that it was not able to fulfill forwarded it with a probability that decreased with the number of hops the request had already travelled in [9]. However, this method decreased the coverage aspect of RIA without a formal analysis of tradeoffs. Resource providers periodically sent resource advertisements along cross-shaped trajectories in [15]. A resource requester only sent a request along a path that intersected with any one of the trajectories. The intersecting node of the advertising and requesting trajectories answered the request. Because it only utilized relay nodes in four directions, the hit ratio (the accuracy aspect) of resource discovery was low in a sparse network. Crossing-layer strategies were adopted by [10] and [14]. The resource discovery protocols in these strategies were integrated with routing protocols. Although energy consumption was reduced, the compatibility of resource discovery methods was greatly limited. Different implementations were needed for different routing protocols. It should be mentioned that, improved centralized and flooding methods are easy to integrate into the proposed adaptive method when they become mature. Since they perform better than their basic versions, the energy efficiency of our adaptive method can also be improved. The simulation results from the adaptive-ex method proved this. [17] provided a comprehensive survey of published work in this area.

As a result, we found there was no existing work that was similar to our proposed method of adaptive resource discovery. However, the importance of adaptive resource discovery based on method (mode) transformations was also emphasized in [17].

8. Conclusions & future work

This paper presented an energy-efficient method of adaptive resource discovery against the background of MCC. According to different network environments, it transforms between centralized and flooding modes to save energy. An extension of the flooding method was introduced as an optional choice. A heuristic algorithm was provided to implement the proposed method. The effectiveness of the new approach was proved through extensive simulations.

Our work was only the first step toward utilizing the proposed adaptive strategy. As discussed in the previous sections, it could be integrated with other improved methods to replace their basic counterparts. It could also be used to optimize other metrics like response time and RIA. Mechanisms that automatically optimize parameters for the proposed algorithm are interesting. Finally, when node mobility is considered, nodes may not be distributed uniformly throughout the area. An appropriate model or heuristic algorithm still needs to be found to estimate energy consumption in the flooding mode.

References

[1] Guan L, Ke X, Song M, Song J. A survey of research on mobile cloud computing. In: Computer and Information Science (ICIS), 2011 IEEE/ACIS 10th International Conference on. IEEE; 2011, p. 387–92.

[2] Vanthournout K, Deconinck G, Belmans R. A taxonomy for resource discovery. Personal and Ubiquitous Computing 2005;9(2):81–9.

[3] Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM; 2012, p. 13–6.

[4] Shi C, Ammar M, Zegura E, Naik M. Computing in cirrus clouds: the challenge of intermittent connectivity. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM; 2012, p. 23–8.

[5] Bluetooth S. Specification of the bluetooth system, version 1.1. http://www bluetooth com 2001;.

[6] Guttman E, Veizades J. Service location protocol, version 2 1999;.

[7] Sailhan F, Issarny V. Scalable service discovery for manet. In: Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on. IEEE; 2005, p. 235–44.

[8] Canali C, Renda M, Santi P, Burresi S. Enabling efficient peer-to-peer resource sharing in wireless mesh networks. Mobile Computing, IEEE Transactions on 2010;9(3):333–47.

[9] Gao Z, Yang X, Ma T, Cai S. Ricffp: an efficient service discovery protocol for manets. Embedded and Ubiquitous Computing 2004;:786–95.

[10] Garcia-Macias J, Torres D. Service discovery in mobile ad-hoc networks: better at the network layer? In: Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on. IEEE; 2005, p. 452–7.

[11] Jara AJ, Lopez P, Fernandez D, Castillo JF, Zamora MA, Skarmeta AF. Mobile digcovery: A global service discovery for the internet of things. In: Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on. IEEE; 2013, p. 1325–30.

[12] Park KL, Yoon UH, Kim SD. Personalized service discovery in ubiquitous computing environments. Pervasive Computing, IEEE 2009;8(1):58–65.

[13] Wang RC, Chang YC, Chang RS. A semantic service discovery approach for ubiquitous computing. Journal of Intelligent Manufacturing 2009;20(3):327–35.

[14] Liang J, Chen J, Zhang T. An adaptive low-overhead resource discovery protocol for mobile ad-hoc networks. Wireless Networks 2011;17(2):437–52.

[15] Tchakarov JB, Vaidya NH. Efficient content location in wireless ad hoc networks. In: Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on. IEEE; 2004, p. 74–85.

[16] Vellore P, Gillard P, Venkatesan R. Performance analysis of bittorrent enabled ad hoc network routing. In: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly. ACM; 2009, p. 196–200.

[17] Ververidis C, Polyzos G. Service discovery for mobile ad hoc networks: a survey of issues and techniques. Communications Surveys & Tutorials, IEEE 2008;10(3):30–45.

[18] Vellore P, Gillard P, Venkatesan R. Probability distribution of multi-hop multipath connection in a random network. In: Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE. IEEE; 2009, p. 1–5.

[19] Ristanovic N, Le Boudec J, Chaintreau A, Erramilli V. Energy efficient offloading of 3g networks. In: Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on. IEEE; 2011, p. 202–11.